Universidad de Valladolid

ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA INFORMÁTICA

DEPARTAMENTO DE INFORMÁTICA

TESIS DOCTORAL

Fault Diagnosis of Hybrid Systems with Dynamic Bayesian Networks and Hybrid Possible Conflicts

Presentada por Noemí Moya Alonso para optar al grado de doctor por la Universidad de Valladolid

Dirigida por:
Dr. Carlos J. Alonso González
Dr. José Belarmino Pulido Junquera

Cuando la vida te presente razones para llorar,
demuéstrale que tienes mil y una razones para reír.

Acknowledgments

Here it is, my dissertation. It's me who is going to become a PhD, true, but it has been possible thanks to many people supporting me and working with me during all these years.

First of all I want to thank from the bottom of my heart my supervisors, Dr. Carlos Alonso and Dr. Belarmino Pulido for all the work they have devoted to guide me along those years. They have worked by my side, they have taught me how to face the challenges we have found in our way and they have prepared me to become a PhD. If I'm writing these lines, it is thank to all their work during the last four years.

I would also want to thank Dr. Gautam Biswas. Thank you for letting me work in ISIS for a period of time, this has been a very fruitful period for my training. I want to thank also Dr. Xenofon Koutsoukos for helping me improve my work with all the meetings and discussions.

Regarding my research stay in Vanderbilt University, I have to thank Kristy, Tonya, Jaclyn and Michelle for helping me with all the paper work, it would have been much more difficult without your help.

Daniel Mack, Stephanie and Jeremiah Weeden-Wright, Chetan Kulkarni, Joshua D. Carl, Daniel Balasubramanian, Sheila Kusnoor and all the guys I met in Nashville, thank you for your friendship. You had always a smile in your face to make my day nicer. Thank you guys!

Back in Spain, I would like to thank my colleagues from the Grupo de Sistemas Inteligentes (GSI) from the Universidad de Valladolid (Arancha, Aníbal, Isaac and Juanjo besides Carlos and Belarmino). I want to make a special mention to Aníbal Bregón, you have been working by my side during the last 4 year, you have been always there ready to answer all my questions, to talk or to share a coffee, thank you!

I would like to thank all the people in the Departamento de Informática from the Universidad de Valladolid who have helped me along this time.

I have to thank the Junta de Castilla y León and the European Social Fund (ESF) for the support they gave me through order EDU/1933/2008, as well as the Spanish MCI TIN2009-11326 grant.

Finally, I want to thank all the loving people in my life, my friends and family, who have been there day by day, no matter my mood. They have encouraged me during all those years. Your support has been very valuable. Muchas gracias a mis padres por apoyarme siempre, por creer en mí y por estar siempre a mi lado. Gracias!. Fran, you believe in myself more than I do and I want to thank you too.

Noemí Moya.
June 2013. Valladolid.

Resumen

Los sistemas híbridos son muy importantes en nuestra sociedad, están presentes en multitud de campos. Este tipo de sistemas pueden desarrollar una tarea de manera autónoma o pueden interaccionar con personas en su trabajo, de ahí la importancia vital de que funcionen en un estado nominal y seguro. La Diagnosis Basada en Modelos (MBD) es una rama de la diagnosis que basa sus decisiones en modelos. Esta tesis se centra en el campo de MBD con técnicas de Inteligencia Artificial, lo que se conoce como comunidad DX. Los sistemas híbridos con los que trabaja esta tesis tienen un comportamiento continuo que está controlado por eventos discretos, encargados de producir cambios en su modo de trabajo.

Actualmente, otros autores han realizado trabajos previos en el campo de la diagnosis de sistemas híbridos. La mayoría de las propuestas necesitan enumerar previamente los posibles modos de funcionamiento del sistema, incluyo aquellos que nunca lleguen a visitarse durante el funcionamiento del mismo. Como solución a este problema, algunos autores proponen la técnica de modelado de los Hybrid Bond Graphs (HBGs) como extensión a los Bond Graphs (BGs). Los HBGs no necesitan todos los modos de un sistema híbrido, se van construyendo en línea, según se actualiza el estado del sistema.

Los fallos que pueden aparecer en un sistema híbrido se pueden dividir en dos grandes grupos: 1) Fallos discretos, y 2) Fallos paramétricos. Los fallos discretos están relacionados con la naturaleza híbrida y discreta del sistema, mientras que los fallos paramétricos, también denominados continuos, están relacionados con los parámetros y sensores del sistema. Hasta ahora, los dos tipos de fallo no se han considerado de manera conjunta y unificada en la misma arquitectura de diagnosis.

El proceso de diagnosis se puede dividir en tres etapas principales: Detección, aislamiento e identificación de fallos. Los Posibles Conflictos (PCs) son una técnica de compilación utilizada en MBD para sistemas continuos. Los PCs proporcionan una descomposición de un sistema en subsistemas con redundancia analítica minimal lo que permite realizar una etapa de aislamiento más eficiente. Los PCs se pueden utilizar para las tareas de detección y aislamiento mediante la matriz de firmas (FSM). Esta matriz relaciona los diferentes parámetros del sistema (candidatos a fallo), con los PCs que los utilizan.

Los PCs proporcionan un modelo computacional que se puede implementar mediante diferentes herramientas. Esta tesis utiliza las Redes Bayesianas Dinámicas (DBNs) para implementar los PCs. Las DBNs tienen la ventaja adicional de que se pueden utilizar durante las tres etapas de diagnosis, incluida la identificación, completando así la aplicabilidad de los PCs. El principal problema que presentan las DBNs es la alta carga computacional que necesitan para realizar la inferencia, los algoritmos de inferencia aproximada, como los Filtros de Partículas (PF) solucionan este inconveniente en parte. Esta tesis proponen derivar DBNs minimales de PCs, lo que permitirá reducir la complejidad de las DBNs mejorando su eficiencia.

La principal contribución de esta tesis es el formalismo de los Posibles Conflictos Híbridos (HPCs). Los PCs se extenderán para sistemas híbridos con lo que la tesis propone su segunda contribución importante: una arquitectura de diagnosis para sistemas híbridos que integra tanto los fallos discretos como los paramétricos. Esta arquitectura está basada en los Posibles Conflictos Híbridos (HPCs) y las DBNs minimales derivadas de ellos. La arquitectura de diagnosis utiliza DBNs durante las tres etapas y además de para sistemas híbridos, también se ha presentado para sistemas continuos, junto con el método para derivar las DBNs minimales de los PCs de forma sistemática.

Finalmente, para probar cada una de las contribuciones realizadas se han utilizado diferentes sistemas en simulación del ámbito hidráulico (sistemas de tanques conectados), del ámbito eléctrico (circuito de orden doce), y un sistema híbrido que abarca varios dominios como es el ROS del entorno aeroespacial. Los resultados obtenidos en todos los casos han sido satisfactorios.

Abstract

Hybrid systems are very important in our society, we can find them in many engineering fields. They can develop a task by themselves or they can interact with people so they have to work in a nominal and safe state. Model-based Diagnosis (MBD) is a diagnosis branch that bases its decisions in models. This dissertation is placed in the MBD framework with Artificial Intelligence techniques, which is known as DX community. The kind of hybrid systems we focus on have a continuous behaviour commanded by discrete events.

There are several works already done in the diagnosis of hybrid systems field. Most of them need to pre-enumerate all the possible modes in the system even if they are never visited during the process. To solve that problem, some authors have presented the Hybrid Bond Graph (HBG) modeling technique, that is an extension of Bond Graphs. HBGs do not need to enumerate all the system modes, they are built as the system visits them at run time.

Regarding the faults that can appear in a hybrid system, they can be divided in two main groups: (1) Discrete faults, and (2) parametric or continuous faults. The discrete faults are related to the hybrid nature of the systems while the parametric or continuous faults appear as faults in the system parameters or in the sensors. Both types af faults have not been considered in a unified diagnosis architecture for hybrid systems.

The diagnosis process can be divided in three main stages: Fault Detection, Fault Isolation and Fault Identification. Computing the set of Possible Conflicts (PCs) is a compilation technique used in MBD of continuous systems. They provide a decomposition of a system in subsystems with minimal analytical redundancy that makes the isolation process more efficient. They can be used for fault detection and isolation tasks by means of the Fault Signature Matrix (FSM). The FSM is a matrix that relates the different parameters (fault candidates) in a system and the PCs where they are used.

PCs provide a computational model that can be implemented by means of different tools. In this dissertation, Dynamic Bayesian Networks (DBNs) have been chosen to implement the PCs and they can be used during the three diagnosis stages. DBNs main problem is the computational burden for inference tasks, this is partially solved using approximate inference like Particle Filter (PF) Algorithms. This dissertation proposes to derive minimal DBNs from PCs which reduces the complexity of the DBNs improving their performance.

The main contribution of this dissertation is the Hybrid Possible Conflicts (HPCs) formalism, PCs have been extended for hybrid systems. Related to that, the dissertation can provide the second important contribution: a diagnosis architecture for hybrid systems integrating discrete and parametric faults. This architecture is based on Hybrid Possible Conflicts and DBNs derived from them. The architecture uses DBNs along the three diagnosis stages. The diagnosis architecture has been also proposed for continuous systems fault diagnosis as well as a method to derive minimal DBNs from Possible Conflicts.

Finally, several simulation systems have been used to test each contribution. The systems come from different fields: Connected tanks (hydraulic), twelfth order electrical circuit (electric), and the ROS, a more complex hybrid system from the aerospace field. The results obtained have been satisfactory with all of them.

Parte I.

Diagnosis de fallos en Sistemas Híbridos con Redes Bayesianas Dinámicas y Posibles Conflictos Híbridos. Resúmenes en Español

Índice general

Capítulo 1
Introducción

1.1. Motivación

La detección, el aislamiento y la identificación de fallos (FDII), también conocido como diagnosis de fallos, es un reto vital en la sociedad. Cada día hay más robots y máquinas que realizan tareas de manera autónoma o semi autónoma, por lo tanto, en caso de que se produzca algún fallo deberá ser diagnosticado cuanto antes para evitar daños y pérdidas tanto físicas como económicas. Algunos de los fallos que podemos encontrar en los sistemas son errores de programación no solucionados previamente o la fatiga física de alguno de los componentes del propio sistema. La diagnosis de fallos permitirá detectar cuanto antes esos fallos, identificarlos y reestablecer un estado seguro en el sistema. El trabajo presentado en esta tesis está centrado en la detección y diagnosis de fallos (FDD) de sistemas dinámicos, cuyo estado interno depende de los estados previos y de las entradas al sistema.

Los sistemas dinámicos se pueden dividir en tres grupos: 1) Sistemas continuos, 2) Sistemas discretos, y 3) Sistemas híbridos. Los sistemas continuos tienen un comportamiento continuo y pueden modelarse mediante Ecuaciones Diferenciales Ordinarias, EDOs (ODEs si se usa el término en inglés). Por su parte, los sistemas discretos tienen un número finito de estados y se pueden modelar como sistemas de eventos discretos a través de autómatas. Los sistemas híbridos comparten características de los dos tipos previos, tienen comportamiento tanto discreto como continuo. En el tipo de sistemas con el que trabajaremos, eventos discretos modifican el comportamiento (continuo) del sistema. Este tipo de sistemas híbridos se encuentran en un amplio rango de aplicaciones industriales, como por ejemplo, sistemas mecánicos, circuitos eléctricos o sistemas embebidos.

Como ya se ha comentado anteriormente, hay varios tipos de sistemas híbridos, pero en este trabajo nos centramos en los sistemas continuos controlados por eventos discretos (muy comunes en sistemas embebidos). En este tipo de sistemas aparecen fallos que pueden ser clasificados en dos grandes grupos: 1) *Fallos continuos*, como una desviación en un sensor o un cambio en un parámetro del sistema, y 2) *Fallos discretos*, como un relé atascado o una válvula que cambia de posición sin recibir la orden necesaria.

Los sistemas híbridos suelen tener un comportamiento dinámico complejo y necesitan un proceso de diagnosis eficiente y fiable desde su puesta en marcha. Por eso, es necesario utilizar herramientas formales de análisis y diseño de diagnosis.

Hay varias ramas trabajando en el campo de la diagnosis de fallos: 1) Diagnosis basada en conocimiento, 2) Diagnosis guiada por datos y 3) Diagnosis basada en modelos [64, 65, 66] (DBM o MBD, según sus siglas en inglés). Según las necesidades de diagnosis expuestas previamente, la MBD es la única que permite cumplirlas y es en esta línea en la que se encuadra el trabajo de esta tesis.

Las técnicas de MBD utilizan modelos en el proceso de diagnosis [26, 27, 19, 54], comparan el comportamiento esperado, estimado por el modelo, con el comportamiento observado para detectar desviaciones.

La diagnosis online de sistemas híbridos no es una tarea fácil debido al comportamiento de este tipo de sistemas. La diagnosis basada en modelos online requiere un proceso de reconfiguración rápido y robusto durante un cambio de modo en el sistema, además debe permitir continuar monitorizando el comportamiento del sistema nominal durante estados transitorios.

Hay varias aproximaciones que proponen solucionar el problema de la diagnosis de sistemas híbridos [4, 7, 36, 18]. Normalmente, modelan los sistemas híbridos como autómatas híbridos, pre-enumerando de manera

explícita todos los posibles modos del sistema. Esta aproximación con autómatas híbridos funciona bien en sistemas pequeños, pero tiene un gran coste computacional para sistemas grandes enumerando modos que puede que nunca sean visitados. Otros métodos alternativos proponen modelar los sistemas híbridos mediante *Hybrid Bond Graphs* (HBGs) [44]. Esta técnica genera los modelos de cada modo de funcionamiento en tiempo de ejecución. Sin embargo, estas técnicas también presentan sus problemas, principalmente en el esfuerzo computacional necesario para la reconfiguración cuando se produce un cambio de modo. Este problema se hace más relevante en los modernos sistemas electrónicos, donde los cambios de modos se producen rápidamente. Roychoudhury et al. [60] proponen una forma de reconfiguración eficiente para sistemas híbridos modelados con HBGs. A pesar de ello, no existe una aproximación para la diagnosis de sistemas híbridos que permita realizar de manera unificada la detección, el aislamiento y la identificación de sistemas híbridos para fallos continuos (paramétricos) y discretos.

Otro problema habitual en el campo de la diagnosis de sistemas reales es que hay que gestionar la incertidumbre, tanto a nivel de modelo como de nivel de ruido en las señales que se miden del sistema. Los sistemas reales trabajan en entornos con ruido. Sus modelos también tienen cierto nivel de ruido, no sólo en las medidas, sino también en los propios parámetros del modelo. Este problema hace que se necesiten herramientas de modelado y simulación robustas. En este trabajo se pretende utilizar el mismo método para las tres etapas de diagnosis: detección, aislamiento e identificación.

Las Redes Bayesianas Dinámicas (DBNs, por sus siglas en inglés) [42, 57] se han elegido como la herramienta para modelar y simular los sistemas gracias a su habilidad de trabajar satisfactoriamente con ciertos niveles de ruido. Además, se pueden utilizar para detección, aislamiento e identificación de fallos, lo que permite realizar la diagnosis de fallos de manera unificada.

Por contra, las DBNs también tienen alguna desventaja. Realizar inferencia exacta requiere un alto coste computacional, pero puede resolverse utilizando métodos de inferencia aproximada, como el algoritmo de Filtro de Partículas (PF, por sus siglas en inglés). Incluso con métodos de inferencia aproximada, el esfuerzo computacional puede ser un problema. Otra gran desventaja de las DBNs es la dificultad de conseguir una convergencia precisa cuando se tienen muchos estados desconocidos.

La principal motivación de esta tesis consiste en mejorar los métodos recientes de diagnosis de sistemas híbridos mediante una solución basada en MBD que permita realizar FDII en entornos con ruido utilizando DBNs. Las DBNs se generarán a partir de subsistemas con redundancia, en lugar de modelar el sistema completo, para reducir la carga computacional necesaria. Las DBNs con PF como algoritmo de inferencia se utilizarán a lo largo de las tres etapas de diagnosis.

1.2. Principales hipótesis

Este trabajo presenta un método eficiente para realizar Detección, Aislamiento e Identificación de Fallos en sistemas continuos controlados por eventos discretos. La herramienta propuesta para modelar los sistemas son las Redes Bayesianas Dinámicas.

Incluso con algoritmos de inferencia aproximada, como el Filtro de Partículas, se puede tener un coste computacional elevado utilizando DBNs. El coste aumenta al aumentar el tamaño y la complejidad de la red. Esta tesis propone los Posibles Conflictos [51] como técnica de descomposición de modelos para analizar la estructura de la DBN del sistema completo y para generar DBNs minimales que modelen los subsistemas sobredeterminados minimales representados por los PCs. Las DBNs derivadas de los PCs tendrán menor número de nodos y arcos, de forma que serán menos complejas, característica deseable en las DBNs ya que conseguirá estimaciones más precisas del estado con una carga computacional menor.

Las DBNs se han utilizado también con sistemas híbridos, incluyendo nodos discretos que modelen la parte discreta de su comportamiento [20, 45], pero esto hace que las DBNs sean más complejas y su convergencia menos precisa.

Este trabajo propone modelar mediante DBNs el comportamiento continuo de los sistemas híbridos. Diferentes DBNs se utilizarán para diferentes modos de funcionamiento o configuraciones del sistema real. Los Posibles Conflictos, que ya se pueden generar a partir de modelos de Bond Graphs, se extenderán para sistemas híbridos, ya que son la herramienta a partir de la que se derivarán las DBNs. En nuestra primera aproximación, proponemos incluir los PCs en la metodología de HBGs [60] para generar los nuevos modelos de manera eficiente tras un cambio de modo.

La hipótesis principal de esta tesis enuncia que:

> Es posible integrar la detección, el aislamiento y la identificación de fallos para sistemas híbridos en entornos con ruido de una manera eficiente mediante la extensión de los Posibles Conflictos para sistemas híbridos y se usarán DBNs para realizar FDII.

La hipótesis principal puede descomponerse en las siguiente hipótesis particulares:

- Se necesita una arquitectura de diagnosis para sistemas híbridos que permita tratar tanto con fallos continuos (paramétricos) como discretos. La arquitectura debe soportar las tres etapas de diagnosis: detección, aislamiento e identificación de fallos, que no han sido consideradas siempre de manera integrada.
- Las Redes Bayesianas Dinámicas son una buena herramienta, junto con el Filtro de Partículas, para realizar las tareas de diagnosis en un marco común. También permiten trabajar con incertidumbre/ruido en el sistema y en el modelo.
- La complejidad de la monitorización, el aislamiento y la identificación de fallos en sistemas híbridos reales se puede reducir extendiendo los PCs a sistemas híbridos, lo que permitirá generar de manera eficiente los PCs para la configuración que tenga el sistema en cada momento.
- El coste computacional de la inferencia con Filtro de Partículas en las DBNs se puede reducir utilizando los PCs para factorizar la DBN del sistema completo en DBNs minimales. Esto también permite evitar el uso de nodos discretos en las DBNs.

1.3. Objetivos

Las hipótesis fijadas en la sección previa conducen a los objetivos que se presentan a continuación. Además aparece otro objetivo que permitirá validar las técnicas desarrolladas utilizando un sistema híbrido real.

1. Factorizar de manera eficiente Redes Bayesianas Dinámicas mediante Posibles Conflictos y aplicarlas a la diagnosis de fallos de sistemas continuos.
2. Desarrollar la teoría de los Posibles Conflictos Híbridos (HPCs).
3. Definir cómo realizar la detección y el aislamiento de fallos con los HPCs.
4. Extender el método para derivar DBNs de PCs para sistemas híbridos.
5. Construir una arquitectura de diagnosis para sistemas híbridos basada en las DBNs derivadas de los HPCs. La arquitectura debe incluir fallos tanto continuos (paramétricos), como discretos, y permitirá realizar detección, aislamiento e identificación.
6. La arquitectura de diagnosis de sistemas híbridos debe probarse finalmente con un sistema híbrido real del ambito aeroespacial.

Durante la tesis también se utilizarán ejemplos de casos de estudio más simples para clarificar definiciones y propuestas.

Los objetivos se han propuesto en orden para diseñar la arquitectura de diagnosis unificada que permitirá confirmar la principal hipótesis.

Capítulo 2
Estado del Arte

La diagnosis de sistemas físicos se aborda desde diferentes aproximaciones: sistemas expertos, aprendizaje o diagnosis basada en modelos entre otras [64, 65, 66]. El trabajo de esta tesis se centra en la diagnosis basada en modelos y dentro de ésta en la comunidad DX, que utiliza técnicas de Inteligencia Artificial para la diagnosis de fallos.

De manera general, la diagnosis se puede definir como el proceso de localizar la causa de una desviación en el comportamiento esperado de un sistema. Esta definición se puede aplicar a la diagnosis de sistemas físicos. Estos sistemas se encuentran frecuentemente en nuestro entorno, incluso controlan tareas de nuestra vida diaria, por lo que es muy importante que su funcionamiento sea el esperado. Cualquier sistema puede sufrir algún tipo de fallo, pero debe detectarse y corregirse antes de que suponga un riesgo para su entorno.

Según Console [16] "la diagnosis es la tarea en la que dado un sistema y un conjunto de observaciones correspondientes a un modo de funcionamiento no correcto, determina qué está mal en el sistema para reestablecer su comportamiento normal."

La diagnosis es un proceso iterativo con 3 fases:

- **Detección de fallos:** Decidir si hay un fallo o no. También proporciona el instante de tiempo en que se produjo el fallo.
- **Aislamiento de fallos:** Localizar el componente que falla.
- **Identificación de fallos:** Averiguar el tipo de fallo que se ha producido y su magnitud.

2.1. Diagnosis Basada en Modelos (MBD)

La diagnosis basada en modelos [26, 27, 19, 54] compara el comportamiento observado de un sistema con el comportamiento esperado que se obtiene de un modelo del mismo (Ver Figura 2.1)

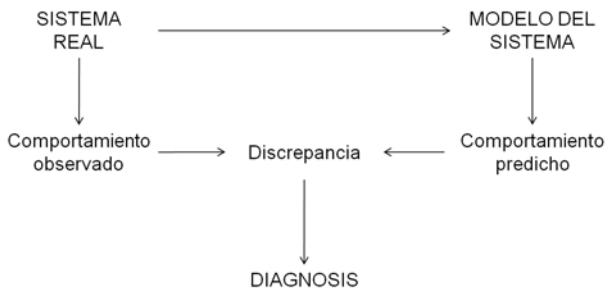


Figura 2.1: Diagnosis Basada en Modelos.

Como cualquier aproximación, la diagnosis basada en modelos tiene sus ventajas y sus inconvenientes. Por un lado, es independiente del dispositivo y de la experiencia. También puede usarse en escenarios con

multiples fallos y permite crear bibliotecas de modelos para poder reutilizarlos. Por otro lado, su principal desventaja es la necesidad de construir modelos precisos para poder realizar una diagnosis adecuada.

El elemento principal en la MBD es el modelo. Según B. Kuipers [34] "un modelo es una (pequeña) descripción finita de una realidad compleja infinita construida con la finalidad de responder unas preguntas concretas."

Tradicionalmente, MBD se ha dividido en dos grandes comunidades en función de las técnicas empleadas para realizar la diagnosis: 1) Comunidad FDI (Sistemas dinámicos e ingeniería de control) [49, 29, 26, 46], y 2) Comunidad DX (Inteligencia Artificial) [54, 27]. Más recientemente ha aparecido una tercera comunidad, BRIDGE [9, 17] que busca un marco común para combinar resultados y técnicas de las dos primeras comunidades (FDI y DX). Las dos primeras aproximaciones a la diagnosis basada en modelos de sistemas continuos utilizan diferentes tipos de modelos y suposiciones sobre la robustez de la solución propuesta en relación a perturbaciones, errores de modelado y ruido.

2.1.1. Métodos probabilísticos en MBD

Dentro de la diagnosis basada en modelos también hay métodos probabilísticos [27, 43]. Algunos de los métodos probabilísticos ya se han utilizado en algoritmos de diagnosis [33, 42, 35, 57, 59, 56, 47]. Las teorías probabilísticas permiten trabajar con incertidumbre relacionada con las hipótesis y las medidas.

El principal problema de la diagnosis con métodos probabilísticos consiste en decidir qué fallo ha ocurrido según las medidas u observaciones disponibles. El razonamiento que se emplea es similar al que sigue el cerebro humano: Un experto puede saber qué síntomas aparecen con qué probabilidad cuando ocurre un fallo en un sistema.

El Teorema de Bayes es la herramienta principal para hacer este razonamiento, permite diagnosticar fallos con incertidumbre en los síntomas y en los propios fallos [61]. Suponiendo que $Sintoma$ y $Fallo$ son dos variables aleatorias, la probabilidad a posteriori de $Fallo$ dado $Sintoma$ ($P(Fallo|Sintoma)$) se puede obtener usando la información causal y las probabilidades a priori: $P(Sintoma|Fallo)$, $P(Sintoma)$ y $P(Fallo)$, respectivamente.

$$P(Fallo|Sintoma) = \frac{P(Sintoma|Fallo)P(Fallo)}{P(Sintoma)}$$

Existen modelos gráficos, como las Redes Bayesianas (BNs) y las Redes Bayesianas Dinámicas (DBNs) que modelan la incertidumbre de manera explícita. La principal ventaja que tienen estos dos modelos es que representan de manera gráfica una factorización eficiente de la distribución conjunta de las variables del modelo, lo que hace posible realizar la diagnosis. Esto es posible porque las redes contienen no solo las dependencias entre variables sino también las independencias.

Las BNs asumen estados estáticos en el sistema: no pueden modelar estados dinámicos ni transiciones entre diferentes estados. Las DBNs utilizan información temporal y permiten modelar estados dinámicos y transiciones entre distintos estados del sistema. Este trabajo utiliza DBNs para aplicar técnicas de MBD a sistemas dinámicos en entornos con incertidumbre.

2.2. MBD de Sistemas Híbridos

El comportamiento de los sistemas híbridos está compuesto por comportamiento continuo y eventos discretos. Estos sistemas tienen distintos modos de funcionamiento o configuraciones en las que funcionan y dentro de cada uno de ellos el comportamiento es continuo. Los eventos discretos se encargan de controlar el cambio de modo de funcionamiento. Sistemas híbridos de este tipo se encuentran en nuestra vida diaria, por ejemplo el sistema de ABS de los frenos del coche o el sistema de llenado de combustible de un avión.

La diagnosis basada en modelos online necesita procesos de reconfiguración rápidos y robustos tras un cambio de modo. Un evento discreto conlleva un cambio de modo, es necesario construir el modelo de la nueva configuración y continuar monitorizando el sistema durante el periodo transitorio sin producir falsas alarmas de fallo (falsos positivos).

La diagnosis de sistemas híbridos se ha afrontado desde las comunidades DX [37, 43] y FDI [15, 36] y se enfrenta a grandes retos, las propuestas de ambas comunidades proponen en cierto modo estimar o modelar

el conjunto de posibles estados del sistema. Además no existe una propuesta que permita tratar de manera conjunta tanto fallos continuos o paramétricos, como posibles fallos discretos (cambios de modo no deseados o cambios de modo no ejecutados).

Este trabajo propone una aproximación para tratar con los problemas de la diagnosis de sistemas híbridos de manera eficiente. En ella no es necesario realizar una pre-enumeración de los modos del sistema y se tratarán de manera uniforme y conjunta los fallos tanto discretos como continuos.

Capítulo 3

Diagnosis basada en consistencia de sistemas continuos mediante Redes Bayesianas Dinámicas Minimales.

3.1. Posibles Conflictos (PCs)

Los Posibles Conflictos (PCs) son un técnica de compilación que permite realizar diagnosis basada en consistencia (CBD) de sistemas continuos. Esta técnica se encuadra en la comunidad DX y se ha demostrado que los PCs son equivalentes a otras técnicas de compilación de la comunidad FDI como las Relaciones con Redundancia Analítica minimales (ARRs minimales, por sus siglas en inglés) o los conjuntos Estructurales Sobredeterminados Minimales (MSOs, según sus siglas en inglés) [5].

Los PCs proporcionan el modelo computacional de un residuo que se puede implementar como un modelo de simulación ejecutable [50], como observadores de estados [52], o incluso como Redes Neuronales [53]. Por otro lado están las Redes Bayesianas Dinámicas (DBNs) como técnica de modelado probabilística para sistemas dinámicos. En esta tesis se propone construir DBNs minimales a partir del modelo computacional de los PCs.

Blanke et al. [10] y Staroswiecki [63] proporcionan definiciones de observabilidad estructural en base a las ecuaciones y variables que definen un modelo. De manera general, un sistema es estructuralmente observable si se puede calcular su estado a partir de las entradas y las observaciones disponibles. Los Posibles Conflictos son estructuralmente observables por definición [39] lo que permite que su modelo computacional se pueda implementar satisfactoriamente como DBNs.

La detección y el aislamiento de fallos de sistemas continuos se puede realizar mediante PCs de manera eficiente. Como se ha dicho anteriormente, cada PC se corresponde con un residuo[1]. Un PC se confirma como conflicto cuando se activa, esto es, cuando el residuo correspondiente deja de ser cero. La detección de un fallo en el sistema consiste en la activación de uno o más PCs. A continuación, el proceso de aislamiento se encargará de obtener el conjunto de corte minimal de los conflictos confirmados y los parámetros del sistema que aparecen en ellos obteniendo el conjunto de candidatos de fallo. Este mismo proceso se puede realizar sea cual sea la implementación que se está usando de los PCs.

3.2. Caso de estudio. Sistema de 3 tanques.

Como caso de estudio sencillo en este capítulo se ha utilizado el sistema de 3 tanques que se presenta en la Figura 3.1.

Este sistema tiene un flujo de entrada en el tanque 1 (f_1) y cada uno de los 3 tanques tiene un flujo de salida en su parte inferior. La Figura 3.1 muestra los parámetros del sistema, además de la disposición de los tanques. Las medidas disponibles en el sistema son: El flujo de salida del tanque 1 (F_1), el flujo entre los tanques 1 y 2 (F_{12}) y el flujo de salida del tanque 3 (F_3).

[1] El residuo se define como la diferencia entre una observación en el sistema y el valor predicho por el modelo.

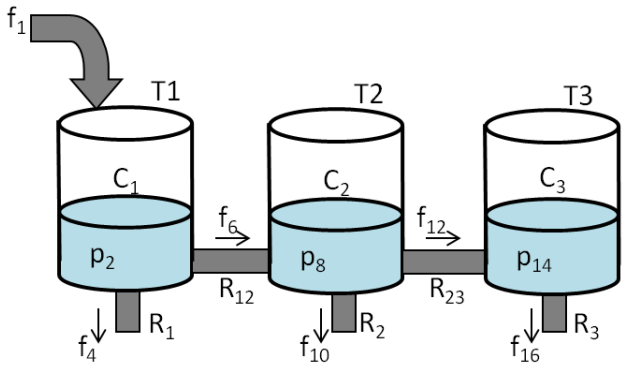


Figura 3.1: Sistema de 3 tanques. Según el esquema, las observaciones del sistema F_1, F_{12} y F_3 se corresponden con f_4, f_6 y f_{16} respectivamente.

El sistema de 3 tanques tiene 3 PCs, uno asociado a cada medida, de forma que el nodo discrepancia (variable estimada por el PC) en cada uno de ellos se corresponde con una de las medidas del sistema. Las Figuras 3.2, 3.3 y 3.4 muestran los MEM asociados a cada uno de los PCs.

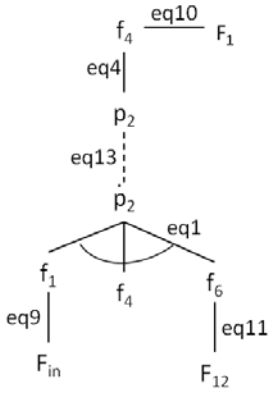


Figura 3.2: PC1 del sistema de 3 tanques.

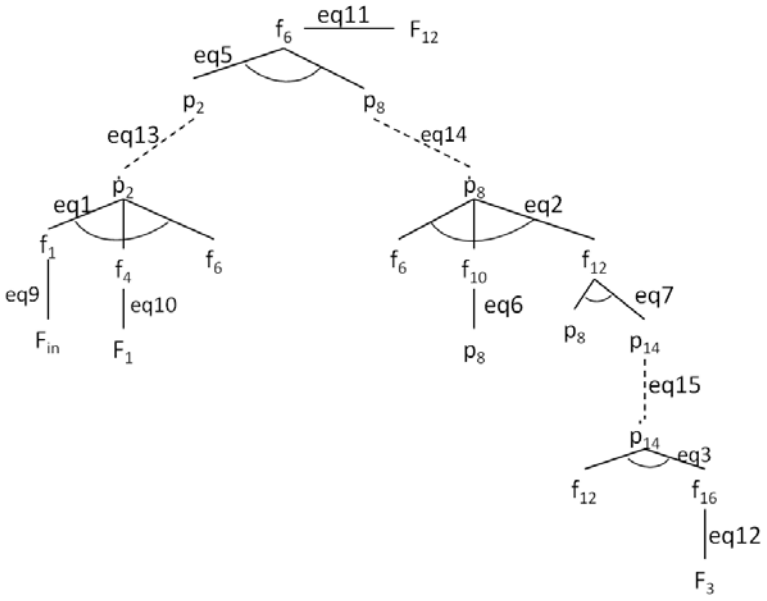


Figura 3.3: PC2 del sistema de 3 tanques.

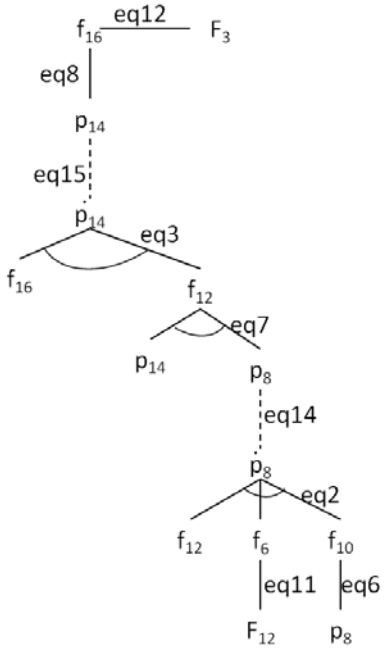


Figura 3.4: PC3 del sistema de 3 tanques.

3.3. Diagnosis de Sistemas Continuos mediante DBNs minimales

Las Redes Bayesianas Dinámicas (DBNs), como se ha indicado en el capítulo anterior, son modelos probabilísticos que permiten representar sistemas dinámicos. En este trabajo se propone derivar las DBNs de los PCs obteniendo DBNs que modelan subconjuntos minimales con redundancia analítica que son por definición estructuralmente observables.

Uno de los principales inconvenientes de las DBNs es el alto coste computacional que tiene el proceso de inferencia y la falta de precisión en la convergencia cuantos más estados desconocidos tiene. En este caso, al obtener las DBNs de los PCs en vez de usar el sistema completo, éstas son menos complejas, lo que reduce también la complejidad de la inferencia; y además tiene menos estados desconocidos que el sistema completo.

La Figura 3.5 presenta la DBN que modela el sistema de 3 tanques de presentado en la sección anterior.

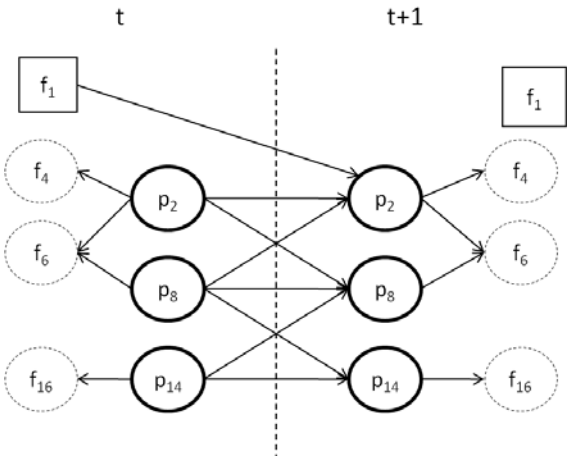


Figura 3.5: DBN del sistema de 3 tanques.

Este trabajo propone un método para derivar de manera eficiente las DBNs a partir de los PCs de un sistema continuo. Esa es una contribución importante del trabajo y en la que se apoyan la mayoría de las demás contribuciones.

Para modelar una DBN es necesario definir tanto su modelo de transición de estados, que define cómo evoluciona el estado de la DBN, como su modelo observacional, que se encarga de definir la relación entre el estado de la DBN y sus observaciones. Estos dos modelos se encuentran en los PCs en las relaciones integrales y en las relaciones algebráicas, respectivamente. El Capítulo 3 presenta el proceso que se debe seguir para obtener cada uno de los dos modelos a partir de un PC.

La Figura 3.6 muestra la DBN asociada al PC1 del sistema de 3 tanques, como se puede ver, su complejidad es mucho menor que la que tiene la DBN que modela el sistema completo (ver Figura 3.5).

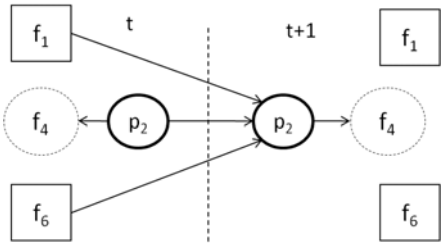


Figura 3.6: DBN minimal derivada del PC1 del sistema de 3 tanques.

Las DBNs también se pueden usar para identificar fallos en parámetros del sistema. En este trabajo se ha seguido la aproximación presentada por Roychoudhury et al. [57], la propuesta consiste en añadir a la DBN inicial, denominada también nominal al emplearse para modelar el sistema en el mismo comportamiento, un

nodo adicional que modela el parámetro en el que se quiere estimar el fallo. El nuevo nodo se relaciona con los nodos iniciales de la DBN según la dependencia o independencia que tenga con las respectivas variables del modelo. La Figura 3.7 muestra un ejemplo de la DBN minimal del PC1 del sistema de 3 tanques utilizada para identificar un fallo en el parámetro C1 (capacidad del tanque 1).

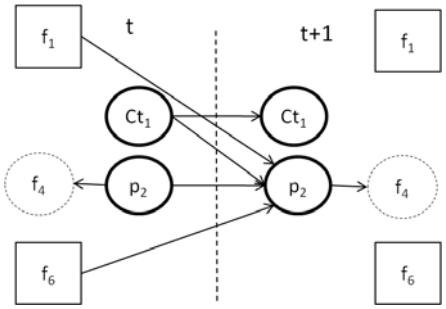


Figura 3.7: DBN minimal derivada del PC1 del sistema de 3 tanques para identificar un fallo en la capacidad del tanque 1.

Las DBNs minimales se han integrado en una arquitectura de diagnosis presentada en el mismo capítulo, cuyo esquema se muestra en la Figura 3.8.

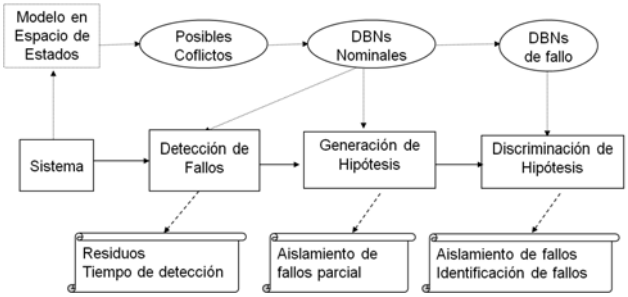


Figura 3.8: Arquitectura de diagnosis que integra DBNs y PCs.

La arquitectura de diagnosis que integra las DBNs minimales permite abordar la detección, el aislamiento y la identificación de los fallos de manera uniforme con la misma herramienta, las propias DBNs minimales. El proceso de aislamiento se realiza siguiendo el proceso que se seguiría con los PCs. En esta arquitectura se refleja otra de las ventajas que tiene el uso de DBNs, también pueden utilizarse para la identificación de fallos, como ya se ha indicado anteriormente. Esta arquitectura constituye la segunda contribución de la tesis.

El proceso de diagnosis que se sigue con la arquitectura propuesta se resume a continuación. Inicialmente, las DBNs minimales realizan el seguimiento o monitorización del sistema de forma que cuando se detecta un fallo, porque el residuo[2] generado por una de las DBNs minimales es distinto de cero se desencadena el proceso de diagnosis. Mediante una matriz que relaciona las DBNs minimales y los parámetros del sistema que aparecen en cada una de ellas (Matriz de firmas) se realiza el proceso de aislamiento construyendo los conjuntos de candidatos minimales. Una vez no se esperan más detecciones o ha pasado un tiempo suficiente desde la última detección activada se procede a realizar la etapa de identificación. Durante la identificación se construye una DBN por cada candidato de fallo y se simulan hasta que todas son descartadas porque no convergen sus residuos (no vuelven a cero) salvo una, que sí converge y además de confirmar el candidato como fallo, proporciona una estimación del mismo.

Al final del capítulo se presenta el estudio realizado sobre un sistema eléctrico de orden doce en el que se ha probado la arquitectura de diagnosis propuesta con DBNs minimales. Al analizar los resultados se observó que las DBNs minimales conseguían un mejor comportamiento que la DBN del sistema completo en detección y

[2]residuo = salida observada en el sistema - salida predicha por la DBN.

aislamiento de fallos, pero en la etapa de identificación el comportamiento de las DBNs minimales no siempre era tan preciso o mejor que el comportamiento obtenido por la DBN que modela el sistema completo. Se sabe que las DBNs obtienen un mejor o peor comportamiento en función de su complejidad y del número de estados desconocidos. Basándonos en esta idea, decidimos desarrollar un método que mejorase la etapa de identificación cuando fuera necesario realizando la fusión de una o más DBNs minimales. El primer requisito para fusionar dos DBNs minimales es que compartan variables de estado, además de entradas y/o medidas, de forma que la proporción de variables de estado frente a observaciones mejore. La Figura 3.9 muestra un ejemplo de una DBN creada a partir de la fusión de dos DBNs minimales.

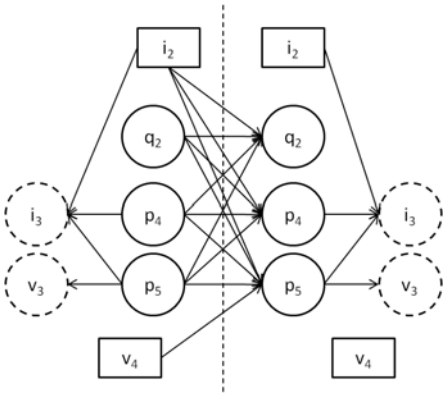


Figura 3.9: DBN creada al fusionar dos DBNs minimales del sistema eléctrico de orden 12.

El capítulo presenta el proceso de fusión en detalle y los resultados que se han obtenido con el sistema eléctrico de orden doce probando fallos en distintos parámetros y con diversas magnitudes.

Capítulo 4

Diagnosis de Sistemas Híbridos con Posibles Conflictos. Aproximación mediante HBG-PCs

4.1. Motivación

Los sistemas híbridos se encuentran en nuestra sociedad de manera muy frecuente. Como ya se ha comentado en capítulos anteriores, es de vital importancia que los sistemas trabajen dentro de su funcionamiento nominal (sin fallo) y que en caso de producirse un fallo, el funcionamiento correcto se reestablezca lo antes posible.

Los sistemas híbridos en los que centramos este trabajo tienen un comportamiento continuo controlado por eventos discretos. Estos sistemas tienen varios modos de trabajo, detro de los cuales su comportamiento es continuo, y reciben eventos discretos que desencadenan el cambio de modo de trabajo o funcionamiento.

En la comunidad DX existen trabajos sobre la diagnosis de sistemas híbridos: Unos se basan en modelado híbrido [37, 43], otros en la estimación de estados híbridos [28, 55] y otros combinan el seguimiento online del estado y la evaluación de residuos [8, 7]. Al analizarlos, todos presentan al menos uno de los siguientes problemas:

- Necesitan pre-enumerar todas las posibles configuraciones o modos de funcionamiento del sistema. Además hay que construir los modelos para todos ellos.
- Necesitan determinar el modo de trabajo actual, tanto la configuración de los actuadores como el comportamiento continuo.

Algunos autores proponen los Hybrid Bond Graphs (HBGs) [38] como técnica de modelado para sistemas híbridos. La principal ventaja de esta técnica es que no necesita pre-enumerar los modos de funcionamiento para poder monitorizar el sistema. Dentro de los HBGs hay dos aproximaciones: 1) Utilizar elementos de unión con causalidad fija para modelar los actuadores cambiando los valores de los parámetros [23, 11, 21, 24], y 2) Utilizar elementos de unión con cambio de causalidad cuando se produce un cambio en un actuador (esto mantiene constantes los valores de los parámetros del sistema) [44]. La propuesta en este trabajo se basa en la segunda aproximación.

En los sistemas híbridos se pueden tener dos tipos de fallos:

- **Fallos discretos:** Están relacionados con los eventos discretos que controlan el sistema. Estos fallos incluyen un cambio en un actuador no deseado o un actuador que no responde al cambio requerido.
- **Fallos continuos (paramétricos):** Son los fallos relacionados con el comportamiento continuo del sistema. Al igual que los fallos considerados en los sistemas continuos, en esta tesis nos centramos en los que afectan a los parámetros del sistema.

En este capítulo se presenta la ampliación de los PCs para sistemas híbridos, tanto su definición como su uso para detección y aislamiento de fallos (discretos y continuos). Además se propone una arquitectura de

diagnosis para sistemas híbridos que permite integrar tanto los fallos discretos como los continuos (paramétricos) de manera unificada. La base de la arquitectura son los Posibles Conflictos para sistemas híbridos y las DBNs derivadas de ellos. Las principales suposiciones sobre las que se asienta nuestra propuesta son:

- La configuración o modo de funcionamiento del sistema antes de que suceda el fallo es conocida.
- Los sistemas híbridos que se pueden diagnosticar con esta propuesta tienen sólo actuadores ON/OFF, es decir, que conectan o desconectan subsistemas, pero no conectan caminos alternativos.
- El modelo HBG del sistema completo tiene asignación causal global válida considerando todos los actuadores activos (ON).

Existen sistemas híbridos con actuadores multiposición, que conectan uno de varios caminos alternativos en cada una de sus posiciones. Para permitir una mayor aplicabilidad de la arquitectura propuesta se presenta un método que permite modelar ese tipo de actuadores multiposición mediante switches ON/OFF.

4.2. Posibles Conflictos Híbridos (HPCs)

Como se ha comentado en la introducción de este capítulo, los Posibles Conflictos se han extendido para utilizarlos, no sólo en sistemas continuos, sino también en sistemas híbridos, sistemas que tienen un comportamiento continuo controlado por eventos discretos.

La ténica de modelado Hybrid Bond Graphs (HBG) es una ampliación de los BGs [31]. Los HBG añaden un tipo de unión particular, denominado switching junction que está controlada por un autómata, cuando su estado es ON, la unión se comporta como una unión 1- ó 0- regular, en el caso de encontrarse en estado OFF, la unión se comporta como una fuente de flujo o esfuerzo cero, respectivamente.

Una de las principales ventajas de los HBGs, que se ha explicado anteriormente, es que permiten modelar los sistemas híbridos sin necesidad de conocer, ni enumerar inicialmente sus modos de funcionamiento, éstos se van construyendo a medida que se van visitando.

Los HPCs se han definido en base a HBGs, de ahí que también se identifiquen como HBG-PCS. En el Capítulo 4 se presenta el marco completo de definiciones y propiedades que se cumplen en él y sobre los que se apoya la proposición principal de esta aproximación:

> El conjunto de Posibles Conflictos Híbridos con k switching junctions OFF se puede derivar del conjunto de Posibles Conflictos Híbridos con todas las switching junctions ON.

En el Capítulo 4 se prueba que esa afirmación es cierta, así como la afirmación en el sentido contrario: Los PCs que se generen a partir de una configuración dada al activar una switching junction que estuviera OFF pertenecen a alguno de los HBG-PCs obtenidos con todas las switching junctions ON.

El método que se propone para derivar de manera eficiente los HBG-PCs de un sistema híbrido considera que todas las switching junctions están a ON y a partir de él deriva el TCG y a continuación los HBG-PCs [12].

4.2.1. Caso de estudio. Sistems de tanques

En el Capítulo 4 se utiliza el sistema híbrido de tanques que se muestra en la Figura 4.1 para ilustrar los conceptos que se van presentando. La Figura 4.2 presenta el modelo BG del sistema.

El sistema de 4 tanques híbrido tiene un flujo de entrada, que puede ser desactivado del tanque 1 o del tanque 3 pero nunca de los dos al mismo tiempo, la fuente de flujo es común para los dos tanques. Además, los tanques 2 y 4 están conectados con los tanques 1 y 3 respectivamente, mediante una tubería situada a una altura h de la base de los tanques. El sistema presenta 4 actuadores ON/OFF: SW_1, SW_2, SW_3 y SW_4. SW_1 y SW_3 son actuadores comandados, reciben una orden para activarse o desactivarse, mientras que SW_2 y SW_4 son actuadores autónomos, la transición de un estado a otro depende de una variable interna del sistema, en este caso concreto la altura del líquido del tanque 1 o 3 respectivamente, ya que cuando esa altura alcane el valor h al que está situada la tubería, el líquido empezará a pasar al tanque contiguo. El sistema tiene 4 medidas, que se correcponden con las presiones en cada uno de los cuatro tanques.

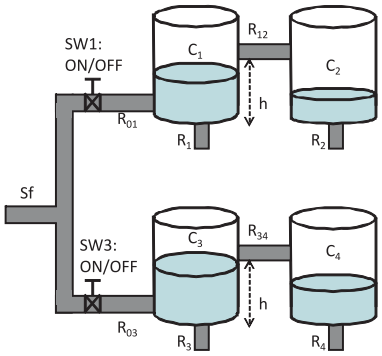


Figura 4.1: Esquema del sistema de 4 tanques híbrido.

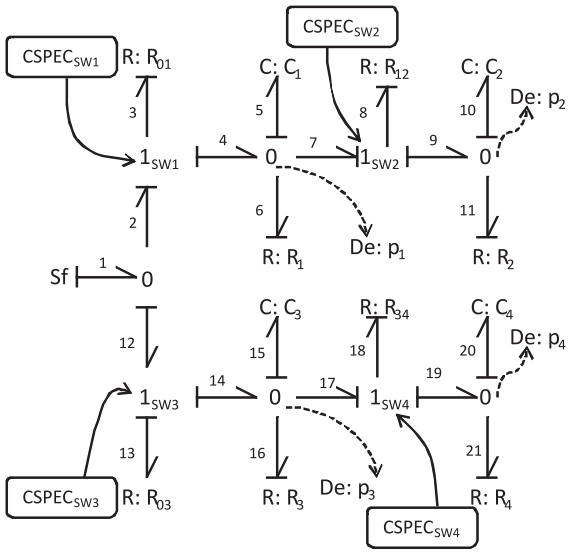


Figura 4.2: Modelo Bond Graph del sistema de 4 tanques híbrido.

Según la propuesta del Capítulo 4, los HPCs se obtienen configurando todas las switching junctions del HBG a ON y derivando el modelo TCG, para a obtener de él los PCs. Ese proceso se ha llevado a cabo con este sistema y la Figura 4.3 presenta el TCG del sistema de 4 tanques híbrido.

Aplicando el proceso explicado en el capítulo, los HBG-PCs obtenidos para el sistema son 4 y se muestran en la Figura 4.4.

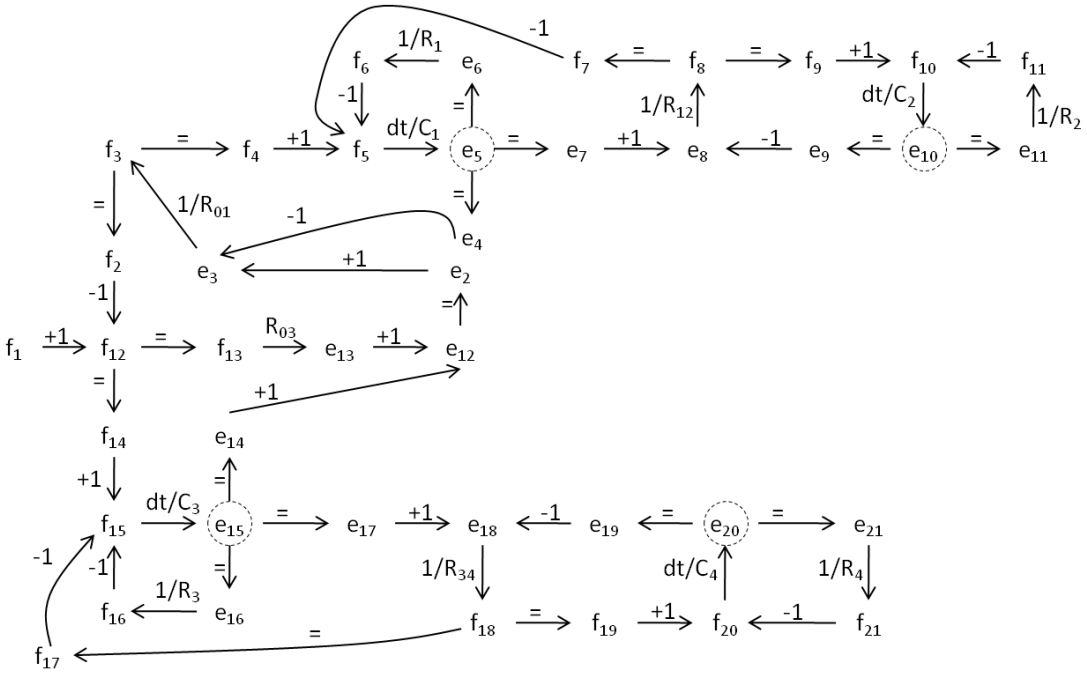


Figura 4.3: Temporal Causal Graph del sistema de 4 tanques híbrido.

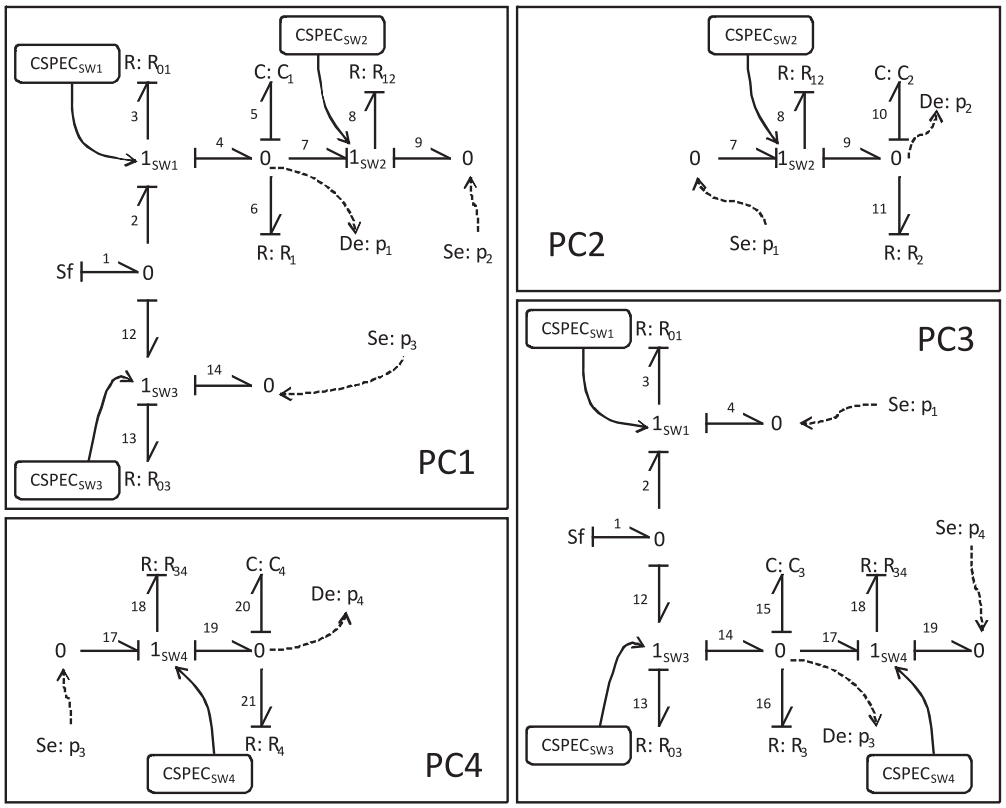


Figura 4.4: HBG-PCs del sistema de 4 tanques híbrido.

4.3. Arquitectura de Diagnosis Unificada para Sistemas Híbridos

El Capítulo 4 también presenta la arquitectura de diagnosis que unifica el tratamiento de fallos discretos y continuos para sistemas híbridos [40]. Inicialmente se define la arquitectura para realizar detección y aislamiento de estos dos tipos de fallos de manera conjunta mediante HBG-PCs. La detección se basa en el proceso de activación de PCs descrito para los sistemas continuos, una vez se obtiene el conjunto de candidatos de fallo, se consideran candidatos preferidos los fallos discretos. Los candidatos paramétricos no se tratan hasta que todos los discretos han sido rechazados.

Como puede extraerse del párrafo anterior, el proceso de diagnosis es muy similar al que se presentó en el caso de sistemas continuos, pero en este caso se está tratando con dos tipos de fallos diferentes: Discretos y paramétricos. En el caso de los fallos paramétricos se ha trabajado con la matriz de firmas ya conocida. Para los fallos discretos se han definido unas matrices similares a la matriz de firmas y que se presentan a continuación. Inicialmente se tiene la Hybrid Fault Signature Matrix (HFSM), en la que se relacionan las switching junctions y los HBG-PCs del sistema, la Figura 4.1 muestra la HFSM del sistema de cuatro tanques híbrido presentado en la sección anterior.

	HBG – PC1	**HBG – PC2**	**HBG – PC3**	**HBG – PC4**
$\mathbf{1_{SW_1}}$	1		1	
$\mathbf{1_{SW_2}}$	1	1		
$\mathbf{1_{SW_3}}$	1		1	
$\mathbf{1_{SW_4}}$			1	1

Cuadro 4.1: Hybrid Fault Signature Matrix (HFSM) del sistema de 4 tanques híbrido, muestra la relación entre las switching junctions del modelo y los HBG-PCs.

La HFSM se utiliza en la etapa de aislamiento de fallos, tras producirse la detección de un fallo, de la misma forma que se utilizaba la matriz de firmas para los fallos paramétricos: Según los HBG-PCs que se hayan activado en la etapa de detección se obtiene el conjunto de corte minimal de esos HBG-PCs activos con las switching junctions, teniendo en cuenta que no todas las switching junctions afectan a todos los HBG-PCS, el aislmiento será más preciso que si se trabajase con el sistema completo.

Como extensión de la HFSM se ha definido la Hybrid Qualitative Fault Signature Matrix (HQFSM), esta incluye información cualitativa sobre la variación del residuo [1]. Con esta información adicional, lo que se consigue es reducir aún más el conjunto de candidatos de fallo discretos, permitiendo que el proceso de identificación de estos fallos se haga de manera más eficiente. La Figura 4.2 muestra la HQFSM del sistema de 4 tanques híbridos centrándose en las switching junctions comandadas (SW_1 y SW_3).

	HBG-PC1	**HBG-PC3**
$\mathbf{1_{SW_1}1 \rightarrow 1}$	+	-
$\mathbf{1_{SW_1}0 \rightarrow 0}$	-	+
$\mathbf{1_{SW_1}0 \rightarrow 1}$	+	-
$\mathbf{1_{SW_1}1 \rightarrow 0}$	-	+
$\mathbf{1_{SW_3}1 \rightarrow 1}$	-	+
$\mathbf{1_{SW_3}0 \rightarrow 0}$	+	-
$\mathbf{1_{SW_3}0 \rightarrow 1}$	-	+
$\mathbf{1_{SW_3}1 \rightarrow 0}$	+	-

Cuadro 4.2: Hybrid Qualitative Fault Signature Matrix (HQFSM) del sistema de 4 tanques híbrido.

Una vez se tiene la arquitectura definida para realizar la detección y el aislamiento de fallos discretos y paramétricos en sistemas híbridos se amplía para incluir DBNs minimales derivadas a partir de los HBG-PCs en cada modo de funcionamiento. Con las DBNs minimales se amplía la arquitectura para considerar también la identificación de los fallos, de la misma forma que ya se presentó en el caso de sistemas continuos.

[1] residuo = variable medida - variable estimada.

El capítulo presenta el estudio realizado sobre el sistema de 4 tanques híbrido (ver Fig. 4.1) para probar el funcionamiento de la arquitectura de diagnosis unificada para sistemas híbridos con fallos discretos y paramétricos.

4.4. Extensión para actuadores multiposición

La herramienta de modelado Hybrid Bond Graphs sólo permite modelar actuadores de tipo ON/OFF, es decir, que tienen dos estadosposibles: conectado o desconectado, pero no permite modelar directamente actuadores multiposición que conecten caminos alternativos en cada una de sus posiciones. Este tipo de actuadores no son raros de encontrar en sistemas híbridos por eso se ha definido una estructura que permite modelar este tipo de actuadores multiposición mediante actuadores ON/OFF.

La Figura 4.5 muestra en a) la estructura general de los actuadores multiposición que se han comentado anteriormente y en b) el esquema que se ha diseñado para poder representarlo mediante switching junctions ON/OFF.

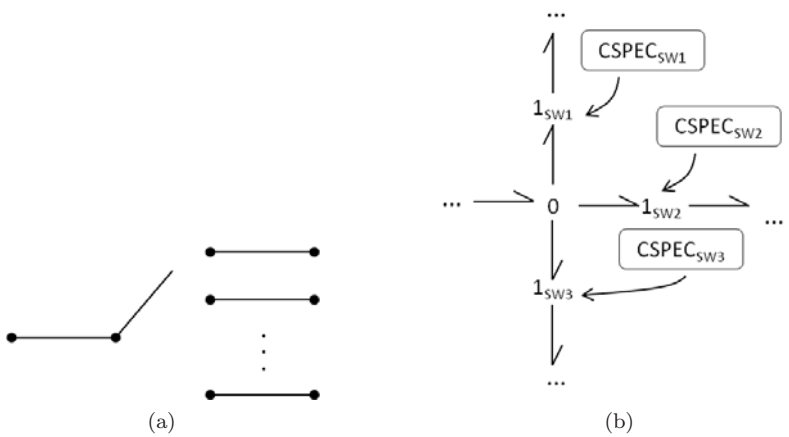


Figura 4.5: a) Estructura genérica de un actuador multiposición. b) Esquema HBG genérico para un actuador de 3 posiciones.

La idea principal del modelo consiste en utilizar una unión de tipo 0 (conexión paralelo) como punto de referencia y a ella conectar tantas switching junctions de tipo 1 como caminos alternativos tenga el actuador multiposición. Las señales que controlan esas switching junctions serán siempre mutuamente expcluyentes, de forma que sólo una de ellas puedes estar activa en un instante concreto. Este modelo se ha probado en el sistema que se presenta en el Capítulo 5, el Sistema de Osmosis Inversa (ROS), que tiene entre sus componentes una válula multiposición.

Capítulo 5

Caso de Estudio. Sistema de Ósmosis Inversa (ROS)

Los capítulos anteriores han ido presentando las diferentes propuestas y contribuciones de esta tesis. En este capítulo se muestra, en un sistema híbrido real del ámbito aeroespacial, la aplicación de dichas contribuciones.

El sistema empleado en este capítulo se denomina Sistema de Osmosis Inversa, según sus siglas en inglés le denominaremos ROS. El ROS está incluido en un Sistema Avanzado de Recuperación de Agua (AWRS) diseñado y construido por el centro espacial Jonhson de la NASA (JSC) como parte del Sistema Avanzado de Apoyo a la Vida (ALS) para misiones tripuladas [48, 58]. La Figura 5.1 muestra un esquema del AWRS. Este sistema trabaja en condiciones de microgravedad y permite obtener agua potable a partir de agua residual.

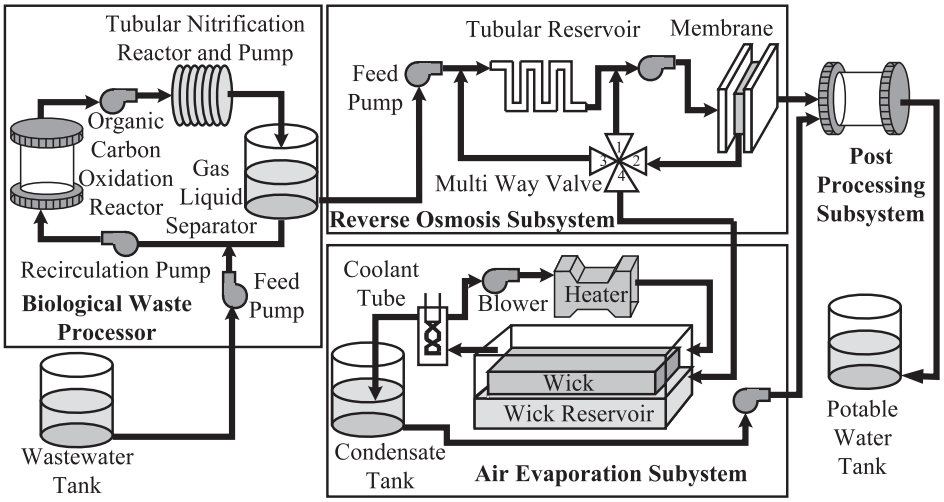


Figura 5.1: Esquema del Sistema Avanzado de Recuperación de Agua (AWRS).

El AWRS está formado por cuatro subsistemas: 1) El procesador de residuos biológicos (BWP), que elimina la materia orgánica y el amoniaco del agua residual; 2) El sistema de ósmosis inversa (ROS), que elimina la materia inorgánica y en suspensión mediante un sistema de filtración por membrana de alta presión; 3) El subsistema de evaporación de agua (AES), que recupera el agua restante en el líquido final del ROS mediante un proceso de evaporación y condensación; 4) El subsistema de postprocesado (PPS) que elimina trazas de impurezas y genera agua potable combinando la salida del ROS y del AES. Como se ha indicado antes, esta tesis se ha centrado en el estudio del ROS.

El ROS recibe agua del BWP, la bomba de entrada está siempre activa, de forma que el flujo es continuo durante todo el ciclo de trabajo. Este flujo de entrada se mezcla con el flujo que se recircula, cuando la bomba de recirculación está activa, y pasa por el depósito tubular. A continuación está situada la membrana encargada de filtrar el líquido para eliminar las impurezas. El agua filtrada pasa al subsistema de postproce-

sado (PPS) y una parte del líquido, que aún no está limpio, sigue el camino hacia una valvula multiposición (*Multi Way Valve* en la Figura 5.1). Según el modo de trabajo activo, la válvula estará en la posición 1, 2 o purga. En la posición 1, el agua se recircula por el camino largo del esquema (antes del depósito tubular), mientras que en la posición 2, el camino de recirculación activo es el corto (después del depósito tubular). La posición purga de la válvula conduce el líquido residual que tiene la membrana al AES. El ciclo de trabajo del ROS está formado por el modo 1 (válvula en posición 1), seguido del modo 2 (válvula en posición 2) y finalizando con el modo purga (válvula en posición purga), al terminar este modo se volvería de nuevo al modo 1 para comenzar un nuevo ciclo.

Como se puede extraer de la explicación anterior, el ROS es un sistema híbrido con suficiente complejidad para poder ilustrar la propuesta de diagnosis de sistemas híbridos presentada en el capítulo 4. Además tiene un actuador multiposición, lo que permite también comprobar la bondad del modelo propuesto para este tipo de actuadores utilizando sólo elementos ON/OFF.

La Figura 5.2 muestra el modelo HBG del ROS. Está dividido en subsistemas para comprender e identificar más fácilmente las distintas partes del sistema. Las señales de control M1, M2 y P modelan los cambios de la válvula multiposición, de forma que también determinan el modo de funcionamiento del ROS en cada momento.

El ROS tiene 5 medidas: 1) Flujo de entrada al sistema (F_FP), 2) La presión en el depósito tubular (P_Back), 3) la presión en la membrana (P_Memb), 4) la presión de salida de la bomba de recirculación (P_Pump) y 5) la concentración del líquido (P_k). La Figura 5.3 muestra las medidas del sistema durante 3 ciclos completos de funcionamiento, como se ha explicado anteriormente, cada ciclo está formado por el modo 1, seguido del modo 2 y finalmente el modo purga. Los datos tienen un 2 % de ruido. Las medidas se muestran en la figura de arriba a abajo en el mismo orden en que se han explicado. El flujo de entrada al sistema (F_FP) es casi continuo a lo largo del tiempo pero en las demás medidas observamos variaciones. La tercera medida en la Figura 5.3, es la presión en la membrana (P_Memb), en ella se pueden diferenciar los tres modos de funcionamiento, durante el primero y el segundo, la medida crece con distinta pendiente en cada una de ellas, mientras que en el modo purga, la presión cae drásticamente y se mantiene estable durante el resto del modo. La concentración del líquido (Pk) se muestra en la parte baja de la gráfica, en ella se pueden identificar claramente los tres modos de funcionamiento, además, es la variable utilizada por el controlador para fijar las señales de control.

Se han derivado los HBG-PCs del sistema (ver Figuras 5.4, 5.5, 5.6, 5.7 y 5.8) y se han obtenido sus DBNs (ver Figuras 5.9, 5.10, 5.11, 5.12 y 5.13). Con ellas se ha comprobado que son capaces de realizar la monitorización del sistema, siguiendo su comportamiento incluso inmediatamente después de un cambio de modo sin producir falsos positivos.

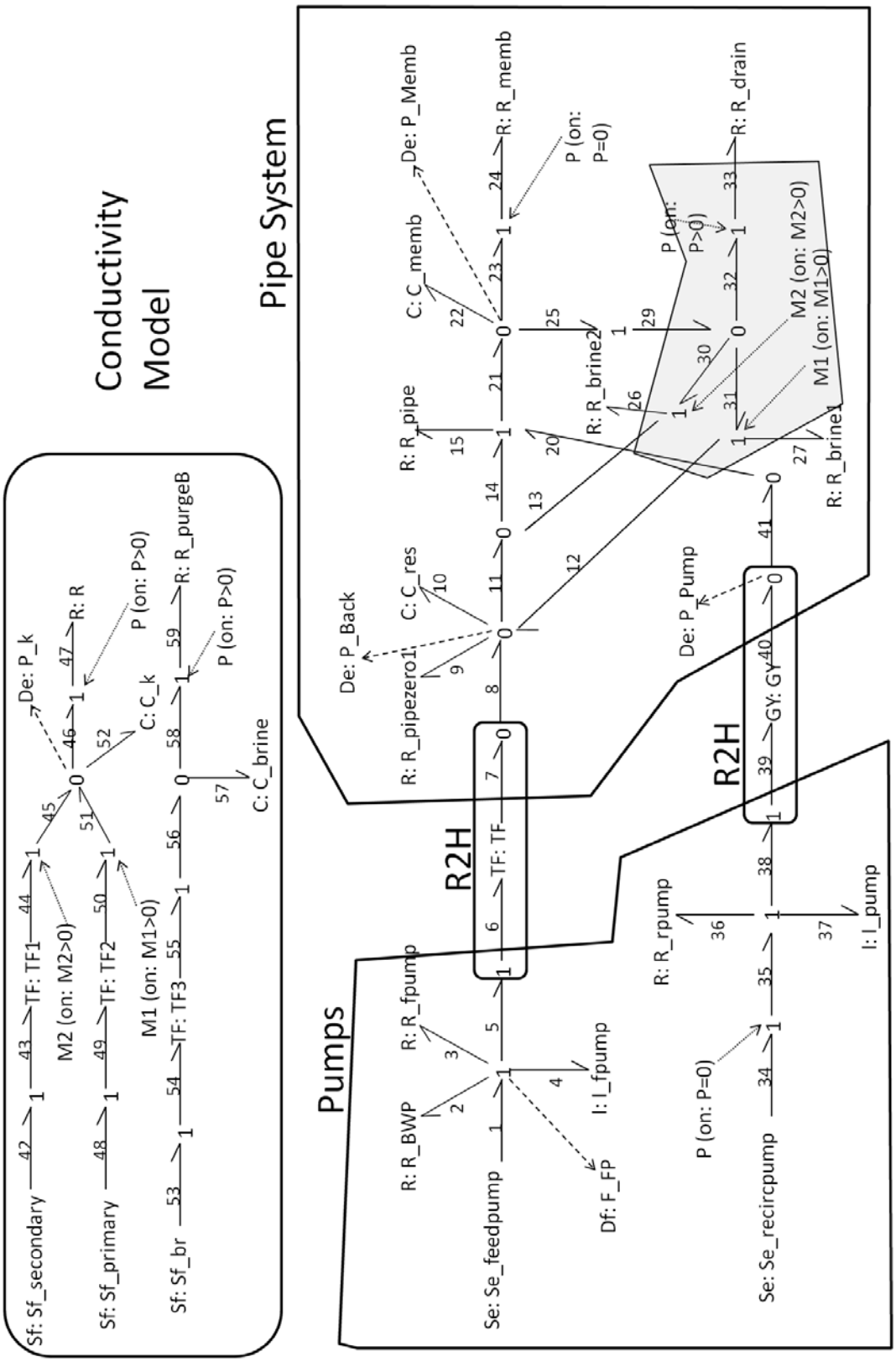


Figura 5.2: Modelo Hybrid Bond Graph del Sistema de Ósmosis Inversa (ROS).

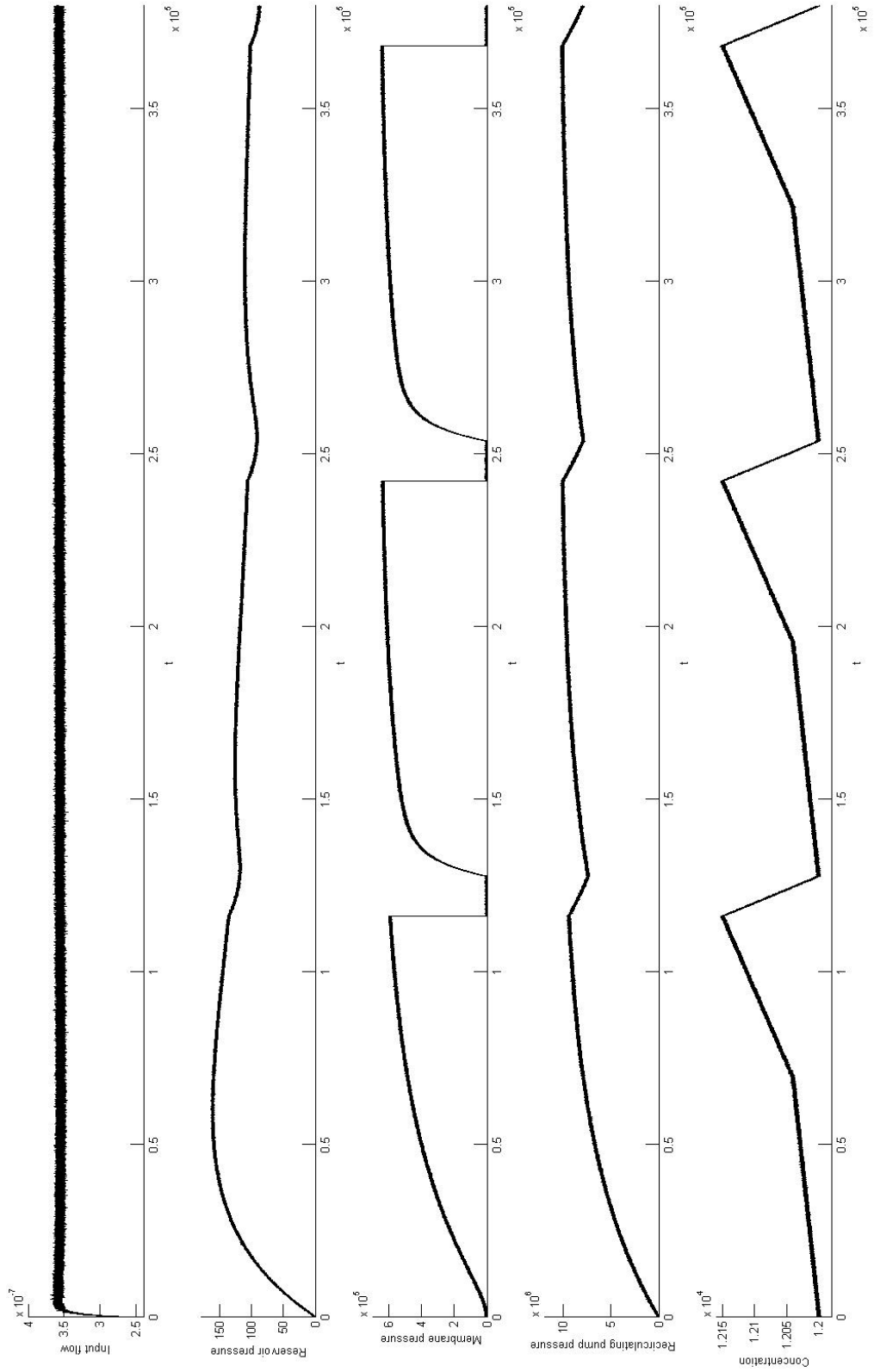


Figura 5.3: Comportamiento nominal del ROS durante 3 ciclos de funcionamiento completos para las medidas disponibles: F_FP, P_Back, P_memb, P_Pump, y P_k, de arriba a abajo en la figura.

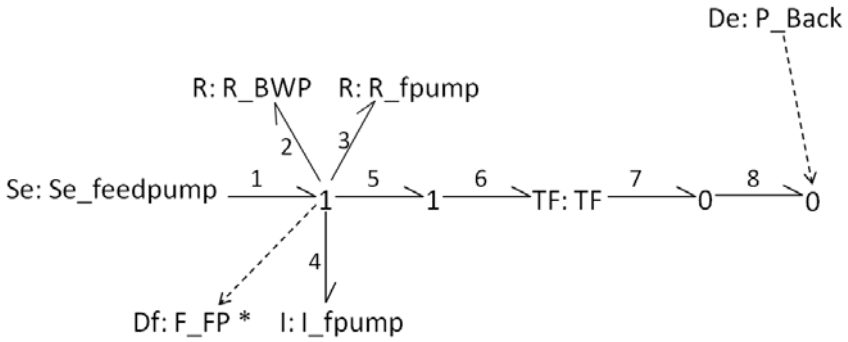


Figura 5.4: Modelo Hybrid Bond Graph del HBG-PC1 del Sistema de Ósmosis Inversa (ROS). El nodo discrepancia, variable estimada, es F_FP y la variable de estado es f_4, que está relacionada con el esfuerzo generalizado en I_fpump.

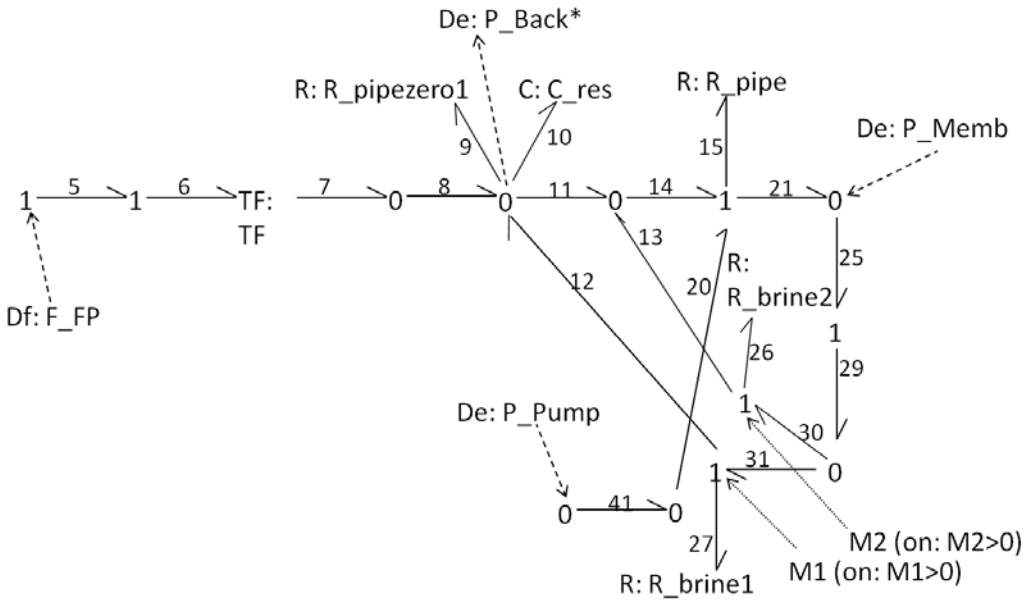


Figura 5.5: Modelo Hybrid Bond Graph del HBG-PC2 del Sistema de Ósmosis Inversa (ROS). El nodo discrepancia, variable estimada, es P_Back y la variable de estado es e_{10}, que está relacionada con la capacidad en el depósito tubular, C_res.

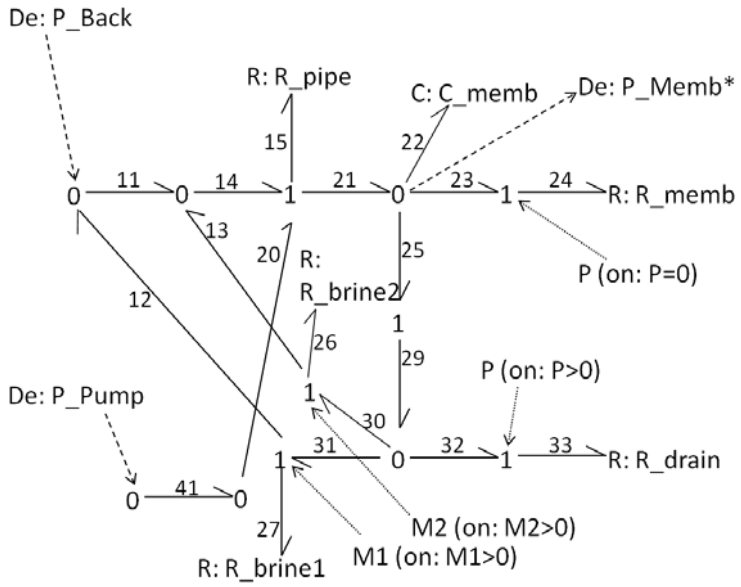


Figura 5.6: Modelo Hybrid Bond Graph del HBG-PC3 del Sistema de Ósmosis Inversa (ROS). El nodo discrepancia, variable estimada, es P_Memb y la variable de estado es e_{22}, que está relacionada con la capacidad en la membrana C_memb.

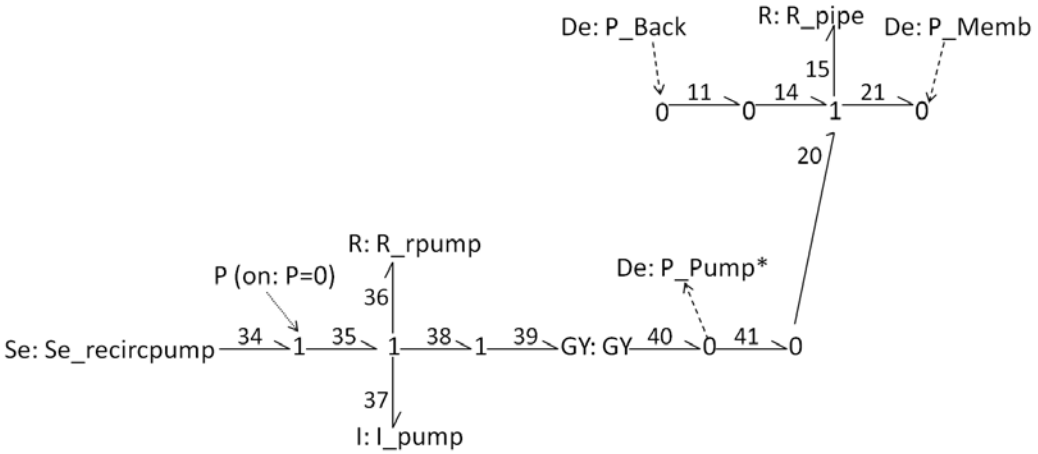


Figura 5.7: Modelo Hybrid Bond Graph del HBG-PC4 del Sistema de Ósmosis Inversa (ROS). El nodo discrepancia, variable estimada, es P_Pump y la variable de estado es f_{37}, que está relacionada con el esfuerzo generalizado en I_pump.

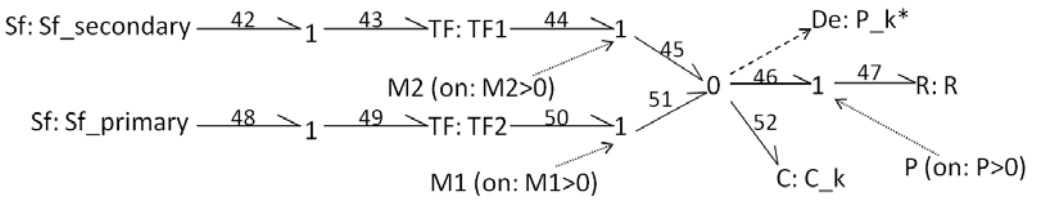


Figura 5.8: Modelo Hybrid Bond Graph del HBG-PC5 del Sistema de Ósmosis Inversa (ROS). El nodo discrepancia, variable estimada, es P_k y la variable de estado es e_{52}, que está relacionada con la concentración en el modelo de conductividad, C_k.

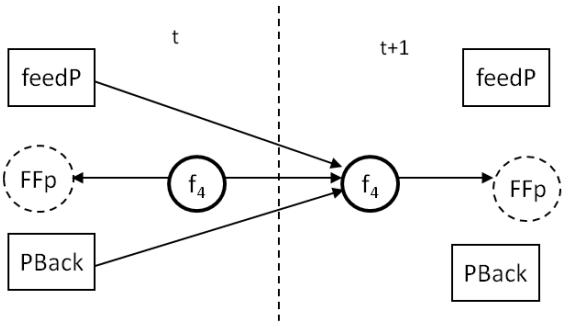


Figura 5.9: Red Bayesiana Dinámica del HBG-PC1 (DBN1) para el ROS en el modo de trabajo 1. La variable de estado es f_4 (flujo en la bomba de entrada), relacionada con el parámetro I_fpump en el modelo HBG.

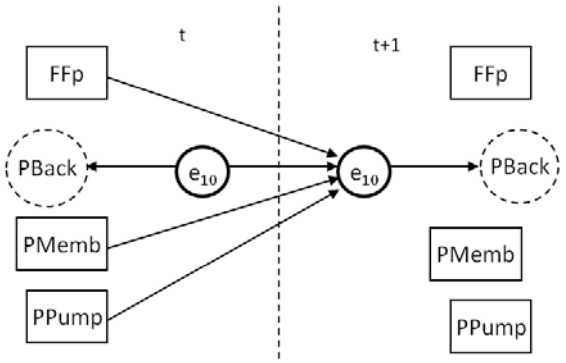


Figura 5.10: Red Bayesiana Dinámica del HBG-PC2 (DBN2) para el ROS en el modo de trabajo 1. La variable de estado es e_{10} (presión en el depósito), relacionada con el parámetro C_res en el modelo HBG.

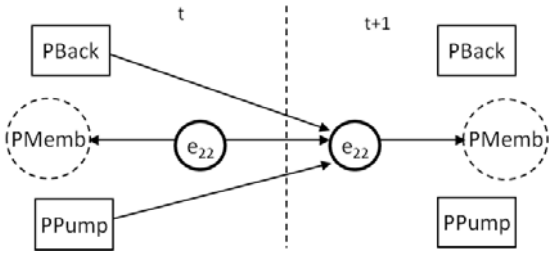


Figura 5.11: Red Bayesiana Dinámica del HBG-PC3 (DBN3) para el ROS en el modo de trabajo 1. La variable de estado es e_{22} (presión en la membrana), relacionada con el parámetro C_memb en el modelo HBG.

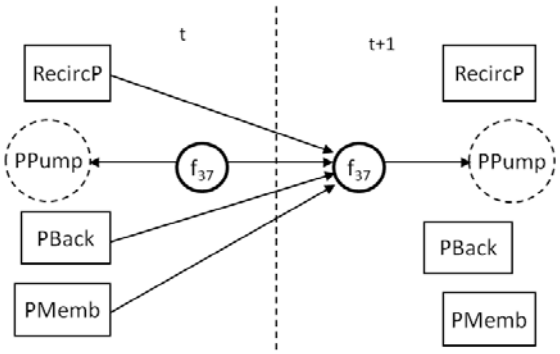


Figura 5.12: Red Bayesiana Dinámica del HBG-PC4 (DBN4) para el ROS en el modo de trabajo 1. La variable de estado es f_{37} (flujo en la bomba de recirculación), relacionada con el parámetro I_pump en el modelo HBG.

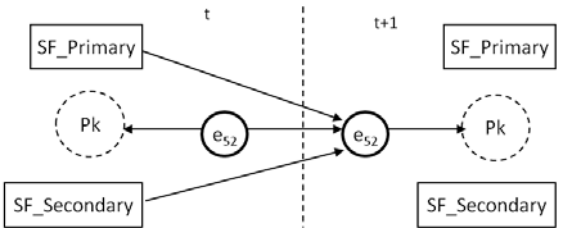


Figura 5.13: Red Bayesiana Dinámica del HBG-PC5 (DBN5) para el ROS en el modo de trabajo 1. La variable de estado es e_{52} (concentración en el modelo de conductividad), relacionada con el parámetro C_k en el modelo HBG.

En cuanto al proceso de diagnosis, se han estudiado tanto los fallos discretos en el sistema, como los fallos continuos (paramétricos) obteniendo resultados satisfactorios y dentro de lo esperado en todos los casos.

El proceso de diagnosis seguido es el que ya se ha presentado en el Capítulo anterior para sistemas híbridos: Los fallos discretos se consideran candidatos preferidos, de forma que si tras la detección del fallo y el proceso de aislamiento se obtienen candidatos tanto discretos como paramétricos en el conjunto de candidatos de fallo, primero se estudian los candidatos discretos. Si tras realizar la identificación con los candidatos discretos, todos ellos son rechazados, entonces se pasa a considerar los candidatos continuos o paramétricos como se explicó en el proceso de diagnosis de sistemas continuos.

Los experimentos comparan el desempeño de la DBN modelando el sistema completo y las DBNs derivadas de los HBG-PCs. En el proceso de detección, tanto de fallos discretos como continuos, el comportamiento es muy similar. En la etapa de aislamiento, las DBNs obtenidas de los HBG-PCs permiten que el proceso sea más eficiente y genere conjuntos de candidatos menores más rápidamente que la DBN modelando el sistema completo. El proceso de identificación de fallos discretos tiene unos resultados similares para cualquiera de las DBNs y en el caso de los fallos continuos, la estimación del parámetro de fallo, al igual que sucedía en los sistemas continuos, es mejor con las DBNs de los HBG-PCs en unos parámetros y con la DBN modelando el sistema completo en otros.

Capítulo 6
Conclusiones

Este capítulo resume las conclusiones extraídas durante la realización de la tesis. Además proporciona un listado de las contribuciones del trabajo que se han ido presentando a lo largo de los capítulos. Al final del capítulo se proponen algunas líneas de trabajo futuro.

6.1. Principales Contribuciones

Arquitectura de diagnosis para sistemas continuos con Redes Bayesianas Dinámicas minimales derivadas de Posibles Conflictos [2, 1, 3] La primera contribución de esta tesis es un método para derivar Redes Bayesianas Dinámicas (DBNs) minimales de Posibles Conflictos (PCs). Utilizando la estructura de los PCs (MEM o TCG) se puede derivar la estructura de la DBN, los parámetros (ecuaciones) de la DBN se obtienen de las etiquetas de los arcos.

Las DBNs minimales se pueden utilizar de manera eficiente durante las 3 etapas del proceso de diagnosis (detección, aislamiento e identificación). Como resultado se obtiene la segunda contribución presentada en el Capítulo 3, la definición de un marco para FDII de sistemas continuos basado en PCs con DBNs como única herramienta de estimación para la detección y la identificación.

Mejorar la identificación de parámetros fusionando DBNs minimales Las DBNs minimales permiten simplificar la etapa de aislamiento, pero en la identificación de los fallos se ha observado que los resultados obtenidos no son siempre tan precisos como las estimaciones de la DBN que modela el sistema completo. De manera general, las DBNs obtienen estimaciones del estado más precisas cuando hay más observaciones disponibles para ajustar esa estimación, pero esto tampoco es cierto en todos los casos. Por eso, se propone fusionar algunas DBNs minimales que compartan variables de estado y/o entradas y medidas para obtener una mejor estimación del parámetro a identificar si se necesita. Este método se presenta en el caso de estudio del Capítulo 3.

Posibles Conflictos para Sistemas Híbridos (HPCs). [13, 12] Extendemos la técnica de compilación de los Posibles Conflictos para utilizarla, no sólo con sistemas continuos, sino también con sistemas híbridos (HPCs). La herramienta de modelado utilizada ha sido Hybrid Bond Graphs (HBGs), que extiende los Bond Graphs con un nuevo tipo de uniones 1- (0-) controladas por un autómata que indica si la unión 1- (0-) está conectada (se comporta como cualquier unión de tipo 1- (0-)), o desconectada (el comportamiento de la unión es el de una fuente de flujo (esfuerzo) a cero).

Los HPCs, también referenciados como HBG-PCs por la técnica de modelado empleada, se han utilizado de manera satisfactoria para realizar la monitorización del sistema como se presenta en el Capítulo 4.

Los HBG-PCs no necesitan pre-enumerar todos los posibles modos de funcionamiento del sistema, solucionando uno de los principales problemas de la diagnosis de sistemas híbridos. Además, si el modelo HBG del sistema tiene asignación causal válida considerando todas las switching junctions ON, se puede derivar de él el conjunto de HBG-PCs que caracteriza todos los HBG-PCs de todos los modos.

Detección, aislamiento e identificación de fallos con HBG-PCs. [40] Los sistemas híbridos presentan dos tipos de fallos, los fallos que afectan a sus parámetros se conocen como fallos paramétricos y su

proceso de diagnosis es el mismo que se utiliza para los sistemas continuos. Por otro lado están los fallos que se deben a la naturaleza discreta de este tipo de sistemas, que pueden sufrir un cambio autónomo de modo de funcionamiento o no obedecer a una orden de cambio de modo. Estos dos tipos de fallos (paramétricos y discretos) se han integrado en una arquitectura de diagnosis unificada basada en HBG-PCs.

El proceso de diagnosis basado en HBG-PCs permite detectar fácilmente fallos discretos sin necesidad de medir directamente la señal de control. Los fallos discretos son candidatos preferidos a los fallos paramétricos y se confirmarán o descartarán primero durante el proceso de diagnosis. Sólo cuando todos los candidatos de fallo discretos se descarten se empezará el proceso de identificación de fallos paramétricos.

Una ventaja adicional es que sólo algunos HBG-PCs detectarán cada fallo, sus residuos se activarán; esos HBG-PCs serán utilizados en las siguientes etapas de diagnosis, mientras tanto, los demás HBG-PCs que no se han visto afectados por el fallo pueden seguir monitorizando el sistema.

Detección, aislamiento e identificación de fallos con DBNs minimales derivadas de HBG-PCs. [40, 41] Los HBG-PCs se han propuesto para la diagnosis de fallos discretos y paramétricos de sistemas continuos en una arquitectura de diagnosis unificada. Los HBG-PCs también presentan el comportamiento de los sistemas híbridos como una colección de comportamientos continuos de manera que las transiciones entre ellos se producen mediante eventos discretos (cambios autónomos o actuadores comandados). Esta simplificación permite utilizar DBNs minimales para modelar el comportamiento de cada modo de funcionamiento, evitando el uso de nodos discretos en la DBN, simplificando el proceso de inferencia y evitando problemas de convergencia. Una vez se tienen configurados los HBG-PCs para el modo de funcionamiento actual, se derivan las DBNs minimales de ellos. Las DBNs minimales se han integrado en la arquitectura de diagnosis para fallos discretos y paramétricos en sistemas híbridos.

Modelar actuadores multiposición mediante uniones ON/OFF (HBGs) Hybrid Bond Graphs, la técnica de modelado utilizada en los HBG-PCs sólo permite modelar uniones de tipo ON/OFF, lo que permite crear directamente modelos con actuadores de tipo ON/OFF. Hay actuadores multiposición en sistemas híbridos que conectan uno de múltiples caminos posibles, este tipo de actuadores no se pueden modelar directamente con HBGs pero en el Capítulo 4 se presenta un método que permite modelar este tipo de actuadores multiposición mediante uniones de tipo ON/OFF.

Casos de estudio

- **Circuito eléctrico de orden doce.** La arquitectura de diagnosis de sistemas continuos utilizando PCs y DBNs minimales se ha evaluado con un circuito eléctrico de orden doce (Capítulo 3).
- **Sistema de Ósmosis Inversa (ROS).** La arquitectura de diagnosis para sistemas híbridos basada en HBG-PCs y DBNs minimales se ha evaluado con el Sistema de Ósmosis Inversa (ROS) (Capítulo 5).

6.2. Conclusiones

Primeramente, se ha confirmado que los Posibles Conflictos (PCs) son una estructura válida y útil para derivar Redes Bayesianas Dinámicas (DBNs) minimales.

Las DBNs minimales se han integrado en una arquitectura de diagnosis unificada para sistemas continuos que permite realizar con la misma herramienta las tres fases del proceso de diagnosis. Los resultados obtenidos con las DBNs minimales en tareas de detección de fallos son equivalentes a los obtenidos con la DBN del sistema completo, pero requieren menos carga computacional. En cuanto al aislamiento de fallos, las DBNs minimales permiten obtener conjuntos de candidatos más precisos (con menos candidatos) una vez se ha detectado el fallo. Por otro lado, en la etapa de identificación de fallos no siempre se obtienen resultados tan precisos como los de la DBN que modela el sistema completo. Para resolver esta debilidad, se propone un método para fusionar DBNs minimales durante las tareas de identificación. En todos los escenarios y tareas, las DBNs minimales necesitan un esfuerzo computacional menor que la DBN que modela el sistema completo.

Los HBG-PCs son la extensión de los PCs para sistemas híbridos. Se ha definido su utilización para realizar tanto monitorización como diagnosis de sistemas híbridos. Los HBG-PCs permiten diagnosticar tanto fallos discretos como paramétricos en un arquitectura unificada, lo que proporciona una importante contribución en el campo de la diagnosis de sistemas híbridos. La técnica de HBG-PCs también simplifica la monitorización del comportamiento del sistema descomponiéndolo en una colección de comportamientos continuos, uno para cada modo de funcionamiento, la transición entre ellos se produce por un evento discreto.

Las DBNs minimales también se pueden derivar de los HBG-PCs, pudiendo utilizar una DBN para cada modo de funcionamiento (comportamiento continuo) obteniendo un desempeño más eficiente y una convergencia más precisa. Estas DBNs minimales se pueden integrar en una arquitectura de diagnosis para la detección, el asilamiento y la identificación de fallos discretos y paramétricos en sistemas híbridos.

Considerando la hipótesis principal de esta tesis y las conclusiones explicadas anteriormente se puede extraer que las conclusiones confirman la hipótesis en la que se enunciaba la posibilidad de integrar la detección, el asilamiento y la identificación de fallos en sistemas híbridos en entornos con ruido de manera eficiente. Para ello se han utilizando los Posibles Conflictos extendidos para sistemas híbridos y DBNs, como establecía la hipótesis.

Hybrid Bond Graphs (HBGs), la herramienta de modelado de los HBG-PCs, sólo utiliza uniones de tipo ON/OFF, lo que restringe bastante la aplicabilidad de la técnica. Para solventar esa restricción se ha propuesto una estructura que permite modelar actuadores multiposición utilizando sólo uniones de tipo ON/OFF.

6.3. Trabajo Futuro

Esta sección resume el trabajo futuro relacionado con las contribuciones de esta tesis. Parte del trabajo que se presenta a continuación ya está actualmente en progreso.

- **Incertidumbre en los parámetros.** Sobre este tema ya se hicieron algunos estudios preliminares [39].

  La diagnosis basada en modelos explota la redundancia analítica del sistema a través de los residuos generados. Para ello, compara las salidas del sistema con las predicciones del modelo. En un sistema sin fallos, los valores de los residuos serían teóricamente cero. En condiciones ideales, residuos diferentes de cero indicarían que el sistema ha sufrido un fallo. Como se puede ver, la generación de residuos robustos es un elemento importante de la MBD.

  La generación de residuos es un gran problema para sistemas no lineales porque es díficil construir modelos muy precisos. La estimación de estados óptima no es factible para sistemas no lineales a día de hoy. Para sistemas no lineales, la sensibilidad de los residuos es un problema difícil de abordar con métodos analíticos y numéricos. Para lidiar con este problema se propone una aproximación estructural. En los trabajos preliminares realizados se ha adoptado esta misma idea, utilizando Bond Graphs [32, 62] como método de modelado de los sistemas no lineales.

  Normalmente, la incertidumbre durante la monitorización de un sistema continuo puede provenir de dos fuentes: (1) Incertidumbre en el propio modelo, (2) incertidumbre en las medidas [25]. La incertidumbre en las medidas, atribuida a los sensores, se suele modelar como ruido gausiano con media cero, en ese caso, el punto clave consiste en estimar la varianza desconocida en las medidas con ruido para hacer más robusta la detección de fallos. Para sistemas lineales, este problema se ha abordado mediante Filtros de Kalman, y para sistemas no lineales, mediante métodos como el Filtro de Kalman Extendido [14], o el Filtro de Partículas [6].

  En este trabajo, asumimos que la estructura del modelo es conocida, y nos centramos en el problema cuando los parámetros no son todos conocidos con certeza. Este escenario es realista e importante, ya que muchos modelos no lineales reales tienen problemas comunes: (1) Problemas de convergencia númerica, y (2) Los valores que se conocen de sus parámetros son aproximados. Desde el punto de vista de la monitorizacion y la generación de residuos del comportamiento de un sistema no linenal, un pequeño cambio en sus parámetros puede suponer cambios mucho mayores en el comportamiento dinámico.

La forma habitual de afrontar la incertidumbre en los parámetros consiste en asumir que el valor del parámetro pertenece a un intervalo conocido, aunque el valor exacto del parámetro sea desconocido. Esta es la suposición que usamos también en nuestro trabajo y empleamos el método de Kam y Dauphin-Tanguy y otros [30, 22] para modelar incertidumbre en parámetros con una aproximación de bond graphs.

- **HBG-PCs aproximación incremental** Actualmente estamos trabajando en una aproximación incremental para construir los HBG-PCs a partir de los modelos HBG-PCs del sistema en el modo anterior. Esta propuesta es menos restrictiva que el método presentado en esta tesis.

  La idea consiste en comenzar con el modo de funcionamiento inicial configurado en el modelo HBG, se le asigna causalidad al modelo y a partir de él se obtienen los HBG-PCs para la configuración actual. Cuando se produzca un cambio de modo, los HBG-PCs se reconfigurarán para el nuevo modo de funcionamiento. Actualmente estamos trabajando en este proceso para que la búsqueda tras cada cambio de modo se haga de manera eficiente y no necesite extenderse a todo el modelo del HBG-PC en principio.

- **HBG-PCs aproximación con causalidad parcial.** Como se ha dicho previamente, asumimos que el modelo HBG del sistema tiene asignación causal válida considerando todas las switching junctions ON. Una propuesta menos restrictiva establece que sólo algunos elementos del sistema tienen causalidad fija y los otros se mantienen sin causalidad hasta que se configura el modo de trabajo actual.

  Esta propuesta se combinará con la aproximación incremental para encontrar el conjunto de HBG-PCs de la configuración actual tras un cambio de modo. También puede utilizarse junto con la propuesta de esta tesis, en la etapa inicial, se derivan de un HBG con causalidad parcial, las estructuras en las que estarán incluidos los HBG-PCs de cada modo, en lugar de partir de un HBG con todas las switching junctions ON. Durante el proceso de diagnosis, cuando se detecte un cambio de modo, esas estructuras serán reconfiguradas.

Bibliografía

[1] C. J. Alonso-González, N. Moya, and G. Biswas. Dynamic Bayesian Network Factors from Possible Conflicts for Continuous System Diagnosis. In *Proceedings of the 14th international conference on Advances in artificial intelligence: spanish association for artificial intelligence*, CAEPIA'11, pages 223–232, Berlin, Heidelberg, 2011. Springer-Verlag.

[2] C.J. Alonso-González, N. Moya, and G. Biswas. Factoring Dynamic Bayesian Networks Using Possible Conflicts. In *Proceeding of the 21th International Workshop on Principles of Diagnosis, DX10*, October 2010.

[3] C.J. Alonso-González, N. Moya, and G. Biswas. Dynamic Bayesian Network Factors from Possible Conflicts for Continuous System Diagnosis. In J.A. Lozano, J.A. Gámez, and J.A. Moreno, editors, *Advances in Artificial Intelligence*, volume 7023 of *Lecture Notes in Computer Science*, pages 223–232. Springer Berlin Heidelberg, 2011.

[4] R. Alur, C. Courcoubetis, T. Henzinger, and P. Ho. Hybrid automata: An algorithmic approach to the specification and verification of hybrid systems. In *Hybrid Systems*, volume 736 of *Lecture Notes in Computer Science*, pages 209–229. Springer Berlin / Heidelberg, 1993.

[5] J. Armengol, A. Bregon, T. Escobet, M. Krysander, M .Nyberg, X. Olive, B. Pulido, and L. Travé-Massuyès. Minimal Structurally Overdetermined sets for residual generation: A comparison of alternative approaches. In *Proceedings of the 7th IFAC Symposium on Fault Detection, Supervision and Safety of Thecnical Processes, SAFEPROCESS09*, pages 1480–1485, Barcelona, 2009.

[6] M.S. Arulampalam, S. Maskell, and N. Gordon. A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking. *IEEE Transactions on Signal Processing*, 50:174–188, 2002.

[7] M. Bayoudh, L. Travé-Massuyès, and X. Olive. Coupling Continuous and Discrete Event System Techniques for Hybrid System Diagnosability Analysis. In *Proceeding of the 2008 conference on ECAI 2008: 18th European Conference on Artificial Intelligence*, pages 219–223, Amsterdam, The Netherlands, The Netherlands, 2008. IOS Press.

[8] E. Benazera and L. Travé-Massuyès. Set-theoretic estimation of hybrid system configurations. *Trans. Sys. Man Cyber. Part B*, 39:1277–1291, October 2009.

[9] G. Biswas, M.O. Cordier, J. Lunze, Travé-Massuyès, and M. Staroswiecki. Diagnosis of complex systems: bridging the methodologies of the FDI and DX communities. *Part B: Cybernetics, IEEE Transactions on Systems, Man, and Cybernetics*, 34(5):2159–2162, 2004.

[10] M. Blanke, M. Kinnaert, J. Lunze, M. Staroswiecki, and J. Schröder. *Diagnosis and Fault-Tolerant Control*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006.

[11] W. Borutzky. Representing discontinuities by means of sinks of fixed causality. In F. E. Cellier and J. J. Granda, editors, *Proceedings of the International Conference on Bond Graph Modeling, ICBGM95*, pages 65–72. SCS Publishing, January 15-18 1995.

[12] A. Bregon, C. Alonso, G. Biswas, B. Pulido, and N. Moya. Fault Diagnosis in Hybrid Systems using Possible Conficts. In *Proc. of the 8th IFAC Symposium on Fault Detection, Supervision and Safety of Technical Processes, SAFEPROCESS12*, Mexico City, Mexico, 2012.

[13] A. Bregon, C. J. Alonso-González, G. Biswas, B. Pulido, and N. Moya. Hybrid Systems Fault Diagnosis with Possible Conflicts. In *Proceedings of the 22nd International Workshop on Principles of Diagnosis, DX11*, pages 36–43, Murnau, Germany, 2011.

[14] E. N. Chatzi and A. W. Smyth. The unscented kalman filter and particle filter methods for nonlinear structural system identification with non-collocated heterogeneous sensing. *Structural Control and Health Monitoring*, 16(1):99–123, 2009.

[15] V. Cocquempot, T. El Mezyani, and M. Staroswiecki. Fault detection and isolation for hybrid systems using structured parity residuals. In *Control Conference, 2004. 5th Asian*, volume 2, pages 1204 – 1212 Vol.2, july 2004.

[16] L. Console. Model-based diagnosis history and state of the art. MONET Excellence Network. Summer School, Bertinoro, Italy, Mayo 2000.

[17] M.O. Cordier, P. Dague, F. Lévy, J. Montmain, M. Staroswiecki, and Travé-Massuyès. Conflicts versus Analytical Redundancy Relations: A comparative analysis of the model based diagnosis approach from the Artificial Intelligence and Automatic Control perspectives. *Part B: Cybernetics, IEEE Transactions on Systems, Man, and Cybernetics*, 34(5):2163–77, 2004.

[18] P.J.L. Cuijpers and M.A. Reniers. Hybrid process algebra. *Journal of Logic and Algebraic Programming*, 62(2):191 – 245, 2005.

[19] R. Davis. Expert systems: Where are we? And where do we go from here? *Artificial Intelligence Magazine*, 3(2):3–22, 1982.

[20] R. Dearden and D. Clancy. Particle filters for real-time fault detection in planetary rovers. In *The 12th International Workshop on Principles of Diagnosis, DX01*, pages 1–6, 2001.

[21] M. Delgado and H. Sira-Ramirez. Modeling and simulation of switch regulated dc-to-dc power converters of the boost type. In *Proceedings of the First IEEE International Caracas Conference on Devices, Circuits and Systems*, pages 84 –88, dec 1995.

[22] M.A. Djeziri, R. Merzouki, B.O. Bouamama, and G. Dauphin-Tanguy. Robust Fault Diagnosis by Using Bond Graph Approach. *Mechatronics, IEEE/ASME Transactions on*, 12(6):599 –611, 2007.

[23] J.P. Ducreux, G. Dauphin-Tanguy, and C. Rombaut. Representing discontinuities by means of sinks of fixed causality. In F. E. Cellier and J. J. Granda, editors, *roceedings of the International Conference on Bond Graph Modeling, ICBGM93*, volume 25, pages 132–136. SCS Publishing, 1993.

[24] P.J. Gawthrop. Hybrid Bond Graphs Using Switched I and C Components, 1997.

[25] A. Gelb. *Applied Optimal Estimation*. MIT Press, Cambridge, MA, USA, 1979.

[26] J. Gertler. *Fault Detection and Diagnosis in Engineering Systems*. Number 9780824794279. CRC Press, first edition edition, May 1998.

[27] W. Hamscher, L. Console, and J. de Kleer. *Readings in model-based diagnosis*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1992.

[28] M.W. Hofbaur and B.C. Williams. Hybrid estimation of complex systems. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, 34(5):2178 –2191, oct. 2004.

[29] R. Isermann. Supervision, fault-detection and fault-diagnosis methods – An introduction. *Control Engineering Practice*, 5(5):639 – 652, 1997.

[30] C. S. Kam and G. Dauphin-Tanguy. Bond Graph models of structured parameter uncertainties. *Journal of the Franklin Institute*, 342(4):379 – 399, 2005.

[31] D.C. Karnopp, D.L. Margolis, and R.C. Rosenberg. *System Dynamics: Modeling and Simulation of Mechatronic Systems*. John Wiley & Sons, Inc., New York, NY, USA, 2006.

[32] D.C. Karnopp, R. C. Rosenborg, and D.L. Margolis. *Systems Dynamics. A Unified approach*. Wiley & Sons, 3rd ed. edition, 2000.

[33] J. De Kleer and B.C. Williams. Diagnosis with behavioral modes. In *IJCAI'89: Proceedings of the 11th International Joint Conference on Artificial Intelligence*, pages 1324–1330, San Francisco, CA, USA, 1989. Morgan Kaufmann Publishers Inc.

[34] B. Kuipers. Qualitative reasoning: modeling and simulation with incomplete knowledge. *Automatica*, 25(4):571–585, 1989.

[35] U. Lerner, R.Paar, D. Koller, and G. Biswas. Bayesian Fault Detection and Diagnosis in Dynamic Systems. In *Proccedings of the AAAI/IAAI*, pages 531–537, 2000.

[36] J. Lunze. Diagnosis of Quantised Systems by Means of Timed Discrete-Event Representations. In *Proceedings of the Third International Workshop on Hybrid Systems: Computation and Control*, HSCC '00, pages 258–271, London, UK, 2000. Springer-Verlag.

[37] P. Mosterman and G. Biswas. Diagnosis of continuous valued systems in transient operating regions. *IEEE Transactions on Systems, Man, and Cybernetics*, 29(6):554–565, 1999.

[38] P.J. Mosterman and G. Biswas. Behavior Generation using Model Switching - A Hybrid Bond Graph Modeling Technique. In *Society for Computer Simulation*, pages 177–182. SCS publishing, 1994.

[39] N. Moya, G. Biswas, C.J. Alonso-González, and X. Koutsoukos. Structural Observability. Application to decompose a System with Possible Conflicts. In *Proceedings of the 21th International Workshop on Principles of Diagnosis, DX10*, pages 241–248, June 2010.

[40] N. Moya, A. Bregon, C. J. Alonso-González, B. Pulido, and G. Biswas. Extendng Hybrid Possible Conflicts to Diagnose Discrete Faults. In *Proceedings of the 23rd Int. WS on Pples. of Diagnosis, DX12*, pages 51–58, Great Malvern, UK, Jul-Aug 2012.

[41] N. Moya, B. Pulido, C. J. Alonso-González, A. Bregon, and D. Rubio. Automatic Generation of Dynamic Bayesian Networks for Hybrid Systems Fault Diagnosis. In *Proceedings of the 23rd Int. WS on Pples. of Diagnosis, DX12*, pages 59–66, Great Malvern, UK, Jul-Aug 2012.

[42] K. P. Murphy. *Dynamic Bayesian Networks: Representation, Inference and Learning*. PhD thesis, University of California, Berkeley, 2002.

[43] S. Narasimhan. Automated Diagnosis of Physical Systems. In *Proceedings of ICALEPCS07*, pages 701–705, 2007.

[44] S. Narasimhan and G. Biswas. Model-Based Diagnosis of Hybrid Systems. *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans*, 37(3):348–361, May 2007.

[45] S. Narasimhan, R. Dearden, and E. Benazera. Combining particle filters and consistency-based approaches for monitoring and diagnosis of stochastic hybrid systems. In *In Proceedings of the 15th International Workshop on Principles of Diagnosis, DX04*, Carcassonne, France, 2004.

[46] R. J. Patton, P. M. Frank, and R. N. Clark. *Issues of Fault Diagnosis for Dynamic Systems*. Springer, 2000.

[47] A. Pernestal, M. Nyberg, and B. Wahlberg. A bayesian approach to fault isolation with application to diesel engine diagnosis. In *In Proceedings of the 17th International Workshop on Principles of Diagnosis, DX06*, pages 211–218, Penaranda de Duero, Spain, 2006.

[48] K. D. Pickering, K. Wines, G. M. PAriani, L. A. Franks, J. Yeh, B. W. Finger, M. L. Campbell, C. E. Verostko, C. Carrier, J. C. Gandhi, and L. M. Vega. Early results of an ontegrated water recovery system test. In *In Proceedings of the 29th International Conference on Environmental Systems*, Orlando, FL, 2001.

[49] A. D. Pouliezos and G. S. Stavrakakis. *Real Time Fault Monitoring of Industrial Processes*. Kluwer Academic Publishers, Norwell, MA, USA, 1994.

[50] B. Pulido, C. Alonso, and F. Acebes. Lessons learned from diagnosing dynamic systems using Possible Conflicts and Quantitative Models. In *Engineering of Intelligent Systems. Fourteenth International Conference on Industrial and Engineering Applications of Artificial Intelligence and Expert Systems (IEA/AIE-2001)*, volume 2070 of *Lecture Notes in Artificial Intelligence*, pages 135–144, Budapest, Hungary, 2001.

[51] B. Pulido and C. Alonso-González. Possible Conflicts: A compilation technique for consistency-based diagnosis. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, 34(5):2192–2206, October 2004.

[52] B. Pulido, A. Bregon, and C. Alonso-González. Analyzing the Influence of Differential Constraints in Possible Conflict and ARR Computation. In *Proceedings of the 13th Conference of the Spanish Association for Artificial Intelligence, CAEPIA 2009*, Seville, Spain, 2009.

[53] B. Pulido, J.M. Zamarreño, A. Merino, and A. Bregon. Using structural decomposition methods to design gray-box models for fault diagnosis of complex systems: A beet sugar factory case study. In A. Bregon and A. Saxena, editors, *Procs. of the First European Conference of the Prognostics and Health Management Society*, pages 225–238, Dresden, Germany, July 2012. www.phmsociety.org.

[54] R. Reiter. A theory of diagnosis from first principles. *Artificial Intelligence*, 32:57–95, 1987.

[55] Th. Rienmüller, M. Bayoudh, M.W. Hofbaur, and L. Travé-Massuyès. Hybrid Estimation through Synergic Mode-Set Focusing. In *Proc. of IFAC Safeprocess'09*, pages 1480–1485, Barcelona, Spain, 2009.

[56] I. Roychoudhury. *Distributed Diagnosis of Continuous Systems: Global diagnosis through local analysis*. PhD thesis, Graduate School of the Vanderbilt University, August 2009.

[57] I. Roychoudhury, G. Biswas, and X. Koutsoukos. Comprehensive Diagnosis of Continuous systems Using Dynamic Bayes Nets. In *Proceeding of the 19th International Workshop on Principles of Diagnosis, DX08*, Blue Mountains, Australia, September 2008.

[58] I. Roychoudhury, G. Biswas, and X. Koutsoukos. Designing Distributed Diagnosers for Complex Continuous Systems. *Automation Science and Engineering, IEEE Transactions on*, 6(2):277 –290, april 2009.

[59] I. Roychoudhury, G. Biswas, and X. Koutsoukos. Factoring Dynamic Bayesian Networks based on Structural Observability. In *In 48th IEEE Conference on Decision and Control (CDC 2009)*, 2009.

[60] I. Roychoudhury, M. J. Daigle, G. Biswas, and Xenofon Koutsoukos. Efficient simulation of hybrid systems: A hybrid bond graph approach. *SIMULATION: Transactions of the Society for Modeling and Simulation International*, April 2010.

[61] S.J. Russell and P. Norvig. *Artificial Intelligence: A Modern Approach*. Prentice-Hall Inc., 2nd ed. edition, 1995.

[62] A. K. Samantaray and B. O. Bouamama. *Model-based Process Supervision: A Bond Graph Approach*. Springer-Verlag, London, 2008.

[63] M. Staroswiecki. A structural view of fault-tolerant estimation. In *Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering*, volume 221, pages 905–914, 2007.

[64] V. Venkatasubramanian, R. Rengswamy, K. Yin, and S.N. Kavuri. A review of process fault detection and diagnosis. Part I: Quantitative model-based methods. *Computers and Chemical Engineering*, 27:293–311, 2003.

[65] V. Venkatasubramanian, R. Rengswamy, K. Yin, and S.N. Kavuri. A review of process fault detection and diagnosis. Part II: Qualitative models and search strategies. *Computers and Chemical Engineering*, 27:313–326, 2003.

[66] V. Venkatasubramanian, R. Rengswamy, K. Yin, and S.N. Kavuri. A review of process fault detection and diagnosis. Part III: Process history based methods. *Computers and Chemical Engineering*, 27:327–346, 2003.

Part II.

Fault Diagnosis of Hybrid Systems with Dynamic Bayesian Networks and Hybrid Possible Conflicts

Contents

List of Figures

List of Tables

Chapter 1
Introduction

1.1 Motivation

Fault detection and diagnosis, FDD, is an important issue in the society. More and more tasks are performed by machines and robots every day. A non detected programming mistake or the physical exhaustion can appear as faults in a system. Fault diagnosis is used to keep safety in those tasks and its environments. Faults have to be detected as soon as possible. They also need to be identified and fixed to drive the system to a safe state. This work is devoted to FDD of dynamic systems whose behaviour changes over time, and its current internal state depends on previous states and the value of system inputs.

Dynamic systems can be divided in three general types: 1) Continuous systems, 2) Discrete systems, and 3) Hybrid systems. Continuous systems have a continuous behaviour and can be modeled using Ordinary Differential Equations (ODEs). Discrete systems have a finite number of states so can be modeled using automata. Hybrid systems have characteristics from the two types previously said, they have continuous and discrete behaviour. Discrete events modify the behaviour of those systems. They can be found in a large range of engineering applications (i.e. mechanical systems, electrical circuits, or embedded computation systems).

There are different types of hybrid systems, in this work, we focus on continuous systems governed by discrete events. This kind of hybrid systems are quite common in embebded systems. Different faults appear in those systems: 1) *continuous faults*, like sensor drifting or parameter changes, and 2) *discrete faults*, like a stuck relay or a valve which autonomously changes its position.

Hybrid systems usually have complex dynamic behaviour and they need a reliable diagnosis process from the very begining of their lifetime. Because of that, general formal tools from diagnosis analysis and design need to be used.

There are several branches working on the field of fault diagnosis: Knowledge-base, data driven and model based approaches [121, 122, 123]. Model-based diagnosis (MBD) is the only one which have the methodology needed to develop the diagnosers previously described. This work is placed in the Model-based diagnosis field.

Model-based diagnosis technique uses models to perform the diagnosis process [54, 56, 36, 100]. Basically, it compares the expected behaviour, estimated with the model, to the observed behaviour to detect deviations.

Online fault diagnosis of hybrid systems is not always an easy task due to its behaviour. Model-based online diagnosis methods require quick and robust reconfiguration processes when a mode change occurs, as well as the ability to keep on tracking the nominal behaviour of the system during transitory states.

Different approaches have been proposed to solve the hybrid systems fault diagnosis problem [6, 9, 74, 34]. Typically, hybrid systems are modeled as hybrid automata, pre-enumerating all the system modes explicitly, even if some of these modes will never be visited during execution. The hybrid automata approach works well for small systems, but it has a high computational cost for large systems enumerating modes that may never be visited. Alternative ways have been proposed to simulate hybrid systems, like hybrid bond graphs [84], where models for the hybrid modes are generated at runtime. Anyway, these approaches also exhibit different problems regarding the online computational effort required for the reconfiguration of the computational structures when a mode change occurs. This problem is especially relevant in modern

electronics-based electrical power systems, where model transitions occur at fast rates. Roychoudhury et al. [106] proposed an approach for efficient reconfiguration of hybrid systems using hybrid bond graph (HBG) models. Moreover, there is not a unified diagnosis approach for hybrid systems to perform fault detection, isolation and identification of continuous (parametric) and discrete faults.

Real-world systems work in noisy environments. Models of systems can have some level of noise, not only in the measurements but also in the model itself. This problem requires robust modeling and simulation tools. The same method wants to be used along the three diagnosis stages (fault detection, isolation and identification).

Dynamic Bayesian Networks have been chosen as the tool to model and simulate the systems due to its ability to deal with noise. They can also be used for fault detection, isolation and identification, which provides a unified approach for fault diagnosis.

On the other hand, DBNs have some disadvantages. Firstly, it is the high computational effort needed to do exact inference but this may be solved using a Particle Filter (PF) algorithm (approximate inference). The computational complexity with approximate inference can also be a drawback of this tool. Another issue about DBNs is the difficulty of getting an accurate convergence when there are several unknown states.

The main motivation for this dissertation consist of improving recent methods for hybrid systems diagnosis by developing a model-based solution that performs FDI in uncertain (i.e., noisy) environments with DBNs. DBNs with PF inference will be used on every aforementioned diagnosis step. To reduce the computational burden of fault isolation and identification, DBNs will be generated from subsystems with some kind of redundancy instead of using the complete system.

1.2 Guidelines and Main Hypothesis

This work presents an efficient method to perform Fault Detection, Isolation and Identification (FDII) within continuous systems governed by discrete events. The proposed tool to model those systems is Dynamic Bayesian Networks (DBNs).

Even using approximate methods (i.e., Particle Filter Algorithm) to perform inference in DBNs, the computational cost can be a problem. This cost will increase as the size of the network increases. Another factor which increases the computational cost of the inference process is the accuracy of the DBN state compared to the actual state of the system. This dissertation proposes Possible Conflicts technique [95] to factorize the DBN modeling the complete system into minimal DBNs modeling PCs (minimal overdetermined subsystems). DBNs with the smaller number of nodes and arcs are assumed to be less complex. This is a desirable characteristic in DBNs to get an accurate and less computational complex convergence. DBNs for hybrid systems have been used with discrete nodes modeling discrete states [41, 86] but this builds larger and more complex DBNs which convergence is more difficult.

This work proposes to use DBNs to model the continuous behaviour of the system. Different DBNs will be used for different system modes or configurations. As we have proposed to derive DBNs from PCs, the Possible Conflicts technique will be extended for hybrid systems. Our first guess proposes to include in the PCs approach the HBGs method [106] to efficiently generate system models when a discrete change occurs, so PCs for each configuration will be efficiently generated after a change in the system mode occurs.

The main hypothesis in this dissertation states that it is possible to integrate fault detection, isolation and identification for hybrid systems in noisy environments in an efficient way. This can be achieve extending PCs to hybrid systems and using DBNs to perform FDII.

This main hypothesis can be splited in some particular hypotheses:

- A diagnosis architecture for hybrid systems to work with parametric (continuous) and discrete faults in a uniform framework is needed. This architecture must support fault detection, isolation and identification, not always considered in previous proposals.

- Dynamic Bayesian Networks, that are a good tool combined with the Particle Filter inference algorithm, can be used in a common framework for all diagnosis tasks. They can also work with uncertainty in the systems, because they can deal with the uncertainty in the measurements (noise) and in the system state. They can also be adapted to deal with uncertainty in the system model (unkonwn parameters).

- The complexity of tracking, fault isolation and fault identification in real-world hybrid systems can be reduced extending the PCs to hybrid systems to efficiently generate the PCs for the discrete configurations of the system.
- Computational cost of Particle Filter inference in DBNs can be reduced by using a Possible Conflicts approach to factorize the system DBN model into smaller (minimal) model fragments and avoiding to include discrete nodes in the network, without losing accuracy.

1.3 Main Goals

The hypotheses stated in the previous section leads to several objectives which can be splited in different tasks. There is another objective which will be helpful to validate the techniques developed using a real complex hybrid system.

1. Factorize efficiently Dynamic Bayesian Networks using Possible Conflicts for fault diagnosis in continuous systems. This will be done developing a method to automatically derive DBNs from PCs.
2. Develop the theory for Hybrid Possible Conflicts (HPCs).
3. Define how to perform fault detection and fault diagnosis with HPCs.
4. Extend the method to derive DBNs from PCs for hybrid systems.
5. Build a diagnosis architecture for hybrid systems based on the DBNs-HPCs approach. This architecture will include discrete and continuous (parametric) faults in a common framework. It will also integrate fault detection, isolation and identification.
6. Test the diagnosis architecture in a real-world hybrid system. Several running examples will be also used to test the different proposals and to clarify definitions. A real system from the aerospace domain will be included in the dissertation to test the accuracy of the developed framework.

A diagnosis architecture for fault detection, isolation and identification of continuous and hybrid systems will be created with the method proposed during the dissertation to efficiently derived DBNs from PCs (HPCs) and it will be evaluated with a real system from the aerospace domain, as it has been said in the last objective. The goals have been proposed in order to design the intended unified architecture, that would confirm our main hypothesis.

1.4 Organization

This dissertation is organized as follows. Chapter 2 presents the state of the art on diagnosis of continuous and hybrid systems, as well as on Dynamic Bayesian Networks. It summarizes the recent research that is being done on fault diagnosis of continuous and hybrid systems in the DX and FDI communities, the two main approaches to Model-based diagnosis. It also focuses the recent work with probabilistic methods in fault diagnosis.

Chapter 3 explains some of the contributions of this dissertation. It deals with continuos systems fault diagnosis. First of all it introduces the Possible Conflicts compilation technique and it briefly explains the DBNs theory. After that, we present the method to derive minimal DBNs from the PCs model and the diagnosis framework for continuous systems that allow to perform fault detection, isolation and identification of continuous systems using DBNs as the modeling tool in all the stages. The last part of the chapter is a case study, a 12^{th} order electrical circuit, that has been used to test the diagnosis architecture previously proposed. During the case study, it is also introduced a method to improve the fault identification stage merging two or more minimal DBNs.

Chapter 4 introduces and describes the Hybrid Possible Conflicts technique, that is the extension of Possible Conflicts for hybrid systems. This chapter also characterize how to perform fault diagnosis of hybrid systems. Considering that hybrid systems have faults in their parameters (continuous faults) and

faults due to its hybrid nature (discrete faults). The chapter also presents a diagnosis framework for fault diagnosis of hybrid systems considering all the diagnosis stages for discrete and parametric faults in a unified way using DBNs as the modeling tool.

Chapter 5 shows the Reverse Osmosis System (ROS) case study. The ROS is a hybrid system from the aerospace field. It has been used to test the diagnosis architecture presented in the previous chapter. The Hybrid Possible Conflicts decomposition of the system, as well as the diagnosis results for discrete and parametric faults are explained in this chapter.

Finally, Chapter 6 summarizes the main contibutions and presents the conclusions of this dissertation. During this dissertation we have pointed out the weaknesses of our proposals and contributions and at the end of this chapter we propose some future work lines to improve the current work.

Chapter 2
State of the Art

This dissertation is included in the Model-Based Diagnosis (MBD) framework. Moreover, it is related to the DX community, which uses Artificial Intelligence techniques for fault diagnosis. This Chapter goes through the main contributions appeared in the different communities of the MBD field: FDI, DX and BRIDGE.

2.1 Background

Diagnosis is a general concept that can be found in many different fields in the society. Illnesses diagnosis in medicine is a well-known example. Doctors analyse pacients' symptoms and they try to figure out which illness each pacient has. Diagnosis can be defined as the process to find the cause of some kind of deviation in the expected behaviour of the considered system.

The diagnosis of physical systems is a particular case included in the previous definition. Physical systems appear everywhere around us. They almost manage some tasks in our lives so it is very important that they work properly. A system can misbehave and that have to be noticed before it risks people's life.

According to Isermann and Ballé [61] a fault is "an unpermitted deviation of at least one characteristic property or parameter of the system from its acceptable/usual/standard condition". They also define the concept of failure as "a permanent interruption of a system ability to perform a required function under specified operating conditions" [61]. Few years later, Console presents a definition of diagnosis: "Diagnosis is the task that given a system and a set of observations corresponding to abnormal behaviour, determines what is wrong with the system, in order to re-establish the system normal behaviour" [32].

There are many different approaches to physical systems diagnosis: expert systems, machine learning or model-based diagnosis among others[121, 122, 123]. This dissertation is focused on Model-based Diagnosis (MBD).

Diagnosis is an iterative process with three stages [14, 60]:

- **Fault detection:** Decide whether there is a fault or not. The time when the fault happens is also given after this stage.
- **Fault isolation:** The faulty component is found.
- **Fault identification:** The task in this stage is to find out the fault type and estimate its scale.

There are some authors, like Gertler [54], who divide the diagnosis proceses only in two stages. They consider fault isolation and fault identification to be part of the same stage, the diagnosis stage.

There are different fault classifications, depending on the criteria chosen for the sorting.

According to the model process [54] there are two types of faults:

- *Additive faults:* They are unknown inputs to the system model. They usually have zero value but when it changes to a value different from zero they modify the output in the system (e.g. leaks in a tank).
- *Multiplicative faults:* They represent changes in some parameters in the system model. The change they produce in the output of the system is related to the input values (e.g. blocks in pipes or power loss).

Looking at the time dependence there are three main groups of faults [14, 60]:

- *Abrupt faults:* They appear suddenly and its scale remains the same along the time.
- *Incipient faults:* The fault magnitude changes gradually along time.
- *Persistent or intermittent faults:* Fault effects are the same along the time or they appear and desappear, respectively.

Finally, there is another classification for faults according to the faulty components [14].

- *System faults:* These faults modify the dynamic properties in the system.
- *Sensor faults:* These faults do not affect the system properties but sensor signals are not correct.
- *Actuator faults:* These faults do not modify the system properties but the controller action is not as expected.

2.2 Model Based Diagnosis (MBD) of continuous systems

Model-based Diagnosis [54, 56, 36, 100] compares the observed behaviour of a system with the expected behaviour derived from a model (Figure 2.1).

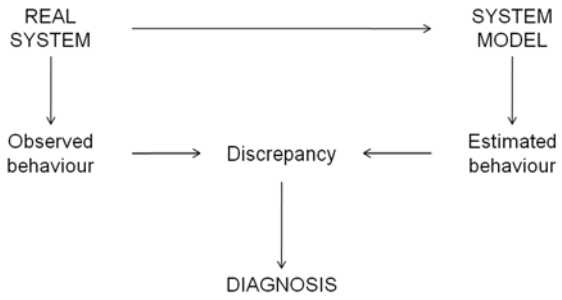


Figure 2.1: Model-based Diagnosis.

MBD has some advantages and disadvantages. On one hand, it is experience and device independent. It can be used with multiple faults scenarios. Model libraries can be created to help reusability and maintenability. On the other hand, accurate models are needed to perform diagnosis in a precise way, this is the main disadvantage. There may be other problems related to the computational cost while deriving possible diagnosis associated with an observed fault.

The main element in MBD is the model. B. Kuipers [71] defines a model as "a (small) finite description of an infinitely complex reality, constructed for the purpose of answering particular questions".

Traditionally, there have been two different communities developing Model-based Diagnosis techniques: (1) FDI community (Systems Dynamics and Control Engineering community) [91, 60, 54, 88], and (2) the Artificial Intelligence Diagnosis (DX) community [100, 56]. Recently, a third community (BRIDGE), has appeared. BRIDGE [12, 33] establishes a common framework to put together results and techniques from the FDI and DX communities. Those three approaches to Model-based Diagnosis of dynamic systems use different kinds of models, and different assumptions concerning robustness of the generated solution regarding disturbances, modeling errors, and noise.

2.2.1 FDI approach

FDI approach is related to Modern Control Engineering. This community has a solid theoretical and mathematical fundamentals. FDI uses mainly analytical numerical models.

FDI performs diagnosis using residuals. A residual can be defined as a fault sign based on a deviation between measurements and values derived from the model [61]. Perfect models and ideal systems will get zero residuals in a non faulty system. Residuals will be different from zero in case a fault occurs. In the real world, systems are not ideal and models are not perfect so there are techniques to decide whether a residual is detecting a fault or the system is in its nominal behaviour.

Residuals in FDI are built using numerical methods producing robust results in presence of noise or external disturbances, as well as modeling errors.

The proccess to use residuals has three steps:

- **Residual generation.** Residual expressions are derived from analytical redundancy in the system and they are generated offline. There are two main categories of residual generation techniques:
  - State estimation: It estimates the state variables of the systems to find discrepancies with the observed behaviour. This approach can be tackled with parity space [30, 54, 110], or observer techniques [49, 87, 88].
  - Parameter estimation[59, 91, 54]: This technique identifies the parameters in the reference model of the system.
- **Residual evaluation.** Once residuals have been generated, an online evaluation process is run using the measurements of the system.
- **Decision procedure.** Once residuals have been evaluated, fault detection is made using some kind of decision logic. As it has been previously said, we will never have neither perfect models nor non-noisy measurements, so residuals will never be zero and some kind of statistical study has to be used to determine whether residuals are equal or different to 0.

  Several authors [69, 68, 92, 52] have proposed extensions to improve the fault detection and isolation stages using additional information (sign, magnitude, delays,...) about the residual signals. These extensions are hard to obtain automatically. The Qualitative Fault Signatures approach by Mosterman and Biswas [76] uses Temporal Causal Graphs (TCGs) to automatically derive the fault signature matrix of a system including temporal information (known as the Qualitative Fault Signature Matrix, QFSM).

2.2.2 DX approach

DX community uses Artificial Intelligence techniques in fault diagnosis. The most common approach among DX community is Consistency Based Diagnosis (CBD), it is associated with its computational paradigm, the General Diagnosis Engine (GDE) [56].

There are several advantages about CBD [37, 114]: (1) It only uses knowledge about system structure and behaviour, (2) it can deal with different types of faults, (3) it can also deal with fault scenarios never seen before, and (4) it is capable to automatically handle multiple-fault diagnosis scenarios.

A DX approach techniques classification is shown in Figure 2.2 [32].

In 1987, Reiter [100] fixed the basis for diagnosis using a logic point of view. That was the theoretical frame for the CBD for static systems. CBD theory is solid for static systems [100, 56, 39, 40] and it is still used. This theory is known as First Principles Diagnosis [37], the available information for the diagnosis is the system descripcion (a logic model of the system) and its measurements. According to the main idea [46, 33], given the system description (SD) and measurements (OBS) which are non consistent with the expected behaviour of the system, the problem is to determine which components in the system ($c \subseteq COMPS$[1]) restore consistency to the set of the system and the measurements when they are assumed not to be operating normaly (ab(c)).

There are three basic concepts in CBD: *system*, *symptom* and *conflict* [100, 39]:

Definition 1. *(System). A system is a triple (SD, COMPS, OBS) where: (1) SD, the system description, is a set of first-order sentences; (2) COMPS, the system components, is a finite set of constants; (3) OBS, the set of observations, is a set of first-order sentences.*

[1] $COMPS$ is the set of components. It is a set of logical constants.

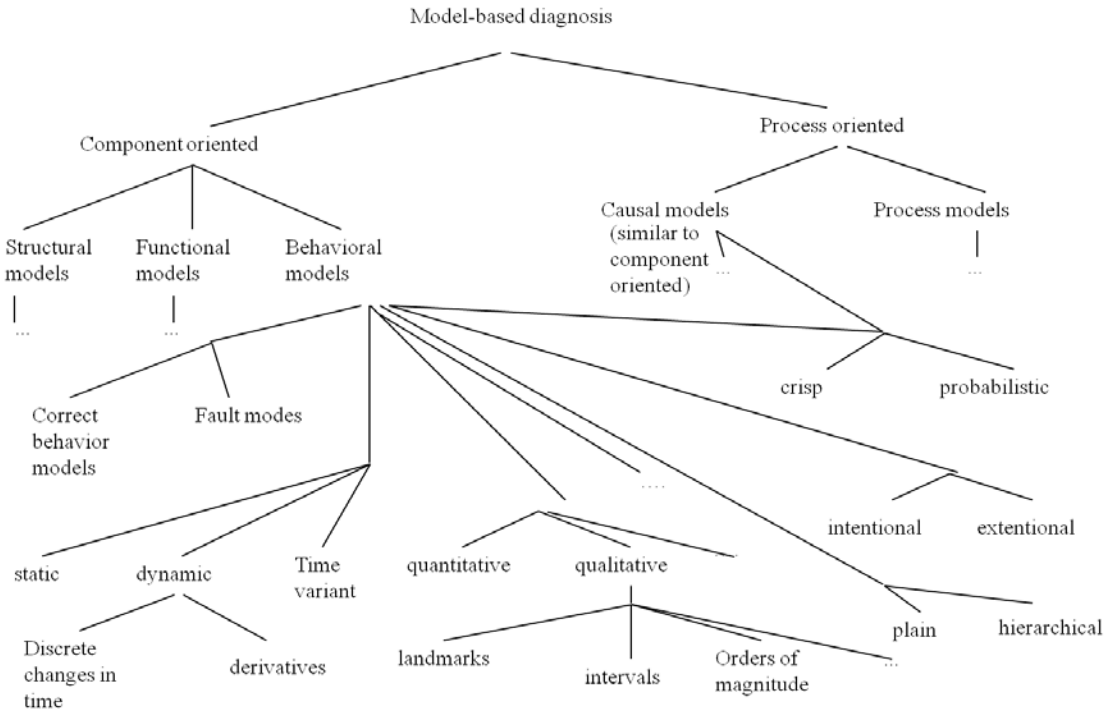


Figure 2.2: Simplified classification for the DX community techniques.

Definition 2. *(Symptom). A symptom is any difference between a prediction made by the inference procedure and an observation.*

Definition 3. *(Conflict). A conflict is a set of correctness assumptions for components which underly a symptom.*

Conflicts provide an intermediate step between symptoms and diagnosis. To define the concept of diagnosis, the literal ab(·) has to be defined. The standard MBD convention [39] states that given a component $c \subseteq COMPS$ that is abnormal, ab(c) is a literal which holds.

A diagnosis is [39] a particular hypothesis for how the system differs from its model, i.e., the diagnosis specifies whether each component of a system is faulty, ab(·), or is working properly,¬ab(·). Formally, a diagnosis can be defined as [39]:

Definition 4. *(Diagnosis). Let $\Delta \subseteq COMPS$. A diagnosis for (SD, COMPS,OBS) is a set of components Δ such that the following is satisfiable:*

$$SD \cup OBS \cup \{[\Lambda_{c \in \Delta} ab(c)] \wedge [\Lambda_{c \in COMPS - \Delta} \neg ab(c)]\}$$

The set of (minimal) diagnoses[2] of a system can be characterized using the set of (minimal) conflicts[3]. This characterization is based on the minimal hitting set definition [33, 100]:

Definition 5. *(Hitting set). A hitting set for a collection C of sets is a set $H \subseteq \cup\{S|S \in C\}$ such that $H \cap S \neq \emptyset$ for each $S \in C$.*

Definition 6. *(Minimal hitting set). A hitting set is minimal if and only if no proper subset of it is a hitting set for C.*

And then, the (minimal) diagnosis for a system can be computed from the set of (minimal) conflicts [33]:

[2]A diagnosis is minimal if does not exist any $\Delta' \subset \Delta$ such that Δ' is a diagnosis [33].
[3]A minimal conflict is a conflict which does not strictly include (set inclusion) any conflict [33].

Theorem 1. *Δ is a (minimal) diagnosis for (SD, COMPS,OBS) if and only if Δ is a (minimal) hitting set for the collection of (minimal) conflicts for (SD, COMPS,OBS).*

There is not a common framework for Consistency Based Diagnosis of dynamic systems [39, 117]. There are some works showing different DX techniques used in this field [120].

Finally, it is important to point out that the basic approach of CBD uses only correct behaviour models, but this simple approach can be extended to include fault modes [115]. This approach is known as Consistency-based Diagnosis with fault modes. It requires a priori description of the set of faults that can occur in the system. This approach allows to know how the system fails, but lacks one of the main advantages of CBD, the ability to deal with faults never seen before unless they somehow characterize unknown faults as proposed by Struss et al. [116]. Recently, several authors, like [125], have proposed extensions for unknown fault mode diagnosis in hybrid systems.

The main concerns in DX have been isolation and identification stages in the diagnosis process. Qualitative or semiqualitative models have been used to deal with uncertainty in the models. The work developed during the last 20 years has been focused on the improvement of the characteristics described in GDE and the basis theory:

- *Diagnosis in dynamic systems*: The dynamic information in the models develops computational problems in the ATMS (the system used to compute minimal conflicts recording dependencies). The problems are mainly when recording dependencies online. There are three main solutions proposed to this problem:
  - Modify the ATMS to avoid problems when recording dependencies online and temporal information. SIDIA [55] and Magellan-MT [44] implement this solution.
  - Use forward propagation to predict the system behaviour. Propagate backwards to find the dependencies only if a discrepancy has been found. This task is usually done following a causal direction: CAEN [119], DYNAMIS [29] or TRANSCEND [76].
  - Compile dependencies offline. Using an approach similar to the FDI structural analysis, the set of components or subsystems which can become a conflict are calculated offline. Those subsystems estimate online variables behaviour looking for a discrepancy [73, 127, 93]. In this dissertation, a technique using this approach is being used, Possible Conflicts [95].
- *Diagnosis in complex systems: There are two main branches working on the diagnosis of complex systems:*
  - Use a hierarchy to decompose the model of the system [38, 2].
  - Reason at different levels of abstraction keeping the results when changing from one level to another [113, 28].
- *Diagnosis with fault modes:* A great advantage of CBD is that it only need models of the non faulty behaviour of the system to perform fault detection and isolation [56]. This can end in a problem, because, once the fault is isolated, there is not enough information to know which component is faulty [45].

  If there is knowledge about how the system fails, i.e. fault modes, and we have some kind of predictive models for those fault modes, then we can use CBD with fault modes. A choice to solve this problem is to reject a fault mode when the estimation of the abnormal behaviour of the system is not consistent with the measurements. Only the consistent modes remain [46, 56]. There are several ways to introduce information about the fault modes:
  - *Non predictive approaches:* They have few estimation ability related to the fault modes. There are some examples: *physical imposibility* [48], it describes a physical impossible behaviour, or *non intermittent* [99], it takes advantage over the information about the intermittent faults.
  - *Predictive approaches:* They use models of the fault modes to estimate the system behaviour when there is a fault. i.e.: *Sherlock* [66] or *GDE+* [116]. Based on the estimation, the inconsistent fault modes are rejected. A fault mode is accepted when all the other modes have been rejected and there is not any unknown fault mode.

Summarizing, unfortunately there is no general architecture suitable for all kind of systems.

2.2.3 BRIDGE approach

The BRIDGE community [12], based on the work by Cordier et al. [33] provides a common framework to researchers from both fields for sharing and combining results and techniques, as it has been previously introduced. The basis for this approach is the comparison of two Model-based Diagnosis techniques (from the two previous approaches): Consistency-based Diagnosis using conflicts [100], and fault detection and isolation with Analytical Redundancy Relations (ARRs) obtained through structural analysis [111, 14].

Both communities have the same common principles but they have different concepts, assumptions, and techniques [33].

The principles they share are:

- The diagnosis process relies on an explicit model of the nominal system behaviour.
- Faults are detected as inconsistencies between the observations and the behaviour predicted by the model.
- Fault isolation stands on interlinking the sets of components which underly every detected inconsistency: conflicts in DX and support for residuals or ARRs, in FDI.

Although Possible Conflicts [95] were developed within the DX approach, and were designed to be equivalent to compiled conflicts, it has been demonstrated that they are also equivalent to minimal ARRs from the FDI approach [96]; hence, PCs perfectly fit in the BRIDGE framework too. It states the basis to develop a diagnosis framework [19]. Additionally, TRANSCEND [76] was introduced in the DX field, but it actually fits within the BRIDGE approach. Recently, PCs and TRANSCEND were combined in the same diagnosis approach [22, 19].

The BRIDGE community is developing a common framework combining ideas and results from FDI and DX approaches, i.e. state observers and conflicts, respectively [96, 16, 19]; or using qualitative methods (DX) to improve the isolation process (FDI).

2.2.4 Probabilistic Methods in MBD

Probabilistic methods can be included in MBD [56, 83].

There are some probabilistic methods already used in diagnosis algorithms [66, 81, 72, 103, 105, 102, 89]. Probabilistic theories can deal with uncertainty related to hypotheses and measurements.

The main problem in the probabilistic diagnosis is how to define which fault has occured according to the system measurements. The reasoning is similar to the expert human reasoning: An expert can know which symptoms appear with which probability when a fault in a system occurs.

The Bayes Theorem is a basic method to dignose faults with uncertainty in symptoms and faults themselves [107]. Assuming that $Symptom$ and $Fault$ are two random variables, the posterior probability of $Fault$ given $Symptom$ ($P(Fault|Symtom)$) can be derived using the causal information and the prior probabilities: $P(Symptom|Fault)$, $P(Symptom)$ and $P(Fault)$, respectively.

$$P(Fault|Symptom) = \frac{P(Symptom|Fault)P(Fault)}{P(Symptom)}$$

There are graphical models, like Bayesian Networks (BNs), and Dynamic Bayesian Networks (DBNs) [81], which model uncertainty explicitly. Above all, they also represent graphically an efficient factorization of the joint distribution of the variables in the model. This can be possible because these models contain not only the causal dependencies among variables, but also the independencies. This is what allows a feasible diagnosis.

BNs assume a static state for the system, they cannot model dynamic states and trasitions among different states. DBNs use temporal information. This dissertation uses DBNs to apply CBD techniques in dynamic systems.

Dynamic Bayesian Networks (DBNs)

Dynamic Bayesian Networks (DBNs) are a suitable tool to model complex dynamic systems with uncertainty [81]. DBNs are an acyclic graph representing a probability model of a dynamic system with discrete time. Nodes are variables in the system. There are two types of edges: (1) Causal dependencies between nodes in the same time stamp, and (2) Dependencies or relations between nodes in two different time stamps. If there is not an edge between two nodes, it is because each one is conditionally independent from the other.

Assuming a First Order Markov Model, the number of time stamps needed to define the temporal behaviour of the system is smaller. Using two time stamps it is possible to represent the whole temporal behaviour.

There are DBNs with discrete random variables whose prior and conditional probabilities are tables. There are also continuous systems, where all the probabilities (prior and conditional) are probability functions. Graphical methods do not impose restrictions over probability distributions or sensor noise [81]. Moreover, they are a compact way to represent systems.

Learning process in MBD can be performed deriving the net structure and the parameter values from models [72, 103]. Particle Filter is the inference aproximate technique which is going to be used in this dissertation to improve the problem with the high computational cost.

R. Dearden and D. Clancy [41] applied the DBNs modeling and diagnosis approach to a planetary rovers, which is a hybrid system (continuous behaviour in several discrete working modes). They implemented a single DBN modeling the whole system including the different working modes. I. Roychoudhury et al. [103] have applied this probabilistic approach to a continuous system from the hydraulic field, they model the whole system whith a DBN to track the system behaviour but when a fault is detected, after generating the set of fault candidates they build a different DBN for each candidate and they simulate them all in parallel to identify the right candidate.

DBNs are useful to estimate the state variables evolution without assuming gaussian distribution for noise and model errors, which is not usually true when a fault appears [8]. The main disadvantage is that they have a high computational cost for inference and learning processes. DBNs have been widly used for FDD [67, 41]: for instance, in hydraulic systems [72, 103, 102], planetary rovers [41], robots [124] and electrical circuits [102], among other fields.

From a FDII point of view, DBNs can also be used for fault identification, because they can find the model which best fits the system behaviour and they allow rejecting the other fault hypotheses, as they do not match the system behaviour. They are useful in fault isolation and identification even when faults have diverse magnitudes.

Particle Filter Particle Filter [81] is an approximate inference method which can be used for real time inference with DBNs.

A Particle Filter algorithm approximates the system state with a number of particles. Those particles are sampled from a distribution and according to the system observations, it assigns a weight to each of them. The final state estimation is the weighted average of all the particles.

With this method, we can use non linear models with random prior probabilities. The working effort to obtain an estimation can be adjusted using a different number of particles.

The Particle Filter algorithm has a weakness, that is called *sample impoverishment* [124, 41]: samples with less weight usually disappear in few iterations. This is a serious problem in diagnosis tasks, because faults are usually represented in less probable (smaller weight) samples.

There is a modified PF algorithm known as *Importance Sampling* [41] that is going to be used in this dissertation. The *Importance Sampling* algorithm can be used in fault isolation and identification tasks with a single DBN modeling all possible faults [85] or with a separate DBNs for each fault candidate [103]. It was more common the use of a single DBN but it turns out that it gets a more complex DBN. The single DBN modeling all possible faults is also more sensitive to sample impoverishment because the differences between the complete DBN and the actual DBN modeling the system can be huge. The work done in this dissertation will use multiple DBNs, one modeling each fault candidate, as it is explained by I. Roychoudhury et al. [103]. At first, the DBN modeling the nominal (non faulty) system will be run until a fault will be detected. At that time, DBNs modeling all single fault hypotheses will be started to perform fault isolation and identification tasks until ideally all the hyptheses, except one, will be rejected.

The main problem with the approach described in the previous paragraph is that each DBN models the whole system. For complex systems, the computational load can be so high that all DBNs modeling the different fault hypothesis cannot be simulated at the same time. This problem will be tackled using Possible Conflicts to factorize the system and its DBN.

There are some approaches to avoid the sample impoverishment problem: (1) *RSPF (Risk-sensitive Particle Filter)*, and (2) *VRPF (Variable-resolution Particle Filter)* [124]. *RSPF* uses a cost model to calculate the cost of deriving each particle, faulty states have a high cost, so this algorithm assure that the generated particles represent that state, even if they have low probability. *VRPF* uses virtual or abstract particles including similar states on each particle. It allows a small number of particles, so that, each one represents a big state space part when the likelihood to belong to that part of the state space is small. When the likelihood of the states belonging to the same virtual particle grows, particles will be redefined to represent individual states.

2.3 MBD of Hybrid Systems

Hybrid systems' behaviour is made up of continuous and discrete event dynamics. Those systems have different working modes with a continuous behaviour on each of them. Discrete events are responsible for changes between modes. There is a different model for each working mode and the hybrid system will be switching between them according to the triggered events. Complex hybrid systems can be found in a large range of engineering applications (i.e. mechanical systems, electrical circuits, embedded computation systems). Anti-lock braking systems (ABS) or a plane fueling system, among others, are some actual examples.

Model-based online diagnosis methods require quick and robust reconfiguration processes when a mode change occurs. A discrete event will trigger the change to another working mode, so the reconfiguration process is needed to build the models for the new working mode. The ability to keep on tracking the nominal behaviour of the system during transitory states is also required.

To introduce hybrid systems we will present first the Discrete Event Systems (DES) where observations and control actions are essentially discrete: on-off components, discrete alarms, etc. Those systems abstract time to points where an event occurs. There are well known methods to diagnose DES. Event based DES diagnosis is built on the diagnoser proposal of [109]. System components are modeled by an automata and the whole system model is obtained by the synchronous composition of the individual automata. The diagnoser is obtained by reducing unobservable events. The diagnoser tracks sequences of observable events and, if possible, links them to unobservable fault events. The key issue to DES diagnosis is to achieve the right automata description level, to limit exponential growth of the diagnoser and still obtain precise diagnosis.

The systems we focus on have continuous behaviour commanded by discrete events.

Hybrid systems modeling and diagnosis have been approached by the FDI, as well as the DX communities during the last 15 years. In the FDI field several approaches have been developed to diagnose hybrid systems [31] or quantized systems [74]. Meanwhile, in the DX field, different proposals have been made based on hybrid modeling [76, 83], hybrid state estimation [58, 101], or combination of online state tracking and residual evaluation [11, 9]. In both communities, the solution requires to somehow model and eventually fully or approximately estimate the set of possible states, and to diagnose the current set of consistent modes. Both steps are computationally very expensive or infeasible for complex systems.

Some researchers use different kinds of automata to model the complete set of modes, and transitions between them. In those cases, the main research topic is hybrid system state estimation due to uncertainties related to either the model parameters or the measurements or both. State estimation can be done using probabilistic (some kind of filter [75, 70] or hybrid automata [58]) or set-theoric approaches [11].

An alternative is to use an automaton just to follow the system mode, but use another approach to diagnose the continuous behaviour: a set of ARRs for each mode [9], or parameterized ARRs for the complete set of modes[10], for instance.

Typically, hybrid systems are modeled as hybrid automata, pre-enumerating all the system modes explicitly, even if some of these modes will never be visited during execution. Moreover, the hybrid automata approach works well for small systems, but incurs in high computational (and space) costs for large systems.

To solve the first problem, some authors have proposed alternative ways for hybrid systems modeling, like hybrid bond graphs [77], where modes in the system do not have to be previously enumerated. There are two main approaches: (1)those who use switching elements with fixed causality, and (2)those who use ideal switching element that change causality on switching.

The first set of approaches model systems with a fixed causal structure and varying parameters at the switching instants. These varying parameters are modeled using different bond graph components, such as sources [15], transformers [47, 42], or storage elements [50]. The great advantage of these approaches is that they do not require the causality reassignment process when a mode changes. Anyway, these approaches implies large variations in the parameters to cover all the cases and the modelled system can be stiff. Solutions for this last problem imply having complex integration methods with large simulation time requirements, what makes it difficult to apply for online simulation.

The second set of approaches consider ideal switching elements whose causality is modified when a mode change occurs. Hence, parameter values are fixed in this approach, avoiding some problems of the previous ones. However, these approaches need a causality reassignment procedure to have a consistent causal assignment for the entire bond graph model [84], what also makes this approach difficult to apply for online simulation. In [106] the mechanisms to determine the bonds and elements of hybrid bond graph models whose causality assignments are invariant across system modes are proposed. Then, this information is used to derive space-efficient reconfigurable block diagram models that may be reconfigured efficiently when mode changes occur. In [106] is has been proven that this solution can be used for efficient simulation of hybrid systems.

As it has been previously explained, there are some works done with hybrid systems. Some problems related to hybrid systems have also been solved, at least partially. The main concern for hybrid systems diagnosis is the integration of parametric [84, 129] and discrete faults [58, 126]. There are few approaches proposing a unified method [31, 35]. This dissertation proposes an approach to deal with this issue in an efficient way, based on the structural behaviour information provided by PCs. Next Chapter introduces and summarizes every essential concept and technique for our diagnosis framework.

Chapter 3

Minimal Dynamic Bayesian Networks for continuous systems Consistency-Based Diagnosis

Possible Conflicts (PCs) are introduced as a compilation technique from the Consistency Based Diagnosis (CBD) community. PCs are a useful tool for fault diagnosis of continuous systems because they provide the minimal computational model for a residual. On the other hand, Dynamic Bayesian Networks (DBNs) are a probabilistic modeling technique for dynamic systems robust to the presence of noise in the observations. The main inconvinient of DBNs is its high computational cost. Its main advantege for diagnosis tasks is that they allow performing fault detection, isolation and identification with the same formalism. This Chapter presents a method to derive DBNs from PCs, as well as a fault diagnosis framework for continuous systems that uses those DBNs. Minimal DBNs reduce computing time of the DBN of the complete system and provide CBD candidates without additional machinery. A merging strategy of minimals DBNs is also proposed to improve fault identification. As a running example, we employ a simple three-tank system to introduce the main concepts. Then, a 12^{th} order electrical system is used as a case study.

3.1 Possible Conflicts (PCs)

Possible Conflicts [95] are a model-based compilation technique that allows to perform online CBD of dynamic systems without a dependency recording engine. It has been shown to be equivalent to other FDI model compilation techniques such as minimal ARRs or MSOs [7]. Basically, PCs identify minimal redundant subsystems. These subsystems are described by a set of equations that, among other properties: (1) have analytical redundancy (this is needed to perform the fault diagnosis) and (2) are minimal in the sense that no proper subset of equations has analytical redundancy. PCs are just over-constrainted sets of equations as they fulfill 1) and 2).

Figure 3.1 shows the three-tank system which is going to be used to clarify some definitions. In this system, there is an input flow (f_1) to tank T1. Each tank has an outflow at the bottom (f_4, f_{10} and f_{16} respectively for T1, T2 and T3). Moreover, there are flows connecting T1 with T2 (f_6), and also T2 with T3 (f_{12}). In the model, the pressure in each tank, p_2, p_8 and p_{14} respectively, is computed along time based on mass balances and tank capacitances: C_1, C_2, C_3. Finally, flows through pipes are modeled in terms of pressure differences and pipe resistances R_1, R_{12}, R_2, R_{23} and R_3 for flows f_4, f_6, f_{10}, f_{12} and f_{16} respectively.

The state space equations and the observational model for the system are presented below (from eq.1 to eq.8):

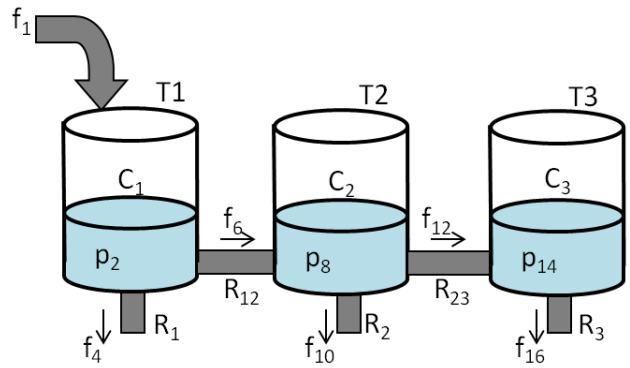


Figure 3.1: Three-tank system.

$$\dot{p}_2(t) = f_1(t) - f_6(t) - f_4(t) \quad \text{(eq.1)}$$
$$\dot{p}_8(t) = f_6(t) - f_{12}(t) - f_{10}(t) \quad \text{(eq.2)}$$
$$\dot{p}_{14}(t) = f_{12}(t) - f_{14}(t) \quad \text{(eq.3)}$$
$$f_4(t) = \frac{p_2(t)}{R_1} \quad \text{(eq.4)}$$
$$f_6(t) = \frac{1}{R_{12}} \cdot (p_2(t) - p_8(t)) \quad \text{(eq.5)}$$
$$f_{10}(t) = \frac{p_8(t)}{R_2} \quad \text{(eq.6)}$$
$$f_{12}(t) = \frac{1}{R_{23}} \cdot (p_8(t) - p_{14}(t)) \quad \text{(eq.7)}$$
$$f_{16}(t) = \frac{p_{14}(t)}{R_3} \quad \text{(eq.8)}$$

In the three-tank system, we have also the following set of observations (diagnosis observational model):

$$f_1(t) = F_{in} \quad \text{(eq.9)}$$
$$f_4(t) = F_1(t) \quad \text{(eq.10)}$$
$$f_6(t) = F_{12}(t) \quad \text{(eq.11)}$$
$$f_{16}(t) = F_3(t) \quad \text{(eq.12)}$$

Finally, just to illustrate PCs computation, we include the integration step relating the state variable between consecutive time steps (differential constraint in integral causality in DX terminology [97]).

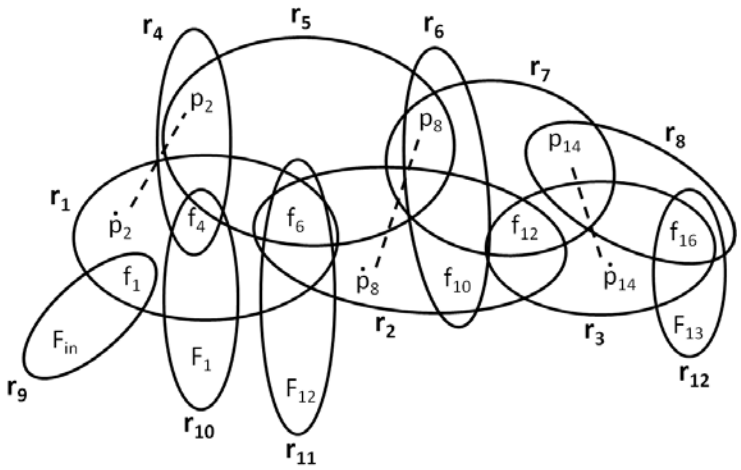


Figure 3.2: H_{SD} of the three-tank system.

$$p_2(t) = p_2(t-1) + \int_{t-1}^{t} \dot{p}_2(t)dt \quad \text{(eq.13)}$$

$$p_8(t) = p_8(t-1) + \int_{t-1}^{t} \dot{p}_8(t)dt \quad \text{(eq.14)}$$

$$p_{14}(t) = p_{14}(t-1) + \int_{t-1}^{t} \dot{p}_{14}(t)dt \quad \text{(eq.15)}$$

Possible Conflicts method works with an abstract representation of a system, obtained describing the structure of the Input/Output equations[1] and its variables with a hypergraph. Hence, system description is the hypergraph $H_{SD} = \{V, R\}$, where V is the set of variables of the system and $R = \{r_1, r_2, \ldots, r_m\}$ is a family of relations in V, where each r_k represents a relation in the model and its elements are the variables that occur at equation r_k. For instance, from equation eq.1 , $\dot{p}_2(t) = f_1(t) - f_6(t) - f_4(t)$, we obtain the relation $r_1 = \{\dot{p}_2, f_1, f_4, f_6, \}$. Figure 3.2 shows the H_{SD} of the three-tank system.

Figure 3.2 also shows the two types of relations managed in the PCs framework: *instantaneous* and *differential* relations. Instantaneous constraints model static relations, like those modeled by algebraic equations. They are represented by continuous arcs in the hyper graph. Differential constraints model dynamic relations and are limited to binary relations between a variable and its first derivative. They are represented by dashed lines and their label is not shown in MECs figures to simplity the figures. They abstract the equations eq.12, eq.14, and eq.15 of the three-tank system.

First step on PCs computation consists of finding all the *Minimal Evaluation Chains, MEC*, contained in H_{SD}. $MECs$ are denoted $H_{ec} = \{V_{ec}, R_{ec}\}$, where $V_{ec} \subseteq V$, $R_{ec} \subseteq R$. $MECs$ are minimal over-constrained subsystems. The existence of a MEC is a necessary condition for analytical redundancy to exist. $MECs$ have the potential to be solved using only local propagation (i.e. solving one equation in one unknown) from the measurements to the subsystem output. Additionally, each MEC identifies, by definition, a subgraph of H_{SD}. There are 3 $MECs$ in the three-tank system. Figures 3.3, 3.4 and 3.5 show the $MEC1$, $MEC2$ and $MEC3$ of the system. These $MECs$ are defined by:

- $H_{MEC_1} = \{\ \{\dot{p}_2, p_2, f_1, f_4, f_6, F_{in}, F_1, F_{12}\},\ \{r_1, r_4, r_9, r_{10}, r_{11}, r_{13}\}\ \}$
- $H_{MEC_2} = \{\ \{\dot{p}_2, \dot{p}_8, \dot{p}_{14}, p_2, p_8, p_{14}, f_1, f_4, f_6, f_{10}, f_{12}, F_{in}, F_1, F_{12}F_{13}\},\ \{r_1, r_2, r_3, r_5, r_7, r_9, r_{10}, r_{11}, r_{12}\}\ \}$

[1]Possible Conflicts are derived from Input/Output equations, but those can be easily obtained from the state space equations.

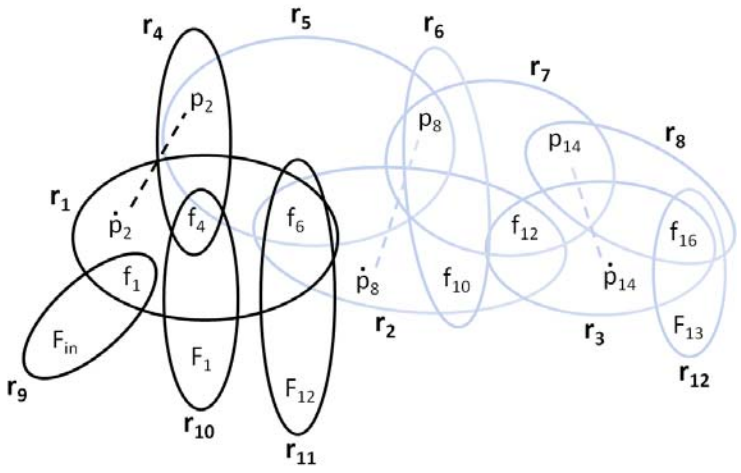


Figure 3.3: MEC1 of the three-tank system, represented as solid hyperarcs in H_{SD}.1.

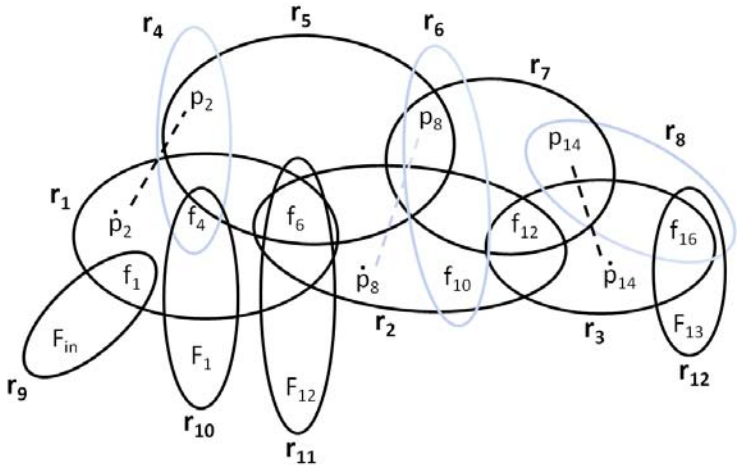


Figure 3.4: MEC2 of the three-tank system, represented as solid hyperarcs in H_{SD}.2.

- $H_{MEC_3} = \{\ \{\dot{p}_8, \dot{p}_{14}, p_8, p_{14}, f_6, f_{10}, f_{12}, f_{16}, F_{12}, F_{13}\}, \{r_2, r_3, r_6, r_7, r_8\}\ \}$

The three $MECs$ are strictly over constrained. $MEC1$, for instance, has five unknown variables, $\dot{p}_2, p_2, f_1, f_4, f_6$, and six relations: $r_1, r_4, r_9, r_{10}, r_{11}, r_{13}$. They are minimal *w.r.t* set inclusion and they might be solved using only local propagation, as we will show in the next paragraph.

Second step of PCs computation requires adding causal knowledge to assure that a MEC, $H_{ec} = \{V_{ec}, R_{ec}\}$, can be solved using local propagation criterion[2]. If it is possible, a *Minimal Evaluation Model (MEM)* is defined, $H_{mem} = \{V_{mem}, R_{mem}\}$, with $V_{mem} = V_{ec}$ and $R_{mem} = \{r_{1_{k1}}, r_{2_{k2}}, \ldots, r_{m_{km}}\}$. $r_{i_{ki}}$ is a causal constraint obtained assigning a causality to $r_i \in R_{ec}$. MEM are directed hyper graphs that specify the order in which equations should be locally solved starting from measurements and inputs to generate the subsystem output.

Figures 3.6, 3.7 and 3.8 show the $MEMs$ for $MEC1$, $MEC2$ and $MEC3$ respectively. Considering $MEM1$, $H_{mem_1} = \{V_{mem_1}, R_{mem_1}\}$ with $V_{mem_1} = \{F_{in}, F_1, F_{12}, f_1, f_4, f_6, \dot{p}_2, p_2\}$ and
$R_{mem_1} = \{\{F_{in}, f_1\}, \{F_{12}, f_6\}, \{f_1, f_4, f_6, \dot{p}_2\}, \{\dot{p}_2, p_2\}, \{p_2, f_4\}\}$. According to $MEM1$, $\dot{p}_2$ can be estimated from input F_{in} and measurement F_{12} plus the initial value of f_4. If initial value of the state is also known,

[2]It should be reminded that in the DX approach, let's say in GDE, using local propagation allows to isolate every single equation or constraint.

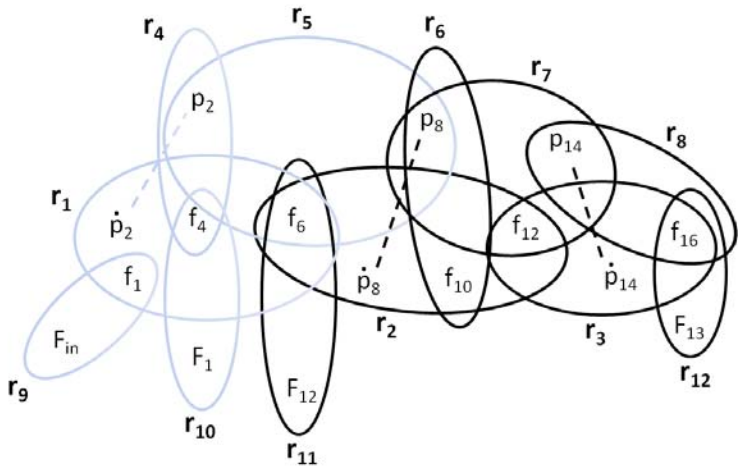


Figure 3.5: MEC3 of the three-tank system, represented as solid hyperarcs in H_{SD}.3.

p_2 can be estimated integrating $\dot{p}_2$. Finally f_4 can be estimated from p_2 and it is also directly measured. Similarly, $MEM2$ allows to estimate f_6 from F_{in} and measurements F_{12} and F_1 and F_3 while $MEM3$ estimates f_{16} from F_{12}. In this example, every causal constraint $r_{i_{ki}}$ impose the causality shown in the initial equation abstracted by r_i. Thus, we have labelled every $r_{i_{ki}}$ in the $MEMs$ figures with the label of the original equation of the system.

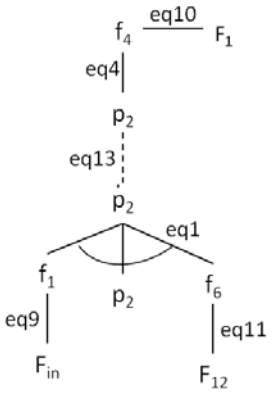


Figure 3.6: PC1 from the three-tank system.

We have been talking about causal assigments and we will see some examples using the three tanks system (Figure 3.1) to clarify what a causal assigment, also called an interpretation, is. Looking at the state space equation eq.5 we can derive the three different interpretations it has depending on which variables in the equation are known and which one is calculated. Those interpretations are shown from equation eq.5-1 to eq.5-3.

$$f_6(t) = \frac{1}{R_{12}} \cdot (p_2(t) - p_8(t)) \quad \text{(eq.5-1)}$$

$$p_2(t) = p_8 + \frac{1}{R_{12}} \cdot f_6(t)) \quad \text{(eq.5-2)}$$

$$p_8(t) = p_2(t) - \frac{1}{R_{12}} \cdot f_6(t) \quad \text{(eq.5-3)}$$

Now, we can show that the $MEMs$ provide a mean to decompose the original system model, $H_{SD} = \{V, R\}$. Each MEM, $H_{mem} = \{V_{mem}, R_{mem}\}$, is uniquely related to the MEC, $H_{ec} = \{V_{ec}, R_{ec}\}$, that originates it. And each MEC identifies, by definition, a subsystem of H_{SD}. Then, the set of $MEMs$ induces a decomposition on H_{SD}. The decomposition is not a partition of the original system, because it is not exhaustive (some variables and relations may not be included in any subsystem) neither exclusive (some relations and variables may belong to various subsystems). However, the decomposition is unique because the set of minimal subsystems with analytical redundancy is unique and the algorithms that compute possible conflicts find every MEC and MEM[3].

Definition 7. *(Possible Conflicts (PCs)). A Possible Conflict is defined as the set of relations in a MEC with at least one MEM.*

Following in detail the definition of PC and the process to derive them [95] it has been shown that PCs fulfill the definition of structural observability in [14, 112]. So, every PC is structurally observable [78].

PCs have been used to decompose the system model and create more efficient simulation or identification tasks [94, 19]. That reason, combined with the fact that they define observable subsystems, makes them a useful tool to derive minimal DBNs. We can use the structure defined in PCs and subsequently, we can derive from that a DBN. The set of equations in a PC define a structurally observable subsystem so we do not need to check the observability of DBNs derived from PCs because they are always structurally observable. Obtaining both the structural and causal models is a capital step in computing PCs. As a consequence, a method to automatically derive both structural and causal models from a graphical description would be a major advantage. For that reason, we have selected Bond-Graphs as our graphical modeling tool.

3.1.1 Deriving PCs from Bond Graph models

Bond Graph (BG) is a domain-independent energy-based topological modeling language for physical systems [65]. A BG is built of primitive elements: storage elements (capacitances, C, and inductances, I), dissipative elements (resistors, R) and elements to transform energy (transformers, TF, and gyrators, GY). Effort and flow sources (Se and Sf) are used to define interactions between the system and the environment. Bonds, drawn as half arrows, have associated two variables (effort and flow). The rate of energy is defined as *effort* $\times$ *flow* for each bond. The primitive elements are connected by ideal 0 or 1 junctions (representing ideal energy parallel or series connections between components).

BGs also allow representing causality in the system adding a vertical *stroke* to the bonds. Causality establishes the cause and effect relationships between the e and f variables of the bonds. Causality is determined by constraints imposed by the incident BG elements. Usually, integral or derivative causality can be arbitrarily selected for capacitance and inductance elements. Where it is possible, integral causality is imposed in our approach. The causality of the whole system can be assigned automatically using the SCAP algorithm [65]. BGs will be formally defined and more deeply explained in Chapter 4.

Figure 3.9 presents the BG model of the three-tank system.

The BG contains also dynamic characteristics information of the dynamic relations between system variables. This information is made explicit in the Temporal Causal Graph (TCG) which can be automatically derived from the BG [76]. The TCG nodes include all state variables, measured variables and input variables. Links between nodes in the TCG are added according to the restrictions modeled by the BG. Figure 3.10

[3]There are algorithms to derive the MECs of a model and all their MEMs [95].

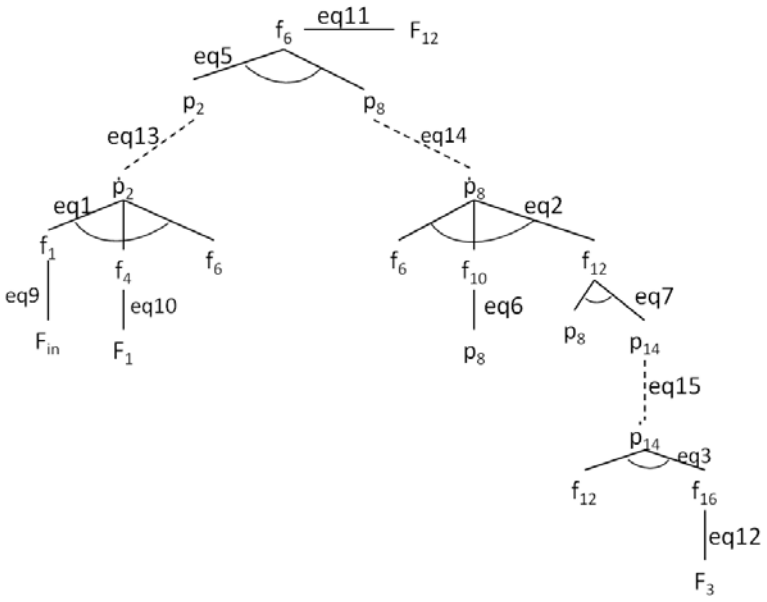


Figure 3.7: PC2 from the three-tank system.

shows the TCG derived from the BG in Figure 3.9. TCGs variables are always generalized efforts and flows. Hence the preassures of the three-tank system, previously denoted as p_2, p_8 and p_{14} are now e_2, e_8 and e_{14}. Due to the local nature of BG modeling, the TCG of 3.10 includes additional flow and effort variables associated to identity equations. These identity equations allow to eliminate these new variables recovering the computationl expresions identified by the $MEMs$ obtained from the Input/Output description of the system.

These modeling tools have been introduced because PCs can also be derived from TCGs [23] and they will be also used in fault diagnosis of hybrid systems.

Once the TCG of a system has been derived from its BG, the process to derive the PCs has two basic steps:

- Mark in the TCG the Input and the Measured variables.
- From each measured variable go backwards in every possible minimal causal path[4] until reaching inputs (sources), other measured variables or already visited nodes.

Figure 3.11 shows (bold path) the PC1 from the three-tank system in Figure 3.1 derived from its TCG with the process previously described. Dotted circles mark measured variables in the system and continuous circles are state variables.

Figures 3.6 and 3.11 present two different representations of the same PC from the same system (PC1 from the three-tank system in Figure 3.1). The computational model is the same in both representations. They have the same information (parameters, variables and relations). In this case, they also have the same causality assignment, they use the same interpretation for the relations. The main difference comes because a TCG uses only one possible causal assignment for the whole BG, while a PC explores every possible causal assignment for a MEC, i.e. a PC can have more than one MEM for the same MEC, while a PC derived from a TCG has only one possible MEM.

[4]Minimal Causal Path will be defined later in this dissertation. It is basically a causal path in a TCG ending in a measured or an already visited variable

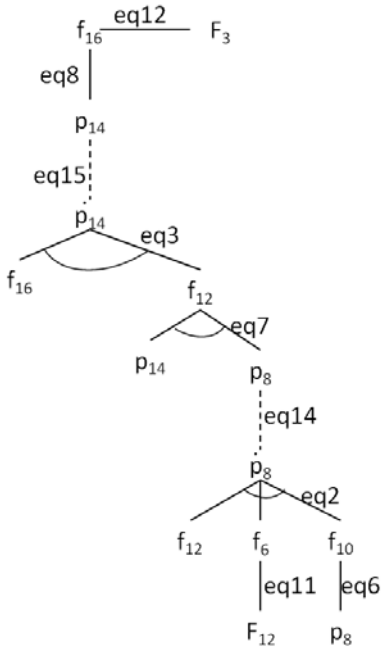


Figure 3.8: PC3 from the three-tank system.

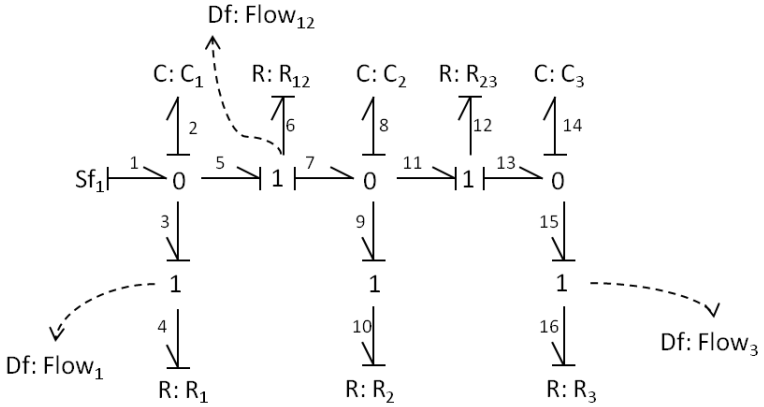


Figure 3.9: Bond Graph model of the three-tank system.

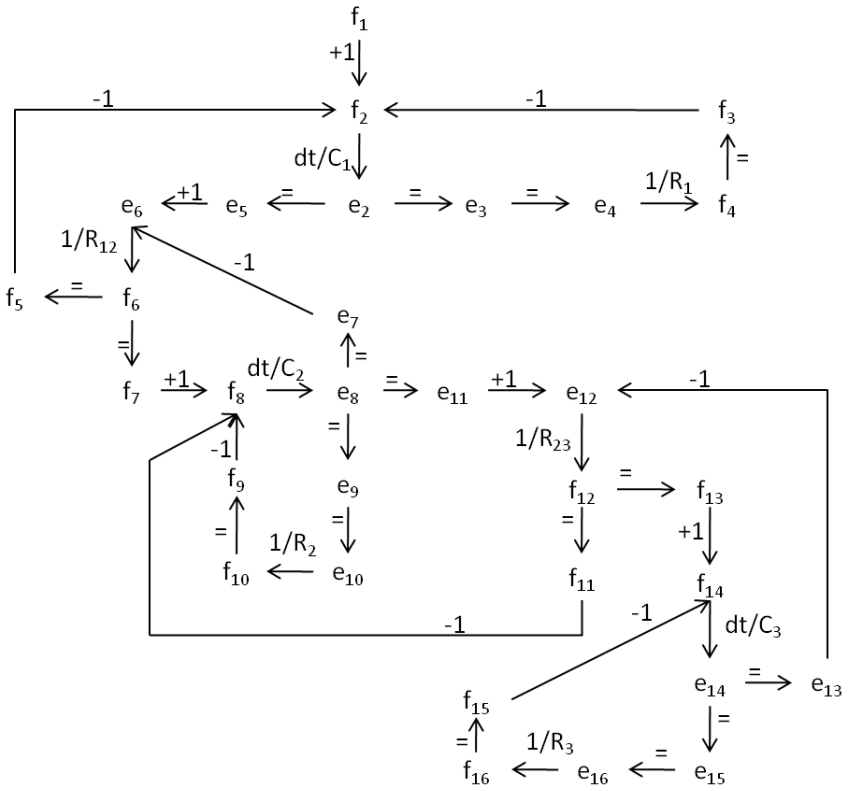


Figure 3.10: Temporal Causal Graph model of the three-tank system.

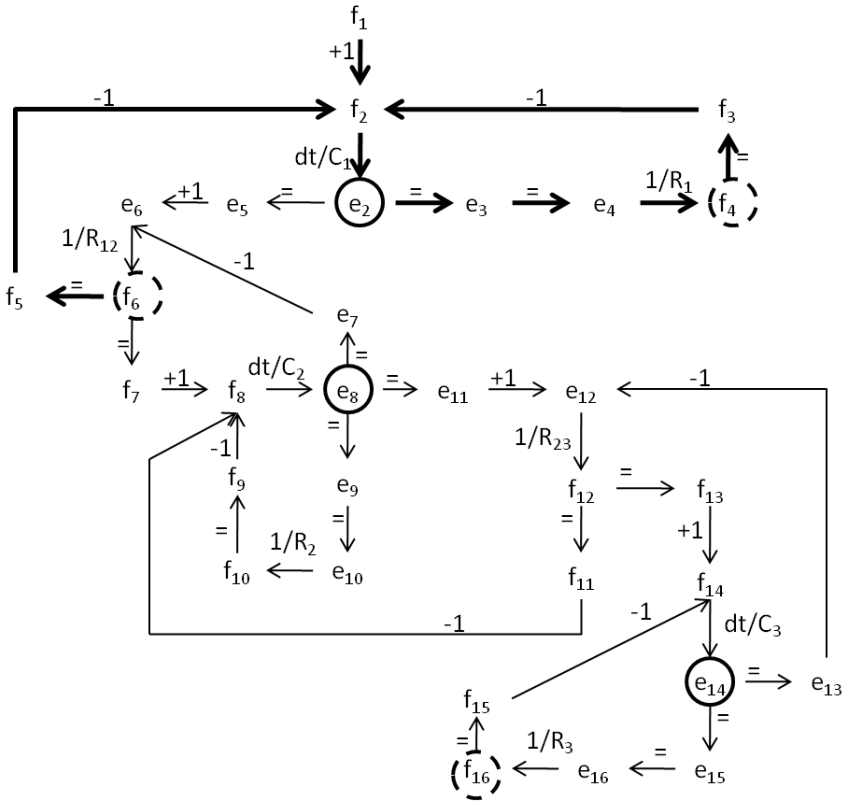


Figure 3.11: PC1 of the three tanks system derived from the TCG.

3.2 Fault Detection and Isolation of continuos systems using PCs

CBD allows performing fault detection and fault isolation using only models of correct behaviour in a two stage process. Fault detection consists of computing every conflict. Fault isolation consists of computing the minimal hitting set of the already found conflicts.

The process described before can be applied to CBD using PCs. In this case, PCs estimate the system behaviour. Residuals[5] are calculated using those estimations and the system measurements. Theoretically, a residual must be zero when the system is in its nominal behaviour but it is different from zero in case a fault occurs in the system. In the real world, residuals are not zero in nominal behaviour because system measurements have noise and models are not perfect, so statistical tests are used to confirm whether a residual is zero or not.

When a residual is confirmed to be different from zero a PC is activated, that is, there is a discrepancy between the PC estimation and the corresponding measurement or estimation. At this point, the PC is confirmed to be a conflict.

The fault isolation stage [100] is performed using the PCs fault signature matrix[6] to calculate the minimal hitting set of the confirmed conflicts. The set of fault candidates is refined as new detections confirmed more PCs as conflicts. The fault signature matrix relates the parameters of the system and the PCs where they are used. Table 3.1 shows the fault signature matrix of the PCs from the three-tank system.

	PC1	**PC2**	**PC3**
C1	1	1	
C2		1	1
C3		1	1
R1	1		
R12		1	
R2		1	1
R23		1	1
R3			1

Table 3.1: Signature matrix of the three-tank system in Figure 3.1

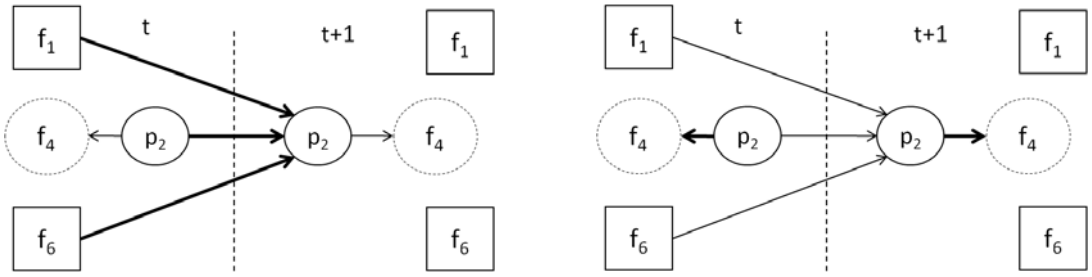
3.3 Dynamic Bayesian Networks

Dynamic Bayesian Networks are a probabilistic temporal model representation of a dynamic system. Basically, a DBN can be defined as a two-slice Bayesian Network (BN). There are two assumptions regarding the system: 1) it is time invariant, and 2) it is a First Order Markov process; in this case, two static and identical BNs connected by arcs (named inter slice arcs) are enough to model the system [81]. Inter slices arcs, which connect nodes of the DBN from different time slices, model the system dynamics. Intra slice arcs, connecting nodes from the same time slice, model instantaneous (algebraic) relations. Similar concepts were used in early work in the DX community for dynamic systems diagnosis using qualitative models [46].

The system variables (X, Z, U, Y) represented in a DBN are the state variables (X), the inputs (U), the observed or measured variables (Y) and, in some cases, other hidden variables (Z). Once we have the nodes, we need to define the parameters of the model, which are the state transition model (graphically represented by the inter slice arcs) and the sensor model (represented by intra slice arcs). Figure 3.12 shows a DBN highlighting the different types of arcs previously explained. The figure also presents the input nodes as squares, the measured variables as dotted circles and the state variables as solid circles.

[5]Residual is the FDI term equivalent to a discrepancy in DX, and can be computed as (Actual measurement - Estimated measurement).

[6]Matrix relating the possible faults in the system and the PCs they affect.



(a) Inter slice arcs (b) Intra slice arcs

Figure 3.12: Bold arrows show the two different types of arcs in a DBN.

The Markov observation model, $P(\mathbf{Y}_t|\mathbf{X}_t;\mathbf{U}_t)$, is derived from intra slice arcs $X_t \rightarrow Y_t$ and $U_t \rightarrow Y_t$, where $X \in \mathbf{X}$, $Y \in \mathbf{Y}$, $U \in \mathbf{U}$, and subscript t represents time. Similarly, inter slice arcs $X_t \rightarrow X_{t+1}$, $X_t \rightarrow X'_{t+1}$, and $U_t \rightarrow X_{t+1}$, where $X' \in \mathbf{X}$, represent the Markov state transition model, $P(\mathbf{X}_{t+1}|\mathbf{X}_t;\mathbf{U}_t)$.

Exact inference in DBNs is not computationally tractable in the general case. Hence, Monte Carlo simulation methods are used for approximate inference, particularly Particle Filter algorithm [67]. The unknown continuous stochastic distribution of the state is approximated by a discrete distribution obtained by weighted samples. After propagation of the state, the weights are updated with current observations. In this work, we assume a Gaussian distribution. Even using approximate inference algorithms, like Particle Filter, DBNs are computationally expensive. This is the disadvantage which is going to be faced in this work by means of the Possible Conflicts.

The variables of the system in Figure 3.1 can be labeled as: $X = \{p_2, p_8, p_{14}\}$, $U = \{f_1\}$, $Y = \{f_4, f_6, f_{16}\}$ and $Z = \{\phi\}$. Figure 3.13 presents the DBN model of the three-tank system.

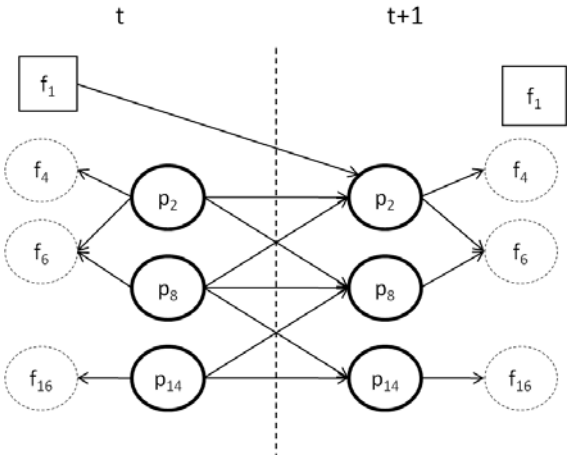


Figure 3.13: DBN of the three-tank system.

The DBN can be manually constructed from conditional independence considerations. In our case, we have derived the state equations of the system from its Temporal Causal Graph (TCG) [72, 103], which is automatically generated from a Bond Graph (BG) model of the system [76]. Using the TCG we can identify the state variables, as well as the inputs and the measurements in the system. To derive the equation to obtain each state variable we can propagate backwards in the TCG until reaching inputs, state variables or already visited variables. For the observations, the process is the same, but it starts in the measurements. The state equations and the observational model can be also used to easily obtain the DBN parameters.

3.4 Minimal DBNs derived from PCs

Dynamic Bayesian Networks (DBNs) are defined by its structure and the coefficients or parameter values. Both may be obtained from models in the consistency-based approach, particularly from Temporal Causal Graphs (TCGs) [72, 103] or from the MEM of a PC [5]. Before describing the proposed method to derive DBNs from PCs, we have to introduce some ideas about the importance observability has in this work.

3.4.1 Observability and System Factorization

Some researchers in MBD are studying how to simplify models from complex systems. This dissertation is focused on the proposal to decompose the model of the complete system in models of subsystems or components. Those models are less complex than the model of the complete system [128, 38, 2, 105, 102].

There are some approaches about systems factorization, one of them uses the idea of *Dissent* [128] to obtain subsystems less complex than the system itself. A *dissent* is a minimal subsystem with analytical redundancy. The previous definition is equivalent to the Possible Conflicts' definition [95] which has been presented earlier in this Chapter.

DBNs decomposition has already been studied. Roychoudhury et al. [105, 102] proposed a method to decompose the DBN of the complete system in serveral factors, less complex than the original DBN. In those works, the authors talk about the concept of observability [14, 112] and they explain its importance to estimate the state of a system from the inputs and the measurements.

Definition 8. *(Structural observability): According to Blanke et al. [14], a necessary and sufficient condition for a system to be structurally observable is that, under derivative causality:*

1. *All the unknown variables are reachable from the known ones.*
2. *The over-constrained and the just-constrained subsystems are causal.*
3. *The under-constrained subsystem is empty.*

There is a less restrictive definition of structural observability presented in [112].

Definition 9. *(Structural observability): According to Staroswiecki [112] a state variable, x, is structurally observable if there exists a matching that is complete on x.*

These two previous definitions of structural observability are a necessary and sufficient condition for a system to be observable, i.e. to be able to estimate the state of the system using inputs and measurements.

Observability must be applied to DBNs, if a system is not structurally observable values for the hidden state variables cannot be estimated using the measurements. In other words, structural observability in a system (or subsystem) guarantees that the state can be obtained from its measurements or observations and inputs.

Possible Conflicts are structurally observable by definition [78]. They have a complete matching on the state variables on the PC. The idea in this dissertation is the application of the Possible Conflicts technique [95] to derive minimal DBNs modeling subsystems from the complete system.

3.4.2 Factoring DBNs with PCs: Minimal DBNs

This dissertation proposes a method to efficently derive the DBN structure and its parameters from the MEM of a PC. The DBN obtained will be smaller than the DBN modeling the complete system, but it will be structurally observable [78].

Given that a MEM provides a computational model of a minimal redundant subsystem, that can be implemented as a simulation model [94] or as an observer [16], a natural extension is to implement the model as a Dynamic Bayesian Network (DBN). A necessary condition to transform the simulation model of a dynamic PC into a DBN is observability. This is always guaranteed because a MEM can compute all its unknown variables from its inputs and, hence, it is structurally observable, as defined in [14, 112], result that has already been shown in [78].

The MEM of a PC can also provide the model of a DBN of the subsystem identified by the PC, as states the following proposition:

Proposition 1. *Those MEM that include the first derivative of a state variable provide the structural description of a DBN for the subsystem defined by the Possible Conflict.*

Requiring that the MEM of the PC includes the first derivative of a state variable is a necessary condition to have a dynamic system. Otherwise, the MEM only includes algebraic relations.

The proof of Proposition 1 is constructive and generally requires two steps:

- **Structure:** Generate the initial DBN from the state space equations of the subsystem defined by the PC.
- **Simplification:** For any state variable which is conditionally dependent only on input nodes, replace that state variable and inputs by a new input node, according to algebraic *MEM* computation.

The key step on DBN generation is the Structure step. Second step of the construction process is not always needed and it just simplifies the network, eliminating state variables that are algebraically estimated from known inputs and observed variables in the *MEM* -although not in the complete system-.

Structure: State space model a MEM. Given a Minimal Evaluation Model, MEM_i, $H_{mem_i} = \{V_{mem_i}, R_{mem_i}\}$, of a PC with model $M_i = (X_i, U_i, < y_i >)$, its state space representation can be expressed in a general way by the tuple $(X_i, U_i, y_i, \mathbf{f_i}, g_i)$, where:

- $X_i =< x_{i_1}, x_{i_2}, \ldots, x_{i_n} >$ is the state vector of the system described by MEM_i,
- $U_i =< u_{i_1}, u_{i_2}, \ldots, u_{i_m} >$ is the input vector of the system described by MEM_i,
- y_i is the output of the system described by MEM_i,
- $\mathbf{f_i}$ is the state transition function of the system described by MEM_i,
- g_i is the observational function of the system described by MEM_i.

with $\{x_{i_1}, x_{i_2}, \ldots, x_{i_n}, u_{i_1}, u_{i_2}, \ldots, u_{i_m}, y_i\} \subseteq V_{mem_i}$, $\{\dot{x}_{i_1}, \dot{x}_{i_2}, \ldots, \dot{x}_{i_n}\} \subseteq V_{mem_i}$, y_i is the discrepancy node of MEM_i, $\{u_{i_1}, u_{i_2}, \ldots, u_{i_m}, y_i\}$ are the only knonwn variables of MEM_i and

$$\dot{X}_i(t) = \mathbf{f_i}(X_i(t), U_i(t)) \qquad (trn)$$
$$y_i(t) = g_i(X_i(t)(t), U_i(t)) \qquad (obs)$$

When MEM_i has no algebraic loops, each *j-th* component of the state transition function, f_{i_j}, is obtained from H_{mem_i} by the following procedure:

- Build the transitional model subgraph, $H_{f_{i_j}} \subseteq H_{mem_i}$, traversing H_{mem_i} from the occurrence of $\dot{x}_{i_j}$ to the first occurrence of either an input or a state variable.
- Eliminate intermediate unknown variables by substitution method starting from $\dot{x}_{i_j}$ to leaf nodes of $H_{f_{i_j}}$, using the equations that label its arcs, in the order specified by $H_{f_{i_j}}$.

Similarly, the output function g_i is obtained from H_{mem_i} by the procedure:

- Build the observational model subgraph, $H_{g_i} \subseteq H_{mem_i}$, traversing H_{mem_i} from the output y_i to the first occurrence of either an input or state variable.
- Eliminate intermediate unknown variables by substitution method starting from output y_i to leaf nodes of H_{g_i}, using the equations that label its arcs, in the order specified by H_{g_i}.

By construction, the causal matching in each MEM guarantees that $\forall i, j$, $H_{f_{i_j}}$ and H_{g_i} can be built for any MEM_i and state variable x_{i_j} such that $\dot{x}_{i_j} \in V_{mem_i}$. Consequently, when MEM_i has no algebraic loops, the analytical expression of f_{i_j} and g_i can always be obtained from MEM_i. If MEM_i has an algebraic loop, we cannot obtain the analytical expression of f_{i_j} and/or g_i. Nevertheless, we still can built $H_{f_{i_j}}$ and H_{g_i}, which provide the structural description of f_{i_j} and g_i, respectively. From these structural descriptions an external solver can compute the value of all the unknown variables in state space formulation. Also, these structural descriptions are enough to obtain the structural description of the DBN.

Simplification of state space variables in a MEM. If z_{i_j} is a state variable of the complete system and $z_{i_j} \in MEM_i$ is conditionally dependent only on input nodes then $\dot{z}_{i_j} \notin \{V_{mem_i}\}$ and z_{i_j} is not an element of X_i.

When MEM_i has no algebraic loops, for each state variable z_{i_j} conditionally dependent only on input nodes, the function $Input\text{-}z_{i_j}(\ldots)$ to compute z_{i_j} from inputs is obtained by the following procedure:

- Build the simplification model subgraph, $H_{Input-z_{i_j}} \subseteq H_{mem_i}$, traversing H_{mem_i} from the occurrence of z_{i_j} to the first occurrence of an input.
- Eliminate all unknown variables by substitution method starting from z_{i_j} to leaf nodes of $H_{Input-z_{i_j}}$, using the equations that label its arcs, in the order specified by $H_{Input-z_{i_j}}$.

If MEM_i has an algebraic loop, we cannot obtain the analytical expression of $Input\text{-}z_{i_j}$ although we still can built $H_{Input-z_{i_j}}$ which provides the structural description of $Input\text{-}z_{i_j}$.

Obtaining Minimal DBNs. Once we have the state equations of the subsystem defined by a PC and the new functions $Input\text{-}z_{i_j}$, to simplify the state variables conditionally dependent only on inputs of the PC, we can define the concept of minimal DBN.

First, we extend the state transition model, equation *trn*, and the observational model, equation *obs*, to account for uncertainties, generating their stochastic versions:

$$\dot{X}_i(t) = \mathbf{f_i}(X_i(t), U_i(t), V_i(t)) \qquad (st\text{-}trn)$$
$$y_i(t) = g_i(X_i(t)(t), U_i(t), w_i(t)) \qquad (st\text{-}obs)$$

where V_i and w_i represents the process and measurement noise vectors, respectively.

Definition 10 (Minimal DBN)**.** *Let MEM_i be a Minimal Evaluation Model of a PC, with state space equations given by equations trn and obs. Let $\{z_{i_1}, \ldots, z_{i_k}\}$ the, possibly empty, set of state variables of the system that are, in MEM_i, conditionally dependent only on inputs of MEM_i . A minimal DBN of the subsystem defined by the PC is the DBN obtained by:*

- *Building the DBN that has st-trn as its state transition model and st-obs as its observational model, $DBN_{structure}$.*
- *If $\{z_{i_1}, \ldots, z_{i_k}\} \neq \emptyset$, replacing each occurrence of z_{i_j} and its inputs by the new input node $Input\text{-}z_{i_j}$.*

We can illustrate the whole process with the PC1 of the three-tank system, in this case we have M_{PC1}=$\{< p_2 >, < F_{in}, F_{12} >, < f_4 >\}$. Figures 3.14 and 3.15 show $H_{f_{1_1}}$ and H_{g_1} for this subsystem, respectively. Hence equation *trn* reduces to equation: $\dot{p}_2(t) = F_{in} - F_{12} - p_2/R_1$. Equation *obs* is $f_4 = p_2/R_1$. Simplification step is not needed. Figure 3.16 shows the Minimal DBN for the subsystem defined by PC1 (DBN1).

None of the PCs of this system needs to perform the *Simplification* step to generate the DBN. Figure 3.17 shows a PC whose transformation on a DBN requires both steps. Figures 3.18 and 3.19 show $H_{f_{1_1}}$ and H_{g_1} for the MEM1 of the two-tank system, respectively. Figure 3.20 shows the simplification model subgraph ($H_{Input-z_{1_1}}$) of this subgraph.

Figure 3.21 shows the intermediate network created applying *Structure* step and Figure 3.22 the final DBN after *Simplification* step.

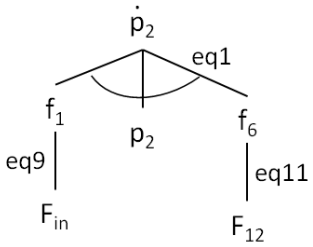


Figure 3.14: Subgraph of the transitional model, $H_{f_{1_1}}$, of the MEM1 of the three-tank system.

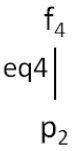


Figure 3.15: Subgraph of the observational model, H_{g_1}, of the MEM1 of the three-tank system.

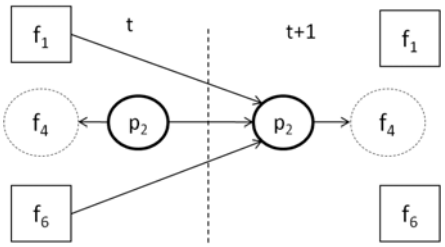


Figure 3.16: Minimal DBN derived from the PC1 of the three-tank system (DBN1).

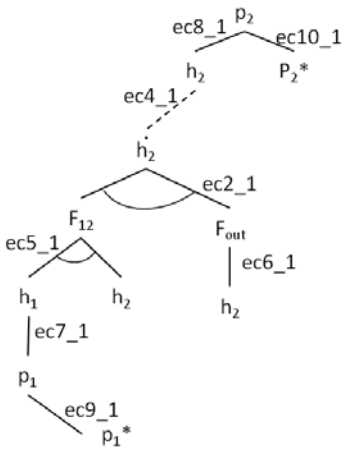


Figure 3.17: Possible conflict of a two tank system with the pressure of each tank measured.

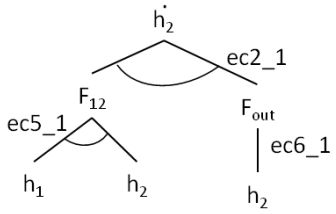


Figure 3.18: Subgraph of the transitional model, $H_{f_{1_1}}$, of the MEM1 of the two-tank system.

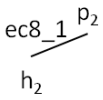


Figure 3.19: Subgraph of the observational model, H_{g_1}, of the MEM1 of the two-tank system.

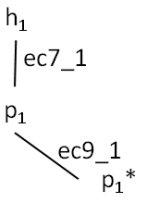


Figure 3.20: Subgraph of the simplification model, $H_{Input-z_{1_1}}$, of the MEM1 of the two-tank system.

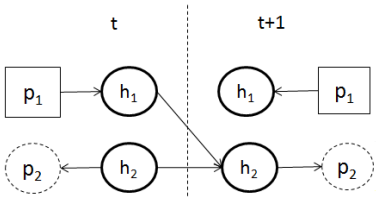


Figure 3.21: Intermediate $DBN_{structure}$ of the possible conflict in Figure 3.17 after the *Structure* step.

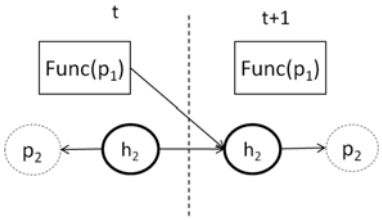


Figure 3.22: DBN of the possible conflict in Figure 3.17 after the *Simplification* step.

3.5 Diagnosis of Continuous Systems with Minimal DBNs

The DBNs derived from the PCs have the same structural information that the PCs, because DBNs equations are obtained just manipulating the model of the corresponding MEM, but no additional equations are introduced. Hence, the set of Minimal DBNs have exactly the same fault detection and fault isolation capabilities than the original set of PCs.

3.5.1 Diagnosis Architecture. Fault Detection, Isolation and Identification with Minimal DBNs

Minimal DBNs allow tackling all the stages of model based diagnosis, that is, fault detection, fault isolation and fault identification, in the Consistency Based Diagnosis framework with fault models in a predictive approach. Figure 3.23 shows the diagnosis architecture.

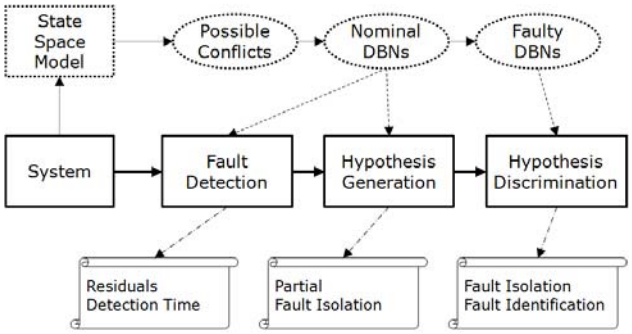


Figure 3.23: The diagnosis architecture integrating DBNs and PCs

Fault Detection

Nominal minimal DBNs are obtained offline from the system model through PCs decomposition. The three resultant minimal DBNs for the three-tank system are shown in Figure 3.16, Figure 3.24 and Figure 3.25. These DBNs are run in parallel to perform fault detection.

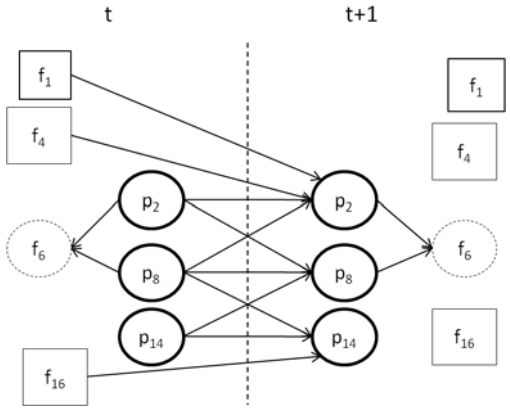


Figure 3.24: DBN of the PC2 of the three-tank system.

The DBN predicted output is compared against existing measurements, i.e. they create our set of residuals. To test residual activation we use a statistical test, the Ztest [13, 53], that is used to confirm whether two different intervals in a signal have the same distribution. It is supposed that both intervals have a Gaussian distribution and knowing the mean and the variance of the first one we test if the second has

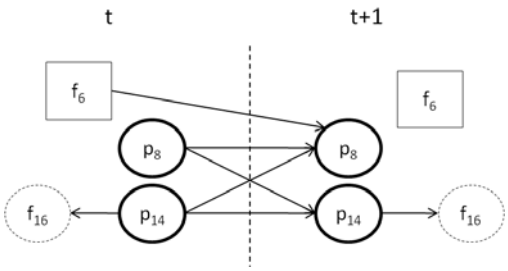


Figure 3.25: DBN of the PC3 of the three-tank system

the same parameter values. It can be done with different confidence values, changing the parameter α in the test.

In this case, the signal we have used is the residual, which is the difference between the measured value of an observed variable and its estimated value, as it was previously mentioned. In case of not knowing the mean and the standard deviation of the intervals equations 3.1 and 3.2 can be used to estimate them.

$$\hat{\mu}_{N_2}(k) = \frac{1}{N_2} \sum_{i=k-N_2+1}^{k} r(i) \tag{3.1}$$

$$\hat{\sigma}^2_{N_1}(k) = \frac{1}{N_1 - 1} \sum_{i=k-N_1+1}^{k} (r(i) - \mu_{N_1}(k))^2 \tag{3.2}$$

The parameters we need to define for the Ztest are the *length of the first interval* (N_1), the *length of the second interval* (N_2), the *delay between the two intervals* ($VarDelay$) and the *value of alpha* (usually, $\alpha = 0.05$).

Fault isolation

The first stage in fault isolation is to compute the set of fault candidates from the minimal hitting set of the fault information in the set of PCs. However, consistency based isolation is not usually able to completely isolate the faulty parameters, unless the system is completely diagnosable [118]. Hence, some practical choice has to be made regarding when to start the fault identification stage. A naive approach consist of launching the identification stage as soon as a fault is detected by any minimal DBN, refining the set of candidates when a new DBN establishes a new conflict and stopping the simulation of the faulty DBNs that have been discarded. However, this approach may incur in a high computational overhead, especially when several minimal DBNs detect the same fault with negligible or small temporal differences. On the other end of the spectrum, it is possible to wait until no further candidate refining is expected, according to the fault signature matrix of the system (Figure 3.1 presents the fault signature matrix for the three-tank system). But this approach may generate long isolation times. Even worst, the process may not end if there is a false negative, with the diagnosis system waiting indefinitely for a PC to become a conflict. Hence, in the experimental setting we have opted by the criterion of start the identification stage as soon as one of the following conditions is satisfied:

1. Current candidates cannot be further refined, according to the fault signature matrix.
2. There has elapsed k time steps from the first fault detection, assuming non intermittent faults.

The parameter k must be tuned according to the system dynamic behaviour.

In a predictive approach, fault isolation requires introducing fault modes. We have opted for a simple abrupt fault model [103].

Abrupt fault An abrupt fault is characterized by a fast change in a parameter value (modeled as $b(t)$ in the equation below). The temporal profile of a parameter with an abrupt fault, $p^a(t)$ is given by:

$$p^a(t) = \begin{cases} p(t) & t < t_f \\ p(t) + b(t) = p(t) + \sigma_p^a & t \geq t_f \end{cases}$$

where σ_p^a models the absolute change of the parameter value.

We have considered 8 abrupt faults in the system in Figure 3.1: in the capacitances of each tank (C1, C2 and C3), in the resistance of the output of each tank (R1, R2 and R3) and in the resistances of the flow between tanks (R12 and R23). The fault signature matrix of the minimal DBNs is the same as the fault signature matrix of the PCs of the system, shown in Table 3.1.

Fault Identification

Fault identification is done using the minimal DBNs for fault modes or faulty Minimal DBNs. Minimal DBNs for each fault mode are obtained from minimal DBNs of the nominal system according to [103, 102] proposal. Nominal minimal DBNs are extended with an additional node for the faulty parameter. If some network node is conditionally dependent on the new node, an edge is added from the new node to the *'not conditionally independent'* node. Figure 3.26 shows the faulty network obtained from PC1 for an abrupt fault in the capacitance of tank 1 (C_1). For each minimal DBN it is necessary to build as many faulty DBNs as indicated in the fault signature matrix.

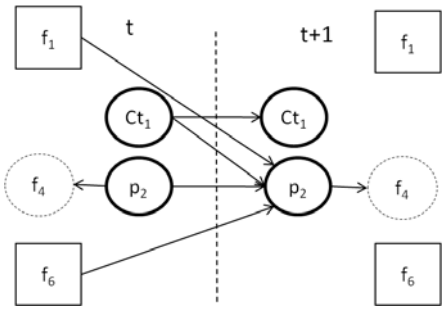


Figure 3.26: DBN of PC1 of the three-tank system with a fault in the capacitance of tank 1.

Fault identification requires tracking the system with faulty DBNs. For each single fault candidate a faulty minimal DBN has to track the system. If a new detection allows reducing the number of single fault candidates, the corresponding fault hypotheses are rejected and the associated Minimal DBNs no longer track the system. Eventually, one of the faulty DBNs will converge[7] identifying the new value of the parameter, while all the other DBNs tracking the fault will not. This will happen in case the fault is diagnosticable, if it is not, two faults could be identified as potential candidates.

We have to take into account the effect of sample impoverishment, that with Importance Sampling algorithm [8] is harmless for tracking the non faulty system. However, this is a serious problem once a fault is detected, because faults are usually represented in less probable (smaller weight) samples. To cope with this difficulty, we have resorted to the *ad hoc* solution adopted in [103]. To perform fault identification of a given parameter, we linearly modify the variance of that parameter, now a node of the DBN, from an initial large value to its expected final value. In this way, at the beginning of the Particle Filtering simulation, we allow for particles with a parameter value far away from its nominal value. Hence, the probability of having particles close to the faulty parameter value is increased. As simulation proceeds, the variance of the parameter diminish, favoring particles close to the current parameter value, according to current observations. To facilitate even more this process, we usually increase the number of particles in the faulty DBNs.

3.5.2 Diagnosis results with the three-tank system

The diagnosis architecture has been initially tested with the three-tank system. This section presents the results obtained.

[7] The residual becomes zero.

Fault Scenarios

We have developed two fault scenarios for the three-tank system: (1) an abrupt fault in the capacitance of tank 1 and (2) an abrupt fault in the resistance out of tank 3. The nominal value of C1 is $1.5708 \cdot 10^{-6}$ and the nominal value of R3 is $2 \cdot 10^{7}$ In both cases the fault magnitude is 10% of the nominal value of the parameter. We have simulated for 10000 time steps, starting with the three tanks empty and injecting the faults at time stamp 2000. Simulink has been used to generate data of the faulty system. A 5% Gaussian noise has been added to sensors. The number of particles used in the Particle Filter algorithm has been 500. Ztest has been applied to decide on network detection and also on network convergence for fault identification.

To easily identify each minimal DBN we will name them as *DBNi*, where i is the PC number where the DBN comes from.

Abrupt fault in C1 Fault detection is performed with the nominal minimal DBNs (see Figure 3.27). The DBN2 detects the fault at time 2001 (Figure 3.27 b)) while the DBN1 detects the fault at time 2002 (Figure 3.27 a)). According to the fault signature matrix of the system, Table 3.1, the DBN3 does not detect the fault (Figure 3.27 c)).

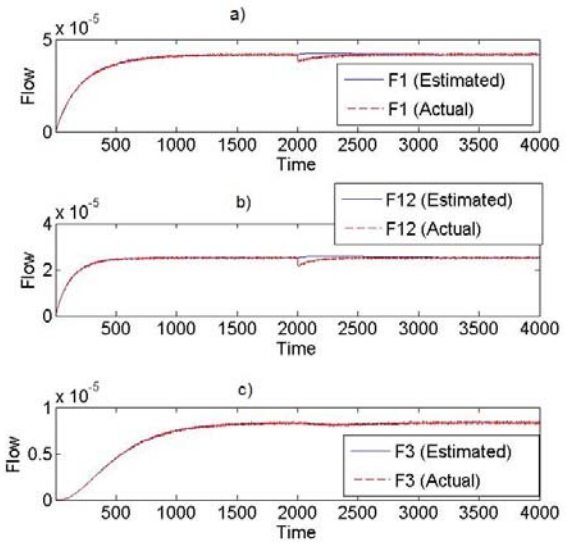


Figure 3.27: Behaviour (observed variable) of the nominal minimal DBNs of a) PC1, b) PC2 and c) PC3 tracking an abrupt fault (10%) in the capacitance of tank 1.

Hence, from time 2002 there is only one single fault candidate: C1. We have run the faulty DBN1 for a fault in C1 (Figure 3.26) starting 50 time steps before the fault is detected, to launch simulation from a known system state with nominal behaviour. The behaviour of the network is shown in Figure 3.28. Convergence time is 389 time steps, (337 after fault injection). Figure 3.28 shows that the faulty parameter converges to an acceptable value many instants before, but ztest is performed on the residual of the observed flow (not shown in the figure) and it also introduces a delay.

Abrupt Fault in R3 Like in the previous scenario, fault detection is performed with the three nominal minimal DBNs. Now, only DBN3 detects the fault, at time 2007, (see Figure 3.29).

In this case, we have 5 single fault candidates (C2, C3, R2, R23 and R3) and we have to run the faulty DBN3 for all these faults to check which one converges.

In Figure 3.30 we can see how the DBN3 with the extra node for the fault in R3 is able to track the state variables and also gives us a pretty good estimation of the parameter after the fault. Convergence time is 378 (321 after fault injection).

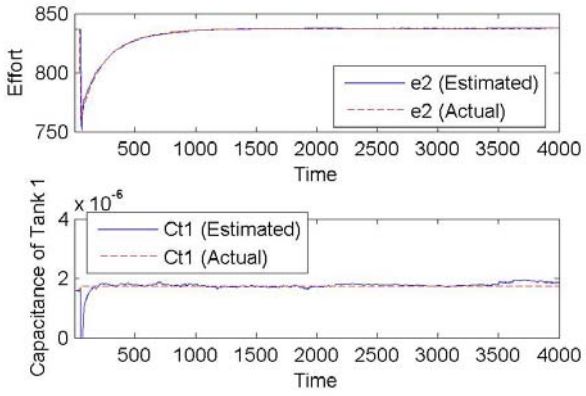


Figure 3.28: State variables tracked with the DBN1 for a fault in capacitance of tank 1. The last chart at the bottom is the estimation of the parameter C1.

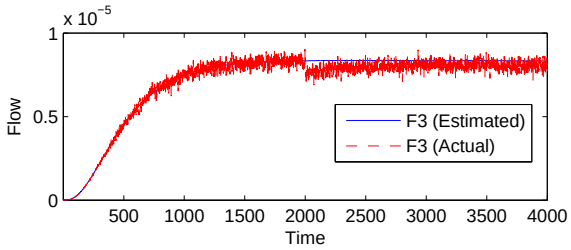


Figure 3.29: Behaviour (observed variable) of the DBN of PC3 for a fault in the resistance out of tank 3.

On the other hand, checking the behaviour of the other DBNs launched to identify this fault we can see charts like the ones in Figure 3.31 where it is shown how the DBN3 estimating a fault in C2 is not able to track the system behaviour, it does not converge.

Comparing complete DBN and minimal DBNs performance

Once we have presented the two fault scenarios tested we will see a quantitative comparison of the performance of the DBN of the complete system and the performance of the minimal DBNs. For the two considered abrupt faults, C1 and R3, mean detection, execution and convergence times are computed. Parameter value convergence is also considered. All experiments have been repeated ten times. Table 3.2, row *Execution Time*, shows execution time for 10000 time steps. As it was to be expected, Minimal DBNs require less computation time than the original DBN. Faults are injected at time 2000. Rows *C1A* and *R3A* show that there are no false positive detections and that detection time is similar for every network considered.

	Complete	**DBN1**	**DBN2**	**DBN3**
Execution Time	84.25	71.14	77.77	78.40
C1A +0.1	2001 (0.0)	2002 (0.0)	2001 (0.0)	
R3A +0.1	2008 (0.0)			2007 (0.0)

Table 3.2: Mean execution and detection time for nominal DBNs (standard deviation in brackets).

For fault identification, simulation starts 50 time steps before the fault is detected and simulation time extends to 8050 time stamps. Table 3.3, first two rows, shows execution time for faulty networks, that are also smaller for minimal DBNs. Second two rows show convergence time. Compared to the complete DBN,

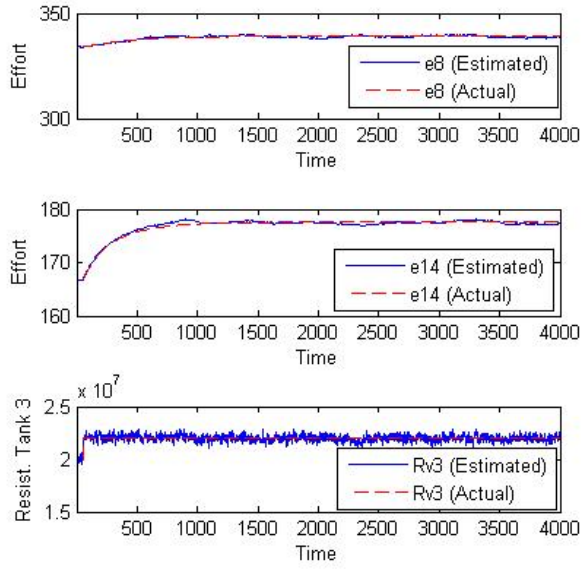


Figure 3.30: State variables tracked with the DBN3 for a fault in resistance out of tank 3. The last chart at the bottom is the estimation of the parameter R3.

convergence time is smaller for the fault in capacitance of tank 1, and it is equal for the fault in resistance R3.

Finally, Table 3.4 shows the Mean Square Error (MSE) of the estimated parameter. Error in this running example is smaller for DBN1 and DBN2 for faults in capacitance of tank 1, but it is slightly bigger for DBN3 for fault of resistance R3.

	Complete	**DBN1**	**DBN2**	**DBN3**
Exec. Time C1	46.66	43.43	45.40	
Exec. Time Rv3	67.15			63.31
C1A +0.1	$1.15 \cdot 10^3$ $(2.1 \cdot 10^3)$	$3.89 \cdot 10^2$ $(2.8 \cdot 10^1)$	$4.95 \cdot 10^2$ $(1.2 \cdot 10^2)$	
Rv3A +0.1	$3.78 \cdot 10^2$ (1.4)			$3.78 \cdot 10^2$ $(8.8 \cdot 10^{-1})$

Table 3.3: Mean execution and convergence time for faulty DBNs (standard deviation in brackets).

The results previously presented are a preliminar work before testing the diagnosis architecture for continuous systems with a 12^{th} order electrical circuit, that is the complete case study presented in next section.

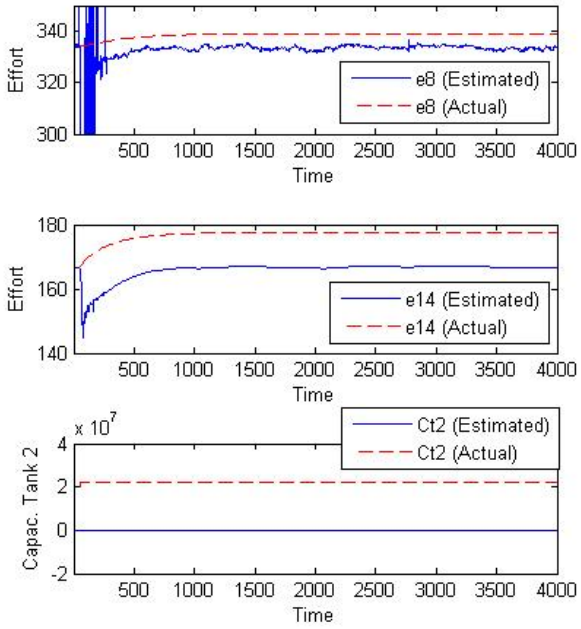


Figure 3.31: State variables tracked with the DBN3 for a fault in the capacitance of tank 2. The last chart at the bottom is the estimation of the parameter C2. The data used to simulate the DBN are from a fault in R3.

	Complete	**DBN1**	**DBN2**	**DBN3**
C1A +0.1	$2.62 \cdot 10^{-8}$ $(2.4 \cdot 10^{-7})$	$2.06 \cdot 10^{-8}$ $(1.6 \cdot 10^{-8})$	$1.67 \cdot 10^{-8}$ $(7.2 \cdot 10^{-9})$	
Rv3A +0.1	$2.77 \cdot 10^{4}$ $(3.6 \cdot 10^{8})$			$2.81 \cdot 10^{4}$ $(9.0 \cdot 10^{3})$

Table 3.4: Mean Square Error of the estimation for the faulty parameter by each DBN (standard deviation in brackets).

3.6 Case Study. 12^{th} order electrical circuit.

This section describes the 12^{th} order electrical circuit shown in Figure 3.32. This system is going to be used as a case study to show how the minimal DBNs decomposition works on a high order system.

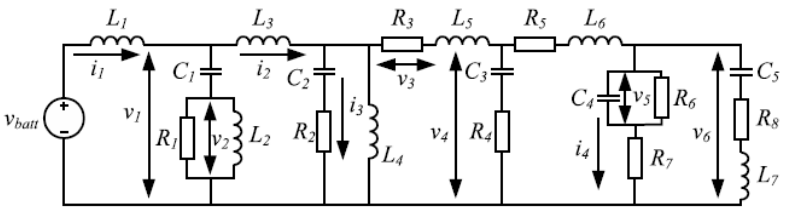


Figure 3.32: 12^{th} Order Electrical Circuit.

The 12^{th} order electrical circuit system has 12 state variables (variables associated to the five condensers, C_i elements, and to the seven inductances, L_i elements), 10 measurements, represented in the circuit by the voltages, v_i, and the currents, i_j, (Figure 3.32) using arrows, and one input, $U = \{v_{batt}\}$. The variables of its state space representation are $X = \{p_1, p_2, p_3, p_4, p_5, p_6, p_7, q_1, q_2, q_3, q_4, q_5\}$, $U = \{v_{batt}\}$ and $Y = \{i_1, v_1, v_2, i_2, i_3, v_3, v_4, i_4, v_5, v_6\}$. p_j and q_j are the generalized state space variables with $p_j = i \cdot L_j$ and $q_j = e \cdot C_j$ [108]. The system dynamics is described by twelve state equations:

$$\dot{p}_1 = e1 - \frac{q1}{C1} - (\frac{p1}{L1} - \frac{p3}{L3} - \frac{p2}{L2}) * R1 \quad \text{(eq.1)}$$

$$\dot{q}_1 = \frac{p1}{L1} - \frac{p3}{L3} \quad \text{(eq.2)}$$

$$\dot{p}_2 = (\frac{p1}{L1} - \frac{p3}{L3} - \frac{p2}{L2}) * R1 \quad \text{(eq.3)}$$

$$\dot{p}_3 = \frac{q1}{C1} + (\frac{p1}{L1} - \frac{p2}{L2} - \frac{p3}{L3}) * R1 - \frac{q2}{C2} - (\frac{p3}{L3} - \frac{p4}{L4} - \frac{p5}{L5}) * R2 \quad \text{(eq.4)}$$

$$\dot{p}_4 = \frac{q2}{C2} + (\frac{p3}{L3} - \frac{p4}{L4} - \frac{p5}{L5}) * R2 \quad \text{(eq.5)}$$

$$\dot{q}_2 = \frac{p3}{L3} - \frac{p4}{L4} - \frac{p5}{L5} \quad \text{(eq.6)}$$

$$\dot{p}_5 = \frac{q2}{C2} + (\frac{p3}{L3} - \frac{p4}{L4} - \frac{p5}{L5}) * R2 - \frac{p5}{L5} * R3 - \frac{q3}{C3} - (\frac{p5}{L5} - \frac{p6}{L6}) * R4 \quad \text{(eq.7)}$$

$$\dot{q}_3 = \frac{p5}{L5} - \frac{p6}{L6} \quad \text{(eq.8)}$$

$$\dot{p}_6 = \frac{q3}{C3} + (\frac{p5}{L5} - \frac{p6}{L6}) * R4 - (\frac{p6}{L6} - \frac{p7}{L7}) * R7 - \frac{q4}{C4} - \frac{p6}{L6} * R5 \quad \text{(eq.9)}$$

$$\dot{q}_4 = \frac{p6}{L6} - \frac{p7}{L7} - \frac{q4}{C4 * R6} \quad \text{(eq.10)}$$

$$\dot{q}_5 = \frac{p7}{L7} \quad \text{(eq.11)}$$

$$\dot{p}_7 = (\frac{p6}{L6} - \frac{p7}{L7}) * R7 + \frac{q4}{C4} - \frac{p7}{L7} * R8 - \frac{q5}{C5} \quad \text{(eq.12)}$$

Plus ten equations for the observational model:

$$i_1 = \frac{p1}{L1} \quad \text{(eq.13)}$$

$$v_1 = \frac{q1}{C1} + (\frac{p1}{L1} - \frac{p3}{L3} - \frac{p2}{L2}) * R1 \quad \text{(eq.14)}$$

$$v_2 = (\frac{p1}{L1} - \frac{p3}{L3} - \frac{p2}{L2}) * R1 \quad \text{(eq.15)}$$

$$i_2 = \frac{p3}{L3} \quad \text{(eq.16)}$$

$$i_3 = \frac{p3}{L3} - \frac{p4}{L4} - \frac{p5}{L5} \quad \text{(eq.17)}$$

$$v_3 = \frac{p5}{L5} * R3 \quad \text{(eq.18)}$$

$$v_4 = \frac{q3}{C3} + (\frac{p5}{L5} - \frac{p6}{L6}) * R4 \quad \text{(eq.19)}$$

$$i_4 = \frac{p6}{L6} - \frac{p7}{L7} \quad \text{(eq.20)}$$

$$v_5 = \frac{q4}{C4} \quad \text{(eq.21)}$$

$$v_6 = \frac{q4}{C4} + (\frac{p6}{L6} - \frac{p7}{L7}) * R7 \quad \text{(eq.22)}$$

Although the state space representations provides the most compact description of a dynamic system equations, PCs formalism works with the Input/Output description of the system, which highly increases the number of dynamic equations. For the sake of completeness, we have exhaustively generated all the Input/Output equations for the 12^{th} order electrical circuit, which yields 89 equations (68 are dynamic equations). We present just a small sample of these equations, which will be later used to illustrate the basic concepts of the approach:

$$\dot{p}_1 = e1 - v_1 \quad (c1)$$

$$\dot{q}_1 = i_1 - i_2 \quad (c7)$$

$$\dot{p}_2 = (i_1 - 1_2 - \frac{p2}{L2}) * R1 \quad (c11)$$

$$\dot{p}_3 = v_1 - \frac{q2}{C2} - R2 * i_3 \quad (c16)$$

$$\dot{q}_2 = i_3 \quad (c33)$$

$$\dot{p}_5 = \frac{q2}{C2} + i_3 * R2 - \frac{p5}{L5} * R3 - v_4 \quad (c37)$$

$$i_1 = \frac{p1}{L1} \quad (c69)$$

$$v_1 = \frac{q1}{C1} + v_2 \quad (c70)$$

$$v_2 = (i_1 - i_2 - \frac{p2}{L2}) * R1 \quad (c75)$$

$$i_2 = \frac{p3}{L3} \quad (c79)$$

$$v_3 = \frac{p5}{L5} * R3 \quad (c82)$$

The DBN modeling the system is obtained from its state space representation, where the state equations provide the transition model and the remaining equations cater for the observational model. Figure 3.33 shows the DBN structure assuming DBN = {U,X,Y}, with the U,X,Y description previously presented.

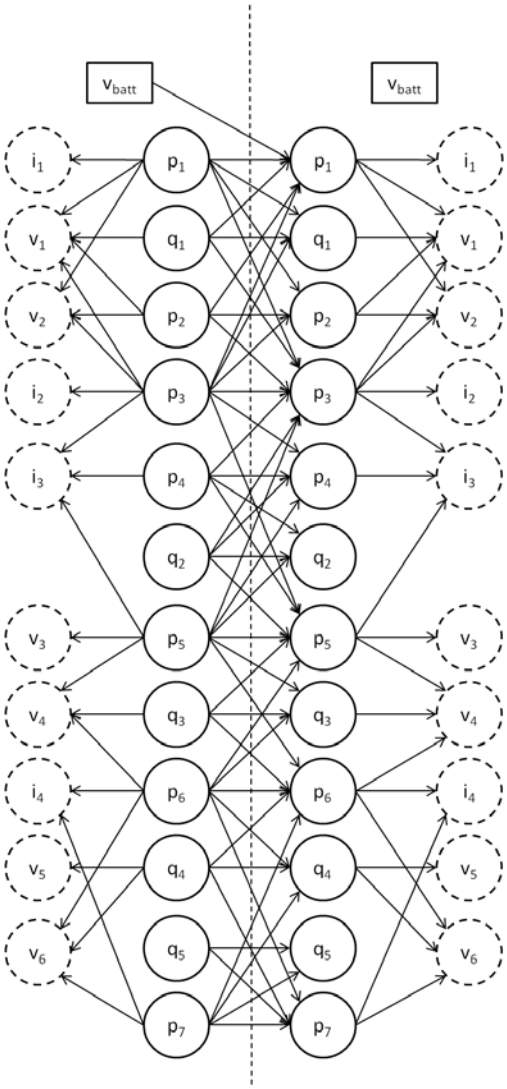


Figure 3.33: Dynamic Bayesian Network of the 12^{th} Order Electrical Circuit.

3.6.1 Possible Conflicts and Minimal DBNs

Possible Conflict method, as it has been previously explained in this Chapter, works with an abstract representation of a system, obtained describing the structure of the Input/Output equations and its variables with a hypergraph. Hence, system description is the hypergraph $H_{SD} = \{V, R\}$, where V is the set of variables of the system and $R = \{r_1, r_2, \ldots, r_m\}$ is a family of subsets in V, where each r_k represents a constraint in the model and its elements are the variables that occur at equation r_k. For instance, from equation $c7$, $\dot{q}_1 = i_1 - i_2$, we obtain the set $c_7 = \{\dot{q}_1, i_1, i_2\}$. Figure 3.34 shows part of the H_{SD} of the 12^{th} order electrical circuit.

To simplify the graphical representation of the hypergraph, a variable, x_i, and its observed value, x_i^*, are collapsed with the notation x_i^*. However, keep in mind that this is a notational convention that stands for the set $\{x_i, x_i^*\}$ and that only x_i belongs to the constraint where it is located in the graphical representation.

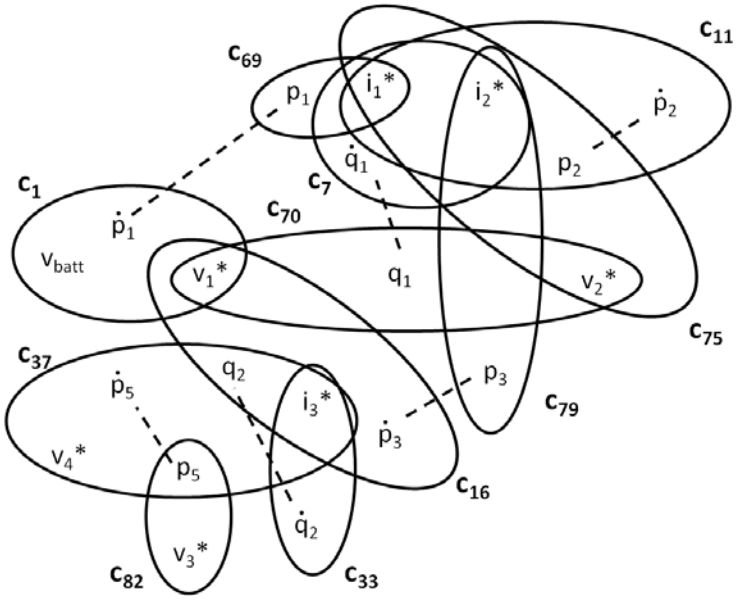


Figure 3.34: Fragment of H_{SD} of the 12^{th} order circuit.

Actually, the set $\{x_i, x_i^*\}$ abstracts the equation $x_i = x_i^*$. In the model based diagnosis community, the set of equations $O_{dm} = \{x_i = x_i^*, \forall x_i \in O\}$, with $O \subset V$ the set of observed variables of the system, is called (diagnosis) observational model. The equations of the diagnosis observational model of the 12^{th} order electrical circuit are presented below:

$$i_1 = {i_1}^* \quad (o1)$$
$$v_1 = {v_1}^* \quad (o2)$$
$$v_2 = {v_2}^* \quad (o3)$$
$$i_2 = {i_2}^* \quad (o4)$$
$$i_3 = {i_3}^* \quad (o5)$$
$$v_3 = {v_3}^* \quad (o6)$$
$$v_4 = {v_4}^* \quad (o7)$$
$$i_4 = {i_4}^* \quad (o8)$$
$$v_5 = {v_5}^* \quad (o9)$$
$$v_6 = {v_6}^* \quad (o10)$$

As it has been previously explained in this Chapter, first step on PCs computation consists on finding all the *Minimal Evaluation Chains, MEC*, contained in H_{SD}. There are 10 $MECs$ in the 12^{th} order electrical circuit. Figures 3.35 and 3.36 show the $MEC1$ and $MEC6$ of the system. Hence, $H_{MEC_1} = \{\{\dot{p}_1, p_1, v_{batt}, v_1, i_1, v_1^*, i_1^*\}, \{\{v_1, v_1^*\}, \{i_1, i_1^*\}, c_1, c_{16}, \{\dot{p}_1, p_1\}\}\}$ and $H_{MEC6} = \{\{\dot{p}_5, \dot{q}_2, p_5, q_2, i_3, v_3, v_4, i_3^*, v_3^*, v_4^*\}, \{\{i_3, i_3^*\}, \{v_3, v_3*\}, \{v_4, v_4^*\}, c_5, c_9, c_{12}, \{\dot{p}_5, p_5\}, \{\dot{q}_2, q_2\}\}\}$.

Second step of PCs computation requires finding *Minimal Evaluation Models (MEMs)* for every MEC. Under integral causality, 10 PCs are found for the 12^{th} Electrical Circuit described in Figure 3.32.

In order to identify the role of the variables involved in a PC computation in a state space context, we will use a variant of the alternative notation introduced in [21] where a model of a system is described by its state, input and output variables.

Definition 11 (State space description of a model)**.** *The state space description of model M_i is given by*

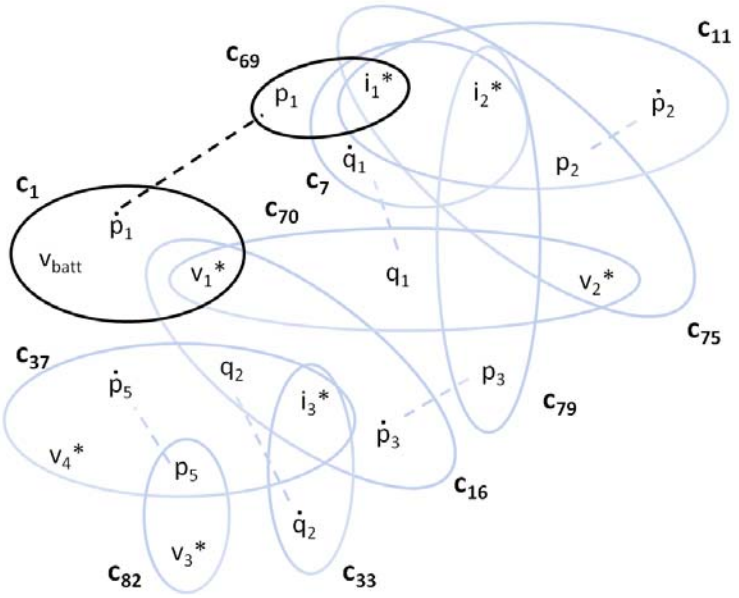


Figure 3.35: MEC1 of the 12^{th} Order Electrical Circuit.

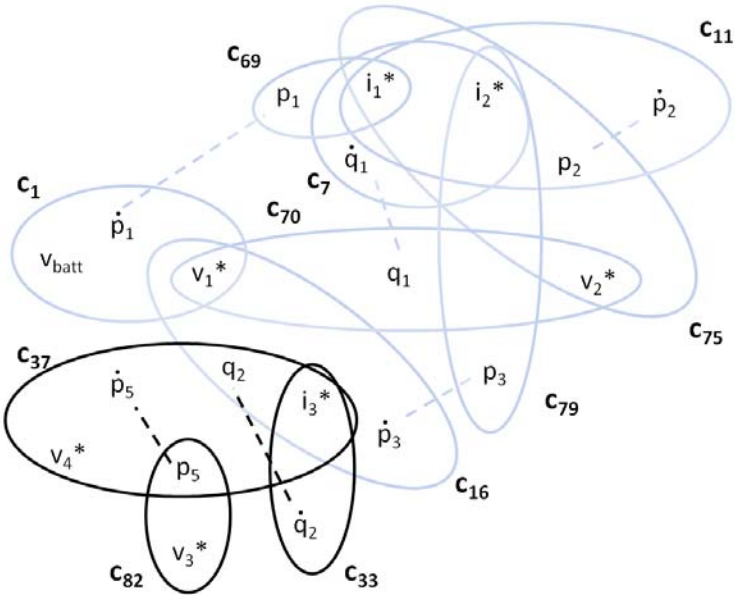


Figure 3.36: MEC6 of the 12^{th} Order Electrical Circuit.

$M_i = (X_i, U_i, Y_i)$ where X_i, U_i and Y_i are the state, input and output variables of the model.

Table 3.5 shows the variables involved in each PC. $PC10$ consists only of instant constraints and does not describe a dynamic subsystem. Hence PCs decomposition provide 9 dynamic subsystems. Among them, size and complexity may vary. $PC1$, with model $M_1 = (\{p_1\}, \{v_{batt}, v_1\}, \{i_1\})$, only includes one state variable and two inputs. On the contrary, $PC9$, with model $M_9 = (\{p_6, p_7, q_4, q_5\}, \{v_4, v_6\}, \{v_5\})$, is one of the most complex with four state variables.

Figures 3.37, 3.38, 3.39, 3.40, 3.41, 3.42, 3.43, 3.44, 3.45, 3.46 present the MEMs of the 10 PCs in this system.

	X	**U**	**Y**
PC1	p_1	v_{batt}, v_1	i_1
PC2	q_1	i_1, i_2, v_2	v_1
PC3	p_2	i_1, i_2	v_2
PC4	p_3, q_2	v_1, i_3	i_2
PC5	q_2, p_4, p_5	i_2, v_3, v_4	i_3
PC6	p_5, q_2	v_4, i_3	v_3
PC7	q_2, q_3, p_5, p_6	i_3, v_3, v_6	v_4
PC8	p_6, p_7, q_5	v_4, v_6	i_4
PC9	p_6, p_7, q_4, q_5	v_4, v_6	v_5
PC10		i_4, v_5	v_6

Table 3.5: State, Input, Output PCs table.

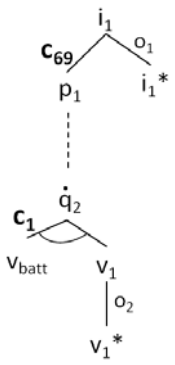


Figure 3.37: MEM from the PC1 of the 12^{th} Order Electrical Circuit.

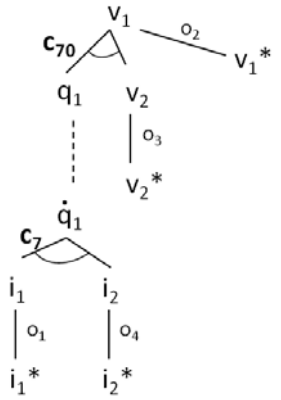


Figure 3.38: MEM from the PC2 of the 12^{th} Order Electrical Circuit.

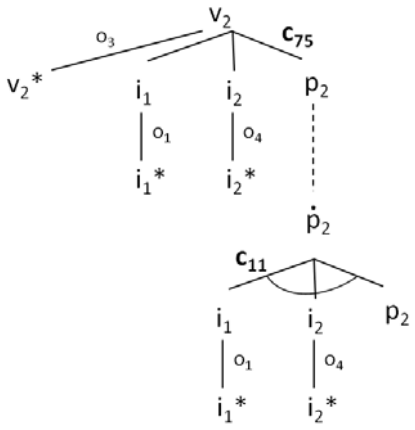


Figure 3.39: MEM from the PC3 of the 12^{th} Order Electrical Circuit.

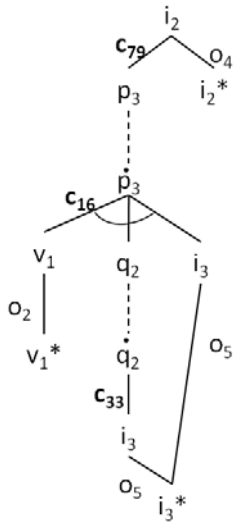


Figure 3.40: MEM from the PC4 of the 12^{th} Order Electrical Circuit.

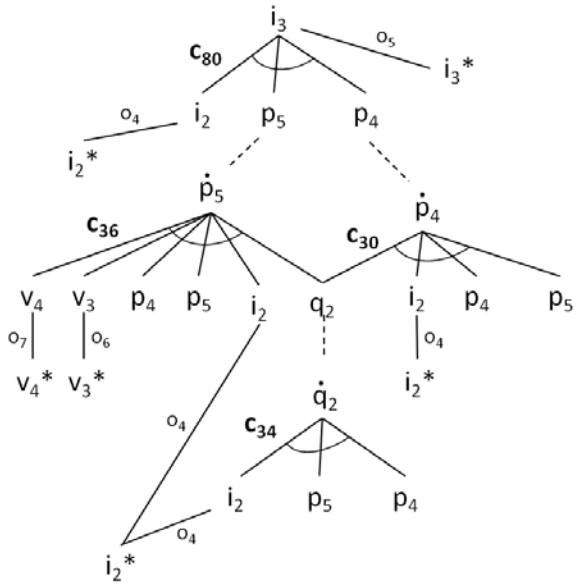


Figure 3.41: MEM from the PC5 of the 12^{th} Order Electrical Circuit.

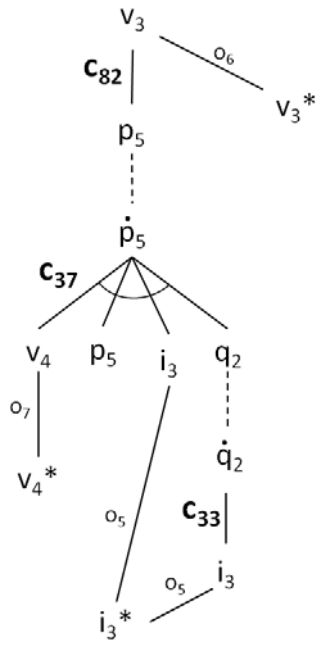


Figure 3.42: MEM from the PC6 of the 12^{th} Order Electrical Circuit.

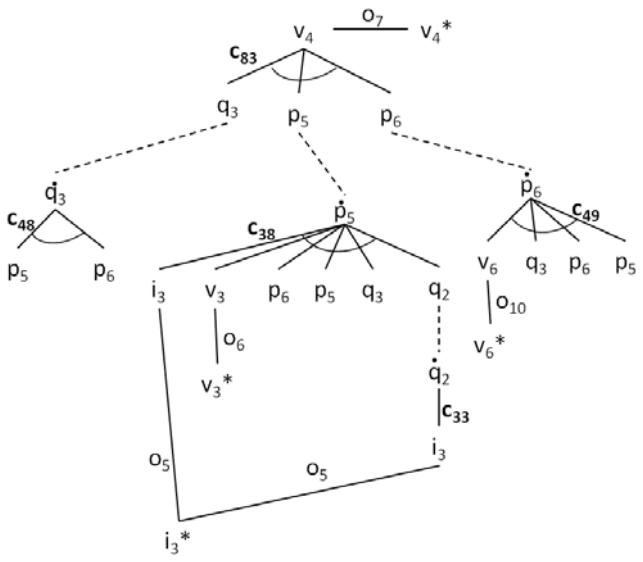


Figure 3.43: MEM from the PC7 of the 12^{th} Order Electrical Circuit.

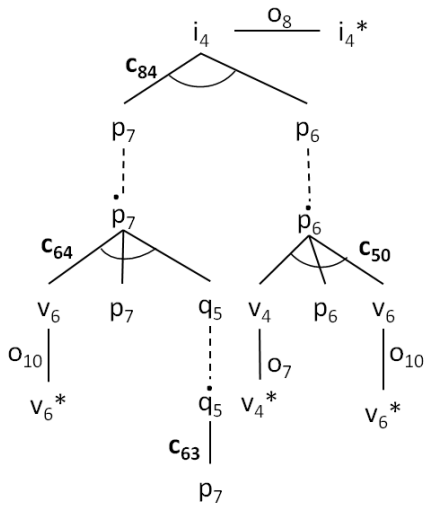


Figure 3.44: MEM from the PC8 of the 12^{th} Order Electrical Circuit.

Figures 3.47 to 3.55 show the 9 DBNs derived from the PCs decomposition using the method previously presented in this Chapter.

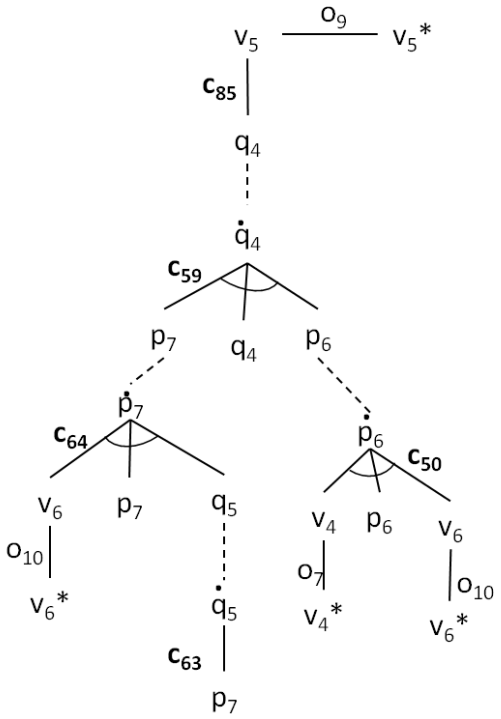


Figure 3.45: MEM from the PC9 of the 12^{th} Order Electrical Circuit.

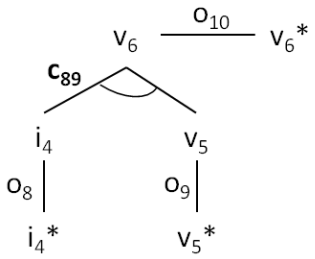


Figure 3.46: MEM from the PC10 of the 12^{th} Order Electrical Circuit.

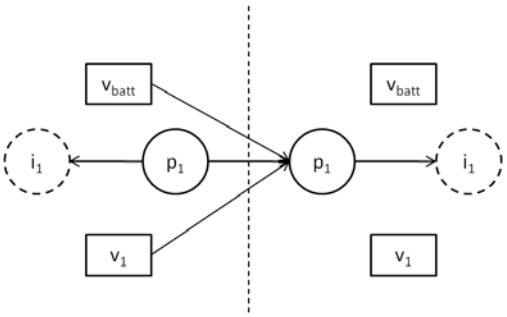


Figure 3.47: DBN derived from the PC1 of the 12^{th} Order Electrical Circuit.

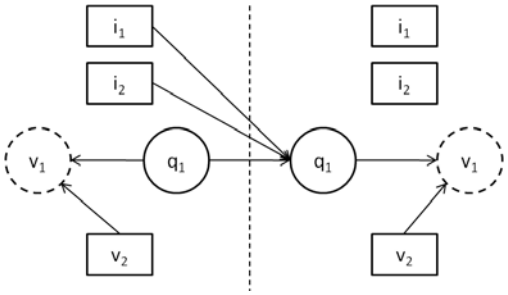


Figure 3.48: DBN derived from the PC2 of the 12^{th} Order Electrical Circuit.

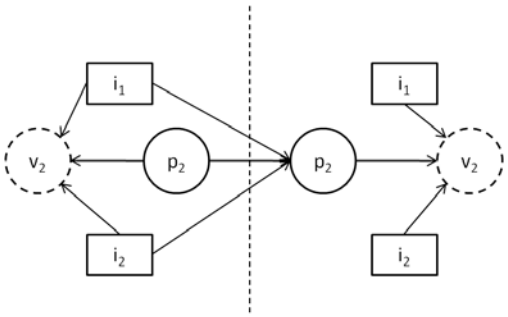


Figure 3.49: DBN derived from the PC3 of the 12^{th} Order Electrical Circuit.

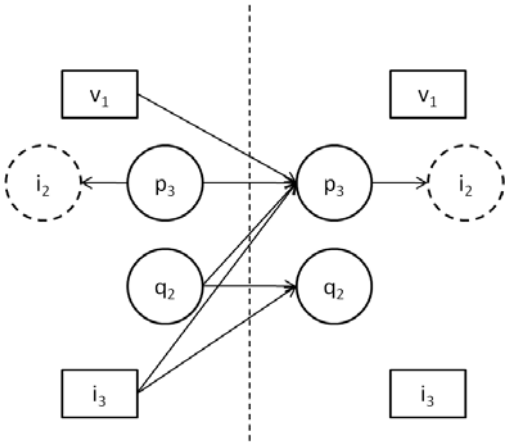


Figure 3.50: DBN derived from the PC4 of the 12^{th} Order Electrical Circuit.

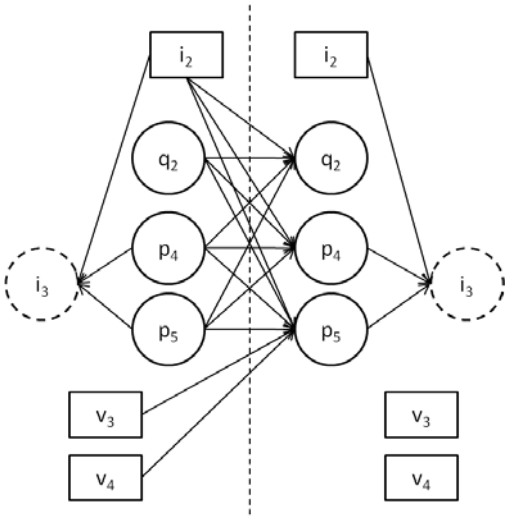


Figure 3.51: DBN derived from the PC5 of the 12^{th} Order Electrical Circuit.

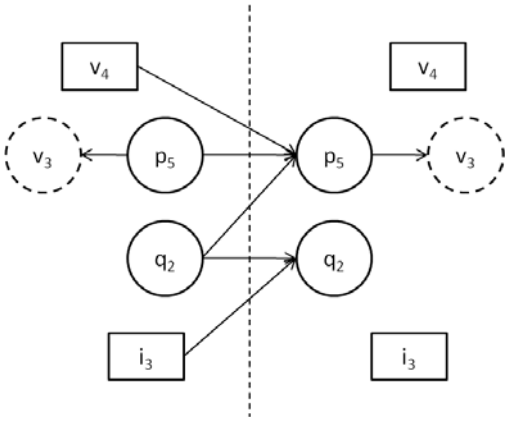


Figure 3.52: DBN derived from the PC6 of the 12^{th} Order Electrical Circuit.

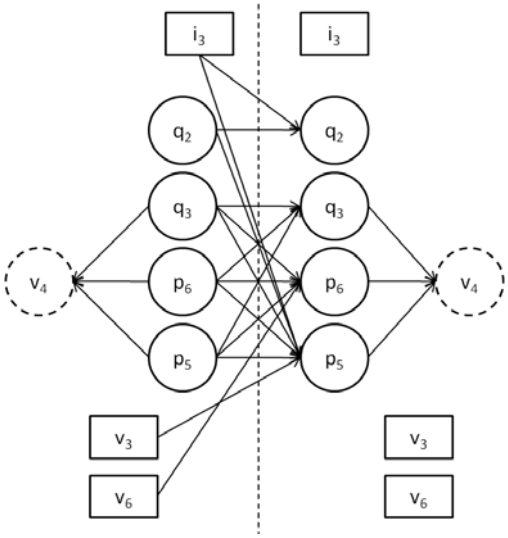


Figure 3.53: DBN derived from the PC7 of the 12^{th} Order Electrical Circuit.

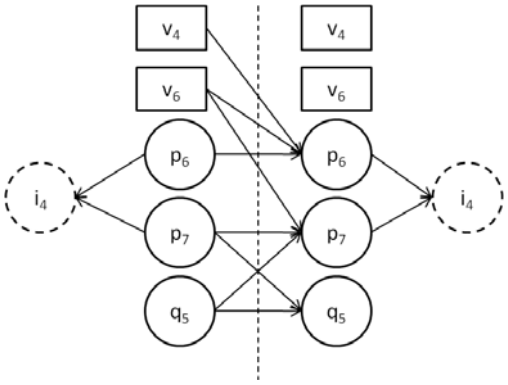


Figure 3.54: DBN derived from the PC8 of the 12^{th} Order Electrical Circuit.

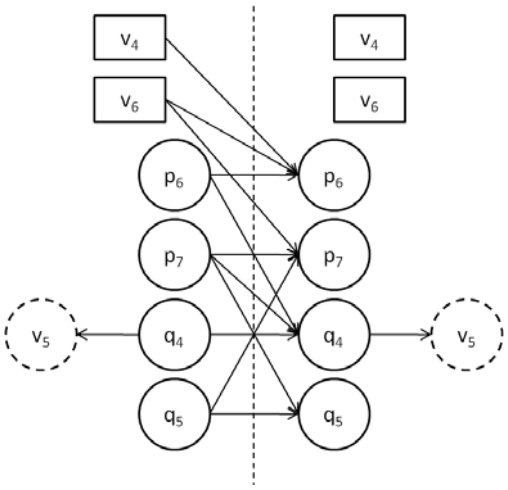


Figure 3.55: DBN derived from the PC9 of the 12^{th} Order Electrical Circuit.

This decomposition is not a partition of the DBN of the complete system because there is some overlap among state variables between several PCs. However, the number of nodes and edges in each DBN derived from a PC is drastically smaller than those in the DBN modeling the complete system.

3.6.2 Diagnosis results

We are going to explain the experimental behaviour of the minimal DBNs and, when pertinent, compares it against the behaviour of the DBN of the complete system.

The overall diagnosis approach is described in Figure 3.23. Fault detection is performed with a Particle Filtering simulation of all the minimal DBNs, which can be run in parallel. Fault isolation is performed with the fault signature matrix that summarizes which components belong to which conflicts. Table 3.6 shows the fault signature matrix of the 12^{th} order electrical circuit. Computing the minimal hitting sets provides fault isolation results using only models of correct behaviour. Fault identification requires simulating faulty DBNs for each fault candidate, which can also be run in parallel. Candidates whose faulty DBN do not converge are rejected until, eventually, only one DBN will converge to the actual system behaviour, computing also an estimate for the faulty parameter.

All the experiments have been run in the same computer, an Intel Core i7 860 at 2.80 GHz with 8 Gb of RAM memory and running Microsoft Windows 7 Professional (64 bits) operating system. The DBN of the complete system has been simulated using 1000 particles, which results in a ratio of approximately 83 particles per state variable. We have kept this ratio on the minimal DBNs. Decision making regarding fault

	PC1	PC2	PC3	PC4	PC5	PC6	PC7	PC8	PC9
$\mathbf{C_1}$		1							
$\mathbf{C_2}$				1	1	1	1		
$\mathbf{C_3}$							1		
$\mathbf{C_4}$									1
$\mathbf{C_5}$								1	1
$\mathbf{L_1}$	1								
$\mathbf{L_2}$			1						
$\mathbf{L_3}$				1					
$\mathbf{L_4}$					1				
$\mathbf{L_5}$					1	1	1		
$\mathbf{L_6}$							1	1	1
$\mathbf{L_7}$								1	1
$\mathbf{R_1}$			1						
$\mathbf{R_2}$				1	1	1	1		
$\mathbf{R_3}$						1			
$\mathbf{R_4}$							1		
$\mathbf{R_5}$							1	1	1
$\mathbf{R_6}$									1
$\mathbf{R_7}$									
$\mathbf{R_8}$								1	1

Table 3.6: Signature matrix of the 12th order electrical system in Figure 3.32

detection and DBN convergence was done with Ztest statistical test [13, 53]. All data has 2% Gaussian noise in the measurements. Simulation starts always from the same and known initial state. All the experiments have been repeated ten times and mean value and standard deviation of the results are presented. References to confidence intervals and statistically significant differences assume a t-student distribution with 9 freedom degrees and $\alpha = 0.95$.

Tracking nominal behaviour

We have tracked the nominal behaviour of the 12^{th} order electrical circuit or 5000 simulation time steps, which accounts for 500 seconds of elapsed real time.

All the DBNs are able to track the nominal behaviour of the system and there is no false positive detection. Table 3.7 shows, in seconds, the elapsed real time required to simulate each DBN for 5000 time steps. Computing time is highly reduced by the Minimal DBNs. If they are run in parallel, worst case computing time is obtained by PC9 which reduces computing time 86.6% respect to the DBN of the complete system. Even if minimal DBNs are sequentially run, total running time sums up to 50.77 seconds which still reduces computing time respect to the DBN of the complete system by 30.7%.

Fault Detection, Isolation and Identification

The fault profiles considered in this case study are the abrupt faults, already presented in this Chapter, and the incipient faults.

Incipient fault An incipient fault is characterized by a gradual, slow drift in a parameter value. The temporal profile of a parameter with an incipient fault, $p^i(t)$ is given by:

$$p^i(t) = \begin{cases} p(t) & t < t_f \\ p(t) + d(t) = p(t) + \sigma_p^i \times (t - t_f) & t \geq t_f \end{cases}$$

	N. Particles	Mean exect. time
Complete	1000	73.28 (2.200)
DBN1	83	3.58 (0.098)
DBN2	83	3.61 (0.086)
DBN3	83	3.67 (0.087)
DBN4	166	4.67 (0.410)
DBN5	249	5.35 (0.160)
DBN6	166	4.60 (0.330)
DBN7	332	9.96 (0.310)
DBN8	249	5.52 (0.470)
DBN9	332	9.81 (0.260)

Table 3.7: Mean execution time (seconds) and standard deviation for all DBNs.

where $p(t)$ represents the nominal parameter value, $d(t)$ is a drift function, and t_f is the point of the fault ocurrence.

The nominal values of the system parameters are: $L_1 = 10$, $L_2 = 20$, $L_3 = 30$, $L_4 = 40$, $L_5 = 50$, $L_6 = 60$, $L_7 = 70$, $C_1 = 100$, $C_2 = 200$, $C_3 = 300$, $C_4 = 400$, $C_5 = 500$, $R_1 = 8 \cdot 10^{-3}$, $R_2 = 7 \cdot 10^{-3}$, $R_3 = 6 \cdot 10^{-3}$, $R_4 = 5 \cdot 10^{-3}$, $R_5 = 4 \cdot 10^{-3}$, $R_6 = 3 \cdot 10^{-3}$, $R_7 = 2 \cdot 10^{-3}$, $R_8 = 10^{-3}$.

We have considered four single fault scenarios: an abrupt fault in C_2, C_2a, an incipient fault in L_3, L_3i, an incipient fault in R_2, R_2i, and an incipient fault in R_4, R_4i. Fault percentage magnitude and drifts are 90%, 5%, 1% and 2% respectively. Drift in L_3i reduces the value of the parameter, which explains the negative faulty value of this parameter: it is the slope of the drifting.

Detection Time The DBN of the complete system is able to detect the four faults injected. The minimal DBNs detect the faults according to the system fault signature matrix. There are neither false positive nor false negative detections. The DBN of the complete system detects C_2a slightly earlier than DBN6[8], although DBN7 incurs in a considerable delay. All the DBNs detect the incipient faults at the same time step.

	$\mathbf{C_2a}$	$\mathbf{L_3i}$	$\mathbf{R_2i}$	$\mathbf{R_4i}$
Complete	4	460	460	460
DBN1				
DBN2				
DBN3				
DBN4	9	460	460	
DBN5	8		460	
DBN6	7		460	
DBN7	54		460	460
DBN8				
DBN9				

Table 3.8: Detection time for DBN of the complete system and minimal DBNs in simulation steps (the standard deviation is always zero).

Partial Isolation Time The isolation approach has been previously presented, the fault signature matrix of the system (Table 3.6) will be used to generate the set of fault candidates as new detections appear. In

[8]Minimal DBNs will be named DBNi, with i the number of the original PC, when we will refer to a specific minimal DBN.

these experiments, the parameter k has been adjusted to 5 simulation time steps. Hence, the identification stage will start as soon as one of the following conditions is satisfied:

1. Current candidates cannot be further refined, according to the fault signature matrix.

2. There has elapsed 5 time steps from the first fault detection, assuming non intermittent faults.

For C_2a, first condition is satisfied when PC6, PC5 and PC4 are confirmed, generating two single fault candidates: $\{C_2, R_2\}$ two time steps after the fault was initially detected by PC6. For L_3 only PC4 is confirmed, generating three fault candidates, $\{C_2,\ L_3,\ R_2\}$, five time steps after the fault is detected. For R_2i PC4, PC5, PC6 and PC7 are simultaneously confirmed and two single fault candidates are created, $\{C_2, R_2\}$, immediately after the fault is detected. For R_4i, only PC7 is confirmed, and seven single fault candidates are generated, $\{C_2, C_5, L_5, L_6, R_2, R_4, R_5\}$, five time steps after the fault is detected.

Identification Time To identify the faulty parameter, faulty minimal DBNs are simulated. It is only necessary to simulate a single faulty network for each fault candidate. However, to gain insight on the behaviour of the minimal DBNs, we have simulated a faulty DBN for each PC that is confirmed and for each fault candidate it contributes. Also, four different faulty DBNs of the complete system are considered.

	$\mathbf{C_2a}$	$\mathbf{L_3i}$	$\mathbf{R_2i}$	$\mathbf{R_4i}$
Complete	507 (88)	371 (148)	301 (0.0)	363 (11)
DBN4	301 (0.0)	394 (35)	302 (3.8)	
DBN5	301 (0.0)		301 (0.0)	
DBN6	343 (0.0)		309 (18)	
DBN7	380 (11)		304 (8.8)	346 (52)

Table 3.9: Faulty DBNs mean time to converge, in simulation time steps. Standard deviation is shown in brackets.

Table 3.9 shows the convergence time, in simulation time steps, of all the faulty DBNs that converge to the injected fault. It is noteworthy that faulty DBNs of candidates different from the injected one do not converge in any case. For C_2a, any minimal DBN reduces the time to converge to the parameter value respect of the DBN of the complete system, although convergence speed depends on the specific DBN used. For L_3i, the convergence of DBN4 is slower than the convergence of the complete DBN. For R_2i, minimal DBNs convergence time is slightly worse than for the complete system, except for DBN5 which converges in the same time than the complete DBN. For R_4i, DBN7 converges quicker than the complete DBN. However, if we consider the dispersion of the mean convergencen time, differences on time are not statistically significant, except for parameter C_2a. DBN4, DBN5 and DBN6 convergence time for C_2a is significantly different from the complete DBN convergence time for parameter C_2a: they converge earlier than the complete DBN and always on the same time instant. Standard deviation is rather large for the complete DBN in faults C_2a and L_3i and for DBN7 in R_4i.

Identification accuracy Table 3.10 presents the estimations obtained with the faulty DBNs. Except for DBN7, which estimations considerably differ from the real faulty parameter values, the behaviour of the faulty DBNs is accurate enough to be used for online fault diagnosis. For C_2a, DBN6 provides the most accurate estimation, although the other minimal DBNs behaves worse than the complete DBN, with DBN7 particularly bad. For L_3i minimal DBNs does not improve on the complete network, although both estimates are comparable. Similar results are obtained for R_2i, although now DBN6 converges to the real value of the faulty parameter. However, for R_4i we can only use DBN7 that incorrectly identifies a value one order of magnitude larger than the real parameter value. Considering the standard deviation of the estimated parameter value, except for DBN7 and the complete DBN for C_2a, all confidence intervals include the real parameter value and the estimated values are not significantly different.

	C_2a	L_3i	R_2i	R_4i
Faulty Value	20	-0.005	0.001	0.002
Complete	27.88 (1.1)	-0.0051 ($5.7 \cdot 10^{-5}$)	0.0011 ($9.3 \cdot 10^{-6}$)	0.0018 ($2.5 \cdot 10^{-4}$)
PC4	37.19 (9.2)	-0.0056 ($6.0 \cdot 10^{-5}$)	0.00083 ($6.5 \cdot 10^{-5}$)	
PC5	35.42 (5.2)		0.0011 ($4.2 \cdot 10^{-5}$)	
PC6	23.76 (4.2)		0.001 ($8.2 \cdot 10^{-5}$)	
PC7	125.58 (5.8)		0.00049 ($2.7 \cdot 10^{-3}$)	0.0204 ($5.7 \cdot 10^{-5}$)

Table 3.10: Mean estimated value of the faulty parameter and the standard deviation.

Summary The results of these experiments show that fault detection and fault isolation stages with minimal DBNs are clearly defined and that minimal DBN improves over the DBN of the complete system, because of their shorter running time and their higher isolation capability. However, due to the overlap of the subsystems, several minimal DBNs may be used to identify the same faulty parameter. Selecting the minimal DBN most suitable to identify a particular faulty parameter seems to be system dependent. The accuracy of the estimation of some minimal DBNs is better or equal than the accuracy achieved by the complete DBN, although most of them provide worse estimations. The time to converge of the minimal DBNs may significantly improve on the complete DBN, but half of the tested minimal DBNs converge on similar or slightly longer time. There is also a tradeoff between accuracy and convergence time because the minimal DBN that achieves the greatest accuracy, DBN6, is not the fastest to converge.

Table 3.11 summarizes the diagnosis results for the four considered scenarios, indicating the total time elapsed from fault injection to final fault identification, selecting the most accurate minimal DBN when there are more than one available.

	C_2a	L_3i	R_2i	R_4i
Detection Time	7	460	460	460
Isolation Time	2	5	0	5
Identification Time	343	394.2	301	346
Total Time	355	859.2	761	811
Faulty Value	20	-0.005	0.001	0.001
Estimated Value	23.76	-0.0056	0.0011	0.02

Table 3.11: Summary of the four fault diagnosis scenarios.

3.6.3 Merging minimal DBNs to improve fault identification

Motivation

Up to now, we have exploited the decomposition induced by the PCs of a system to generate a decomposition of the DBN of a complete system creating a minimal DBN from each dynamic PC. We have shown that this approach allows performing online Consistency Based Diagnosis (fault identification and initial fault isolation) with the set of minimal DBNs in an efficient manner, because simulation time of every subsystem is smaller than simulation time of the DBN of the complete system and they can be run in parallel. Initial isolation capability of the approach is higher than that of the complete DBN, which requires starting the identification process to isolate faults, because these DBNs are minimal and isolability decision logic is provided by CBD. CBD exploits minimality of subsystems to obtain maximun isolation information using only models of correct behaviour.

However, using minimal subsystems do not take advantage of the analytical redundancy of the system, redundancy that generally improves convergence of any identification algorithm.

Experimental results presented in previous subsection show that minimal DBNs can be used to perform fault identification, because the faulty DBNs that models the current fault is the only DBN that converges.

Minimal faulty DBNs may even improve identification stage compared to the complete DBN like first scenario shows, because DBN6 noticeably improves on convergence speed and parameter value estimation compared to the complete DBN. However, this good behaviour of minimal DBNs for fault identification is not consistently achieved, because the opposite behaviour may also be found. See, for instance, the fourth scenario, with fault $R_4 i$, where minimal DBN parameter value estimation is one order of magnitude bigger than the actual value of the faulty parameter, while the complete DBN estimation differs of the real faulty value just in the magnitude of the parameter value.

Hence, experimental results confirm the hypothesis that minimal subsystems may not always be the best option for fault identification with DBNs.

In order to improve fault identification algorithms behaviour, we propose to generate subsystems merging PCs of the system, and afterwards, generating new DBNs. Our intuition is that merging overlapping PCs, that is, PCs that have at least one common state variable, may improve the identification stage with DBNs because of three causes. First, sharing at least one common state variable means that merging them may exploit the analytical redundancy of the system, because there are options that the shared state variables are computed by different paths in each PC. Second, merging two PCs, we increase the number of outputs, which may improve Particle Filtering simulation because of a bigger observational model. Third, there are opportunities that some observed variables, which are inputs in the PCs, does not occur in the resulting subsystem, because analytical redundancy may allow estimating its value from other inputs. Given that observations are affected by noise, reducing the number of observations that are used as inputs to the subsystem will reduce the global noise introduced into the subsystem, improving again Particle Filtering simulation.

A criterion to merge minimal DBNs.

Merging PCs to obtain multiple output subsystems was first proposed in [18] in the context of distributed diagnosis to generate decentralized distributed diagnosis systems. The proposed merging strategy does not apply to this problem because it looks for different goals. Moreover, it requires the complete system to be globally diagnosticable, which is not usually the case. But the method to merge two PCs was already established in the aforementioned work.

Definition 12 (PCs Merging)**.** *Given PC1 and PC2, with models* $M_1 = \{X_1, U_1, \{y_1\}\}$ *and* $M_2 = \{X_2, U_2, \{y_2\}\}$*, respectively, we denote PC1-2 to the subsystem whose model is* $M_{1-2} = \{X_{1-2}, U_{1-2}, \{y_1, y_2\}\}$*, where* $X_{1-2} = \bigcup(X_1, X_2)$ *and* $U_{1-2} = \bigcup(U_1, U_2) - \{y_1, y_2\}$*.*

Take into account that although the merging procedure is described in terms of the models state/input/output description, it has to be performed at equation level. We will illustrate the merging procedure with PC5 and PC6, whose behaviour is described by equations $c30$, $c34$, $c36$ and $c80$ and $c33$, $c37$ and $c82$, respectively.

$$\dot{p}_4 = \frac{q2}{C2} + (i_2 - \frac{p4}{L4} - \frac{p5}{L5}) * R2 \quad (c30)$$

$$\dot{q}_2 = i_3 \quad (c33)$$

$$\dot{q}_2 = i_2 - \frac{p4}{L4} - \frac{p5}{L5} \quad (c34)$$

$$\dot{p}_5 = \frac{q2}{C2} + (i_2 - \frac{p4}{L4} - \frac{p5}{L5}) * R2 - v_3 - v_4 \quad (c36)$$

$$\dot{p}_5 = \frac{q2}{C2} + i_3 * R2 - \frac{p5}{L5} * R3 - v_4 \quad (c37)$$

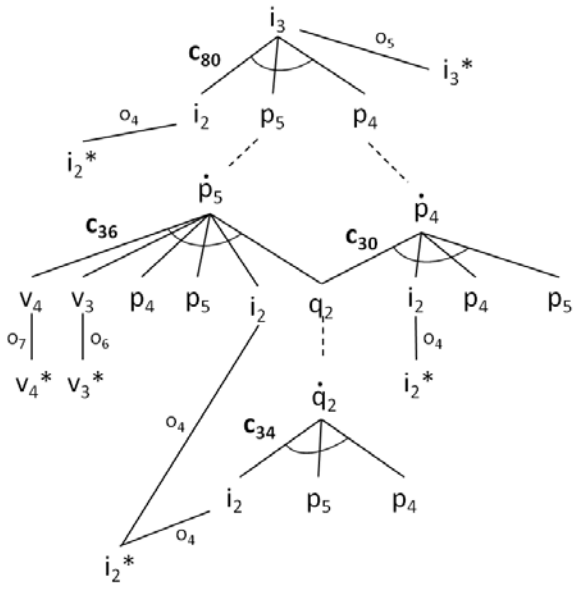


Figure 3.56: MEM5 of the 12^{th} Order Electrical Circuit.

$$i_3 = i_2 - \frac{p4}{L4} - \frac{p5}{L5} \quad (c80)$$

$$v_3 = \frac{p5}{L5} * R3 \quad (c82)$$

Merging PC5 and PC6 generates PC5-6, described by equations $c30$, $c34$, $c41$, $c80$ and $c82$.

$$\dot{p}_4 = \frac{q2}{C2} + (i_2 - \frac{p4}{L4} - \frac{p5}{L5}) * R2 \quad (c30)$$

$$\dot{q}_2 = i_2 - \frac{p4}{L4} - \frac{p5}{L5} \quad (c34)$$

$$\dot{p}_5 = \frac{q2}{C2} + (i_2 - \frac{p4}{L4} - \frac{p5}{L5}) * R2 - \frac{p5}{L5} * R3 - v_4 \quad (c41)$$

$$i_3 = i_2 - \frac{p4}{L4} - \frac{p5}{L5} \quad (c80)$$

$$v_3 = \frac{p5}{L5} * R3 \quad (c82)$$

The structural effect of merging two PCs is better appreciated if we look at the $MEMs$' structure of the involved PCs. Consider PC5, whose model is $M_5 = \{\{q_2, p_4, p_5\}, \{i_2, v_3, v_4\}, \{i_3\}\}$ and whose MEM is shown in Figure 3.56, and PC6, whose model is $M_6 = \{\{q_2, p_5\}, \{i_3, v_4\}, \{v_3\}\}$ and whose MEM is described in Figure 3.57. The subsystem generated merging both PCs is PC5-6, whose model is $M_{5-6} = \{\{q_2, p_4, p_5\}, \{i_2, v_4\}, \{i_3, v_3\}\}$ and whose MEM is shown in Figure 3.58.

Figure 3.58 shows how the analytical redundancy of the system allows to completely eliminate v_3 and i_3 from the inputs of the model of *MEM5-6*. To compute $\dot{p}_5$ PC5 employs equation $c36$ that needs v_3 as

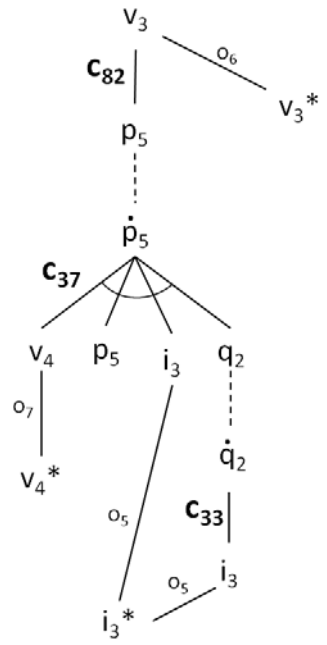


Figure 3.57: MEM6 of the 12^{th} Order Electrical Circuit.

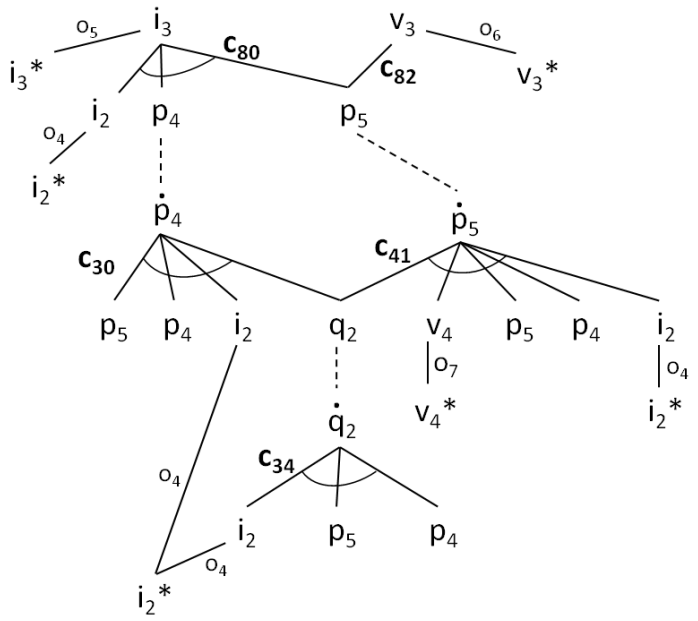


Figure 3.58: MEM of the merging of PC5 and PC6 from the 12^{th} Order Electrical Circuit.

input while PC6 has equation $c37$ with i_3 as input. Introducing equation $c82$ in $c36$ we obtain equation $c41$, v_3 is no longer an input and $c37$ is no longer needed, eliminating the input i_3. *MEM5-6* computes $\dot{p}_5$ with equation $c41$ and it computes the same outputs that the minimal PC5 and PC6 although with two inputs less.

Once introduced the method to merge two PCs, which is easily extendable to merge any two subsystems, the next issue we face is deciding which PCs, and how many PCs, are going to be merged. This is a key aspect, because the merging procedure induces a lattice structure over the set of PCs. Hence, we have to consider $(2^k - k)$ potential new subsystems, being k the number of minimal PCs, which easily grows to a huge number. In the 12^{th} order electrical circuit, with 9 PCs, we have to consider $(2^9 - 9)$ potential new subsystem, as the result of merging different number of PCs. Although the real number of subsystems that we have to consider may be smaller, because we will only merge subsystems that share at least one state variable, the number of potential subsystems is still huge. Hence, some criteria is needed to decide on which PCs to merge.

Taking into account the intuitions stated in the motivation of this section, and considering how PC merging works, we have developed the following structural heuristic to select the PC to be merged with another subsystem.

Definition 13 (Structural Merging Heuristic for parameter Identification, SMHI)**.** *When merging a PC with a subsystem to improve fault identification, and this PC may be chosen from a set of available PCs, select the PC to be merged according to the following criteria:*

1. *Merge only subsystems that have, at least, one state variable in common.*
2. *If possible, select pairs of subsystems that satisfy the following restrictions, i) one output of the first subsystem is an input of the second subsystem, and ii) one output of the second subsystem is an input of the first subsystem.*
3. *If several pairs satify criteria 1 and 2 (or criterion 1 when no pair satisfies criterion 2) choose the pair that generates the subsystem with the minimum number of state variables.*
4. *If several pairs also satisfy criterion 3, choose the pair that generates the subsystem with the minimum number of inputs.*
5. *If several pairs also satisfy criterion 4, choose one at random.*

This heuristic, specifically designed to improve parameter identification considering only structural information, only merges subsystems that share at least one state variable, to assure that we are dealing with subsystems that overlap and, consequently, its merging may exploit the analytical redundancy of the system. Then, the fundamental criterion of the heuristic is criterion 2, which assure that the effect of merging a PC with another subsystem always add one output to the subsystem while simultaneously decreasing the total number of inputs, which is the primary effect that we look for with this heuristic. If there is more than one pair of subsystems that satisfy criterion 2, then we sequentially apply criteria that may improve fault identification, ranking first minimizing the total number of state variables on the resulting subsystem, because we prefer smaller subsystems to reduce computational load of Particle Filtering simulation. It must be noted that there is the possibility that no two PCs satisfy criterion 2. In this particular case, main criterion becomes criterion 3, which prefers pairs that generate subsystems with the smallest number of state variables.

We can illustrate the effect of this heuristic in the 12^{th} order electrical circuit. Assume that we are interesting on merging PCs from PC4 to PC7. Given that any pair of these PCs shares at least one state variable, any element of their pairwise combination satisfies criterion 1. Table 3.12 shows the model description of the resultant subsystems. From them, there are four pairs that also satisfy criterion 2: PC4-5, PC5-6, PC5-7 and PC6-7. If no other restriction applies, like for instance the need to include PC4 or PC7 because we want to identify a faulty parameter that only occurs on the fault signature matrix of one of those PCs, like L_3 or C_3, SMHI will select PC5-6 because it is the one with the lesser number of state variables. Figures 3.56, 3.57 and 3.58 show the overall effect of criterion 2 when creating PC5-6.

	X	U	Y
PC4	p_3, q_2	v_1, i_3	i_2
PC5	q_2, p_4, p_5	i_2, v_3, v_4	i_3
PC6	p_5, q_2	v_4, i_3	v_3
PC7	q_2, q_3, p_5, p_6	i_3, v_3, v_6	v_4
PC4-5	q_2, p_3, p_4, p_5	v_1, v_3, v_4	i_2, i_3
PC4-6	q_2, p_3, p_4	v_1, v_4, i_3	i_2, v_3
PC4-7	q_2, q_3, p_3, p_5, p_6	v_1, v_3, v_6	i_2, v_4
PC5-6	p_2, p_4, p_5	i_2, v_4	i_3, v_3
PC5-7	q_2, q_3, p_4, p_5, p_6	i_2, v_3, v_6	i_3, i_4
PC6-7	q_2, q_3, p_5, p_6	i_3, v_6	v_3, v_4

Table 3.12: State, Input, Output merged PCs table.

An algorithm to merge minimal DBNs

Algorithm 1 describes a greedy procedure to merge minimal DBNs based on the proposed structural merging heuristic. Its inputs are the set of PCs, the fault signature matrix and the set of faulty parameters that the factors must identify. The algorithm returns a sequence of merged subsystems, each one obtained merging the previous one with the best PC, according to the *structural merging heuristic* . The algorithm is made of two basic procedures, *selectFirstSubSys* and *mergePCs*. *selectFirstSubSys* determines the first subsystem of the sequence. *mergePCs* builds the sequence iteratively adding the PC selected by SMHI to the current subsystem.

Algorithm 1: Merging DBN-PCs. Obtains the sequence of PCs to be merged.

Require: *PCs*, list of PCs to consider; *FaultyP*, list of faulty parameters; *signatureMatrix*, the fault signature matrix.

Ensure: *listMerging*, sequence of merging proposed by SMHI.

1: *preferCandidates* = {*pcs*, *pcs* ∈ *PCs* & *pcs* is sensitive to some element from *FaultyP* according to *signatureMatrix* };
2: *otherCandidates* = *PCs* \ *preferCandidates*;
3: $InitSubSys = selectFirstSubSys(PCs, preferCandidates, FaultyP, signatureMatrix)$;
4: *preferCandidates* = *preferCandidates* \ already merged *PCs*;
5: $listMerging = mergePCs(InitSubSys, preferCandidates, otherCandidates)$;
6: **return** *listMerging*;

Selecting the first two PCs to create the first non minimal subsystem is critical on the behaviour of the algorithm. Algorithm *selectFirstSubSys* performs this task. First, it checks if there are some PCs that can identify all the considered faults. If this is the case, the SMHI is applied to merge two of these PCs, unless there is only a single PC that includes all the faulty parameters, in which case this PC is automatically selected as the Initial subsystem. If no PC accounts for all the faulty parameters, we consider merging as many PCs as needed to create an initial subsystem sensitive to all the faulty parameters that we want to identify. To do that, we first compute the set of PCs sensitive to each fault, $setF_i$ in line 9. Then, we obtain all the minimal hitting sets of the collection of $setF_i$. Finally, we select the minimal hitting set with less state variables and less input variables.

The algorithm *mergePCs* only have to select the best PC to be merged with the last found subsystem, starting from InitialSubSys and iteratively applying SMHI. The algorithm includes an additional heuristic: merge first with PCs that can also identify the faulty parameters, and consider PCs that cannot identify the faulty parameters afterwards. This heuristic forces merging first subsystems that overlap on the faulty parameter vicinity so that we exploit analytical redundancy related to the faulty parameter. According to SMHI, *mergePCs* stops if no PC shares a state variable with the current subsystem.

Algorithm 2: Select or create the first subsystem to start up the merging process (*selectFirstSubSys*)

Require: *PCs*, list of PCs; *preferCandidates*, list of PCs sensitive to at least one fault; *FaultyP*, list of faulty parameters; *signatureMatrix*, the fault signature matrix.
Ensure: *InitSubSys*, the PC or PC merging to start with.

selectPreferCandidates = {*pcs*, *pcs* ∈ *preferCandidates* & *pcs* is sensitive to all elements from *FaultyP* };
if (|*selectPreferCandidates*| == 1) **then**
  InitSubSys = PC from *selectPreferCandidates*;
else
  if (|*selectPreferCandidates*| > 1) **then**
    InitSubSys = the *best* merging of two PC from *selectPreferCandidates*;
  else
    for each $fault_i \in FaultyP$ **do**
      $setF_i$ = {*pcs*, *pcs* ∈ *preferCandidates* & *pcs* is sensitive to $fault_i$ };
    end for
    InitialSubSysCandidates = collection of Minimal Hitting Set of $setF_i$;
    SmallSubSysCandidates = elements of *InitialSubSysCandidates* with minimum number of state variables;
    InitSubSys = element of *SmallSubSysCandidates* with minimun number of inputs;
  end if
end if
return *InitSubSys*;

Algorithm 3: MergePCs routine (*mergePCs*)

Require: *currentSubSys*, subsystem to start the process; *preferCandidates* list of preferred candidates to merge; *otherCandidates* list of other PCs canditates.
Ensure: *listM*, sequence of merging proposed by SMHI.

listM = { *currentSubSys* }
while (∃ *pc* ∈ *preferCandidates* & *stateVar(pc)* ∩ *stateVar(currentSubSys)* ≠ ϕ) **do**
  bestPC = Select best *pc* ∈ $preferCandidates$ to merge with *currentSubSyst*;
  Remove *bestPC* from *preferCandidates*;
  currentSubSys = $merge(bestPC, currentSubSys)$;
  Add *currentSubSys* to *listM*;
end while
while (∃ *pc* ∈ *otherCandidates* & *stateVar(pc)* ∩ *stateVar(currentSubSys)* ≠ ϕ) **do**
  bestPC = Select best *pc* ∈ $otherCandidates$ to merge with *currentSubSys*;
  Remove *bestPC* from *otherCandidates*;
  currentSubSys = $merge(bestPC, currentSubSys)$;
  Add *currentSubSys* to *listM*;
end while
return listM;

Faulty parameter	**Sequence of merged PCs**				
$\mathbf{C_2a}$	PC5-6	PC4-5-6	PC4-5-6-7	PC4-5-6-7-8	PC4-5-6-7-8-9
$\mathbf{L_3i}$	PC4-5	PC4-5-6	PC4-5-6-7	PC4-5-6-7-8	PC4-5-6-7-8-9
$\mathbf{R_2i}$	PC5-6	PC4-5-6	PC4-5-6-7	PC4-5-6-7-8	PC4-5-6-7-8-9
$\mathbf{R_4i}$	PC5-7	PC5-6-7	PC4-5-6-7	PC4-5-6-7-8	PC4-5-6-7-8-9

Table 3.13: Sequence of merged subsystems for the four scenarios.

Table 3.13 shows the sequence found by Algorithm 1 for the four scenarios of our case study and its nine PCs. For parameters C_2 and R_4, that algorithm has to consider four candidates to create the initial subsystem, PC4, PC5, PC6 and PC7, the four PCs that can detect these faults. In this case the initial subsystem is PC5-6, whose MEM is shown in Figure 3.58. Parameter L_3 is only detected by PC4 and it becomes the initial subsystem. Afterwards, SMHI selects PC5 as the best PC to be merged with it, to create subsystem PC4-5. Similarly, PC7 is the only PC that includes parameter R_4 and it becomes the initial subsystem. For PC7, SMHI chooses PC5 creating the subsystem PC5-7. The four sequences converge in the third step to PC4-5-6-7. Then, Algorithm 1 first add PC8 and finally PC9. It stops with subsystem PC4-5-6-7-8-9 because PC1, PC2 and PC3 do not share any state variable with this last subsystem.

Testing merged DBNs.

Convergence times			**Sequence of Merged DBNs**				
Faulty P.	**Complete DBN**	**Best Min. DBN**	**DBN5-6**	**DBN4-5-6**	**DBN4-5-6-7**	**DBN4-5-6-7-8**	**DBN4-5-6-7-8-9**
C_2a	507 (88)	343 (4.2)	331 (91)	318 (35)	377 (120)	301 (0.0)	327 (83)
R_2i	301 (0.0)	309.1 ($8.2\cdot10^{-5}$)	301 (0.0)	301 (0.0)	301 (0.0)	301 (0.0)	301 (0.0)
	Complete DBN	**Best Min. DBN**	**DBN4-5**	**DBN4-5-6**	**DBN4-5-6-7**	**DBN4-5-6-7-8**	**DBN4-5-6-7-8-9**
L_3i	371 (0.0)	394.2 ($6.1\cdot10^{-5}$)	324 (34)	330 (17)	460 (210)	310 (7.4)	332 (16)
	Complete DBN	**Best Min. DBN**	**DBN5-7**	**DBN5-6-7**	**DBN4-5-6-7**	**DBN4-5-6-7-8**	**DBN4-5-6-7-8-9**
R_4i	363 (11)	346 ($5.7\cdot10^{-3}$)	358 (37)	360 (18)	323 (5.9)	324 (2.7)	359 (28)

Table 3.14: Convergence time for the merged DBNs in the four scenarios. Standard deviation is shown in brackets.

Convergence values			**Sequence of Merged DBNs**				
Faulty P.	**Complete DBN**	**Best Min. DBN**	**DBN5-6**	**DBN4-5-6**	**DBN4-5-6-7**	**DBN4-5-6-7-8**	**DBN4-5-6-7-8-9**
C_2a(20)	27.9 (1.1)	23.8 (4.2)	21.2 (2.8)	22.29 (1.0)	23.1 (1.1)	23.5 (6.7)	22.54 (0.8)
R_2i(0.001)	0.0011 ($9.3\cdot10^{-6}$)	0.0010 ($8.3\cdot10^{-5}$)	0.0010 ($3.8\cdot10^{-5}$)	0.0011 ($3.1\cdot10^{-5}$)	0.0011 ($2.1\cdot10^{-5}$)	0.0011 ($1.6\cdot10^{-5}$)	0.0011 ($8.1\cdot10^{-6}$)
	Complete DBN	**Best Min. DBN**	**DBN4-5**	**DBN4-5-6**	**DBN4-5-6-7**	**DBN4-5-6-7-8**	**DBN4-5-6-7-8-9**
L_3i(-0.005)	-0.0051 ($5.8\cdot10^{-5}$)	-0.0056 ($6.0\cdot10^{-5}$)	-0.0055 ($2.2\cdot10^{-4}$)	-0.0052 ($3.4\cdot10^{-5}$)	-0.0051 ($1.2\cdot10^{-4}$)	-0.0058 ($1.6\cdot10^{-4}$)	-0.0052 ($2.7\cdot10^{-5}$)
	Complete DBN	**Best Min. DBN**	**DBN5-7**	**DBN5-6-7**	**DBN4-5-6-7**	**DBN4-5-6-7-8**	**DBN4-5-6-7-8-9**
R_4i(0.002)	0.0018 ($2.5\cdot10^{-4}$)	0.0204 ($5.7\cdot10^{-3}$)	0.0019 ($3.9\cdot10^{-4}$)	0.0020 ($3.0\cdot10^{-4}$)	$4.0\cdot10^{-5}$ ($9.5\cdot10^{-5}$)	$5.9\cdot10^{-7}$ ($6.5\cdot10^{-5}$)	0.0020 ($2.3\cdot10^{-4}$)

Table 3.15: Convergence value for the merged DBNs in the four scenarios. Standard deviation is shown in brackets.

Tables 3.14 and 3.15, show, respectively, the time to converge and the estimated value of the faulty

parameter for all the merged DBNs proposed by SMHI, and allows comparison with the complete DBN and with the minimal DBN that provides the most accurate estimation of the faulty parameter.

For fault C_2a the first merged DBN of its sequence, DBN5-6, obtains the best estimation of the faulty parameter, although DBN4-5-6 performance is comparable, it has a smaller standard deviation and converges slightly quicker. Both of them improve on the best minimal DBN for this fault, DBN6. The improvement against the complete DBN is rather important as shown in table 3.16. Adding more minimal DBNs deteriorate fault identification, except for the last subsystem that obtains an acceptable estimated value with the smallest standard deviation of the sequence.

The behaviour of the subsystems follows a different pattern for R_2i. All faulty DBNs require the same time to converge and only DBN5-6 converges to the real faulty value. The remaining subsystems converge to the same value than the complete faulty DBN. Improvement of DBN5-6 over the complete DBN is small.

For fault L_3i, the first subsystem of its sequence improves on the only minimal DBN that can identify it, DBN4. The most accurate estimation is obtained by DBN4-5-6-7 although the overall best behaviour is achieved by subsystems DBN4-5-6 and DBN4-5-6-7-8-9 which converge quicker than the complete DBN to a comparable value.

Fault R_4i could only be identified by DBN7, although it converges to a value one order of magnitude larger than the real faulty value. The first subsystem of the sequence, DBN5-7 highly improves the behaviour of DBN7 computing a good estimation of the parameter value. Overall best behaviour is achieved by DBN5-6-7 and DBN4-5-6-7-8-9, that estimate the real faulty parameter and converge slightly quicker than the complete DBN. The behaviour of the remaining DBNs of the sequence is deficient.

Considering the general pattern of the merged DBNs sequences behaviour, several interesting features can be pointed out. First, in every case the first subsystem of the sequence improves or equals the best minimal DBN. We have made additional tests to check the behaviour of other combinations of two minimal subsystems and, in every case, the initial subsystem selected by SMHI outperforms them. Second, improvement on the estimation is not monotonous and adding some subsystem to a well behaved one may even severely deteriorate the behaviour. Combination of three minimal DBNs behaves consistently well in the four scenarios, with good estimations and equal or smaller convergence times than the complete DBN. This consistently good behaviour is also reproduce by the last subsystem, combining subsystems 4, 5, 6, 7, 8 and 9 with the additional advantages that it obtains the smallest standard deviation in the four scenarios and that it is the same subsystem for every system parameter, except for R_1, L_1 and L_2.

Error rate			**Sequence of DBNs**				
Faulty P.	**Complete DBN**	**Best min. DBN**	**2 DBNs**	**3 DBNs**	**4 DBNs**	**5 DBNs**	**6 DBNs**
C_2a	39.4	18.80	**5.85**	11.45	15.65	17.70	12.70
R_2i	10.00	**0.00**	**0.00**	10.00	10.00	10.00	10.00
L_3i	**2.00**	12.00	10.00	4.00	**2.00**	16.00	4.00
R_4i	10.00	920.00	5.00	**0.00**	97.98	99.97	**0.00**

Table 3.16: Error rate (%) for the DBNs sequence in the four scenarios.

Tables 3.16 and 3.17 summarize the behaviour of all the subsystems in terms of the number of minimal DBNs merged. Table 3.16 shows the error rate of the DBNs. It clearly shows that minimal DBNs do not necessarily improve the estimation of the complete DBN, although we can find a configuration that improves or equals the behaviour of the complete DBN. It also shows that, in the tested scenarios, merging two minimal DBNs improves the estimation of the best minimal DBN. Merging three and six DBNs yields consistently good behaviour. Particularly interesting is merging six DBNs, because the resulting DBN is the same in the four scenarios and it is the last DBN generated by the SMHI because it does not share any state variable with DBN1, DBN2 or DBN3. Table 3.17 shows the relative improvement of convergence time of the different DBNs respect to the DBN of the complete system. A negative value means that a DBN converges faster than the DBN of the complete system. The table shows that most of the DBNs converge faster than the complete DBN. Significant exceptions are the best minimal DBNs for R_2i and L_3i. Only one merged DBN

Convergence time improvement			Sequence of DBNs			
Faulty P.	**Best min. DBN**	**2 DBNs**	**3 DBNs**	**4 DBNs**	**5 DBNs**	**6 DBNs**
$\mathbf{C_2a}$	-32.4	-34.8	-37.3	-25.8	-40.7	-35.5
$\mathbf{R_2i}$	2.7	0.0	0.0	0.0	0.0	0.0
$\mathbf{L_3i}$	6.3	-12.7	-10.9	24.0	-16.3	-10.5
$\mathbf{R_4i}$	-4.8	-1.5	-1.0	-11.2	-10.8	-1.3

Table 3.17: Convergence time improvement (%) comparing DBNs of the sequence and the complete DBN in the four scenarios.

is slower than the complete DBN. We can see that merging the best minimal DBN with the minimal DBNs selected by SMHI always reduces convergence time with respect to the complete DBN. And merging three and six DBNs consistently reduces convergence time in the three scenarios. For R_2i merged DBNs require the same time to converge than the complete DBN.

3.7 Discussion and conclusions

We have presented a method to decompose the DBN of a dynamic system into several minimal DBNs, based on the Possible Conflicts framework. The decomposition consist of finding the PCs of the dynamic system and transforming the computational models provided by their MEMs in DBNs, that are minimal because PCs are minimal. These minimal DBNs allow performing online CBD of dynamic systems, using only models of correct behaviour. Hence, we can perform fault detection and partial fault isolation with these minimal DBNs. This is a clear advantage compared to other approaches to fault diagnosis with DBN models [41] that requires simulating the DBN of the complete system extended with additional nodes to identify the values of potential faulty parameters or resorting to additional techniques to perform fault isolation [103]. This factoring has another practical advantage: simulation computing time can be highly reduced, as the results of the experiments have shown. This saving in computing time depends, at least, on the relative size of the minimal DBNs respetc to the DBN of the complete system, which in turns depends on the system redundancy and ovelaping, i.e. the number of sensors and their allocation. With few sensors, like in the running example of the three-tank system, minimal DBNs executing time is comparable to the excuting time of the DBN of the complete system. In systems with high redundancy, like the 12^{th} order electrical circuit, the saving may be important.

We have also developed a method to perform fault identification of dynamic systems based on merging minimal DBNs. The method is based on merging the MEMs of two or more PCs and then transforming the resulting MEM in a DBN. Experimental results show that minimal DBNs are not always the best option to perform fault identification. Convergence speed and accuracy usually improves just merging two PCs. We have developed a heuristic, SMHI, oriented to identify the best initial merging and the best merging strategy to add more minimal DBNs until we obtain a subsystem that does not overlap with any other minimal DBN. Improvement of the subsystems behaviour along the sequence is not monotonous, which seems to be a consequence of exploiting only structural information. Sensitivity of the DBNs to changes in the faulty parameters should be taken into account if we need to find the best subsystem for fault isolation. Nevertheless, the results of the experiments previously discussed suggest that the last subsystem of the sequence has the potential to improve the accuracy obtained with the complete DBN reducing also convergence time, while being competitive with the behaviour of the best minimal DBN. From a practical point of view, having a small number of subsystems to identify all the parameters of the system simplifies the design of the diagnosis system, and we can always resort to a particular merged subsystem if our priority is maximizing the accuracy of parameter fault identification. However, more experimental work should be done to confirm the existence of a common merged subsystem that performs well for a given set of faulty parameters.

Although some other methods have been proposed to decompose a DBN, nearly all of them obtain a

decomposition that approximates the original DBN. In contrast, the decomposition induced by the PCs of the system generates exact DBNs, in the sense that all the minimal DBNs are obtained from the state space equations model of the system and no approximation is introduced. To the best of our knowledge only the proposal of [105] generates an exact factoring of the original DBN. A recent work that proposes exact simplification of Bayesian Networks is [27]. It develops a method based on variable elimination that is related to PCs search of MEMs for a given MEC. However, it only applies to static bayesian networks and does not pretend to decompose the original DBN but to compile part of it to improve simulation time.

The approach of [105] is rather different from the PCs decomposition. It is based on searching conditional independent subsystem that allows partition of the complete DBN in a set of factors. Additionally, each factor has to be structurally observable; otherwise, state estimation is not feasible. Hence, the process to derive the factors has three steps: 1) Generate a maximal factoring; 2) Test observability of each factor; 3) Merge unobservable factors. The first step, finding the maximal factoring, requires substituting some state variables for algebraic equations involving some measurements. There are some freedom degrees in this initial selection. Consequently, the decomposition induced is not unique but depends on a user selected parameter. Moreover, the method cannot cope with faulty parameters occurring at the algebraic equations used to eliminate state variables. In contrast, Minimal DBNs do not have this limitation: what they can detect and diagnose only depends on the diagnosability of the complete system. Finally, as long as the complete system is redundant PCs decomposition can find a minimal DBN to decompose the system, although if redundancy limits to the presence of one sensor this minimal DBN may comprise the whole system. On the contrary, the method of [105] generates initial factors that may be unobservable. Afterwards, it uses a heuristic to merge factors looking for observable factors. Because this search is not exhaustive, it does not guarantee to find an observable factor until it reconstructs the initial complete DBN. Hence, it may not find a decomposition of the system when there is one.

As a summary, we can point out the following conclusions:

- We have proposed a new approach to decompose the DBN of a system into a set of minimal DBNs. This decomposition has the advantage of being exact and unique.
- From the point of view of the application of DBNs for fault diagnosis of dynamic systems, the proposed approach allows performing consistency based diagnosis, i.e. fault detection and partial fault isolation, using only models of correct behaviour. From a practical point of view it facilitates online CBD because computing time of the minimal DBNs is smaller than the computing time of the complete DBN, specially if we distribute minimal DBNs computation.
- A merging strategy of minimal DBNs has been proposed to find subsystems with good behaviour for fault identification. Experimental results have shown that when minimal DBNs or the DBN of the complete system do not produce accurate enough estimations of a faulty parameter, the proposed strategy may find subsystems that outperform them, both in terms of accuracy of the estimation and smaller computing time.

Chapter 4

Hybrid systems fault diagnosis with Possible Conflicts: a HBG-PC approach.

Possible Conflicts can be used in Consistency-based Diagnosis of Continuous systems. At the same time, many systems in our daily lives have some kind of discrete behaviour. Hence, it seems necessary to extend the PCs approach for diagnosing hybrid systems (we focus on those systems that have continuous behaviour commanded by discrete events, that generates different working modes).

Modeling and diagnosis of hybrid systems is a well-known field in Model-based diagnosis. In this dissertation we will focus our modeling approach on Hybrid Bond Graphs, HBGs, that have several advantages over other methodologies, specially that we do not need to enumerate every possible mode. In HBG modeling, discrete behaviour is introduced as special junctions in the Bond Graph, known as switching junctions.

In this Chapter we propose to characterize Possible Conflicts in the Bond Graph framework, to later extend the definition for hybrid systems using HBGs, leading to the Hybrid Possible Conflict, HPC or HBG-PC, concept. Main contribution of the new approach is that the set of HPCs in the system for any working mode can be obtained from the set of HPCs when every switching junction in the system model is set to ON. Main advantage of the approach is that we can track system behaviour by tracking smaller subsystems and analyzing only the discrete changes locally to each HPC.

The presence of switching junctions in hybrid systems introduce a new potential family of faults: discrete faults, that are defined as faults in actuators. These faults introduce highly non-linear behaviour in the system, that must be quickly isolated. In this Chapter, we present a unifying framework for Fault Detection, Isolation and Identification of both parametric and discrete faults using HBG-PCs. In this proposal we are capable to isolate discrete faults combining both structural and qualitative information from the set of active Possible Conflicts, and consequently update the new set of PCs for the new working mode. The proposal is tested in simulation for a four-tank system.

4.1 Motivation

Hybrid systems are frequently found in many engineering fields. It is vital that they work in a nominal and safe state, that is the reason why fault diagnosis in hybrid systems has to be accurate and efficient. The hybrid systems we focus on have a continuous behaviour controlled by discrete events. The behavioural model of this kind of hybrid systems is made up of the continuous behavioural model of each working mode and the discrete events which trigger the changes between them.

There are many proposals in the DX community for fault diagnosis of hybrid systems. They are based on hybrid modeling [76, 83], hybrid state estimation [58, 101], or combination of online state tracking and residual evaluation [11, 9]. Analysing all of them, they present at least one of the following difficulties:

- It is necessary to pre-enumerate all possible configurations or modes in the system, as well as building models for all of them.

- They need to determine somehow the actual woking mode, including the actuators' configuration and the continuous behaviour.

Some authors have proposed hybrid bond graphs [77] as a modeling technique for hybrid systems. The main advantage is that they do not need a pre-enumeration of all possible modes to track the system behaviour. Among hybrid bond graphs (HBGs) there are two main approaches: (1) Use switching elements with fixed causality to model the actuators changing the parameter values [47, 15, 42, 50], and (2) Use ideal switching elements with a change in causality when a change in a switch happens (there is not any change in parameter values due to the mode change)[84]. The second approach will be used in our proposal, as we will explain later in this chapter.

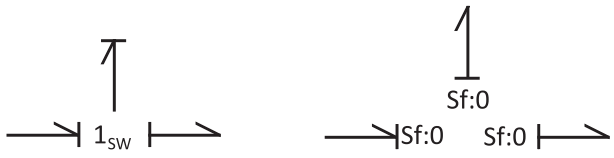
There are some assumptions that must be presented first:

- The actual system configuration, i.e. working mode, is known before a fault occurs.
- The hybrid systems that can be diagnosed with this approach have only discrete actuators: ON/OFF. That is, they connect or disconnect a path or a subsystem but they do not connect alternative paths or subsystems.
- The hybrid bond graph model of the complete system considering all switches are ON has a global valid causal assignment.

4.2 Hybrid Bond Graph Modeling

Bond Graph (BG) modeling approach have been presented in Chapter 3 as a domain-independent energy-based topological modeling language for physical systems [64].

Hybrid Bond Graphs (HBGs) extend BGs by including idealized switching junctions to allow mode changes in the system. If a switching junction is set to *ON*, it behaves as a regular junction. When it changes to *OFF*, all bonds incident on the junction are deactivated forcing 0 flow (or effort) for 1 (or 0) junctions. Those junctions are implemented as a finite state machine *control specification (CSPEC)*. Transitions between the CSPEC states can be triggered by endogenous or exogenous variables, called guards. CSPECs capture controlled and autonomous changes as described in [106]. Figure 4.1 shows the two configurations of the ideal switching junction 1.



(a) *ON* configuration (b) *OFF* configuration

Figure 4.1: Semantics of a switching junction 1.

4.3 Structural decomposition for hybrid systems

As we previously described, main advantage of using hybrid bond graph models is that pre-enumeration of modes in the system is not necessary. However, the main concern of this modeling approach when applied to fault diagnosis of hybrid systems [84] is related to the high time consuming task of causality reassignment for the entire bond graph model. Moreover, during this causality reassignment process, the diagnoser needs to stop tracking the behaviour of the system, making it sensitive to miss faults that occur during (or immediately after) such reassignment process. This drawback will be solved in our proposal, as it will be explained later.

Typically, changes in causality do not propagate, or only a small part of the model needs causality to be reassigned [106]. Moreover, when causality needs to be reassigned, the changes will be local to the hybrid junction. Hence, if we are able to identify the complete set of minimal subsystems for a hybrid system model,

only a subset of these subsystems will be susceptible to suffer from causality reassignment, while the rest of the subsystem will remain with no modifications and will be able to keep on tracking the behaviour of the system.

The proposal of this dissertation consists of decompossing a hybrid system using Possible Conflicts. Then, we have to extend the PCs formalism to the BG framework and more precisely we will use HBGs for Hybrid Systems diagnosis. We will call this new concept Hybrid Possible Conflict, or HBG-PC. In order to define HBG-PCs we proceed first by defining PCs in the BG framework for continuous behaviour estimation, then extending the definition to cope with hybrid behaviour.

4.3.1 Possible Conflicts for Hybrid Systems (HBG-PCs)

Definition 14. *(BG) A BG is a connected graph that is made up of elements and bonds:* $\{E, B\}$*, where* $E = St \cup M$*. M applies for sensors* (De, Df) *and St, the set of structural elements, is made up of* $St = S \cup PSV \cup J_t$*. S represents Effort or Flow Source elements (Se, Sf). PSV applies for passive elements (resistance, R, capacitance, C, or inductance elements, I).* J_t *is the total set of junctions:* $J_t = J \cup T$*, where T are transducers (transformers TF, and gyrators GY), and J are the set of 0- and 1-junctions.*

Each one of these elements is connected through a set of bonds, $B \subset E \times E$*. Not every relation between elements* e_i*,* e_j *is allowed for each bond* $b_k \in B$*:*

- *for each* $(e_i, e_j) \in B$*,* $e_i \in J_t$ *or* $e_j \in J_t$ *or* $(e_i, e_j) \in Jt$

Exceptionally there could be combinations of one source and one passive element that would not respect that generic rule, but we do not consider those systems as significant for the fault diagnosis field.

Usually, BGs are extended by adding a number to each bond, in order to facilitate the enumeration of each effort and flow variable.

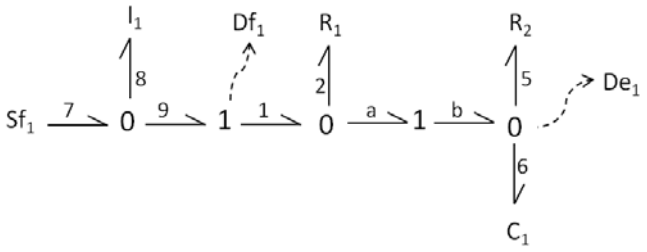


Figure 4.2: Bond Graph model of a system.

Figure 4.2 shows an example of a bond graph model. Each $b_k \in B$ represent a relation or equation among system (effort and flow) variables. The set of equations provided by a valid BG model can be expressed as state-space equations or as input-output models. The elements in $S \cup PSV$ provide the behavioural model by means of the set of its constituent equations. The elements of J_t provide the structural model of the system. The set M determines which variable in the system can be observed (the so called observational model in the FDI approach to MBD and that we have called the diagnosis observational model in Chapter 3).

A detailed description of BGs as modeling tools, and the rules to design valid BGs can be found in [65, 25, 24]. A formal description of the BG graphical model can be found in [84].

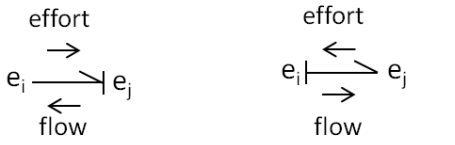
Causality expresses computational dependencies between effort and flow variables in a BG [106]. A BG with a valid global causal assignment and without sensors defines a just-determined set of equations, where S elements are the exogeneous variables or inputs [108].

A BG with a valid causal assignment, VCA, is known as a Causally Enhanced BG [82] or Causal BG [108]:

Definition 15. *(Causal BG) It is a* $BG = \{E, B\}$*, where each bond,* $b_i \in B \subset E \times E$ *is extended with a label causality* $= \{"effort", "flow"\}$*, that signals which variable (effort or flow) fix the causality in the bond:* $b_i = (e_i, e_j, causality)$*.*

The information provided by this definition can be graphically represented as vertical strokes in the bonds, determining which effort/flow is computed in a given bond [108]. For instance, given two elements, e_i, e_j, and a causality assignment, effort/flow:

- $(e_i, e_j, "effort")$ means that e_i imposes the effort and e_j imposes the flow, and the causal stroke is represented as in Figure 4.3 a).
- conversely, $(e_i, e_j, "flow")$ means that e_i imposes the flow and e_j imposes the effort, and the causal stroke is represented as in Figure 4.3 b).



(a) e_i imposes the effort in the bond.

(b) e_i imposes the flow in the bond.

Figure 4.3: Causality assignments marked with a vertical stroke.

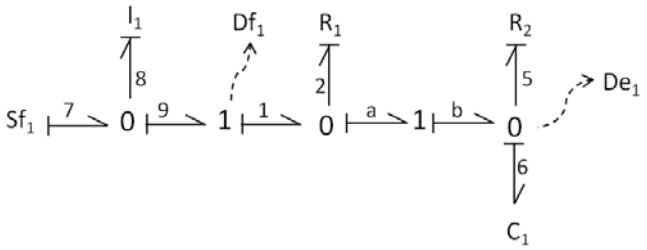


Figure 4.4: Bond Graph model of a system with a valid global causal assignment.

Figure 4.4 presents and example of a BG with a valid global causal assignment. From now on we will talk about BGs as Casual BGs. Only when it is necessary we will make a difference between Causal and Non-causal BGs.

An important property of a 1- (0-) junction is that only one bond imposes the flow (effort) at the junction. This bond is called *determining* bond. In the graphical representation, the determining bond of a 1- junction has the causal stroke out of the junction while the determining bond of a 0- junction has the causal stroke in the junction. The remaining bonds of the 1- (0-) junction have the causal stroke in (out of) the junction. For instance, in Figure 4.4 bond 9 is the determining bond of the 1- junction where flow is measured and bond 6 is the determining bond of the 0- junction where effort is measured.

There are systematic rules to assign a valid global causality to a given BG [108]. However, a valid BG may not have a VCA. The absence of a VCA in a BG is usually due to the fact that the system being modeled is not physically feasible, like for instance if we directly or indirectly connect in parallel two effort sources without a passive element. This absence of VCA might also be due to some modeling simplifications. If a valid causal assignment exists it can be provided by the SCAP algorithm [65] or other similar algorithms. Once causality is introduced, the equations must be ordered to provide a valid simulation model.

Adding sensors M to an extended valid BG introduces analytical redundancy in the system model, because we can at least estimate and observe each variable related to a sensor. As it is the usual procedure in Fault Detection and Diagnosis, sensors can be the source of discrepancies. This is the main idea behind building ARRs or DBGs for FDI using BGs [108], because using sensors as output and sources of discrepancies provides structural independent residuals. PCs also rely on these concepts although they are not defined on the BG framework. Extending the concept of PCs to BGs requires finding the set of subsystems in a BG

with minimal analytical redundancy which in turns requires introducing the following terms: Degenerated Junction, Sub Bond-Graph and Redundant Sub Bond-Graph.

Definition 16. *(Degenerated junction (J_d)) A degenerated 1- junction (equivalently 0-) is a one-port element that must be obtained from a valid 1- junction in a BG that is connected to a Df (equivalently De) sensor or a source, Sf (equivalently Se). Given a bond, b, and a measurement Df_1, the 1- degenerated junction introduces the following model:*

- *a) $f_b = Df_1$ (instead of the set of equalities $f_a = f_b$, $f_c = f_b$, that an original three-port 1- junction with determining bond b would have provided). If b is linked to a source, the expressions would be similar ($f_b = Sf_1$).*
- *b) there is no restriction for the conjugated variable, e_b (instead of $e_b = e_a + e_c$).*

It can be seen that degenerated junctions provide the value for exactly one variable, of exactly the same type (effort/flow) of the adjacent measurement or source. Also, creation of a degenerated junction requires dualizing the sensor.

Definition 17. *(Sub Bond Graph (sBG))A sub Bond Graph, sBG, derived from a bond graph, $BG = \{E, B\}$, is a partial connected subgraph: $sBG = \{E', B'\} \mid E' = St' \cup M'$ and $St' = S' \cup PSV' \cup J'$ with $S' \subseteq S$, $PSV' \subseteq PSV$, $M' \subseteq M$ and $B' \subseteq E' \times E' \subset B$. $J' = J'_o \cup J'_d$, $J'_o \subseteq J_t$, and J'_d is a set of zero or more degenerated junctions. Additionally if $j_d \in J'_d$ was derived from $j_o \in J$ then $j_o \notin J'_o$.*

To simplify notational burden, we will denote $sBG \subset BG$ when sBG is derived from a BG.

sBG is a partial subgraph from a BG that is made of some of the constituent elements of BG, and also a set of junctions J_o from the original BG. But there is also a potentially empty set of new junctions, J_d, that we call degenerated. These new types of 1- and 0- junctions will be used to split the BG in terms of sources or measurements and they are used to determine the value of a flow/effort variable. If j_d is a degenerated junction derived from an original junction $j_o \in J$, then by definition $j_o \notin J'_o$. Figure 4.5 shows a sBG obtained from the BG shown in 4.4. It has only one degenerated junction with a dualize effort sensor.

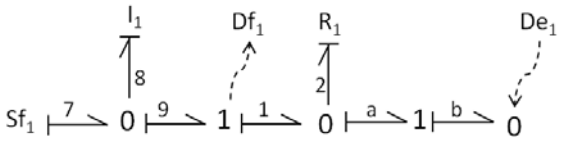


Figure 4.5: SubBG derived from the BG model in Figure 4.4

As it was mentioned before, each valid BG, without sensors, defines a just-determined set of equations. Once we introduce observations or sensors in M, there will be redundant information. This can be also the source of discrepancies or residuals that can be used for diagnosis purposes.

Depending on the number of measurements in the sBG, it will provide a just-determined or an over-determined, but never an underdetermined, set of equations, derived from its constituent and structural equations[1]. Because we look for analytical redundancy, we define:

Definition 18. *(Redundant Bond Graph (RBG)) A RBG is defined as a sBG whose underlying model has analytical redundancy.*

The subBG of Figure 4.5 is also a RBG, due to the presence of sensor Df_1.

Now we have the tools to define a bond graph subsystem that contains the minimal necessary elements to find a PC:

[1]This property of a sBG must be enforced by construction. Later on, we will provide the algorithms to build PCs based on sBGs that guarantee such requirement.

Definition 19. *(BG Possible Conflict (BG-PC)) Given a valid BG with a VCA, a BG-PC is a RBG, $\{E_{pc}, B_{pc}\}$, $E_{pc} = St_{pc} \cup M_{pc}$, such that $BG - PC \subset BG$ and BG-PC is minimal in the sense that $\nexists$ $RBG' \subset BG$ with $RBG' \subset$ BG-PC.*

The existence of a BG-PC $\subset$ BG requires that BG had analytical redundancy. Note that this definition implies that $\exists\ d' \in M_{pc}$ such that the VCA of BG-PC allows estimation of d' from dualized sensors and/or sources: d' is the discrepancy node. The discrepancy node is unique (otherwise BG-PC is not minimal).

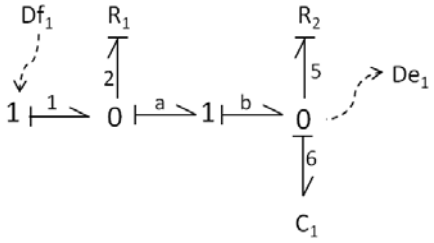


Figure 4.6: BG-PC derived from the BG model in Figure 4.4

Figure 4.6 shows an example of a BG-PC derived from the BG shown in Figure 4.4. The set of equations of a BG-PC is by definition overdetermined minimal in the sense that if we eliminate any of its equations we can not compute all its variables. However, to fully understand the calculation process of the discrepancy node, d', within the BG-PC, usually a causal graph is used. We will introduce a special causal graph which will allow us to analyze the effects of changing causality (or adding or removing bonds due to changes in switching junctions) from a purely structural and causal point of view.

Hence, we need additional concepts already introduced in [1] and that are defined from a causal graph named Temporal Causal Graph, or TGC, [76]. TCGs can be automatically obtained from a BG model with a VCA and are defined as:

Definition 20. *(Temporal Causal Graph (TCG)) A TCG derived from a bond graph model, BG, is a directed labelled graph $< V, L, B >$. V is a set of vertices or nodes, $V = E \cup F$. E and F are the sets of effort, respectively flow, variables in BG. L is the label set $\{=, 1, -1, p, p^{-1}, pdt, p^{-1}dt\ \}$ (where p are the names of the parameters, St U T, in BG). $B \subset V \times L \times V$ is a set of edges derived from the set of bonds in BG.B.*

The dt specifier indicates a temporal edge relation, which implies that a vertex affects the derivative of its successor vertex accross the temporal edge.

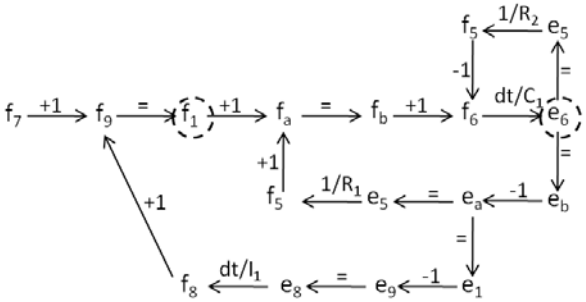


Figure 4.7: Temporal Causal graph derived from the BG in Figure 4.4 (dotted circles mark sensors).

Figure 4.7 presents the temporal causal graph derived from the BG model example presented in Figure 4.4. In the TCG in Figure 4.7 we have marked with a dotted circle the sensors of the BG model. Those variables that can be meassured or estimated are the source of analitycal redundancy. Temporal causal graphs are very useful for fault isolation and identification because they allow us to establish relationships between qualitative fault signatures for residuals in the presence of faults, and parameters involved in such

signature estimation [76, 84]. It is known that PCs can be directly computed from a TCG derived from a BG [23].

Now, we can introduce the definitions that will help analizing BG-PCs, already introduced in [20]:

Definition 21. *(Causal Path in a TCG) A causal path in a TCG model $< V, L, B >$, between two nodes, v_1, v_n, is $CP = \{V', B'\}$, $V' = \{v_1, \ldots, v_n\} \subset V$, and $B' \subset B$, such that the start node $v_1 \in M \cup S$ in BG, the end node $v_n \in M$ in BG, and the edges $\{(v_i, v_{i+1}) | i = 1, ..., n-1\} \subset B$ form a directed path from the start to the end node.*

We do not include the labels in the definition for the sake of readability because they are not relevant for our reasoning.

Definition 22. *(Minimal Causal Path in a TCG) A causal path in a TCG between two nodes, v_1 and v_n, is minimal if the following holds:*

- *Only the start, v_1, and the end node, v_n, can be a source or a measurement, i.e., $v_i \notin M \cup S$ for $2 \leq j \leq n-1$.*
- *A node can only appear once in a causal path, i.e., $v_i \neq v_j$, $\forall i, j$ such that $i \neq j$.*

Definition 23. *(Closed Causal Path) A causal path in a TCG, V_{cp}, between two nodes, v_i and v_n, is a closed causal path if the following conditions are fulfilled:*

1. *$v_i = v_n$.*
2. *$v_j \notin M \cup S$ for $2 \leq j \leq n-1$.*

Definition 24. *(Minimal Causal Subgraph (MCS)) A minimal causal subgraph in a TCG is a directed graph with a set of vertices $V_{mcs} = \{v_1, v_2, \ldots, v_n\} \subseteq V$, where*

1. *one node, $v_n \in M$, is the destination node,*
2. *one or more nodes, $\{v_{s1_1}, \ldots, v_{sk_1}\} \subset M \cup S$ are start nodes, and*
3. *it is made up of the set of minimal causal paths $\{v_{sh_1}, v_{sh_2}, \ldots, v_n\}$ and closed causal paths $\{v_m, \ldots, v_m\}$ found from the set of start nodes to the destination node, v_n.*

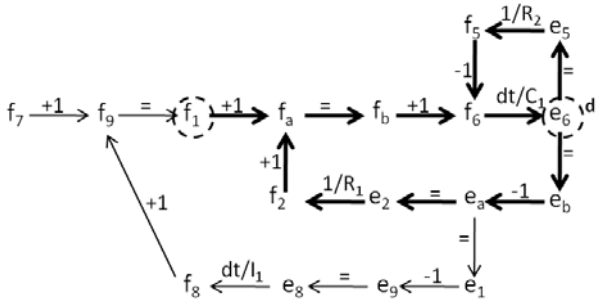


Figure 4.8: Minimal Causal Subgraph (MCS), in bold, derived from the TCG in Figure 4.7.

Figure 4.8 shows a MCS derived from the TCG of Figure 4.7. The destination node, which is measured, is marked with superscript **d**. This figure also shows that this MCS is the TCG of the BG-PC of Figure 4.6 (although in general the TCG of a BG-PC may include additional equations and variables).

A MCS describes how to estimate its destination variable from a minimal number of sensors and /or sources. The set of equations of a MCS are overdetermined minimal in the sense that none of its proper subsets is redundant.

We are going to extend the concept of MCS to describe minimal causal graphs that allow estimation of any destination node from a minimal set of variables, not necessarily limited to sensors and/or sources. Some

of these variables will be named *inputs*, generalizing the concept of input to any desired variable. These generalized inputs are not the real inputs of the system, i.e. they are not limited to sources. They are conceived as a device to break a path at the desired variable, which becomes an input:

Definition 25. *(Input Causal Path (ICP)). A causal path in a TCG model $< V, L, B >$, between two nodes, v_1, v_n, is $ICP = \{V', B'\}$, $V' = \{v_1, \ldots, v_n\} \subset V$, and $B' \subset B$, such that the input node $v_1 \notin M \cup S$ in BG, the end node $v_n \in M$ in BG, and the edges $\{(v_i, v_{i+1}) | i = 1, ..., n-1\} \subset B$ form a directed path from the start to the end node.*

Definition 26. *(Minimal Input Causal Path (MICP)). An input causal path in a TCG model $< V, L, B >$, between two nodes, v_1, v_n, is minimal if the following holds:*

- *Only the end node, v_n, can be a source or a measurement, i.e., $v_i \notin M \cup S$ for $1 \leq j \leq n-1$.*
- *A node can only appear once in a causal path, i.e., $v_i \neq v_j$, $\forall i, j$ such that $i \neq j$.*

Definition 27. *(Minimal Input Causal Subgraph (MICS)) A minimal input causal subgraph in a TCG is a directed graph with a set of vertices $V_{mics} = \{v_1, v_2, \ldots, v_n\} \subseteq V$, where*

1. *one node, $v_n \in M$, is the destination node,*
2. *a potentially empty set $START = \{v_{s1_1}, \ldots, v_{sk_1}\} \subset M \cup S$ are start nodes,*
3. *a potentially empty set $INPUT = \{v_{i1_1}, \ldots, v_{il_1}\} \mid INPUT \cap (M \cup S) = \emptyset$ are input nodes,*
4. *$START \cup INPUT \neq \emptyset$ and*
5. *it is made up of the set of minimal causal paths, $\{v_{sh_1}, \ldots, v_n\}$, minimal input causal paths, $\{v_{ij_1}, \ldots, v_n\}$ and closed causal paths $\{v_m, \ldots, v_m\}$ found from the set of start and input nodes to the destination node, v_n, with the restrictions:*
   - $\nexists \{v_{sh_1}, \ldots, v_n\}\ MCP \mid \exists\ v_i \in INPUT\ \&\ v_i \in \{v_{sh_1}, \ldots, v_n\}$
   - $\nexists \{v_{ij_1}, \ldots, v_n\}\ MICP \mid \exists\ v_i \in START\ \&\ v_i \in \{v_{ij_1}, \ldots, v_n\}$

By definition, if we add to the BG model of the system a sensor to every input node of a MICS, it becomes a MCS.

Similarly, the former input causal paths and subgraphs can be extended to the case when the destination node is not measured, transforming them in potential causal paths and subgraphs. For the sake of brevity, we only provide the definition of Possible Minimal Causal Subgraph:

Definition 28. *(Possible Minimal Causal Subgraph (PMCS)) A possible minimal causal subgraph in a TCG is a directed graph with a set of vertices $V_{pmcs} = \{v_1, v_2, \ldots, v_n\} \subseteq V$, where*

1. *one node, $v_n \notin M$, is the destination node,*
2. *a potentially empty set $START = \{v_{s1_1}, \ldots, v_{sk_1}\} \subset M \cup S$ are start nodes,*
3. *a potentially empty set $INPUT = \{v_{i1_1}, \ldots, v_{il_1}\} \mid INPUT \cap (M \cup S) = \emptyset$ are input nodes,*
4. *$START \cup INPUT \neq \emptyset$ and*
5. *it is made up of the set of minimal causal paths, $\{v_{sh_1}, \ldots, v_n\}$, minimal input causal paths $\{v_{ij_1}, \ldots, v_n\}$ and closed causal paths $\{v_m, \ldots, v_m\}$ found from the set of start and input nodes to the destination node, v_n, with the restrictions:*
   - $\nexists \{v_{sh_1}, \ldots, v_n\}\ MCP \mid \exists\ v_i \in INPUT\ \&\ v_i \in \{v_{sh_1}, \ldots, v_n\}$
   - $\nexists \{v_{ij_1}, \ldots, v_n\}\ MICP \mid \exists\ v_i \in START\ \&\ v_i \in \{v_{ij_1}, \ldots, v_n\}$

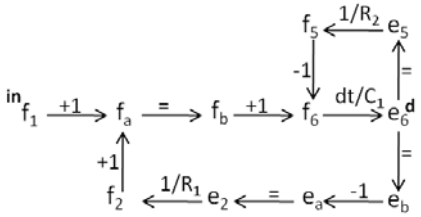


Figure 4.9: Possible Minimal Causal Subgraph.

By definition, if we add to the BG model of the system a sensor to the destination node of a PMCS, it becomes a MICS which can be further transformed into a MCS if we add a sensor to every input node of the MICS.

Figure 4.9 shows a Possible Minimal Causal Subgraph (PMCS) obtained from the MCS of Figure 4.8 eliminating the sensor at f_1, the sensor at e_6 and with f_1 as input node. We will mark the input variables with the superscript **in** and the destination variables with the superscript **d**.

Due to the definitions of MCS and $PMCS$, the underlying set of equations of a $PMCS$ has the following properties:

- If $INPUT \neq \emptyset$, the equations of $PMCS$ are underdetermined.
- If $INPUT = \emptyset$, the equations of $PMCS$ are just determined.
  - If we add sensors to every input node we transform them in start nodes and $INPUT = \emptyset$.
  - Similarly if we know the value, $value_i$ of every $u_i \in INPUT$ and we add the equations $u_i = value_i$ $1 \leq i \leq l$, the resulting set of equations is just determined minimal.
- If the inputs are known and we add a sensor to the destination node, the set of equations is overdetermined minimal and the $PMCS$ becomes a MCS.

A special case may occur if we add a source to the destination node of a PMCS. It represents a non valid causal assignment and only makes sense if theres is a diferent VCA such that $\exists\ PMCS'$ with former destination node a start node and $\exists\ v_i$, a former input node, and v_i is the new destination node.

We will introduce some notational conventions that will be used later on:

- $PMCS\_v_n$: $PMCS$ with destination node v_n.
- $PMCS^{\emptyset}\_v_n$: $PMCS$ with destination node v_n and $INPUT = \emptyset$.

Now, we extend the notion of parametric and non-parametric ARRs to PCs and MCS:

Definition 29. *(Non-parametric Minimal Causal Subgraph) A non parametric MCS is a MCS which labels have no passive elements. There are three options for its set of variables:*

- *All flow variables, including, at least, a flow source and a flow sensor.*
- *All effort variables, including, at least, an effort source and an effort sensor.*
- *Sequences of effort variables and sequences of flow variables if there are gyrators. To simplify this dissertation, we will suppose that there are no transducers in the system. But the discussion is essentially the same with transducer elements. Hence, we will assume that non-parametric MCS does not have both effort and flow variables.*

The concept of non-parametric $PMCS$ can be defined in similar terms.

From the previous definitions we can easily derive the following characterization:

Definition 30. *(Non parametric BG-PC) A BG-PC with discrepancy node d is non parametric if and only if* $\exists$ *a non parametric* $PMCS^{\emptyset}\_d$

Consequently if a BG-PC is non-parametric the following assertions hold:

- It contains at least one sensor.
- It has to contain, at least, a source or an additional sensor.

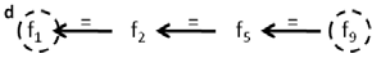


Figure 4.10: Non-parametric Minimal Causal Subgraph.

Figure 4.10 shows a non-parametric Minimal Causal Subgraph.

We will assume we are always referring to parametric PMCS. If it is non-parametric, we will make it explicit.

As it was previously mentioned, HBGs are an extension of BGs to model hybrid behaviour. In order to model discrete behaviour we have special 1-j or 0-j junctions that are known as switching junctions, sw-j. First, we assume that the set of hybrid states can be characterized by the value of the sw-j in the system.

To diagnose hybrid systems, we should be able to find and use the set of PCs valid for each operation mode (or discrete state). To do so, we would need to analyse the behaviour of the system in each operation mode: i.e. for each valid configuration of the set of sw-j. The number of configurations can be very high, hence it is not usually feasible to compute the whole set of PCs for every possible mode. To avoid this drawback, we introduce the notion of HBG-PC. As it will be shown later, HBG-PCs allows computing all PCs when every sw-j is ON, reducing the need of model reconfiguration to the HBG-PCs that include the switching junctions that changes their state.

Definition 31. *(HBG-PC:) It is a BG-PC, where some elements* $J_{sw} \subset J'$ *are switching junctions, and has one global valid causal assignment for at least one hybrid state.*

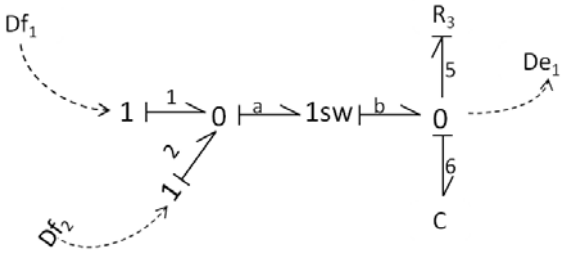


Figure 4.11: HBG-PC.

Figure 4.11 shows an HBG-PC, that has two sensors, Df_1 and Df_2, as inputs, one 1- switching junction, $1sw$, and the discrepancy node is the effort sensor, De_1. The reader can see that the left-hand side of the 1- switching junction has only non-parametric information, while the right-hand side has parametric components.

In this dissertation we impose that the system has a global valid causal assignment when every sw-j is ON in order to compute HBG-PCs. In that case, we can build the associated TCG from the BG [76], then look for MCSs related to measurements in the TCG [23, 19, 1]. From a MCS we can obtain a HBG-PC including each junction, source, sensor and passive element that contributes with a link to the MCS and their incident bond. In this process, all dualized sensors incident to a junction provide a degenerated junction, instead of the original one.

Every time a discrete change happens in the system, we need to recompute the new causal assignment. This process can be sped up using block diagrams to build the computational model of the BG, and running H-SCAP algorithms [106]. Thus, HBG-PCs can be used to efficiently track hybrid systems and also to perform online fault detection, isolation and identification.

> The main claim of this work is that such reconfiguration will be limited to the HBG-PCs found when every switching junction is set to ON. As a consequence, we only will need to track a limited number of subsystems, and each subsystem will have a limited number of switching junctions, thus making the problem tractable.

Next section provides the theoretical concepts and properties related to HBGs and HBG-PCs that support the main claim: the set of HBG-PCs in the system can be derived from the set of HBG-PCs present in the system when every switching junction is set to ON. Proposition 2 will show the effects of changes in switching junctions from ON to OFF, and viceversa, for both parametric and non parametric BG-PCs.

Finally, Section 4.6 will explain how these concepts will be used in our framework for hybrid systems diagnosis for both discrete and parametric faults, and illustrate the approach with a four-tank system.

4.3.2 Main properties of PMCSs in HBGs

Showing that the reconfiguration due to changes of a discrete mode is limited to the HBG-PCs where the switches that change their state are allocated requires introducing several properties of PMCS in HBG-PCs, using the variables of the switching junction bonds as input and/or destination. But first, we have to introduce several definitions:

Definition 32. *(VCA_v_n) It is a valid causal assignment of the HBG-PC such that $\exists\ PMCS\_v_n \subseteq TCG$.*

Definition 33. *($Eq(PMCS\_v_n)$) It is the set of equations abstracted by the arcs of $PMCS\_v_n$, without their causality mark.*

For instance:

- $e_1 = e_2 + e_3$ is the same as $e_3 = e_1 - e_2$.
- $e = f \cdot R$ is the same as $f = e/R$.
- $e = 1/C \cdot \int f$ is the same as $de/dt = 1/C \cdot f$.
- $f = 1/I \cdot \int e$ is the same as $de/dt = f \cdot I$[2].

Definition 34. *($Var(PMCS\_v_n) \subseteq V$) It defines the set of variables of $PMCS\_v_n$.*

Definition 35. *($Start(PMCS\_v_n)$) It defines the set of start variables of $PMCS\_v_n$.*

Definition 36. *($Input(PMCS\_v_{i_n})$) It defines the set of input variables of $PMCS\_v_n$.*

Now, we can introduce the *Equivalence property*, that is the fundamental property of PMCS in linear systems:

Property (Equivalence property)**.** *Let BG be a valid BG model with a VCA, and TCG its derived temporal causal graph such that $\exists\ PMCS_y$ with $y \in V$. Let $INPUT = Input(PMCS\_y)$ and $START = Start(PMCS\_y)$. And let $TCG\_y \subseteq TCG$ the temporal causal subgraph obtained from TCG eliminating all the variables not included in $var(PMCS\_y)$ and all the edges not included in $eq(PMCS\_y)$*[3]*. Then $\forall\ u \in INPUT \mid \exists\ VCA\_u$, $\exists\ PMCS\_u$ and $\exists\ TCG\_u$ with:*

[2] Dynamic equations inversion only makes sense when both causalities (integral and derivative) are valid.

[3]
- There is a one-one relation between the edges in TCG connecting e_i (f_i) and f_i (e_i), and the equations of passive elements in TCG_y. It remains the same, changing the direction and inverting the relation, for different causal assignments.
- The causality in the flows or efforts balances (1- or 0- junctions) determines the connectivity of the variables involved.

- *$PMCS\_u$:*
  1. $Var(PMCS\_u) = Var(PMCS\_y)$
  2. $Eq(PMCS\_u) = Eq(PMCS\_y)$
  3. $Start(PMCS\_u) = START$
  4. $Input(PMCS\_u) = INPUT \setminus \{u\} \cup \{y\}$
- *and TCG_u:*
  1. $Var(TCG\_u) = Var(TCG\_y)$
  2. *Links between e_i/f_i in TCG_u: links between e_i/f_i in TCG_y inverting those required by the new VCA_u.*
  3. *Flow (effort) balance links (due to 1- and 0-j juntions) according to new VCA_u .*

Similarly, $\forall\ x \in START \mid \exists\ VCA\_x$, in the temporal causal graph TCG' of the system obtained removing the sensor or source of x, $\exists\ PMCS\_x$ and $\exists\ TCG\_x$ with:

- *$PMCS\_x$:*
  1. $Var(PMCS\_x) = Var(PMCS\_y)$
  2. $Eq(PMCS\_x) = Eq(PMCS\_y)$
  3. $Start(PMCS\_x) = START \setminus \{x\}$
  4. $Input(PMCS\_x) = INPUT \cup \{y\}$
- *and TCG_x:*
  1. $Var(TCG\_x) = Var(TCG\_y)$
  2. *Links between e_i/f_i in TCG_x: links between e_i/f_i in TCG_y inverting those required by the new VCA_x.*
  3. *Flow (effort) balance links (due to 1- and 0-j untions) according to new VCA_x .*

■

The last stage of TCG_u and TCG_x reconstruction changes the topology of the TCG.

Proof 1. *(Equivalence Property). We will demonstrate the equivalence property for the case where the new destination node was a former input, assuming $INPUT \neq \emptyset$. The proof for the case where the new destination node was a former start node is simpler because start nodes are meassured or sources.*

The BG model have a VCA_y and in its derived $TCG\ \exists\ PMCS\_y$ with $INPUT = input(PMCS_y) \neq \emptyset$. By definition, $Eq(PMCS\_y)$ are underdetermined, but adding a sensor to each $u' \in INPUT$, we obtain a set of equations Eq_y_I that is just determined. Moreover, this set of equations is:

- *Linear,*
- *determined (i.e.: at most one solution),*
- *consistent (it has a solution, that is true if it comes from a physical system),*
- *allows computing all variables of $Var(PMCS\_y)$ from $START$ and the new added sensors by substitution method (If there is any algebraic loop, we can use an equation solver or super-component, this proof is independent of the method used to solve the system equations)*

If $\exists\ VCA\_u$ then, imposing in Eq_y_I the causality VCA_u, removing the sensor of u and adding a new sensor to y, we can compute u from $START$ and the current added sensors. If we eliminate all the added sensors, we obtain $PMCS\_u$ with $Input(PMCS\_u) = INPUT \setminus \{u\} \cup \{y\}$:

- *Just change the causality of those equations where VCA_y and VCA_u differs (can always be done in a linear system when both assignments are valid). For dynamic equations in capacitances and inductances would mean changing from integral to derivative causality. This could develop into numerical problems, but it should be reminded that we are analyzing the structure of the set of equations in the TCG.*
- *We are using the same variables and equations, just a different valid causal assignment.*

Additionaly, the TCG_u can be easily obtained from $Eq(PMCS\_u)$ and VCA_u.

□

The meaning of the equivalence property of PMCS is illustrated in Figure 4.12 where the $PMCS\_f_a$ in c) with $\{e_a\} = Input(PMCS\_f_a)$ can be transformed into $PMCS\_e_a$ in f) with $\{f_a\} = Input(PMCS\_e_a)$.

Proposition 1. *The equivalence property induce an equivalence relation.*
Given a $PMCS\_y$ with a VCA_y. Let $V_1 = \{x | x \in Start(PMCS\_y) \cup \{y\}, \exists VCA\_x\}$
The equivalence property induce an equivalence relation on the set of $\{PMCS\_x, x \in V_1\}$

It can be easily shown that the previous property is symmetrical, reflexive and transitive.

PMCS are a useful tool to analyze how changes in causality influence computation in a system. In particular, there are several important properties of PMCS with input or destination node at a switching junction. In order to analize these kind of systems we have introduced a generic decomposable system, presented in Definition 37.

Definition 37. *(Generic decomposable HBG, ($gdHBG$)) Let $gdHBG$ be a Hybrid Bond Graph of a linear system with valid causal assignment describing a just determined system and including a $1sw$ that is not in a closed casual path (cycle when set to ON). $gdHBG$ admits the decomposition $HBG{=}BG_1 \cup 1sw \cup BG_2$ with:*

- *$BG_1 \neq \emptyset$, $BG_2 \neq \emptyset$ and $BG_1 \cap BG_2 = \emptyset$,*
- *1sw: It represents 1-j switching junction,*
- *Graphically:* HBG: $BG_1 \xrightarrow{a}$ 1sw $\xrightarrow{b} BG_2$

And let:

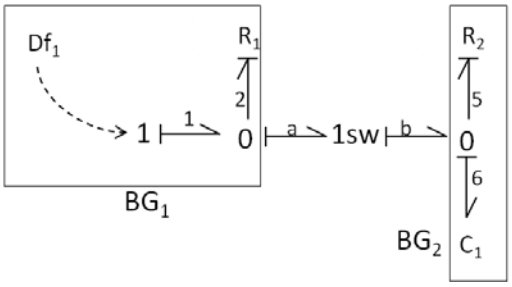
- *$PS_1 \subset gdHBG$ be the partial subsystem defined by:* PS_1: $BG_1 \xrightarrow{a}$
- *$PS_2 \subset gdHBG$ be the partial subsystem defined by:* PS_2: $\xrightarrow{b} BG_2$
- *BG'_1:* BG'_1: $BG_1 \xleftarrow{a}$ Sf:0
- *BG'_2:* BG'_2: Sf:0 $\xrightarrow{b} BG_2$

Generic decomposable systems satisfy properties 1 to 4 and Corollaries 1 and 2, that we are going to introduce in the following paragraphs.

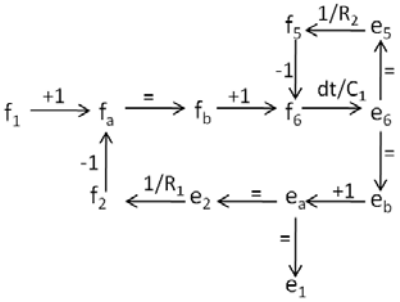
Property 1 (Equivalence property in generic decomposable systems)**.** *If $\exists$ parametric $PMCS\_f_a$ in $PS_1 \subset gdHBG$ with $\{e_a\} = Input(PMCS\_f_a)$ and $\exists$ VCA_e_a in PS_1 then $\exists$ parametric $PMCS\_e_a$ in PS_1 with $\{f_a\} = Input(PMCS\_e_a)$.*

■

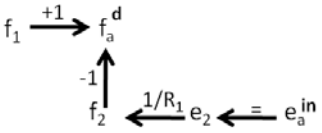
This is a direct application of the Equivalence property of PMCS to the left side subsystem of a gdHBG. A similar result can be stated if $\exists$ parametric $PMCS\_e_a$ in PS_1.



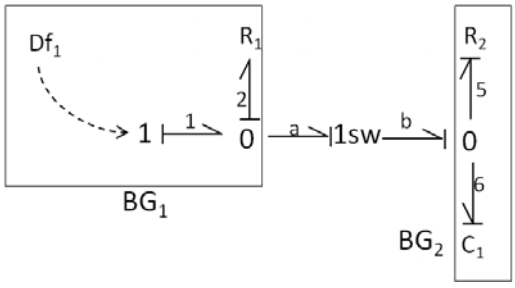
(a) HBG causal assignment 1



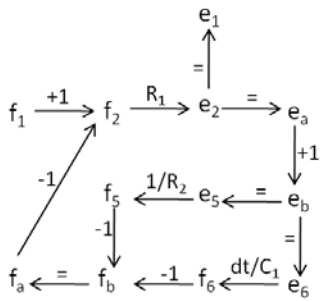
(b) TCG causal assignment 1.



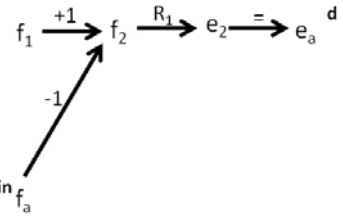
(c) Parametric $PMCS\_f_a$ with $\{e_a\} = Input(PMCS\_f_a)$ and $\{f_1\} = Start(PMCS\_f_a)$.



(d) HBG causal assignment 2



(e) TCG causal assignment 2.



(f) Parametric $PMCS\_e_a$ with $\{f_a\} = Input(PMCS\_e_a)$ and $\{f_1\} = Start(PMCS\_e_a)$.

Figure 4.12: Property 1. HBG in figure a) has a causal assignment and its TCG is shown in b) with the corresponding $PMCS\_f_a$ shown in c). Changing the causal assignment, as in d), there is a TCG shown in e) and the corresponding $PMCS\_e_a$ presented in f).

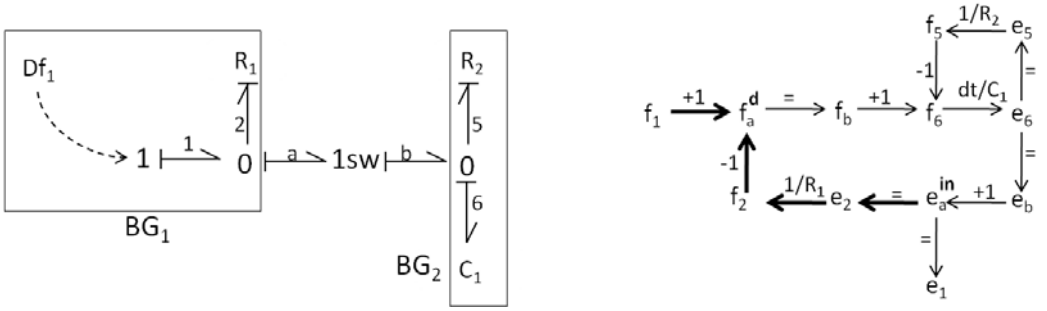
Property 2 (In a generic decomposable system we can not have simultaneously a parametric and non parametric PMCS to a variable of the bond of the 1-sw both at the same side of the switching)**.** *If* $\exists$ *parametric* $PMCS\_f_a$ *in* $PS_1 \subset gdHBG$ *with* $\{e_a\} = Input(PMCS\_f_a)$ *then* $\nexists$ *non parametric* $PMCS^{\emptyset}\_f_a$ *in* PS_1.

If $\exists$ *non parametric* $PMCS^{\emptyset}\_f_a$ *in* $PS_1 \subset gdHBG$ *then* $\nexists$ *parametric* $PMCS\_f_a$ *in* PS_1 *with* $\{e_a\} = Input(PMCS\_f_a)$*.(The former argument also applies to this case).*

■

This is because if $\exists$ parametric $PMCS\_f_a$ with $\{e_a\} = Input(PMCS\_f_a)$ and $\exists$ non parametric $PMCS^{\emptyset}\_f_a$, both in PS_1, then we have two independent causal subgraphs in PS_1 to compute f_a and PS_1 would be overdetermined. Hence gdHBG would not be just determined.

A similar result stands for parametric and non parametric $PMCS\_e_a$. Figure 4.13 shows that there is only one causal graph to a destination variable in any side of the switching junction of a just determined system.



(a) HBG (b) TCG parametric $PMCS\_f_a$

Figure 4.13: Property 2. HBG in figure a) has a causal assignment and there is a parametric $PMCS\_f_a$ shown in b). The presence of an additional non parametric $PMCS^{\emptyset}\_f_a$ in PS_1 would make this part of the system overdetermined.

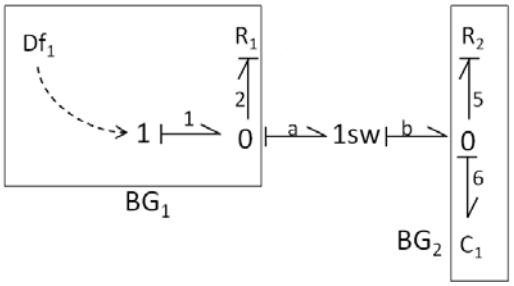
Property 3 (In a generic decomposable system we can not have simultaneously a parametric PMCS to a variable of the bond of the 1-sw and a non parametric PMCS to its conjugated variable both at the same side of the switching)**.** *If* $\exists$ *parametric* $PMCS\_f_a$ *in* $PS_1 \subset gdHBG$ *with* $\{e_a\} = Input(PMCS\_f_a)$ *then* $\nexists$ *non parametric* $PMCS^{\emptyset}\_e_a$ *in* PS_1*.*

■

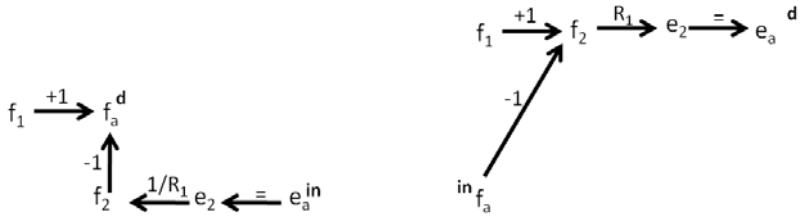
This is because if $\exists$ parametric $PMCS\_f_a$ and $\exists$ non parametric $PMCS^{\emptyset}\_e_a$ both in PS_1, then PS_1 is able to compute e_a and f_a. Whatever the causality at the junction, $e_b = e_a$ and $f_b = f_a$, without using any equation from PS_2. But PS_2 must be able to compute e_b, or f_b or to impose a relationship between e_b and f_b. Then, $gdHBG$ would be overdetermined.

A similar result can be shown if $\exists$ parametric $PMCS\_e_a$ in PS_1.

Figure 4.14 b) shows a parametric $PMCS\_f_a$ in PS_1. The presence of an additional non parametric $PMCS^{\emptyset}\_e_a$ in PS_1 would make the system overdetermined.



(a) HBG.



(b) TCG. Parametric $PMCS\_f_a$ in PS_1. (c) TCG. Parametric $PMCS\_e_a$ in PS_1.

Figure 4.14: Property 3. HBG in figure a) has a causal assignment and there is a $PMCS\_f_a$ shown in b). It can be transformed to obtain the parametric $PMCS\_e_a$ in c). The presence of an additional non parametric $PMCS^{\emptyset}\_e_a$ in PS_1 would make the system overdetermined.

Corollary 1. *The existence of a parametric $PMCS$ with destination node a variable of the switching junction at one side of the switching in a gdHBG, prevents the existence of a non parametric $PMCS^{\emptyset}$ to any variable of the bond at the same side of the switching.*

This is a consequence of properties 1, 2 and 3. If $\exists$ parametric $PMCS\_f_a$ $[PMCS\_e_a]$ with $\{e_a\} = Input(PMCS\_f_a)$ $[\{f_a\} = Input(PMCS\_e_a)]$, then $\nexists$ non parametric $PMCS^{\emptyset}\_f_a$ nor $PMCS^{\emptyset}\_e_a$.

Corollary 2. *The existence of a non parametric $PMCS$ with destination node a variable of the switching junction at one side of the switching in a gdHBG, prevents the existence of a parametric $PMCS$ to any variable of the bond at the same side of the switching.*

This is a consequence of Corollary 1, because if there is a parametric $PMCS$ to any variable of the bond of the switching junction at one side of the switching, then it is not possible to have a non parametric $PMCS$ at the same side of the switching.

Property 4 (Setting the switching junction to OFF in a parametric gdHBG, generates a just-determined subsystem at the parametric side of the junction)**.** *If $\exists$ parametric $PMCS\_f_a$ with $\{e_a\} = Input(PMCS\_f_a)$ in $PS_1 \subset gdHBG$, then BG'_1 is just determined.*

■

Figure 4.15 illustrates this property and facilitates following its proof.

Proof 2. *(Property 4).*

Assume that a is the determining bond at the switching junction (see Figure 4.15 a)).

Let $Eq(PS_1) = Eq(PMCS\_f_a) \cup Remainder$ with $Remainder = Eq(PS_1) \setminus Eq(PMCS\_f_a)$ (see Figure 4.15 a), b) and c)).

Then $Eq(BG'_1) = Eq(PMCS\_f_a) \cup Remainder \cup \{f_a = 0\}$ because it is obtained from PS_1 changing causality and adding the zero flow source (see Figure 4.15 d) and e)).

By property 1, given that BG'_1 has a VCA, $\exists$ $PMCS\_e_a$ with $\{f_a\} = Input(PMCS\_e_a)$ in BG'_1 with $Eq(PMCS\_e_a) = Eq(PMCS\_f_a)$ (see Figure 4.15 f)). Consequently we also have $Eq(BG'_1) = Eq(PMCS\_e_a) \cup Remainder \cup \{f_a = 0\}$.

Define $Border(Remainder) = \{v | v \in Var(PMCS\_f_a)$ and $v \in Var(eq_i)$ and $eq_i \in Remainder\}$.

Given that gdHBG is just determined, $Eq(PS_1)$ is just determined given e_a (e_a in gdHBG is computed by PS_2 for the causal assignment considered). Because $Eq(PMCS\_f_a)$ is just determined given e_a, Remainder has to be just determined given every $v \in Border(Remainder)$.

By definition, $Eq(PMCS\_e_a) \cup \{f_a = 0\}$ is just determined: we can compute every $v \in Border(PMCS\_f_a)$ in BG'_1 − actually every $v \in Var(PMCS\_f_a) = Var(PMCS\_e_a)$ −.

Then, $Eq(BG'_1)$ is just determined, because its set of equations can be split in two disjoint just determined sets of equations (that share the variables of $Border(Remainder)$, which are computed by $Eq(PMCS\_e_a)$ in BG'_1).

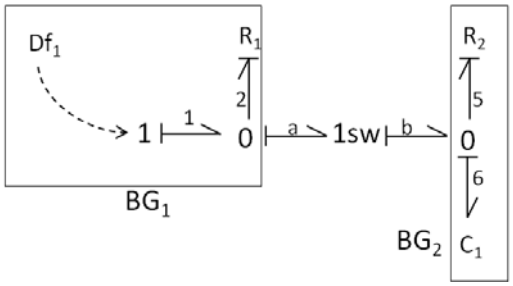
A similar result is obtained if $\exists$ parametric $PMCS\_e_a$ in PS_1 with alternative causality at the switching.

□

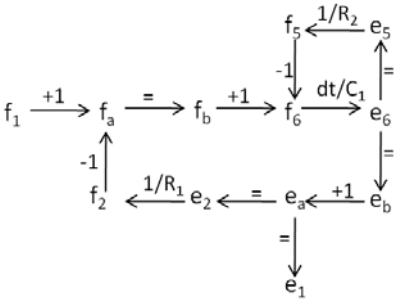
Now, we are ready to examine the effect of turning to OFF a switching junction on the subsystems it connects. First we will define a decomposable HBG-PC:

Definition 38. *(Generic decomposable HBG-PC, (gdHBG-PC)) Let gdHBG-PC be a HBG-PC of a linear system with valid causal assignment describing a minimal overdetermined system and including a $1sw$ that is not in a closed casual path (cycle when set to ON). gdHBG-PC admits the decomposition gdHBG-PC=$BG_R \cup 1sw \cup BG_{NR}$ with:*

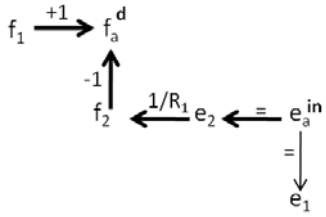
- *$BG_R \neq \emptyset$, $BG_{NR} \neq \emptyset$ and $BG_R \cap BG_{NR} = \emptyset$,*
- *BG_R includes the discrepancy node.*
- *BG_{NR} does not include the discrepancy node.*



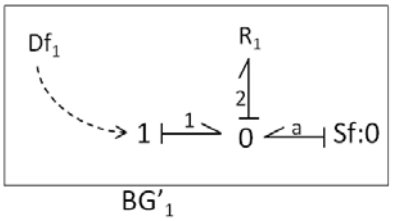
(a) HBG switching junction ON.



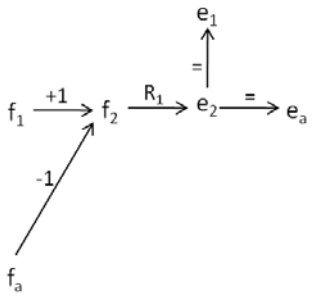
(b) TCG switching junction ON.



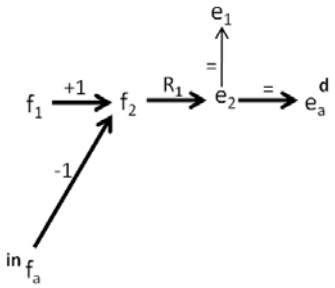
(c) PMCS switching junction ON. Parametric $PMCS\_f_a$.



(d) HBG switching junction OFF.



(e) TCG switching junction OFF.



(f) PMCS switching junction OFF. Parametric $PMCS\_e_a$.

Figure 4.15: Property 4. HBG in figure a) has the switching junction ON, its corresponding TCG is in b). The corresponding $PMCS\_f_a$ is shown in c). The HBG with the switching junction OFF is presented in d), its TCG is shown in e) and the corresponding $PMCS\_e_a$ is presented in f).

- *1sw: It represents 1-j switching junction,*

And let:

- $PS_R \subset$ *gdHBG-PC be the partial subsystem defined by:* PS_R: $BG_R \xrightarrow{a}$
- $PS_{NR} \subset$ *gdHBG-PC be the partial subsystem defined by:* PS_{NR}: $\xrightarrow{b} BG_{NR}$
- BG'_R: $BG_R \xleftarrow{a} Sf{:}0$
- BG'_{NR}: $Sf{:}0 \xrightarrow{b} BG_{NR}$

If we remove the sensor of the discrepancy node in a gdHBG-PC, we obtain a special gdHBG that is a just determined system:

Definition 39. *(Generic decomposable non redundant HBG-PC, (gdnrHBG-PC)) Let gdnrHBG-PC be a HBG-PC obtained by removing the discrepancy node $\{d\}$ from a gdHBG-PC. Then, gdnrHBG-PC admits the decomposition gdnrHBG-PC=$(BG_R \setminus \{d\}) \cup 1sw \cup BG_{NR}$ with:*

- $BG_R \setminus \{d\} \neq \emptyset$, $BG_{NR} \neq \emptyset$ *and* $(BG_R \setminus \{d\}) \cap BG_{NR} = \emptyset$,
- $BG_R \setminus \{d\}$ *now, does not include the discrepancy node.*
- BG_{NR} *does not include the discrepancy node.*
- *1sw: It represents 1-j switching junction.*

Assume that a is the determining bond at the switching junction of gdnrHBG-PC: $BG_R\setminus\{d\} \overset{a}{\longmapsto} 1_{SW} \overset{b}{\longmapsto} BG_{NR}$ *And let:*

- $PS_R \setminus \{d\} \subset gdnrHBG - PC$ *be the partial subsystem defined by:* $PS_R\setminus\{d\}$: $BG_R\setminus\{d\} \overset{a}{\longmapsto}$
- $PS_{NR} \subset gdnrHBG - PC$ *be the partial subsystem defined by:* PS_{NR}: $\overset{b}{\longmapsto} BG_{NR}$
- $BG'_R \setminus \{d\}$: $BG_R\setminus\{d\} \overset{a}{\longmapsfrom}$ Sf:0
- BG'_{NR}: Sf:0 $\overset{b}{\longmapsto} BG_{NR}$

A gdnrHBG-PC is also a gdHBG and it satisfies Properties 1 to 4. It also satisfies the additional Property 5:

Property 5. *Let HBG be a gdnrHBG-PC with a the determining bond at the switching junction. Then:*

I The system which causal assignment does not change, BG'_{NR} in this setting, is just determined.

II If the system which causality changes, $BG'_R \setminus \{d\}$ in this setting, has a valid causal assignment, then $\nexists$ non parametric $PMCS^{\emptyset}\_e_a$ in $PS_R \setminus \{d\}$ and there are two possibilities:

a If $\exists$ parametric $PMCS\_f_a$ in $PS_R \setminus \{d\}$ with $Input(PMCS\_f_a) = \{e_a\}$ then $BG'_R \setminus \{d\}$ is just determined.

b If $\exists$ non parametric $PMCS^{\emptyset}\_f_a$ in $PS_R \setminus \{d\}$, then $BG'_R \setminus \{d\}$ contains a subsystem of flow equations overdetermined, minimal. And we cannot compute e_a in $BG'_R \setminus \{d\}$.

■

Figure 4.16 and 4.17 show the two possibilities presented in property 5 for the subsystem which causality changes when it has a VCA: if it is parametric then it is just determined (Figure 4.16) and when it is non parametric, there is an overdetermined subsystem of flows and the efforts cannot be calculated (Figure 4.17).

Proof 3. *(Property 5).*

*We will prove first part **I** of property 5:*

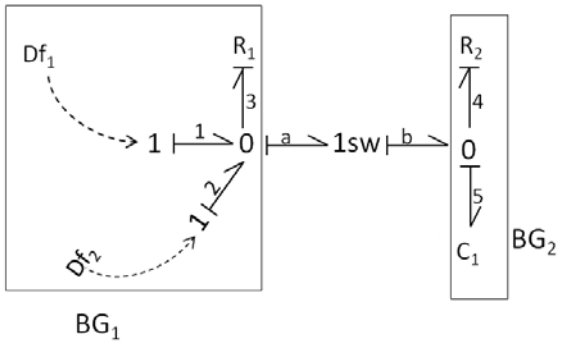
I. The system which causal assignment does not change, BG'_{NR} in this setting, is just determined.

In HBG we have that the flow f_b is fixed by $PS_R \setminus \{d\}$, that is $f_b = f_a$ and given f_b, PS_{NR} is just determined (otherwise the complete system is not).

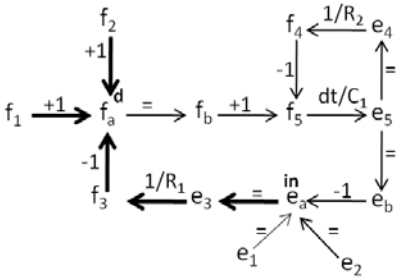
Now, in BG'_{NR}, we have $f_b = 0$ and the same causal assignment. Hence, BG'_{NR} is just determined.

*Now we proceed with part **II** of property 5:*

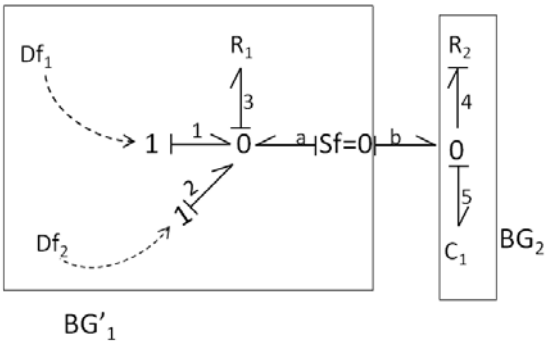
II. If the system which causality changes, $BG'_R \setminus \{d\}$ in this setting, has a valid causal assignment, then $\nexists$ non parametric $PMCS^{\emptyset}\_e_a$ in $PS_R \setminus \{d\}$.



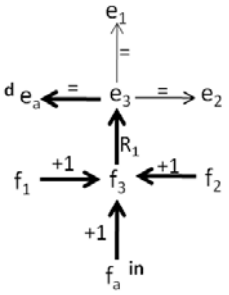
(a) HBG switch ON



(b) TCG switch ON. Parametric $PMCS\_f_a$ in bold.



(c) HBG switch OFF



(d) TCG switch off. Parametric $PMCS\_e_a$ in bold.

Figure 4.16: Property 5. HBG in figure a) has a valid causal assignment. The TCG for that causality has a parametric $PMCS\_f_a$ shown in b). The causality changes when the switch is OFF for $BG'_R \setminus \{d\}$, as in c). There is a parametric $PMCS\_e_a$ shown in d).

Because of the causal assignment of the HBG, $\exists$ $PMCS\_e_b$ in PS_{NR} (parametric or non parametric) and $e_a = e_b$.

If there is also a non parametric $PMCS^{\emptyset}\_e_a$ in $PS_R \setminus \{d\}$, then there are two independent ways to compute e_a and HBG would include an overdetermined subset of equations. Then HBG would not be just determined.

*We continue now with part **IIa** of property 5:*

II.a. If** $\exists$ **parametric** $PMCS\_f_a$ **in** $PS_R \setminus \{d\}$ **with** $Input(PMCS\_f_a) = \{e_a\}$ **then** $BG'_R \setminus \{d\}$ **is just determined.

By property 1, $\exists$ parametric $PMCS\_e_a$ with $Input(PMCS\_e_a) = \{f_a\}$ in $BG_R \setminus \{d\}$. By property 4, the equations of $BG'_R \setminus \{d\}$ are just determined.

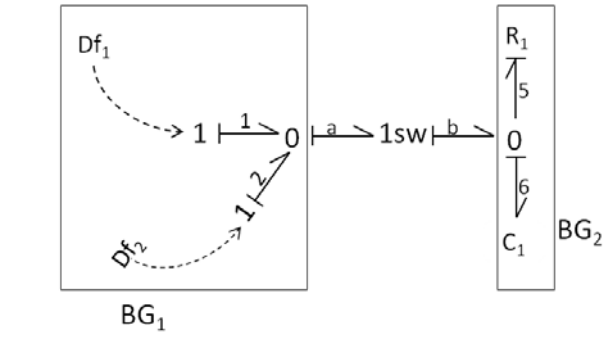
*Finally, we will prove part **IIb** of property 5:*

II.b. If** $\exists$ **non parametric** $PMCS^{\emptyset}\_f_a$ **in** $PS_R \setminus \{d\}$**, then** $BG'_R \setminus \{d\}$ **contains a subsystem of flow equations overdetermined minimal and we cannot compute** e_a **in** $BG'_R \setminus \{d\}$**.

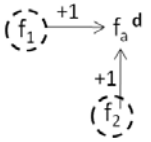
If $PS_R \setminus \{d\}$ contains a non parametric $PMCS^{\emptyset}\_f_a$, then $BG'_R \setminus \{d\}$ can compute f_a from sources and sensors.

However f_a is a source in $BG'_R \setminus \{d\}$, that has a VCA. Then, it has to be possible to transform $PMCS^{\emptyset}\_f_a$ from all sources and sensors, except one, to at least this one sensor, with f_a as input. Let's call $PMCS^{trans}$, this new $PMCS$. (Note that if this transformation is not possible, $BG'_R \setminus \{d\}$ has no VCA).

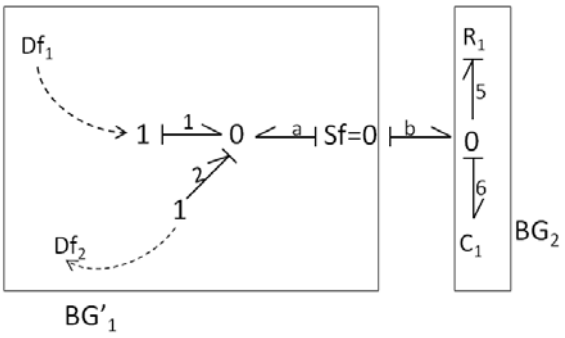
However, $Eq(PMCS^{trans})$ is just determined given f_a. Given that in $BG'_R \setminus \{d\}$ $PMCS^{trans}$ computes



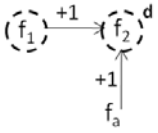
(a) HBG switch ON



(b) TCG switch ON. Non parametric $PMCS\_f_a$.



(c) HBG switch OFF



(d) TCG switch off. Non parametric $PMCS\_f_a$.

Figure 4.17: Proterty 5. HBG in figure a) has a valid causal assignment. The $PMCS\_f_a$ for that causality is in b). The causality changes when the switch is OFF for $BG'_R \setminus \{d\}$, as in c). There is a non parametric $PMCS\_f_2$ ($PMCS^trans$) shown in c), there is an overdetermined subset of flow equations in BG'_1 and e_a cannot be calculated.

a measured variable, $BG'_R \setminus \{d\}$ is overdetermined minimal.

And we cannot compute e_a in $BG'_R \setminus \{d\}$ because by Property 5.II $\nexists$ non parametric $PMCS^{\emptyset}\_e_a$ in $BG'_R \setminus \{d\}$ and given that $\exists$ non parametric $PMCS^{\emptyset}\_f_a$ in $PS_R \setminus \{d\}$, corollary 2 ensures that $\nexists$ parametric $PMCS\_e_a$ with $\{f_a\} = Input(PMCS\_e_a)$ in $BG'_R \setminus \{d\}$.

□

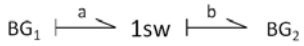
Property 5 is essential to analyze the behaviour of the subsystem where causality changes because of the switching. Now, we can state the fundamental Proposition 3 which predicts the effect of turning only one 1sw to OFF.

Properties 1-5 will help to analyze the structural effect of turning one switching junction from ON to OFF.

Analyzing the structural effect of turning one switching junction from OFF to ON, requires reconstructing an HBG-PC from subsystems that were unconnected when the 1- switching junction was set to OFF. We need to introduce some additional properties to support that process. Particularly, we want to characterize the PMCS of a 1- switching junction of a HBG model of a physical system for each possible causal assignment at the switching, for parametric and non parametric subsystems.

The following properties 6 and 7 tackle this issue, assuming that the HBG model admits the decomposition of a $gdHBG$ as defined in 37, although we relax the conditions on its set of equations, that will be separately stated.

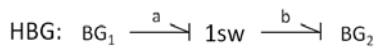
Property 6 (PMCS of a generic HBG that admits the following decomposition and have an indetermined number of sources and sensors, when a is the determining bond of the switching)**.** *Considering:*



- *If HBG is at least determined (all flows and efforts can be computed).*
  - *At bond a:* $\exists$ $PMCS\_f_a$ *that can be:*
    - *Parametric: then* $\{e_a\} = Input(PMCS\_f_a)$ *and* $PMCS\_f_a$ *has no element from* BG_2.
    - *Non parametric: then we have* $PMCS^{\emptyset}\_f_a$ *that has no element from* BG_2 *(computing efforts requires* BG_2*).*
  - *At bond b:* $\exists$ $PMCS\_e_b$ *that can be:*
    - *Parametric: then* $\{f_b\} = Input(PMCS\_e_b)$ *and* $PMCS\_e_b$ *has no element from* BG_1.
    - *Non parametric: then we have* $PMCS^{\emptyset}\_e_b$ *that has no element from* BG_1 *(computing flows requires* BG_1*).*
- *If HBG is underdetermined but all flows can be computed:* $\exists$ $PMCS^{\emptyset}\_f_a$ *in* BG_1, $\forall f_i \in BG_2$ $\exists PMCS\_f_i$ *with* $\{f_b\} = Input(PMCS\_f_i)$ *in* BG_2*, both non parametric and the efforts of their bonds can not be computed.*
- *If HBG is underdetermined but all efforts can be computed:* $\forall e_i \in BG_1$ $\exists$ $PMCS\_e_i$ *with* $\{e_a\} = Input(PMCS\_e_i)$ *in* BG_1, $\exists$ $PMCS^{\emptyset}\_e_b$ *in* BG_2 *both non parametric and the flows of the their bonds can not be computed.*

■

Property 7 (PMCS of a generic HBG that admits the following decomposition and have an indetermined number of sources and sensors, when b is the determing bond of the switching)**.** *PMCS when determining bond is b (b imposes the flow at the junction).*



- *If HBG is at least determined (all flows and efforts can be computed).*
  - *At bond a:* $\exists$ $PMCS\_e_a$ *that can be:*
    - *Parametric: then* $\{f_a\} = Input(PMCS\_e_a)$ *and* $PMCS\_e_a$ *has no element from* BG_2.
    - *Non parametric: then we have* $PMCS^{\emptyset}\_e_a$ *that has no element from* BG_2 *(computing flows requires* BG_2*).*
  - *At bond b:* $\exists$ $PMCS\_f_b$ *that can be:*
    - *Parametric: then* $\{e_b\} = Input(PMCS\_f_b)$ *and* $PMCS\_f_b$ *has no element from* BG_1.
    - *Non parametric: then we have* $PMCS^{\emptyset}\_f_b$ *that has no element from* BG_1 *(computing flows requires* BG_1*).*
- *If HBG is underdetermined but all flows can be computed:* $\exists$ $PMCS^{\emptyset}\_f_b$ *in* BG_2, $\forall f_i \in BG_1$ $\exists PMCS\_f_i$ *with* $\{f_a\} = Input(PMCS\_f_i)$ *in* BG_1*, both non parametric, and the efforts of their bonds can not be computed.*
- *If HBG is underdetermined but all efforts can be computed:* $\exists$ $PMCS^{\emptyset}\_e_a$ *in* BG_1, $\forall e_i \in BG_2$ $\exists$ $PMCS\_e_i$ *with* $\{e_b\} = Input(PMCS\_e_i)$ *in* BG_2*, both non parametric, and the flows of the their bonds can not be computed.*

■

4.3.3 Characterizing the set of HBG-PCs for all modes with the HBG-PCs for all switching junctions ON

An advantage of combining ideal switching junctions and the PCs formalism is that changing the state of one switching junction only influences the HBG-PCs that contain the switching junction. HBG-PCs that do not contain the switching junction can be used to track the system during the discrete change of the state. Moreover, the effect of the switching on the HBG-PCs where it is located is local to those HBG-PCs and do not require analyzing from scratch the complete system model to obtain new HBG-PCs. In this dissertation we assume that the HBG of the complete model has a VCA when all sw-j are set to ON.

Although it may seem a very restrictive assumption, it is satisfied in many systems. For those systems that violate the assumption, we could still apply the proposed method working with a modified version of the original HBG model, HBG'. HBG' can always be obtained from HBG adding as many *virtual* resistances to the system as necessary to create an alternative system with valid causal assignment when every switch is set to ON. We have not explored this approach in this dissertation, which remains an open research option.

Next proposition states the fundamental property of HBG-PCs:

Proposition 2. *Let HBG be the model of a dynamic linear system that has no switching junction in a closed casual path. If the HBG has a valid causal assignment when all the switching junctions are set to ON, then the set of parametric Possible Conflicts of HBG can be computed, for any configuration of the switching junctions, from the set of parametric HBG-PCs of the system with all the switching junctions set to ON.*

To prove proposition 2, we need to introduce some additional notation. Let $PCsw_1sw_2sw_3 \dots sw_n$ denote the set of PCs of a hybrid system with n switching junction for the specified switching junction configuration (i.e. if $n = 4$, $PC1111$ denotes the set of PCs for the mode with all switching junctions ON). Let $PC < k >$ denote the family of PCs for configurations with k switching junctions OFF. Hence, $PC < 0 >= \{PC1111\}$, $PC < 1 >= \{PC0111, PC1011, PC1101, PC1110\}$, and so on. Finally, let $PC < k >^*$ be an element of $PC < k >$, i.e., $PC < k >^*$ is the set of PCs for a particular mode configuration included in $PC < k >$. Specifically, $PC < 0 >^* = \{PC1111\}$.

We can proof proposition 2 decomposing it in the following propositions 3 and 4:

Proposition 3. *The set of Possible Conflicts for any $PC < k+1 >^*$, $0 \leq k \leq n-1$, can be systematically derived from the set of Possible Conflicts of some $PC < k >^*$ for linear systems with valid causal assignment when every switching junction is set to ON, except for some non parametric PCs.*

Proposition 4. *The set of Possible Conflicts for any $PC < k >^*$, $0 \leq k \leq n-1$, can be systematically derived from the set of Possible Conflicts of some $PC < k+1 >^*$ for linear systems with valid causal assignment when one switching junction is set to ON, except for some non parametric PCs.*

Proof 4. *(Proposition 3)*

Proposition 3 can be proven by complete induction on the number of switching junctions set to OFF (k). The proof from k to $k+1$ is essentially the same as the proof from 0 to 1, that we provide in the next paragraphs.

*Let **pc** $\in PC < 0 >^*$ the PC defined by HBG-PC when all the switching junctions in HBG-PC are set to ON. Assume that one switching junction, sj_i in HBG-PC switches from ON to OFF. We have to check whether **pc** is still a PC of some element of $PC < 1 >^*$. The following lemma holds:*

Lemma 1. *If one $sj_i \in$ HBG-PC switches from ON to OFF, **pc** may disappear or transform in a new PC losing some components, but never split into two parametric PCs, although a new non-parametric PC may appear.*

We will prove Lemma 1, without losing generality, for the case when sj_i is a 1- switching junction (similar proof for 0- switching junction) with only two bonds (similar for n bonds, $n \geq 2$).

Given that HBG-PC includes a 1- switching junction that is not in a closed causal path, removing from HBG-PC the sensor at the discrepancy node, d, we obtain a gdnr$HBG-PC$, $HBG-PC \setminus \{d\}$ that admits the decomposition described in definition 38. $HBG-PC \setminus \{d\}$ is just determined and has a VCA. Consequently, Properties 1 to 5 apply to this system.

The effect of changing the state of the i- switching junction on **pc** *depends on whether the change of causality affects the part of the system that contains the discrepancy node. Hence, we have to consider two different configurations:*

1. $BG_R \setminus \{d\}$ *changes causality when switching:* $\mathrm{BG_R\backslash\{d\}} \overset{a}{\longmapsto} 1_{SW} \overset{b}{\longmapsto} \mathrm{BG_{NR}}$
2. BG_{NR} *changes causality when switching:* $\mathrm{BG_{NR}} \overset{b}{\longmapsto} 1_{SW} \overset{a}{\longmapsto} \mathrm{BG_R\backslash\{d\}}$

$PS_R \setminus \{d\}$: $\mathrm{BG_R\backslash\{d\}} \overset{a}{\longleftarrow}$

PS_{NR}: $\overset{b}{\longrightarrow} \mathrm{BG_{NR}}$

We have to analyze both configurations.

1. $BG_R \setminus \{d\}$ ***changes causality when switching:***

   $\mathrm{BG_R\backslash\{d\}} \overset{a}{\longmapsto} 1_{SW} \overset{b}{\longmapsto} \mathrm{BG_{NR}}$

   Setting the 1- switching junction OFF, the system becomes:

   $\mathrm{BG_R\backslash\{d\}} \overset{a}{\longleftarrow} \mathrm{Sf{:}0} \overset{b}{\longmapsto} \mathrm{BG_{NR}}$

   There are two subsystems to analyze:

   (a) BG'_{NR}: $\mathrm{Sf{:}0} \overset{b}{\longmapsto} \mathrm{BG_{NR}}$

   There has not been a change of causality at the non redundant part of HBG-PC. Then, by Property 5.I, BG'_{NR} is just determined. Hence, no new PC appears at the non redundant part of the HBG-PC.

   (b) $BG'_R \setminus \{d\}$: $\mathrm{BG_R\backslash\{d\}} \overset{a}{\longleftarrow} \mathrm{Sf{:}0}$

   There has been a change in causality at the redundant part of HBG-PC. Then **pc** *may transform or may be lost, appearing a new non parametric PC.*

   First, note that in this case $\nexists$ non parametric $PMCS^{\emptyset}\_f_a$ in $PS_R \setminus \{d\}$. This is because with the discrepancy node $d \in PMCS^{\emptyset}\_f_a$, restoring the sensor of d we would have an overdetermined subsystem in BG_R (inverting $PMCS^{\emptyset}\_f_a$ to destination d) and HBG-PC would not be minimal. Consequently, if $\exists$ non parametric $PMCS\_f_a$ in $PS_R \setminus \{d\}$ it has to be one of:

   - *non parametric $PMCS\_f_a$ with $d \in Input(PMCS\_f_a)$*
   - *non parametric $PMCS\_f_a$ with $d \notin Input(PMCS\_f_a)$*

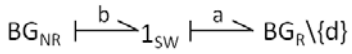
   Then, depending on the existence of a parametric $PMCS\_f_a$ we have:

   i. *If $\exists$ parametric $PMCS\_f_a$ in $PS_R \setminus \{d\}$ with $\{e_a\} = Input(PMCS\_f_a)$:*
      - *By Property 5.IIa, $BG'_R \setminus \{d\}$ is just determined.*
      - *Restoring back the measure at the discrepancy node we have an overdetermined minimal system. Hence, there is a PC at $BG'_R \setminus \{d\}$, with the same discrepancy node. However, the nature of this PC also depends on:*
        - *If additionally $\exists$ non parametric $PMCS'\_f_a \subset$ parametric $PMCS\_f_a$ with $d \in Input(PMCS'\_f_a)$, the overdetermined part of the subsystem is given by the non parametric $PMCS'\_f_a$. Then* **pc** *transform in a non parametric PC.*
        - *If $\nexists$ non parametric $PMCS'\_f_a \subset$ parametric $PMCS\_f_a$ with $d \in Input(PMCS'\_f_a)$, the PC is parametric.*

ii. If $\nexists$ parametric $PMCS\_f_a$ in $PS_R \setminus \{d\}$ with $\{e_a\} = Input(PMCS\_f_a)$ then necessarily $\exists$ non parametric $PMCS'\_f_a$ and we have to consider two cases:

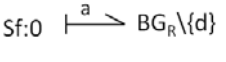
- *If $d \in Input(PMCS\_f_a)$, **pc** transform in a non parametric PC.*
- *If $d \notin Input(PMCS\_f_a)$, **pc** disappears and a new non parametric PC is found when the switch turns to OFF.*

*2. BG_{NR} **changes causality when switching:***



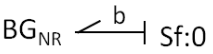
Again, there are two subsystems to analize:

(a) Analysis of $BG'_R \setminus \{d\}$



*Now, by Property 5.I, $BG'_R \setminus \{d\}$ is just determined. Hence, restoring the sensor we obtain BG'_R overdetermined minimal. As a consequence, **pc** remains at the same discrepancy node, although it has lost the elements from BG_{NR}.*

(b) Analysis of BG'_{NR}



There has been a change in causality at the non redundant part of HBG-PC. The appearance of a new PC depends on its parametric or non parametric nature:

- *$\exists$ parametric $PMCS\_f_a$ in PS_{NR} with $\{e_a\} = Input(PMCS\_f_a)$. Then by Property 5.IIa, BG'_{NR} is just determined. Hence, no new PC appears.*
- *$\exists$ non parametric $PMCS^{\emptyset}\_f_a$ in PS_{NR}. Then by Property 5.IIb, BG'_{NR} contains a subsystem of flow equations overdetermined, minimal. Hence a non parametric new PC appears, at a new discrepancy node.*

Corollary 3. *If HBG-PC defines a **pc** with all switches to ON and one $1sw$ turns to OFF.*

1. If the HBG-PC does not contain a non parametric Possible Causal Subgraph to a flow of the $1sw$:

- *If there is a valid causal assignment, **pc** remains, losing some of its components.*
- *No new PC can appear.*

2. If the HBG-PC contains a non parametric Possible Causal Subgraph to a flow of the $1sw$:

- *Even with a valid causal assignment **pc** disappears if the discrepancy node lies on the subsystem which causality changes, remaining if its causality does not change.*
- *A non parametric new PC always appears.*

Corolary 3 proofs Lemma 1.

□

This same result can be applied to switching junctions with more than two bonds (there is only one determining bond in the junction, all the others have the same behaviour). And also for 0- switching junctions considering it behaves as a zero effort source when it switches to OFF.

Proposition 4 can be proven by complete induction on the number of switching junctions set to ON (k). The proof from $k+1$ to k is the same as the proof from 1 to 0.

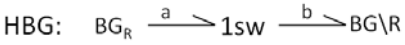
Now we have all switching junctions set to ON except for some 1sw_i that is set to OFF. We have to show that if there exist a parametric **pc** with this 1sw_i set to OFF, when 1sw_i turns to ON, there exists a HBG-PC that defines a parametric PC, pc', that includes **pc**.

Lemma 2. *Let* $\mathbf{pc} \in PC < 1 >^*$ *be a parametric PC. Then, there is a HBG-PC that defines a parametric PC,* $pc' \in PC < 0 >^*$ *and* $\mathbf{pc} \subset pc'$.

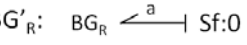
We will proof Lemma 2, without losing generality, for the case when sw_i is a 1sw (similar proof for 0sw) with only two bonds (similar for n bonds, $n \geq 2$)

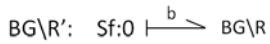
To prove Lemma 2 we have to introduce the necessary structural elements to isolate a HBG-PC from the complete HBG model when the switching junction is set to ON building from two unconnected subsystem, one of them a BG-PC defining **pc**. We start introducing a generic decomposable redundant HBG:

Definition 40. *(Generic decomposable redundant HBG, (gdrHBG)) Let gdrHBG be a Hybrid Bond Graph of a linear system with valid causal assignment describing an over determined system, not necessarily minimal, and including a* $1sw$ *that is not in a closed casual path (cycle when set to ON). gdrHBG admits the decomposition* $gdrHBG = BG_R \cup 1sw \cup BG \setminus R$*, with: with:*

- $BG_R \neq \emptyset$, $BG \setminus R \neq \emptyset$ *and* $BG_R \cap (BG \setminus R) = \emptyset$,
- BG_R *is a redundant subsystem that contains the discrepancy node of* **pc**.
- $BG \setminus R = HBG \setminus \{BG_R \cup 1sw\}$
- *1sw: It represents 1-j switching junction,*
- *Graphically, independently of the causality at the switching junction:* 

And let:

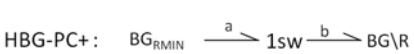
- BG'_R: 

- $BG \setminus R'$: 

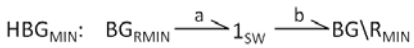
Definition 41. *(Stepping HBGs) Stteping HBGs are the set of subsystems found reconstructing the complete gdrHBG from a given* **pc** .

- *When the switch is set to OFF, exists a BG-PC that defines* **pc**. *Lets call* BG'_{RMIN} *this BG-PC and* PS_{RMIN} *the partial subsystem obtained from* BG'_{RMIN} *removing the flow source but keeping its bond:*

- *When the switch is set to ON,* BG_{RMIN} *is connected to* $BG \setminus R$. *Define* $HBG - PC+ \subseteq gdrHBG$ *as the system* $BG_{RMIN} \cup 1sw \cup (BG \setminus R)$.



- *Define* $HBG_{MIN} \subseteq HBG - PC+$, $HBG_{MIN} = BG_{RMIN} \cup 1sw \cup (BG \setminus R_{MIN})$, *with* $BG \setminus R_{MIN} \subseteq (BG \setminus R)$.



We have to show that $PS_{RMIN} \subset HBG_{MIN} \subseteq HBG - PC+ \subseteq gdrHBG$ and that HBG_{MIN} is a HBG-PC. This requires tracking the transformation of the MCS that defines the $\mathbf{pc} \in BG'RMIN$ (PS_{RMIN}) when the switching junction turns to ON. We provide the proof for the case when **pc** is parametric. For the, non parametric case, we will present a counter example showing that a non parametric **pc** may disappear setting the switching junction to ON.

1. **PMCS of** BG'_{RMIM}.

   BG'_{RMIN} is overdetermined minimal. Let d be the sensor at the discrepancy node of **pc**. Removing this sensor we obtain $BG'_{RMIN} \setminus \{d\}$ that is just determined.

   Given that we consider only the parametric case and $BG'_{RMIN} \setminus \{d\}$ is just determined, $\exists\ PMCS\_e_a$ parametric, with $\{f_a\} = Input(PMCS\_e_a)$ in $BG'_{RMIN} \setminus \{d\}$:

   - $\exists\ PMCS\_e_a$ (otherwise, we cannot compute e_a) and the set of equations Eq($PMCS\_e_a$) is just determined.
   - If $f_a \notin Input(PMCS\_e_a)$, then $f_a \notin Var(PMCS\_e_a)$ because a source can only occur at the start or the input of a PMCS. In this case $BG'_{RMIN} \setminus \{d\}$ is not just determined.
     - $Eq(PMCS\_e_a)$ would have a unique solution and changing the value of the flow source $BG'_{RMIN} \setminus \{d\}$ would admit solutions with different values of f_a and the same value of e_a.

2. **PMCS of** $HBG - PC + \setminus\{d\}$.

   $HBG - PC+$ is the first subsystem to consider when setting the switching junction to ON connects BG'_{RMIN} to the rest of the system, $BG \setminus R$.

   The complete system, $gdrHBG$ has a VCA (we assume a VCA with all switches set to ON). Then, $HBG - PC+$ and $HBG - PC + \setminus\{d\}$ have at least the same VCA.

   The nature of the PMCS that we can find at the switching junction of $HBG - PC + \setminus\{d\}$ depends on two factors: the nature of the original $gdrHBG$ (which determines the nature of BG'_{RMIN} and $BG \setminus R'$, providing three different cases as properties 6 and 7 state) and which subsystem changes it causal assignment at the switching (which adds another to cases). We are going to analyze them attending to the change of causality at the switching.

3. **Property 8: when switching changes causality of** BG_{RMIN}**,** $\exists\ PMCS\_f_a$ **parametric, with** $\{e_a\} = Input(PMCS\_f_a)$ **and no elements in** $BG \setminus R$ **and** $\exists\ PMCS\_e_b$ **parametric or non parametric, with no element in** BG_{RMIN}**, both in** $HBG - PC + \setminus\{d\}$.

   This configuration happens when bond a is the determining bond of the switching junction:

   Previous step 2 shows that $\exists\ PMCS\_e_a$ parametric, with $\{f_a\} = Input(PMCS\_e_a)$ in $BG'_{RMIN} \setminus \{d\}$

   Given that $HBG - PC + \setminus\{d\}$ has a VCA, Property 1 guarantees that $\exists\ PMCS\_f_a$ parametric, with $\{e_a\} = Input(PMCS\_f_a)$ in $HBG - PC + \setminus\{d\}$ with no elements in $BG \setminus R$.

   By Property 6,there are three possibilities for PMCS in $BG \setminus R$ and bond b:

   (a) $\exists\ PMCS\_e_b$ that can be:
   - Parametric: then $\{f_b\} = Input(PMCS\_e_b)$ and $PMCS\_e_b$ has no element from BG_{RMIN} (its effect is reflected by f_b).
   - Non parametric: then we have $PMCS^{\emptyset}\_e_b$ that has no element from BG_{RMIN} (computing flows requires BG_{RMIN}).

   (b) $\exists\ PMCS^{\emptyset}\_f_a$ in BG_{RMIN}, $\forall f_i \in BG_{RMIN} \setminus \{d\}$ $\exists PMCS\_f_i$ with $\{f_b\} = Input(PMCS\_f_i)$ in $BG_{RMIN} \setminus \{d\}$, both non parametric and the efforts of their bonds can not be computed. However, this case is incompatible with $BG'_{RMIN} \setminus \{d\}$ parametric and minimal.

   (c) $\forall e_i \in BG_{RMIN}$ $\exists\ PMCS\_e_i$ with $\{e_a\} = Input(PMCS\_e_i)$ in BG_{RMIN}, $\exists\ PMCS^{\emptyset}\_e_b$ in $BG_{RMIN} \setminus \{d\}$ both non parametric and the flows of the their bonds can not be computed. However, this case is incompatible with $BG'_{RMIN} \setminus \{d\}$ parametric and minimal.

Hence only case a) applies: $\exists\ PMCS\_e_b$ parametric or non parametric with no element in BG_{RMIN}.

4. **Property 9: When switching does not change causality of BG_{RMIN}, $\exists\ PMCS\_e_a$ parametric, with $\{f_a\} = Input(PMCS\_e_a)$ in $BG'_{RMIN}\backslash\{d\}$ and $\exists\ PMCS\_f_b$ parametric or non parametric, with no element in BG_{RMIN}, both in $HBG - PC + \backslash\{d\}$.**

   This configuration happens when bond b is the determining bond of the switching junction:

   $BG'_{RMIN}\backslash\{d\}$: $BG_{RMIN}\backslash\{d\} \xleftarrow{a}$ Sf:0 HBG-PC+$\backslash\{d\}$: $BG_{RMIN}\backslash\{d\} \xrightarrow{a}$ 1sw $\xrightarrow{b}$ BG\R

   Previous step 2 shows that $\exists\ PMCS\_e_a$ parametric, with $\{f_a\} = Input(PMCS\_e_a)$ in $BG'_{RMIN} \setminus \{d\}$

   By Property 7, there are three possibilities for PMCS in $BG \setminus R$ and bond b:

   (a) $\exists\ PMCS\_f_b$ that can be:
   - Parametric: then $\{e_b\} = Input(PMCS\_f_b)$ and $PMCS\_f_b$ has no element from BG_{RMIN} (its effect is reflected by f_b).
   - Non parametric: then we have $PMCS^{\emptyset}\_f_b$ that has no element from BG_{RMIN} (computing flows requires BG_{RMIN}).

   (b) $\exists\ PMCS^{\emptyset}\_f_b$ in $BG_{RMIN} \setminus \{d\}$, $\forall f_i \in BG_{RMIN}\ \exists PMCS\_f_i$ with $\{f_a\} = Input(PMCS\_f_i)$ in BG_{RMIN}, both non parametric, and the efforts of their bonds can not be computed. However, this case is incompatible with $BG'_{RMIN} \setminus \{d\}$ parametric and minimal.

   (c) $\exists\ PMCS^{\emptyset}\_e_a$ in BG_{RMIN}, $\forall e_i \in BG_{RMIN} \setminus \{d\}\ \exists\ PMCS\_e_i$ with $\{e_b\} = Input(PMCS\_e_i)$ in $BG_{RMIN} \setminus \{d\}$, both non parametric, and the flows of the their bonds can not be computed. However, this case is incompatible with $BG'_{RMIN} \setminus \{d\}$ parametric and minimal.

   Hence only case a) applies: $\exists\ PMCS\_f_b$ parametric or non parametric with no element in BG_{RMIN}.

Now, we are ready to prove Lemma 2.

I: switching changes causality of BG_{RMIN}

By property 3, $\exists\ PMCS\_f_a$ parametric, with $\{e_a\} = Input(PMCS\_f_a)$ in $HBG - PC + \backslash\{d\}$ with no elements in $BG \setminus R$.

Also by property 3, $\exists\ PMCS\_e_b$ parametric or non parametric with no element in BG_{RMIN}. Then we can define $BG \setminus R_{MIN} \subseteq BG \setminus R$ as the minimal sBG that includes all the elements of $PMCS\_e_b$.

Then, we can build the system $HBG_{MIN} \setminus \{d\}$:

$HBG_{MIN}\backslash\{d\}$: $BG_{RMIN}\backslash\{d\} \xrightarrow{a}$ 1sw $\xrightarrow{b}$ $BG\backslash R_{MIN}$

$HBG_{MIN} \setminus \{d\}$ is just determined, because Eq($PMCS\_f_a$) plus $e_a = e_b$ is just determined given e_b and Eq($PMCS\_e_b$) plus $f_b = f_a$ is just determined given f_a.

Adding the sensor at the discrepancy node, d, we obtain HBG_{MIN} overdetermined minimal and hence a HBG-PC.

When the switching junction is set to OFF, HBG_{MIN} defines **pc**.

II: switching does no change causality of BG_{RMIN}

Now by property 4, $\exists\ PMCS\_e_a$ parametric, with $\{f_a\} = Input(PMCS\_e_a)$ in $BG'_{RMIN} \setminus \{d\}$

Also by property 4, $\exists\ PMCS\_f_b$ parametric or non parametric, with no element in BG_{RMIN}, in PMCS of $HBG - PC + \backslash\{d\}$. Then we can define $BG \setminus R_{MIN} \subseteq BG \setminus R$ as the minimal sBG that includes all the elements of $PMCS\_f_b$.

Then, we can build the system $HBG_{MIN} \setminus \{d\}$:

$HBG_{MIN}\backslash\{d\}$: $BG_{RMIN\backslash\{d\}} \xrightarrow{a}$ 1sw $\xrightarrow{b}$ $BG\backslash R_{MIN}$

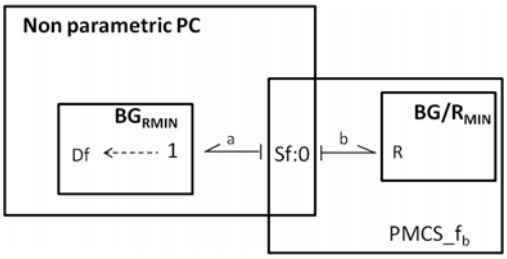
$HBG_{MIN} \setminus \{d\}$ is just determined, because Eq($PMCS\_e_a$) plus $f_a = f_b$ is just determined given f_b and Eq($PMCS\_f_b$) plus $e_b = e_a$ is just determined given e_a.

Adding the sensor at the discrepancy node, d, we obtain HBG_{MIN} overdetermined minimal and hence a HBG-PC.

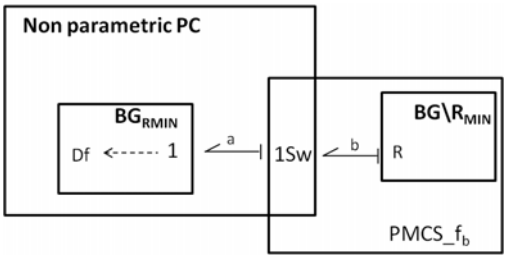
When the switching junction is set to OFF, HBG_{MIN} defines **pc**.
Proofs **I** and **II** demonstrate Lemma 2.

We have performed a similar analysis when the **pc** that exist when the switching junction is set to OFF is non parametric. In this case, the non parametric PC may disappear under certain circumstances. However, in this dissertation we limit to show an example of two different cases: (1) the non parametric **pc** disappears, and (2) the non parametric **pc** remains.
Case 1) The non parametric **pc** disappears:

- Switching junction set to OFF: Non parametric **pc** at BG_R (that in the example is equal to BG_{RMIN} and $\exists PMCS\_e_b$ parametric in $BG' \setminus R$ (that in the example is equal to $BG' \setminus R_{MIN}$) with f_b at its input.

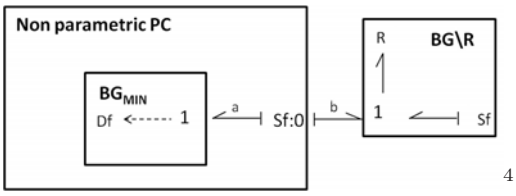


- Switching junction set to ON: The complete system is not a HBG-PC. Basically, because to compute e_b we need Df and we lose redundancy (that was due to $Sf:0$).



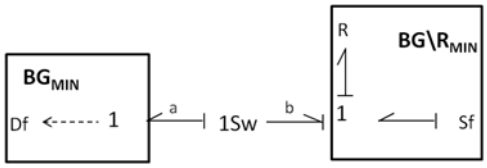
Case 2) The non parametric **pc** remains:

- Switching junction set to OFF: Non parametric **pc** at BG_R and $\exists PMCS\_f_b$ non parametric in $BG \setminus R$ and bond b.



[4]

- Switching junction set to ON: The complete system is a HBG-PC, although non parametric.



[4]BG\R has not a valid causal assignment when the switching junction is OFF.

4.4 HBG-PCs Generation

Once the HBG-PC structure is defined and its behaviour analyzed, we need to explain how to build those minimal redundant subsystems for hybrid systems. The complete system is modeled as a HBG, as we have previously stated.

As explained before, the more efficient way to derive the HBG-PCs consist of assuming all switching junctions are ON in the HBG modeling the whole system. From that HBG, which can be seen as a BG once all the switching junctions are configured to its ON position, it is possible to automatically derive the TCG [76]. In Chapter 3 it has been summarized the algorithm presented in [23] to derive PCs from a TCG. The algorithm is applied in the TCG to obtain the TCG structure of each PC.

Chapter 3 deals with continuous systems while now, we are working on hybrid systems, so the structures we have derived from the TCG are the PCs assuming all switching junctions are ON (the working mode of the system have all the actuators ON), but we need their HBG models (HBG-PCs) to configure the switching junctions according to the system working mode.

Algorithm 4 presents the pseudocode to derive the HBG-PC from its TCG representation.

Algorithm 4: Derive the HBG-PC from its TCG for discrepancy node d.

Require: TCG model of a PC (*tcg*);HBG of the complete system (*hbg*)
Ensure: *hbg-pc*

```
1: Start in the discrepancy node d;
2: Mark in hbg the junction associated to d;
3: /* Search deep first in tcg */
4: while not visited all nodes in tcg do
5:    Get next element, deep first, in tcg;
6:    if it is an edge then
7:       Mark in hbg the system parameters in the edge;
8:    else
9:       /* It will be a node: e_i or f_i */
10:      Mark in hbg the bond related to the node;
11:   end if
12: end while
13: Mark in hbg the sources and the sensors in tcg;
14: hbg-pc = Remove from hbg all the componens not marked;
15: return hbg-pc
```

The HBG-PCs obtained as Algorithm 4 presents will be used to track and monitor system behaviour for fault diagnosis tasks.

4.5 Running Example: A hybrid four-tank system

To show significant concepts of the approach, we considered the hybrid four-tank system shown in Figure 4.18. The system has input flow to tanks 1 and 3. This source is common for both tanks but it can be deactivated separately for one of them. Tanks 2 and 4 are connected to tanks 1 and 3, respectively, through a connecting pipe placed at a distance h above the base of the tanks. Figure 4.18 shows the four-tank system, and Figure 4.19 shows its HBG model.

The system has four switching junctions: SW_1, SW_2, SW_3 and SW_4. $CSPEC_{SW1}$ and $CSPEC_{SW3}$ are controlled ON/OFF transitions, while $CSPEC_{SW2}$ and $CSPEC_{SW4}$ are autonomous transitions. Both kinds of transitions are represented using a finite state machine. Figure 4.20 shows: a) the automaton associated with $CSPEC_{SW1}$ and b) the automaton representing the autonomous transition in $CSPEC_{SW2}$. Since the system is symmetric, automata for $CSPEC_{SW3}$ and $CSPEC_{SW4}$ are equivalent to the ones shown in Figure 4.20.

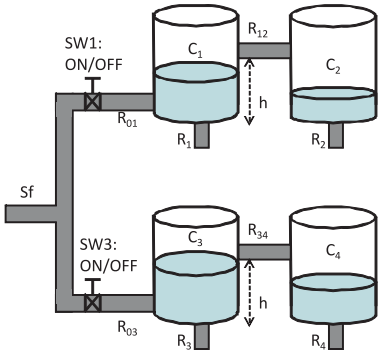


Figure 4.18: Schematics of the four-tank system

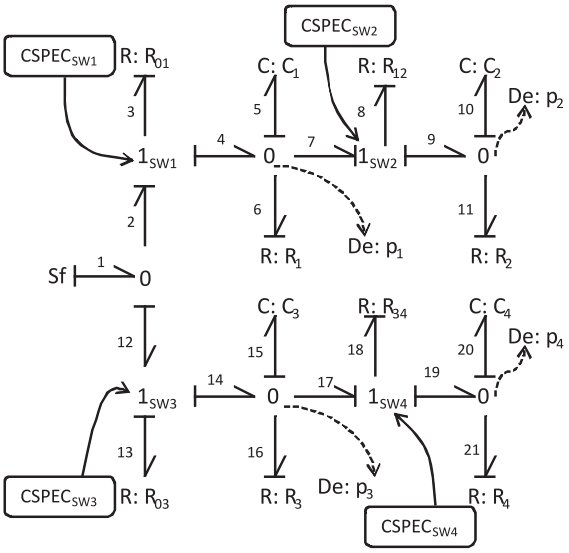


Figure 4.19: Bond graph model of the plant.

Assuming all switching junctions are *ON*, the Temporal Causal Graph of the system can be automatically computed using the bond graph model (see Figure 4.21).

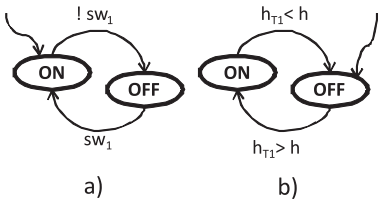


Figure 4.20: a) Automaton associated with the ON/OFF switching junction SW_1; b) Automaton representing the autonomous transition in SW_2.

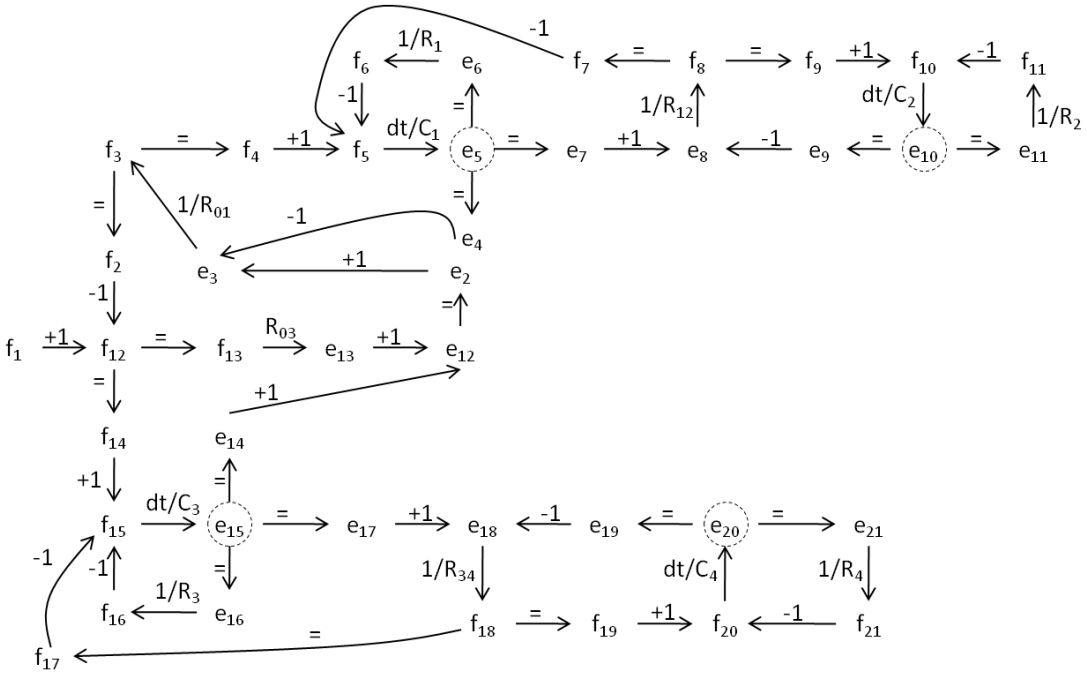


Figure 4.21: Temporal Causal Graph of the four-tank system.

4.6 Hybrid systems fault diagnosis

The four-tank hybrid system (Figure 4.18) have four HBG-PCs. Each one of them estimates one of the measured variables (p_1, p_2, p_3, or p_4). Figure 4.22 shows the HBG-PCs.

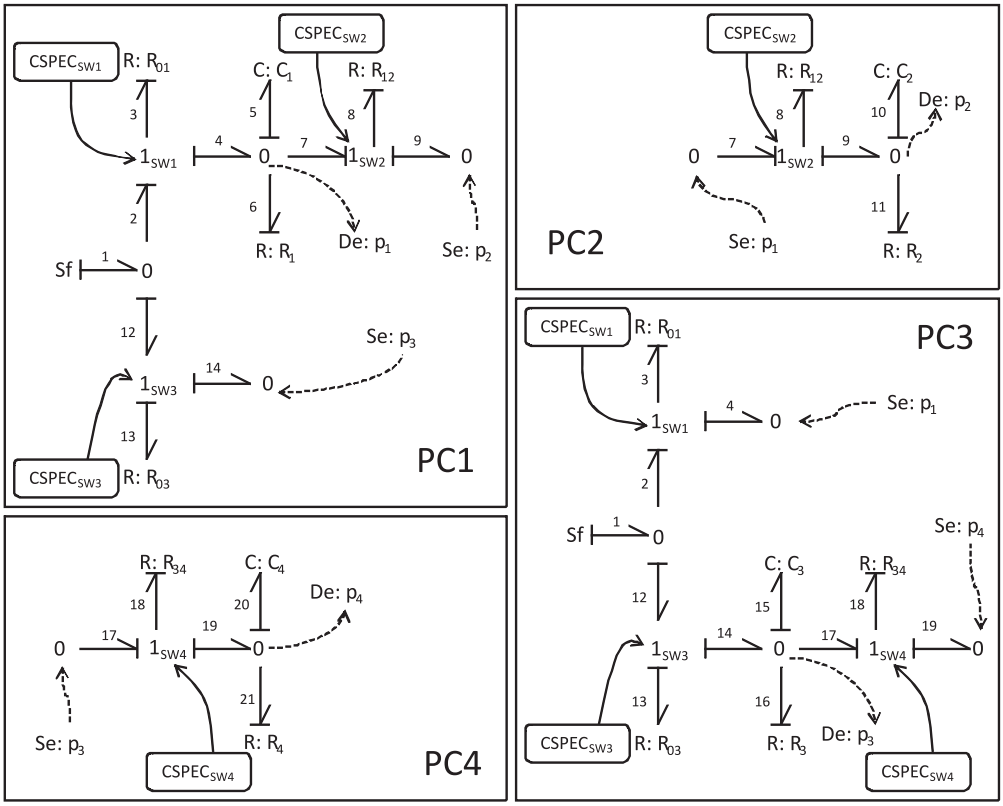


Figure 4.22: HBG-PCs found for the four-tank system.

4.6.1 Tracking system behaviour

HBG-PCs can be used to track hybrid system behaviour[17]. As a first step, the complete set of HBG-PCs in the system assuming that all switching junctions are set to *ON* are computed, as it has been previously explained. When a mode change occurs we will not need to run the HBG-PCs computation again, just an update of the current HBG-PCs, to accomplish the new configuration changes, will be necessary. Moreover, since mode changes effects only propagate locally, the great advantage of model decomposition is that mode changes only affect to a subset of HBG-PCs. Table 4.1 shows the switching junctions (in rows) included in the HBG-PCs (in columns) for the four-tank system. The signature matrix in Table 4.1 relates HBG-PCs and sw-j, hence it can be used to relate faults affecting sw-j and HBG-PCs, for that reason we have termed this structure as the Hybrid Fault Signature Matrix (H-FSM). As shown in Table 4.1, each switching junction is only included in two HBG-PCs, while the other two HBG-PCs are not affected by mode changes in the junction. The HBG-PCs that are not affected by the mode change, will not need causality reassignment, and will be able to remain tracking the system without any change. Then, when a HBG-PC includes a particular switching junction and a mode change occurs in such switching junction, the local causality reassignment and model update for the HBG-PC must occur quickly (before the true state and the state estimation diverge) to avoid false positives. In this work, HBG-PCs are implemented as reconfigurable block diagram (BD) models as proposed in [106].

First, to perform quick model reconfiguration, it is important to notice that computational structures for some elements of the HBG-PCs remain invariant across all system modes. For example, causality of

source elements (Se, Sf) and storage elements (C, I) of the bond graph model remain constant across all the system modes. These elements are identified offline. Using such information, the number of components in the PC that are susceptible to suffer from online causality reassignment is reduced. For the rest of the elements, those whose causality may vary, the Hybrid Sequential Causal Assignment Procedure (Hybrid SCAP) algorithm is used for efficient causality reassignment by quickly reconfiguring the BD models.

Details of the algorithm and the process to compute Block Diagrams from Bond Graph models can be found in [106].

	HBG – PC1	**HBG – PC2**	**HBG – PC3**	**HBG – PC4**
$\mathbf{1_{SW_1}}$	1		1	
$\mathbf{1_{SW_2}}$	1	1		
$\mathbf{1_{SW_3}}$	1		1	
$\mathbf{1_{SW_4}}$			1	1

Table 4.1: Hybrid Fault Signature Matrix (HFSM) of the four-tank system showing the relations between switching junctions and each HBG-PC.

In case a fault occurs, the HBG-PC residuals must be significantly different to zero. That triggers the fault detection and isolation stage.

We will explain in the following two subsections our FDI proposal for hybrid systems for both parametric and discrete faults. We will illustrate the proposal using the four-tank system in the running example.

4.6.2 Fault detection and isolation. Parametric faults

Fault isolation of parametric faults is performed by means of the Reduced Qualitative Fault Signature Matrix (RQ-FSM). Table 4.3 shows the RQ-FSM for the four-tank system in the mode where each switch is ON, it contains the relation between HBG-PCs and system parameters. For a given mode, the Q-FSM can be online computed from the TCG associated to an HBG-PC [76]. In this table each column represents a measurement in the TCG obtained from the original HBG, which is also the source of a discrepancy for an HBG-PC. The table shows the qualitative fault signatures as computed in TRANSCEND [76], except that it is minimal: each column represent the expected effect on the measurement (HBG-PC discrepancy node or output) for the set of faults, in rows [23].

	HBG – PC1	**HBG – PC2**	**HBG – PC3**	**HBG – PC4**
$\mathbf{C_1}$	1	1		
$\mathbf{C_2}$		1		
$\mathbf{C_3}$			1	1
$\mathbf{C_4}$				1
$\mathbf{R_{01}}$	1		1	
$\mathbf{R_{03}}$	1		1	
$\mathbf{R_1}$	1			
$\mathbf{R_2}$		1		
$\mathbf{R_3}$			1	
$\mathbf{R_4}$				1
$\mathbf{R_{12}}$	1	1		
$\mathbf{R_{34}}$			1	1

Table 4.2: (Parametric) Fault Signature Matrix of the four-tank system.

Fault detection shows the possible conflicts which actually become conflicts, i.e. their residuals have been activated. In the fault isolation stage, the set of fault candidates is obtained as the minimal hitting set of the conflicts using the RQ-FSM.

	HBG – PC1	HBG – PC2	HBG – PC3	HBG – PC4
$\mathbf{C_1^+}$	$-+$			
$\mathbf{C_2^+}$		$-+$		
$\mathbf{C_3^+}$			$-+$	
$\mathbf{C_4^+}$				$-+$
$\mathbf{R_{01}^+}$	$0-$		$0+$	
$\mathbf{R_{03}^+}$	$0+$		$0-$	
$\mathbf{R_1^+}$	$0+$			
$\mathbf{R_2^+}$		$0+$		
$\mathbf{R_3^+}$			$0+$	
$\mathbf{R_4^+}$				$0+$
$\mathbf{R_{12}^+}$	$0-$	$0-$		
$\mathbf{R_{34}^+}$			$0+$	$0-$

Table 4.3: Reduced Qualitative Fault Signature Matrix (RQ-FSM) of the four-tank system.

4.6.3 Fault detection and isolation. Discrete faults

For the system configuration where all the switches are *ON*, the relation between the HBG-PCs and the switching junctions can be seen in the HFSM in Table 4.1. This is an important information that we will use for discrete faults isolation.

Discrete faults in this work are defined to be faults in discrete actuators, which means, commanded mode switches that do not perform the correct action. There are four faulty situations to be considered, where SW_i denotes the switching junction i of the system.

1. SW_i $1 \rightarrow 1$: SW_i is stuck ON (1).
2. SW_i $0 \rightarrow 0$: SW_i is stuck OFF (0).
3. SW_i $0 \rightarrow 1$: Autonomous switch ON (SW_i is OFF (0) and it switches to ON itself (1)).
4. SW_i $1 \rightarrow 0$: Autonomous switch OFF (SW_i is ON (1) and it switches to OFF itself (0)).

Situation 3 (autonomous switch ON) is not feasible in most of the actuators nowadays, because they have a safety position which keeps it OFF in case of malfunction. Regarding situation 4, it is equivalent to a 100% parametric fault in some scenarios.

As we previously mentioned when we stated the assumptions in this work, the operation mode of the system is known before a fault occurs. The magnitude related to the actuator can be sometimes measured, but it can be inaccurate or even wrong. Because of that, we do not use that information in our proposal.

Since discrete faults generally have a bigger and potentially more dangerous influence in the system behaviour, in our approach we will try to isolate them, before considering parametric faults. To do so, we will borrow some ideas from Dressler and Struss [46] in terms of exoneration (although fault signature matrix analysis using exoneration is the classical FDI approach [33]), and from Raiman [98]

Fault candidates concerning discrete faults will be preferred over parametric faults, hence, we need to either confirm them or reject them at the early stage of the fault isolation and identification process. To speed up the discrimination stage, we will use structural information in the HFSM: the relation between HBG-PCs and the switching junctions, as expressed in Table 4.1, and we will introduce qualitative information about the sign of the residuals[5] of each HBG-PC and its commanded actuators, what is called Hybrid Qualitative Fault Signature Matrix (HQFSM). Table 4.4 presents the HQFSM for our case study, signs represent the variation of the residual, which is built using the measured value in the actual system and the estimation for the measurement in the hypothetical mode we will be if there is actually a fault.

We are going to focus on the commanded mode changes; in our case study, there are only 2 commanded actuators: SW_1 and SW_3. So that, the HQFSM will be only built for SW_1 and SW_3.

[5]Residuals are calculated as the actual value of the measurement - the estimated value.

	HBG-PC1	HBG-PC3
$\mathbf{1_{SW_1}1 \rightarrow 1}$	+	-
$\mathbf{1_{SW_1}0 \rightarrow 0}$	-	+
$\mathbf{1_{SW_1}0 \rightarrow 1}$	+	-
$\mathbf{1_{SW_1}1 \rightarrow 0}$	-	+
$\mathbf{1_{SW_3}1 \rightarrow 1}$	-	+
$\mathbf{1_{SW_3}0 \rightarrow 0}$	+	-
$\mathbf{1_{SW_3}0 \rightarrow 1}$	-	+
$\mathbf{1_{SW_3}1 \rightarrow 0}$	+	-

Table 4.4: Hybrid Qualitative Fault Signature Matrix (HQFSM).

Based on the activated residuals for the set of active HBG-PCs, the structural information in the HQFSM (Table 4.4), and the QFSM (Table 4.3), we build the current set of fault candidates. This set can contain both discrete and parametric faults. If there are not discrete faults as fault candidates, we perform regular fault isolation and identification as described in [19] (See Chapter 3). Otherwise, we consider discrete faults as preferred candidates.

Assuming switch values are not measured, discrete faults will be tested to confirm or discard them, hence we look at the HFSM (Table 4.1) to identify affected SW_i according to the activated PCs. Meanwhile, the HBG-PCs tracking the system before the detection time will continue doing it to update the set of candidates in case of new activations.

We look at the QFSM of activated HBG-PCs in Table 4.3. Those qualitative signatures that do not match observed signatures can be rejected. For each discrete fault whose qualitative signature matches the HQFSM (Table 4.4), we build a new potential mode and its HBG-PCs. Since switch values are not measured, we need to simulate them during a period σ_t[6]. Eventually, during that σ_t period, the HBG-PCs from the actual mode will converge. If all the HBG-PCs of a candidate mode converge, their residuals are deactivated, the discrete fault is identified, so the initial HBG-PCs are stopped and the HBG-PCs from the new mode continue tracking the system. If none of the tested modes converges the fault is assumed to be parametric and the common parametric fault detection and isolation procedure will be performed using the Reduced QFSM (Table 4.3) to obtain an isolation as accurate as possible.

As mentioned above, discrete faults usually introduce high nonlinearities in the system outputs, that should be easily detected if magnitudes related to the failing switch were measured, generating almost instantaneous detection for discrete faults. In case the measurement of the switching element is available, exoneration could be applied. However, if we do not assume that the magnitude can be measured, the variable related to the discrete fault might not be available for detection and isolation, and there could be delays in the residual response related to capacitor effects. Hence, we need to wait an additional period of time to confirm or to reject discrete faults.

4.7 Results for hybrid four-tank system

The hybrid four-tank system (Figure 4.18) has four HBG-PCs (Figure 4.22). The relation between HBG-PCs and the switching junctions, assuming all of them are ON have been presented in the HFSM (Table 4.1), while the qualitative information related to HBG-PCs and the system parameters appears in the RQ-FSM (Table 4.3).

To illustrate the validity of the approach we have ran different fault scenarios to cover most of the faults in the system. We also introduced several controlled and autonomous transitions in the system to test how HBG-PCs are capable to efficiently transition between different states in the system without triggering wrong fault detections. Two particularly interesting scenarios have been chosen to illustrate the fault isolation capabilities of the HBG-PCs approach: (*i*) a discrete fault (occurring in switching junction SW_1); and (*ii*) a parametric fault (occurring in resistance R_{01}). Both experiments have been run during 700 s using a

[6] σ_t will be empirically obtained for each system.

sampling period of 1 s; the level of noise in the measurements is 5%.

4.7.1 Tracking nominal behaviour

First of all, we will show the validity of the HBG-PCs approach to track the hybrid system behaviour. We have implemented the four hybrid PCs for the four-tank system and run different simulation experiments. Figure 4.23 shows the diagnosis results obtained for one of these experiments. First column in Figure 4.23 compares the four measurements against its estimation provided by the corresponding HBG-PC, while second column shows the residual obtained for each HBG-PC.

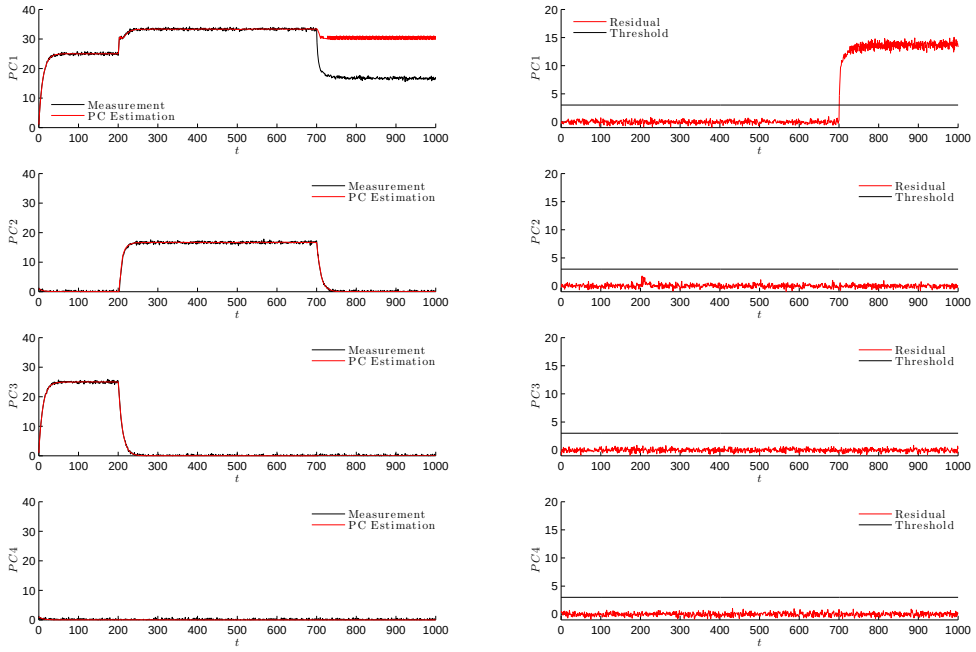


Figure 4.23: Diagnosis experiment for the four-tank system.

In this experiment, we assume that the water tanks are initially empty, and start to fill at constant rate. Hence, the initial configuration of the system is SW_1 and SW_3 set to ON, and SW_2 and SW_4 set to OFF. Tanks 1 and 3 start to fill, and approximately at time 50 seconds both tanks reach stationary state. At this time, the level in tank 1, h_{T1}, and the level in tank 3, h_{T3}, are lower that the height of the connecting pipes, h, and consequently, there is no flow through the connecting pipes, and tanks 2 and 4 remain empty.

At time 200 seconds, the mode of the system changes, and controlled junction SW_3 is set to OFF. Simultaneously, HBG-PC1 and HBG-PC3, which contain SW_3, change their modes, and quickly reassign causality by running Hybrid SCAP. Causality reassignment is done quickly using the Block Diagrams, and, as shown in Figure 4.23, HBG-PC1 is able to correctly estimate the level of tank 1 immediately after the mode change. The situation is even better for HBG-PC3 where no change in the causality is needed, and the HBG-PC remains the same just eliminating some equations. Regarding HBG-PC2 and HBG-PC4, since both HBG-PCs do not contain the switching junction SW_3, none of them is affected by the mode change.

Since SW_3 has been set of OFF, the level of tank 3 decreases until it becomes zero, while the level of tank 1 increases. At time 210 seconds, the level of tank 1 reaches the height of the connecting pipe between tanks 1 and 2. At this point, the autonomous junction SW_2 transitions to ON mode and water begins to fill the tank 2. Two HBG-PCs, HBG-PC1 and HBG-PC2, are affected by this mode change. In both cases, the block diagrams update the models of the HBG-PCs quickly, and both of them are able to correctly estimate

the measurements for the new mode. Note that residual for HBG-PC2 slightly increases at time 210 seconds due to the delay related to the HBG-PC2 model update.

At time 250 seconds, the system reaches again stationary state. Then, at time 700 seconds a 20% leak in tank 1 occurs. As a consequence, the level of tank 1 decreases, while the estimation of HBG-PC1 does not. Hence, residual of HBG-PC1, that is the only one containing C_1 as a fault candidate, activates, triggering the fault isolation procedure. Regarding HBG-PC2, since the level of the tank 1 decreases due to the fault, at time 705 seconds the autonomous junction SW_2 transitions again to *OFF* mode, and HBG-PC2 changes mode again. The HBG-PC model is updated immediately and it is able to correctly estimate the level of tank 2 for the new mode.

In these tests about the ability to track nominal and faulty behaviour, we did not consider discrete faults, and we did not generate new HBG-PCs. That is done in the following tests.

4.7.2 Discrete fault in SW_1

In this scenario, illustrated in Figures 4.24, 4.25 and 4.26, we assume that the water tanks are initially empty, and start to fill at constant rate. Hence, the initial configuration of the system is SW_1 and SW_3 set to *ON*, and SW_2 and SW_4 set to *OFF*. Tanks 1 and 3 start to fill, and approximately at time 50 s both tanks reach stationary state (Fig. 4.24). At time 500 s, an autonomous switch *OFF* fault is introduced in switching junction SW_1. At the next time step, both HBG-PC1 and HBG-PC3 residuals start to deviate, and at time 502 s the fault is detected (see Fig. 4.24). Some seconds later, at time 507 s a $0-$ signature is derived for HBG-PC1 residual, and a $0+$ signature is derived for HBG-PC3 residual, triggering the discrete fault isolation mechanisms. As explained in the previous section, our discrete fault isolation algorithm first assumes that a discrete fault has occurred in the system. In Table 4.1 we can see that switching junctions SW_1 and SW_3 are fault candidates, so we have to check the HQFSM in Table 4.4 using the qualitative signatures. Looking at Table 4.4 we can see that for $0-$ and $0+$ signatures in HBG-PC1 and HBG-PC3, respectively; only four discrete fault candidates are possible: stuck *OFF* in SW_1, autonomous transition to *OFF* in SW_1, stuck *ON* in SW_3, and autonomous transition to *ON* in SW_3. Moreover, since we assume single fault, and we know the state of the switching junctions before the detection (for these experiments both switching junctions were set to *ON* mode), only two of the four fault candidates are possible, and the candidates set is reduced to autonomous transition to *OFF* in SW_1 and stuck *ON* in SW_3.

At this point, the hybrid diagnosis framework creates two different instances of the HBG-PCs in the system, one for each fault candidate. It quickly reassigns causality by running Hybrid SCAP for the mode transitions, and tracks the system for an empirically determined time interval (in this work, since the dynamic of the system is quite fast we have fixed an interval of 10 s) to isolate the fault. The evolution of each fault candidate is shown in Figures 4.25 and 4.26, for SW_1 and SW_3 fault candidates, respectively. First column in the figures compares measurements of the deviated residuals (HBG-PC1 and HBG-PC3) against its estimations provided by the corresponding HBG-PC, while second column shows the residual obtained for each one[7]. Looking at Figure 4.25 we clearly see that if we hypothesize an autonomous transition to *OFF* in SW_1, the HBG-PCs estimations move to a stationary state with similar values to the measured ones. Looking at the residuals. we can also see this effect, their values become almost zero, thus confirming the fault in SW_1 as the real fault in the system. On the other hand, if a stuck *ON* fault in SW_3 fault is hypothesized (see Figure 4.26), the residual activation remains present, and consequently this fault candidate will be discarded after the σ_t period of time.

[7]HBG-PC2 and HBG-PC4 are not shown here since none of these HBG-PCs is affected by the fault.

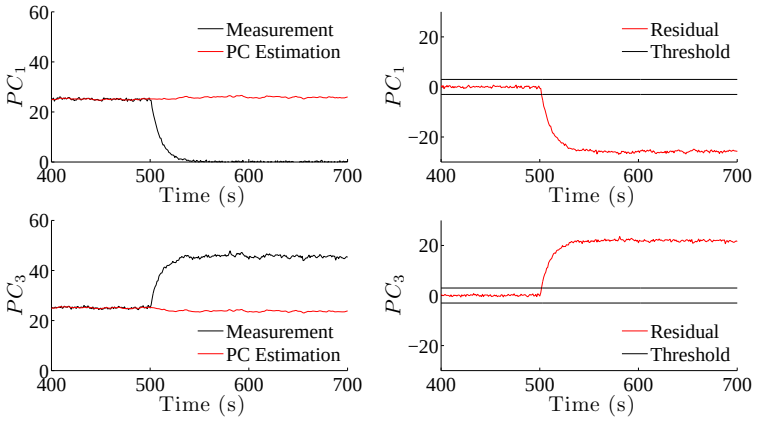


Figure 4.24: Measurements and estimations of HBG-PC1 and HBG-PC3, and their corresponding residuals, when an autonomous transition to OFF in SW_1 occurs.

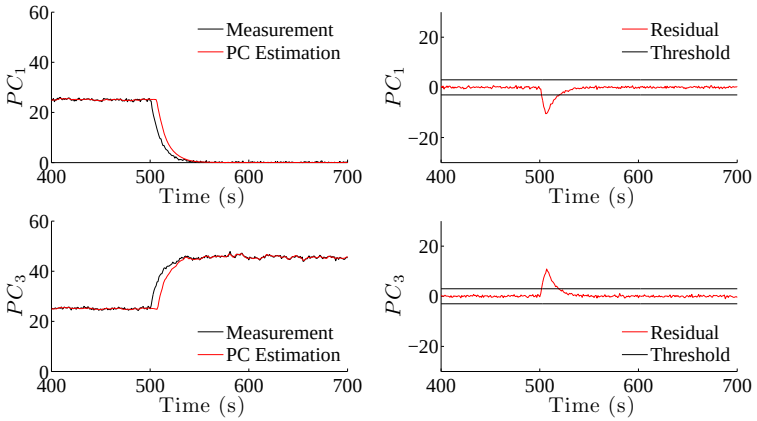


Figure 4.25: Measurements and estimations of HBG-PC1 and HBG-PC3, and their corresponding residuals, when an autonomous transition to OFF in SW_1 occurs and an autonomous transition to OFF in SW_1 is hypothesized.

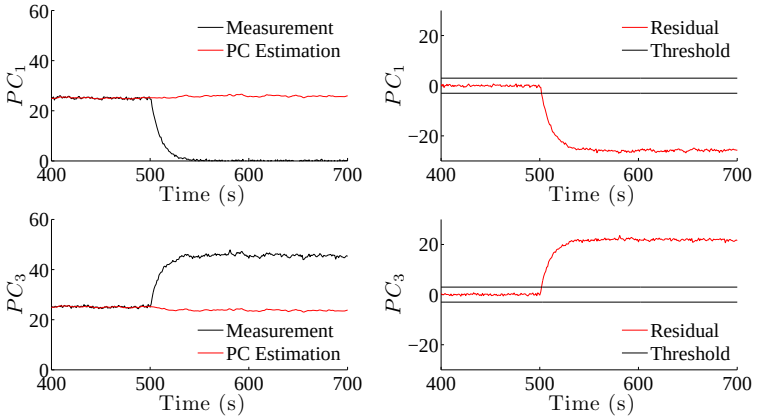


Figure 4.26: Measurements and estimations of HBG-PC1 and HBG-PC3, and their corresponding residuals, when an autonomous transition to OFF in SW_1 occurs and a stuck ON in SW_3 is hypothesized.

4.7.3 Parametric fault in R_{01}

The second scenario, illustrated in Figures 4.27, 4.28 and 4.29 presented in this chapter starts in the same initial configuration of the system with SW_1 and SW_3 set to *ON*, and SW_2 and SW_4 set to *OFF*. Tanks 1 and 3 fill in to the stationary level, and later, at time 500 s, a 20% blockage fault occurs in R_{01} (see Figure 4.27). Similarly to the previous example, at the next time step, both HBG-PC1 and HBG-PC3 residuals start to deviate, and at time 505 s the fault is detected. Some seconds later, at time 511 s a 0− signature is derived for HBG-PC1 residual, and a 0+ signature is derived for HBG-PC3 residual. Since the signatures obtained for this example are the same signatures obtained for the previous example, the fault isolation algorithm works similarly, assuming first that a discrete fault has occurred, and isolating autonomous transition to *OFF* in SW_1 and stuck *ON* in SW_3 as the fault candidates. The hybrid diagnosis framework creates two different instances of the HBG-PCs in the system, reassigns causality by running Hybrid SCAP for the mode transitions, and tracks the system for the empirically determined time interval of 10 s. The evolution of each fault candidate is shown in Figures 4.28 and 4.29, for SW_1 and SW_3 fault candidates, respectively. For this scenario, by looking at the figures, we see that none of the discrete fault candidates can be confirmed as the true fault in the system, as a result, the isolation algorithm discards a discrete fault in the system. Next step in the algorithm is to hypothesize parametric faults. Looking at Table 4.3, we see that the fault signatures obtained for HBG-PC1 and HBG-PC3 only match a fault in R_{01}, thus confirming R_{01} as the true fault in the system.

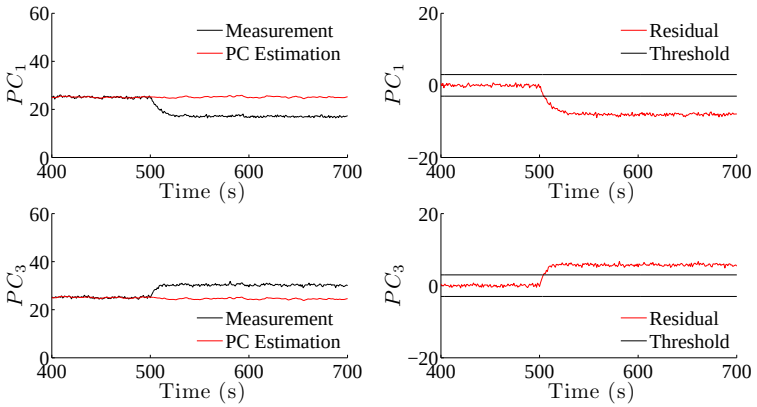


Figure 4.27: Measurements and estimations of HBG-PC1 and HBG-PC3, and their corresponding residuals, when a parametric fault in R_{01} occurs.

In scenarios where there are more than one parametric fault candidate the isolation process will calculate the set of fault candidates and the identification stage will be needed to confirm the actual fault in the system.

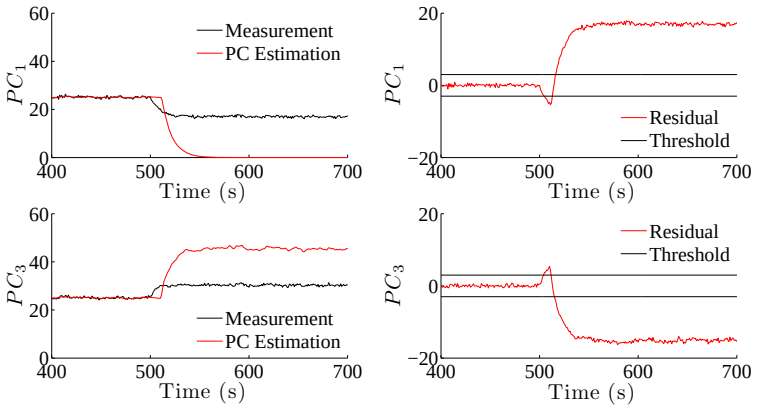


Figure 4.28: Measurements and estimations of HBG-PC1 and HBG-PC3, and their corresponding residuals, when a parametric fault in R_{01} occurs and an autonomous transition to OFF in SW_1 is hypothesized.

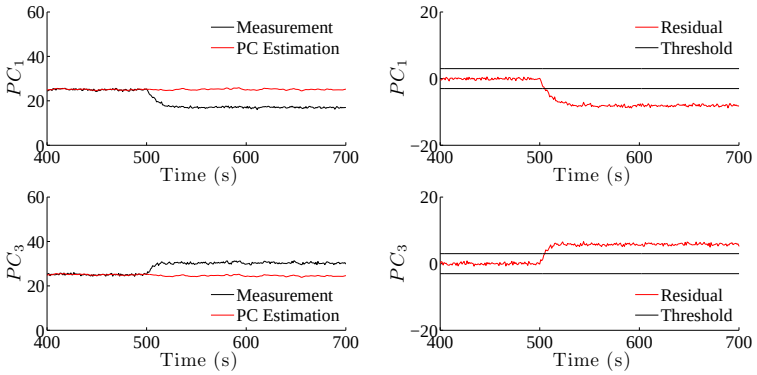


Figure 4.29: Measurements and estimations of HBG-PC1 and HBG-PC3, and their corresponding residuals, when a parametric fault in R_{01} occurs and a stuck ON in SW_3 is hypothesized.

4.8 Diagnosis Architecture using minimal DBNs and HBG-PCs

Dynamic Bayesian Networks (DBNs) are used to model, simulate and diagnose continuous systems. They can also have some discrete state variables or nodes to model discrete behaviour [41, 86]. Chapter 3 has introduced the convergence and high need of resources problems with DBNs. There are some solutions already explained to those problems: Approximate inference (PF algorithms), and using minimal DBNs instead of DBNs modeling the complete system, that is one of the contributions of this dissertation. Regarding hybrid systems, DBNs with continuous and discrete nodes modeling the whole system behaviour have even harder convergence problems.

Minimal DBNs have been presented in Chapter 3 as a useful tool for continuous systems fault diagnosis. They reduce the computational resources needed to obtain an accurate convergence that allows to perform fault detection, isolation and identification satisfactorily. This section proposes to derive minimal DBNs for hybrid systems and integrate them in the diagnosis architecture for discrete and parametric faults in hybrid systems presented previously in this Chapter.

HBG-PCs decompose the hybrid system behaviour in a collection of continuous behaviours managed by discrete events that trigger the change between them. This characteristic can be exploited to simplify the minimal DBNs for hybrid systems.

Figure 4.30 shows how minimal DBNs derived from HBG-PCs can be used to track the system behaviour without giving false positives when a mode change occurs. They can also perform fault detection (Fig. 4.30) and identification (Fig. 4.31) integrated in the diagnosis architecture previously presented in this Chapter.

The diagnosis of hybrid systems for discrete and parametric faults have already been presented. DBNs can be used in every diagnosis stage and they will be generated for each HBG-PC after every change in the working mode.

Regarding the identification stage there are two different faults to identify: 1) discrete faults, and 1) parametric faults. The discrete fault candidates identification consist of configuring the system model in the hypothetical mode and waiting for a period of time empirically fixed. If the residuals remain active, the hypothetical working mode is not the actual one, otherwise, the working mode of the system has been identified and the actual configuration is updated. This process will be done with minimal DBNs derived from HBG-PCs for each candidate working mode. The fault identification of parametric faults is the same as the stage presented for continuous systems (see Chapter 3): the nominal DBNs will be augmented with a node modeling the faulty parameter, where they will give the fault estimation. Next Chapter will present a complete case study where the minimal DBNs are used in diagnosis of a hybrid system. Meanwhile, the four-tank hybrid system in Figure 4.18 has been used to show the applicability of our proposal.

4.8.1 Four-tank system example

Simulated data has been generated with 5% level of noise, during 1000 s with a sample period of 0.1. We run several experiments with different mode configurations and different faults, varying the size and time of fault occurrence. Results for all these situations were equivalent to the example presented next.

Figure 4.30 shows the results obtained for one of the experiments run. First column (Figure 4.30) compares the three measurements and its estimation by the minimal DBNs, while second column shows the residual obtained for each minimal DBN. DBN4 has not been included in the figure as this HBG-PC is always deactivated during the experiment.

Initially, water tanks are empty, and start to fill at constant rate. Hence, the initial configuration of the system is SW_1 and SW_3 set to *ON*, and SW_2 and SW_4 set to *OFF*. Tanks 1 and 3 start to fill, and approximately at instant 500 sampling periods both tanks reach stationary state. At this time, the level in tank 1, h_{T1}, and the level in tank 3, h_{T3}, are lower than the height of the connecting pipes, h, and consequently, there is no flow through the connecting pipes.

At instant 2000 sampling steps, controlled junction SW_3 is set to *OFF*, so the system mode changes. Simultaneously, HBG-PC1 and HBG-PC3, which contain SW_3, change their modes, and quickly reassign causality by running Hybrid SCAP [17]. Once the new HBG-PCs have been generated, the corresponding DBNs are built. As shown in Figure 4.30, DBN1 and DBN3 are able to correctly estimate the level of tank 1 and 3, respectivley, immediately after the mode change. Regarding HBG-PC2 and HBG-PC4, since both

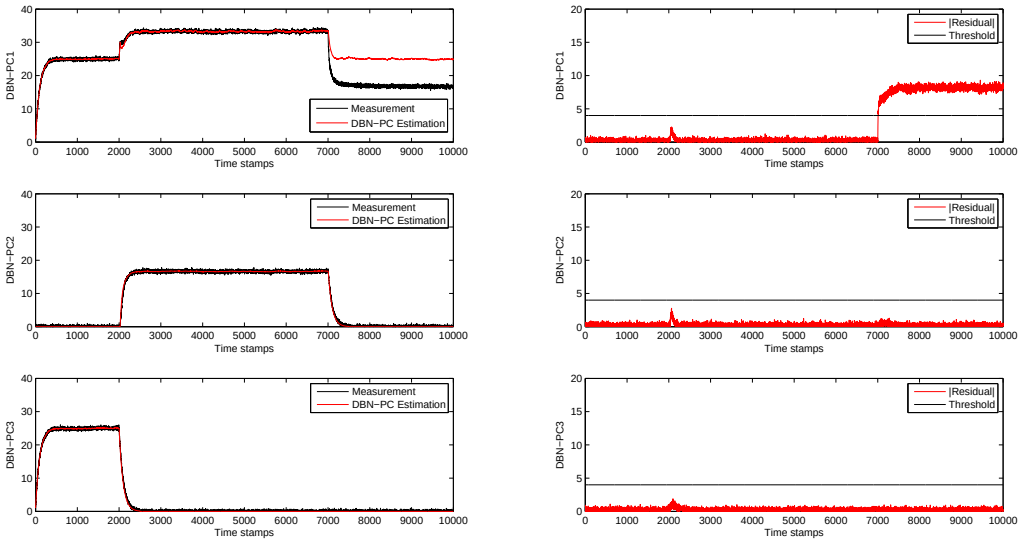


Figure 4.30: Tracking the four-tank hybrid system behaviour using minimal DBNs. The graphs on the left show the three HBG-PCs estimations corresponding to the pressure of tank 1 to 3, respectively (HBG-PC4 is not shown because tank 4 is always empty and is not connected to the rest of the system during the experiment). Graphs on the right present the residuals of each HBG-PC estimation. Residual is only activated after a fault occurs.

HBG-PCs do not contain the switching junction SW_3, none of them is affected by the mode change so their DBNs do not need to be generated and continue tracking the system behaviour.

SW_3 has been set to *OFF*, so the level of tank 3 decreases until it becomes zero, while the level of tank 1 increases. At instant 2100 sampling periods, the level of tank 1 reaches the height of the connecting pipe between tanks 1 and 2. At this point, the autonomous junction SW_2 transitions to *ON* mode and water begins to fill tank 2. HBG-PC1 and HBG-PC2 are affected by this mode change. In both cases, the models of the HBG-PCs are updated and the DBNs are generated. Both of them are able to correctly estimate the measurements for the new mode.

At instant 7000 sampling periods a 20% leak in tank 1 occurs. As a consequence, the level of tank 1 decreases, while the estimation of DBN1 does not. Hence, residual of DBN1, which is the only one containing C_1 as a fault candidate, activates, triggering the fault isolation procedure. Regarding DBN2, since the level of tank 1 decreases due to the fault, at instant 7050 sampling periods the autonomous junction SW_2 transitions again to *OFF* mode, and HBG-PC2 changes mode again. The HBG-PC model is updated immediately and DBN2 is built; it is able to correctly estimate the level of tank 2 for the new mode.

Nine sampling periods after fault inyection (0.9 seconds) DBN1 detects a fault. After the discrete fault isolation and identification procedure and according to FSM in Table 4.5, the set of fault candidates is $\{C_1^+, R_1^+, R_{03}^+\}$.

DBN1 was extended with a node for the faulty parameter which needs to be identified as explained in [5]. In this scenario, three DBNs have been built, one for each fault candidate. Figure 4.31 shows the results obtained using the DBN1 to estimate C_1. The DBN is able to track the system behaviour and to obtain an estimation for the parameter quickly converging to a 19.3% fault in C_1. Minimal DBNs built to estimate R_1 and R_{03} were not able to converge. Hence, the candidates were discarded.

	$HBG-PC1$	$HBG-PC2$	$HBG-PC3$	$HBG-PC4$
C_1^+	$0+$			
C_2^+		$0+$		
C_3^+			$0+$	
C_4^+				$0+$
R_{01}^+	$0-$		$0+$	
R_{03}^+	$0+$		$0-$	
R_1^+	$0+$			
R_2^+		$0+$		
R_3^+			$0+$	
R_4^+				$0+$
R_{12}^+	$0-$	$0-$		
R_{34}^+			$0+$	$0-$

Table 4.5: Reduced Qualitative Fault Signature Matrix.

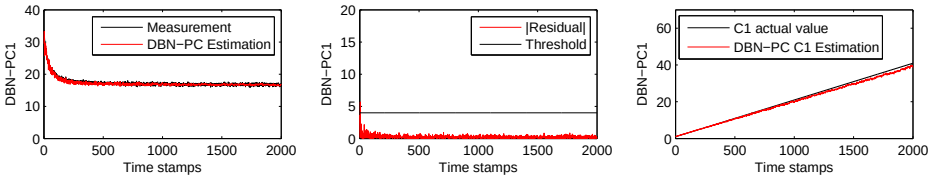


Figure 4.31: Identifiying an incipient or progressive fault in tank 1 capacitance, C_1. First graph in the left shows the DBN1 output, the pressure in tank 1, the second graph presents the residual of DBN1 and the third one shows the estimation for the parameter C_1.

4.9 Extension for multiposition actuators

Hybrid Bond Graphs' switching junctions model ON/OFF switches but there are hybrid systems which have some kind of multiposition actuator. The first challenge in this proposal is to model a multiposition valve or any other discrete actuator that connects one and only one of multiple paths at the same time to be able to use HBGs to explote its advantages regarding reconfiguration efficiency. The approach presented in this dissertation assumes that any switch which connects one "input path" with one and only one "output path" among several paths can be modeled as a group of ON/OFF switches connected in the appropiate way. When one of those multiposition switches has a position in the activated state it connects something in the system at the same time it disconnects any other parts related to its other positions.

Based on that idea the multiposition switch can be seen as a zero junction in an HBG model connected to the same number of one switching junctions as alternative paths the physical model has. There are a control specification (CSPEC) for each switching juntion which models the command to activate or not the switching junction. Only one CSPEC is going to be ON at any same time. Figure 4.32 shows a generalization of the previous explanation (a) and a generic HBG schematic of that structure (b).

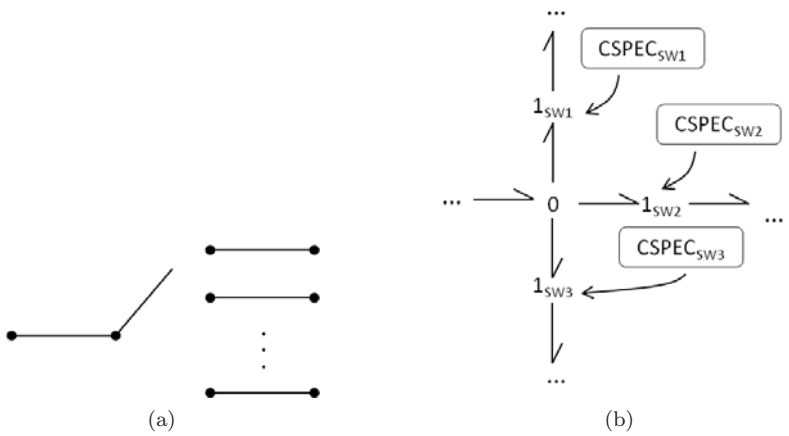
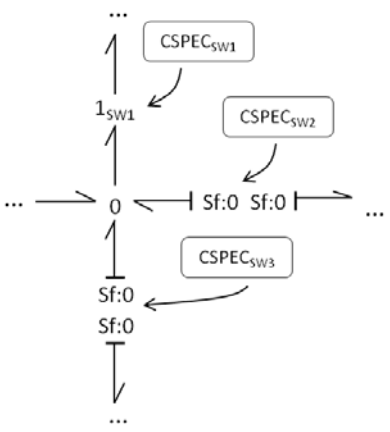
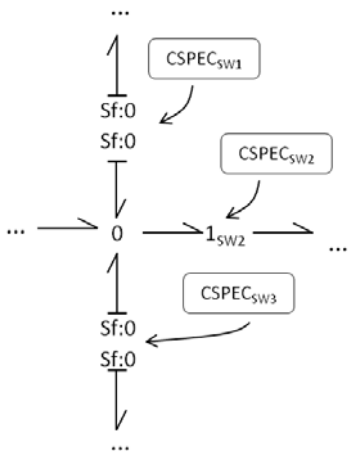


Figure 4.32: a) General structure of a multiposition switch. b) HBG generic schematic of a 3-position switch.

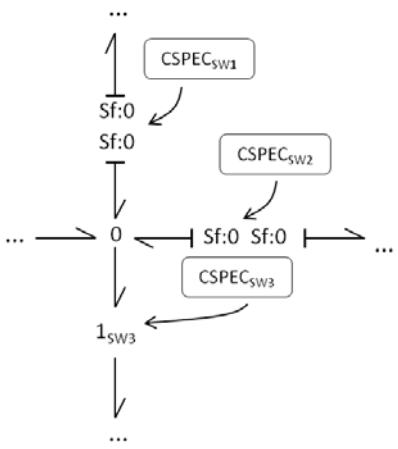
Looking at Figure 4.32.a) and the previous explanation, the main characteristic of the multiposition valves or discrete actuators considered in this dissertation is that they connect one and only one path among several at any time. This can be seen in the control signals of the actuator. Assuming 0 means OFF and 1 means ON, there is going to be only one path ON (1) at each time. In a three position valve, this means there are three possible control signals assuming non faulty behaviour: 100, 010 and 001. Figure 4.33 shows the three configurations previously described (a)100; b)010; c)001). It can be possible to have all paths disconnected (000, in the three position valve example) but this position depends on the actuator itself.



(a) CSPEC1 = *ON*, CSPEC2 = *OFF* and CSPEC3 = *OFF*



(b) CSPEC1 = *OFF*, CSPEC2 = *ON* and CSPEC3 = *OFF*



(c) CSPEC1 = *OFF*, CSPEC2 = *OFF* and CSPEC3 = *ON*

Figure 4.33: Configuration modes of a three position switch.

4.10 Discussion and conclusions

This Chapter has presented the Possible Conflicts' extension for hybrid systems (HBG-PCs). It has been introduced using the Bond Graph modeling tool and they have been successfully applied for a simulation four-tank system. The Hybrid Bond Graph (HBG) approach can model only ON/OFF actuators, that limits the use of this proposal. The chapter also proposed a method to model multiposition actuators using the ON/OFF tools of the HBGs.

To formalize the HBG-PCs some definitions, general properties and propositions have been presented and proven. This Chapter formally states that assuming all switching junctions are ON does not lead to lose HBG-PCs, i.e. the set of HBG-PCs in the hybrid system at any given time can be directly obtained from this original set of HBG-PCs providing the necessary configuration in terms of control specification commands.

There is an important advantage derived from the HBGs, the configuration modes do not need to be previously enumerated, the model is configured as the mode changes, i.e, if one mode is never visited, it will never be built.

The HBG-PCs divide the system in minimal redundant subsystems, some of them have common elements, but they model different parts of the original system. This can be exploited regarding the fault diagnosis efficiency, a mode change will not affect all the HBG-PCs so, while the HBG-PCs affected by the change are reconfigured, the others can continue tracking the system behaviour.

The HBG-PCs have been integrated in a diagnosis architecture for hybrid systems. This architecture allows performing fault detection and isolation of discrete and parametric faults in a unified way. In our approach, we do not measure the control variable in the actuator so we have to hypothesize the inputs.

Moreover, minimal DBNs have been also introduced in the diagnosis architecture to extend it for fault identification tasks.

Finally, it is possible to generate specific algorithms to derive quickly the set of ODEs (Ordinary Differential Equations) required to model the continuous behaviour of each working mode using the HBGs and HBG-PCs concepts [80], this allows to generate the minimal DBNs quickly after a mode change.

The method proposed to derive the HBG-PCs from the HBG of the complete system imposes a valid causal assignment in the HBG considering all actuators are ON. This cannot make sense in some systems, where there are alternative paths which are not going to be ON at the same time, but this does not mean their HBG model does not have a causal assignment with that assumption, i.e, the ROS (the system presented in next Chapter) has mutually exclusive paths and it actually has a valid causal assignment considering all switching junctions are ON. Moreover, as it will be presented later, the future work is directed to eliminate the assumption and derive the HBG-PCs even when there is not a valid causal assignment in the HBG with all the actuators ON.

There is also an important assumption in this work, the working mode before a fault occurs is known, but we keep tracking the system with either simulation or some kind of observer (particle filter). As a consequence, we track both, the continuous behaviour and the discrete behaviour (due to commanded changes or as a result of autonomous transitions).

Chapter 5

Case Study. Reverse Osmosis System (ROS)

Last Chapter has introduced Hybrid Possible Conflicts (HBG-PCs). It has also explained a fault diagnosis framework for hybrid systems using the HBG-PCs. This Chapter shows how that framework is applied to a real-world system, a Reverse Osmosis System (ROS). The ROS is a subsystem of an Advanced Water Recovery System (AWRS) which receives water with impurities and by means of a membrane filter cleans the inorganic and particulate matter from the water.

5.1 Reverse Osmosis System (ROS). Introduction

The Advanced Water Recovery System (AWRS) has been designed and built at the NASA Jonhson Space Center (JSC) as part of the Advanced Life Support (ALS) System for long duration manned missions [90, 104]. Figure 5.1 shows the schematic of the AWRS, it works in microgravity conditions and it obtains potable water from wastewater.

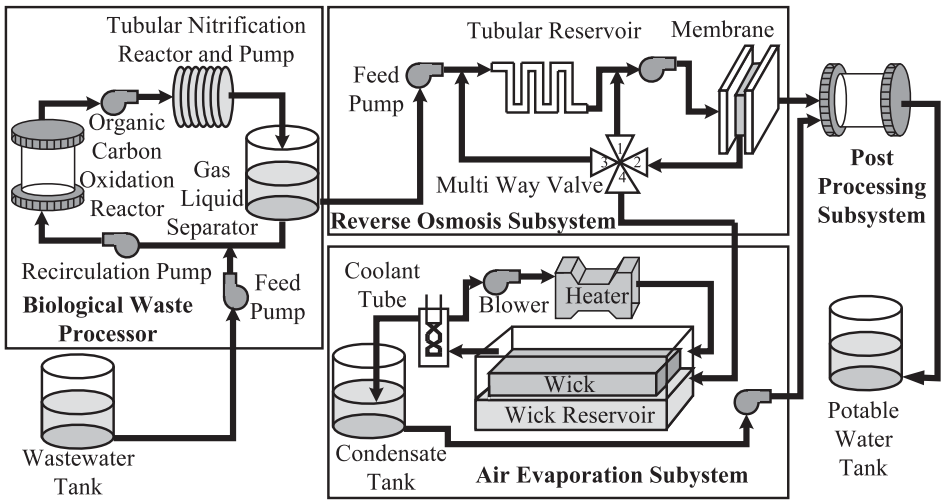


Figure 5.1: Schematic of the Advanced Water Recovery System.

The AWRS has four subsystems: 1) the Biological Waste Processor (BWP), which removes organic matter and ammonia from the wastewater; 2) the Reverse Osmosis System (ROS), which removes inorganic and particulate matter by means of a high pressure membrane filtration system; 3) the Air Evaporation Subsystem (AES), which recovers the remaining water in the ROS' brine using an evaporation and condensation process; 4) the Post Processing Subsystem (PPS) which removes trace impurities and generates potable water from combined output from the ROS and the AES. This dissertation focuses on the ROS subsystem.

The ROS receives input flow from the BWP subsystem, the feed pump is always ON, so this flow is continuous during all the working cycle. This flow mixed with the recirculated flow, when the recirculation pump is ON, goes through the tubular reservoir. After it, there is a membrane responsible of filtering the flow to remove impurities. The filtered water goes to the Post Processing Subsystem (PPS) and some of the flow, which is not clean enough goes to a multiposition valve. Depending on the working mode, the valve will be in position 1, 2 or purge. Position 1 recirculates flow in the long path (before the tubular reservoir), position 2 recirculates flow in the short path (after the tubular reservoir) and position purge takes the wastewater accumulated in the membrane to the AES. There is a fourth working mode, called slough, used to clean the membrane. In this mode, the water is recirculated counter-clockwise in the smaller loop. The slough mode occurs periodically but it is not part of the regular working cycle of the ROS. The work in this dissertation is focused on the working cycle made of the working mode 1 (the valve is in position 1), followed by the working mode 2 (the valve is in position 2) and finally, the working mode purge (the valve is in position purge), and starting again.

The ROS has a multiposition valve which is the responsible for the path followed by the dirty flow coming out of the membrane. Chapter 4 have presented a method to model a multiposition actuator using ON/OFF switching junctions.

Figure 5.2 shows the HBG model of the ROS. It is divided into subsystems to easily understand the different parts of the system: Pumps, pipe system, rotation to hydraulic (R2H) and the conductivity model. Control signals $M1$, $M2$ and P model the changes of the multiposition valve, and consequently, determines the current operation mode in the ROS. The multiposition valve is modeled in the HBG in Figure 5.2 in the highlighted area in the pipe system. The three control signals are mutually exclusive, i.e. only one can be *ON* at any time. As can be seen in Figure 5.2 the conductivity subsystem model is not connected to the remaining subsystems; they only share the control signals. The conductivity subsystem is less representative and less error prone, then, we have focused on the big subsystems below.

During the working mode 1 control signal $M1$ is equal to 1 ($M2$=0 and P=0), in this mode, the recirculation pump is ON and the connected path is the one with resistance R_{brine1}. Regarding the conductivity model, the active path is the one with the source Sf_primary. For working mode 2, $M2$=1 ($M1$=0 and P=0), the recirculation pump is also ON but in this case, the path connected is the one with the parameter R_{brine2}, for the conductivity model, Sf_secondary is the connected source. The third working mode, mode purge, has signal P=1 ($M1$=0 and $M2$=0); in this mode, the recirculating pump is OFF and this is modeled disconnecting the source (Se_recircpump) and the resistance of the membrane (R_{memb}), the resistance R_{drain} appears in the pipe system as well as R and R_{purgeB} in the conductivity model.

There are five measurements in the system: 1) The outflow of the feed pump, which is the input flow in the system (F_FP) , 2) the pressure in the tubular reservoir (P_Back), 3) the pressure in the membrane (P_Memb), 4) the pressure out of the recirculating pump (P_Pump) and 5) the concentration of the liquid (P_k). Figure 5.3 shows the system measurements during three complete working cycles (each cycle is made of mode 1, mode 2 and mode purge). The data has 2% noise. They are drawn from the top to the bottom of the figure in the same order they have been explained before. The input flow in the system (F_FP) is almost continuous across time but the other four measurements have a different behaviour during each working mode. Looking at the third measurement in Figure 5.3, the pressure in the membrane (P_Memb), we can identify the three working modes, during the first and the second modes, the measurement is increasing, at different rates on each of them, while in the third mode, the purge mode, the pressure of the membrane drops drastically and it remains steady during the whole mode. The concentration of the liquid (Pk) is shown at the bottom of the graph, in this measurement we can better identify the three working modes, it has another important characteristic, it is the variable used by the controller to fix the control signals.

The values of the main parameters in the system are presented on Table 5.1. The system is described by three sets of equations, one for each working mode; the equations are presented in Appendix A.

The system can be modeled using three DBNs, one for each working mode. The DBN modeling the system configuration in mode 1 is shown in Figure 5.4.

Parameter	Description	Nominal value
C_{Memb}	Capacitance of the membrane	$2.611 \cdot 10^{-10}$
C_{Res}	Capacitance of the tubular reservoir	$4.3511 \cdot 10^{-9}$
C_k	Capacitance in the conductivity model	$1.695 \cdot 10^{5}$
GY	Element to transform the energy of the recirculation pump	1
TF	Element to transform the energy of the feed pump	1
I_{fpump}	Efficiency in the feed pump	$2.4821 \cdot 10^{12}$
I_{pump}	Efficiency in the recirculation pump	9600
R_{fpump}	Resistance in the feed pump	$2.835 \cdot 10^{10}$
R_{bwp}	Resistance in the pipe from the BWP subsystem	$2.283 \cdot 10^{9}$
R_{pipe}	Resistance in the pipe carrying water to the membrane	$2.8544 \cdot 10^{13}$
R_{Memb}	Resistance in the membrane	$2.46 \cdot 10^{12}$
R_{brine1}	Resistance in the pipe carrying water from the valve in the long loop	$9.1011 \cdot 10^{12}$
R_{brine2}	Resistance in the pipe carrying water from the valve in the short loop	$9.1011 \cdot 10^{12}$
R_{drain}	Resistance in the pipe carrying water from the valve to the AES	$4.1362 \cdot 10^{9}$
R_{rpump}	Resistance in the recirculation pump	2

Table 5.1: Parameters in the Reverse Osmosis System.

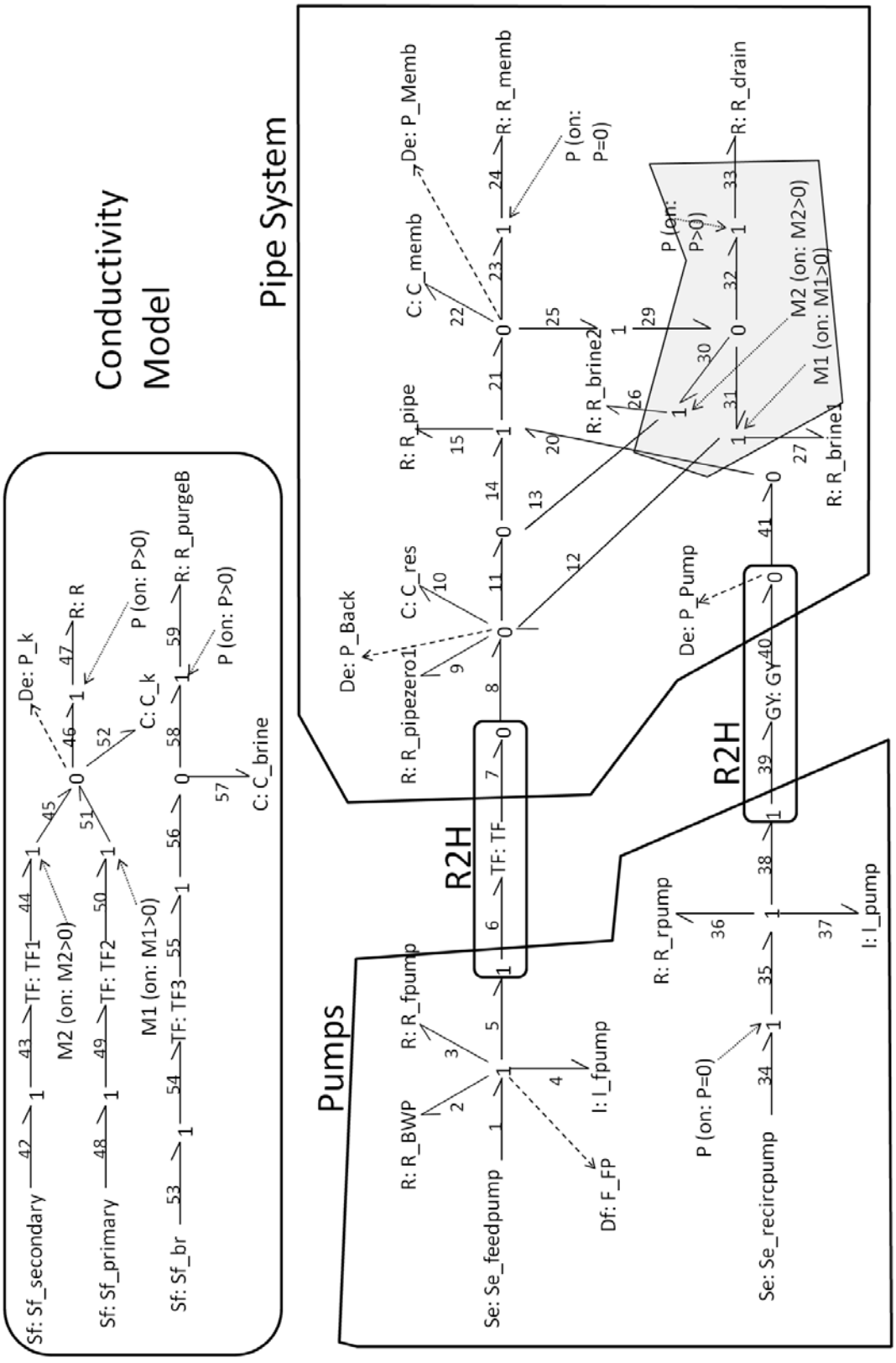


Figure 5.2: Hybrid Bond Graph Model of the Reverse Osmosis System (ROS).

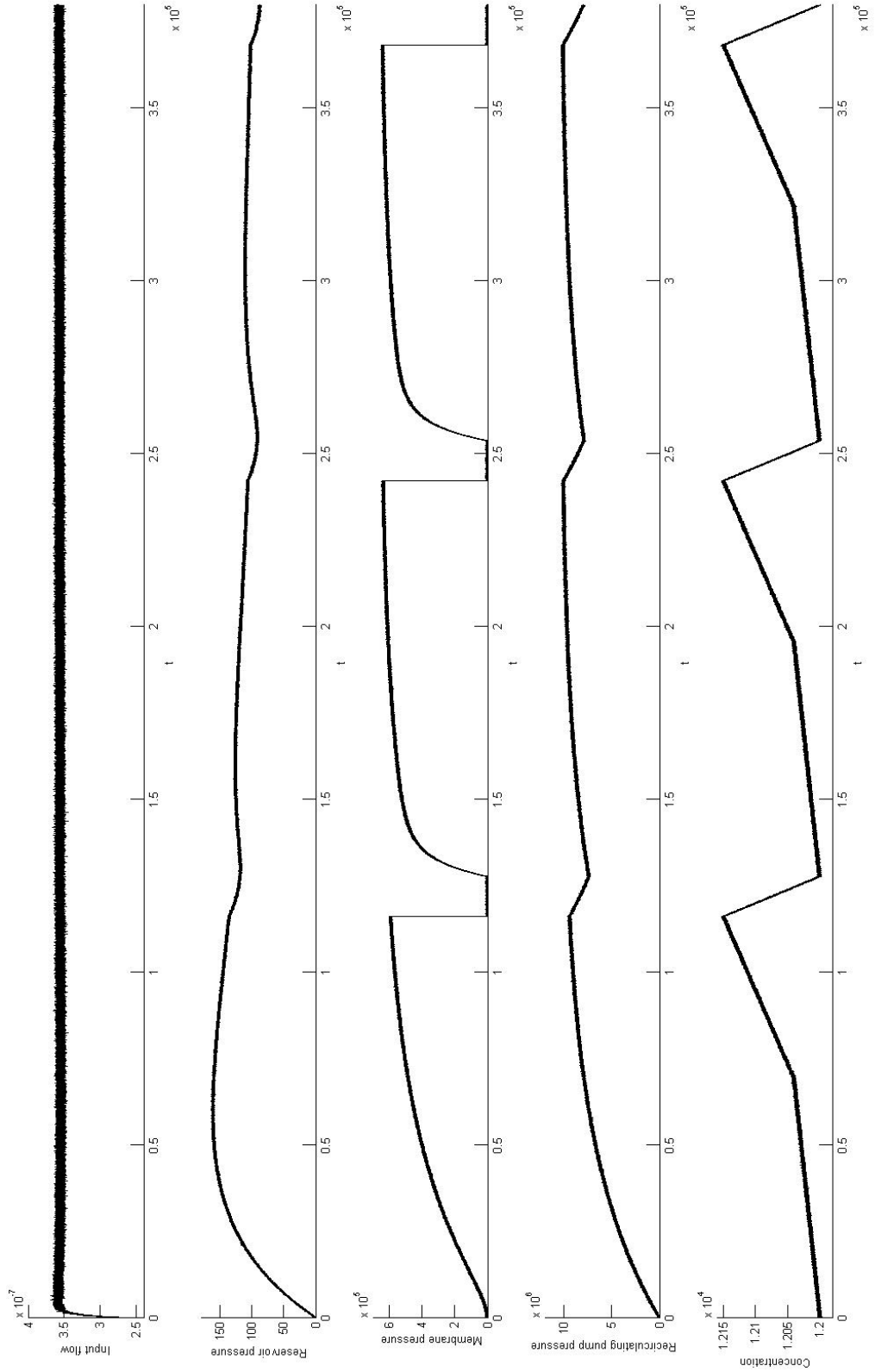


Figure 5.3: Nominal behaviour of the ROS during three complete working cycles for available measurements: F_FP, P_Back, P_memb, P_Pump, and P_k, from top to down in the figure.

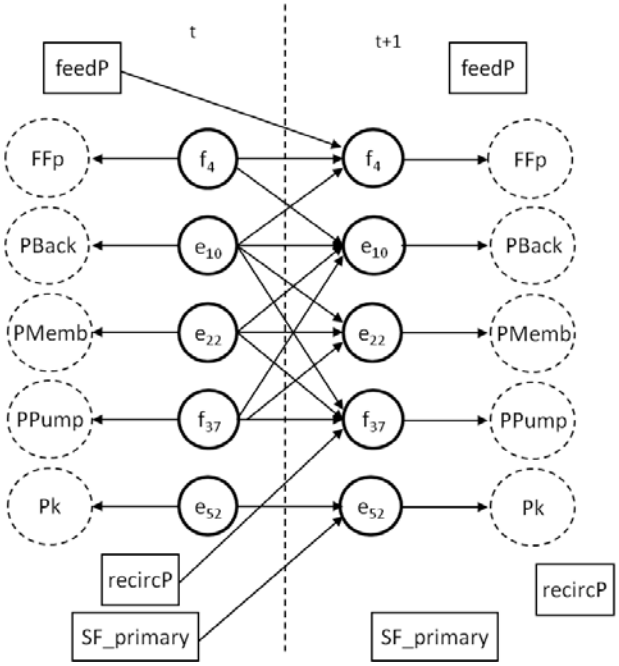


Figure 5.4: DBN modeling the Reverse Osmosis System (ROS) in mode 1.

5.2 Hybrid Possible Conflicts and Dynamic Bayesian Networks

The ROS HBG model has causal assignment considering all switching junctions are set to ON. Because of that, its Hybrid Possible Conflicts (HBG-PCs) have been derived efficiently using the HBG of the system presented in Figure 5.2 and the method presented in Chapter 4. There are five HBG-PCs, each one of them has one of the system measurements as discrepancy node. The HBG-PCs derived following the method described in Chapter 4 are presented from Figure 5.5 to Figure 5.9.

Each HBG-PC models a redundant subsystem of the physical system. HBG-PC1 (Figure 5.5) models the subsystem from the feedpump to the sensor in the reservoir (PBack).

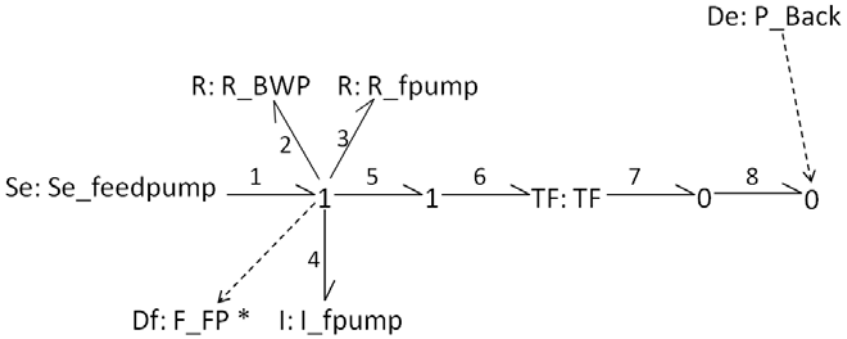


Figure 5.5: Hybrid Bond Graph Model of HBG-PC1 from the Reverse Osmosis System (ROS). The discrepancy node, estimated variable, is F_FP and the state variable is f_4, that is related to the generalized effort in I_fpump.

HBG-PC2 (Figure 5.6) represents the subsystem modeling the reservoir and the two recirculating paths.

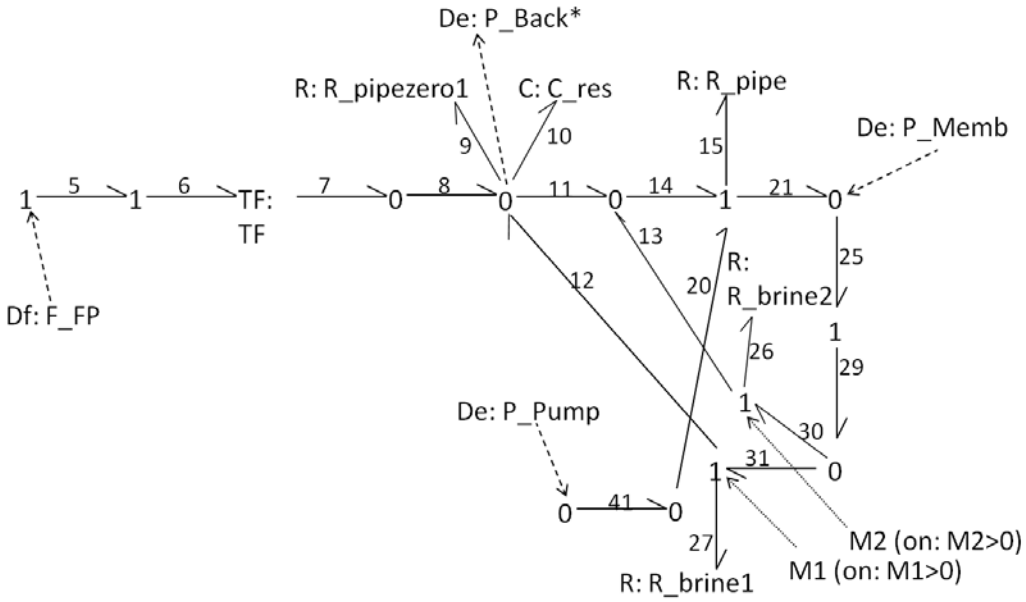


Figure 5.6: Hybrid Bond Graph Model of HBG-PC2 from the Reverse Osmosis System (ROS). The discrepancy node, estimated variable, is P_Back and the state variable is e_{10}, that is related to the capacitance in the reservoir, C_res.

HBG-PC3 (Figure 5.7) models the membrane subsystem.

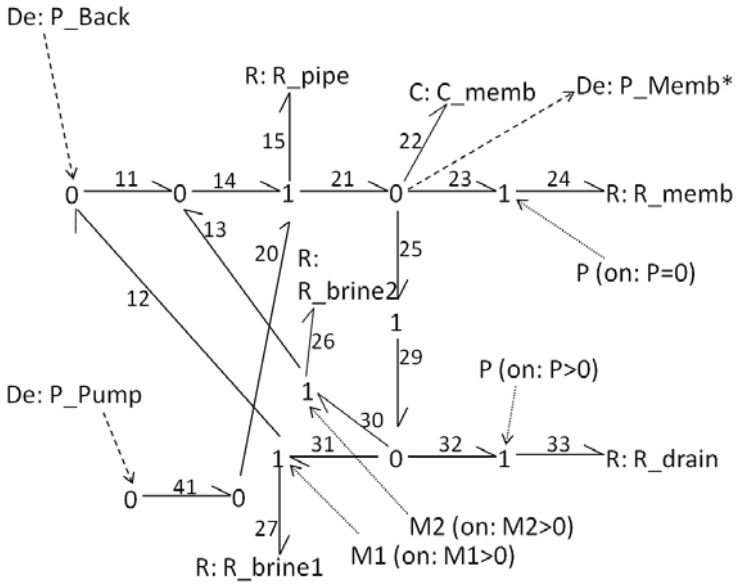


Figure 5.7: Hybrid Bond Graph Model of HBG-PC3 from the Reverse Osmosis System (ROS). The discrepancy node, estimated variable, is P_Memb and the state variable is e_{22}, that is related to the capacitance in the membrane C_memb.

HBG-PC4 (Figure 5.8) models the recirculating pump.

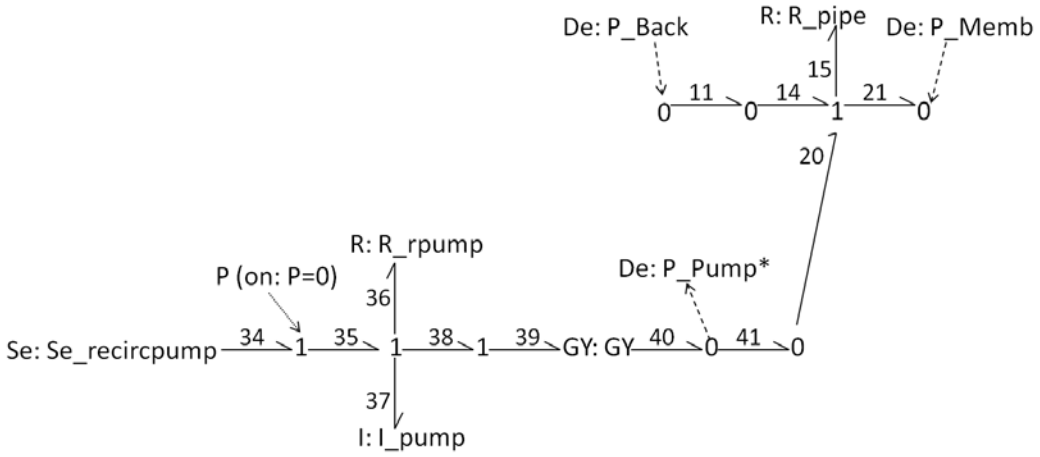


Figure 5.8: Hybrid Bond Graph Model of HBG-PC4 from the Reverse Osmosis System (ROS). The discrepancy node, estimated variable, is P_Pump and the state variable is f_{37}, that is related to the generalized effort in I_pump.

HBG-PC5 (Figure 5.9) represents the conductivity model of the system.

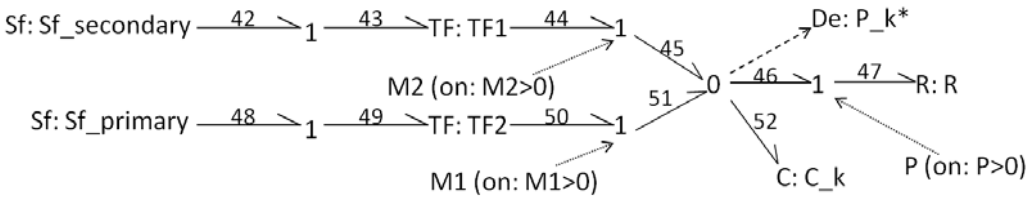


Figure 5.9: Hybrid Bond Graph Model of HBG-PC5 from the Reverse Osmosis System (ROS). The discrepancy node, estimated variable, is P_k and the state variable is e_{52}, that is related to the concentration in conductivity model C_k.

The previous paragraphs present a general idea of the subsytem modeled by each HBG-PC but there is some information we can obtain just analyzing the HBG model of each HBG-PC. The system configuration and the sensors placement contribute to the fact that all the HBG-PCs in this system have just one state variable. This shows very little overlaping between the HBG-PCs and, of course, between the subsystems they represent.

Table 5.2 presents a summary of the state variables, inputs, outputs, parameters and actuators included in each HBG-PC. The actuators in the system are the multiposition valve and the recirculating pump, as it has been explained, the valve has three positions: 1, 2 and purge, corresponding to the control signals M1, M2 and P in mode ON, respectively. The column *actuators* shows the actuators and the positions that influence each HBG-PC, in case there is not any control signal, the HBG-PC is affected during the three working mode.

	State variables	**Inputs**	**Output**	**Actuators**	**Param.**
HBG-PC1	f_4	$Se_{feedpump}$, $PBack$	FFP	-	R_{BWP}, R_{fpump}
HBG-PC2	e_{10}	FFP, $PMemb$, $PPump$	$PBack$	Valve (M1,M2)	$R_{pipezero1}$, C_{res}, R_{pipe}, R_{brine1}, R_{brine2}
HBG-PC3	e_{22}	$PBack$, $PPump$	$PMemb$	Valve	R_{pipe}, C_{memb}, R_{memb}, R_{drain}, R_{brine1}, R_{brine2}
HBG-PC4	f_{37}	$PBack$, $PMemb$, $Se_{recircpump}$	$PPump$	Recirculation pump	R_{pipe}, R_{rpump}, I_{pump}
HBG-PC5	e_{52}	$Sf_{primary}$, $Sf_{secondary}$	Pk	Valve	R, C_k

Table 5.2: Summary of the state variables, inputs, outputs, actuators and parameters included in each HBG-PC.

All of those HBG-PCs have a causal assignment in the three working modes (mode 1, mode 2 and mode purge). There is a particular case for HBG-PC4 in mode purge. All the experiments in this dissertation have integral causality to calculate the state variables. In this particular case, there is only causal assignment considering derivative causality in the element *Ipump*. Because of that, there is not going to be one equation

with integral causality to calculate state variable f_{37} for mode purge. The change in causality in *Ipump* affects to the DBN4 as well as the DBN of the complete system. DBN4 does not exist in mode purge (f_{37} is the only state variable in DBN4) and DBN of the complete system does not have the equation to estimate the value for the state variable f_{37} and for the observation *Ppump* during the mode purge.

The Hybrid Fault Signature Matrix (HFSM) and the (parametric) Fault Signature Matrix (FSM) for each working mode show the HBG-PCs affected by the switching junctions and the system parameters, respectively. Those matrices are shown in Tables from 5.3 to 5.6 and they will be used for diagnosis tasks. The discrete faults have been grouped by the element in the system that can actually fail: the recirculation pump and the multiposition valve. We have previously explained the relation between the control signals and the actuators.

	HBG-PC1	**HBG-PC2**	**HBG-PC3**	**HBG-PC4**	**HBG-PC5**
Recirc. Pump			1	1	
Valve		1	1		1

Table 5.3: Hybrid Fault Signature Matrix of the ROS.

	HBG-PC1	**HBG-PC2**	**HBG-PC3**	**HBG-PC4**	**HBG-PC5**
R_{bwp}	1				
R_{fpump}	1				
I_{fpump}	1				
R_{rpump}				1	
I_{pump}				1	
$R_{pipezero1}$		1			
C_{res}		1			
R_{pipe}		1	1	1	
C_{memb}			1		
R_{memb}			1		
R_{brine1}		1	1		
C_k					1

Table 5.4: Fault Signature Matrix of the ROS for Mode 1.

	HBG-PC1	HBG-PC2	HBG-PC3	HBG-PC4	HBG-PC5
R_{bwp}	1				
R_{fpump}	1				
I_{fpump}	1				
R_{rpump}				1	
I_{pump}				1	
$R_{pipezero1}$		1			
C_{res}		1			
R_{pipe}		1	1	1	
C_{memb}			1		
R_{memb}			1		
R_{brine2}		1	1		
C_k					1

Table 5.5: Fault Signature Matrix of the ROS for Mode 2.

	HBG-PC1	HBG-PC2	HBG-PC3	HBG-PC4	HBG-PC5
R_{bwp}	1				
R_{fpump}	1				
I_{fpump}	1				
R_{rpump}				1	
$R_{pipezero1}$		1			
C_{res}		1			
R_{pipe}		1	1	1	
C_{memb}			1		
R_{drain}			1		
C_k					1
R					1

Table 5.6: Fault Signature Matrix of the ROS for Mode Purge.

DBNs have been chosen as the estimation model for fault detection, isolation and identification, as it has been explained in Chapters 3 and 4. DBNs have been derived from the HBG-PCs (see Figures from 5.5 to 5.9) using the method presented in Chapter 3. There is a minimal DBN for each HBG-PC and for each working mode. When a mode change occurs, it automatically changes the minimal DBNs of the HBG-PCs that contain the switching junction to the minimal DBNs of the new working mode. Some of those DBNs are presented in Figures from 5.10 to 5.14, they correspond to the system working on mode 1.

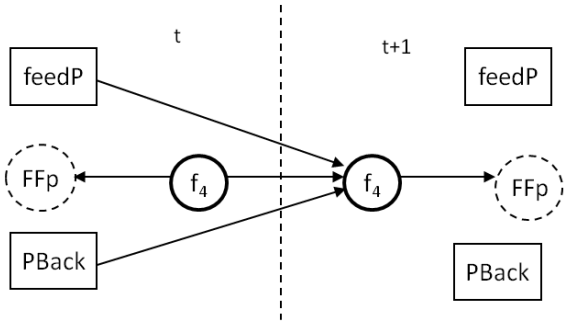


Figure 5.10: Dynamic Bayesian Network of HBG-PC1 (DBN1) from the Reverse Osmosis System (ROS) for the working mode 1. The state variable is f_4 (flow in the feed pump), related to the parameter I_fpump in the BG model.

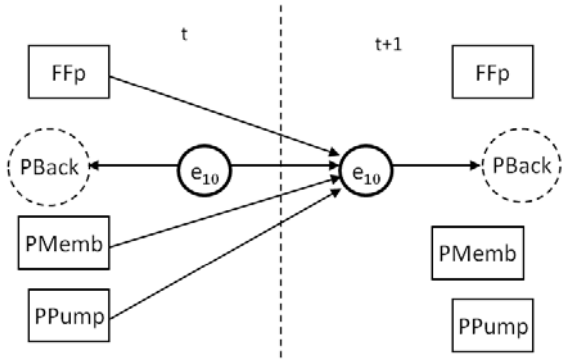


Figure 5.11: Dynamic Bayesian Network of HBG-PC2 (DBN2) from the Reverse Osmosis System (ROS) for the working mode 1. The state variable is e_{10} (pressure in the reservoir), related to the parameter C_res in the BG model.

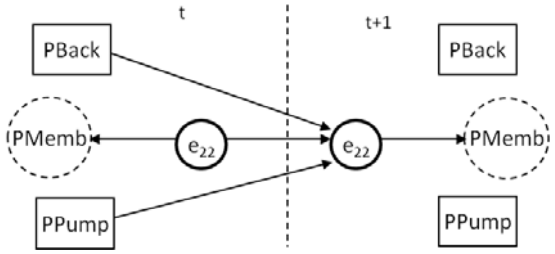


Figure 5.12: Dynamic Bayesian Network of HBG-PC3 (DBN3) from the Reverse Osmosis System (ROS) for the working mode 1. The state variable is e_{22} (pressure in the membrane), related to the parameter C_memb in the BG model.

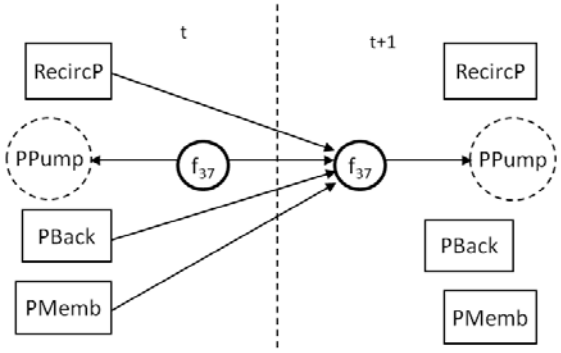


Figure 5.13: Dynamic Bayesian Network of HBG-PC4 (DBN4) from the Reverse Osmosis System (ROS) for the working mode 1. The state variable is f_{37} (flow in the recirculation pump), related to the parameter I_pump in the BG model.

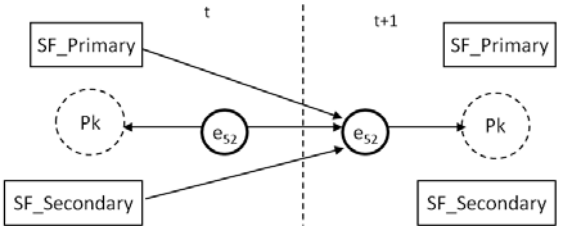


Figure 5.14: Dynamic Bayesian Network of HBG-PC5 (DBN5) from the Reverse Osmosis System (ROS) for the working mode 1. The state variable is e_{52} (concentration in the conductivity model), related to the parameter C_k in the BG model.

5.3 Tracking nominal behaviour

The first objective to achieve with the minimal DBNs derived from HBG-PCs is tracking the nominal behaviour of the system. Once this is tested, the fault diagnosis process can be done using the minimal DBNs derived from the HBG-PCs, this will be explained in next section.

The simulated behaviour, as presented in Figure 5.3, was used as input for FDI to both the DBN modeling the complete system and the minimal DBN for each HBG-PC. A Ztest [13, 53] has been used to check that all those DBNs are able to track the system behaviour even during mode changes and no false positives have been obtained. Figures from 5.15 to 5.17 present how some minimal DBNs track the system behaviour.

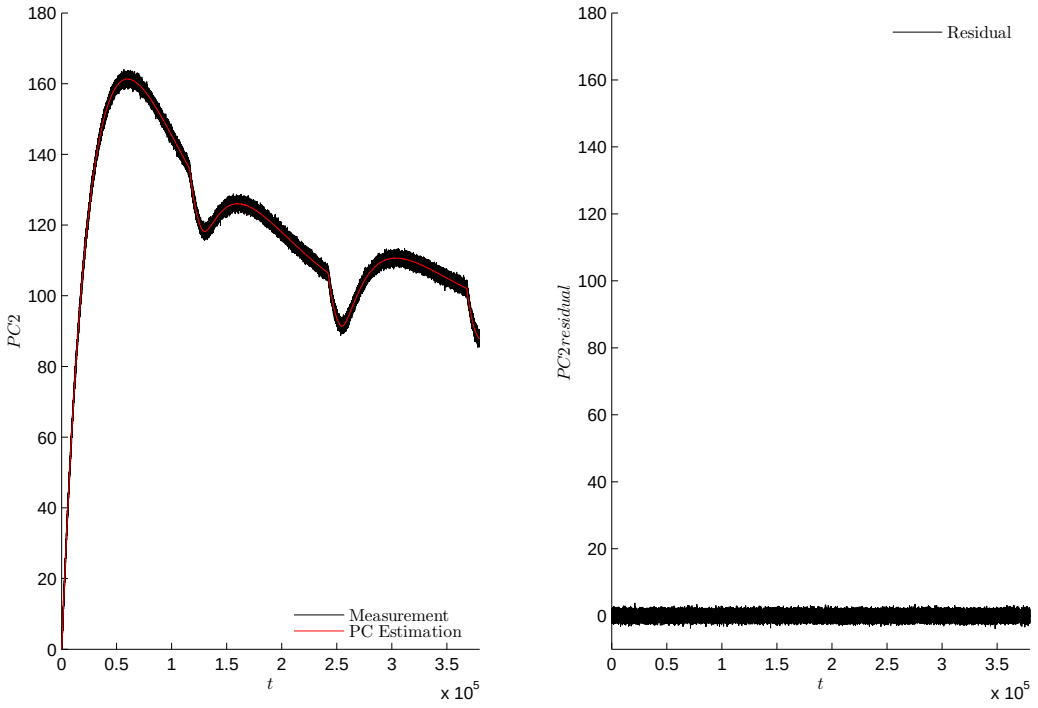


Figure 5.15: DBN2 estimation compared to the ROS measurement along three working cycles. The system measurement is the pressure in the membrane (PMemb).

Figure 5.16 shows the DBN4 behaviour tracking the system. It also shows the particular situation previously explained where there is no DBN for HBG-PC4 in mode purge because the state variable estimated in that PC has derivative causality. The same problem appears in the DBN modeling the complete system. That DBN in mode purge has only 4 state variables, instead of 5 and there is no estimation for measurement *Ppump*, directly related to the state variable in derivative causality. This fact can be observed in the figure because during those intervals there is no values for *Ppump*.

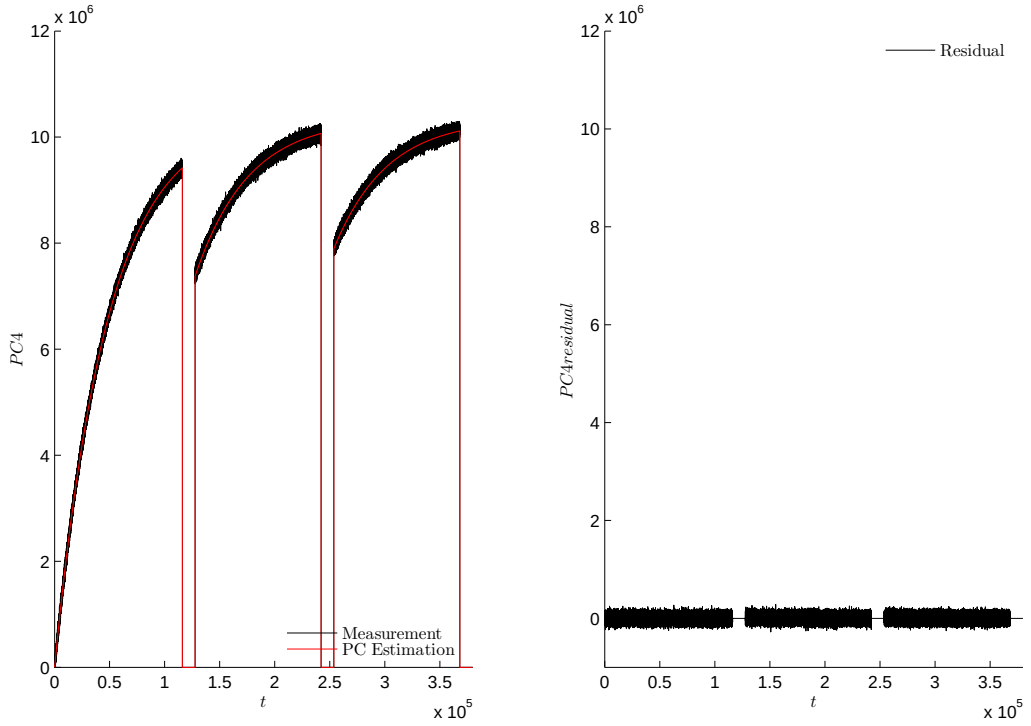


Figure 5.16: DBN4 estimation compared to the ROS measurement along three working cycles. The system measurement is the pressure out of the recirculating pump ($PPump$). In mode purgue the state variable has derivative causality and there is not DBN4.

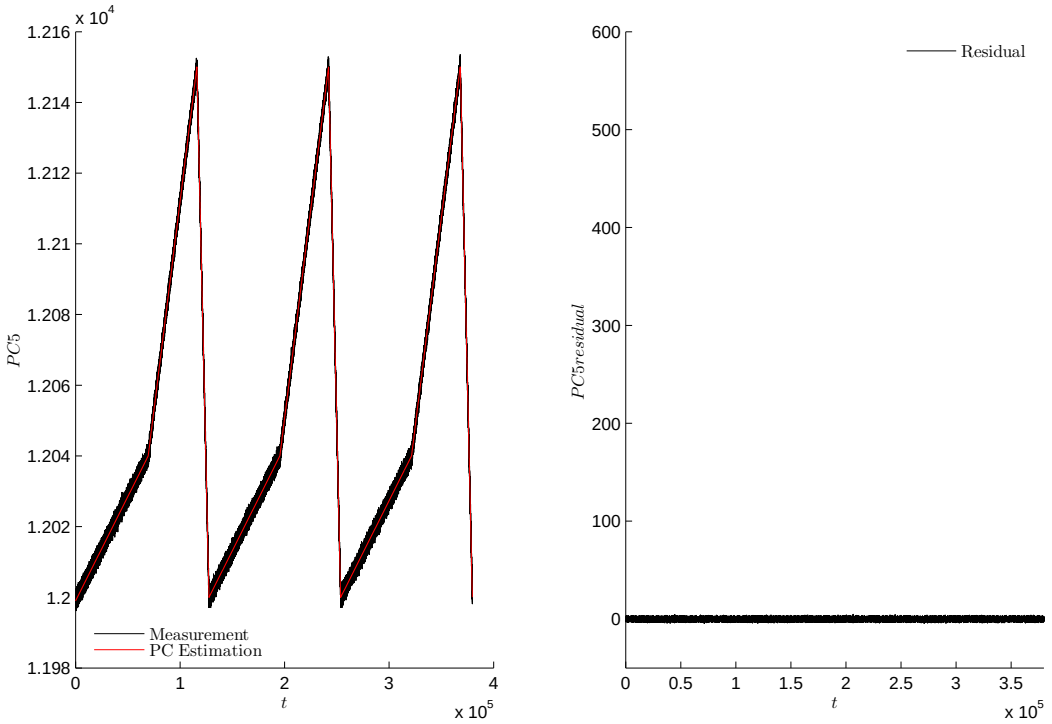


Figure 5.17: DBN5 estimation compared to the ROS measurement along three working cycles. The system measurement is the concentration (Pk).

5.4 Diagnosis results

The data for the experiments have been generated from the Simulink model of the ROS using a 0.5 sample time and adding a 2% gaussian noise to the measurements. All the experiments have been repeated ten times, the results presented are average values from those repetitions. The standard deviation is shown in brackets. Decision making regarding fault detection and convergence have been done with a ztest [13, 53].

5.4.1 Discrete faults

The ROS has three working modes, as it has been explained before. Discrete faults are related to them. To be more precise, they are faults in the discrete actuators involved in the mode changes: The recirculation pump and the multiposition valve.

The recirculation pump has two states, ON and OFF, and it switches from one to the other by means of a control signal. According to that, it can fail in two different ways: 1) It autonomously changes its state, and 2) It does not obey the order from the control signal to change the state. To summarize, the recirculating pump has 4 possible faults, assuming 1 means ON and 0 means OFF, the possible faults can be labeled as:

- **Pump 1 → 0:** The pump is ON and it autonomously switches to OFF.
- **Pump 0 → 1:** The pump is OFF and it autonomously switches to ON.
- **Pump 1 → 1:** The pump is stuck ON (The pump receives the order to switch OFF but it remains ON).
- **Pump 0 → 0:** The pump is stuck OFF (The pump receives the order to switch ON but it remains OFF).

On the other hand, there is the multiposition valve with three states: M1, M2 and P. This valve is also commanded by three control signals: M1, M2 and P that determines the corresponding position. So it can fail in the same ways as the recirculating pump: 1) Change autonomously its state, and 2) Do not obey the order from the control signal. In this actuator, there are 3 different states, so there are more combinations characterizing the faults. Considering 100 means the valve is in position M1; 010 means the valve is in position M2; and 001 means the valve is in position P:

- **Valve 100 → 010:** The valve is in position M1 and it autonomously switches to position M2.
- **Valve 100 → 001:** The valve is in position M1 and it autonomously switches to position P.
- **Valve 100 → 100:** The valve is stuck in position M1.
- **Valve 010 → 100:** The valve is in position M2 and it autonomously switches to position M1.
- **Valve 010 → 001:** The valve is in position M2 and it autonomously switches to position P.
- **Valve 010 → 010:** The valve is stuck in position M2.
- **Valve 001 → 100:** The valve is in position P and it autonomously switches to position M1.
- **Valve 001 → 010:** The valve is in position P and it autonomously switches to position M2.
- **Valve 001 → 001:** The valve is stuck in position P.

The HFSM has been presented in Table 5.3 and it will be used to generate discrete fault candidates. This matrix can be augmented with qualitative information to obtain the Hybrid Qualitative Fault Signature Matrix (HQFSM) in Table 5.7. Sign +(-) means the residual generated by the HBG-PC in the column has a positive (negative) sign, while the asterisk (*) means the residual sign cannot be computed unambiguously. HQFSM will be used to discriminate among discrete fault candidates.

	HBG-PC1	HBG-PC2	HBG-PC3	HBG-PC4	HBG-PC5
Pump 1 → 0			+	-	
Pump 0 → 1			-	+	
Pump 0 → 0			+	-	
Pump 1 → 1			-	+	
Valve 100 → 010		+	*		*
Valve 100 → 001		-	*		-
Valve 100 → 100		+	*		*
Valve 010 → 100		+	*		*
Valve 010 → 001		-	*		-
Valve 010 → 010		+	*		+
Valve 001 → 100		+	*		+
Valve 001 → 010		+	*		+
Valve 001 → 001		-	*		-

Table 5.7: Hybrid Qualitative Fault Signature Matrix (HQFSM) of the ROS.

Fault detection Tables 5.8 and 5.9 summarize the fault detection results regarding discrete faults. Each row in table 5.8 summarizes results for a discrete fault: second column indicates the working mode when the fault arose, while third column presents in which working modes the fault was detected. There is one value (working mode) for each DBN that detects the fault, corresponding to the four HBG-PCs (HBG-PC1 is not affected by the discrete actuators) and the complete system, respectively.

Table 5.9 contains the average detection time stamps of ten DBNs for each minimal DBN[1] and the DBN of the complete system for the considered faults shown in the first column (the standard deviation is always zero, so it has been omitted).

	Working Mode	**Detection Mode**
Pump 1 → 0	M1	M1/M1/M1
Pump 0 → 1	P	-/-/P
Pump 0 → 0	M1	M1/M1/M1
Pump 1 → 1	P	-/-/P
Valve 100 → 010	M1	P/P/M1/M1
Valve 100 → 001	M1	-/M1/M1/M1
Valve 100 → 100	M2	P/P/M2/M2
Valve 010 → 100	M2	P/P/M2/M2
Valve 010 → 001	M2	-/M2/M2/M2
Valve 010 → 010	P	P/P/P/P
Valve 001 → 100	P	P/P/P/P
Valve 001 → 010	P	P/P/P/P
Valve 001 → 001	M1	M1/M1/M1/M1

Table 5.8: Fault detection of discrete faults in the ROS. Fault detection working modes of the possible discrete faults of the system.

	HBG-PC2	**HBG-PC3**	**HBG-PC4**	**HBG-PC5**	**Complete System**
Pump 1 → 0		106	3068		568
Pump 0 → 1		ND	ND		983
Pump 0 → 0		840	10068		57
Pump 1 → 1		ND	ND		3
Valve 100 → 010	106070	105592		3456	3456
Valve 100 → 001	ND	20		2309	568
Valve 100 → 100	47012	46632		905	905
Valve 010 → 100	40819	40582		1086	1086
Valve 010 → 001	ND	21		1421	2
Valve 010 → 010	1411	1032		1465	1013
Valve 001 → 100	2222	82		133	63
Valve 001 → 010	1828	74		379	55
Valve 001 → 001	63	2		1438	215

Table 5.9: Fault detection of discrete faults in the ROS. Fault detection time stamps of the possible discrete faults of the system (the standard deviation is not shown because it is always zero).

[1]DBN1 is not shown because it is not affected by discrete faults and it does not give any false positive.

Looking at detection time stamp values, the minimal DBNs have basically the same behaviour than the DBN of the complete system. More specific, regarding efficiency, the minimal DBNs obtain a better detection time in three scenarios, there is a draw in other three scenarios. Finally, the DBN of the complete system gets an earlier detection time in seven out of thirteen scenarios but in three of them the difference with the best minimal DBN is less than ten seconds.

Regarding correctness in the fault detection process, two issues must be pointed out. First, there are two scenarios where minimal DBNs can not detect the discrete fault; those particular situations are the faults in the recirculation pump, when it autonomously switches to ON state ($0 \rightarrow 1$), and when it is stuck ON($1 \rightarrow 1$). The recirculating pump behaviour is closely related to the purge mode, it changes its position when starting/ending the purge mode (pump is switched OFF/ON, respectively). It is also affected by the part of the system that changes causality during purge mode so there is not minimal DBN modeling the recirculation pump during that period. Those characteristics make the minimal DBNs not sensitive to those changes in the recirculating pump.

Second, PC2 is not able to detect two faults in the valve: 1) Valve $100 \rightarrow 001$ and 2) Valve $010 \rightarrow 001$. Looking at PC2 model (Figure 5.6) we can see that it is affected by the multiposition valve, but only in working modes 1 and 2. This is also presented in Table 5.2.

On the other hand, minimal DBNs will help in the isolation stage reducing the set of fault candidates and avoiding considering all possible faults in the system as candidates.

We will be using exoneration in the discrete faults isolation stage, so the experimental HQFSM in Table 5.10 must be introduced. This matrix presents the actual behaviour of the HBG-PCs and the DBNs derived from them, instead of the theoretical behaviour. Comparing Tables 5.7 and 5.10 we can observe that in the second one, the cells corresponding to the HBG-PCs that do not detect a fault, which have already been explained, are empty, meaning that those HBG-PCs and the DBNs derived from them do not detect the corresponding fault. The experimental HQFSM in Table 5.10 will be used in fault isolation tasks.

	HBG-PC1	**HBG-PC2**	**HBG-PC3**	**HBG-PC4**	**HBG-PC5**
Pump 1 → 0			+	-	
Pump 0 → 1					
Pump 0 → 0			+	-	
Pump 1 → 1					
Valve 100 → 010		+	*		*
Valve 100 → 001			*		-
Valve 100 → 100		+	*		*
Valve 010 → 100		+	*		*
Valve 010 → 001			*		-
Valve 010 → 010		+	*		+
Valve 001 → 100		+	*		+
Valve 001 → 010		+	*		+
Valve 001 → 001		-	*		-

Table 5.10: Experimental Hybrid Qualitative Fault Signature Matrix (HQFSM) of the ROS.

Isolation and Identification After fault detection stage, the isolation task is performed. Regarding the detection information, besides detection time, presented in Table 5.9, there are also the signatures generated after that detection (Table 5.11).

The isolation process have been explained in Chapter 4. Summarizing the isolation and identification process: Discrete fault candidates are preferred candidates over parametric ones. Exoneration will be used to generate the set of discrete fault candidates. In case all discrete fault candidates are rejected, parametric candidates will be tested.

Considering the signatures in Table 5.11 and the experimental HQFSM in Table 5.10 the set of discrete fault candidates have been generated, Table 5.12 present the simulated fault in first column and the fault

candidates in the second column. To build the table we have taken into account that our assumption is that the current mode is known. Hence, some potential fault candidates are not consistent with the current state, and they are rejected.

	HBG-PC1	**HBG-PC2**	**HBG-PC3**	**HBG-PC4**	**HBG-PC5**
Pump 1 → 0			+	-	
Pump 0 → 1					
Pump 0 → 0			+	-	
Pump 1 → 1					
Valve 100 → 010		+	*		*
Valve 100 → 001			*		-
Valve 100 → 100		+	*		*
Valve 010 → 100		+	*		*
Valve 010 → 001			*		-
Valve 010 → 010		+	*		+
Valve 001 → 100		+	*		+
Valve 001 → 010		+	*		+
Valve 001 → 001		-	*		-

Table 5.11: Fault signatures derived after fault detection for discrete faults.

Fault	**Candidates**
Pump 1 → 0	Pump 1 → 0
Pump 0 → 1	Pump 0 → 1
Pump 0 → 0	Pump 0 → 0
Pump 1 → 1	Pump 1 → 1
Valve 100 → 010	Valve 100 → 010
Valve 100 → 001	Valve 100 → 001
Valve 100 → 100	Valve 100 → 100
Valve 010 → 100	Valve 010 → 100
Valve 010 → 001	Valve 010 → 001
	Valve 010 → 100
Valve 010 → 010	Valve 010 → 010
Valve 001 → 100	Valve 001 → 100
	Valve 001 → 010
Valve 001 → 010	Valve 001 → 010
	Valve 001 → 100
Valve 001 → 001	Valve 001 → 001

Table 5.12: Discrete fault candidates for the discrete faults tested in the ROS.

As it has been explained in Chapter 4, for each fault, the HBG-PCs are configured to simulate each fault candidate. In this particular case, the minimal DBNs for each HBG-PC are generated and run for a period $\sigma_t = 6000$ time stamps. None of the fault candidates which were not the actual fault converges in any case. Table 5.13 presents the isolation and identification results for the fault candidate that converges, which is the actual fault introduced in the system.

	HBG-PC2	HBG-PC3	HBG-PC4	HBG-PC5
Pump 1 → 0		5511	*	
Pump 0 → 0		5123	*	
Valve 100 → 010	2391	2401		2390
Valve 100 → 001		3771		1660
Valve 100 → 100	2891	2891		2891
Valve 010 → 100	2109	3190		2132
Valve 010 → 001		3260		1535
Valve 010 → 010	3325	6290		3260
Valve 001 → 100	4730	5360		5495
Valve 001 → 010	3890	3865		5390
Valve 001 → 001	1313	3070		1325

Table 5.13: Fault convergence time of discrete faults isolation and identification in the ROS. Convergence time is presented in time stamps (the standard deviation is not shown because it is always zero). The asterisk marks the situations where there is not a DBN able to simulate it, because the system equations can only be solved using derivative causality.

Table 5.13 summarizes the identification results of the discrete faults in the ROS. The table shows the average time stamps of 10 experiments (the standard deviation is always zero, so it has been omited). As it has been previously said, the actual fault is the single candidate that converges so Table 5.13 shows only the information of that candidate. The system dynamics are slow, so it is possible to say that all of them converge fast, which means that the residuals are deactivated.

5.4.2 Parametric faults

The parameters in the ROS have been introduced in Table 5.1. All the faults have been simulated as a 10% abrupt change in its nominal value.

Most of the parameters in the system are present in every working mode. Hence, their associated faults can be introduced and afterwards be detected in any mode. On the contrary, I_{pump}, R_{memb}, R_{brine1}, R_{brine2}, R_{drain} and R appear only in some of the working modes so the faults in those parameters have been tested twice: first when the fault appears in the mode in which the parameter is active, and second, when the fault appears in a mode in which the parameter is not active so it will not influence the system behaviour until the appropriate working mode is active again. Table 5.14 shows the relation between parameters and the working mode when they appear.

Parameter	**Working Mode**
I_{pump}	Mode 1, Mode 2
R_{Memb}	Mode 1, Mode 2
R_{brime1}	Mode 1
R_{brime2}	Mode 2
R_{drain}	Mode purge
R	Mode purge

Table 5.14: Physical parameters in the ROS and the working modes in which they actually are present in the system. Parameters not shown in this table appear along the three working modes.

Fault detection Due to the problem presented above, the experiments performed for fault diagnosis results have been divided in two groups: 1) The fault appears in a working mode when the parameter is part of the system, and 2) The fault appears in a working mode when the parameter is not in the system. Results are presented in Tables 5.15 and 5.16 for the first group and Tables 5.17 and 5.18, for the second group.

The experiments have confirmed that a fault is detected only when the parameter is actually in the system, that is because at that moment, the parameter appears in the system equations, so its value affects the system behaviour.

	Working Mode	Detection Mode
R_{bwp}	M1	M1/-
R_{fpump}	M1	M1/M1
I_{fpump}	P	P/P
R_{rpump}	M2	M2/M2
I_{pump}	M1	M1/M1
$R_{pipezero1}$	M2	M2/M2
C_{Res}	M1	M1/M1
R_{pipe}	M1	M1/M1/-/M1
C_{Memb}	M2	M2/M2
R_{Memb}	M1	M1/M1
R_{brine1}	M1	M1/M1/M1
R_{brine2}	M2	M2/M2/M2
R_{drain}	P	P(+1)/ P(+1)
C_k	M1	M1/M1
R	P	P/P

Table 5.15: Fault detection of parametric faults in the ROS. Third column represents fault detection working modes of the parametric faults of the system; second column represent the working mode when the fault happend. The parameters are present in the system.

	HBG-PC1	HBG-PC2	HBG-PC3	HBG-PC4	HBG-PC5	Complete System
R_{bwp}	1711					ND
R_{fpump}	78					489
I_{fpump}	3					42
R_{rpump}				3031		3056
I_{pump}				3		10
$R_{pipezero1}$		3111				3285
C_{Res}		4				4
R_{pipe}		1410	696	ND		420
C_{Memb}			22			3
R_{Memb}			933			659
R_{brine1}		8318	2564			5568
R_{brine2}		2725	2039			2039
R_{drain}			1251304			1151454
C_k					2	2
R					1245	1245

Table 5.16: Fault detection of parametric faults in the ROS. Fault detection time stamps of the parametric faults of the system when the fault occurs in a working mode when the parameter is in the system (the standard deviation is not shown because it is always zero).

Tables 5.15 and 5.16 present the detection results when the parametric fault appears in a working mode when the parameter is part of the system. Looking at Table 5.15, in columns there is the parameter affected by the fault, the working mode when the fault occurs and the working mode when the fault is detected, there is one value (working mode) for each DBN that detects the fault (including the DBN modeling the complete system).

Table 5.16 shows the average detection time stamps for ten DBNs obtained for each parametric fault for each minimal DBN and the DBN derived from the complete system (the standard deviation is not shown because it is always zero). All the parametric faults are detected in the same working mode when they appear except the fault in resistance R_{drain}, which is detected in the same working mode but after a complete working cycle.

Regarding DBNs performance in parametric fault detection, the minimal DBNs obtain detection times comparable to the values from the DBN modeling the complete system. The minimal DBNs gets an earlier detection time than the complete system DBN in seven out of fifteen scenarios. Both obtain the same detection time in four of them. Finally, the DBN of the complete system obtains an earlier detection time in four out of fifteen.

DBN4 does not detect the fault in R_{pipe} but it is detected by other minimal DBNs and that allows to calculate the set of fault candidates. On the other hand, the DBN modeling the complete system does not detect the fault in R_{bwp}, so this fault will not be detected in case of using that DBN in the diagnosis process.

Tables 5.17 and 5.18 show the results obtained when faults in some parameters appear when those parameters are not used in that working mode (Standard deviation has been omitted because it is always zero). Table 5.17 shows that the faults are always detected when the parameter appears in the system. Analysing the average detection time, the faults are detected at very early time stamps of the corresponding working mode. The DBN modeling the complete system usually obtains earlier values, this reflects the influence of the inputs used in each DBN compared to the number of state nodes.

	Working Mode	**Detection Mode**
I_{pump}	P	M1/M1
R_{Memb}	P	M1/M1
R_{brine1}	M2	M1/ M1/ M1
R_{brine2}	M1	M2/M2/M2
R_{drain}	M2	P/P
R	M2	P/P

Table 5.17: Fault detection of parametric faults in the ROS. Third column represents fault detection working modes of the parametric faults of the system; second column represent the working mode when the fault happend. The parameters are not present in the system.

	HBG-PC1	**HBG-PC2**	**HBG-PC3**	**HBG-PC4**	**HBG-PC5**	**Complete System**
I_{pump}				5112		5184
R_{Memb}			10942			8388
R_{brine1}		51588	49131			28903
R_{brine2}		32388	31052			29929
R_{drain}			20825			20793
R					32625	31832

Table 5.18: Fault detection of parametric faults in the ROS. Fault detection time stamps of the parametric faults of the system when the fault occurs in a working mode when the parameter is not in the system (the standard deviation is not shown because it is always zero).

Isolation and Identification The isolation and identification stage of parametric faults starts just after discarding all discrete fault candidates. At this point, the isolation process calculates the set of parametric fault candidates using the FSM of the actual working mode (Tables 5.4, 5.5 and 5.6). The minimal DBNs for the corresponding working mode and the fault candidates are built and simulated to estimate the actual fault and to discard the other candidates. The results obtained for all the parametric fault candidates are summarized from Table 5.20 to Table 5.24 [2].

In those tables, we have organized the information as follows: First column shows the actual fault together with the actual parameter value (in brackets). Second column presents the fault candidates after the isolation stage. The third column introduces the DBN that is being used to identify the fault. The last three columns summarize the DBN behaviour regarding the identification task, presenting the convergence time, the parameter estimation and the normalized mean squared error (MSE), respectively. DBNs that do not converge have a hyphen in the corresponding cell.

Parameter	**Table**
I_{fpump}	Table 5.20
R_{fpump}	Table 5.20
R_{bwp}	Table 5.20
R_{rpump}	Table 5.24
I_{pump}	Table 5.24
$R_{pipezero1}$	Table 5.21
R_{Memb}	Table 5.22
R_{drain}	Table 5.22
R_{pipe}	Table 5.23
R_{brine1}	Table 5.23
R_{brine2}	Table 5.23
C_k	Table 5.24
R	Table 5.24
C_{Res}	Table 5.21
C_{Memb}	Table 5.22

Table 5.19: Summary of the parameter in the system (Column1) and table summarizing the identification results (Column 2) for the ROS.

[2]Due to the size of the table including all the parametric faults in the system, it has been divided using different tables for each HBG-PC (See Table 5.19 for the relation between parameter and table with the results)

Actual Fault	**Candidates**	**DBNs**	**Conv. Time**	**Param. Estim.**	**Norm. MSE**
I_{fpump} $(2.73 \cdot 10^{12})$	I_{fpump}	**PC1**	5564.4 (238.16)	$3.47 \cdot 10^{13}$ $(2.13 \cdot 10^{14})$	$7.3 \cdot 10^{-1}$ (0.74)
		CS	-	-	-
	R_{fpump}	PC1	-	-	-
		CS	-	-	-
	R_{bwp}	PC1	-	-	-
		CS	-	-	-
R_{fpump} $(3.12 \cdot 10^{10})$	R_{fpump}	**PC1**	5460 (0.0)	$3.11 \cdot 10^{10}$ $(7.8 \cdot 10^{7})$	$4.9 \cdot 10^{-1}$ $(1.7 \cdot 10^{-4})$
		CS	5460 (0.0)	$3.12 \cdot 10^{10}$ $(2.0 \cdot 10^{7})$	$9.4 \cdot 10^{-1}$ $(2.0 \cdot 10^{-12})$
	I_{fpump}	PC1	-	-	-
		CS	-	-	-
	R_{bwp}	PC1	5460 (0.0)	$5.03 \cdot 10^{9}$ $(1.4 \cdot 10^{8})$	$4.9 \cdot 10^{-1}$ $(3.5 \cdot 10^{-4})$
		CS	5460 (0.0)	$5.12 \cdot 10^{9}$ $(3.2 \cdot 10^{7})$	$9.4 \cdot 10^{-1}$ $(2.5 \cdot 10^{-12})$
R_{bwp} $(2.5 \cdot 10^{9})$	R_{bwp}	PC1	5459 (0.0)	$2.50 \cdot 10^{9}$ $(4.6 \cdot 10^{7})$	$9.9 \cdot 10^{-1}$ (0.05)
		CS	5459 (0.0)	$2.52 \cdot 10^{9}$ $(4.4 \cdot 10^{7})$	$7.4 \cdot 10^{-1}$ $(4.9 \cdot 10^{-5})$
	R_{fpump}	**PC1**	5459 (0.0)	$2.86 \cdot 10^{10}$ $(3.7 \cdot 10^{6})$	$9.8 \cdot 10^{-1}$ $(8.2 \cdot 10^{-4})$
		CS	5459 (0.0)	$2.86 \cdot 10^{10}$ $(2.8 \cdot 10^{7})$	$7.4 \cdot 10^{-1}$ $(5.5 \cdot 10^{-5})$
	I_{fpump}	PC1	-	-	-
		CS	5459 (0.0)	$3.61 \cdot 10^{12}$ $(5.2 \cdot 10^{13})$	$7.4 \cdot 10^{-1}$ $(7.5 \cdot 10^{-5})$

Table 5.20: Isolation and Identification results for parametric faults in ROS. Faults identified by Possible Conflict 1 and the complete system (CS).

Actual Fault	**Candidates**	**DBNs**	**Conv. Time**	**Param. Estim.**	**Norm. MSE**
$R_{pipezero1}$ $(1.1 \cdot 10^{9})$	$R_{pipezero1}$	**PC2**	5213.1 (777.6)	$2.05 \cdot 10^{9}$ $(1.2 \cdot 10^{9})$	$9.1 \cdot 10^{-1}$ (1.3)
		CS	5540 (0.0)	$1.70 \cdot 10^{9}$ $(1.4 \cdot 10^{9})$	$9.0 \cdot 10^{-1}$ (0.002)
	C_{res}	PC2	-	-	-
		CS	5549 (0.0)	$1.30 \cdot 10^{-5}$ $(1.8 \cdot 10^{-4})$	$9.0 \cdot 10^{-1}$ $(1.7 \cdot 10^{-4})$
	R_{pipe}	PC2	6575 (0.0)	$2.97 \cdot 10^{13}$ $(2.6 \cdot 10^{11})$	$9.9 \cdot 10^{-1}$ $(8.0 \cdot 10^{-4})$
		CS	5459 (0.0)	$2.87 \cdot 10^{13}$ $(5.2 \cdot 10^{10})$	$9.1 \cdot 10^{-1}$ $(2.4 \cdot 10^{-4})$
	R_{brine2}	PC2	5459 (0.0)	$8.14 \cdot 10^{12}$ $(6.0 \cdot 10^{11})$	$9.3 \cdot 10^{-1}$ (0.05)
		CS	5459 (0.0)	$9.14 \cdot 10^{12}$ $(4.7 \cdot 10^{10})$	$9.2 \cdot 10^{-1}$ $(2.7 \cdot 10^{-4})$
C_{res} $(4.79 \cdot 10^{-6})$	C_{res}	**PC2**	5560 (0.0)	$4.11 \cdot 10^{-6}$ $(2.2 \cdot 10^{-6})$	$7.2 \cdot 10^{-1}$ (1.51)
		CS	5962 (0.0)	$6.56 \cdot 10^{-6}$ $(7.7 \cdot 10^{-6})$	$9.2 \cdot 10^{-1}$ (0.1)
	$R_{pipezero1}$	PC2	-	-	-
		CS	-	-	-
	R_{pipe}	PC2	-	-	-
		CS	-	-	-
	R_{brine1}	PC2	-	-	-
		CS	-	-	-

Table 5.21: Isolation and Identification results for parametric faults in ROS. Faults identified by Possible Conflict 2 and the complete system (CS).

Actual Fault	**Candidates**	**DBNs**	**Conv. Time**	**Param. Estim.**	**Norm. MSE**
R_{memb} $(2.7 \cdot 10^{12})$	R_{memb}	**PC3**	5784.5 $(1.8 \cdot 10^{3})$	$2.45 \cdot 10^{12}$ $(8.6 \cdot 10^{11})$	$7.19 \cdot 10^{-1}$ (2.662)
		CS	5465 $(2.4 \cdot 10^{2})$	$2.73 \cdot 10^{12}$ $(8.0 \cdot 10^{9})$	$7.0 \cdot 10^{-1}$ $(5.7 \cdot 10^{-5})$
	R_{pipe}	PC3	-	-	-
		CS	-	-	-
	C_{memb}	PC3	-	-	-
		CS	-	-	-
	R_{brine1}	PC3	-	-	-
		CS	-	-	-
R_{drain} $(4.6 \cdot 10^{9})$	R_{drain}	**PC3**	5459 (0.0)	$2.43 \cdot 10^{9}$ $(7.1 \cdot 10^{9})$	$8.4 \cdot 10^{-1}$ $(1.5 \cdot 10^{-15})$
		CS	5459 (0.0)	$9.33 \cdot 10^{9}$ $(1.5 \cdot 10^{10})$	$8.3 \cdot 10^{-1}$ $(5.5 \cdot 10^{-5})$
	R_{pipe}	PC3	5855.5 (66.1)	$2.92 \cdot 10^{13}$ $(2.0 \cdot 10^{11})$	$9.6 \cdot 10^{-1}$ (0.3)
		CS	5459 (0.0)	$2.96 \cdot 10^{13}$ $(4.9 \cdot 10^{11})$	$9.0 \cdot 10^{-1}$ (0.07)
	C_{memb}	PC3	5459 (0.0)	$5.61 \cdot 10^{-9}$ $(6.0 \cdot 10^{-9})$	$9.6 \cdot 10^{-1}$ $(5.8 \cdot 10^{-4})$
		CS	-	-	-
C_{memb} $(2.9 \cdot 10^{-10})$	C_{memb}	**PC3**	5954.1 $(2.7 \cdot 10^{2})$	$3.10 \cdot 10^{-10}$ $(2.9 \cdot 10^{-9})$	$7.2 \cdot 10^{-1}$ (1.4)
		CS	5459 (0.0)	$2.31 \cdot 10^{-10}$ $(2.3 \cdot 10^{-9})$	$7.4 \cdot 10^{-1}$ (0.2)
	R_{pipe}	PC3	-	-	-
		CS	-	-	-
	R_{memb}	PC3	-	-	-
		CS	-	-	-
	R_{brine2}	PC3	-	-	-
		CS	-	-	-

Table 5.22: Isolation and Identification results for parametric faults in ROS. Faults identified by Possible Conflict 3 and the complete system (CS).

Actual Fault	**Candidates**	**DBNs**	**Conv. Time**	**Param. Estim.**	**Norm. MSE**
R_{pipe} $(3.14 \cdot 10^{13})$	R_{pipe}	**PC2**	5460 (0.0)	$3.15 \cdot 10^{13}$ $(1.1 \cdot 10^{11})$	$5.2 \cdot 10^{-1}$ (0.001)
		PC3	5459 (0.0)	$3.15 \cdot 10^{13}$ $(4.4 \cdot 10^{10})$	$5.8 \cdot 10^{-1}$ $(1.6 \cdot 10^{-4})$
		CS	5459 (0.0)	$3.16 \cdot 10^{13}$ $(3.0 \cdot 10^{10})$	$7.2 \cdot 10^{-1}$ $(7.6 \cdot 10^{-5})$
	R_{brine1}	PC2	-	-	-
		PC3	-	-	-
		CS	-	-	-
R_{brine1} $(1.00 \cdot 10^{13})$	R_{brine1}	PC2	5471.1 (38.3)	$6.18 \cdot 10^{13}$ $(1.6 \cdot 10^{14})$	$9.8 \cdot 10^{-1}$ (2.2)
		PC3	7342.6 $(1.4 \cdot 10^{3})$	$1.42 \cdot 10^{13}$ $(1.6 \cdot 10^{13})$	$8.6 \cdot 10^{-1}$ (2.5)
		CS	5459 (0.0)	$1.09 \cdot 10^{13}$ $(9.9 \cdot 10^{11})$	$7.8 \cdot 10^{-1}$ $(4.8 \cdot 10^{-4})$
	R_{pipe}	PC2	-	-	-
		PC3	-	-	-
		CS	-	-	-
R_{brine2} $(1.00 \cdot 10^{13})$	R_{brine2}	**PC2**	5560 (0.0)	$1.04 \cdot 10^{13}$ $(6.4 \cdot 10^{11})$	$7.5 \cdot 10^{-1}$ $(2.0 \cdot 10^{-4})$
		PC3	5549 (0.0)	$2.80 \cdot 10^{13}$ $(1.7 \cdot 10^{10})$	$8.9 \cdot 10^{-1}$ (0.01)
		CS	5449 (0.0)	$1.02 \cdot 10^{13}$ $(5.3 \cdot 10^{10})$	$7.7 \cdot 10^{-1}$ $(6.8 \cdot 10^{-5})$
	R_{pipe}	PC2	-	-	-
		PC3	-	-	-
		CS	-	-	-

Table 5.23: Isolation and Identification results for parametric faults in ROS. Faults identified by Possible Conflicts 2 and 3 (both the same faults) and the complete system (CS).

Actual Fault	**Candidates**	**DBNs**	**Conv. Time**	**Param. Estim.**	**Norm. MSE**
R_{rpump} (2.20)	R_{rpump}	**PC4**	5461 (0.0)	2.18 (0.02)	$9.5 \cdot 10^{-1}$ ($5.9 \cdot 10^{-4}$)
		CS	5463 (0.0)	2.20 (0.03)	$8.3 \cdot 10^{-1}$ ($1.9 \cdot 10^{-4}$)
	R_{pipe}	PC4	-	-	-
		CS	5459 (0.0)	$2.85 \cdot 10^{13}$ ($1.3 \cdot 10^{12}$)	$3.8 \cdot 10^{-1}$ (0.6)
	I_{pump}	PC4	-	-	-
		CS	-	-	-
I_{pump} ($1.06 \cdot 10^4$)	I_{pump}	**PC4**	5460 ($1.8 \cdot 10^2$)	$1.00 \cdot 10^4$ ($3.6 \cdot 10^3$)	$5.1 \cdot 10^{-1}$ (2.1)
		CS	5458 (0.0)	$2.07 \cdot 10^4$ ($1.9 \cdot 10^3$)	$7.5 \cdot 10^{-1}$ ($3.2 \cdot 10^{-4}$)
	R_{rpump}	PC4	-	-	-
		CS	7026 (86.7)	50.41 (5.3)	$9 \cdot 10^{-1}$ (1.11)
	R_{pipe}	PC4	-	-	-
		CS	5459 (0.0)	$3.17 \cdot 10^{13}$ ($1.0 \cdot 10^{12}$)	$8.4 \cdot 10^{-1}$ (0.3)
C_k ($1.86 \cdot 10^5$)	C_k	**PC5**	5506.4 (56.9)	$1.23 \cdot 10^3$ ($5.5 \cdot 10^4$)	$8.6 \cdot 10^{-1}$ (0.1)
		CS	-	-	-
R (0.61)	R	**PC5**	5661.3 (373.2)	0.94 (0.7)	$2.2 \cdot 10^{-1}$ (0.1)
		CS	-	-	-
	C_k	PC5	5459.4 (0.8)	$2.15 \cdot 10^5$ ($1.1 \cdot 10^4$)	$8.6 \cdot 10^{-1}$ (0.003)
		CS	-	-	-

Table 5.24: Isolation and Identification results for parametric faults in ROS. Faults identified by Possible Conflict 4 and faults identified by Possible Conflict 5 and the complete system (CS).

Analysing the identificacion results of parametric faults it has been seen a situation that never happened before, some of the DBNs estimating a fault candidate different than the actual fault converge. This means that the DBN outputs converge to the actual measurements in the system according to the ztest. The main reason for that behaviour seems to be that this system has all the state variables direcly measured and, those are the only measurements in the whole system. Because of that distribution of the observations, the information in each DBN is the minimal information needed to have part of the system dynamics. DBNs usually obtain better estimations as they have more information about the system to adjust their weights, so for the ROS they have very few extra information to improve the estimation of an additional state node (the faulty parameter). Moreover, looking at the system model, the parameters that present that problem are not diagnosticable, as there are not measurements between those parameters in the system, it is not possible to clearly identify the actual fault. This problem does not depend on the identification method used.

Due to this situation, besides the convergence decided by the ztest, there is another measurement used to decided what is the actual fault identified in the experiment if some of the candidates converge. That measurement is the Normalized Mean Square Error (NMSE)[57] of the measurements' estimation, it is shown in the sixth column of the Tables from 5.20 to 5.24.

The Normalized Mean Square Error (MSE) of a univariable signal [57] is calculated as:

$$\sqrt{\frac{\frac{\sum_{k=1}^{N}(\widehat{y_k}-y_k)^2)}{N}}{\sigma_N^2}}$$

Where the numerator in the root is the MSE of the variable and the denominator (σ_t^2) is the variance of that variable.

Minimal DBNs have only one output, so the previous equation have been used to calculate the NMSE. On the contrary, the DBN of the complete system has five outputs, so the NMSE have been calculated as the mean value of the NMSE of each output. This value is closer to zero as the estimation is better and it is close to one, when the estimation is more different than the measured value. According to that, in case several DBNs converge giving an estimation for the faulty parameter, the one with the smallest NMSE will be chosen, in case of a draw in the NMSE value, the candidate with the smaller standard deviation will be chosen.

Analysing the identification results, there are eight faults where only the actual fault converges (I_{fpump}, R_{memb}, R_{pipe}, R_{brine1}, R_{brine2}, C_k, C_{res}, C_{memb}). There are also three faults where the DBN modeling the complete system does not converge to the actual fault, while the minimal DBNs always give an estimation for the correct fault candidate. Regarding the accuracy of the estimations considering the NMSE of the DBN output in case several DBNs converge, the minimal DBNs obtain a more accurate estimation in nine out of fifteen faults while the DBN modeling the complete system gets it for three parameters.

There is a particular situation for resistances R_{fpump} and R_{bwp}. They are placed in the model in the same junction, this means they are modeled as two resistances in series connection. There is not any sensor between both of them so they are not diagnosticable. Because of that, for each of those faults, the DBNs identifying the other resistance also converge, moreover, for R_{bwp} the NMSE is smaller for the DBN1 estimating R_{fpump}.

It can be seen that most of the fault candidates where the wrong DBN converges are resistances (R_{fpump}, R_{bwp}, $R_{pipezero1}$, R_{drain}, R_{rpump}, R and I_{pump}). Those parameters are not direcly related to the state variables so in this system, they are not directly related to the measurements either.

Table 5.25 complements the convergence information, it summarizes, for each fault, the best minimal DBN and the DBN model of the complete system regarding the parameter estimation. Due to the number of parameters, the table has been divided in three parts, each one has the Root Mean Squared Error (RMSE) in the parameter estimation for the minimal DBN and the DBN fo the complete system, if the DBN converges.

According to Table 5.25 the minimal DBNs converge always giving an estimation for the faulty parameter while the DBN modeling the complete system does not converge for three out of fifteen faults. The RMSE calculated for the estimation is comparable for the minimal DBN and for the DBN of the complete system, there are not a big difference on the values.

Faulty Parameter	R_{bwp}	R_{fpump}	I_{fpump}	R_{rpump}	I_{pump}
Minimal DBN used	PC1	PC1	PC1	PC4	PC4
PC convergence	$\surd$	$\surd$	$\surd$	$\surd$	$\surd$
PC mse	$2.61 \cdot 10^{10}$*	$6.89 \cdot 10^{7}$	$7.05 \cdot 10^{12}$	$2.92 \cdot 10^{-2}$	$5.17 \cdot 10^{2}$
CS convergence	$\surd$	$\surd$	χ	$\surd$	$\surd$
CS mse	$5.41 \cdot 10^{8}$	$4.36 \cdot 10^{8}$	-	$2.84 \cdot 10^{13}$*	$1.20 \cdot 10^{4}$
Faulty Parameter	$R_{pipezero1}$	C_{res}	R_{pipe}	C_{memb}	R_{memb}
Minimal DBN used	PC2	PC2	PC2	PC3	PC3
PC convergence	$\surd$	$\surd$	$\surd$	$\surd$	$\surd$
PC mse	$7.48 \cdot 10^{8}$	$7.62 \cdot 10^{-7}$	$7.18 \cdot 10^{11}$	$1.11 \cdot 10^{-9}$	$3.51 \cdot 10^{10}$
CS convergence	$\surd$	$\surd$	$\surd$	$\surd$	$\surd$
CS mse	$1.10 \cdot 10^{9}$*	$4.85 \cdot 10^{-6}$	$7.61 \cdot 10^{11}$	$1.88 \cdot 10^{-10}$	$4.21 \cdot 10^{10}$
Faulty Parameter	R_{brine1}	R_{brine2}	R_{drain}	C_k	R
Minimal DBN used	PC3	PC2	PC3	PC5	PC5
PC convergence	$\surd$	$\surd$	$\surd$	$\surd$	$\surd$
PC mse	$1.85 \cdot 10^{13}$	$9.02 \cdot 10^{11}$	$1.29 \cdot 10^{10}$	$1.01 \cdot 10^{5}$	$2.89 \cdot 10^{-2}$
CS convergence	$\surd$	$\surd$	$\surd$	χ	χ
CS mse	$5.11 \cdot 10^{11}$	$6.49 \cdot 10^{11}$	$9.02 \cdot 10^{11}$	-	-

Table 5.25: Root Mean Squared Error in the parameter estimation. Comparison of the best minimal DBN and the DBN model of the complete system.

5.5 Discussion and conclusions

This chapter has presented a complete case study of a real-world system where the proposal in Chapter 4 is applied. The Reverse Osmosis System (ROS) is a complex system with a multiposition actuator and the system dynamics are not trivial.

The architecture presented in Chapter 4 has been successfully applied to a hybrid system combining the HBG-PCs and the minimal DBNs derived from them (Chapter 4). The diagnosis results are satisfactory for discrete as well as small parametric faults (10% of their nominal value).

Discrete and parametric faults in the system have been tested for the three diagnosis stages: Fault detection, isolation and identification.

- Fault detection of discrete faults is quite fast, as it was thought and after simulating the candidate modes, the actual mode converges quickly deactivating its residuals, while the other simulated modes do not converge.
- Continuous or parametric faults detection time is similar using the DBN modeling the complete system and the DBNs derived from the HBG-PCs.
- Regarding fault isolation, minimal DBNs allow to calculate smaller sets of fault candidates so the isolation process is more efficient in this case.
- Regarding fault identification, the results are similar to those obtained for continuous systems: there is no clear winner. The minimal DBNs converge to a more accurate estimation for some parameters, while the DBN modeling the complete system obtains better estimation for others.

.

First conclusion is that the proposed framework based on HBG-PCs and implemented using minimal DBNs was capable of successfully track the hybrid system behaviour across several changes in the working mode, without producing false detection results. At the same time, the diagnosis system is able to detect and to isolate faults related to parameters that are not related to the current working mode. Those are detected once the parameter is used again.

Second conclusion is related to the influence of the ROS structure, and its influence in our approach. The ROS model has five state variables, that can be measured. Hence, our approach produce five almost isolated HBG-PCs. The first effect is that some parameters that can not be structurally isolated also produce poor identification results. Due to the lack of measurements, that was the expected result. Moreover, given that both parameters share the same type (resistances) and are sequentially placed in the system, the results are fully comprehensible. The second effect is that the lack of shared state-variables in the HBG-PCs makes difficult to test our HBG-PC merging proposal for fault identification improvement.

Third conclusion is that a sensitivity analysis might be necessary in those systems where the effect of faults is really weak in several HBG-PCs. In this case, given the small size of the faults, and the level of noise in the system, it was necessary to calculate the experimental HQFSM based on simulated data, to complement the theoretical HQFSM that can be computed from the TCG. This situation can be expected in presence of non-linear behaviours too.

Finally, the ROS has a collection of working modes that can be identified given the value of different operation commands (in our case, values for control signals $M1$, $M2$, and P). This fact has simplified our task for knowing the current state before starting fault isolation. As future work we want to test the approach in a real complex system with several autonomous transitions, thus evaluating the suitability of the DBN approach and/or adding additional tools for hybrid state estimation.

There are future work related to the causality used in the model. Our work has been done assuming integral causality and other approaches may prefer derivative causality. The problem observed in the ROS is that a change in the causality due to a change in a working mode can affect not only the algebraic equations but also a state equation changing between integral and derivative causality. This effect can occur in any model and it will be an issue to study no matter the modeling tool.

The work presented in this Chapter has shown that the proposed diagnosis architecture for hybrid systems considering discrete and parametric faults in a unified way obtains the expected results with a real-world complex system.

Chapter 6
Conclusions

This chapter summarizes the conclusions obtained during the dissertation. It also shorten the contributions that have been presented in previous chapters, some of them have already been published. Some future work ideas are presented at the end.

6.1 Main Contributions

Diagnosis architecture for continuous sytems using Minimal Dynamic Bayesian Networks derived from Possible Conflics [4, 3, 5] The first contribution is a method to derive minimal DBNs from PCs presented in Chapter 3. Using the PCs structure (MEM or TCG) we derive the DBN structure, while the parameters (equations) of the DBN are obtained from the labels in the arcs or edges.

Minimal DBNs can be used efficiently along all the stages of the diagnosis process (Fault detection, isolation and identifcation - FDII). As a result of that, the second contribution in this dissertation is the definition of a PC-based FDII framework for continuous systems diagnosis that has DBNs as the unique estimation method for fault detection and fault identification.

Improving the parameter identificacion stage merging minimal DBNs Minimal DBNs simplify the isolation stage providing smaller sets of fault candidates. But regarding fault identification, it has been observed that the results obtained are not always as accurate as the DBN modeling the complete system. As a general idea, DBNs obtain a more accurate state estimation when there are more observations to adjust the estimation. Because of that, the process of merging some minimal DBNs sharing state variables and/or inputs and measurements provides an improvement in the parameter estimation when needed. This method is presented in Chapter 3 for the 12^{th} order electrical system case study. Based on a heuristic it selects two DBNs that share state variables and the parameter that is being studied, those DBNs are merged to obtain a DBN that models as one subsystem the two minimal redundant subsystems.

Possible Conflicts for Hybrid systems (HPCs). [18, 17] The Possible Conflicts compilation technique has been augmented to be used for hybrid systems (HPCs). Hybrid Bond Graphs (HBGs) have been used as the modeling technique. HBGs extend Bond Graphs including a special type of 1- (0-) junction managed by an automata that indicates if the 1- (0-) junction is active (it is a regular 1- (0-) junction) or it is disconnected (there is zero flow (effort) source in the junction). The guard conditions of those automata represent the events that trigger the mode change in hybrid systems.

PCs of hybrid systems (HPCs o HBG-PCs) have been theoretically characterized and have been succesfully used for tracking continuous systems behaviour and changing model based on the discrete events that trigger transitions of hybrid systems as it is shown in Chapter 4.

HBG-PCs do not need a preenumeration of all possible modes in the system that solves one of the main concerns regarding hybrid systems fault diagnosis. Moreover, if the HBG model of the system have valid causal assignment with all switching junctions set to ON, it allows to derive the set of HBG-PCs that characterize all the HBG-PCs in any mode.

Fault detection and isolation of parametric and discrete faults with HBG-PCs. [79] Hybrid systems present two different types of faults. On one hand, the faults that affect their parameters, known as parametric faults. The diagnosis process for parametric faults is the same used for continuos systems. On the other hand, this kind of systems can suffer faults due to their discrete nature, they can autonomously change from one working mode to another or they may not obey a command to change the working mode. This two types of faults have been considered in a unified diagnosis architecture based only on HBG-PCs.

The diagnosis process based on HBG-PCs allows to easily detect discrete faults without directly measuring the control signal. Discrete faults are preferred fault candidates so they will be confirmed or discarded in the early stages of the diagnosis process. Only when all discrete fault candidates have been discarded, the isolated parametric faults are considered. Additionally, if and only if some of the HBG-PCs of the system detect a fault, then their residuals are activated in the detection stage, those HBG-PCs will be used for fault isolation while the rest of the HBG-PCs can continue tracking the system behaviour.

Fault detection, isolation and identification with minimal DBNs derived from HBG-PCs. [79, 80] HBG-PCs have been proposed for hybrid systems fault diagnosis of discrete and parametric faults in a unified architecture (Chapter 4). HBG-PCs also present the hybrid systems behaviour as a continuous behaviour in each working mode together with transitions from one mode to another managed by the discrete events, considered in this approach as changes in autonomous or commanded actuators. This simplification allows using DBNs to model the continuous behaviour of each working mode, avoiding discrete nodes in the DBN, that simplifies the DBN inference process and solves most of the convergence problems. Once the HBG-PCs are configured for the actual working mode, minimal DBNs are derived from them. The minimal DBNs integrated with the diagnosis architecture for discrete and parametric faults, allows to perform fault detection, isolation and identification of discrete and parametric faults in hybrid systems using minimal DBNs.

Model multiposition actuators using ON/OFF switching junctions (HBGs) Hybrid Bond Graphs, the modeling technique used in HBG-PC has only ON/OFF switching junctions, so it is possible to directly model ON/OFF actuators. There are multiposition actuators in hybrid systems that connect one and only one among multiple paths. Those actuators cannot be directly modeled with HBGs but Chapter 4 presents a method to model this type of multiposition actuators using ON/OFF switching junctions. This method, easily extends the HBG-PCs technique not only for ON/OFF actuators but also for that type of multiposition actuators.

Case studies The aforementioned contributions have been tested in two case studies:

- **12^{th} Order Electrical Circuit.** The diagnosis architecture for continuous systems using PCs and minimal DBNs have been tested with a 12^{th} order electrical circuit (Chapter 3).
- **Reverse Osmosis System (ROS).** The diagnosis architecture for hybrid systems using HBG-PCs and minimal DBNs have been tested with a Reverse Osmosis System (ROS) (Chapter 5).

6.2 Conclusions

A unified diagnosis architecture for Fault Detecion Isolation and Identification (FDII) for hybrid systems has been proposed in this dissertation. The type of hybrid systems considered have a continuous behaviour commanded by discrete events. The main assumption in this work states that the working mode of the system is known before a fault occurs.

Hybrid Possible Conflicts (HPCs) formalism has been presented in this work. They define the extension of Possible Conflicts (PCs) for hybrid systems. HPCs are based on Hybrid Bond Graphs (HBGs) modeling technique and they are also named as HBG-PCs. The main contribution of HBG-PCs for hybrid systems fault diagnosis is that they do not need to pre-enumerate all the possible working modes in the system. Moreover, they can be built offline and after a mode change is detected, only the HBG-PCs that contain the switching junctions that have changed its state have to be reconfigured. Another advantage of this approach

is that this reconfiguring is local to the affected HBG-PCs. HBG-PCs can be used for Fault Detection and Isolation of discrete and parametric faults. HBG-PCs have allowed introducing the concept of Hybrid Fault Signature Matrix (HFSM), which has made possible integration of discrete and parametric faults in the already known framework of parametric fault isolation from the Fault Signature Matrix (FSM).

Regarding DBNs, the tool chosen to implement the simulation model provided by PCs, the dissertation proposes a method to derive the DBN model (structure and parameters) from the PC. The DBNs derived from PCs are named minimal DBNs as they represent a minimal subsystem with analytical redundancy. Minimal DBNs have been successfully applied for FDII of continuous systems.

Finally, the minimal DBNs have been integrated in the diagnosis architecture for hybrid systems. They are derived from the current set of HBG-PCs and they can be used for FDII of discrete and parametric faults.

The contributions presented in the previous paragraphs have been tested with several case studies. Systems from the hydraulic field and a twelfth order electrical circuit have been used with the FDII architecture for continuous systems. A Reverse Osmosis System has been used to test the FDII architecture for hybrid systems integrating discrete and parametric faults. Moreover, this system has a multiposition actuator that connects one and only one among serveral paths at the same time. The dissertation proposes a method to model that kind of actuators using only ON/OFF switches.

6.3 Future Work

Regarding the contributions obtained in this dissertation, there is some future work and some work already in progress that can be summarized in this section:

- **Parameter Uncertainty.** Some initial studies were done in this field [78].

  Model based diagnosis exploits analytical redundancy of the system generating residuals by comparing the outputs of the system with the predictions of the model. In a fault free system, residual values are theoretically zero. Under ideal conditions, non-zero residual values are indicative of a system failure. Consequently, robust residual generation is an important element on MBD.

  Residual generation becomes a harder problem for nonlinear systems because accurate and precise models are harder to build. Optimal state estimation is not possible for nonlinear systems. For general nonlinear systems, residual sensitivity becomes an issue that is hard to address using analytical and numerical methods. To cope with this problem, structural approach to system analysis have been proposed as an alternative approach. In our work, we adopted this idea, so we used the topological Bond Graph [65, 108] approach to model nonlinear systems.

  Typically, uncertainties when using an observer to track dynamic systems behaviour can be attrributed to two sources: (1) uncertainty in the model, and (2) uncertainty in the measurements [51]. Uncertainties in measurements, attributed to sensors, are typically modeled as zero mean Gaussian noise, and the key is to estimate the unknown variance in the noisy measurements to increase the robustness for fault detection while tracking dynamic systems behaviour. This problem has been addressed for linear systems using Kalman filters, and nonlinear systems using methods such as the Extended Kalman Filter [26], Unscented Kalman Filter [62], and Particle Filter [8]. Modeling uncertainties can be attributed to: (1) structural uncertainty in the system model, i.e., the system equation form or all of the causal relations between system variables may not be explicitly known, and (2) parameter uncertainty, where the system structure is known, but the values of the parameters of the model may not be all known.

  Fault diagnosis of nonlinear systems becomes even harder when the system model is uncertain. Assuming that the structure of the model is known and focusing on the problem when model parameters are uncertain is a realistic scenario. Many real systems are nonlinear and they have some common problems: (1) Numerical convergence issues associated with non linear models, and (2) Their parameter values may only be known approximately. From the point of view of tracking system behaviour and residual generation for nonlinear systems, small changes in system parameter values may produce widely varying dynamic behaviour. Besides, in reality, system measurements are noisy.

Typically, parameter uncertainty is handled by assuming the parameter value comes from a known interval, but the exact value of the parameter is unknown. This assumption together with the method developed by Kam and Dauphin-Tanguy and others [63, 43] in the bond graph framework to model parameter uncertainties are being used to improve the fault diagnosis of non linear systems.

- **HBG-PCs Incremental approach.** The actual HBG-PCs theory assumes there is a valid causal assignment with all switching junctions set to ON to derive the set of HBG-PCs. After that initial step, the working mode is configured in the HBG-PCs.

  When a mode change occurs, the proposal of this disseration states that the new set of HBG-PCs can be built searching in the HBG-PCs structures derived offline assuming all switching junctions are set to ON. The incremental approach supposes that they can be built more efficienly: The actual mode will be configured in the HBG-PCs and after a mode change, the HBG-PCs affected by the change will be updated, moreover, it will be needed to seach in the non parametric HBG-PCs to check whether they are affected by the mode change or not. This will provide a method to efficiently reconfigure the HBG-PCs after a mode change even if the visited mode has never been visited before.

- **Partially causal HBG-PCs approach.** As we have previously said, we assume the HBG modeling the system has a valid causal assignment setting all switching junctions to ON. We are actually working on a less restrictive proposal for models that do not fulfill that requirement. The partially causal approach will not consider the causal assignment of some components to perform the offline analysis of the system to derive the HBG-PCs.

Appendix A
Reverse Osmosis System (ROS) Equations

Mode 1

$$\dot{f}_4 = [e_1 - \frac{e_{10}}{TF} - f_4 * (R_{bwp} + R_{fpump})] * \frac{1}{I_{fpump}} \tag{A.1}$$

$$\dot{e}_{10} = [\frac{f_4}{TF} - \frac{f_{37}}{R_{pipe} * GY} + \frac{e_{22}}{R_{pipe}} + \frac{e_{22} - e_{10}}{R_{brine1}} - \frac{e_{10}}{R_{pipezero1}}] * \frac{1}{C_{res}} \tag{A.2}$$

$$\dot{e}_{22} = [\frac{f_{37}}{R_{pipe} * GY} + \frac{e_{10} - e_{22}}{R_{pipe}} - \frac{e_{22}}{R_{Rmemb}} - \frac{e_{10}}{R_{brine1}}] * \frac{1}{C_{memb}} \tag{A.3}$$

$$\dot{f}_{37} = [e_{34} - f_{37} * R_{rpump} - \frac{1}{GY * R_{pipe}} * [e_{10} + \frac{f_{37}}{GY} - e_{22}]] * \frac{1}{I_{pump}} \tag{A.4}$$

$$\dot{e}_{52} = \frac{f_{48}}{C_k * TF_2} \tag{A.5}$$

$$FFP = f_4 \tag{A.6}$$

$$PBack = e_{10} \tag{A.7}$$

$$PMemb = e_{22} \tag{A.8}$$

$$PPump = \frac{f_{37}}{GY} \tag{A.9}$$

$$Pk = e_{52}Se_{feedpump} = e_1Se_{recircpump} = e_34Sf_{primary} = f_48 \tag{A.10}$$

Mode 2

$$\dot{f}_4 = [e_1 - \frac{e_{10}}{TF} - f_4 * (R_{bwp} + R_{fpump})] * \frac{1}{I_{fpump}} \tag{A.11}$$

$$\dot{e}_{10} = [\frac{f_4}{TF} - \frac{f_{37}}{R_{pipe} * GY} + \frac{e_{22}}{R_{pipe}} + \frac{e_{22} - e_{10}}{R_{brine2}} - \frac{e_{10}}{R_{pipezero1}}] * \frac{1}{C_{res}} \tag{A.12}$$

$$\dot{e}_{22} = [\frac{f_{37}}{R_{pipe} * GY} + \frac{e_{10} - e_{22}}{R_{pipe}} - \frac{e_{22}}{R_{Rmemb}} - \frac{e_{22} - e_{10}}{R_{brine2}}] * \frac{1}{C_{memb}} \tag{A.13}$$

$$\dot{f}_{37} = [e_{34} - f_{37} * R_{rpump} - \frac{1}{GY * R_{pipe}} * [e_{10} + \frac{f_{37}}{GY} - e_{22}]] * \frac{1}{I_{pump}} \tag{A.14}$$

$$\dot{e}_{52} = \frac{f_{42}}{C_k * TF_1} \tag{A.15}$$

$$FFP = f_4 \tag{A.16}$$

$$PBack = e_{10} \tag{A.17}$$

$$PMemb = e_{22} \tag{A.18}$$

$$PPump = \frac{f_{37}}{GY} \tag{A.19}$$

$$Pk = e_{52}Se_{feedpump} = e_1Se_{recircpump} = e_34Sf_{secondary} = f_37 \tag{A.20}$$

Mode Purge

$$\dot{f}_4 = [e_1 - \frac{e_{10}}{TF} - f_4 * (R_{bwp} + R_{fpump})] * \frac{1}{I_{fpump}} \tag{A.21}$$

$$\dot{e}_{10} = [\frac{f_4}{TF} - \frac{e_{10} + PPump}{R_{pipe}} - \frac{e_{10}}{R_{pipezero1}}] * \frac{1}{C_{res}} \tag{A.22}$$

$$\dot{e}_{22} = [\frac{e_{10} + PPump}{R_{pipe}} - \frac{e_{22}}{R_{Rdrain}}] * \frac{1}{C_{memb}} \tag{A.23}$$

$$\dot{e}_{52} = -\frac{e_{52}}{C_k * R} \tag{A.24}$$

$$FFP = f_4 \tag{A.25}$$

$$PBack = e_{10} \tag{A.26}$$

$$PMemb = e_{22} \tag{A.27}$$

$$Pk = e_{52}Se_{feedpump} = e_1 \tag{A.28}$$

Bibliography

[1] B. Pulido C. Alonso-González A. Bregon, G. Biswas and H. Khorasgani. A Common Framework for Compilation Techniques Applied to Diagnosis of Linear Dynamic Systems. *Submitted to IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 2013.

[2] A. Aldea. *The use of scheduling and hierarchical modelling techniques for time-limited Model-based diagnosis*. PhD thesis, Heriot-Watt University, Edimburgh, Scotland, UK., 1994.

[3] C. J. Alonso-González, N. Moya, and G. Biswas. Dynamic Bayesian Network Factors from Possible Conflicts for Continuous System Diagnosis. In *Proceedings of the 14th international conference on Advances in artificial intelligence: spanish association for artificial intelligence*, CAEPIA'11, pages 223–232, Berlin, Heidelberg, 2011. Springer-Verlag.

[4] C.J. Alonso-González, N. Moya, and G. Biswas. Factoring Dynamic Bayesian Networks Using Possible Conflicts. In *Proceeding of the 21th International Workshop on Principles of Diagnosis, DX10*, October 2010.

[5] C.J. Alonso-González, N. Moya, and G. Biswas. Dynamic Bayesian Network Factors from Possible Conflicts for Continuous System Diagnosis. In J.A. Lozano, J.A. Gámez, and J.A. Moreno, editors, *Advances in Artificial Intelligence*, volume 7023 of *Lecture Notes in Computer Science*, pages 223–232. Springer Berlin Heidelberg, 2011.

[6] R. Alur, C. Courcoubetis, T. Henzinger, and P. Ho. Hybrid automata: An algorithmic approach to the specification and verification of hybrid systems. In *Hybrid Systems*, volume 736 of *Lecture Notes in Computer Science*, pages 209–229. Springer Berlin / Heidelberg, 1993.

[7] J. Armengol, A. Bregon, T. Escobet, M. Krysander, M .Nyberg, X. Olive, B. Pulido, and L. Trav-Massuys. Minimal Structurally Overdetermined sets for residual generation: A comparison of alternative approaches. In *Proceedings of the 7th IFAC Symposium on Fault Detection, Supervision and Safety of Thecnical Processes, SAFEPROCESS09*, pages 1480–1485, Barcelona, 2009.

[8] M.S. Arulampalam, S. Maskell, and N. Gordon. A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking. *IEEE Transactions on Signal Processing*, 50:174–188, 2002.

[9] M. Bayoudh, L. Travé-Massuyès, and X. Olive. Coupling Continuous and Discrete Event System Techniques for Hybrid System Diagnosability Analysis. In *Proceeding of the 2008 conference on ECAI 2008: 18th European Conference on Artificial Intelligence*, pages 219–223, Amsterdam, The Netherlands, The Netherlands, 2008. IOS Press.

[10] M. Bayoudh, L. Travé-Massuyès, and X. Olive. Diagnosis of a Class of Non Linear Hybrid Systems by On-line Instantiation of Parameterized Analytical Redundancy Relations. In *20th International Workshop on Principles of Diagnosis, DX09*, pages 283–289, Stockholm, Sweden, 2009.

[11] E. Benazera and L. Travé-Massuyès. Set-theoretic estimation of hybrid system configurations. *Trans. Sys. Man Cyber. Part B*, 39:1277–1291, October 2009.

[12] G. Biswas, M.O. Cordier, J. Lunze, Travé-Massuyès, and M. Staroswiecki. Diagnosis of complex systems: bridging the methodologies of the FDI and DX communities. *Part B: Cybernetics, IEEE Transactions on Systems, Man, and Cybernetics*, 34(5):2159–2162, 2004.

[13] G. Biswas, G. Simon, N. Mahadevan, S. Narasimhan, J. Ramirez, and G. Karsai. A robust method for hybrid diagnosis of complex systems. In *5th IFAC Symposium on Fault Detection, Supervision and Safety of Technical Processes (SAFEPROCESS)*, pages 1125–1131, 2003.

[14] M. Blanke, M. Kinnaert, J. Lunze, M. Staroswiecki, and J. Schröder. *Diagnosis and Fault-Tolerant Control*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006.

[15] W. Borutzky. Representing discontinuities by means of sinks of fixed causality. In F. E. Cellier and J. J. Granda, editors, *Proceedings of the International Conference on Bond Graph Modeling, ICBGM95*, pages 65–72. SCS Publishing, January 15-18 1995.

[16] A. Bregon. *Integration of FDI and DX techniques within Consistency-based Diagnosis with Possible Conflicts*. PhD thesis, Universidad de Valladolid, Valladolid, Spain, May 2010.

[17] A. Bregon, C. Alonso, G. Biswas, B. Pulido, and N. Moya. Fault Diagnosis in Hybrid Systems using Possible Conficts. In *Proc. of the 8th IFAC Symposium on Fault Detection, Supervision and Safety of Technical Processes, SAFEPROCESS12*, Mexico City, Mexico, 2012.

[18] A. Bregon, C. J. Alonso-González, G. Biswas, B. Pulido, and N. Moya. Hybrid Systems Fault Diagnosis with Possible Conflicts. In *Proceedings of the 22nd International Workshop on Principles of Diagnosis, DX11*, pages 36–43, Murnau, Germany, 2011.

[19] A. Bregon, G. Biswas, and B. Pulido. A Decomposition Method for Nonlinear Parameter Estimation in TRANSCEND. *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on*, 42(3):751–763, 2012.

[20] A. Bregon, G. Biswas, B. Pulido, C. Alonso-Gonzalez, and H. Khorasgani. A Common Framework for Compilation Techniques Applied to Diagnosis of Linear Dynamic Systems. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, submited.

[21] A. Bregon, M. Daigle, I. Roychoudhury, G. Biswas, X. Koutsoukos, and B. Pulido. An Event-based Distributed Diagnosis Framework using Structural Model Decomposition. *Submitted to Artificial Intelligence. ARTINT-D-12-00113R1*, 2013.

[22] A. Bregon, B. Pulido, and G. Biswas. Efficient On-line Fault Isolation and Identification in TRANSCEND for nonlinear systems. In *Proceedings of the 20th International Workshop on Principles of Diagnosis, DX09*, pages 291–298, Stockholm, Sweden, 2009.

[23] A. Bregon, B. Pulido, G. Biswas, and X. Koutsoukos. Generating Possible Conflicts from Bond Graphs Using Temporal Causal Graphs. In *Proceeding of the 23rd European Conference on Modelling and Simulation, ECMS09*, Madrid, Spain, 2009.

[24] J.F. Broenink. Introduction to Physical Systems Modelling with Bond Graphs. *SiE Whitebook on Simulation methodologies*, 1999.

[25] F.E. Cellier and R.T. McBride. Object-oriented Modeling of Complex Physical Systems Using the Dymola Bond-graph Library. In *In Proceedings of the 6th SCS Intl. Conf. on Bond Graph Modeling and Simulatio, ICBGM03*, pages 157–162, Orlando, Florida, US, 2003.

[26] E. N. Chatzi and A. W. Smyth. The unscented kalman filter and particle filter methods for nonlinear structural system identification with non-collocated heterogeneous sensing. *Structural Control and Health Monitoring*, 16(1):99–123, 2009.

[27] Mark Chavira and Adnan Darwiche. Compiling Bayesian networks using variable elimination. In *Proceedings of the 20th International Joint Conference on Artifical Intelligence*, IJCAI'07, pages 2443–2449, San Francisco, CA, USA, 2007. Morgan Kaufmann Publishers Inc.

[28] L. Chittaro, G. Guida, C. Tasso, and E. Toppano. Functional and teleological knowledge in the multimodeling approach for reasoning about physical systems: A case study in diagnosis. *IEEE Transactions on Systems, Man and Cybernetics*, 6(23):1718–1751, 1993.

[29] L. Chittaro, R. Ranon, and J.L. Cortes. Ship over troubled waters: Functional reasoning with influences applied to the diagnosis of a marine technical system. In *In Proceedings of the 7th International Workshop on Principles of Diagnosis, DX-96*, pages 69–78, Val Morin, Quebec, Canada, 1996.

[30] E. Y. Chow and A. S. Willsky. Analytical redundancy and the design of robust failure detection systems. *IEEE Transactions on Automatic Control (AC)*, 7(29):603–614, 1984.

[31] V. Cocquempot, T. El Mezyani, and M. Staroswiecki. Fault detection and isolation for hybrid systems using structured parity residuals. In *Control Conference, 2004. 5th Asian*, volume 2, pages 1204 – 1212 Vol.2, july 2004.

[32] L. Console. Model-based diagnosis history and state of the art. MONET Excellence Network. Summer School, Bertinoro, Italy, Mayo 2000.

[33] M.O. Cordier, P. Dague, F. Lvy, J. Montmain, M. Staroswiecki, and Travé-Massuyès. Conflicts versus Analytical Redundancy Relations: A comparative analysis of the model based diagnosis approach from the Artificial Intelligence and Automatic Control perspectives. *Part B: Cybernetics, IEEE Transactions on Systems, Man, and Cybernetics*, 34(5):2163–77, 2004.

[34] P.J.L. Cuijpers and M.A. Reniers. Hybrid process algebra. *Journal of Logic and Algebraic Programming*, 62(2):191 – 245, 2005.

[35] M. Daigle, X. Koutsoukos, and G. Biswas. An event-based approach to hybrid systems diagnosability. In *In Proceedings of the 19th International Workshop on Principles of Diagnosis*, pages 47–54, 2008.

[36] R. Davis. Expert systems: Where are we? And where do we go from here? *Artificial Intelligence Magazine*, 3(2):3–22, 1982.

[37] R. Davis. Diagnostic Reasoning Based on Structure and Behavior. *Artificial Intelligence Magazine*, 24:347–410, 1984.

[38] R. Davis and W. Hamscher. *Model-based reasoning: troubleshooting*, pages 297–346. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1988.

[39] J. de Kleer and J. Kurien. Fundamentals of model-based diagnosis. In *In Proceedings of the 5th IFAC Symposium on Fault Detection, Supervision and Safety for Technical Processes, SAFEPROCESS03*, Washington D.C., USA, 2003.

[40] J. de Kleer, A.K. Mackworth, and R. Reiter. *Readings in Model Based Diagnosis*, chapter Characterizing diagnosis and systems, pages 54–65. Morgan-Kauffman, 1992.

[41] R. Dearden and D. Clancy. Particle filters for real-time fault detection in planetary rovers. In *The 12th International Workshop on Principles of Diagnosis, DX01*, pages 1–6, 2001.

[42] M. Delgado and H. Sira-Ramirez. Modeling and simulation of switch regulated dc-to-dc power converters of the boost type. In *Proceedings of the First IEEE International Caracas Conference on Devices, Circuits and Systems*, pages 84 –88, dec 1995.

[43] M.A. Djeziri, R. Merzouki, B.O. Bouamama, and G. Dauphin-Tanguy. Robust Fault Diagnosis by Using Bond Graph Approach. *Mechatronics, IEEE/ASME Transactions on*, 12(6):599 –611, 2007.

[44] O. Dressler. Model-based Diagnosis on Board: Magellan-MT Inside. In *In Working Notes of the International Workshop on Principles of Diagnosis, DX94*, Goslar, Germany, 1994.

[45] O. Dressler. On-line diagnosis and monitoring of dynamic systems based on qualitative models and dependency-recording diagnosis engines. In *In Proceedings of the Twelfth European Conference on Artificial Intelligence, ECAI-96*, pages 461–465, 1996.

[46] O. Dressler and P. Struss. *The consistency-based approach to automated diagnosis of devices*, pages 267–311. Center for the Study of Language and Information, Stanford, CA, USA, 1996.

[47] J.P. Ducreux, G. Dauphin-Tanguy, and C. Rombaut. Representing discontinuities by means of sinks of fixed causality. In F. E. Cellier and J. J. Granda, editors, *roceedings of the International Conference on Bond Graph Modeling, ICBGM93*, volume 25, pages 132–136. SCS Publishing, 1993.

[48] G. Friedrich, G. Gottlob, and W. Nejdl. *Physical impossibility instead of fault models*, pages 159–164. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1992.

[49] E. Alcorta García and P. M. Frank. Deterministic nonlinear observer-based approaches to fault diagnosis: A survey. *Control Engineering Practice*, 5(5):663 – 670, 1997.

[50] P.J. Gawthrop. Hybrid Bond Graphs Using Switched I and C Components, 1997.

[51] A. Gelb. *Applied Optimal Estimation*. MIT Press, Cambridge, MA, USA, 1979.

[52] E. Gelso, S. Castillo, and J. Armengol. Structural analysis and consistency techniques for robust model-based fault diagnosis. Technical Report 20, Institut dInformática i Aplicacions, Universitat de Girona, 2008.

[53] E. R. Gelso, G. Biswas, S. M. Castillo, and J. Armengol. A Comparison of Two Methods for Fault Detection: a Statistical Decision, and an Interval-based Approach. In *Proceeding of the 19th International Workshop on Principles of Diagnosis, DX08*, Blue Mountains, Australia, 2008.

[54] J. Gertler. *Fault Detection and Diagnosis in Engineering Systems*. Number 9780824794279. CRC Press, first edition edition, May 1998.

[55] T. Guckenbiehl and G. Schaefer-Richter. SIDIA: Extending prediction based diagnosis to dynamic models. In *In Proceedings of the 1st International Workshop on Principles of Diagnosis, DX90*, pages 74–82, Stanford, California, USA, 1990.

[56] W. Hamscher, L. Console, and J. de Kleer. *Readings in model-based diagnosis*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1992.

[57] L. J. Herrera, H. Pomares, I. Rojas, O. Valenzuela, and A. Prieto. TaSe, a Taylor series-based fuzzy system model that combines interpretability and accuracy. *Fuzzy Sets and Systems*, 153(3):403–427, Aug. 2005.

[58] M.W. Hofbaur and B.C. Williams. Hybrid estimation of complex systems. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, 34(5):2178 –2191, oct. 2004.

[59] R. Isermann. Fault diagnosis of machines via parameter estimation and knowledge processing: tutorial paper. *Automatica*, 29(4):815–835, jul 1993.

[60] R. Isermann. Supervision, fault-detection and fault-diagnosis methods – An introduction. *Control Engineering Practice*, 5(5):639 – 652, 1997.

[61] R. Isermann and P. Ballé. Trends in the application of model-based fault detection and diagnosis of technical processes. *Control Engineering Practice*, 5(5):709–719, 1997.

[62] S. J. Julier and J. K. Uhlmann. Unscented Filtering and Nonlinear Estimation. In *Proceedings of the IEEE*, volume 92, pages 401–422, March 2004.

[63] C. S. Kam and G. Dauphin-Tanguy. Bond Graph models of structured parameter uncertainties. *Journal of the Franklin Institute*, 342(4):379 – 399, 2005.

[64] D.C. Karnopp, D.L. Margolis, and R.C. Rosenberg. *System Dynamics: Modeling and Simulation of Mechatronic Systems*. John Wiley & Sons, Inc., New York, NY, USA, 2006.

[65] D.C. Karnopp, R. C. Rosenborg, and D.L. Margolis. *Systems Dynamics. A Unified approach*. Wiley & Sons, 3rd ed. edition, 2000.

[66] J. De Kleer and B.C. Williams. Diagnosis with behavioral modes. In *IJCAI'89: Proceedings of the 11th International Joint Conference on Artificial Intelligence*, pages 1324–1330, San Francisco, CA, USA, 1989. Morgan Kaufmann Publishers Inc.

[67] D. Koller and U. Lerner. *Sequential Monte Carlo Methods in Practice*, chapter Sampling in factored dynamic systems. Springer, 2001.

[68] J. Kościelny and K. Zakroczymski. Fault isolation method based on time sequences of symptom appearance. In *In Proceedings of the th IFAC Safaprocess'2000*, Budapest, Hungary, 2000.

[69] J. M. Kościelny. Fault isolation in industrial processes by the dynamic table of states method. *Automatica*, 31(5):747–753, May 1995.

[70] X. Koutsoukos, J. Kurien, and F. Zhao. Estimation of Distributed Hybrid Systems Using Particle Filtering Methods. In *In Hybrid Systems: Computation and Control (HSCC 2003). Springer Verlag Lecture Notes on Computer Science*, pages 298–313. Springer, 2003.

[71] B. Kuipers. Qualitative reasoning: modeling and simulation with incomplete knowledge. *Automatica*, 25(4):571–585, 1989.

[72] U. Lerner, R.Paar, D. Koller, and G. Biswas. Bayesian Fault Detection and Diagnosis in Dynamic Systems. In *Proccedings of the AAAI/IAAI*, pages 531–537, 2000.

[73] E. Loiez and P. Taillibert. Polynomial temporal band sequences for analog diagnosis. In *In Proceedings of the 15th International Joint Conference on Artificial Intelligence, IJCAI97*, pages 474–479, Nagoya, Japan, 1997.

[74] J. Lunze. Diagnosis of Quantised Systems by Means of Timed Discrete-Event Representations. In *Proceedings of the Third International Workshop on Hybrid Systems: Computation and Control*, HSCC '00, pages 258–271, London, UK, 2000. Springer-Verlag.

[75] S. Mcilraith. Diagnosing Hybrid Systems: a Bayesian Model Selection Approach. In *Proceedings of the 11th International Workshop on Principles of Diagnosis (DX'00)*, pages 140–146, 2000.

[76] P. Mosterman and G. Biswas. Diagnosis of continuous valued systems in transient operating regions. *IEEE Transactions on Systems, Man, and Cybernetics*, 29(6):554–565, 1999.

[77] P.J. Mosterman and G. Biswas. Behavior Generation using Model Switching - A Hybrid Bond Graph Modeling Technique. In *Society for Computer Simulation*, pages 177–182. SCS publishing, 1994.

[78] N. Moya, G. Biswas, C.J. Alonso-González, and X. Koutsoukos. Structural Observability. Application to decompose a System with Possible Conflicts. In *Proceedings of the 21th International Workshop on Principles of Diagnosis, DX10*, pages 241–248, June 2010.

[79] N. Moya, A. Bregon, C. J. Alonso-González, B. Pulido, and G. Biswas. Extendng Hybrid Possible Conflicts to Diagnose Discrete Faults. In *Proceedings of the 23rd Int. WS on Pples. of Diagnosis, DX12*, pages 51–58, Great Malvern, UK, Jul-Aug 2012.

[80] N. Moya, B. Pulido, C. J. Alonso-González, A. Bregon, and D. Rubio. Automatic Generation of Dynamic Bayesian Networks for Hybrid Systems Fault Diagnosis. In *Proceedings of the 23rd Int. WS on Pples. of Diagnosis, DX12*, pages 59–66, Great Malvern, UK, Jul-Aug 2012.

[81] K. P. Murphy. *Dynamic Bayesian Networks: Representation, Inference and Learning*. PhD thesis, University of California, Berkeley, 2002.

[82] S. Narasimhan. *Model based diagnosis of hybrid systems*. PhD thesis, Graduate School of the Vanderbilt University, Nashville, TN, US, Aug 2002.

[83] S. Narasimhan. Automated Diagnosis of Physical Systems. In *Proceedings of ICALEPCS07*, pages 701–705, 2007.

[84] S. Narasimhan and G. Biswas. Model-Based Diagnosis of Hybrid Systems. *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans*, 37(3):348–361, May 2007.

[85] S. Narasimhan and L. Brownston. HyDE: a general framework for stochastic and hybrid model-based diagnosis. In *In Proceedings of the 18th International Workshop on Principles of Diagnosis, DX07*, pages 162–169, Nashville, TN, US, 2007.

[86] S. Narasimhan, R. Dearden, and E. Benazera. Combining particle filters and consistency-based approaches for monitoring and diagnosis of stochastic hybrid systems. In *In Proceedings of the 15th International Workshop on Principles of Diagnosis, DX04*, Carcassonne, France, 2004.

[87] K. Ogata. *Modern Control Engineering*. Prentice Hall, 4th edition, 2001.

[88] R. J. Patton, P. M. Frank, and R. N. Clark. *Issues of Fault Diagnosis for Dynamic Systems*. Springer, 2000.

[89] A. Pernestal, M. Nyberg, and B. Wahlberg. A bayesian approach to fault isolation with application to diesel engine diagnosis. In *In Proceedings of the 17th International Workshop on Principles of Diagnosis, DX06*, pages 211–218, Penaranda de Duero, Spain, 2006.

[90] K. D. Pickering, K. Wines, G. M. PAriani, L. A. Franks, J. Yeh, B. W. Finger, M. L. Campbell, C. E. Verostko, C. Carrier, J. C. Gandhi, and L. M. Vega. Early results of an ontegrated water recovery system test. In *In Proceedings of the 29th International Conference on Environmental Systems*, Orlando, FL, 2001.

[91] A. D. Pouliezos and G. S. Stavrakakis. *Real Time Fault Monitoring of Industrial Processes*. Kluwer Academic Publishers, Norwell, MA, USA, 1994.

[92] V. Puig, J. Quevedo, T. Escobet, and B. Pulido. On the Integration of Fault Detection and Isolation in Model Based Fault Diagnosis. In *Proceeding of the 16th International Workshop on Principles of Diagnosis, DX05*, pages 227 – 232, Pacific Grove, CA, USA, 2005.

[93] B. Pulido and C. Alonso. An alternative approach to dependency-recording engines in consistency-based diagnosis. *In Artificial Intelligence: Methodology, Systems, and Applications, AIMSA-00*, 1904 of LNAI:111–120, 2000.

[94] B. Pulido, C. Alonso, and F. Acebes. Lessons learned from diagnosing dynamic systems using Possible Conflicts and Quantitative Models. In *Engineering of Intelligent Systems. Fourteenth International Conference on Industrial and Engineering Applications of Artificial Intelligence and Expert Systems (IEA/AIE-2001)*, volume 2070 of *Lecture Notes in Artificial Intelligence*, pages 135–144, Budapest, Hungary, 2001.

[95] B. Pulido and C. Alonso-González. Possible Conflicts: A compilation technique for consistency-based diagnosis. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, 34(5):2192–2206, October 2004.

[96] B. Pulido, A. Bregon, and C. Alonso-González. Analyzing the Influence of Differential Constraints in Possible Conflict and ARR Computation. In *Proceedings of the 13th Conference of the Spanish Association for Artificial Intelligence, CAEPIA 2009*, Seville, Spain, 2009.

[97] B. Pulido, A. Bregon, and C. Alonso-González. Analyzing the influence of differential constraints in Possible Conflicts and ARR Computation. In P. Meseguer, L. Mandow, and R. Gasca, editors, *Current Topics in Artificial Intelligence*, volume 5988 of *LNCS*, pages 11–21. Springer Berlin, 2010.

[98] O. Raiman. Diagnosis as a trial. In *Proc. of the Zero-th Int. WS on Pples. of Diagnosis (DX-89)*, IBM Scientific Center, Paris, France, 1989.

[99] O. Raiman, J. de Kleer, V. Saraswat, and M. Shirley. *Characterizing non-intermittent faults*, pages 170–175. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1992.

[100] R. Reiter. A theory of diagnosis from first principles. *Artificial Intelligence*, 32:57–95, 1987.

[101] Th. Rienmüller, M. Bayoudh, M.W. Hofbaur, and L. Travé-Massuyès. Hybrid Estimation through Synergic Mode-Set Focusing. In *Proc. of IFAC Safeprocess'09*, pages 1480–1485, Barcelona, Spain, 2009.

[102] I. Roychoudhury. *Distributed Diagnosis of Continuous Systems: Global diagnosis through local analysis*. PhD thesis, Graduate School of the Vanderbilt University, August 2009.

[103] I. Roychoudhury, G. Biswas, and X. Koutsoukos. Comprehensive Diagnosis of Continuous systems Using Dynamic Bayes Nets. In *Proceeding of the 19th International Workshop on Principles of Diagnosis, DX08*, Blue Mountains, Australia, September 2008.

[104] I. Roychoudhury, G. Biswas, and X. Koutsoukos. Designing Distributed Diagnosers for Complex Continuous Systems. *Automation Science and Engineering, IEEE Transactions on*, 6(2):277 –290, april 2009.

[105] I. Roychoudhury, G. Biswas, and X. Koutsoukos. Factoring Dynamic Bayesian Networks based on Structural Observability. In *In 48th IEEE Conference on Decision and Control (CDC 2009)*, 2009.

[106] I. Roychoudhury, M. J. Daigle, G. Biswas, and Xenofon Koutsoukos. Efficient simulation of hybrid systems: A hybrid bond graph approach. *SIMULATION: Transactions of the Society for Modeling and Simulation International*, April 2010.

[107] S.J. Russell and P. Norvig. *Artificial Intelligence: A Modern Approach*. Prentice-Hall Inc., 2nd ed. edition, 1995.

[108] A. K. Samantaray and B. O. Bouamama. *Model-based Process Supervision: A Bond Graph Approach*. Springer-Verlag, London, 2008.

[109] M. Sampath, R. Sengupta, S. Lafortune, K. Sinnamohideen, and D. Teneketzis. Failure diagnosis using discrete-event models. *IEEE Transactions on Control Systems Technology*, 2(4):105–124, Mar. 1996.

[110] M. Staroswiecki. Model based FDI: the control approach. In *Plenary lecture, Bridge Workshop*, Sancicario, Italy, March, 2001.

[111] M. Staroswiecki. *Fault Diagnosis and Fault Tolerant Control*, chapter Structural Analysis for Fault Detection and Isolation and for Fault Tolerant Control. Encyclopedia of Life Support Systems. Eolss Publishers, Oxford, UK, 2002.

[112] M. Staroswiecki. A structural view of fault-tolerant estimation. In *Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering*, volume 221, pages 905–914, 2007.

[113] P. Struss. *Diagnosis as a process*, pages 408–418. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1992.

[114] P. Struss. Model-based diagnosis for industrial applications. In Colloquium-Applications of Model-based Reasoning, 1997.

[115] P. Struss. *Model-based Problem Solving*, volume 3 of *Foundations of Artificial Intelligence*, chapter 10, pages 395–465. Elsevier B.V., 2008.

[116] P. Struss and O. Dressler. Physical negation: Introducing fault models into the General Diagnostic Engine. In *In Proceedings of the 11th International Joint Conference on Artifical Intelligence, IJCAI89*, pages 1318–1323, Detroit, Michigan, USA, 1989.

[117] L. Travé-Massuyès and Bridge-Task-Group. Deliverable MBD2: Common Diagnostic Framework Specification. Technical report, MONET2: Network of Excellence on Model Based Systems and Qualitative Reasoning, 2003.

[118] L. Travé-Massuyès, T. Escobet, and X. Olive. Diagnosability analysis based on component supported Analytical Redundancy Relations. *IEEE Transactions on Systems, Man and Cybernetics, Part A : Systems and Humans*, (6):1146–1160, Nov 2006.

[119] L. Travé-Massuyès, T. Escobet, and J. Quevedo. The causal qualitative fault detection and diagnosis system CAEN and its application in the gas turbine domain. In QMFDI Vacation School. DAMADICS Excellence Network, 2000.

[120] P. Terenziani V. Brusoni, L. Console and D. Dupre. A spectrum of definitions for temporal Model-based Diagnosis. In *In Proceedings of the 7th International Workshop on Principles of Diagnosis, DX-96*, pages 44–52, Val Morin, Quebec, Canada, 1996.

[121] V. Venkatasubramanian, R. Rengswamy, K. Yin, and S.N. Kavuri. A review of process fault detection and diagnosis. Part I: Quantitative model-based methods. *Computers and Chemical Engineering*, 27:293–311, 2003.

[122] V. Venkatasubramanian, R. Rengswamy, K. Yin, and S.N. Kavuri. A review of process fault detection and diagnosis. Part II: Qualitative models and search strategies. *Computers and Chemical Engineering*, 27:313–326, 2003.

[123] V. Venkatasubramanian, R. Rengswamy, K. Yin, and S.N. Kavuri. A review of process fault detection and diagnosis. Part III: Process history based methods. *Computers and Chemical Engineering*, 27:327–346, 2003.

[124] V. Verma, G. Gordon, R. Simmons, and S. Thrun. Real-time fault diagnosis [robot fault diagnosis]. *Robotics & Automation Magazine*, 11(1):56–66, June 2004.

[125] M. Wang and Dearden R. Detecting and Learning Unknown Fault States in Hybrid Diagnosis. In *Proceeding of the 20th International Workshop on Principles of Diagnosis, DX09*, pages 19–26, Stockholm, Sweden, 2009.

[126] W. Wang, L. Li, D. Zhou, and K. Liu. Robust state estimation and fault diagnosis for uncertain hybrid nonlinear systems. *Nonlinear Analysis: Hybrid Systems*, 1(1):2–15, 2007.

[127] T. Washio, M. Sakuma, and M. Kitamura. A new approach to quantitative and credible diagnosis for multiple faults of components and sensors. *Artif. Intell.*, 91(1):103–130, 1997.

[128] B.C. Williams and B. Millar. Decompositional Model-based Learning and its analogy to diagnosis. In *Proceedings of (AAAI-98)*, 1998.

[129] F. Zhao, X. Koutsoukos, H. Haussecker, J. Reich, and P. Cheung. Monitoring and fault diagnosis of hybrid systems. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, 35(6):1225 –1240, dec. 2005.