

PONTIFICIA UNIVERSIDAD CATOLICA DE CHILE ESCUELA DE INGENIERIA

MEDICIÓN INFERENCIAL EN SISTEMAS DE TRANSPORTE PÚBLICO EMPLEANDO MODELOS DINÁMICOS HÍBRIDOS

FRANCISCO REYES LEIVA

Tesis presentada a la Dirección de Investigación y Postgrado como parte de los requisitos para optar al grado de Magister en Ciencias de la Ingeniería

Profesor Supervisor: ALDO CIPRIANO ZAMORANO

Santiago de Chile, Abril 2012

© MMXII, FRANCISCO REYES LEIVA



PONTIFICIA UNIVERSIDAD CATOLICA DE CHILE ESCUELA DE INGENIERIA

MEDICIÓN INFERENCIAL EN SISTEMAS DE TRANSPORTE PÚBLICO EMPLEANDO MODELOS DINÁMICOS HÍBRIDOS

FRANCISCO REYES LEIVA

Miembros del Comité: ALDO CIPRIANO ZAMORANO RICARDO GIESEN ENCINA JUAN IGNACIO YUZ EISSMANN JUAN CARLOS MUÑOZ ABOGABIR

Tesis presentada a la Dirección de Investigación y Postgrado como parte de los requisitos para optar al grado de Magister en Ciencias de la Ingeniería

Santiago de Chile, Abril 2012

© MMXII, FRANCISCO REYES LEIVA

A mi familia, Camila y amigos

AGRADECIMIENTOS

Agradezco a mi profesor supervisor por todo su apoyo y conocimiento compartido con mi persona durante el tiempo que se realizó este trabajo. Su experiencia me ayudó a clarificar mis ideas y siempre mirar un poco más allá.

También me gustaría agradecer a mi círculo más cercano que me dio su apoyo mediante refuerzos positivos y cariño incondicional. En especial quiero agradecer a Camila, mi familia y mis amigos Renato, Mariajesus, Joaquín, Kaipi, Lurys, Toño y Santiago.

Finalmente agradecer a FONDEF por financiar el proyecto ADI-32 en el que se enmarca esta investigación.

Índice General

AGRADEC	CIMIENTOS	IV
Lista de Fig	guras	VII
Lista de Tal	plas	IX
RESUMEN	1	X
ABSTRAC	Τ	XI
Capítulo 1.	INTRODUCCIÓN	1
1.1. Mo	otivación	1
1.2. De	scripción del Problema	2
1.3. Ob	jetivos	3
1.4. Pri	ncipales Contribuciones	4
1.5. Or	ganización del Documento	5
Capítulo 2.	MARCO TEÓRICO	6
2.1. Sis	temas Híbridos	6
2.2. Est	timación de Estado en Sistemas Híbridos	8
2.2.1.	Métodos basados en Multimodelos	9
2.2.2.	Métodos basados en MHE	10
2.2.3.	Métodos basados en Filtro de Partículas	13
2.2.4.	Análisis Comparativo	16
2.3. Ide	entificación de Sistemas Híbridos	17
2.3.1.	Planteamiento del Problema	18
2.3.2.	Algoritmo de Identificación	19
2.3.3.	Otras Metodologías	20
2.3.4.	Análisis Comparativo	22
2.4. He	rramientas Computacionales	23

Capítulo 3.	MEDICIÓN INFERENCIAL EN SISTEMA HIDRÁULICO	25
3.1. Plan	ta a Analizar	25
3.1.1.	Representación MLD	27
3.1.2.	Representación PWA	28
3.2. Estir	nación de Estado	28
3.2.1.	Estimación de Horizonte Móvil	29
3.2.2.	Filtro de Partículas	29
3.2.3.	Análisis Comparativo	30
3.3. Ident	tificación de Sistemas	32
3.4. Obse	ervaciones Respecto del Problema	34
Capítulo 4.	APLICACIÓN SISTEMA TRANSPORTE	36
4.1. Mod	elo	36
4.2. Plan	teamiento del Problema	42
4.3. Expe	erimentos y Resultados	44
4.3.1.	Análisis de Sensibilidad	45
Capítulo 5.	CONCLUSIONES Y TRABAJO FUTURO	49
5.1. Resu	Iltados y Comentarios Generales	49
5.2. Tema	as de Investigación Futura	51
BIBLIOGRA	FÍA	52
ANEXO A.	PRODUCTOS DE LA TESIS	60
A.1. Pape	er ITS Magazine	60
ANEXO B.	CÓDIGO HYSDEL EJEMPLO BENCHMARK	74

Lista de Figuras

2.1.	Sistema pendular con obstáculo. La línea punteada marca la frontera que define los dos modos de operación del sistema, donde el obstáculo genera una disminución	1
	del largo de la cuerda.	6
2.2.	Observador basado en Multi-Modelos. El observador consiste en 2 estimadores, uno discreto y uno continuo, separando así el problema de estimación	9
2.3.	Algoritmo Filtro de Partículas. La estimación pasada genera a través de la predicción una serie de partículas que son remuestreadas según su semejanza con la medición para así obtener una función de distribución de probabilidad del estado actual	13
		15
3.1.	Esquema Planta 4 estanques. Los estanques se alimentan de las bombas $a \ge b \ge$ el flujo es direccionado por las válvulas $\gamma_a \ge \gamma_b$	25
3.2.	Esquema de variables en planta. La acción de una de las válvulas actúa como perturbación, lo que también es estimado por el observador.	29
3.3.	Estimación de las variables de estado continuas. La estimación de estas variables posee un error bajo, cercano a 1.4×10^{-2} , principalmente debido al uso de un horizonte de estimación que capta la dinámica del sistema.	30
3.4.	La estimación de δ_2 se ve afectada fuertemente por el ruido presente en el sistema presentando un comportamiento de tipo <i>chattering</i>	30
3.5.	Estimación de las variables de estado continuas. La estructura del sistema perjudica más a la estimación de h_3 dado que δ_2 es desconocido; aún así la estimación obtenida es precisa y valida por lo tanto el uso de Filtro de Partículas.	31
3.6.	Resultado de estimación por Filtro de Partículas. Los resultados muestran un pequeño desfase en la detección del cambio en δ_2 . El buen desempeño afecta de igual manera la estimación de h_3 .	31

3.7.	El espacio de estados continuo se particiona en 4 regiones que definen los modos	
	de operación de la planta. Los modos de operación reflejan el efecto de la	
	conexión entre los tanques.	32
3.8.	El proceso de identificación determina aproximadamente modo las regiones del	
	sistema, incluso en presencia de ruido.	34
4.1.	Interfaz de usuario de SimDinMetro. El simulador se ha modificado para permitir	
	una programación más ágil de otros sistemas de transporte para una posible	
	aplicación en minería.	42
4.2.	Estructura del sistema a observar. Las variables de entrada y salida se asumen	
	conocidas en todo momento, lográndose estimar los pasajeros que esperan en las	
	estaciones y viajan en los trenes, variables que a su vez también alimentan el	
	observador	43
4.3.	Etapas en el proceso de estimación. Las variables estimadas deben utilizarse en	
	estimaciones futuras, introduciendo un comportamiento dinámico y retroalimentad	lo
	a la estimación.	44
4.4.	Evolución del número de pasajeros en los trenes 2 y 16 a lo largo de la línea	
	de Metro. Es posible obtener una estimación precisa, donde los límites de la	
	capacidad de los trenes de pasajeros (1300) se respetan	46
4.5.	Evolución del número de pasajeros en las estaciones 12 y 16 de la línea de Metro	
	a medida que los trenes dejan las estaciones. La estimación difiere del valor real	
	debido a la incertidumbre en la matriz origen-destino usada para simulación y	
	aquella usada para la estiamción.	47
4.6.	Error medio de estimación con barras de error que indican la desviación estándar.	
	El error de estimación es mayor en las estaciones debido a la incertidumbre de	
	la matriz origen-destino, sin embargo este error disminuye en los extremos de la	
	línea	48

Lista de Tablas

2.1.	Comparación de Métodos de Estimación	17
2.2.	Comparación de Métodos de Identificación	22
3.1.	Parámetros del sistema	27
3.2.	Cotas y punto de operación del sistema	28
3.3.	Comparación resultados de estimación	32
3.4.	Parámetros a identificar	33
3.5.	Parámetros identificados	34
4.1.	Parámetros del modelo	37
4.2.	Variables de estado del modelo	38
4.3.	Análisis de sensibilidad en σ_T y la diferencia entre la ODM de simulación y	
	estimación	46

RESUMEN

Luego de la puesta en marcha el año 2007 del Transantiago, el sistema de transporte público en la ciudad de Santiago, han surgido diversos esfuerzos desde el área académica para contribuir al mejoramiento del servicio que se entrega a los usuarios. Debido a que variadas aplicaciones ligadas al control del sistema de buses y metro requieren del conocimiento de variables del sistema que no siempre es posible medir, ya sea por inexistencia de sensores o por limitaciones económicas, surge la necesidad de generar técnicas que permitan obtener de manera indirecta estas variables no medidas.

Por otra parte la formulación de modelos híbridos ha potenciado la disciplina de Control Automático debido a la capacidad para describir en un modelo único el comportamiento de diversos modos de operación del sistema que se desea caracterizar, permitiendo crear soluciones más adecuadas a problemas complejos y abriendo paso a la generación de nuevas herramientas y aplicaciones. La coexistencia de variables continuas y discretas, inherente a la formulación híbrida, hace necesaria una reformulación de las herramientas actuales de control, estimación de estados e identificación de sistemas.

Entre los objetivos del presente trabajo está el desarrollar una metodología que permita obtener variables no medibles de un sistema de Metro, tales como el número de pasajeros que viajan en los trenes y de los que esperan en las estaciones. Los resultados obtenidos muestran que la utilización de un enfoque de estimación de estado en sistemas híbridos permite acceder con la precisión necesaria a las variables de interés, posibilitando la implementación de estrategias de control de flota más avanzadas. Investigaciones futuras involucran una extensión de la metodología para una aplicación a sistemas de buses en corredores o una red completa de transporte.

Palabras Claves: Transporte Público, Estimación de Estados, Modelos Híbridos, Identificación de Sistemas, Filtro de Partículas.

ABSTRACT

After the implementation of Transantiago, the new public transport system in Santiago, there have been various efforts from the academic area to improve the service delivered to users. Within this context, especially since various applications related to the control of bus and subway system require knowledge of system variables that are not always possible to measure, either for lack of sensors or limited economic resources, there is a need for generating techniques to estimate system variables

In the recent time the development of hybrid models have opened a new path in the implementation of the various areas within the automatic control, such as model-based control. The ability to formulate a single model for the behavior of different modes of operation in a single system can generate appropriate solutions to complex problems, making way for new tools. This new range of solutions brings a mathematical challenge due to the coexistence of continuous and discrete variables, generating the need to reformulate the current tools of control, state estimation and system identification.

This paper seeks to create a framework that allows to get state variables from a Metro system that are not possible to measure, such as the number of passengers traveling on trains and the number of passengers waiting at stations. The results show that the use of an approach of state estimation in hybrid systems allows access, with the necessary precision, to the variables of interest, leading to a possible implementation of most advanced fleet management strategies. Further investigation involve the extension to Bus corridor systems or a complete transportation network.

Keywords: Public Transportation, State Estimation, Hybrid Models, System Identification, Particle Filter.

Capítulo 1. INTRODUCCIÓN

1.1. Motivación

La modelación de procesos es un pilar fundamental de la disciplina de Control Automático pues permite realizar análisis de estabilidad, controlabilidad, simulaciones y la aplicación de técnicas de diseño de controladores más avanzados, como aquellas basados en modelo y optimización. Si bien la modelación supone una simplificación de la realidad, su objetivo es lograr la máxima representatividad de la dinámica del proceso, y es por esto que en la última década la formulación de sistemas híbridos ha cobrado fuerza. Los sistemas híbridos poseen estados continuos y discretos, es decir, la dinámica del proceso se debe tanto a la evolución temporal de sus variables de estado como a la ocurrencia de eventos en el sistema o su entorno. El impacto positivo que ha generado esta herramienta de modelación se debe principalmente a su versatilidad para representar procesos que involucren diferentes estados de operación, acción de válvulas on/off, ocurrencia de fallas, etc.

La modelación híbrida requiere de la reformulación de diversas teorías, tales como la estimación de estado y la identificación de sistemas, de manera de incluir los efectos de las variables discretas. También se han realizado esfuerzos para el desarrollo de observadores híbridos, siendo este aún un tema en estudio. La presente tesis se centra en estudiar las metodologías de medición inferencial en sistemas híbridos y aplicarlas a un sistema de transporte público. Estos sistemas son susceptible de representarse mediante un modelo híbrido, dado que su evolución dinámica se gobierna por la ocurrencia de eventos como la llegada de un tren a una estación. Una de las potenciales aplicaciones consiste en obtener directamente todas las variables de estado del sistema necesarias para efectuar un control de flota más eficiente, a menor costo que la medición directa, en los casos en que es técnicamente posible.

1.2. Descripción del Problema

Los sistemas de transporte público se caracterizan por poseer diferentes modos de operación. Por ejemplo, un bus puede encontrarse detenido en un paradero, detenido en un semáforo o desplazándose por su ruta. Este comportamiento diverso puede ser caracterizado, de acuerdo a la teoría de sistemas dinámicos, por modelos dinámicos híbridos, y así fue realizado en Cortés et al. (2010), donde se implementó un control predictivo híbrido que permite optimizar el funcionamiento de un sistema de transporte público basado en buses, en el que se regula el intervalo de tiempo entre buses minimizando los tiempos de espera y viaje de los usuarios. Un trabajo similar se realizó en Delgado, Muñoz, Giesen, y Cipriano (2009), donde se proponen estrategias de control sobre una flota de buses en un corredor unidireccional. Este tipo de soluciones requiere en su implementación el acceso a todas las variables de estado del sistema, que en estos casos incluyen la posición y cantidad de pasajeros que transporta cada bus del sistema y la cantidad de pasajeros que esperan en las paradas, siendo estas últimas en el caso de Cortés et al. (2010) un valor discreto y en el modelo de Delgado et al. (2009) un valor continuo.

Tanto en Cortés et al. (2010) como en Delgado et al. (2009) se realizan suposiciones sobre el acceso al estado del sistema, considerando que el sistema de control cuenta con información completa del estado pasado y presente del sistema. Con esta información se determina la acción de control a efectuar en cada una de las detenciones de los buses en los paraderos, buscando minimizar una función objetivo en un horizonte de predicción de los estados futuros del sistema. El proceso de predicción se basa en utilizar los valores esperados de las variables de estado futuras usando los valores medios de las perturbaciones, que en este caso corresponden al arribo de pasajeros. De esta manera, conociendo el tiempo en que el bus hará su próxima detención se estima la cantidad de pasajeros que lo esperarán en dicha parada. Esta metodología realiza dos supuestos relevantes en relación al problema que resuelve esta investigación: el primero es suponer conocido el estado presente y pasado del sistema y el segundo es omitir el proceso de estimación del modo de operación del sistema.

El problema que resuelve esta tesis es la implementación de un sistema de medición inferencial para un modelo híbrido de un sistema de transporte público de trenes urbanos. Se considera el caso en que se conoce la posición de los trenes y se busca estimar la cantidad de usuarios que viajan en ellos y los que esperan en las estaciones. La diferencia de esta metodología respecto a los anteriores trabajos realizados en relación a estimación de demanda en sistemas de transporte reside en eliminar los supuestos de conocimiento completo del sistema y de unificar los modos de operación del sistema mediante un único modelo. Para ello se utilizará una representación híbrida del sistema, permitiendo describir más de un modo de operación e incluir el uso de variables discretas, como el número de pasajeros. La metodología de estimación se diferencia de aquella usada en la realización de predicciones al incluir no sólo los valores medios de las perturbaciones, sino que la distribución de probabilidad de éstas. Además el proceso de estimación no asume conocido el estado pasado del sistema, realizando un procedimiento de predicción del estado presente en base a mediciones y estimaciones pasadas y posteriormente un proceso de ajuste basado en las mediciones actuales. Esta metodología no es original y es conocida en el área de Control Automático como Estimación de Estado; la contribución innovativa se encuentra en aplicar dicha metodología a un sistema híbrido de transporte público.

Para resolver el problema planteado esta investigación se organiza en tres secciones. La primera está dedicada a la revisión de la literatura existente con el objetivo de estudiar las principales herramientas propuestas y utilizadas. Posteriormente las herramientas seleccionadas se prueban y analizan en una planta hidráulica de estanques y bombas, estrategia comúnmente utilizada en la disciplina de Control Automático para *benchmarking*. Finalmente, y luego que las herramientas han sido analizadas de manera cualitativa y cuantitativa, se propone y evalúa una metodología o esquema de estimación de estados para un sistema de Metro.

1.3. Objetivos

Como primer objetivo se encuentra estudiar y comparar las diversas alternativas existentes en la literatura para observar e identificar un sistema híbrido. El segundo objetivo de esta tesis es desarrollar una metodología de medición inferencial para un sistema de transporte público que permita estimar variables que no es posible medir mediante métodos existentes, tales como la cantidad de pasajeros que viajan en los trenes de una red de Metro.

Sobre el primero de los objetivos cabe mencionar que para subsanar la dificultad matemática que presentan los modelos híbridos se suelen utilizar: a) Filtro de Kalman Extendido junto a una orientación Multi-Modelos; b) Un estimador tradicional más una Máquina de Estados; c) Filtro de Partículas, y d) Estimación de Horizonte Móvil. Esta tesis se centra en los dos últimos métodos, pues la literatura los considera más generales y capaces de proporcionar mejores resultados. Las comparaciones de desempeño de los diferentes métodos de inferencia en sistemas híbridos se han realizado en modelos simples y de baja escala, por lo que siguiendo ese camino se propone el uso de una planta hidráulica tipo *benchmark*.

El segundo objetivo se centra en la descripción de un sistema de transporte público, a través de un modelo híbrido. El resultado final de la tesis, un observador de estado para un sistema de transporte, podría permitir una futura implementación en sistemas de control de flota más avanzados. Cabe mencionar que la validación del observador y su metodología se realiza a través de simulaciones basadas en datos reales del Metro de Santiago.

1.4. Principales Contribuciones

En concordancia con los objetivos planteados las principales contribuciones de la presente investigación son:

- El estudio de las más recientes herramientas de estimación de estado e identificación de parámetros para sistemas híbridos.
- Evaluación conjunta de metodologías de estimación de estado e identificación de parámetros en una misma planta *benchmark*.
- La creación de una metodología de estimación para el número de pasajeros que se encuentran en una red de Metro, ya sea, esperando en las estaciones o viajando en los trenes.

1.5. Organización del Documento

El documento se organiza en capítulos, donde el segundo se centra en presentar el estado del arte de las soluciones a los problemas de estimación de estado e identificación en sistemas híbridos. El capítulo tres presenta un ejemplo de la aplicación de las soluciones descritas a un sistema hidráulico tipo *benchmark*, para el cual se analizan las capacidades y limitaciones de las técnicas descritas. En el capítulo que le sigue se muestra una aplicación de éstas a un sistema de transporte público, siendo éste el mayor aporte de la tesis. Finalmente en el capítulo cinco se presentan comentarios y conclusiones generales del trabajo realizado y posibles aplicaciones e investigaciones futuras.

Capítulo 2. MARCO TEÓRICO

2.1. Sistemas Híbridos

La ventaja de los sistemas híbridos está en su capacidad de modelar eventos y diferentes modos de operación presentes en un mismo proceso. La Figura 2.1 muestra un ejemplo correspondiente a un péndulo, dónde la presencia del obstáculo en el techo genera dos modos de operación. El primero ocurre cuando la cuerda no se encuentra en contacto con el obstáculo y el segundo ocurre cuando la masa entra en la zona punteada causando que la cuerda disminuya su largo al pivotear en el obstáculo. Una descripción híbrida permite describir ambos comportamientos en un mismo sistema, siendo esta la principal contribución de los sistemas híbridos.



FIGURA 2.1. Sistema pendular con obstáculo. La línea punteada marca la frontera que define los dos modos de operación del sistema, donde el obstáculo genera una disminución del largo de la cuerda.

En la literatura existen principalmente 2 formulaciones que permiten esta representación. La más intuitiva de ellas se basa en funciones lineales por partes (PWA, Piecewise Affine Systems) (Sontag, 1981). El comportamiento de la planta se modela mediante un conjunto de funciones lineales que actúan bajo ciertas condiciones del estado o las entradas. Así, la dinámica de la planta se rige por una función que divide el espacio conjunto de estados y entradas en poliedros disjuntos, de manera de representar la acción de eventos o modos de operación. La ecuación 2.1 muestra una descripción lineal a tramos de un sistema, donde χ_i representa uno de los poliedros del espacio posible de entradas y estados, el comportamiento del sistema en el poliedro activo se define mediante las matrices A_i , B_i y C_i y las constantes f_i y g_i :

$$x(t+1) = A_i x(t) + B_i u(t) + f_i \qquad \text{para} \begin{bmatrix} x(t) \\ u(t) \end{bmatrix} \in \chi_i$$

$$y(t) = C_i x(t) + g_i \qquad (2.1)$$

Volviendo al ejemplo de la Figura 2.1, los poliedro χ_i corresponden a las regiones divididas por la linea punteada y las matrices A_i al resultado de linealizar el sistema en cada una de estas regiones.

Por otro lado, la formulación MLD (Mixed Logical Dynamical) (Bemporad y Morari, 1999) permite transcribir el comportamiento del sistema usando variables discretas e inecuaciones. Su estructura principal involucra variables continuas, variables enteras o binarias que se entrelazan mediante ecuaciones e inecuaciones lineales, creando un modelo que se expresa en forma de un conjunto de restricciones de igualdad y desigualdad en variables enteras y continuas. Dicha estructura permite el uso de este tipo de modelos en problemas que involucren optimización, tales como el control predictivo o un observador de estados. Las ecuaciones 2.2 muestran una descripción híbrida de un sistema, donde las variables de estado son $x \in \mathbb{R}^n$ y $\delta \in 0, 1$, la entrada es $u \in \mathbb{R}^m$. $z \in \mathbb{R}^l$ son variables auxiliares continuas a tramos que permiten representar las acción de las variables δ sobre x, donde dicha relación es descrita a través de la inecuación con las matrices E.

$$x(t+1) = Ax(t) + B_1u(t) + B_2\delta(t) + B_3z(t) + B_5$$

$$y(t) = Cx + D_1u(t) + D_2\delta(t) + D_3z(t) + D_5$$

$$E_2\delta(t) + E_2z(t) \le E_1u(t) + E_4x(t) + E_5$$
(2.2)

La dinámica del sistema se modela mediante las matrices A, B, C y D, mientras que las matrices E permiten determinar el valor de $\delta(t)$ en función de x(t) y z(t). Esto puede entenderse como el proceso equivalente de evaluar en qué región χ_i se encuentra el sistema que se realiza en la representación PWA. En efecto la equivalencia entre estas formulaciones se encuentra demostrada en Bemporad, Ferrari-Trecate, y Morari (2000) y permite compartir los análisis de estabilidad y controlabilidad de los sistemas.

2.2. Estimación de Estado en Sistemas Híbridos

Algunas estrategias de control, como MPC (Model Predictive Control) o LQR (Linear Quadratic Regulator), requieren del conocimiento del estado completo del sistema a controlar. Dado que no siempre es posible medir todas las variables de estado, se hace necesaria la utilización de observadores. La idea principal radica en utilizar el modelo, las entradas y las salidas medidas para realizar predicciones del estado y luego ajustar dichas estimaciones a partir de las mediciones. Herramientas matemáticas que resuelven este problema se encuentran presentes en la literatura desde 1960 a través del Filtro de Kalman, utilizado ampliamente en sistemas lineales dado que demuestra ser la estimación óptima bajo ruido blanco gaussiano. Para sistemas no lineales se encuentra disponible el Filtro de Partículas (Gordon, Salmond, y Smith, 1993) y el Filtro de Kalman Extendido (Extended Kalman Filter). Estos métodos no permiten tratar directamente con sistemas con variables discretas, por lo que en la literatura se han propuesto diversas alternativas las que serán revisadas más adelante.

Otra arista de interés son las condiciones de observabilidad y controlabilidad, es decir, que a través de las mediciones es posible conocer el sistema de estados por completo mediante inferencia y que mediante las variables manipuladas es posible llevar al sistema a cualquier estado. Para el caso de sistemas continuos lineales este problema se encuentra solucionado, existiendo expresiones cerradas que evalúan el sistema en términos de la capacidad de ser observado y/o controlado. Sin embargo, el estudio de estas propiedades es un desafío matemático en sistemas híbridos dado que el sistema opera en diferentes modos. Por ello, el problema de la observabilidad se ha analizado en diversos artículos a través del estudio de clases particulares de sistemas híbridos; por ejemplo, en Barbot, Djemai, y Manamanni (2007) se muestra el diseño de un observador de estado en sistemas donde los estados continuos evolucionan en función de los discretos, pero éstos evolucionan de manera independiente de los continuos. Utilizando un modelo determinístico no lineal es posible determinar condiciones de observabilidad y la implementación de un observador que permite estimar estados continuos y discretos a la vez. Por su parte en Bemporad et al. (2000) se presentan contraejemplos sobre sistemas en forma PWA o MLD que demuestran que en diversos casos no es posible asegurar observabilidad y controlabilidad.

El desafío que se presenta en la estimación de estados en sistemas híbridos es un tema reciente en la literatura, presentándose diversas técnicas desde el año 2000 en adelante. Los métodos de estimación descritos a continuación pueden clasificarse en tres grandes grupos: a) los que utilizan Estimación con Multi-Modelos (MME); b) aquéllos basados en Estimación de Horizonte Móvil (MHE) y c) los que se basan en el Filtro de Partículas (PF).

2.2.1. Métodos basados en Multimodelos

La estrategia tras de estos métodos consiste en la utilización de una representación PWA de la planta. La estimación puede realizarse de dos maneras diferentes. Una de ellas es la que se describe en Balluchi, Benvenuti, Benedetto, y Sangiovanni-Vincentelli (2002) y Cinquemani, Micheli, y Picci (2004), en los que el observador se compone de dos estructuras que permiten estimar el estado o modo de operación y luego de ello utilizar un observador de sistemas continuos. Esta estructura se aprecia en la Figura 2.2.



FIGURA 2.2. Observador basado en Multi-Modelos. El observador consiste en 2 estimadores, uno discreto y uno continuo, separando así el problema de estimación.

Una segunda metodología basada en Multi-Modelos consiste en estimar el estado en base al conocimiento de las transiciones entre estados (variables discretas) y el uso de un estimador lineal para cada uno de estos estados. Este tipo de estimadores se presenta en Alessandri y Coletta (2001, 2003), donde el sistema se describe como una secuencia temporal de diferentes modelos lineales. Cabe mencionar que la transición temporal entre modelos se asume arbitraria y conocida, siendo el problema planteado en el artículo el análisis de estabilidad y la optimalidad del observador. Siguiendo la misma estructura de la Figura 2.2, Hamdi, Manamanni, Messai, y Benmahammed (2008) presenta un modelo basado en Redes Diferenciales de Petri que permiten estimar el modo de operación del sistema levantando la restricción impuesta en el estudio de Alessandri y Coletta. Los resultados obtenidos por Hamdi et al. muestran una capacidad de estimación de gran precisión y rapidez de convergencia. Para la aplicación de detección de fallas, se presenta en Cinquemani y Micheli (2006) una metodología para realizar estimación de estados en sistemas híbridos donde la transición entre modos de operación sigue el comportamiento de una cadena de Markov.

Por último cabe mencionar que sobre la base de modelos PWA o de Linear Switching Systems (LSS), se han realizado estudios que buscan determinar condiciones de observabilidad del sistema y estabilidad del estimador. Así Di Benedetto, Di Gennaro, y D'Innocenzo (2009) formula una definición de observabilidad de sistemas híbridos, planteando para sistemas de eventos discretos un algoritmo de verificación de observabilidad y para sistemas LSS plantea el mínimo conjunto de variables continuas que deben ser medidas para estimar el estado del sistema completo. Por otro lado Juloski, Heemels, y Weiland (2007) presenta diversos diseños de estimadores para modelos PWA derivando condiciones suficientes para asegurar que la dinámica del error se encuentra acotada asintóticamente.

2.2.2. Métodos basados en MHE

MHE (Moving Horizon Estimation) plantea la estimación de estados de un sistema como un problema de optimización que debe resolverse en cada instante de tiempo y que busca minimizar los errores entre la predicción y la medición. La Estimación de Horizonte

Móvil para sistemas híbridos fue formulada inicialmente por Bemporad, Mignone, y Morari (1999) y se basa en la descripción MLD del sistema. Esto se debe a que la estructura de ecuaciones e inecuaciones de un modelo MLD se incluye en el proceso de estimación como restricción al problema de optimización. La ecuación 2.3 muestra la estructura de un problema de MHE aplicado a sistemas híbridos. $J(x_t)$ es una función cuadrática que penaliza los errores de predicción y las variaciones en las variables estimadas, asegurando así la estimación del sistema y la estabilidad del problema de optimización. Es posible notar que con respecto a la ecuación 2.2 se han agregado las variables ξ , ς y ϕ , que modelan el ruido de proceso, ruido de medición y perturbaciones discretas respectivamente:

$$\begin{split} \min_{x_t} & J(x_t) \\ \text{s.a} \\ \hat{x}(t+k+1|t) &= A\hat{x}(t+k|t) + B_1 u(t+k) + B_2 \hat{\delta}(t+k|t) + B_3 \hat{z}(t+k|t) \\ &+ B_6 \hat{\phi}(t+k|t) + B_5 + \hat{\xi}(t+k) \\ \hat{y}(t+k|t) &= C\hat{x}(t+k|t) + D_1 u(t+k) + D_2 \hat{\delta}(t+k|t) + D_3 \hat{z}(t+k|t) \\ &+ D_6 \hat{\phi}(t+k|t) + D_5 + \hat{\varsigma}(t+k) \\ &E_2 \hat{\delta}(t+k|t) + E_2 \hat{z}(t+k|t) \leq E_1 u(t+k) + E_4 \hat{x}(t+k|t) + E_6 \hat{\phi}(t+k|t) + E_5 \\ \end{split}$$
(2.3)

Esta última ecuación permite también entender la relación entre un problema de MHE y uno de MPC, ya que al igual que en control predictivo existe un horizonte móvil sobre el cual opera la optimización. Sin embargo, en MHE el horizonte opera en el sentido inverso y se aplica hacia los datos pasados y no futuros como en MPC.

De esta manera se estima el estado presente en términos de las entradas y mediciones pasadas y el modelo del sistema, que opera como restricción. Al igual que MPC, si MHE opera con horizonte infinito, es decir, para cada instante se utilizan todos los datos existentes, la estimación es óptima. El problema de esto es que el tiempo de cómputo es cada vez mayor e impide su operación en tiempo real. Una de las ventajas que posee MHE sobre otras metodologías es la capacidad de incluir restricciones sobre las variables, como por ejemplo, de no negatividad o de cotas. Estas ventajas son analizadas con rigurosidad en Rawlings y Bakshi (2006), donde se comparan MHE y EKF en términos de su predicción, tiempo de cómputo y convergencia.

Las aplicaciones de MHE a sistemas híbridos son variadas, pero poseen un factor común: el tamaño de los sistemas sobre los cuales se aplica es reducido. Esto se debe a la dificultad numérica de resolver en cada instante de tiempo un problema de optimización mixta cuadrática. Similar a las técnicas que permiten al control MPC Distribuido abarcar sistemas más complejos, Farina, Ferrari-Trecate, y Scattolini (2010a, 2010b) presenta una estructura de observador basado en MHE para sistemas lineales distribuidos, sin considerar el caso de la estimación de estados para sistemas híbridos.

La dificultad planteada anteriormente no significa que su utilización sea inviable en procesos industriales; de hecho Valdes-Gonzalez y Flaus (2001) estima el estado en un proceso de fermentación genérico que posee 3 modos de operación, aplicando MHE con una variante que permite optimización global por intervalos (Interval Moving Horizon State Estimation). En la misma línea industrial, Rawlings realiza estudios de aplicación de MHE para resolución en tiempo real. En Rao, Rawlings, y Lee (2001); Rao, Rawlings, y Mayne (2003) y Haseltine y Rawlings (2005) se presentan condiciones de estabilidad y las ventajas comparativas de MHE sobre EKF, dada la posibilidad de manejar restricciones y la dinámica de las perturbaciones. Pina y Botto (2006), por otro lado, aplica MHE a un modelo híbrido de un generador a vapor, donde incluso son desconocidas algunas entradas al sistema Thomas, Dumur, Buisson, Falinower, y Bendotti (2003) aplica MHE a detección de fallas de sensores en el control de un generador de similares características. Finalmente cabe mencionar un artículo de Ferrari-Trecate, Mignone, y Morari (2002) que presenta una aplicación de MHE a un sistema híbrido descrito en su forma PWA. El trabajo muestra cómo es posible sacar provecho de la equivalencia entre la descripción MLD y PWA para formular condiciones de observabilidad sobre el estimador, como por ejemplo un tamaño mínimo para el horizonte de la estimación bajo condiciones de suavidad del mapa PWA y ausencia de ruido en las mediciones.

2.2.3. Métodos basados en Filtro de Partículas

Como se mencionó al comienzo de esta sección, el Filtro de Partículas (PF, Particle Filter) ha sido ampliamente utilizado para estimar estados en sistemas no lineales o donde las perturbaciones son no gaussianas, sin requerir de modificaciones en su estructura. El Filtro de Partículas se basa en realizar una serie de simulaciones (partículas) del sistema a partir de la entrada y el estado anterior del sistema. Estas simulaciones son recogidas y ponderadas según su cercanía con las mediciones, formando así una distribución de probabilidad del estado actual, sobre la cual se toma el valor medio o el valor de mayor probabilidad (moda). La Figura 2.3 muestra un esquema del algoritmo descrito, por lo general las variantes de PF se destacan por diferir en la técnica utilizada para remuestrear la función de distribución de probabilidad a priori para generar la distribución a posteriori.



FIGURA 2.3. Algoritmo Filtro de Partículas. La estimación pasada genera a través de la predicción una serie de partículas que son remuestreadas según su semejanza con la medición para así obtener una función de distribución de probabilidad del estado actual.

La flexibilidad de PF radica en que el sistema y las perturbaciones no gaussianas son utilizados directamente para generar las partículas, evitando así tener que realizar simplificaciones al modelo o suposiciones sobre el ruido o las perturbaciones. Esta flexibilidad se revisa en Arulampalam, Maskell, y Gordon (2002), donde se analizan variantes del filtro de partículas para la estimación óptima y subóptima de sistemas no lineales y con perturbaciones no gaussianas, estimaciones que son comparadas con el Filtro de Kalman Extendido. El estudio demuestra las potencialidades de PF, pero omite toda aplicación a sistemas híbridos.

Generalmente este problema se abarca utilizando partículas discretas y continuas, lo que genera distribuciones Multi-Modelos en la estimación, como muestra el Algoritmo 1, donde \hat{z}_t^i es la partícula i-esima de los estados discretos en el tiempo t, \hat{x}_t^i es la partícula i-esima de los estados continuos en el tiempo t y w_t^i es el peso que pondera la cercanía de la partícula i-esima a la medición en el tiempo t. En Doucet, Gordon, y Krishnamurthy (2001) se trata el problema en sistemas PWA donde las transiciones ocurren siguiendo un proceso Markoviano, y presentándose un algoritmo que evita utilizar las técnicas existentes a través de combinar un remuestreo secuencial, un esquema de selección y métodos de Monte Carlo en cadenas de Markov.

Algoritmo 1: Filtro de Partículas

Etapa de muestreo secuencial;

```
for i \leftarrow 1 to N do
Crear partículas según el modelo;
```

```
\hat{z_t}^{i} \sim Pr(z_t | z_{t-1}^{i});\hat{x_t}^{i} \sim P(dx_t | x_{t-1}^{i}, z_{t-1}^{i});
```

end

for $i \leftarrow 1$ to N do

Evaluar importancia según las mediciones y_t ;

 $w_t^{\ i} \propto p(y_t | x_{t-1}^i, z_{t-1}^i);$

end

Etapa de Selección;

Replicar/Descartar partículas del set $[x_{t-1}^i, z_{t-1}^i]_{i=1}^N$;

según su alto/bajo peso w_t^i , de modo de obtener nuevas N partículas;

En Blom y Bloem (2004) se plantea una método de Filtro de Partículas basado en Multi Modelos que utiliza las ecuaciones bayesianas exactas dada las mediciones, permitiendo así que las partículas generadas sean utilizadas únicamente para estimar los estados continuos. La utilización de las ecuaciones condicionales exactas permite además mejorar la estimación de los estados en condiciones donde las probabilidades condicionales de transición son bajas, problema que tradicionalmente se soluciona aumentando el número de partículas y por lo tanto la carga computacional del método.

En Funiak y Williams (2003) se propone una estrategia que reúne la capacidad escalable del Filtrado Multimodal Gaussiano y las potencialidades de muestreo del Filtro de Partículas, logrando así unificar los observadores de cadenas de Markov y el proceso de re-muestreo del Filtro de Partículas, generando una estimación que muestrea los modos de operación y a partir de ello aplica un Filtro de Kalman para el modo detectado. Los resultados muestran que la metodología utilizada supera al tradicional Filtro de Partículas.

En cuanto a aplicaciones, PF resulta adecuado para una vasta gama de problemas, pues dada su flexibilidad y tratamiento de modelos híbridos es posible aplicarlo a procesos industriales, diagnóstico de fallas y sistemas donde los sensores entregan señales discretas, como alarmas u ocurrencia de eventos (Koutsoukos, 2003). Morales-Menéndez, Freitas, y Poole (2003) presentan una aplicación de PF al control de una planta de intercambio de calor en un proceso industrial con mediciones reales, sobre las cuales se identifican los parámetro de un modelo JMLS (Jump Markovian Linear System) y con el que finalmente se aplica la estimación de tanto estados continuos como discretos utilizando una variante de Filtro de Partículas que incorpora una etapa de re-muestreo basado en el teorema de Rao-Blackwell. El estudio muestra resultados similares a los obtenidos por Freitas (2002), al utilizar esta misma metodología con el fin de diagnosticar fallas. Gracias a la aplicación del teorema de Rao-Blackwell (RBPF) logra resultados con mejor precisión debido a que disminuye la varianza de las distribuciones a posteriori del modo de operación. Mayor información sobre diagnóstico de fallas utilizando Filtro de Partículas se encuentra en Arogeti, Danwei, y Chang (2010), Daigle, Koutsoukos, y Biswas (2010), Wang, Zhou, y Li (2006) y Zhao, Koutsoukos, Haussecker, Reich, y Cheung (2005).

Finalmente y como se mencionó en la sección 2.2.2, existen en la literatura aplicaciones de Filtro de Partículas a sistemas híbridos distribuidos. Específicamente nos referimos al trabajo realizado por Samadi, Salahshoor, y Safari (2009), donde se aprovecha la estructura de red que se generar a partir de los sensores y se realizan estimaciones locales en cada nodo sin la necesidad de tener que romper el modelo en submodelos. Se presenta además un modelo de aprendizaje que actualiza los parámetros de todos los nodos en función del desempeño global del sistema. Por otro lado Koutsoukos, Kurien, y Zhao (2003) desarrolla un esquema de estimación en un reactor criogénico como parte de un sistema distribuido de detección de fallas.

2.2.4. Análisis Comparativo

Presentados los métodos, surge la necesidad de realizar una comparación de modo de mostrar las diferencias, fortalezas y debilidades de cada uno de estos. En esta sección se realiza una comparación cualitativa entre Estimación de Horizonte Móvil y Filtro de partículas, los dos métodos con mayores potencialidades para la familia de aplicaciones de interés de esta tesis. En la sección 3.2.3 se realiza un análisis cuantitativo a través de un ejemplo tipo *benchmark*.

MHE se presenta como un método para estimar sistemas lineales con diferentes modos de operar y donde la descripción del modelo debe realizarse siguiendo una descripción MLD. Es posible además incluir restricciones al modelo utilizando su estructura basada en inecuaciones. Las perturbaciones del sistema pueden poseer un comportamiento que no es necesariamente gaussiano y presentarse de manera continua o discreta (fallas o acciones no medidas). La gran desventaja de esta herramienta radica en que la representación MLD del sistema agrega un gran número de variables auxiliares al problema de optimización. Además al ser un problema mixto de norma cuadrática el tiempo de cómputo crece de manera polinomial con el número de variables, restringiendo por lo tanto el tamaño del horizonte de estimación.

El Filtro de Partículas, por otro lado, posee una flexibilidad que le permite ser utilizado tanto en sistemas lineales como no lineales y con perturbaciones estocásticas de diversas características. De hecho es posible observar sistemas cuyas transiciones entre modos de operación son estocásticas. El manejo de restricciones tampoco es problema para esta herramienta, pues éstas pueden incluirse en la etapa de generación de partículas. El problema que presenta PF es lograr obtener mediante las partículas una representación fiel de la distribución de probabilidad del estado, problema que se suele resolver aumentando el número de partículas lo que genera un tiempo de cómputo mayor. La Tabla 2.1 resume las características analizadas en los párrafos anteriores.

Al referirnos al problema que esta tesis pretende resolver, estimar la cantidad de usuarios viajando y esperando en estaciones, por su flexibilidad y capacidad de incorporar perturbaciones no gaussianas el Filtro de Partículas y la Estimación de Horizonte Móvil parecen ser los adecuados. Sin embargo, dada la estructura y tamaño del sistema de transporte público la carga computacional sugiere que el Filtro de Partículas es la elección apropiada.

TABLA 2.1. Comparación de Métodos de Estimación

Método	Sistema	Carga Computacional	Perturbaciones	Tolerancia al Ruido
MME	Lineal	+ + +	Gaussianas	+
MHE	Lineal	++	No Gaussianas	++
PF	No Lineal	+	No Gaussianas	+ + +

2.3. Identificación de Sistemas Híbridos

La identificación de sistemas es otra metodología inferencial que a través del ajuste de un modelo paramétrico al comportamiento del sistema o planta permite acceder a variables no medidas. Los métodos tradicionales de identificación serán clasificarse según el modelo que buscan ajustar, se formula éste en el espacio de estados o como modelo entrada-salida. También pueden caracterizarse de acuerdo a si el modelo a ajustar es lineal o no lineal y cuantos parámetros deben ser optimizados. El proceso de identificación de un sistema híbrido se complica por las mismas razones que en estimación de estado: la presencia de variables discretas y la no linealidad de los procesos debido a la existencia de diferentes modos de operación. En la literatura es posible distinguir diversos métodos que buscan resolver este problema, los que comparten una estrategia similar, que consiste en determinar en un comienzo los modos de operación definiendo una partición espacial de los datos (regresores) y luego ajustar submodelos lineales a aquellos modos. Estas estrategias tienen un modelo base de entrada salida, es decir, no se utiliza la identificación en el espacio de estados de sistemas híbridos.

Los métodos existentes se diferencian entre aquellos que plantean el proceso de identificación como un problema de optimización numérica, aquellos que lo resuelven de manera algebraica, aquellos que realizan un planteamiento en términos de probabilidades condicionales y los que utilizan herramientas de clustering (Niessen, Juloski, Ferrari-Trecate, y Heemels, 2004; Juloski, Heemels, et al., 2005). En esta sección se analizará en profundidad esta última alternativa.

2.3.1. Planteamiento del Problema

El problema de identificación que estudiaremos consiste en ajustar un modelo lineal e invariante en el tiempo, en este caso de entrada salida, a una serie de datos. Se asume que los datos obedecen a un sistema descrito por la ecuación 2.4, donde $\epsilon(k)$ es ruido blanco gaussiano de desviación estándar σ y la función f es el mapa del sistema PWA de s modos de operación.

$$f(x(k)) = \begin{cases} \theta_1^T[x(k) \ 1], & \text{si } x(k) \in \chi_1 \\ \vdots & \vdots \\ \theta_s^T[x(k) \ 1], & \text{si } x(k) \in \chi_s \end{cases}$$
(2.4)
$$y(x(k)) = f(x(k)) + \epsilon(k)$$

El vector de regresores se compone de valores de mediciones y entradas pasadas, como muestra la ecuación 2.5:

$$x(k) = [y(k-1) \ y(k-2) \ \dots \ y(k-n_a) u(k-1) \ u(k-2) \ \dots \ u(k-n_b)]$$
(2.5)

El problema consiste en lograr, dado los datos y los parámetros s, n_a y n_b , reconstruir el mapa f de la ecuación 2.4 obteniendo el mínimo error posible. Obtener el mapa f implica estimar los parámetros $\{\theta\}_{i=1}^{i=s}$ y las particiones del espacio $\{\chi\}_{i=1}^{i=s}$ representado por un conjunto de inecuaciones de la forma $Mx \le m$.

2.3.2. Algoritmo de Identificación

Ferrari-Trecate, Muselli, Liberati, y Morari (2001) plantea un algoritmo de resolución para el problema presentado en 2.3.1 que se divide en 3 etapas, las cuales se describen en el Algoritmo 2.

Algoritmo 2: Identificación de Sistemas PWA

Construir los Clusters C_i

for $i \leftarrow 1$ to N do

$$\forall (\tilde{x}, \tilde{y}) \in \mathcal{C}_i, \parallel x(j) - \tilde{x} \parallel \leq \parallel x(j) - \hat{x} \parallel, \forall (\hat{x}, \hat{y}) \in \mathcal{S} \setminus \mathcal{C}_i$$

end

Encontrar $\hat{\theta}_i$ para cada \mathcal{C}_i

for $i \leftarrow 1$ to N do

$$\theta_i = (\Phi_i^T \Phi_i)^{-1} \Phi_i^T y_{\mathcal{C}_i} \Phi_i^T = \begin{bmatrix} x_1^T & x_2^T & \dots & x_c^T \\ 1 & 1 & \dots & 1 \end{bmatrix}$$

end

Encontrar la división poliedral de f

for $i \leftarrow 1$ to s do $[M_{i1} \dots M_{is}] x \leq [m_{i1} \dots m_{is}]$

end

El algoritmo se inicia con la formación, en el espacio $y(k) \times x(k)$, grupos C_i con c elementos a partir de datos adyacentes. Aplicándose regresión lineal se obtienen los parámetros del modo que está localmente activo en ese cluster (θ_i).

En la siguiente etapa del algoritmo los parámetros de regresión se agrupan en s clusters mediante *k-means*; en esta etapa se utiliza el conocimiento del número de modos de operación de la planta a identificar. Una vez encontrados los grupos de datos éstos se etiquetan. Hasta este punto se conoce a qué modo de operación pertenece cada par ordenado $y(k) \times x(k)$ y los parámetros del sistema afín que los reúne.

Utilizando la información de los grupos C_i , que relaciona los datos y los parámetros de regresión, se procede a la tercera etapa del algoritmo. Para cada grupo de datos en el espacio $y(k) \times x(k)$ se realiza una regresión lineal con el fin de determinar los parámetros $\theta_{i=1}^{s}$. Los autores del algoritmo justifican esta nueva regresión pues las medias obtenidas por k-means no utilizan todos los datos, pues sólo se basan en los grupos de tamaño c y por ende son poco representativos.

Realizado el proceso de regresión falta estimar la partición del espacio de f. Para esto los autores realizan un proceso de clasificación lineal, es decir, se buscan los hiperplanos que logran dividir el conjunto de datos con el mínimo error posible. La división creada por estos hiperplanos corresponde a la partición de χ . Esta división se realiza utilizando técnicas de clasificación como Suported Vector Machine (SVM) o Linear Discriminant Analysis (LDA) (Vapnik, 1998).

2.3.3. Otras Metodologías

Paoletti, Juloski, Ferrari-Trecate, y Vidal (2007) presenta en un artículo tutorial los principales algoritmos de identificación, distinguiendo principalmente tres métodos adicionales de identificación de sistemas híbridos, uno basado en optimización, otro en filtro de partículas y otro de resolución explícita.

Así como el problema de estimación de estado se resuelve utilizando una estrategia basada en optimización, como es MHE, Bemporad, Roll, y Ljung (2001) plantea el proceso de identificación como un problema de optimización mixta usando la estructuración MLD. La principal desventaja que presenta este acercamiento es que la optimización considera una cantidad de variables significativa para problemas de pocas dimensiones, propiedad intrínseca de la representación MLD, y que el tamaño del problema de optimización crece de manera exponencial con los datos. Lauer, Bloch, y Vidal (2010, 2011) convierte el problema mixto en un problema continuo a través de un replanteamiento de la función objetivo, permitiendo disminuir la carga computacional drásticamente. Sin embargo, esta técnica no es capaz de inferir el número de modos de operación, el que debe ser definido por el usuario. En Feng, Lagoa, Ozay, y Sznaier (2010) se analiza el efecto del ruido en el proceso de identificación y las conclusiones alcanzadas muestran que la co-existencia de modos, debido al ruido en los datos, dificultan fuertemente el proceso de optimización.

Juloski, Weiland, y Heemels (2005) presentan una estrategia similar al Filtro de partículas analizado en la sección 2.2.3. Este planteamiento Bayesiano considera que los parámetros a identificar poseen una distribución de probabilidad desconocida, utilizando el concepto detrás del filtro de partículas se estima el valor de los parámetros a través de una probabilidad condicional a priori y posteriori. El problema mayor presente en este método es que no es posible estimar el número de modos de operación y el orden de los sub-modelos. Dado que se basa en Filtro de Partículas los parámetros de ajuste son las distribuciones de probabilidad de los parámetros y del ruido de proceso.

El desconocimiento de la cantidad de modos de operación se resuelve en Vidal, Soatto, Ma, y Sastry (2003), donde se presenta una estrategia de solución explícita del problema de identificación en ausencia de ruido. Este método permite identificar modelos en que las transiciones son aleatorias usando modelos SARXS (Switched Auto-Regressive eXogenous System). Vidal (2004) estudia la identificación de un modelo en que las transiciones son dependientes de los datos de entrada usando modelos PWARX (Piecewise Auto-Regressive eXogenous System), donde una vez más el ruido no está presente en los datos. Ma y Vidal (2005) reúnen ambos resultados y estudian el efecto del ruido en los datos, concluyendo que éste genera un impacto muy negativo en el desempeño del algoritmo. Finalmente Bemporad, Garulli, Paoletti, y Vicino (2003, 2004) proponen un algoritmo que se basa en encontrar los parámetros que permiten que el error de identificación no supere un valor predefinido δ , resolviendo el sistema de inecuaciones 2.6. Este algoritmo requiere del número de subsistemas y los órdenes de los submodelos, siendo su parámetro de ajuste el valor de δ

$$|y(k) - [x(k) \ 1]\theta(k)| \le \delta \qquad k = 1, \dots, N$$
 (2.6)

2.3.4. Análisis Comparativo

La Tabla 2.2 presenta un resumen de los métodos revisados, considerándose como índices de comparación la diferencia con respecto a los valores reales de los parámetros $(\Delta \theta)$, el efecto negativo que produce la sobreestimación del número de parámetros (Δn_{ab}) , la tolerancia al ruido (NT) y si el método es capaz de inferir los modos de operación del sistema (MI).

TABLA 2.2. Comparación de Métodos de Identificación

Método	$\Delta \theta$	Δn_{ab}	NT	MI
Clustering	+	++	+ + +	No
Error Acotado	+	+	+ + +	No
Bayesiano	++	++	+	No
Algebraico	++	+	+	Si

El análisis realizado permite afirmar que el ruido en los datos de identificación afecta negativamente a cada uno de los métodos, pero con mayor fuerza al propuesto por Vidal et al. (2003). Niessen et al. (2004) propone utilizar dicho algoritmo como una herramienta para alimentar las condiciones iniciales de los otros métodos, de modo de disponer de una estimación de las características del sistema a identificar, cuando se desconoce la estructura y dimensión del sistema. Los métodos basados en clustering presentan mayor tolerancia al ruido y se recomienda su uso en casos en que se desconoce la física del sistema y se desea

ajustar un modelo predeterminado. Sin embargo, se debe considerar que se presentan comportamientos erráticos frente a una elección errada de los órdenes de los modelos (Juloski, Heemels, et al., 2005). Similarmente el algoritmo de Error Acotado es una herramienta útil ante un sistema donde el modelo dinámico se desconoce, pues es posible calibrar el *tradeoff* entre la complejidad y la precisión sin la presuposición de un modelo, pero la elección de estos parámetros puede ser muy compleja (Niessen et al., 2004). Finalmente el método de identificación basado en Filtro de Partículas es recomendado en casos en que se conozca la dinámica del sistema y las características estadísticas del ruido, pues una elección equivocada de éstos modelos genera resultados incorrectos (Paoletti et al., 2007).

2.4. Herramientas Computacionales

Las herramientas computacionales disponibles serán descritas según el área en la que aportan, distinguiéndose la descripción, simulación, identificación y estimación de sistemas híbridos.

La modelación o descripción de sistemas híbridos puede ser un proceso complicado e intrincado debido a la conversión de proposiciones lógicas en inecuaciones, especialmente en el formato MLD. Existe una herramienta especializada (Torrisi y Bemporad, 2004) que permite describir estos sistemas en un lenguaje de alto nivel conocido como HYSDEL (Hybrid System Description Language). El código es luego compilado y transcrito a formato MLD y PWA basado en la conexión de diversos modelos lineales interconectados por reglas if - else, proposiciones lógicas y máquinas de estados que se describen en el código escrito por el usuario. Su disponibilidad es gratuita en la web bajo Licencia Pública General (GPL).

La simulación de sistemas híbridos, incluyendo el de un controlador híbrido, puede ser realizada en diferentes plataformas, tales como Matlab y Simulink, softwares que entregan un ambiente de desarrollo que permite la programación de funciones, sistemas e interfaces de usuario mediante paquetes de herramientas (Toolboxes). Entre ellos se encuentran los desarrollados por en la universidad ETH de Zurich, el Toolbox MPT y HYSDEL; el primero

incluye dentro de sus funcionalidades la programación de controladores híbridos lineales y cuadráticos que pueden ser incluidos en Simulink. Esta plataforma de simulación posee una herrameinta, SimEvents, que facilita la simulación de sistemas basados en eventos y que ha demostrado ser útil en la descripción de sistemas tipo *queue-server* (Clune, Mosterman, y Cassandras, 2006; Cassandras, Clune, y Mosterman, 2006; Gray, 2007). Una plataforma alternativa a Matlab y Simulink es la utilización de la librería de descripción de modelos Modelica que junto con la librería de simulación desarrollada por Sanz, Urquia, Cellier, y Dormido (2012) permite la modelación y simulación de sistemas híbridos siguiendo una descripción orientada a objetos.

Para realizar un proceso de identificación es posible emplear el Toolbox MPT que incluye la posibilidad de identificar sistemas PWA utilizando el algoritmo propuesto por Ferrari-Trecate et al. (2001) y descrito en la sección 2.3.2. Sin embargo, no existe una herramienta computacional dedicada a la estimación de estado en sistemas híbridos.

En el desarrollo de la presente tesis se hará uso de Matlab como herramienta de desarrollo y simulación para la solución del problema hidráulico tipo *benchmark* del capítulo 3, y también se emplearán los Toolboxes HYSDEL y MPT. Sin embargo, en el desarrollo del problema de transporte se utilizará SimDinMetro, una aplicación desarrollada por Grube, Núñez, y Cipriano (2011) que permite la simulación de la red de Metro de Santiago. Este software posee entre sus características la comunicación con Matlab, herramienta que se utilizará para la programación del observador de estado.

Capítulo 3. MEDICIÓN INFERENCIAL EN SISTEMA HIDRÁULICO

El objetivo de este capítulo es dar una visión cuantitativa de los métodos existentes en la literatura. Para ello se presenta un ejemplo de las técnicas de medición inferencial descritas en el capítulo anterior sobre la simulación de un sistema tipo *benchmark* basado en el utilizado en Alvarado et al. (2011). La planta a analizar se caracteriza por ser no lineal y poseer 8 modos de operación, definidos por sus entradas, perturbaciones y estados. Primeramente se describe el sistema y sus diferentes representaciones (PWA y MLD) para luego presentarse los resultados de identificación y estimación de estado.

3.1. Planta a Analizar

La planta a analizar es un sistema de 4 estanques distribuidos como muestra la Figura 3.1.



FIGURA 3.1. Esquema Planta 4 estanques. Los estanques se alimentan de las bombas a y b y el flujo es direccionado por las válvulas $\gamma_a y \gamma_b$

El sistema se alimenta mediante 2 bombas independientes y las variables medidas son las alturas en los estanques 1 y 2. Las válvulas 'a' y 'b' dividen el flujo en razón γ , con
lo que se alimenta a los tanques superiores; sin embargo esta división de flujo puede ser controlada de manera on/off externamente, siendo ésta una de las características híbridas del sistema. La posición en la válvula 'a' se considera una variable manipulada y la posición en la válvula 'b' corresponde a una variable externa.

Las ecuaciones de estado de esta planta se presentan a continuación, donde las variables q_i y h_i son continuas y las variables δ_i son binarias:

$$A\frac{dh_{1}}{dt} = -a_{1}\sqrt{2gh_{1}} + a_{3}\sqrt{2gh_{3}} + \delta_{1}\gamma_{a}q_{a}$$

$$+ a_{12}sign(h_{2} - h_{1})\sqrt{2g|\delta_{4}(h_{2} - h_{v}) - \delta_{3}(h_{1} - h_{v})|}$$

$$A\frac{dh_{2}}{dt} = -a_{2}\sqrt{2gh_{2}} + a_{4}\sqrt{2gh_{4}} + \delta_{2}\gamma_{b}q_{b}$$

$$- a_{12}sign(h_{2} - h_{1})\sqrt{2g|\delta_{4}(h_{2} - h_{v}) - \delta_{3}(h_{1} - h_{v})|}$$

$$A\frac{dh_{3}}{dt} = -a_{3}\sqrt{2gh_{3}} + (1 - \delta_{2}\gamma_{b})q_{b}$$

$$A\frac{dh_{4}}{dt} = -a_{4}\sqrt{2gh_{4}} + (1 - \delta_{1}\gamma_{a})q_{a}$$
(3.1)

Los parámetros A y a_i son el área transversal de los estanques y de las salidas de agua, respectivamente. Las variables binarias δ_1 y δ_2 son entradas del sistema y regulan el comportamiento de las válvulas, mientras que las variables binarias δ_3 y δ_4 siguen la siguiente lógica.

$$\delta_3 = 1 \iff h_1 \ge h_v$$

$$\delta_4 = 1 \iff h_2 \ge h_v$$
(3.2)

La Tabla 1 muestra los valores de los parámetros utilizados en el modelo

Además de lo mencionado anteriormente, por razones de diseño los estanques poseen una altura de 1.2 metros, por lo que las variables de estado se encuentran acotadas entre 0 y 1.2 [m]. Adicionalmente las bombas no pueden extraer agua de pozo a una razón mayor a 3.26 y 4 $[m^3/h]$ para q_a y q_b respectivamente.

TABLA 3.1. 1	Parámetros	del	sistema
--------------	------------	-----	---------

Parámetro	Valor	Unidades
А	0.06	$[m^2]$
a_1	1.310×10^{-4}	$[m^2]$
a_2	1.507×10^{-4}	$[m^2]$
a_3	9.267×10^{-5}	$[m^2]$
a_4	8.816×10^{-5}	$[m^2]$
a_{12}	9.042×10^{-5}	$[m^2]$
γ_a	0.3	_
γ_b	0.4	_
h_v	0.7	[m]
g	9.81	$[m^2/s^2]$

3.1.1. Representación MLD

La representación MLD del sistema, ver ecuación 2.3, es no lineal y por lo tanto es necesario aplicar alguna técnica de linealización. Bemporad et al. (1999) propone aproximar la raíz cuadrática mediante una recta que une los extremos, desde el mínimo hasta el máximo de las alturas.

Sin duda es posible utilizar una mejor aproximación, como los mismos autores señalan. Por ello se plantea como estrategia de linealización una aproximación de Taylor de primer orden en torno a un punto de operación. El punto de operación que se utilizará es el mismo planteado en Alvarado et al. (2011) y que se detalla en la Tabla 3.2. Realizado este procedimiento se fijan las cotas para las entradas y los estados, necesarias para elaborar el modelo MLD del sistema. Dichos valores se muestran también en la Tabla 3.2.

Utilizando la herramienta HYSDEL se obtiene a partir de un código de alto nivel (ver Anexo B) las matrices que están involucradas en el sistema hidráulico de 4 estanques ya linealizado; detallarlas va más allá de la intención de esta sección por lo que sólo serán descritas en términos de sus dimensiones. Siguiendo un modelo similar a la ecuación 2.3 se obtiene que el modelo cuenta con 40 restricciones de desigualdad, 4 variables binarias

Variable	Valor Máximo	Valor Mínimo	Punto Operación
h_1	1.2	0	0.6538
h_2	1.2	0	0.6524
h_3	1.2	0	0.6594
h_4	1.2	0	0.6588
Q_a	3.6	0	1.63
Q_b	4	0	2

TABLA 3.2. Cotas y punto de operación del sistema

(δ), 6 variables semicontinuas (z), 3 entradas (flujos y válvula a), 1 perturbación discreta (válvula b) y 4 variables continuas que son justamente las alturas en los estanques.

3.1.2. Representación PWA

Dado que el sistema posee 4 variables binarias, la división del mapa PWA del sistema posee 16 subespacios (2^4). El modelo a utilizar sigue la forma del presentado en 2.1, con 16 conjuntos de matrices *A*, *B*, *f* y *g*:

$$x(t+1) = A^{i}x(t) + B^{i}u(t) + f^{i}$$

$$y(t) = C^{i}x(t) + g^{i}$$

$$para\begin{bmatrix} x(t) \\ u(t) \end{bmatrix} \in \chi^{i}$$
(3.3)

La obtención de estas matrices se realiza a partir del mismo código HYSDEL y por lo tanto sigue el mismo proceso de linealización.

3.2. Estimación de Estado

A continuación se presentan los resultados obtenidos observando el sistema descrito en la sección 3.1. Las variables de estado a estimar son las alturas de los estanque 3 y 4, las variables binarias auxiliares δ_3 y δ_4 y las variable binaria δ_2 que define la perturbación sobre la bomba 'b'. De este modo vemos que el problema busca estimar variables tanto continuas como discretas, ya sea de estado o perturbaciones externas. Las secciones siguientes muestran los resultados obtenidos para los métodos de MHE y PF, para luego presentar un análisis comparativo. La Figura 3.2 muestra la estructura descrita.



FIGURA 3.2. Esquema de variables en planta. La acción de una de las válvulas actúa como perturbación, lo que también es estimado por el observador.

3.2.1. Estimación de Horizonte Móvil

Los resultados obtenidos por medio de esta metodología de estimación debe ser analizados considerando la descripción cuantitativa de la sección 2.2.4, donde se explícita que MHE posee una sensibilidad mayor al ruido del sistema, lo que se traduce en que la estimación de la variable δ_2 posee un comportamiento conocido como *chattering*. Este efecto se observa en la Figura 3.4 entre los segundos 3000 y 4000. La estimación se realiza con un horizonte de N = 3 muestras. Los resultados para las alturas de los estanques 3 y 4 se muestran en la Figura 3.3.

3.2.2. Filtro de Partículas

Se presentan a continuación los resultados obtenidos para la estimación de estado para el caso de un observador basado en Filtro de Partículas, considerando simulaciones que consideran un total de 200 partículas. En la Figura 3.6 es posible apreciar el desempeño del estimador, tanto en las variables continuas h_3 y h_4 como en la discreta δ_2 .



FIGURA 3.3. Estimación de las variables de estado continuas. La estimación de estas variables posee un error bajo, cercano a 1.4×10^{-2} , principalmente debido al uso de un horizonte de estimación que capta la dinámica del sistema.



FIGURA 3.4. La estimación de δ_2 se ve afectada fuertemente por el ruido presente en el sistema presentando un comportamiento de tipo *chattering*

3.2.3. Análisis Comparativo

Es posible apreciar que los resultados son concordantes con el análisis cualitativo de la Tabla 2.1, presentando la estimación MHE una menor tolerancia al ruido del sistema y un error de estimación menor debido al uso de un horizonte de estimación que permite recoger la dinámica de evolución del sistema. La Tabla 3.3 muestra el error de estimación sobre las variables continuas del sistema utilizando como índice de desempeño el error medio cuadrático (RMS). Se observa que si bien los errores son del mismo orden de magnitud para las variables no medidas, h_3 y h_4 , la carga computacional es mayor en el sistema de



FIGURA 3.5. Estimación de las variables de estado continuas. La estructura del sistema perjudica más a la estimación de h_3 dado que δ_2 es desconocido; aún así la estimación obtenida es precisa y valida por lo tanto el uso de Filtro de Partículas.



FIGURA 3.6. Resultado de estimación por Filtro de Partículas. Los resultados muestran un pequeño desfase en la detección del cambio en δ_2 . El buen desempeño afecta de igual manera la estimación de h_3 .

estimación por MHE, tardando 120 segundos contra tan solo 90 segundos del Filtro de Partículas.

El análisis arroja como conclusión que la estimación por MHE es más precisa en la estimación de las variables continuas debido a que cuenta con mayor información de la evolución de la planta; sin embargo posee una tolerancia menor al ruido que el Filtro de partículas debido a que este último utiliza información estadística del ruido y las perturbaciones. Además la estimación por MHE posee mayor carga computacional.

Variable	Error RMS PF	Error RMS MHE
h_1	6.6×10^{-3}	5.9×10^{-3}
h_2	2.1×10^{-2}	6.4×10^{-3}
h_3	5.9×10^{-2}	1.6×10^{-2}
h_4	1.2×10^{-2}	1.3×10^{-2}
Tiempo	90	120
Cómputo	20	120

TABLA 3.3. Comparación resultados de estimación

3.3. Identificación de Sistemas

Para evaluar el desempeño de la identificación de sistemas híbridos se propone como planta a analizar una variación de la presentada en la sección 3.1, que considera sólo los estanques 1 y 2 en el proceso de identificación, siendo los diferentes modos de operar los estados definidos por las variables δ_1 y δ_2 en la ecuación 3.1. Las regiones que definen esta partición del espacio se muestran en la Figura 3.7.



FIGURA 3.7. El espacio de estados continuo se particiona en 4 regiones que definen los modos de operación de la planta. Los modos de operación reflejan el efecto de la conexión entre los tanques.

El proceso de identificación debe evaluarse en al menos 2 ámbitos; el primero tiene relación con la estimación de los parámetros θ y el segundo con la capacidad de detectar la división del espacio de estados. Los resultados que se muestran a continuación corresponden a la identificación de la altura en el estanque 1. El caso del estanque 2 es análogo a este.

La identificación de sistemas PWARX supone una relación lineal entre los regresores y la salida del sistema; por lo tanto para evaluar el desempeño de la identificación en términos de los parámetros θ , se procedió a linealizar la planta en puntos de operación en cada una de las regiones mencionadas anteriormente. De esta manera se crean valores ideales de θ sobre los cuales es posible realizar una comparación. Los valores encontrados luego de este proceso se muestran en la Tabla 3.4.

TABLA 3.4. Parámetros a identificar

Región	$ heta_1$	θ_2	θ_3	$ heta_4$
χ_1	-2.99×10^{-2}	0	1.5	0.65
χ_2	-6.33×10^{-2}	0	1.5	0.95
χ_3	-6.33×10^{-2}	3.34×10^{-2}	1.5	0.95
χ_4	-2.99×10^{-2}	3.34×10^{-2}	1.5	0.65

El método utilizado para identificar la planta es el que se propone en Ferrari-Trecate et al. (2001) y que está disponible como parte del Toolbox de Identificación de Sistemas Híbridos (HIT). Para la simulación se recogieron 60 datos con un ruido de medición de desviación estándar 0.071 [m]. Por otro lado el parámetro de clústering del algoritmo, c, se escogió con un valor de 6. Los resultados se muestran en la Tabla 3.5, donde es posible observar que el error de identificación es bastante elevado para los valores correspondientes al parámetro del flujo proveniente de la bomba a. Cabe mencionar que el algoritmo tarda 10 minutos por cada variable a identificar (h_1 y h_2), alejándose considerablemente de los tiempos de identificación de sistemas no híbridos.

Finalmente la evaluación de las regiones que dividen el espacio de estados se realizará en términos visuales, mostrándose en la Figura 3.8 el resultado de la identificación.

TABLA 3.5. Parámetros identificados

Regiór	θ_1	θ_2	$ heta_3$	$ heta_4$
χ_1	-1.49×10^{-2}	$1.10{\times}10^{-4}$	23.5	0.75
χ_2	-4.83×10^{-2}	4.05×10^{-5}	36.2	0.82
χ_3	-7.18×10^{-2}	2.94×10^{-2}	21.95	0.89
χ_4	-1.21×10^{-2}	3.04×10^{-2}	10.35	0.72

Es posible observar que los 4 subespacios son determinados con una precisión aceptable, teniendo en cuenta la presencia de ruido en las mediciones y la baja cantidad de datos disponibles.



FIGURA 3.8. El proceso de identificación determina aproximadamente modo las regiones del sistema, incluso en presencia de ruido.

3.4. Observaciones Respecto del Problema

Las evidencias obtenidas en la literatura y la aplicación de diferentes metodologías en relación a medición inferencial en sistemas híbridos muestran que para abordar el problema

de estimar el número de pasajeros en trenes y estaciones es necesario tener en consideración algunas características del sistema, el caso del sistema de Metro. En este caso la variables discretas no son tratadas como variables de estado auxiliares que permiten diferenciar modos de operación, sino que son una variable principal que toma diversos valores no solamente binarios. Esta característica implica que, dado el avance actual, no es posible utilizar un enfoque de identificación de sistemas, pues el espacio de estados se encuentra dividido en demasiados subespacios. Además la experiencia muestra que este proceso requiere de mediciones con pocos datos, lo cual disminuye la representatividad de cada uno de los subespacios a medida que estos aumentan. De esta manera, el enfoque debe ser mediante el desarrollo de un observador de estado.

El Filtro de Partículas, si bien presentó un desempeño inferior a MHE, muestra ser flexibile ante el modelo, el ruido y las perturbaciones, características que ningún otro método posee. Esta flexibilidad implica una programación más simple y más eficaz y un tiempo de cómputo más adecuado para una posible implementación en tiempo real. Además la representación MLD de sistemas híbridos posee la desventaja de crecer en complejidad significativamente con la cantidad de variables discretas que se posee, desventaja que aplica de manera similar a lo discutido para un enfoque de identificación de sistemas.

Capítulo 4. APLICACIÓN A SISTEMA DE TRANSPORTE

El presente capítulo analiza un caso particular de estimación de estados que tiene relación con sistemas de transporte público. En específico se presenta un modelo de un sistema de Metro, simulado utilizando una herramienta creada en una investigación pasada, y posteriormente se muestra un método de estimación del número de pasajeros en los trenes y estaciones del modelo. Este capítulo se organiza comenzando con la descripción del modelo, luego continúa con una breve referencia a la herramienta de simulación y finalmente se centra en describir la estrategia de estimación y sus resultados.

4.1. Modelo

Siguiendo el esquema planteado en Grube et al. (2011) el sistema se caracteriza con tres entidades principales: pasajeros, trenes y estaciones. Estos elementos se combinan entre sí siguiendo un enfoque de programación orientada a objetos. Así, cada estación pertenece a una línea determinada y se define mediante el par (s, i), donde $s \in \{1, 2, ..., S_i\}$ e $i \in \{1, 2, ..., L\}$, siendo S e I los conjuntos de estaciones y líneas respectivamente. Los trenes de cada línea se numeran usando el índice $m \in \{1, 2, ..., M_i\}$, donde M_i corresponde a la cantidad de trenes que dispone la línea i. El modelo de los pasajeros es macroscópico y por lo tanto no se hace necesario diferenciarlos de manera individual; el sistema los reúne en grupos según su estación de arribo y destino. Si el lector desea revisar los diferentes enfoques de simulación de sistemas de Metro se puede referir a Núñez, Reyes, Grube, y Cipriano (2011), artículo que se encuentra anexado (A.1) a este documento como resultado de la investigación inicial realizada.

Los pasajeros arriban a las estaciones de acuerdo a una distribución de Poisson compuesta con tiempo variable, lo que significa que llegan en grupos de tamaño variable separados por intervalos de tiempo distribuidos exponencialmente. Las llegadas se define por el par (λ_g, λ_l) , donde λ_g es el tamaño promedio del grupo, y λ_l es la frecuencia de llegada de grupos, por lo tanto, la tasa de llegada de pasajeros equivalente es $\lambda = \lambda_g \cdot \lambda_l$. Con el fin de considerar la variabilidad de la demanda de los sistemas de Metro durante todo el día la estructura del modelo permite cambios de tiempo discreto en los valores de las tasas de llegada, esto es, la tasa $\lambda(t)$ es función del tiempo; con fines de simplicidad en la notación el índice de tiempo se omite en adelante.

En esta investigación el modelo se ha simplificado para representar un sistema de Metro de una sola línea, utilizando los parámetros presentados en la tabla 4.1 y las variables de estado definidas en la tabla 4.2. El modelo posee dentro de sus supuestos que el los trenes permanecen en las estaciones sólo debido al intercambio de pasajeros y que su desplazamiento en las vías no se ve interrumpido debido a fallas o desperfectos.

C:	Capacidad del tren m.
λ_s :	Tasa de arribo de pasajeros a
	la estación s.
$\alpha, \beta, \gamma, \delta$:	Coeficientes que determinan
	los tiempos de subida y baja-
	da de pasajeros.
$M_{(s1,s2)}$:	Elementos de la matriz
	Origen-Destino con la pro-
	porción de pasajeros que
	viajan de la estación $s1$ a la
	estación s2.
L_s :	Largo del tramo que une la
	estación s con la estación $s +$
	1.
v:	Velocidad media del tren en
	la línea.

TABLA 4.1. Parámetros del modelo

El modelo se basa en ocurrencia de eventos relacionados con la llegada de los trenes y los pasajeros en las estaciones, similar a lo presentado en Doucet et al. (2001) y descrito en la sección 2.2.3. Cuando un evento se lleva a cabo las variables de estado del sistema se actualizan con el fin de obtener el nuevo estado del sistema. A continuación se detallan las ecuaciones utilizadas en el proceso de actualización.

Dado un conjunto de valores iniciales para las variables de los estados, el proceso comienza con la determinación del evento más próximo, para esto es necesario determinar el mínimo entre el próximo arribo, ya sea de un tren o de pasajeros, a una estación. La

TABLA 4.2. Variables de estado del modelo

$sa_m[k]$:	Indice que determina la esta-		
	ción en la que el tren m se		
	ubica en el tiempo k .		
$T_m[k]$:	Tiempo de detención en la		
	estación en la que el tren m		
	se encuentra en el tiempo k .		
$ps_{(s,s1)}[k]$:	Número de pasajeros espe-		
())	rando en la estación s cuyo		
	destino es la estación $s1$ en		
	el tiempo k.		
$pm_{(m,s)}[k]$:	Pasajeros que viajan en el		
- ()) 2 2	tren m a la estación s en el		
	tiempo k.		
$ta_m[k]$:	Tiempo de arribo del tren m		
2.5	a la estación siguiente k .		
$ta_s[k]$:	Tiempo de arribo de los pa-		
	sajeros a la estación s en el		
	tiempo k.		

ecuación 4.1 muestra este procedimiento, donde tna_m y tna_s son el tiempo de llegada de los trenes y los pasajeros a una estación, respectivamente.

$$t[k] = \min_{m \in M, s \in S} (tna_m[k], tna_s[k])$$
(4.1)

El instante en el que cada tren llega a la estación siguiente se calcula a partir del tiempo anterior, el tiempo de detención y haciendo uso de la velocidad media del tren en los segmentos de línea entre las estaciones:

$$tna_{m}[k] = ta_{m}[k-1] + T_{m}[k-1] + \frac{L_{sa_{m}[k-1]}}{v}$$

, $\forall m \in M$ (4.2)

En la misma forma, el instante en que los pasajeros llegan a una estación se calcula mediante la realización de un proceso aleatorio, donde se considera el tiempo anterior más la realización de una variable aleatoria η que distribuye $exp(\lambda_s)$:

$$tna_s[k] = ta_s[k-1] + \eta , \ \forall s \in S$$

$$(4.3)$$

Comparando estos valores se establece el tiempo t[k] en que ocurre el próximo evento y se determina el tren o estación que lo gatilló. En adelante se usará el superíndice * para señalar al tren (m^*) sobre el cuál se realizan las actualizaciones y la estación (s^*) a la cual arribaron los pasajeros.

Si el evento gatillante corresponde al arribo de un tren a una estación la actualización es como sigue:

$$m^* = \arg\min_{m \in M} \left(tna_m[k] \right) \tag{4.4}$$

$$ta_{m^*}[k] = t[k] \tag{4.5}$$

Dado que el tren gatillante ha arribado a la estación s^* su índice debe ser actualizado usando la ecuación 4.6, esta representación asume que los trenes conservan el índice de la estación anterior que visitaron hasta visitar una nueva, es decir, en los tramos entre estaciones el índice de estación se ve inalterado.

$$sa_{m^*}[k] = sa_{m^*}[k-1] + 1 \tag{4.6}$$

Al llegar a la estación se determina el número de pasajeros que desean descender mediante la ecuación 4.7, que corresponde al total de pasajeros cuyo destino es la estación s^* :

$$pd[k] = pm_{(m^*, sa_{m^*})}[k-1]$$
(4.7)

El número de pasajeros que abordan el tren se ve limitado por la capacidad del tren, por lo tanto, el proceso de embarque se simula siguiendo el principio FIFO (First In First Out). Las ecuaciones 4.8 muestran cómo se actualiza la disponibilidad en el tren ($c^*[k]$) y la cantidad de pasajeros que logran subir para viajar a su destino s1, $pb_{s1}[k]$. Para ello se utilizan las variables, $pmt^*[k]$ y $ps'_{(sa_m^*,s1)}$ que respectivamente corresponden al número de pasajeros que están en el tren en el momento de la llegada y el número de pasajeros esperando en la estación truncado según la capacidad disponible del tren.

$$c^{*}[k] = C - (pmt^{*}[k] - pd[k])$$

$$pb_{s1}[k] = ps'_{sa_{m^{*}[k]},s1}[k] , \ \forall s1 \in S$$
(4.8)

De esta manera se actualiza el número de pasajeros que se encuentran en el tren m^* . Los pasajeros que bajan del tren deben son descontados, mientras que los que suben se asignan según su estación de destino:

$$pm_{(m^*,s)}[k] = \begin{cases} 0 & if \ s = sa_{m^*}[k] \\ pm_{(m^*,s)}[k-1] + pb_s[k] & if \ s \neq sa_{m^*}[k] \\ , \forall s \in S \end{cases}$$

$$(4.9)$$

El número de pasajeros en la estación se actualiza según el número de pasajeros que subió al tren:

$$ps_{(s^*,s1)}[k] = ps_{(s^*,s1)}[k-1] - pb_{s1}[k], \forall s1 \in S$$
(4.10)

El proceso de subida y bajada de pasajeros determina el tiempo de permanencia del tren recién arribado a la estación, de acuerdo con la ecuación 4.11. Este tiempo de espera considera la cantidad de pasajeros que suben y bajan del tren, y los pasajeros en la estación y el tren al momento del arribo (Wilson y Lin, 1993).

$$T_{m}[k] = \alpha + \beta \sum_{s1 \in S, s1 > s^{*}} pb_{s1}[k] + \gamma pd[k] +$$

$$\delta(\sum_{s1 \in S, s1 > s^{*}} ps_{(s^{*}, s1)}[k-1] + pmt^{*}[k])$$
(4.11)

Por otro lado, si el evento gatillante se debe al arribo de pasajeros a una estacion, la actualización contempla únicamente al número de pasajeros en ella,

$$s^* = \arg\min_{s \in S} \left(tna_s[k] \right) \tag{4.12}$$

$$ta_{s^*}[k] = t[k] \tag{4.13}$$

El número de pasajeros que arriba a la estación se determinan considerando el tamaño del grupo y la matriz origen-destino. Así, para cada pasajero, su destino se obtiene al azar de manera proporcional a lo indicado por la matriz origen-destino. El número de pasajeros actualizado esta dado por la ecuación 4.14, donde $psa_{(s^*,s1)}[k]$ indica el número de pasajeros que arribo a s^* cuyo destino es s1.

$$ps_{(s^*,s1)}[k] = ps_{(s^*,s1)}[k-1] + psa_{(s^*,s1)}[k] ,$$

$$\forall s1 \in S$$
(4.14)

El modelo presentado anteriormente se programó en un simulador dinámico llamado SimDinMetro, cuya interfaz de usuario se muestra en la siguiente figura. El simulador mostró ser fiable al sistema de Metro de Santiago, trabajo que puede observarse en Grube et al. (2011). Este simulador es utilizado como plataforma de simulación del sistema a observar y utilizando su interfaz con Matlab se logra ejecutar el algoritmo de estimación que se describe a continuación.



FIGURA 4.1. Interfaz de usuario de SimDinMetro. El simulador se ha modificado para permitir una programación más ágil de otros sistemas de transporte para una posible aplicación en minería.

4.2. Planteamiento del Problema

El problema que se resuelve presenta la estructura que se muestra en la Figura 4.2, donde se destacan las variables de entrada y salida del sistema presentado en la sección 4.1 que son utilizadas para realizar la estimación. De esta manera el problema consiste en utilizar el modelo presentado en la sección 4.1 para estimar el estado en sistemas híbridos de manera de acceder a variables que no es posible medir, que en este caso corresponden a la cantidad de pasajeros en las estaciones $(ps_{(s,s1)}[k])$ y el número de pasajeros en los trenes de un sistema de metro $(pm_{m,s})[k]$).

Es posible distinguir dos particularidades en la estructura planteada. La primera tiene relación con la arquitectura del observador, que se basa en la presentada por Balluchi et al. (2002), pues permite sacar provecho de la estructura del sistema. Este observador discreto detecta el modo de operación del sistema correspondiente al momento en que un tren hace entrada a alguna estación y la información requerida para esto es la posición de los trenes.



FIGURA 4.2. Estructura del sistema a observar. Las variables de entrada y salida se asumen conocidas en todo momento, lográndose estimar los pasajeros que esperan en las estaciones y viajan en los trenes, variables que a su vez también alimentan el observador.

Una vez que se detecta el modo de operación de interés, el filtro de partículas realiza una estimación del número de pasajeros que bajan del tren en base a la ecuación 4.11. El procedimiento de estimación y actualización de las variables del sistema luego del arribo de un tren a una estación se muestra en la Figura 4.3, donde se destacan las ecuaciones del modelo utilizadas. El proceso de estimación propiamente tal se realiza por medio de un filtro de partículas que simula diferentes tiempos de detención utilizando la ecuación 4.11 mediante una distribución binomial de la cantidad de pasajeros que descienden del tren. Es importante mencionar que si bien en esta investigación se ha decidido utilizar una representación discreta del número de pasajeros la metodología presentada permite una modelación continua sin que esto signifique modificaciones mayores. El cambio necesario consiste en reemplazar la distribución de probabilidad del estado discreta (binomial) por una continua (normal), siendo el algoritmo de selección y de actualización de variables el mismo que se describió anteriormente.



FIGURA 4.3. Etapas en el proceso de estimación. Las variables estimadas deben utilizarse en estimaciones futuras, introduciendo un comportamiento dinámico y retroalimentado a la estimación.

La estructura de la red de Metro posee características de un sistema distribuido, pues es posible crear subsistemas a partir de las líneas que lo componen; además dentro de cada línea una vía es independiente de la otra en términos de la cantidad de pasajeros que hay en sus andenes y trenes. Teniendo en cuenta esta propiedad se dispuso a simular un sistema de Metro simplificado que contempla una línea con 5 estaciones y 10 trenes.

4.3. Experimentos y Resultados

El método de estimación propuesto fue probado a través de una simulación con datos reales del sistema de Metro de Santiago de Chile. Dado que la metodología desarrollada es para sistemas de Metro con sólo una línea, la simulación utiliza datos de sólo una de las cinco líneas de la red de Metro de Santiago, la Línea 1. Esta línea consta de 24 estaciones, de las cuales tres son de combinación con otras líneas, demanda que no es tomada en

cuenta. La simulación se lleva a cabo en hora punta, en un lapso de tiempo que permite que todos los trenes pasen por todas las estaciones.

En una implementación real de este método la matriz de origen-destino es desconocida, por lo tanto en esta simulación, una ODM se utiliza para simular el sistema y otra ODM con al menos 15 % de la diferencia se utiliza en el proceso de estimación.

Las Figuras 4.4 y 4.5 muestran los resultados obtenidos, donde el número de pasajeros en la estación se grafica para cada la llegada de los trenes y los pasajeros que viajan en el tren se representan en cada parada del tren en las estaciones. La incertidumbre introducida por la falta de conocimiento de la matriz origen-destino de pasajeros que entran en la estación afecta la precisión de la estimación de pasajeros en la estación, ya que incluye la división entre las vías de la estación. En la estimación de los pasajeros que viajan en los trenes, el error se debe a que la ODM se utiliza para predecir la cantidad de pasajeros que van a descender del tren en las estaciones que debe visitar el tren posteriormente. Sin embargo, la incertidumbre se reduce en los extremos de la línea y por lo tanto el error en la estimación disminuye.

Se puede observar que los errores en la estimación se reducen a lo largo de la evolución del sistema, esto se debe dos propiedades del sistema. En primer lugar, la incertidumbre en la matriz origen-destino es menor en los extremos de la línea, por lo que la estimación del número de pasajeros que viajan mejora en las predicciones. En segundo lugar, la condición inicial de los trenes en la estación de partida es conocido (el tren comienza vacío) y también los pasajeros pueden tomar sólo una pista por lo que la estimación es precisa. Este efecto se puede ver en la Figura 4.6 donde las estaciones terminales presentan un error de estimación menor.

4.3.1. Análisis de Sensibilidad

Finalmente se incluye un análisis del efecto que producen en la estimación la perturbación en la medición de T y el desconocimiento de la matriz origen-destino de la simulación. El error medio de la estimación del número de pasajeros en los trenes y las estaciones se



(a) Número de pasajeros viajando en el tren 2. (b) Número de pasajeros viajando en el tren 16.

FIGURA 4.4. Evolución del número de pasajeros en los trenes 2 y 16 a lo largo de la línea de Metro. Es posible obtener una estimación precisa, donde los límites de la capacidad de los trenes de pasajeros (1300) se respetan.

TABLA 4.3. Análisis de sensibilidad en σ_T y la diferencia entre la ODM de simulación y estimación.

		ODM %				
		0 15 20 25				
	0	0	-10	-23	-41	
-	0.1	-5	-12	-39	-50	
$ $ O_T	1	-19	-29	-48	-69	
	10	-21	-31	-62	-87	

muestra en La Tabla 4.3 bajo distintos valores de σ_T y el porcentaje de diferencia entre las Matrices Origen-Destino usadas en la simulación y en la estimación, los datos ennegrecidos corresponden al caso mostrado por las Figuras 4.4 a 4.6.



(a) Número de pasajeros esperando en la estación 12. (b) Número de pasajeros esperando en la estación 16

FIGURA 4.5. Evolución del número de pasajeros en las estaciones 12 y 16 de la línea de Metro a medida que los trenes dejan las estaciones. La estimación difiere del valor real debido a la incertidumbre en la matriz origen-destino usada para simulación y aquella usada para la estiamción.

Es posible apreciar que la estimación presenta un sesgo positivo, por lo que tiende a subestimar el número de pasajeros en el sistema. Se ha incluido el caso en el que no existe ruido de medición en el tiempo de detención y que las ODM de simulación y estimación no difieren con el fin de realizar un chequeo de sanidad de la metodología, mostrándose que ante una medición exacta y con el conocimiento completo de la ODM el sistema es observable, mostrando un error de estimación nulo. Cabe mencionar que los parámetros del filtro de partículas (número de partículas, distribuciones, etc) se han dejado constantes y no se han ajustado a cada caso con el fin de observar el efecto de las perturbaciones. Es claro a través de la Tabla que el factor más importante y degradante en el proceso de estimación es la diferencia que existe entre la matriz origen-destino usada para el proceso de estimación



(a) Error de estimación medio en los trenes del sistema. (b) Error medio de estimación en el número de pasajeros que esperan en las estaciones.

FIGURA 4.6. Error medio de estimación con barras de error que indican la desviación estándar. El error de estimación es mayor en las estaciones debido a la incertidumbre de la matriz origen-destino, sin embargo este error disminuye en los extremos de la línea.

y aquella que realmente gobierna la dinámica del sistema, es por ello que acompañar este desarrollo con una metodología de estimación en línea de la ODM es fundamental. La literatura es basta en investigaciones sobre estimación en línea de matrices origen-destino, sin embargo se han centrado en transporte privado o en intersecciones de carreteras (Zhou y Mahmassani, 2003; Lin y Chang, 2007). Zúñiga (2011) propone una metodología de estimación en línea para un sistema de Metro que se basa en datos históricos y mediciones de la subida y bajada de pasajeros en las estaciones para estimar y predecir la matriz origen-destino. Esta metodología puede aportar al proceso de estimación un valor más preciso y actualizado de la matriz origen-destino, generando un efecto positivo en la precisión del método.

Capítulo 5. CONCLUSIONES Y TRABAJO FUTURO

5.1. Resultados y Comentarios Generales

En la presente tesis se presentaron las diferentes formas de representar los sistemas híbridos y los métodos de estimación de estados aplicados tanto a un ejemplo tipo *benchmark* como a un sistema de transporte público basado en una red de Metro. El uso de una representación híbrida se justifica debido a la presencia de diversos modos de operación presentes en el sistema de Metro y el uso de herramientas de estimación de estado se deben a que aplicaciones de control avanzado en gestión de flota requieren conocimiento completo del sistema, supuesto que no siempre es posible sostener en términos de implementación.

Los antecedentes bibliográficos recopilados en esta investigación contemplan un estudio de las maneras de representar sistemas híbridos distinguiéndose dos grandes modelos, uno que resulta más intuitivo (PWA) y otro más complejo que permite utilizar técnicas de estimación de estados más precisas (MLD). Ambos modelos son equivalentes y su conversión se puede realizar con ayuda de un lenguaje de descripción de alto nivel llamado HYS-DEL. El marco teórico descrito incluye una revisión de las diferentes técnicas de medición inferencial en sistemas híbridos, incorporando la estimación de estado y la identificación de sistemas. Cada una de estas áreas muestra ser de interés actualmente por la comunidad científica y su desarrollo aún no termina. Al respecto es importante destacar que el presente trabajo no pretende aportar en la creación de nuevas técnicas, si no una aplicación innovadora de lo que existe actualmente.

El ejemplo *benchmark* en el que se aplicaron las técnicas descritas anteriormente permite cuantificar las apreciaciones cualitativas que se presentan en la literatura de cada uno de los métodos de inferencia en sistemas híbridos. Este ejemplo muestra las ventajas y desventajas absolutas y comparativas de las diferentes técnicas. Así, por ejemplo, se logró determinar que el ruido en las mediciones es altamente perjudicial en el proceso de identificación de sistemas híbridos y que la mejor técnica es aquélla basada en clustering. En relación al proceso de estimación de estado la aplicación a un sistema *benchmark* hizo notar las ventajas del filtro de partículas sobre estimación de horizonte móvil en cuanto a estimación de variables discretas se refiere. Logrando obtener así una decisión informada con hechos cualitativos sobre qué metodología utilizar en el sistema de Metro, decisión que se caracteriza por la flexibilidad y rapidez de cómputo del primer método.

Como innovación destacamos la utilización de metodologías de estimación de estados a un modelo híbrido de red de Metro con el fin de lograr estimar el número de pasajeros que viajan en los trenes y aquéllos que esperan en las estaciones. El modelo utilizado para describir y simular una red de Metro se basó en un trabajo anterior de efectividad y representatividad comprobada. Los resultados que se obtuvieron son de alta precisión, con un error medio de estimación menor a 30 pasajeros en la estimación del número de pasajeros que viajan en los trenes y estaciones; sin embargo, el desconocimiento de la matriz de origen destino que determina los viajes de los pasajeros que entran a las estaciones afecta la estimación de la cantidad de pasajeros que esperan en las estaciones. Esto último se refleja en un error de estimación con mayor varianza y valor medio.

Un análisis de sensibilidad muestra además que la incertidumbre en la matriz de origen destino es el parámetro que afecta mayormente el desempeño de la estimación, pudiendo duplicar el error medio con un aumento de incertidumbre de tan sólo el 10%. Esta sensibilidad sugiere incluir una metodología de estimación dinámica de la matriz origen-destino similar a la propuesta por Zúñiga (2011). A pesar del desconocimiento de la ODM los errores de estimación son pequeños en comparación con las capacidades de los trenes y estaciones, especialmente en las estaciones de los extremos de la línea, donde la incertidumbre es menor debido a propiedades intrínsecas del sistema; los trenes comienzan con un estado conocido (vacíos) y en el extremo final de la línea la matriz origen destino se conoce con mayor precisión (los pasajeros tienen menos posibilidades de destinos).

El presente trabajo muestra que es posible utilizar métodos de estimación de estado en sistemas híbridos, en específico el filtro de partículas, con especial interés en la estimación de variables que no es posible medir en sistemas de transporte como una línea de Metro. La

obtención de dichas variables es fundamental si se desea implementar sistemas de control avanzado como MPC con el objetivo de mejorar la calidad de servicio de los sistemas.

5.2. Temas de Investigación Futura

Es posible distinguir dos grandes áreas de profundización de la investigación realizada. Una de ellas contempla el mejorar el desempeño del estimador mediante nuevas técnicas o la utilización de un modelo más completo para describir el tiempo de detención de los trenes. Junto con esto se debe contemplar el realizar pruebas en una red de Metro de mayor tamaño y complejidad (estaciones de combinación, etc.). Otro trabajo futuro de validación en terreno, incluye la utilización de diversos sensores que permitan obtener las variables medidas (posición de trenes, tiempo de detención, etc.) y las variables a estimar para realizar un análisis de confiabilidad de las estimaciones.

Como área de expansión de este trabajo se propone aplicar la metodología de estimación a una red de buses en corredores, de modo de estimar la cantidad de gente que viaja en ellos y aquellos que esperan en los paraderos. Se contempla además un posible uso en otros sistemas de transporte como los de la gran minería, ya sea en su modalidad de camiones, trenes o correas transportadoras. Esta posible extensión se sustenta en la flexibilidad tanto de la formulación como en la resolución del problema de estimación, pues el modelo requiere de una expresión que relacione la variable de interés con una medición y una serie de variables medibles y el filtro de partículas es completamente transparente a las características de este modelo, pudiendo ser lineal, no lineal, discreto, continuo o híbrido.

Finalmente se propone la realización de una investigación futura que reúna los resultados alcanzados en esta tesis con aquéllos relacionados con el control de flota de una red de Metro. Importante es destacar que el modelo a utilizar para dicha investigación debe permitir una descripción distribuida de la red de Metro, pues la metodología mostrada en esta tesis realiza la estimación de estado de manera local, tan sólo la estación que se encuentra activa debido a la llegada de un tren. Es decir, el sistema completo del Metro es dividido en subsistemas, estaciones, donde la comunicación entre estos ocurre mediante los trenes.

BIBLIOGRAFÍA

Alessandri, A., y Coletta, P. (2001). Switching observers for continuous-time and discrete-time linear systems. En *Proceedings of the American Control Conference*. (Vol. 3, p. 2516 -2521).

Alessandri, A., y Coletta, P. (2003). Design of observers for switched discrete-time linear systems. En *Proceedings of the American Control Conference*. (Vol. 4, p. 2785 - 2790).

Alvarado, I., Limon, D., La Pena, D. M. de, Maestre, J. M., Ridao, M. A., Scheu, H., y cols. (2011). A comparative analysis of distributed mpc techniques applied to the hd-mpc four-tank benchmark. *Journal of Process Control*, *21*(5), 800–815.

Arogeti, S., Danwei, W., y Chang, B. L. (2010). Mode identification of hybrid systems in the presence of fault. *IEEE Transactions on Industrial Electronics*, 57(4), 1452 -1467.

Arulampalam, M. S., Maskell, S., y Gordon, N. (2002). A tutorial on particle filters for online nonlinear/non-gaussian bayesian tracking. *IEEE Transactions on Signal Processing*, *50*, 174–188.

Balluchi, A., Benvenuti, L., Benedetto, M. D., y Sangiovanni-Vincentelli, A. (2002). Design of observers for hybrid systems. En C. Tomlin y M. Greenstreet (Eds.), *Hybrid Systems: Computation and Control* (Vol. 2289, p. 59-80). Berlin, Heidelberg: Springer-Verlag.

Barbot, J., Djemai, M., y Manamanni, N. (2007). State observer and observability conditions for a class of hybrid continuous-discrete dynamic system. En *Proceedings* of the 46th IEEE Conference on Decision and Control. (p. 708-713).

Bemporad, A., Ferrari-Trecate, G., y Morari, M. (2000). Observability and controllability of piecewise affine and hybrid systems. *IEEE Transactions on Automatic Control*, 45(10), 1864 - 1876.

Bemporad, A., Garulli, A., Paoletti, S., y Vicino, A. (2003). A greedy approach to identification of piecewise affine models. En *Proceedings of the 6th International Conference on Hybrid Systems: Computation and Control* (pp. 97–112). Berlin, Heidelberg: Springer-Verlag.

Bemporad, A., Garulli, A., Paoletti, S., y Vicino, A. (2004). Data classification and parameter estimation for the identification of piecewise affine models. En *Proceedings of the 43rd IEEE Conference on Decision and Control.* (Vol. 1, p. 20 - 25).

Bemporad, A., Mignone, D., y Morari, M. (1999). Moving horizon estimation for hybrid systems and fault detection. En *Proceedings of the American Control Conference*. (Vol. 4, p. 2471 -2475).

Bemporad, A., y Morari, M. (1999). Control of systems integrating logic, dynamics, and constraints. *Automatica*, *35*(3), 407–427.

Bemporad, A., Roll, J., y Ljung, L. (2001). Identification of hybrid systems via mixed-integer programming. En *Proceedings of the 40th IEEE Conference on Decision and Control.* (Vol. 1, p. 786 -792).

Blom, H., y Bloem, E. (2004). Particle filtering for stochastic hybrid systems. En *43rd IEEE Conference on Decision and Control.* (Vol. 3, p. 3221 - 3226).

Cassandras, C. G., Clune, M. I., y Mosterman, P. J. (2006). Hybrid System Simulation with SIMEVENTS. En *Analysis and Design of Hybrid Systems* (pp. 267–269). Amsterdam: Elsevier. Cinquemani, E., y Micheli, M. (2006). State estimation in stochastic hybrid systems with sparse observations. *IEEE Transactions on Automatic Control*, *51*(8), 1337–1342.

Cinquemani, E., Micheli, M., y Picci, G. (2004). State estimation and prediction in a class of stochastic hybrid systems. En *Proceedings of the 16th International Symposium on Mathematical Theory of Networks and Systems*.

Clune, M., Mosterman, P., y Cassandras, C. (2006). Discrete event and hybrid system simulation with simEvents. En *Proceedings of the 8th International Workshop on Discrete Event Systems* (pp. 386–387).

Cortés, C. E., Sáez, D., Milla, F., Núñez, A., y Riquelme, M. (2010). Hybrid predictive control for real-time optimization of public transport systems' operations based on evolutionary multi-objective optimization. *Transportation Research Part C: Emerging Technologies*, *18*(5), 757 - 769.

Daigle, M. J., Koutsoukos, X. D., y Biswas, G. (2010). An event-based approach to integrated parametric and discrete fault diagnosis in hybrid systems. *Transactions of The Institute of Measurement and Control*, *32*(5), 487-510.

Delgado, F., Muñoz, J., Giesen, R., y Cipriano, A. (2009). Real-time control of buses in a transit corridor based on vehicle holding and boarding limits. *Transportation Research Record: Journal of the Transportation Research Board*, 2090(1), 59–67.

Di Benedetto, M. D., Di Gennaro, S., y D'Innocenzo, A. (2009). Discrete state observability of hybrid systems. *International Journal of Robust and Nonlinear Control*, 19(14), 1564–1580.

Doucet, A., Gordon, N., y Krishnamurthy, V. (2001). Particle filters for state estimation of jump markov linear systems. *IEEE Transactions on Signal Processing*, *49*(3), 613-624. Farina, M., Ferrari-Trecate, G., y Scattolini, R. (2010a). Distributed moving horizon estimation for linear constrained systems. *IEEE Transactions on Automatic Control*, *55*(11), 2462 -2475.

Farina, M., Ferrari-Trecate, G., y Scattolini, R. (2010b). Moving-horizon partitionbased state estimation of large-scale systems. *Automatica*, *46*(5), 910 - 918.

Feng, C., Lagoa, C., Ozay, N., y Sznaier, M. (2010). Hybrid system identification: An sdp approach. En *Proceedings of the 49th IEEE Conference on Decision and Control.* (p. 1546 -1552).

Ferrari-Trecate, G., Mignone, D., y Morari, M. (2002). Moving horizon estimation for hybrid systems. *IEEE Transactions on Automatic Control*, 47(10), 1663 - 1676.

Ferrari-Trecate, G., Muselli, M., Liberati, D., y Morari, M. (2001). Identification of piecewise affine and hybrid systems. En *American Control Conference* (p. 3521-3526).

Freitas, N. de. (2002). Rao-blackwellised particle filtering for fault diagnosis. En *Proceedings of the IEEE Aerospace Conference*. (Vol. 4, p. 4-1767 - 4-1772).

Funiak, S., y Williams, B. (2003). Multi-modal particle filter for hybrid systems with autonomous mode transitions. En *Proceedings of 14th International Workshop Principles Diagnosis*. (Vol. DX03).

Gordon, N., Salmond, D., y Smith, A. (1993). Novel approach to nonlinear/nongaussian bayesian state estimation. En *IEE Proceedings of Radar and Signal Processing*. (Vol. 140, p. 107 -113).

Gray, B. M. A. (2007). Discrete Event Simulation: a Review of SimEvents. *Computing in Science & Engineering*, *9*(6), 62–66.

Grube, P., Núñez, F., y Cipriano, A. (2011). An event-driven simulator for multiline metro systems and its application to Santiago de Chile metropolitan rail network. *Simulation Modelling Practice and Theory*, *19*(1), 393 - 405.

Hamdi, F., Manamanni, N., Messai, N., y Benmahammed, K. (2008). Hybrid observer design for switched linear systems using differential petri net. En *Proceedings of the 16th Mediterranean Conference on Control and Automation*. (p. 95 -100).

Haseltine, E. L., y Rawlings, J. B. (2005). Critical evaluation of extended kalman filtering and moving-horizon estimation. *Industrial & Engineering Chemistry Research*, *44*(8), 2451-2460.

Juloski, A., Heemels, M., y Weiland, S. (2007). Observer design for a class of piecewise linear systems. *International Journal of Robust and Nonlinear Control*, *17*(15), 1387 -1404.

Juloski, A., Heemels, W., Ferrari-Trecate, G., Vidal, R., Paoletti, S., y Niessen, J. (2005). Comparison of four procedures for the identification of hybrid systems. En M. Morari y L. Thiele (Eds.), *Hybrid Systems: Computation and Control* (Vol. 3414, p. 354-369). Springer Berlin / Heidelberg.

Juloski, A., Weiland, S., y Heemels, W. (2005). A bayesian approach to identification of hybrid systems. *IEEE Transactions on Automatic Control*, *50*(10), 1520 -1533.

Koutsoukos, X. (2003). Estimation of hybrid systems using discrete sensors. En *Proceedings of 42nd IEEE Conference on Decision and Control.* (Vol. 1, p. 155 - 160).

Koutsoukos, X., Kurien, J., y Zhao, F. (2003). Estimation of distributed hybrid systems using particle filtering methods. En *Hybrid Systems: Computation and Control. Springer Verlag Lecture Notes on Computer Science* (pp. 298–313). Springer.

Lauer, F., Bloch, G., y Vidal, R. (2010). Nonlinear hybrid system identification with kernel models. En *Proceedings of the 49th IEEE Conference on Decision and Control.* (p. 696 -701).

Lauer, F., Bloch, G., y Vidal, R. (2011). A continuous optimization framework for hybrid system identification. *Automatica*, 47(3), 608 - 613.

Lin, P.-W., y Chang, G.-L. (2007). A generalized model and solution algorithm for estimation of the dynamic freeway origin-destination matrix. *Transportation Research Part B: Methodological*, *41*(5), 554 - 572.

Ma, Y., y Vidal, R. (2005). Identification of deterministic switched arx systems via identification of algebraic varieties. En M. Morari y L. Thiele (Eds.), *Hybrid Systems: Computation and Control.* Springer Berlin / Heidelberg.

Morales-Menéndez, R., Freitas, N. de, y Poole, D. (2003). Estimation and control of industrial processes with particle filters. En *Proceedings of the 2003 American Control Conference*. (Vol. 1, p. 579 - 584).

Niessen, J., Juloski, A., Ferrari-Trecate, G., y Heemels, W. (2004). Comparison of three procedures for the identification of hybrid systems. En *Proceedings of the IEEE International Conference on Control Applications*. (Vol. 1, p. 643 - 648).

Núñez, F., Reyes, F., Grube, P., y Cipriano, A. (2011). Simulating railway and metropolitan rail networks: From planning to on-line control. *IEEE Intelligent Transportation Systems Magazine*, 2(4), 18–30.

Paoletti, S., Juloski, A., Ferrari-Trecate, G., y Vidal, R. (2007). Identification of hybrid systems. *European Journal of Control*, *13*(2-3), 242–260.

Pina, L., y Botto, M. A. (2006). Simultaneous state and input estimation of hybrid systems with unknown inputs. *Automatica*, 42(5), 755 - 762.

Rao, C. V., Rawlings, J. B., y Lee, J. H. (2001). Constrained linear state estimation–a moving horizon approach. *Automatica*, *37*(10), 1619 - 1628.

Rao, C. V., Rawlings, J. B., y Mayne, D. Q. (2003). Constrained state estimation for nonlinear discrete-time systems: Stability and moving horizon approximations. *IEEE Transactions on Automatic Control*, *48*, 246–258.

Rawlings, J. B., y Bakshi, B. R. (2006). Particle filtering and moving horizon estimation. *Computers & Chemical Engineering*, *30*(10-12), 1529 - 1541.

Samadi, M., Salahshoor, K., y Safari, E. (2009). Distributed particle filter for state estimation of hybrid systems based on a learning vector quantization algorithm. En *IEEE International Conference on Control and Automation*. (p. 1449 -1453).

Sanz, V., Urquia, A., Cellier, F. E., y Dormido, S. (2012). Modeling of hybrid control systems using the DEVSLib Modelica library. *Control Engineering Practice*, *20*(1), 24–34.

Sontag, E. (1981). Nonlinear regulation: The piecewise linear approach. *IEEE Transactions on Automatic Control*, 26(2), 346–358.

Thomas, J., Dumur, D., Buisson, J., Falinower, C., y Bendotti, P. (2003). Moving horizon state estimation of hybrid systems. application to fault detection of sensors of a steam generator. En *Proceedings of the 2003 IEEE Conference on Control Applications* (Vol. 2, p. 1375 - 1380).

Torrisi, F., y Bemporad, A. (2004). Hysdel — a tool for generating computational hybrid models for analysis and synthesis problems. *IEEE Transactions on Control Systems Technology*, *12*, 235–249.

Valdes-Gonzalez, H., y Flaus, J. (2001). State estimation in a bioprocess described by a hybrid model. En *Proceedings of the 2001 IEEE International Symposium on Intelligent Control* (p. 132 -137).

Vapnik, V. N. (1998). *Statistical learning theory* (1.^a ed.). Wiley.

Vidal, R. (2004). Identification of pwarx hybrid models with unknown and possibly different orders. En *Proceedings of the 2004 American Control Conference*. (Vol. 1, p. 547 -552).

Vidal, R., Soatto, S., Ma, Y., y Sastry, S. (2003). An algebraic geometric approach to the identification of a class of linear hybrid systems. En *Proceedings of the 42nd IEEE Conference on Decision and Control* (Vol. 1, p. 167 - 172).

Wang, W., Zhou, D., y Li, Z. (2006). Robust state estimation and fault diagnosis for uncertain hybrid systems. *Nonlinear Analysis*, 65(12), 2193 - 2215.

Wilson, N. H. M., y Lin, T. (1993). Dwell-time relationships for light rail systems. *Transportation Research Record*, *1361*, 296–304.

Zhao, F., Koutsoukos, X., Haussecker, H., Reich, J., y Cheung, P. (2005). Monitoring and fault diagnosis of hybrid systems. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics, 35*(6), 1225 -1240.

Zhou, X., y Mahmassani, H. S. (2003). Dynamic origin-destination demand estimation using multi-day link traffic counts for planning applications. *Journal of Transportation Research Board*(1831), 30–38.

Zúñiga, F. (2011). Estimación y predicción de matrices dinámicas de viaje sobre un corredor de transporte público, utilizando datos históricos e información en tiempo real. Tesis de Magister en Ciencias de la Ingeniería, Pontificia Universidad Católica de Chile.

ANEXO A. PRODUCTOS DE LA TESIS

A.1. Paper ITS Magazine



© LUSHPIX

Abstract—Train-based systems are the principal means of public transportation in many of the world's cities, and continue to grow in the face of rising demand. Expanding infrastructure is costly, however, and at a certain point becomes unsustainable. When this occurs the only feasible solution is to improve the management system. This is done by using either offline or online intelligent transportation systems which requires prior analysis and testing. These previous activities are not easy to carry out in the transportation system itself because of high costs and possible drawbacks. The usual solution in these cases involves conducting simulations. Simulating a train system is a complex problem for which several software applications have been designed, using different models, programming approaches, and simplifications. Therefore, selecting the best simulator for testing a particular intelligent system is a hard task that needs atention. In this work, the requirements that a simulator must fulfill in order to be suitable for testing a particular system are stated. For each class of application, examples of available simulators are given and their main characteristics are then analyzed. Finally, as a practical example, the problem of evaluating skip-stop policies in a multi-line Metro system is studied using a novel event-driven simulator.

Keywords—Public transportation, Railway systems, Metro systems, Computer-aided simulation, Intelligent transportation systems.

Digital Object Identifier 10.1109/MITS.2010.939923 Date of publication: 4 February 2011
Simulating Railway and Metropolitan Rail Networks: From Planning to On-Line Control

Felipe Núñez, Francisco Reyes, Pablo Grube, and Aldo Cipriano College of Engineering, Pontificia Universidad Católica de Chile E-mails: fenunez@ing.puc.cl, frreyes@uc.cl, pgrube@ing.puc.cl, and aciprian@ing.puc.cl

I. Introduction

ransportation has become a major issue all around the world. Increasing traffic congestion, as a direct result of excessive car usage, has significantly reduced the efficiency of transportation infrastructure and increased travel time, air pollution, and fuel consumption. As a result, cities have been faced with the necessity of improving public transportation systems to reduce car usage and, thus, restore the system's efficiency. Many of the largest cities in the world have a multi-modal system consisting of buses and light or metropolitan trains, both of them in charge of transporting a continuously rising demand. One option for serving this huge demand is to expand the infrastructure; however, there is a practical limit in such a strategy, especially for train systems. When it is not possible to continue an infrastructure expanding plan, including stations construction, the only feasible solution for serving passengers with a proper quality of service is to improve the trains' management system by using intelligent transportation systems. These systems can be based on offline methods, such as pre-programming schedules; or online methods, which employ system status information obtained online during operation including trains' positions, number of passengers at stations and on trains, and eventual disturbances affecting the system's normal operation, among others.

The introduction of a new planning or management system, be it on or off line, requires prior testing that, at least in the initial stage, is not easy to carry out in the transportation system itself because of the costs involved and the effects on users. The usual solution in these cases involves conducting simulations, and in fact there are many simulation platforms available for these types of systems, each of them designed for a specific purpose, thus different in terms of model implemented, programming structure, and information delivered as output. These differences between simulation platforms make it extremely difficult to select the best one for testing a given intelligent system. The fact that each system aims to improve a specific part of the transportation system makes imperative that the platform selected for testing can simulate the system with a proper level of detail; and moreover, can deliver useful information for the evaluating process. Taking this fact into account, it is possible to build a list of requirements that a simulation platform must fulfill in order to be suitable for testing. These requirements should be the main point to consider when looking for a simulation platform.

In this work, the requirements that a simulator must fulfill in order to be suitable for testing a particular intelligent transportation system are stated. For each type of system, examples of available simulators are presented and their main characteristics are then analyzed to provide an overview of each simulation platform. Finally, as a practical example, the problem of evaluating skip-stop policies in a multi-line Metro system is studied using SimDinMetro, a novel event-driven simulator specially designed for evaluating on-line control algorithms.

II. General Definitions

When analyzing simulators and their specifications, the user is faced with several terms which often refer to programming techniques, scale of detail, and/or the nature of the model implemented. Then, it is worth defining what a railway is, what a metropolitan rail network is, and what the most used simulating terminology refers to.

A. Railway and Metropolitan Rail Networks

Train-based transportation systems cannot be considered as one big set. Differences between their operating strategy and infrastructure suggest that a sub-division should be done. In this work, transportation systems based on trains are broken down into two sub-sets: railway systems, and metro systems.

The first difference between them worth being considered lies on geo-demographics; railway systems operate transporting passengers between urban areas-that is why they are sometimes called inter-urban systemswhile metro operates inside a metropolitan area. This geo-demographic difference provides a first parameter to classify train systems. However, perhaps the main differences between both systems lie on infrastructure. Railway lines can be unidirectional or bidirectional lines depending on the geographical location and rail system purpose, while metro systems operate on a fully bidirectional line. Differences in infrastructure are not only limited to lines; in terms of stations, clear differences can be recognized: interlocking systems of large railway stations are extremely complex, while in metro tracks crossing is a rare phenomenon. Another main difference is that railway systems have longer inner-station segments, thus, a lower density in terms of stations per kilometer. This is a direct consequence of the inter-urban characteristic of railway systems which are designed to cover longer distances than metro systems. Another important difference is that stations in metro systems are prepared to serve a continuous demand during the whole operating period, while in railway systems stations serve passengers only at specific instants. Regarding the operational strategies, railway systems operate using a pre-programmed schedule, known as timetable, while metro is a high-frequency system where the only information available is the system's starting and ending hours. Considering the differences stated above, it is not surprising that intelligent systems used are also different, not only in terms of implementation, but also in terms of objectives.

B. Differences on Simulating Scale

All simulators are based on models of the process under study, and train systems are not the exception. The model used in the simulator designing stage fully determines the simulator's capabilities and attributes. Several models for trains, railway, and metro systems are available; the reader is referred to [1], [2] for more information about modeling.

When analyzing simulators' capabilities and specifications, the user will find concepts like macroscopic or microscopic simulations. These concepts refer to the level of detail the model on which the simulator is based considers; in other words, the level of detail at which the interacting entities are treated.

In macroscopic simulations similar basic entities are grouped in clusters whose properties attain to all the individual entities. This allows characterizing the process using variables related more to the process itself than to basic entities. A good example is highway analysis where cars are treated as a continuous flux rather than considering each car as an individual entity with its own direction and speed. This strategy is useful when the user is interested only in the general behavior of the system.

On the other hand, in microscopic simulations the model considers independently each individual basic entity such as persons, vehicles or trains. This strategy is useful when the user is interested in the individual behavior of each entity. Usually, microscopic simulations are more demanding than macroscopic simulations in terms of computation and processing time.

Besides macroscopic and microscopic approaches, a hybrid scheme can be found. This hybrid scheme combines microscopic characterization for some entities and a macroscopic one for others. In transportation systems, hybrid approaches are commonly used, being typical to consider passengers as a continuous flux characterized using rates, while transporting units (cars, trains, etc) are treated as individual entities. A less common, worth mentioning



FIG 1 OpenTrack. Source [10].

though, hybrid approach considers entities individually but the interaction between entities of different nature is done on a macroscopic scale [3].

For more information about microscopic and macroscopic models, and their application to traffic systems, the reader is referred to [4].

C. Differences on Time Treatment

Time treatment is also a key point in simulations. The simulation process must take into account the evolution of time, which is not always considered as a continuous variable. Time is treated using two main paradigms: considering only those time instants when significant events happen, or considering the process as a continuous evolution of variables, then considering the time as a continuous variable. When the former paradigm is used, the simulation is said to be event-driven, while when the latter is used, the simulation is continuous.

Event-driven simulations are based on a set of important incidents which are defined as events. When an event takes place, simulation time and state variables of the model are updated and remain unmodified until the next event happens. Therefore, in event-driven simulations the time interval between two steps of calculation is based on events' occurrences and no longer constant [5]; thus, the processing is only conducted at irregular moments of discrete time. What may happen between steps is not taken into account. The advantage of this is to reduce significantly the processing requirements, although details about the process are naturally lost. Event-driven simulations can also be synchronous or asynchronous, meaning that in the latter predefined high priority entities are simulated first. In [6] a set of disparate problems are solved using asynchronous eventdriven simulations.

On the other hand, continuous simulations consist of integrating dynamic differential or difference equations that define the system's behavior at each moment in time. Time evolves at well-defined periodic time instants and state variables are updated at every step. As a consequence, continuous simulations are often more processing demanding than event-driven ones. Sometimes the use of continuous simulations is required though, as when studying the dynamic behavior of a train along the line.

As with scale of detail, time treatment can also be implemented following a hybrid scheme. A good example is given in [7] where the simulation is based on events which fire simulations of continuous dynamic systems.

III. Signaling Management

In railway systems, it is extremely important to count with a reliable signaling system in charge of controlling trains' movement along the line. Signaling management problems consist of finding the signal configuration which ensures safety and, at the same time, optimizes the system's performance. In order to find the optimal configuration, simulations are used.

A. Requirements

In order to be suitable for testing signaling systems, simulators must be able to:

- Generate basic signaling—stops signals and speed limits at least—and to generate restrictions on trains' movements based on those signals.
- Deliver reports regarding system's performance, including active restrictions caused by signals, travel times, and congestion situations.

Moreover, in order to assist the user in designing signaling systems, the simulator should be able to:

- Optimize the signals configuration with respect to a performance index.
- Validate the signaling system, meaning eliminate unnecessary signals.

Signaling management analysis requires trains to be treated as individual entities, i.e., requires the use of microscopic simulation. It also requires a microscopic model of the line, which must consider each section between stations with a great level of detail. It should be noted that the signaling management problem is independent of passengers; therefore, the way the simulator considers passengers microscopic or macroscopic—is not an issue to be considered. Also, there is no restriction on time treatment.

B. Available Simulators

Within the big universe of available simulators, it is possible to find several software applications suitable for signaling testing. Among them, there are some which allow the user to implement signaling systems, but do not provide support for optimization. One example is Bahn [8], a shareware program used for designing and testing train or streetcar transportation networks, which focuses more on learning. Bahn is able to simulate large railway or metro systems using dynamic models for trains. In fact, there are available models for the most popular train brands such as Gotha, Siemens, and Alstom. Other example is RailNet [9], a commercial software used for testing static, dynamic, or hybrid signaling systems. RailNet is also able to simulate different faults in any section of the system which makes it a very attractive option for operating analysis. RailNet is part of a complete commercial suite (see Figure 2) designed to analyze railway systems.

Other set of simulators are those which provide the user with tools to optimize and validate signaling systems. One of them is OpenTrack [10], which is very useful for analyzing the effects of installing new infrastructure, establishing schedules, experimenting with different signal systems, and analyzing the effects of failures [11]. OpenTrack itself is a microscopic model which simulates rail system operations based on user defined train, infrastructure, and timetables databases. The software uses a mixed discrete-continuous simulation process and object-oriented programming. The purpose of OpenTrack is to provide a microscopic platform for railroad simulating. Figure 1 shows a real implementation in OpenTrack. Another available option is Railsim [12], a commercial software which offers tools to validate signaling systems, and also delivers a complete report of incidents caused by the signaling system, including trains' delays. Ontrack [13], yet another commercial software, provides tools to design safety signaling systems. Finally, Rail//SYS [14] has been designed to study the effect of signaling systems on trains' schedules.



FIG 2 RailNet structure. Source [9].

It should be noted that simulators able to simulate and evaluate signalling systems are oriented mainly to railway systems. This is not surprising, given that signalling is a key point in railway systems, while in metro systems signalling is merely a supporting system used for safety purposes.

IV. Timetable Generation

In train-based transportation systems, especially in railway systems, it is customary to publish departure and arrival times. Punctuality is an important measure of operation performance in railway systems, and is often used as the main quality indicator. Therefore, in order to provide a high quality of service to customers, the system should operate following the timetable as close as possible. Timetable generation is a complex task; it is basically an optimization problem which involves trains, signaling, and passengers. To generate a robust and optimal timetable, simulations are used.

A. Requirements

In order to be suitable for timetable generation, the selected simulator must be able to:

- Generate and optimize timetables.
- Simulate trains' behavior using a microscopic scale, i.e., to simulate the system considering each train as an independent entity.
- Deliver reports of the system's performance when operating using a given timetable.

Timetable generation does not require a specific level of detail on passengers; thus, passengers can be treated using a microscopic or a macroscopic approach. Regarding time treatment, simulators used can be either event-driven or continuous. It should be noted that to test the robustness of the constructed timetables, macroscopic approaches are used, which simulate the entire railway network.

B. Available Simulators

One example of a simulator suitable for timetable generation is RailSys [15], which allows the user to design timetables and then to simulate the system. It integrates a combined timetable-infrastructure manager with synchronous microscopic simulation and automatic dispatching. RailSys has been successfully applied in timetable construction, infrastructure planning, and planning of logistics for large scale projects [16]. As output RailSys delivers a report where the user can appreciate the quality of service provided when operating using a given timetable. However, optimization of timetables is not supported. Similar capabilities are provided by RailNet and Rail//Sys. More powerful tools are those included in Topsu-RDM [17] and SiSyFe [18], which have an algorithm capable of generating optimal timetables. Deeper analysis can be made using OpenTrack which develops multiple independent montecarlo simulations to help in finding the optimal timetable,



FIG 3 Strategy to optimize power consumption. Source [22].

and then delivers as output reports of a schedule's robustness. Railsim conducts sensitivity analysis of timetables by introducing disturbances (delays, faults, etc) on the system and then evaluating the effects in terms of quality of service. Rail Traffic Controller [19] is another option to evaluate timetables; it detects congestion and suggests variations to the timetable to avoid congestion problems.

Timetable generation is also a field more related to railway than metro systems. The main reason is because in metro systems arrivals and departures are not guided by a fixed timetable, yet are controlled on-line by using a dynamic traffic management system.

V. Energy Management

One field that has gained attention during recent years is energy management. The objective here is to operate the system with the higher quality of service, while minimizing energy consumption. Simulation allows studying energy consumption of different operating strategies.

A. Requirements

- In order to be suitable for studying energy management techniques, the simulator must be able to:
- Simulate the system using a dynamic train model which considers power consumption.
- Simulate the system using a continuous time approach rather than an event-driven one.
- Deliver reports of system's performance, especially power consumption

Moreover, to assist the user in designing the best energy policy, the simulator should be able to:

 Optimize the power consumption subject to operating restrictions by manipulating variables such as trains speed.

When analyzing energy consumption, it is possible to use either a microscopic or a macroscopic approach. In the former case, each train is treated as an individual entity and the objective is to optimize the energy consumption of each train. On the other hand, when using a macroscopic scheme, trains are treated as a set and they are characterized using a power density defined as the amount of power consumed per line or per track segment. In this case the simulation may be oriented to analyze the power system rather than the transportation system itself.

B. Available Simulators

Several simulators deliver energy consumption information. OpenTrack is able to calculate the total power consumption of the system. Moreover, by using OpenPowerNet [20], an academic add-on that co-simulates the railway network, detailed energy consumption can be obtained. OnTrack also has a special module in charge of determining the powerflux of the whole railway system. One simulator which allows the user to obtain information of individual trains' energy consumption is the one designed by The Beijing Jiao-Tong University [21].

Energy consumption has also gained the attention of the academic community, as example, in [22] Railsim is used to simulate Zagreb's railway system with the aim to minimize energy losses. Figure 3 shows the flux diagram of the strategy presented in [22].

Energy consumption is a hard task, mainly because it requires counting with a detailed model of trains and the electric network which provides energy to the system, making it difficult to design an optimal energy policy suitable for all railway or metro systems.

VI. On-Line Control

On-line control has become one of the most studied disciplines around the world. It refers to a continuous modification of variables of the system to achieve a desired objective, the control objective. In train-based systems, for both railway and metro, several approaches have been implemented; one example is the automatic control strategy implemented on line 14 of the Paris metro system, which can be found in [23]. A more advanced example can be found in [24] where model predictive control is used to address the problem of scheduling.

In this work, all those strategies, or operating policies, which use information of the system obtained during operation, are considered as on-line control.

A. Requirements

In order to be suitable for testing an on-line control strategy, the selected simulator must be able to:

- Simulate the system using a microscopic scale for entities to be controlled.
- Simulate in real-time the system under analysis.
- Manipulate in real-time variables selected as manipulated variables.
- Deliver reports of system's performance.
- Moreover, in order to assist the user in determining control laws, the simulator should be able to:
- Optimize a cost function with respect to the set of manipulated variables.
- Provide the user with a set of possible manipulated variables to control the system during simulations. It is worth mentioning that some

on-line control approaches may require a particular paradigm for time treatment or a particular scale of detail that other approaches do not. On-line control is a wide field, then, it is not possible to establish strong conditions neither for time treatment nor for scale of detail.

B. Available Simulators

There are several simulators suitable for on-line control, varying from academic simulators to complex commercial platforms. Examples of these platforms are the Quadstone Paramics [25] and PTV Vissim micro-simulators [26], which are highly complex and costly. While these platforms are useful for definitively validating of algorithms, they are not at all practical for carrying out rapid testing during the development phase.

Due to the huge variety of simulators suitable for online control, it is worth introducing them based on control strategies. The simplest on-line control strategy is headway control, which aims to regularize the time between successive train departures from a given station and is mainly implemented in metro systems. Examples of simulators able to implement headway control are: the one developed at The University of Palermo [27], the one developed at Pontifical University of Comillas [28], and the one developed at The University of Beijing Jiao-Tong. These simulators deliver performance reports which include passengers' waiting and travel time. However, they cannot simulate faults on trains or at stations. Nevertheless, Kutty [29], vet another academic-developed software, is able to analyze the system's performance when faults take place. In [30] a fault analysis is presented for the Tokyo metro system. Figure 4 describes a control strategy



FIG 4 Implementation in Kutty. Source [29].

Table 1. Summary of simulators.				
Simulation Objective	Origin	Simulator		
Signaling Management	Commercial	Railnet		
		OpenTrack		
		Railsim		
		OnTrack		
		Rail//SYS		
	Shareware	Bahn		
Timetable Generation	Commercial	RailNet		
		OpenTrack		
		Railsim		
		Rail//SYS		
		Railsys		
		Rail Traffic Controller		
	Academic	SiSyFe		
		Topsu-RDM		
Energy Management	Commercial	OpenTrack		
		Railsim		
		OnTrack		
	Academic	Univ. Beijing Jiao-Tong		
On-line Control	Commercial	Rail Traffic Controller		
	Academic	Topsu-RDM		
		SiSyFe		
		Univ. Beijing Jiao-Tong		
		Univ. Palermo		
		Pontifical Univ. Comillas		
		Kutty		
		SimDinMetro		

implementation using Kutty. SimDinMetro [51] is able to simulate complex multi-line metro systems. It delivers as outputs passengers waiting and travel times, and occupation densities for stations and trains. SimDinMetro allows the user to manipulate: the train speed along routes, the holding time at stations, and the dispatch time for trains from terminal stations, by using control algorithms programmed in Matlab. The Matlab interface allows implementing control strategies varying from simple heuristics to complex control algorithms.

Table 2. Simulation results.			
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~	Normal Operation	Skip-stop	
Average waiting time [s]	52.4	104.0	
Average travel time [s]	714.6	543.6	
Average total time [s]	767.0	647.6	

In the on-line control field, the most powerful simulators are Rail Traffic Controller, SiSyFe, and Topsu-RDM. Rail Traffic Controller is suitable for railway systems; it has a train-movement-logic named meet-pass N-train Logic, which allows redefining routes on-line when disruptions take place. This redefinition is obtained by optimizing a cost function which includes passengers' waiting and travel times. SiSyFe includes a multi-agent control module capable of on-line reprogramming of schedules for railway systems. Finally, Topsu-RDM has an on-line delay module which aims to stabilize the system when conflicts occur.

It can be seen that simulators suitable for on-line control are mainly academic. This fact suggests that on-line control is still under development.

# VII. Application Example: Evaluating Skip-Stop Policies Using SimDinMetro

# A. Problem Definition

Skip-stop policies can be included in on-line control techniques. It refers to an operating strategy where trains traveling along the line stop only at some stations. First, the Metro line is broken down into two or more subsets, each of them formed by consecutive or non-consecutive stations. In the same way, trains are divided by assigning a routeone of the several subsets-to each train. The objective of skip-stop policies is to improve service in highly demanded stations by means of serving those stations with more than one route. One characteristic of skip-stop policies is that not extra resources are used; the improvement is achieved due to a redistribution of resources. However, skip-stop policies success is not guaranteed and simulation experiments should be conducted before implementing such a strategy.

# B. Simulator Description

The simulator is based on the event-driven model described in [31]. A deeper description of the simulator is presented in Appendix A in order to provide the reader with an overview of how the simulator works. The software has been designed to simulate the global operation of a metropolitan rail system with a proper level of detail. This makes it possible to conduct a satisfactory initial analysis of the effects of strategies like increasing the number of trains in operation, injecting trains at different points along the line, etc., when faced with different conditions of operation and demand, and for any initial system configuration. As outputs, a series of important indicators can be obtained for system planning, the most important of which are waiting times, travel times, and stations and trains occupation densities. The software also provides a description of the simulation's total development for its subsequent analysis. The simulator allows the user to fully define the fleet's management strategy, using any desired degree of information available. This can range from sticking to schedules planned beforehand, to complex control techniques, with intermediate alternatives like, for example, heuristics that only use local information from the station and/or train under analysis. For implementing the model, object-oriented programming [32] is used. The simulator allows users to simulate metro systems using as configuration inputs: the time-variant passengers' arrival rate for each station, the origin-destination matrix, the number of trains per line, and parameters defining trains' movement dynamic behavior, trains' capacity, and crowd behavior. The simulator also allows manipulating during the simulations: the train speed along routes, the holding time at stations, and the dispatch time for trains from terminal stations; using control algorithms programmed in Matlab.

The main operating interface, shown in Figure 5 for Santiago's Metro network, is the core of the simulator where the user can select the different interfaces and configuration boxes; it allows the user to save and load configuration parameters and initial conditions in the form of text files. It also allows the user to define configuration parameters: simulation's start and finish times, and the random number generating seed.

#### C. Practical Results

As a case study, simulations of the Santiago de Chile Metro, formed by 5 lines (Line 1, Line 2, Line 4, Line 4A, Line 5), 93 stations, and 7 transfer stations are presented. Santiago de Chile Metro is the most important element of the city transportation system. Metro operates from 6:00 to 23:45 hours on weekdays (8:00 to 22:30 hours on weekends), and transports 2.3 million passengers per day. Due to the importance of Metro several efforts have been made for improving operation in terms of service quality, including the use of state of the art fleet management strategies and a continuous upgrading of the measurement system. The metro measurement system is able to identify the hourly demand per station due to the use of electronic tickets and pre-paid cards. However, destination stations can not be identified directly from tickets, thus requiring the use estimations that are made based on surveys. Metro conducts surveys continuously during the year and constructs the origin-destination matrices, used on fleet management, based on these surveys.

The simulator was calibrated to simulate the entire network of the Santiago de Chile Metro using real time-



FIG 5 SimDinMetro main interface.

variant passenger arrival rates, which correspond to a normal business day with a 15 minute period. Real origindestination matrices obtained from surveys and real train and station parameters are also used in calibration.

Testing of a skip-stop policy in line 5 of the Metro system is proposed. Line 5 is a bidirectional line formed by 18 stations and covering 15.08 km. One characteristic of line 5 is an important difference between passengers' demand at stations. In fact, some stations are clearly overloaded during the rush period (7:00-8:00 AM) while others present a minimum demand during the same period.

The skip-stop policy for line 5 is designed by breaking down line 5 into two subsets, each of them containing 6 of the 12 least demanded stations, and both of them containing the 6 most demanded stations. The choice of the stations belonging to each subset is based on the work presented in [33] where the optimization problem aims to minimize travel time considering restrictions on passenger travel route and capacity of trains. In order to evaluate the usefulness of this skip-stop policy, simulations of the metro system were conducted in two scenarios: normal operation and skip-stop operation.

Table II shows the results obtained in both scenarios in terms of average waiting time, average travel time, and average total time – the total time spent by a passenger inside the metro system – for line 5. It can be seen that waiting time increases its value when a skip-stop policy is operating. On the other hand, travel time experiences a significant decrease which cause a decrease in total time. Results obtained suggest that skip-stop improves the efficiency of line 5 and thus should be implemented during rush hours.

# **VIII. Conclusions**

Simulation has become a powerful tool for testing intelligent systems in the planning stage. Simulation has been successfully applied in different fields, not only for testing operating strategies, but also for training of operators, drivers, and human personnel in general. In fact, there are several simulators available for different purposes, as can

# Appendix A. Event-Driven Simulator for Multi-Line Metro Systems

The system is characterized using three main entities: passengers, trains, and stations. The set of trains M and the set of stations S are broken down into subsets  $M_i$  and  $S_i$  respectively, each of them associated to a line  $i \in I$ . Subsets of trains and stations form a complete partition of the universe, which means that the intersection between subsets is empty and the union is the universe. For modeling purposes trains are numbered from m = 1 to m = #(M) while stations are indexed consecutively with 0 being the first station and  $\#(S_i) - 1$  the last in line *i*. Then, each station is completely defined by the pair (s, i).

Passengers arrive at the stations according to a composite Poisson distribution with time-variable rates, meaning that they arrive in variable-size groups separated by exponentially-distributed intervals of time. Arrivals are defined by the pair ( $\lambda_g$ ,  $\lambda_l$ ), where  $\lambda_g$  is the mean size of the group, and  $\lambda_l$  is the group rate, thus, the equivalent passengers' arrival rate is:  $\lambda = \lambda_g \cdot \lambda_l$ . In order to consider the variability of Metro system's demand throughout the day, which is related to rush and lull periods, the model structure allows discrete time changes in the values of arrival rates yielding a time-dependent rate  $\lambda(t)$ ; in sake of simplicity the time index is omitted in the following.

The behavior of multi-line metro systems is described using the parameters presented in Table III and the state variables defined in Table IV.

The model of the simulator is based on events related to arrivals of trains and passengers at stations (Figure 6). When an event takes place the system's state variables are updated to obtain the state of the system after the new event. What follows is a description of the internal behavior of the simulator, based on equations and flux diagrams.

Given a set of initial values for the states variables, the instant at which each train arrives at the next station is calculated assuming constant train speed over the line segments between stations:

$$tna_m(k) = ta_m(k-1) + td_m(k-1) + \frac{L_{(sa_m(k-1),i)}(k-1)}{v_i}$$



# Table 3. Model parameters

С	Train capacity.
$\lambda_{(s, i)}$	Passenger arrival rate at station $(s, i)$ .
α, β, γ, δ	Coefficients determining train boarding and deboarding times.
M(s1, <i>i</i> 1), (s2, <i>i</i> 2)	Origin-destination matrix elements indicating the proportion of passengers boarding at station (s1, i1) whose destination is station $(s2, i2)$ .
L _(s, i)	Length of line segment following station ( <i>s</i> , <i>i</i> ).
Vi	Train speed between stations of line <i>i</i> .

#### Table 4. Model state variables.

<i>t</i> ( <i>k</i> )	Instant at which event k occurs.
sa _m (k)	Index indicating station in which train $m$ is located at time of event $k$ .
$td_m(k)$	Passenger boarding and deboarding time at station in which train $m$ is located at time of step $k$ .
$ps_{(s, i), s1}(k)$	Passengers on station $(s, i)$ whose destination is station $(s1, i)$ at time of event $k$ .
$pm_{m,(s,i)}(k)$	Passengers carried by train $m$ traveling to station ( $s$ , $i$ ) at time of event $k$ .
$ta_m(k)$	Moment of arrival of train $m$ at the station in which it is located at time of event $k$ .
$tw_{(s, i)}(k)$	Total passenger wait time between events $k-1$ and $k$ in station ( <i>s</i> , <i>i</i> ).
$tt_m(k)$	Total travel time of passengers between events $k-1$ and $k$ in train $m$ .
$ta_{(s,i)}(k)$	Moment of arrival of passengers at the station $(s, i)$ at time of event $k$ .

$$\forall m \in M, i \mid m \in M_i \tag{1}$$

In the same form, the instant at which passengers arrive at a station is calculated by generating realizations of the stochastic process:

$$tna_{(s,i)}(k) = ta_{(s,i)}(k-1) + \eta \sim exp(\lambda_{(s,i)}), \ \forall s \in S_i, \forall i \in I \ (2)$$

Comparing these values determines the train  $m^*(k)$  that arrives first at a station and triggers the next train event, or the station  $(s^*, t^*)(k)$  at which passengers arrive and triggers the next passenger event, and also establishes the physical time t(k) at which this occurs:

$$t(k) = \min_{\substack{m \in M, s \in S_{b}, i \in I}} (tna_{m}(k), tna_{(s, i)}(k))$$
(3)



FIG 7 Train event process.

If there was a train event, the simulator fires the process described in Figure 7. The simulator updates the train's state variables, waiting and travel times, and simulates passengers boarding and deboarding. The number of passengers boarding the train is limited by the train's capacity, thus, the boarding process is simulated following the FIFO principle.

The final step is to apply a holding time to the train. This time has a minimum which depends on the passengers getting off the train

be seen in Table I. The analysis of simulators made in this work indicates that the user should consider the particular orientation of the intelligent system to be tested before selecting a simulator. It was stated that in order to be suitable for testing a particular application, a simulator must fulfill a list of requirements. These requirements should be used as a guide when looking for a simulation platform. Considering the analysis of available simulators made, it can be said that for mature fields like timetable generation or signaling management, simulators are mostly commercial, while for emerging fields like on-line control, simulators are mostly academic.

The practical example presented shows how to use a simulator for testing on-line control policies. Particularly, a skip-stop strategy was tested using a novel eventdriven simulator: SimDinMetro. It was demonstrated that the skip-stop strategy improves the operation of Santiago metro's line 5.

Future trends indicate that intelligent systems are now aiming to improve the operation of multi-modal transportation systems. Therefore, future simulators should be able to simulate multi-modal systems. Following this line, SimDinMetro is now being improved to simulate omnibusmetro systems.



and those who are boarding the train. It should be noted that the train's movement on the lines is restricted by safety constraints, which ensure a minimum distance between trains.

If there was a passenger event, the simulator updates waiting times and the number of passengers on the station, and then generates the destinations using a random process based on the origindestination matrix. The process is described in Figure 8.

Transfers are addressed using a random process which generates the transfer time required for passengers to change from one line to the other. Once the transfer time has expired, passengers are incorporated to the destiny line using the normal passenger event process.

## Acknowledgment

This work was funded by CONICYT ADI-32 project: "Real-Time Intelligent Control for Integrated Transit Systems".

# **About the Authors**



*Felipe Núñez* received his B.Sc. and M.Sc. degrees in Electrical Engineering from the Pontificia Universidad Católica de Chile in 2007 and 2008, respectively. Currently, he is working towards his Ph.D. degree at The University of California. He is author of

several journal and conference articles, mainly related to process control. His research interests include model predictive control, distributed control, hybrid systems, fuzzy systems, and mineral processing.



*Francisco Reyes* obtained his Bachelor of Engineering Science degree in 2010 from the Pontificia Universidad Católica de Chile's College of Engineering, where he is currently working towards his Master of Engineering Science degree. His current research

focuses on Intelligent Transportation Systems and his main research interests include automatic control and computer simulation.



**Pablo Grube** obtained his Master of Engineering Science degree in 2009 from the Pontificia Universidad Católica de Chile's College of Engineering, where he also received his electrical engineering degree in 2008. He is currently working as operating engineer

at the K+S AG Group, Germany. His main research interests include automatic control and computer simulation.



*Aldo Cipriano* received the electrical engineering and the Master degrees from the Universidad de Chile, and the Dr. Ing. degree from the Technische Universität München, in 1973, 1974, and 1981, respectively. He has been Dean of the College of Engineering and

is currently Vice Dean and Professor of Automation and Control Engineering at the College of Engineering, Pontificia Universidad Católica de Chile. Dr. Cipriano has led several R&D projects and has worked as a consultant in Chile and other countries. He is a past President of the Chilean Automatic Control Society (ACCA) and Senior Member of the IEEE.

#### References

- C. Goodman, L. Siu, and T. Ho, "A review of simulation models for railway systems," *IEE Conf. Publ.*, vol. 1998, no. CP453, pp. 80–85, 1998.
- [2] J. Törnquist, "Computer-based decision support for railway traffic scheduling and dispatching: a review of models and algorithms," in *Proc. 5th Workshop on Algorithmic Methods and Models for Optimization of Railways*, L. G. Kroon and R. H. Möhring, Eds. Dagstuhl, Germany: Internationales Begegnungs-und Forschungszentrum für Informatik (IBFI), 2006.
- [5] E. Casalicchio, E. Galli, and S. Tucci, "Macro and micro agentbased modeling and simulation of critical infrastructures," in *Proc. Complexity in Engineering*, 2010 (COMPENG'10), pp. 79–81, 22–24, 2010.
- [4] D. Helbing, A. Hennecke, V. Shvetsov, and M. Treiber, "Micro- and macro-simulation of freeway traffic," *Math. Comput. Model.*, vol. 35, no. 5–6, pp. 517–547, 2002.
- [5] N. Giambiasi and J. C. Carmona, "Generalized discrete event abstraction of continuous systems: GDEVS formalism," *Simulation Model. Prac. Theory*, vol. 14, no. 1, pp. 47–70, 2006.
- [6] J. B. Ghosh, "Asynchronous simulation of some discrete time models," in WSC'84: Proc. 16th Conf. Winter Simulation. Piscataway, NJ: IEEE Press, 1984, pp. 466–469.
- [7] D. Hong and Y. Seo, "A hybrid simulation model for manufacturing systems using event-state-operation transitions," in *IEEE Int. Conf. Industrial Engineering and Engineering Management (IEEM 2008)*, 8–11, 2008, pp. 1413–1417.
- [8] (May 2010). JB BAHN Simulation of Railway and Streetcar Networks [Online]. Available: http://www.jbss.de/hpg_eng.htm
- [9] (May 2010). Railnet [Online]. Available: http://www.azurenet.com/ RailNETSts.htm
- [10] (May 2010). Opentrack [Online]. Available: http://www.opentrack.ch

- [11] A. Nash and D. Hrlimann, "Railroad simulation using opentrack," in *Computers in Railways IX*, J. Allan, C. A. Brebbia, R. J. Hill, G. Sciutto, and S. Sone, Eds. Southampton, U.K.: WIT Press, 2004, pp. 45–54.
- [12] (May 2010). Railsim [Online]. Available: http://www.railsim.com
- [13] (May 2010). Ontrack [Online]. Available: http://www.transdecsys. com/overview.htm
- [14] (May 2010). Rail//sys [Online]. Available: http://www.ttgtransportationtechnology.com
- [15] (May 2010). Railsys [Online]. Available: http://www.rmcon.de/ englisch/einsatz.html >
- [16] A. Radtke and D. Hauptmann, "Automated planning of timetables in large railway networks using a microscopic data basis and railway simulation technique," in *Computers in Railways IX*, J. Allan, C. A. Brebbia, R. J. Hill, G. Sciutto, and S. Sone, Eds. Southampton, U.K.: WIT Press, 2004, pp. 615–625.
- [17] A. Berger, R. Hoffmann, U. Lorenz, and S. Stiller, "TOPSU-RDM a simulation platform for online railway delay management," in *Simutools '08: Proc. 1st Int. Conf. Simulation Tools and Techniques for Communications, Networks and Systems and Workshops.* Brussels, Belgium: ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2008, pp. 1–8.
- [18] M. Fontaine and D. Gauyacq, "SISYFE: a toolbox to simulate the railway network functioning for many purposes. Some cases of application," in *Proc. World Congr. Railway Research*, 2001.
- [19] (May 2010). Real Traffic Controller [Online]. Available: http://www. berkeleysimulation.com/home.html
- [20] (Sep. 2010). *Openpowernet* [Online]. Available: http://www.openpowernet.de
- [21] M. Baohua, J. Wenzheng, C. Shaokuan, and L. Jianfeng, "A computeraided multi-train simulator for rail traffic," in *Proc. IEEE Internation*al Conf. Vehicular Electronics and Safety, 2007 (ICVES), Dec. 2007, pp. 1–5.
- [22] T. Capuder, L. Lugaric, J. Brekalo-Strbic, and S. Krajcar, "Optimizing the train power system in Zagreb," in *Proc. Vehicle Power and Propul*sion Conf., 2009 (VPPC '09) IEEE, 7–10, 2009, pp. 41–45.
- [23] M. Ghantous-Mouawad, W. Schon, J. Boulanger, and G. Churchill, "Converting a conventional metro line into automated operation: identifying and managing the safety process of the traffic in migration phase," in *Proc. 1st Institution of Engineering and Technology Int. Conf. System Safety*, 6–8, 2006, pp. 10.
- [24] B. de Schutter and T. van den Boom, "Model predictive control for railway networks," in *Proc. 2001 IEEE/ASME Int. Conf. Advanced Intelligent Mechatronics*, vol. 1, 2001, pp. 105–110.
- [25] Quadstone Paramics 6.5 Modeller User Guide. Edinburgh, U.K.: Quadstone Paramics Ltd., 2006.
- [26] *PTV Vision Tutorial, VISSIM Basic Network.* Karlsruhe, Germany: PTV Planung Transport Verkehr AG, 2008.
- [27] M. Paolucci and R. Pesenti, "An object-oriented approach to discreteevent simulation applied to underground railway systems," *Simulation*, vol. 72, no. 6, pp. 372–383, 1999.
- [28] A. Fernandez, J. de Ponga, F. de Cuadra, and A. Garcia (May 2010). Multi-Train Simulator for Regulation Purposes [Online]. Available: www.iit.upcomillas.es/docs/00AFC01.pdf
- [29] A. Tomoeda, M. Komatsu, I. Y. Yoo, M. Uchida, R. Takayama, and K. Nishinari, "Real-time railway network simulator "kutty"," in ACRI '08: Proc. 8th Int. Conf. Cellular Automata for Research and Industry. Berlin, Germany: Springer-Verlag, 2008, pp. 433–440.
- [30] A. Tomoeda, M. Komatsu, I. Y. Yoo, M. Uchida, R. Takayama, R. Jiang, and K. Nishinari, "Real-time railway network simulation and alleviating congestion of crowded trains," in *Proc. ICCAS-SICE*, 2009, 18–21, 2009, pp. 4932–4936.
- [31] P. Grube, F. Núñez, and A. Cipriano, "An event-driven simulator for multi-line metro systems and its application to Santiago de Chile metropolitan rail network," *Simulation Model. Practice Theory*, to be published.
- [32] J. Sharp, Microsoft Visual C# 2008 Step by Step. Microsoft Press, 2007.
- [33] J. Adasme, "Metro de Santiago Proyecto Operación Expresa," in Proc. Bus Rapid Transit Int. Workshop, Santiago, Chile, Oct. 2008.

# ANEXO B. CÓDIGO HYSDEL EJEMPLO BENCHMARK

```
SYSTEM fourtanks {
    INTERFACE {
         PARAMETER {
             REAL hv = 0.7; /*[m]*/
             REAL S = 0.06; /*[m2]*/
             REAL a1 = 1.31e-4; /*[m2]*/
             REAL a_2 = 1.507e - 4;
             REAL a3 = 9.267e-5;
             REAL a4 = 8.816e-5;
             REAL a12 = 9.0415e-5;
             REAL qa = 0.3;
             REAL gb = 0.4;
             REAL h10 = 0.6538;
             REAL h20 = 0.6524;
             REAL h30 = 0.6594;
             REAL h40 = 0.6588; /*[m]*/
             REAL q = 9.81; /*[m/s2]*/
             REAL Ts = 5;
             REAL hmax = 1.2;
             REAL Qamax = 3.26/3600;
             REAL Qbmax = 4/3600;
             REAL k1 = 0.5 \times a1/sqrt(2 \times q \times h10) \times 2 \times q;
             REAL k^2 = 0.5 * a^2 / sqrt(2 * g * h^{20}) * 2 * g;
             REAL k12 = a12 \cdot sqrt(2 \cdot q/(hmax-hv));
             REAL k3 = 0.5 \times a3/sqrt(2 \times q \times h30) \times 2 \times q;
             REAL k4 = 0.5 * a4/sqrt(2 * q * h40) * 2 * q;
         }
         STATE {
             REAL h1 [0-h10, hmax-h10];
             REAL h2 [0-h20, hmax-h20];
             REAL h3 [0-h30, hmax-h30];
             REAL h4 [0-h40, hmax-h40];
         }
         INPUT {
             REAL Qa [-Qamax/2, Qamax/2], Qb [-Qbmax/2, Qbmax/2];
             BOOL d1, d2;
         }
         OUTPUT {
             REAL y1, y2;
         }
    } /* end interface */
    IMPLEMENTATION {
         AUX {
```

```
REAL z1, z2;
   REAL zqa1, zqb1;
   REAL zqa2, zqb2;
   BOOL d3, d4;
}
AD {
   d3 = hv - h10 \le h1;
   d4 = hv - h20 \le h2;
}
DA {
   z1 = {
       IF d3 THEN
       h1 +h10 - hv
      ELSE
        0
   };
   z_{2} = \{
       IF d4 THEN
       h2 +h20 - hv
       ELSE
       0
   };
   zqa1 = {
       IF d1 THEN
       ga*Qa
       ELSE
        0
   };
   zqb1 = {
      IF d2 THEN
       gb*Qb
      ELSE
       0
   };
   zqa2 = {
       IF d1 THEN
       (1-ga)*Qa
       ELSE
       ga*Qa
   };
   zqb2 = \{
      IF d2 THEN
       (1-gb) *Qb
       ELSE
          gb*Qb
   };
}
```

```
CONTINUOUS {
           h1 = h1 + Ts/S*(zqa1 + k4*h4 - k1*h1 + k12*(z2 - z1));
           h2 = h2 + Ts/S*(zqb1 + k3*h3 - k2*h2 - k12*(z2 - z1));
           h3 = h3 + Ts/S*(zqb2 - k3*h3);
           h4 = h4 + Ts/S*(zqa2 - k4*h4);
        }
       OUTPUT {
           y1 = h1;
           y2 = h2;
        }
       MUST {
            -h1 - h10 <= 0; h1 - hmax + h10 <= 0;
           -h2 - h20 \le 0; h2 - hmax + h20 \le 0;
           -h3 - h30 <= 0; h3 - hmax + h30 <= 0;
           -h4 - h40 \le 0; h4 - hmax + h40 \le 0;
           -Qa - Qamax/2 <= 0; Qa - Qamax/2 <= 0;
           -Qb - Qbmax/2 <= 0; Qb - Qbmax/2 <= 0; }
    } /* end implementation */
} /* end system */
```