

Simulador Dinámico de Sistemas de Tren Metropolitano

Pablo Grube, Aldo Cipriano
 Escuela de Ingeniería, Pontificia Universidad Católica de Chile
 Santiago, Chile
 E-mail: pgrube@ing.puc.cl, aciprian@ing.puc.cl

RESUMEN

Se presenta un simulador dinámico de trenes metropolitanos (metros) basado en eventos y en programación orientada a objetos, con capacidad para interactuar con programas SCILAB para fines de diseñar y evaluar sistemas de control en tiempo real. Se describe la forma de ingresar los datos y la interfaz de usuario y se demuestra cómo utilizar el simulador desarrollando un ejemplo de aplicación de control.

Palabras Clave: Simulación, Metro, Control en tiempo real.

ABSTRACT

An event-based, object-oriented dynamic simulator for metropolitan trains, with the ability to interact with SCILAB programs, is presented. It is intended to assist in the design and evaluation of real-time control systems. The means to enter the data and the user interface are described. A demonstration of how to use the simulator is also included through an example of a control application.

Keywords: Simulation, Metro, Real-time control.

1. INTRODUCCIÓN

Los trenes metropolitanos, o metros, constituyen el principal medio de transporte público en muchas ciudades del mundo. Los aumentos en los requerimientos de su demanda han motivado la expansión de este tipo de sistemas; sin embargo, aumentar la infraestructura es caro y llega un momento en que se hace insostenible. Entonces la única solución factible consiste en mejorar el sistema de gestión de los trenes, para lo cual existen dos tipos de técnicas: aquellas determinadas fuera de línea, por ejemplo pre-programando itinerarios (*schedules*) en base a información histórica, incluyendo estrategias como hacer que ciertos trenes no paren en ciertas estaciones; y técnicas que hacen uso de información del estado del sistema obtenida en línea durante la operación del mismo. Esta información puede incluir posición de los trenes, números de pasajeros en estaciones y metros, eventuales perturbaciones a la marcha normal del sistema, etc. En algunos casos, las decisiones relevantes las pueden tomar operadores, aunque esto no siempre es posible o deseable. Por ejemplo, no es práctico tomar una decisión cada vez que un tren llega a una estación; aún si no es esto lo que se desea, el operador no es capaz de considerar toda la información relevante para tomar las mejores decisiones, como sí puede hacerlo un sistema de control computacional. Un compromiso entre un sistema totalmente manejado por un operador y uno totalmente manejado por *software* son los sistemas de apoyo a la gestión, en los que el *software* le da al operador sugerencias en tiempo real de las acciones que puede tomar, pero es éste quien finalmente decide si las acepta o las ignora.

En cualquier caso, la introducción de cualquier nueva técnica de planificación o control, sea en o fuera de línea, exige realizar previamente ensayos que, al menos en una etapa inicial, no son fáciles de hacer en sistemas reales debido a los grandes costos que esto implica. Como es usual en estos casos, la solución consiste en realizar simulaciones, y de hecho existen bastantes plataformas de simulación para este tipo de sistemas. Entre ellas se puede mencionar OpenTrack, que resulta muy útil para hacer análisis de los efectos de instalar nueva infraestructura, construir horarios, probar diferentes sistemas de señales o analizar efectos de fallas (Hürlimann, 2001). Otra alternativa más enfocada al aprendizaje es BAHN, un programa *shareware* escrito por Jan Bochmann que permite diseñar y probar redes de transporte de trenes o tranvías.

Sin embargo, muy pocos simuladores ofrecen facilidades para diseñar y evaluar estrategias de control en línea. Ejemplos de estas plataformas son los micros simuladores de alta complejidad y costo Quadstone Paramics (Quadstone, 2000) y PTV Vissim (PTV, 2001), los que además no incluyen soporte específicamente para metros. Aunque estas plataformas son útiles para validar en forma definitiva los algoritmos creados, resultan poco prácticas para hacer pruebas rápidas durante el desarrollo.

En este artículo se presenta un simulador de sistemas de metro de tamaño reducido, rápido y fácil de usar, que permite diseñar y evaluar estrategias de control tanto en como fuera de

línea. En su versión actual permite controlar la velocidad de los trenes en las líneas, el tiempo de *holding* o retención de los mismos en las estaciones y el tiempo de despacho de los trenes desde las estaciones terminales de la línea.

El simulador está programado en lenguaje C# (Microsoft), el cual tiene la ventaja de ser fácil de usar, proveer un buen soporte para la programación orientada a objetos y dar facilidades para el desarrollo de interfaces gráficas. Los algoritmos de control deben ser programados en SCILAB, un popular paquete de *software* gratuito para cómputo numérico.

2. CARACTERÍSTICAS DEL SIMULADOR

2.1 Paradigmas Empleados

Para definir el modelo se empleó programación *orientada a objetos* (Stroustrup, 1997). Se trata de un paradigma de programación muy popular que consiste en definir *clases* de objetos diseñados específicamente para realizar determinada funcionalidad. Cada clase define un conjunto de *datos miembros* que representa toda la información necesaria para definir el estado de un objeto de la clase correspondiente. Además, implementa un conjunto de funciones miembro, o *métodos*, que determinan el comportamiento del objeto, y que pueden ser llamados por otras partes del programa para actuar sobre los datos miembros. Por un lado, esto permite modularizar el código, y por otro, los objetos abstractos constituyen una representación conveniente de elementos reales, como estaciones de metro, trenes, etc. que comparten una clase común pero de la que existen múltiples instancias u objetos.

Por otro lado, se optó por un esquema de simulación *orientado a eventos*, contrapuesto al paradigma de simulación continua (Hill y Yates, 1992). Esto significa que, en lugar de resolver por integración las ecuaciones dinámicas que definen el comportamiento del sistema para cada instante de tiempo, el procesamiento sólo se realiza en instantes irregulares de tiempo discretos. En cada uno de estos instantes, o *pasos*, se actualizan las variables de estado del sistema y se calcula el instante en que debe ocurrir el próximo paso. Lo que pueda ocurrir entre pasos no se toma en cuenta. Esto da la ventaja de reducir considerablemente los requisitos de capacidad de procesamiento, aunque naturalmente se pierde detalle del proceso. El uso de este paradigma se justifica pues todos los sucesos principales en esta aplicación (llegadas y salidas de trenes de estaciones, llegadas de pasajeros, etc.) ocurren en instantes bien definidos, y lo que transcurre entre dos eventos, como por ejemplo la forma exacta en que se desplazan los trenes a través de las vías, no es relevante, pues no afecta mayormente las decisiones de control.

2.2 Características de la Simulación

El *software* puede simular simultáneamente cualquier número de líneas conectadas entre sí a través de estaciones de combinación. A continuación se describen diversos aspectos del mismo:

- Las líneas son bidireccionales y pueden contener cualquier número de estaciones, incluyendo dos estaciones terminales en cada extremo. Los trenes se van desplazando por la línea y parando en las estaciones un tiempo que depende tanto de la ocupación de pasajeros (ver más adelante) como de la estrategia de control que se emplee.
- En los tramos de líneas entre estaciones, se asume que éstos se mueven con velocidad constante, de modo que el tiempo que tardan en recorrer el trayecto es simplemente:

$$t = v/d + t_A \quad (1)$$

donde v es la velocidad, d la distancia y t_A un tiempo aleatorio. Esto representa evidentemente una simplificación del caso real, en el cual los trenes tienen un perfil de velocidad a lo largo de la línea, pero si se interpreta la velocidad simplemente como “distancia recorrida” / “tiempo empleado” la aproximación debiera ser suficientemente cercana a la realidad para los propósitos de evaluar la operación del sistema a nivel global, sin entrar en detalles sobre el movimiento en cada momento de los trenes. t_A es, como ya se indicó, una variable aleatoria que distribuye normal y permite incorporar un componente estocástico al movimiento de los trenes; su media y desviación estándar deben ser fijados por el usuario y son únicos para todos los tramos de la línea.

- Aunque la velocidad de los trenes es en principio constante, puede ser cambiada debido a acciones de control, como se explica en la sección 2.3. Esto implica que puede haber trenes con velocidades distintas e incluso un mismo tren puede comenzar un tramo con cierta velocidad y terminarlo con otra. Aquí se considera, simplemente, que la velocidad cambia bruscamente de un valor a otro.
- Los trenes van numerados según su posición relativa, y en condiciones normales este orden no puede cambiarse, es decir, los trenes no pueden adelantarse entre sí. Si un tren arriba a una estación que está ocupada entra en “modo de espera” hasta que la estación se desocupe y recién entonces ingresa en ella. Algo parecido ocurre si los trenes se alcanzan unos a otros dentro de la línea.
- En los extremos de las líneas existen “colas de maniobra” en las cuales es posible almacenar un determinado número de trenes definido por el usuario. Al abandonar una estación terminal, los trenes tardan un tiempo fijo en llegar a la respectiva cola, e igualmente tardan un tiempo fijo en abandonarla y llegar a la primera estación en el sentido contrario. El momento en que se despacha el primer tren que se encuentra en la cola está determinado por el control, como se explica en la sección 2.3. Si la cola está llena, un tren que se encuentre en la estación terminal correspondiente no la abandona hasta que el primer tren de la cola la abandone; igualmente, si la primera estación en la vía contraria está ocupada, el primer tren en la cola se queda ahí hasta que se desocupe.
- Además, en los tramos de línea entre estaciones el usuario puede definir “vías laterales” paralelas a algunos tramos de línea en que los trenes se pueden mantener temporalmente estacionados. Al ser despachado un tren, el control puede definir que en lugar de funcionar normalmente, salte las estaciones y se dirija a determinada vía lateral de la línea donde se queda hasta que el control le permita abandonarla y reintegrarse a la operación normal. El mecanismo de control del tren se explica en la sección 2.3. El usuario debe definir en qué puntos del tramo de línea se encuentran la entrada y salida de la vía lateral. Al igual que los depósitos de los terminales, cada vía lateral tienen cierta capacidad y, si ésta es excedida, el siguiente tren que quiera entrar en ella se quedará parado en espera, bloqueando la vía principal en el punto en que se encuentra. Las vías laterales permiten tener a los trenes que esperan en ellas aguardando a que cierta parte de la línea se congestione más y en ese momento “soltarlos” para servir rápidamente a la parte congestionada de la línea sin tener que pasar por el resto.
- Los pasajeros llegan a las estaciones según un proceso de Poisson compuesto; es decir, llegan en grupos separados por intervalos de tiempo que distribuyen exponencial. El tamaño de cada grupo distribuye Poisson. Por lo tanto, se requieren dos parámetros para determinar la llegada de pasajeros a

cada estación: el número de grupos que llegan por unidad de tiempo y el tamaño medio de estos grupos. El valor de ambos parámetros puede ir variando a lo largo de la simulación en forma discreta si así lo desea el usuario, que puede especificar valores distintos para distintos intervalos de tiempo.

- Al arribar un tren a una estación, debe permanecer allí un tiempo mínimo (*dwell time*), el cual está dado por un modelo simple del tipo que se puede encontrar en (Lin y Wilson, 1992):

$$T = \alpha + \beta P_S + \gamma P_B + \delta P_O + T_A \quad (2)$$

donde P_S es el número de pasajeros que aborda el tren (generalmente, el número de pasajeros que hay en la estación, a menos que el tren haya llegado al tope de su capacidad), P_B es el número de pasajeros que bajan en la estación, P_O es el total de pasajeros que se encuentran tanto en la estación como en el metro, ya sea que suban/bajen o no, con el fin de considerar el efecto de congestión que produce un número excesivo de pasajeros, y T_A es un tiempo aleatorio de distribución normal que permite incorporar un componente estocástico al *dwell time* y cuyos parámetros deben ser determinados por el usuario en forma única para toda la línea. α , β , γ y δ son parámetros también definidos por el usuario. La capacidad de los trenes también es un parámetro constante para todos los trenes de la línea.

- El *dwell time* anterior no es necesariamente el tiempo que el tren permanecerá en la estación; éste puede ser mayor si el control especifica un tiempo de retención o *holding*, como se explica en la sección 2.3. Sin embargo, el tren no puede salir de la estación antes de pasado el *dwell time*.
- En cada tren los pasajeros están divididos en grupos según su estación de destino. Cada vez que un pasajero sube a un tren, se decide aleatoriamente a qué estación se dirige en base a las probabilidades dadas por una matriz origen-destino que el usuario debe especificar para la línea. Se especifica una matriz origen-destino distinta para cada sentido (vía) de la línea. Cuando un tren llega a una estación, todos los pasajeros del grupo correspondiente deben bajar.
- Al llegar un tren a una estación de combinación, se considera que una proporción fija de los pasajeros que ahí descienden hace la transferencia a la otra línea. Si el número de estos pasajeros es mayor que 5, se dividen en 5 grupos de distintos tamaños (siguiendo una curva normal). A cada grupo se le asigna un tiempo que demora en hacer la combinación, en base a los parámetros de tiempo medio y desviación estándar ingresados por el usuario: el primer grupo tarda el tiempo medio menos 2 desviaciones estándar, el segundo el tiempo medio menos 1 desviación estándar, el tercero justo el tiempo medio, etc. Además, si el número de pasajeros transferidos supera un mínimo definido por el usuario, se genera “atocamiento” y se penaliza el tiempo que demora cada grupo en hacer la combinación en una cantidad proporcional al total de pasajeros que ya han pasado. Al pasar el tiempo correspondiente a cada grupo, los pasajeros que contiene se agregan a la estación en la línea opuesta.
- El número de metros en funcionamiento y su posición inicial son especificados manualmente por el usuario, quien puede situarlos inicialmente en cualquier estación, tramo de línea, rama o terminal, indicando valores como el tiempo de llegada a la estación actual o su posición inicial dentro de un tramo de línea para permitirle al *software* decidir cómo deben comportarse al comenzar la simulación.

2.3 Control en Tiempo Real

El simulador puede operar en modo de “lazo abierto”, en el cual los trenes tienen todos una velocidad fija, se detienen un tiempo de *holding* o *stop* predeterminado en todas las estaciones y, al llegar a los terminales, son despachados a intervalos regulares desde ellos, todos éstos parámetros que pueden ser fijados por el usuario. Sin embargo, el objetivo principal del simulador es permitir al usuario escribir sus propias funciones de control que pueden, según los objetivos buscados, incluir información en tiempo real de la operación. Estas funciones deben ser escritas en SCILAB en archivos .SCI, respetando, las variables de entrada y salida definidas por el simulador. Éste accede a los servicios de SCILAB (y, a través de éstos, a la función escrita por el usuario) a través de las DLLs provistas por SCILAB.

Actualmente el sistema de control puede actuar sobre 4 tipos de “variables manipuladas”: *tiempos de holding*, *velocidades de línea*, *tiempos hasta el despacho desde los terminales del siguiente metro* y *tiempo hasta el despacho desde las vías laterales*. Las primeras dos se definen para cada tren, mientras que la tercera está asociada a las líneas y la última a las vías laterales. En el caso de controlar el despacho desde las vías laterales, se puede manipular también qué trenes serán dirigidos al momento del despacho hacia una de estas vías en lugar de operar normalmente.

Cada vez que un tren cambia de estado, es decir llega o sale de una estación, arriba a un depósito terminal o a una rama, etc., se invoca al sistema de control para que actualice las variables manipuladas. Para hacerlo, éste recaba información sobre el estado del sistema completo, lo cual implica obtener la siguiente información de cada tren: *estado* (en estación, en línea o en espera), *índice* del tren y *línea* a la que pertenece. El índice numera a los trenes ordenadamente según su posición relativa. Además, si está en una estación, se incluyen el *tiempo de llegada* a ella, el *número de pasajeros* con que llegó y el *número de los que había en la estación al momento del arribo*, así como el *tiempo de holding* pre-asignado al tren. Si el tren está en un tramo de línea, incluyendo el caso en que está esperando para entrar a una estación ocupada, se incluyen su *velocidad*, su *posición* y el *número de pasajeros* que lleva. En cuanto a las estaciones, se explicitan la *línea* a la que pertenecen, el *número de pasajeros* que hay en ellas, así como el *tiempo de la última llegada* y la *última salida de un metro* de cada estación. Además de todo esto, se incluyen el *tiempo actual* y el *índice* y *línea* a que pertenece el metro que gatilló la llamada al sistema de control. Finalmente, se pueden definir 6 parámetros opcionales para la función de control, según lo requiera el usuario.

Toda esta información se almacena en sendas matrices de SCILAB que se entregan como argumentos a la función definida por el usuario. Se espera que esta función use algún criterio de control para generar una matriz de salida con las variables manipuladas que deben cambiar de valor. Esta matriz debe tener tantas filas como controles se pretendan cambiar; en cada fila se incluye el tipo de variable que se quiere alterar (*holding*, *velocidad* o *tiempo para despacho de terminal* o *vía lateral*), la *línea* y el *índice* del tren (este último sólo si corresponde) y el nuevo valor que debe tomar la variable de control. El programa realiza inmediatamente los cambios requeridos y luego procede con la simulación.

3. INTERFAZ DE OPERACIÓN

La interfaz, que se muestra en la figura 1, permite al usuario crear gráficamente nuevas líneas, estaciones, combinaciones, etc. con el mouse. Además, haciendo clic en los distintos elementos es posible acceder a cuadros de diálogo para ingresar los parámetros correspondientes, como en la figura 2, que corresponde a los parámetros generales de una línea, incluyendo capacidad de los trenes, parámetros α , β , γ y δ del modelo de *dwell time*, matrices origen-destino y los demás que ya se mencionaron en las sección 2.2. Cuadros similares pueden invocarse para parametrizar estaciones, estaciones de combinación y vías laterales. Otros valores que pueden fijarse directamente en la ventana principal son los tiempos de inicio y fin de la simulación y la semilla del generador de números aleatorios. Dejar constante el valor de la semilla permite correr la simulación bajo las mismas condiciones (por ejemplo, los pasajeros arribarán a las estaciones de exactamente la misma manera) pero con estrategias de control distintas. Además, es posible guardar las líneas representadas para volver a usarlas en una ocasión posterior.

Una vez ingresadas las líneas, el siguiente paso consiste en ingresar las condiciones iniciales, esto es, el número y posición inicial de los trenes y la cantidad inicial de pasajeros en las distintas estaciones y trenes. Para ello, se debe hacer clic sobre un botón especial y a continuación seleccionar las estaciones, tramos de línea, ramas o terminales en que se desea agregar un tren o especificar la cantidad inicial de pasajeros, además de parámetros como el tiempo de llegada del tren a cierta estación o su posición dentro de un tramo de línea. Finalmente, se puede definir el nombre y ubicación en disco del archivo .M que contiene la función de control escrita por el usuario, cuáles variables manipuladas se desea que ésta controle (*holding*, velocidad de trenes y/o tiempos para despacho de terminales/vías laterales) así como los valores de los 6 parámetros opcionales que se le pasan.

Una vez seleccionados estos valores, se da inicio a la simulación. Terminada ésta, es posible acceder a estadísticas por línea o por estación, incluyendo:

- Tiempo de espera promedio de pasajeros (por estación o total de la línea)
- Tiempo de viaje promedio
- Ocupación promedio de las estaciones y trenes
- Frecuencia promedio de trenes en una estación
- Número total de trenes que se quedan esperando para entrar a una estación por estar ésta ocupada al llegar a ella

Las estadísticas anteriores pueden obtenerse para el total de la simulación o sus promedios a lo largo de intervalos de 15 minutos.

Además de las estadísticas, se pueden obtener 2 tipos de representación gráfica del desarrollo de la simulación. El primero es el diagrama distancia-tiempo de cada línea, un ejemplo del cual se muestra en la figura 3. Aquí, el eje X representa el tiempo y el eje Y la distancia a lo largo de la línea; cada línea representa un tren distinto a medida que se desplaza por la línea. También es posible obtener un gráfico de la cantidad de pasajeros en cada estación en función del tiempo como muestra la figura 4.

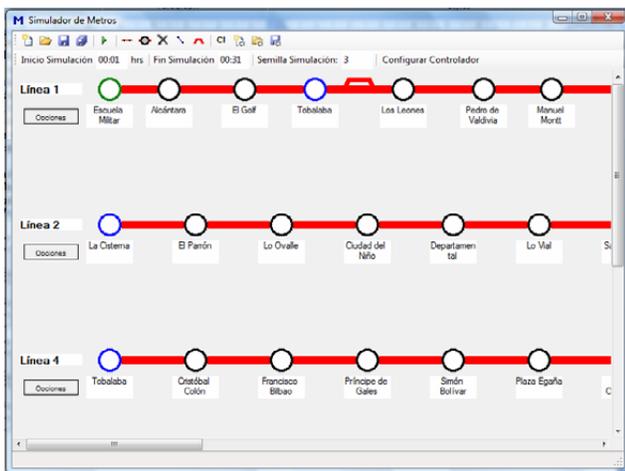


FIGURA 1: Ventana Principal de la Interfaz de la Aplicación

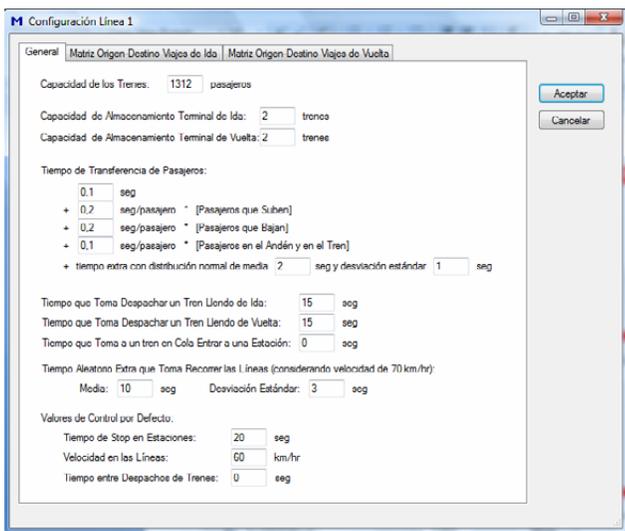


FIGURA 2: Cuadro de Configuración de los Parámetros de la Línea

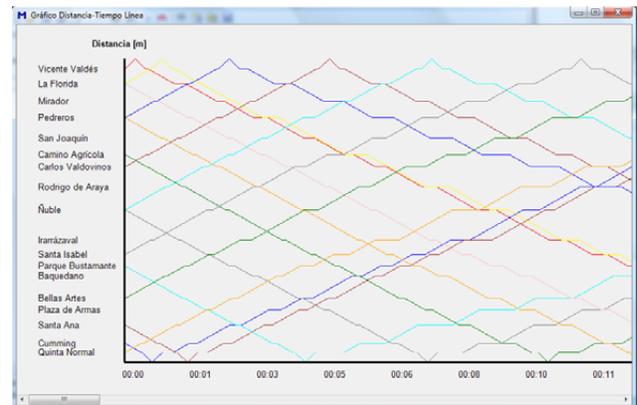


FIGURA 3: Diagrama Distancia-Tiempo de una Línea

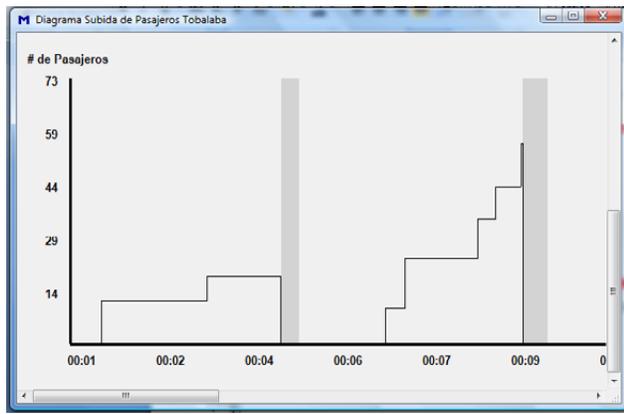


FIGURA 4: Diagrama de Subida de Pasajeros en una Estación

Finalmente, es posible almacenar un registro del desarrollo de la simulación en forma de un archivo de datos SCILAB (un archivo .DAT). Almacena una matriz para cada estación y cada tren. Cada fila de las matrices representa un instante de tiempo en que ocurrió algún suceso en el tren o estación correspondiente, como la llegada de un pasajero o el arribo a una estación. Las distintas columnas contienen los valores de las variables de estado relevantes, como total de pasajeros, posición del tren, etc.

4. EJEMPLO DE APLICACIÓN

Para mostrar cómo opera el simulador se propone un sencillo esquema de control en tiempo real basado en regularización del *headway* de salida de estaciones, es decir, el tiempo entre sucesivas salidas de trenes desde cierta estación. Se pretende que este *headway* sea siempre constante e igual a H , un parámetro de control. La estrategia es la siguiente: cada vez que un tren arriba a una estación en un instante t_{lleg} , se computa el instante de la salida del tren anterior a esa misma estación, t_{ant} . El *headway* h se calcula como sigue:

$$h = \begin{cases} 0, & t_{lleg} \geq t_{ant} + H \\ t_{ant} + H - t_{lleg}, & t_{lleg} < t_{ant} + H \end{cases} \quad (3)$$

Esto significa que el control intentará, a través del *holding* de los trenes en cada estación, mantener el *headway* de salida de ésta constante. Hay que notar que si el tren llega muy tarde, el *holding* es cero para intentar hacerlo pasar lo más rápido posible considerando que el tren puede requerir un tiempo mínimo para que suban y bajen todos los pasajeros.

El control fue probado para una línea similar a la Línea 5 del Metro de Santiago, con 18 estaciones en cada sentido, con un largo total de 14.5 km. Se consideran en total 26 trenes, inicialmente repartidos uniformemente a lo largo de la línea, los cuales se mueven entre estaciones con una velocidad promedio de 70 km/hr. Los pasajeros arriban a las estaciones con una tasa promedio de 0.25 pasajeros/seg en un sentido y de 0.16 pasajeros/seg en el otro; las desviaciones estándar de estas tasas entre estaciones son de 0.26 pasajeros/seg y 0.15 pasaj/seg, respectivamente. El modelo de *dwell time* considera los parámetros $\alpha=5$ seg, $\beta=\gamma=0.5$ seg/pasaj, $\delta=0$. El tiempo de simulación es de media hora.

Considerando el largo de la línea y el número de trenes, éstos debieran estar a aprox. 1.2 km de distancia uno de otro para estar perfectamente repartidos a lo largo de la línea. A una velocidad de 70 km/hr, esto significa un *headway* de por lo menos 1 min entre trenes sucesivos, el cual debiera aumentar debido al *dwell time* en estaciones.

En la tabla 1 se muestran los resultados de la simulación para distintos valores del parámetro de *headway* H , junto con el tiempo promedio de espera de pasajeros en estaciones y de viaje en trenes. Como comparación, se incluye también el caso sin control en que el *holding* es constante para todas las estaciones e igual a 15 seg; los resultados son claramente peores y, a diferencia del caso controlado, empeoran al aumentar el tiempo de simulación debido a que los trenes tienden a juntarse (fenómeno conocido como “*bunching*”) haciendo aumentar cada vez más los tiempos de espera: el sistema es inestable en lazo abierto. Lo anterior se puede observar en la figura 5, que muestra los diagramas distancia-tiempo para los casos con y sin control.

TABLA 1: Resultados de las Simulaciones

Parámetro H (segundos)	Tiempo promedio de espera en estaciones (seg/pasaj)	Tiempo promedio de viaje en trenes (min/pasaj)
Sin Control	78.50	10.09
70	44.43	8.78
80	37.18	8.23
90	25.48	8.36
100	24.23	8.91
110	24.78	9.37
120	24.83	9.90
130	25.19	10.34

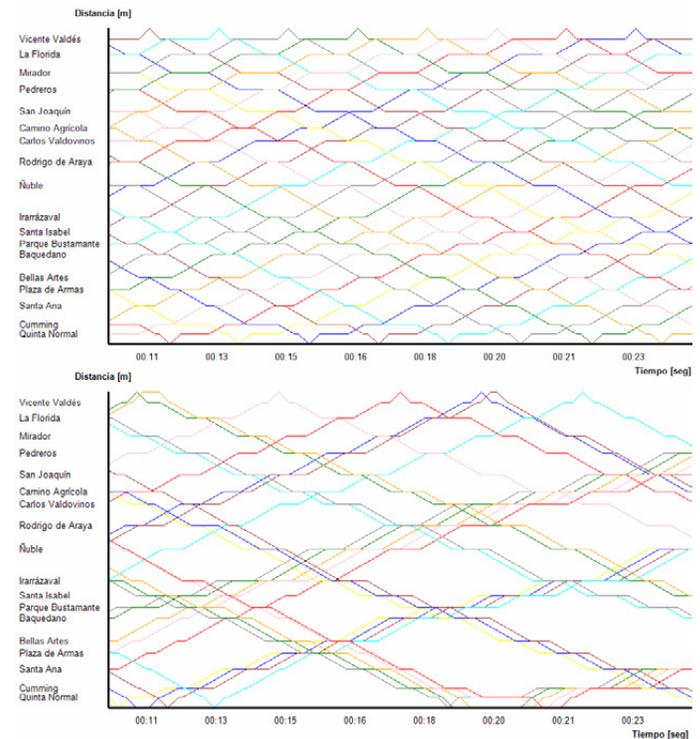


FIGURA 5: Diagramas Distancia-Tiempo para el Caso Sin Control (abajo) y Con Control (arriba)

En las figuras 6 y 7 se grafican los resultados de la tabla 1; de aquí se observa que un buen valor para el parámetro H , que permite un compromiso entre tiempos de espera y de viaje, es de $H=90$ seg.

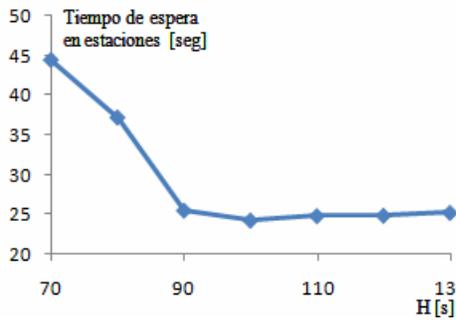


FIGURA 6: Tiempo Promedio de Espera de Pasajeros en Estaciones en Función de Parámetro H

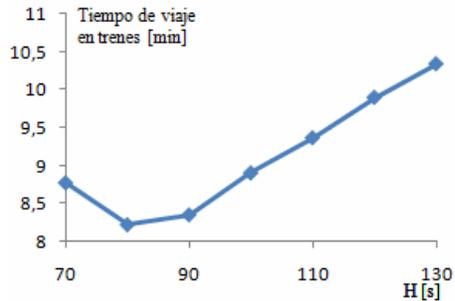


FIGURA 7: Tiempo Promedio de Viaje de Pasajeros en Trenes en Función de Parámetro H

5. CONCLUSIONES Y RECOMENDACIONES

El simulador simplifica varios detalles de la operación de un sistema de trenes metropolitanos, con el fin de obtener simulaciones rápidas y facilitar el desarrollo y evaluación de sistemas de control.

Entre las mejoras consideradas se incluye: a) incorporar un perfil de velocidades de los metros en los tramos de línea más acorde con la realidad; b) modelar en forma más precisa los tiempos de subida y bajada de pasajeros; c) incorporar funciones de cálculo de energía requerida para el movimiento de los metros para incorporarlo como un resultado adicional de la simulación; d) eliminar la restricción de que todos los pasajeros que caben tienen que terminar de subir al metro para que éste pueda cerrar sus puertas y partir; e) considerar fallas u otros eventos que limitan la operación normal; f) incluir facilidad para probar en el simulador estrategias de operación expresa o *station skipping* (haciendo que los trenes se salten algunas estaciones), así como de *short-turning*, es decir, puentes para que los trenes se puedan devolver en ciertos puntos de la línea yendo en el sentido contrario al que iban; g) combinar el simulador con estrategias de control específicas para desarrollar una herramienta útil para el apoyo en línea a las decisiones de los operadores del sistema.

Actualmente se trabaja también en la incorporación al simulador de estrategias de control más elaboradas, basadas en sistemas inteligentes y control predictivo.

AGRADECIMIENTOS

Esta investigación ha sido financiada por el proyecto Anillo de Investigación en Ciencia y Tecnología ACT-32 "Real Time Intelligent Control for Integrated Transit Systems". Agradecemos también al Departamento de Proyectos del Metro de Santiago por sus sugerencias y observaciones.

REFERENCIAS

- Hill, R.J. y Yates, T.K. (1992) Modelling Railway Block Signalling Systems Using Discrete-event Simulation. **Railroad Proceedings of the 1992 ASME/IEEE Spring Joint Conference**, 31, 1-9.
- Hürlimann, D. (2001) **Objektorientierte Modellierung von Infrastrukturelementen und Betriebsvorgängen im Eisenbahnwesen**. PhD Thesis, Eidgenössische Technische Hochschule Zürich.
- Lin, T.H. y Wilson, N.H.M. (1992) Dwell Time Relationships for Light Rail Systems. **Transportation Research Record**, 1361, 287-295.
- Microsoft (2007) **Microsoft Developer Network**, Microsoft Corporation, Washington, U.S.A.
- PTV (2001) **VISSIM User Manual-Version 3.60**, PTV Planung Transport Verkehr AG, Karlsruhe, Germany.
- Quadstone (2000) **PARAMICS v3.0 User Guide and Reference Manual**, Quadstone Ltd., Edinburgh, U.K.
- Stroustrup, B. (1997) Abstraction Mechanisms. **The C++ Programming Language, Special Edition**, 3ª Ed., Addison-Wesley, Massachusetts, U.S.A.