

La simulación con FlexSim, una fuente alternativa para la toma de decisiones en las operaciones de un sistema híbrido

Isaías **Simón-Marmolejo**
Francisca **Santana-Robles**
Rafael **Granillo-Macías**
Víctor Manuel **Piedra-Mayorga**

Universidad Autónoma del Estado de Hidalgo,
Escuela Superior de Ciudad Sahagún.
Carretera Cd. Sahagún-Otumba s/n,
Ciudad Sahagún, Hidalgo.
MÉXICO.

correo electrónico (email): isaias_simn@hotmail.com

Recibido 17-02-2012, aceptado 31-07-2012.

Resumen

La principal aportación de este documento es abordar conceptos clave en relación con el uso del software FlexSim dentro de un modelo de simulación, como una fuente alternativa de efectividad en la toma de decisiones de las operaciones de un sistema productivo. El estudio incluye una revisión de la literatura, conceptos de modelado y simulación, áreas de aplicación, la caracterización de FlexSim y la descripción para la construcción de un modelo en el software FlexSim. Este trabajo se desarrolla partiendo del entendimiento y uso del software, modelación, ejecución y análisis de las operaciones de un sistema híbrido (entidades discretas y material fluido). El modelo aquí descrito ilustra claramente la amplia precisión y capacidad de FlexSim como herramienta de mejora, además muestra lo sencilla que es su programación y lo elegante que luce su animación 3D. Este artículo busca dar una idea clara del impacto que tiene la simulación dentro de un sistema de producción y a su vez logre un entendimiento claro del uso de FlexSim.

Palabras clave: FlexSim, simulación, toma de decisiones.

Abstract

(The FlexSim Simulation, an Alternative Source for Decision-Making in the Operations of a Hybrid System)

The main contribution of this paper is to address key concepts relating to the use of FlexSim software within a simulation model,

as an alternative source of effectiveness in decision-making operations of a production system. The study includes a review of the literature, concepts of modeling and simulation application areas, the characterization of FlexSim and description for the construction of a model in FlexSim software. This work is developed based on the understanding and use of software, modeling, execution and analysis of the operations of a hybrid (discrete and fluid material entities). The model described here clearly illustrates the wide accuracy and FlexSim as improvement tool, and shows how simple and elegant programming that looks its 3D animation. This paper seeks to give a clear idea of the impact of simulation within a production system and in turn achieving a clear understanding of the use of FlexSim.

Key words: FlexSim, simulations, decision-making.

1. Introducción

Para poder comprender la realidad y la complejidad que un sistema puede conllevar, ha sido necesario construir artificialmente objetos y experimentar con ellos dinámicamente antes de interactuar con el sistema real. La simulación por computadora puede verse como el equivalente computarizado a ese tipo de experimentación. Para lo cual es necesario construir modelos que representen la realidad de tales modelos que puedan ser interpretados por un ordenador. Las operaciones, procesos o servicios de varias clases que existen en el mundo real (o de forma hipotética) son generalmente llamados sistemas y para estudiarlos de forma científica frecuentemente se hace un conjunto de suposiciones acerca de cómo éstos trabajan. Estas suposiciones, que generalmente toman la forma de relaciones matemáticas o lógicas constituyen un modelo que es usado para intentar ganar un entendimiento de cómo se comporta el sistema correspondiente. Si las relaciones que componen el modelo son bastante simples, puede ser posible usar métodos matemáticos (tales como, cálculo o teoría de probabilidad) para obtener información exacta en cuestiones de interés, a esto se le llama solución analítica. Sin embargo, la mayoría de los sistemas reales son complejos, por lo que deben ser estudiados por medio de simulación para permitir que estos modelos sean realistas y puedan ser evaluados de forma analítica. Las áreas de aplicación de la simulación son numerosas y diversas; es usual encontrar aplicaciones en ingeniería, economía, medicina, biología, ecología o ciencias sociales.

El mostrar un panorama concreto de simulación desde una perspectiva aleatoria en modelos prácticos utilizando el software FlexSim, requiere del entendimiento de los conceptos elementales que componen esta analogía. En este contexto, es necesario tener

una buena comprensión teórica de los conceptos de modelado y simulación. Además de contar con conocimientos en el desarrollo experimental, en esta investigación se describen los conceptos principales del tema en estudio.

El objetivo de este artículo es exhortar el uso de la simulación en la evaluación de un sistema productivo en un ambiente virtual, con la intención de lograr la máxima eficiencia en los sistemas de producción, minimizar costos, mejorar la calidad, reducir el lapso de tiempo entre la fabricación y la entrega de los productos al cliente, proyectar escenarios catastróficos y extremos. Así como la aplicación de alternativas educativas para la formación de personal en diferentes áreas de una organización, estudiantes, cateóricos e investigadores.

2. Revisión de la literatura

2.1. Definición de simulación

Para el desarrollo del presente trabajo resulta conveniente iniciar definiendo el término simulación. La simulación es la representación de algún proceso o sistema real a través del tiempo, ya sea hecha manualmente o en una computadora, la simulación involucra la generación de un historial artificial de un sistema y su observación para obtener inferencias relacionadas con las características operativas del sistema real [1]. Asimismo, de acuerdo con [2] la simulación por computadora se refiere a los métodos para estudiar una gran variedad de modelos de sistemas del mundo real mediante la evolución numérica usando un software diseñado para imitar las operaciones o características del sistema, a menudo en el transcurso del tiempo.

La simulación inicia con un modelo. Un modelo es una descripción física o matemática de un sistema, objeto o evento y usualmente representa un punto particular de acción en el tiempo [3]. Asimismo, los modelos son empleados para estudiar y determinar la representación de un sistema real de manera abstracta con la intención de predecir el comportamiento del mismo. Sin embargo, dado que un sistema es un conjunto complejo, integrado de elementos interconectados, el cual forma parte de otro sistema superior y se compone de sistemas de orden inferior [4], se dificulta el nivel de abstracción pues la mayoría de los sistemas reales son muy complejos para permitir evaluaciones analíticas, por lo que los sistemas deben ser estudiados mediante la simulación [5].

Un modelo de simulación representa la operación de algún proceso o sistema del mundo real, a través del tiempo, ya sea realizada manualmente o por medio de una computadora. La simulación involucra la generación de una historia artificial de un sistema y su observación para obtener inferencias relacionadas con las características operativas del sistema real [1]. Este modelo, comúnmente toma la forma de un conjunto de supuestos respecto a la operación del sistema, estos supuestos son expresados en forma de relaciones matemáticas y lógicas entre los objetos de interés del sistema. Entonces el sistema es modelado identificando las características de sus eventos y mostrando un juego de rutinas

que dan una descripción detallada del sistema en estudio. Es así como los eventos de la simulación evolucionan en el tiempo (reloj de simulación) ejecutando los eventos en orden creciente del tiempo de ocurrencia: un evento es algo que sucede en un instante de tiempo (simulado) que puede cambiar atributos, variables o acumuladores estadísticos. Asimismo, un evento discreto es un momento específico del tiempo [6]. De esta forma, el estado de los elementos de un sistema cambia continuamente o sólo en determinados estados del tiempo. El flujo del agua que entra y sale en una planta tratadora de aguas residuales y, la entrada y salida de clientes que pagan la despensa en un cajero son ejemplos de cambios continuos y discretos, respectivamente. Esto es, en el primer ejemplo el agua no muestra un espacio de tiempo entre litro y litro, por otro lado, la llegada de las personas al cajero de un supermercado muestran tiempos variados entre una y otra.

Posteriormente, una vez desarrollado, validado y verificado el modelo de simulación, éste puede ser utilizado para investigar una amplia variedad de preguntas del tipo ¿Qué pasa si...?, acerca del mundo real [2]. Una vez realizado lo anterior, se procede con la planeación de los experimentos que se harán mediante el modelo de simulación establecido. Después de haber concluido el modelo de simulación, los clientes o dueños del sistema real tendrán una herramienta que les permitirá proyectar distintos escenarios de simulación en diferentes periodos de tiempo. No obstante, la simulación también puede ser utilizada para estudiar sistemas en su etapa de diseño (antes de que dichos sistemas sean construidos). De esta manera, los modelos de simulación pueden ser utilizados tanto como un instrumento de análisis para predecir los efectos de cambios en sistemas existentes en lapsos de tiempo distintos, o bien como un instrumento de diseño para predecir el comportamiento de sistemas nuevos en proyecciones futuras.

Como opinión particular, se considera que un sistema es modelado y simulado con la finalidad de identificar sus componentes, su estructura y observar su conducta (interacción y efecto de sus elementos) en términos de sinergia, con el propósito de crear, planificar, controlar y predecir el comportamiento de distintos escenarios que contribuyan a soluciones alternativas en efectos presentes para un futuro sustentable.

2.2. La simulación a través del tiempo

La palabra simulación, en su concepto actual, se remonta hacia fines de 1940 cuando Von Neumann y Ulam definieron el término "análisis de Monte Carlo" para aplicarlo a una técnica matemática que usaban al resolver ciertos problemas de protección nuclear que eran demasiado costosos o muy complicados para resolverse experimentalmente o analíticamente [7]. Sin embargo, la simulación analógica fue muy evidente durante los años 50, la cual se definía entonces como *Simulation Programming Languages* (SPL) o bien simulación por ordenador [8].

La necesidad de resultados cada vez más precisos y rápidos estimuló el desarrollo de cientos de nuevas técnicas de simu-

lación en los últimos 60 años. Además, con el desarrollo de la computadora de gran velocidad, a principios de 1950, la simulación tomó otro significado, ya que surgió la posibilidad de experimentar con modelos matemáticos (que describen sistemas de interés) en una computadora. Su sorprendente evolución brindó la herramienta necesaria que permite la implementación de los métodos de simulación más complejos de una manera simple y eficiente [3]. El primer simulador fue *General Simulation Program* (GSP) acreditado por K.D. Tocher en conjunto con D.G. Owen en 1960 [9]. Mientras que en el periodo de 1961 a 1965 surgieron nuevos simuladores los cuales incluían declaraciones del principal lenguaje de programación llamada FORTRAN (*FORmula TRANslator System* es un lenguaje de programación adaptado especialmente al cálculo numérico y a la computación científica desarrollado por John Backus e IBM en 1954), el cual se tenía como idioma hasta entonces [10]: *General Purpose System Simulator* (GPSS) y SIMULA descritos por Wexelblatt [11], SIMSCRIPT enunciado por Marcowitz *et. al.* [12], *Control and Simulation Language* (CSL) el cual contribuía con la solución de problemas complejos dentro de las empresas industriales y comerciales [13] y algunos más enunciados en la publicación de Nance [8]. En esa época el diseñador del software utilizaba instrucciones de lenguaje y macros para especificar al sistema su lógica de operación. Por lo que el modelador tenía inicialmente que entender todo el conjunto de argumentos antes de poder hacer un seguimiento de las variables de estado del sistema. Esto implicaba la construcción de macros complicados, se requería de mucho tiempo y los costos eran excesivos. Además, los modelos que empleaban estos lenguajes de simulación no contaban con animación [14]. Posteriormente, la aparición de un nuevo lenguaje popular y de uso general llamado PASCAL desarrollado por Niklaus Wirth entre los años 1968 - 1969, estimuló el diseño posterior de paquetes de simulación basados en este lenguaje [8]: Bryant [15,16] desarrolló el lenguaje de programación SIMPAS como un lenguaje de simulación de redes el cual fue diseñado para ser altamente portátil y orientado a eventos. Otro programa sobresaliente fue el lenguaje de simulación de redes llamando INTERACTIVE, el cual permitía la utilización de símbolos gráficos además de que la construcción y la ejecución eran interactivos [17].

Por otro lado, un primer lenguaje que permitió añadir animación fue SIMAN el cual fue creado por C. Dennis Pegden en 1982. Sin embargo, para crear animación SIMAN utilizaba un segundo lenguaje independiente llamado CINEMA [12]. SIMAN y CINEMA más tarde (en el año 2000) se combinaron para crear al software Arena.

En 1986, WITNESS lanzó el primer producto con nivel de construcciones prefabricadas junto con una mejor representación visual. Este nuevo software redujo aún más la necesidad de una programación directa en el lenguaje de simulación. La nueva generación de simuladores hasta entonces era más fácil de usar, pero aún carecían de flexibilidad y la programación de código en el ordenador no estaba disponible para el usuario. A finales

de los años 80 existían en el mercado diferentes simuladores para computadora basados en Disk Operating System (DOS) los cuales fueron creados principalmente para usos comerciales, incluyendo WITNESS, ProModel, SLAM y SIMFACTORY. En esa misma época, existieron en el mercado una amplia variedad de software de simulación; no obstante, muchos de ellos eran difíciles de usar, debido limitada capacidad gráfica, lo que ocasionaba el uso máximo de la capacidad de la computadora al momento de ejecutar los modelos de simulación [3].

2.3. La nueva generación de simuladores

En la década de los 90, surgieron diferentes simuladores. En Europa surgió el Simple++, Simul8 y Taylor II (el primer simulador 3D para ejecutarse en una PC). En Estado Unidos, se crearon muchos programas de simulación de los cuales los más notables son: Extend y Simcad. En 1998 Taylor ED fue puesto en el mercado. Este último software se destacó por ser el primer software de simulación orientado a objetos en 3D con realidad virtual, el cual también operaba con la plataforma del software de Microsoft Windows. Posteriormente, en el año 2003 fue lanzado el software FlexSim, el cual resultó ser sustancialmente diferente respecto a los simuladores anteriores tanto en su lenguaje de simulación como en su arquitectura [3].

2.4. Simulación de fluidos

Dentro de algunos procesos industriales se emplea material fluido, tales como en los procesos de plantas químicas y refinerías de petróleo. Al simular este tipo de sistemas debe tomarse en cuenta que el estado de los elementos del sistema cambian continuamente en el tiempo. No obstante, la mayoría de los trabajos de investigación publicados en áreas de simulación están centrados principalmente en la simulación de eventos discretos (aquellos que cambian en determinados instantes del tiempo). Además, no todos los sistemas simulados son necesariamente continuos o discretos, sino una combinación de ambos, a lo que algunos autores llaman como modelos híbridos o de cambio continuo-discreto [2,5].

Algunos paquetes de simulación que tienen la capacidad de controlar tanto eventos discretos como continuos en la misma aplicación son: FlexSim, Extend, Arena, y WITNESS [3].

3. Simulación con FlexSim

3.1. Descripción del software

El software FlexSim fue desarrollado por Bill Nordgren, Cliff King, Roger Hullinger, Eamonn Lavery y Anthony Johnson. FlexSim permite modelar y entender con precisión los problemas básicos de un sistema sin la necesidad de programaciones complicadas, esto debido a que ofrece una forma sencilla al desarrollar el modelo de simulación. Se enlistan algunas razones por las cuales FlexSim es una buena alternativa como herramienta en simulación:

- Su amplia sección de preconstruidos permiten abordar situaciones mucho más complejas sin tener que escribir código de software.

- El software se orienta a objetos lo que admite una mayor visualización del flujo de producción.
- Todo el proyecto se desarrolla en un ambiente tridimensional (3D), además de permitir importar infinidad de objetos de distintos paquetes de diseño, incluyendo AutoCAD, ProE, Solid Works, Catia, 3D Studio, AC3D, Rivot, Google Sketch-Up, etc.
- Otra razón importante es que no sólo se pueden simular sistemas discretos, sino que también se admite la simulación de fluidos o modelos combinados continuo-discreto.
- La generación de distintos escenarios y condiciones variadas son fáciles de programar.
- Las distribuciones de probabilidad se pueden representar con gran precisión en lugar de valores promedio para representar fielmente la realidad.
- Las gráficas, los reportes y todo lo que se refiere a los estadísticos se puede revisar a detalle.

3.2. Aplicaciones de FlexSim

FlexSim ha contribuido con aplicaciones de clase mundial en temas de salud, sistemas de logística tales como operaciones de contenedores en puertos, simulaciones distribuidas en varios equipos dentro de una empresa manufacturera, en la minería, en centros aeroespaciales e incluso se ha adaptado a la industria del servicio (hoteles, hospitales, supermercados, o muchas otras industrias) para simular la administración y operación de los recursos humanos.

Asimismo se ha demostrado en diferentes casos de diversos corporativos alrededor del mundo (Volkswagen, USAA, EATON, Coca-Cola, DAIMLER, Foth, DHL, Disney, MICHELIN, MATTEL, QUALCOMM, VALE, CATERPILLAR, DISCOVER, Pfizer, U.S. ARMY, U.S. AIR FORCE, DU PONT, Gillette, ABB, FREIGHTLINER, CARRIER, ORACLE, Tetra Pak, IBM, NASA, FedEx, AVON, Whirlpool, ALCAN, Remington, BAKER HUGHES, etc.) que FlexSim es una herramienta clave para mejorar los resultados al dar respuesta acertadas a los problemas planteados [18].

4. Uso y manejo del software FlexSim

En la presente sección, para poder describir el uso y manejo del software se desarrollará un caso de estudio sencillo de un modelo híbrido.

En muchos modelos se ha ilustrado cómo manipular material que se encuentra en cantidades discretas (cajas, esferas, piezas, etc.), pero dentro de muchos sistemas reales los materiales no necesariamente son piezas discretas sino fluidos. Un fluido o líquido puede ser casi cualquier cosa que se mida por peso o volumen (leche, agua, arroz, azúcar, etc.). En este apartado se presentan los conceptos fundamentales para construir un modelo básico de fluidos en FlexSim. La construcción de un modelo con estas características requiere de mucha atención, pues a pesar de ser relativamente muy sencillo, existen detalles que no se deben pasar por alto.

4.1. Descripción del caso de estudio

Una plata productora de lácteos elabora un postre llamado DasaFrut en unas de sus líneas de producción. La preparación del postre requiere de dos ingredientes: por un lado utiliza leche pasteurizada contenida en cubetas de 25 litros cada una y por otro lado, toma el preparado de fruta ya estabilizada en cubetas de cinco litros. La leche pasteurizada es tomada del Almacén1 y enviada hasta el Tanque FluidTank1. Para realizar dicho envío, es necesario que un operario tome las cubetas del Almacén1 y las lleve hasta un ItemToFluid1 para que la leche sea vertida a través de una tubería al Tanque FluidTank1. El preparado de fruta estabilizado es enviado hacia un segundo Tanque FluidTank2, donde otro operario tome las cubetas del Almacén2 y las lleve hasta un segundo ItemToFluid2 para que la fruta sea vertida a través de una tubería al Tanque FluidTank2. Una vez contenidos los ingredientes en ambos tanques respectivamente, la leche y la fruta se conduce hacia el área de mezclado por tuberías en proporciones variadas. Después de que el producto es mezclado pasa a un proceso de fermentación y finalmente se entrega a un FluidToItem para crear porciones de 500 ml. Las porciones salientes de la máquina FluidToItem ahora son llamadas Postre y necesitan de un recurso Combiner para ser envasadas una en cada frasco (véase figuras 1 y 2).

4.2. Descripción de los objetos del modelo

- Tanto los flowitems (cubetas de leche pasteurizada) como los flowitems (cubetas de preparado de fruta) llegan a una tasa exponencial (0, 10, 0) segundos.
- El contenido máximo del ItemToFluid1 es de 100 litros y del ItemToFluid2 es de 20 litros.
- Las unidades de leche pasteurizada por cada unidad discreta es de: 25 litros por cada flowitem (cubetas de leche pasteurizada) que entre al ItemToFluid1.
- Las unidades de preparado de fruta por cada unidad discreta es de: cinco litros por cada flowitem (cubetas de preparado de fruta) que entre al ItemToFluid2.
- La tubería FluidPipe1 que conduce al tanque FluidTank1 tiene una capacidad de contenido máximo de 12 litros y la velocidad de transferencia del ItemToFluid1 al FluidTank1 es de 10 litros por segundo.
- La tubería FluidPipe2 que conduce al tanque FluidTank2 tiene una capacidad de contenido máximo de 20 litros y la velocidad de transferencia del ItemToFluid2 al FluidTank2 es de 2 litros por segundo.
- La capacidad de tanque FluidTank1 es de 100 litros y la de FluidTank2 de 20. Para ambos tanques la marca de tanque inferior es de 1 litro y la marca de tanque superior es de 85 y 18 litros, respectivamente.
- Las dos tuberías (FluidPipe3 y FluidPipe4) que conduce al FluidMixer tienen una capacidad de contenido máximo de cinco litros y la velocidad de transferencia de los tanques FluidTank1 y FluidTank2 al FluidMixer es de 10 y dos litros por segundo, respectivamente.

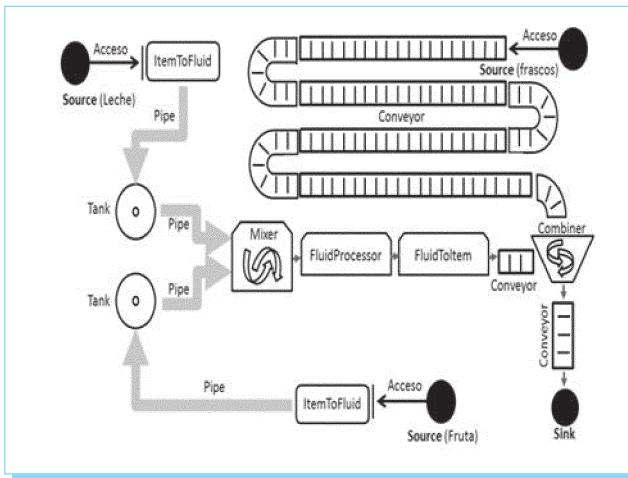


Fig. 1. Diagrama de flujo de producción del modelo de fluidos.

- Descripción para el mezclado:
 - Mixer Steps:
 - Paso 1: Entrada 1, 5 segundos de retraso
 - Paso 2: Entrada 2, no hay tiempo de retardo
 - Mixer Recipe
 - Leche pasteurizada: 10 litros, paso 1
 - Preparado de fruta: 3 litros, paso 2
- El contenido máximo y la velocidad del FluidProcessor no es relevante, manténgase como está definido.
- El contenido máximo del FluidToItem es de 0.5 litros y las unidades de líquido por unidad discreta entregadas de 0.5 litros por flowitem, estos nuevos flowitems se llaman Postre.
- Los flowitems (frascos) llegan al sistema cada 1.5 segundos de un tercer almacén y son enviados para una banda transportadora hasta el Combiner.
- Al Combiner llegan tanto frascos como flowitems postres y son envasados a la velocidad de 3 600 frascos por hora, es decir, un frasco por segundo.

4.3. Descripción de términos para el modelado de fluidos

Para poder iniciar con la construcción del modelo, es necesario entender algunos términos que se encuentran en las librerías que ofrece el software FlexSim y que son necesarios para el desarrollo del modelo:

– Fluid. En FlexSim un fluido es cualquier materia que no es fácil o eficazmente modelado como un flowitem (un tipo de entidad) discreto. Generalmente es un material que se mide por peso o volumen. Además, los materiales fluidos también pueden representar objetos que son muy numerosos y que como flowitems no son prácticos de manejar, por ejemplo miles de tomates formados en una línea de lavado pueden ser modelados como materia fluido sin la necesidad de utilizar un flowitem para cada tomate que sobrecargue y disminuya la velocidad del sistema.

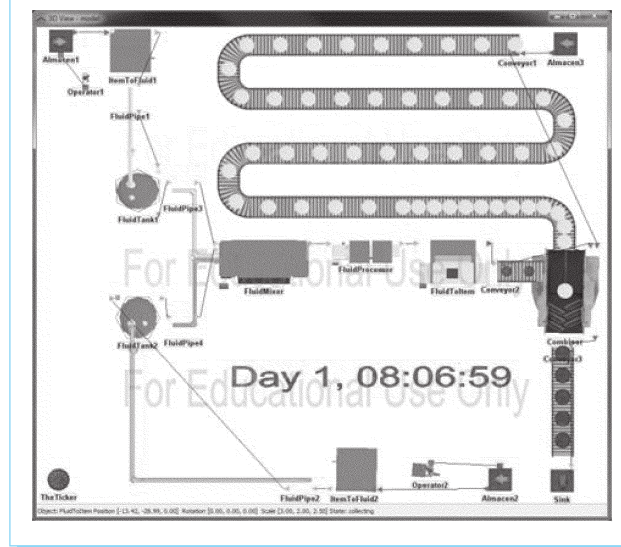


Fig. 2. Imagen del modelo final en ejecución.

- Fluid Objects. Existen 11 objetos que están diseñados para manejar material fluido. Nueve de éstos no pueden interactuar con los objetos discretos de FlexSim, pero dos de ellos (ItemToFluid y FluidToItem) están diseñados para funcionar como interfaz entre los objetos de fluidos y los objetos discretos, éstos permiten al modelador convertir flowitems en el líquido y el líquido en flowitems.
- Tick. Los Fluid Objects envían y reciben material en intervalos fijos. Estos intervalos se llaman "Ticks". Al final de cada Tick, los Fluid Objects calculan la cantidad de material que enviaron y recibieron durante ese periodo de tiempo.
- Tick Time. Duración de cada Tick. El modelador puede establecer este valor a un valor que sea apropiado para su modelo. Entre más corto sea el Tick Time el modelo será más preciso, pero también puede hacer que éste sea más lento. Un valor más largo permitirá una mayor rapidez en la ejecución del modelo, pero el costo será una pérdida de precisión. Entonces el equilibrio de precisión y velocidad está sujeto a las necesidades del modelador.
- Rate. Tasa de velocidad máxima a la que el material entra o sale de un objeto. En general, los objetos fluidos tienen tanto una velocidad de entrada como una velocidad de salida, la cual está separada una de otra. En algunos objetos, la velocidad a la que el material entra afectará la velocidad a la que sale. Para estos objetos, al modelador no se le da la oportunidad de editar la velocidad de salida. La velocidad real a la que el material entra o sale se basa en varios factores: la velocidad de salida del objeto con un flujo a contracorriente o corriente arriba, la velocidad de entrada del objeto corriente abajo, la cantidad de material disponible para enviar y la cantidad de espacio disponible en el objeto corriente abajo.
- Object Rate. Ésta es la tasa o velocidad máxima a la que el material puede entrar o salir de un objeto a través de todos los

puertos de entrada o salida juntos. Los objetos tienen típicamente una velocidad diferente para los puertos de entrada y los puertos de salida. Si al final de cualquier Tick, el objeto calcula que la cantidad de material enviada o recibida ha alcanzado la velocidad máxima del objeto, el envío y recepción de material será suspendido por el Tick, incluso si hay puertos que aún no envían o reciben material.

- Port Rate. Es la tasa de velocidad máxima a la que el material puede entrar o salir de cualquier puerto en el objeto. Los objetos normalmente tienen diferentes velocidades en sus puertos de entrada y puertos de salida. Este valor únicamente se aplica a todos los puertos de entrada o salida. Este parámetro no puede ser alterado para afectar puertos individuales.
- Port Scale Factor: Este número es empleado para cambiar la velocidad de un puerto pero de manera individual. El valor de cada puerto es multiplicado por la velocidad máxima del puerto para encontrar la velocidad máxima actual por cada puerto.

4.4. Construcción del modelo

Para la construcción del modelo se inicia creando una nueva hoja de trabajo (New Model) en FlexSim. Posteriormente, se selecciona de las librerías los recursos necesarios y se arrastran al área de trabajo (véase figura 2). Los objetos necesarios para este caso de estudio son:

- 3 Sources de nombre Almacen1, Almacen2 y Almacen3 (Discrete Objects).
- 2 Operators de nombre Operator1 y Operator2 (Discrete Objects).
- 2 ItemToFluids de nombre ItemToFluid1 y ItemToFluid2 (Fluid Objects).
- 2 FluidPipes de nombre FluidPipe1 y FluidPipe2 (Fluid Objects).
- 2 FluidTanks de nombre FluidTank1 y FluidTank2 (Fluid Objects).
- 2 FluidPipes más de nombre FluidPipe3 y FluidPipe4 (Fluid Objects).
- 1 FluidMixer de nombre FluidMixer (Fluid Objects).
- 1 FluidProcessor de nombre FluidProcessor (Fluid Objects).
- 1 FluidToItem de nombre FluidToItem (Fluid Objects).
- 3 Conveyors de nombre Conveyor1, Conveyor2 y Conveyor3 (Discrete Objects).
- 1 Combiner para envasar el postre en los frascos (Discrete Objects).
- 1 Sink de nombre Sink (Discrete Objects).
- 1 VisualTool para colocar un texto que indique el avance del tiempo (Discrete Objects).

Observar que cuando arrastra el primer objeto de fluidos al área de trabajo, automáticamente un Ticker es creado en la posición (0,0) del área de trabajo. Este mismo, puede moverse de un punto a otro dentro el área de trabajo, pero por ningún motivo lo elimina pues es necesario para trabajar con los objetos de fluidos.

4.5. Conexión de objetos

Después de arrastrar los objetos al área de trabajo, es necesario conectarlos. La conexión de objetos fluidos se hace de la misma forma con la que conecta objetos discretos: presionando la tecla A y dando clic en los objetos a conectar se crea una conexión de entrada/salida y la tecla S crea una conexión de puerto central. Recuerde que los objetos deben ser conectados de acuerdo con el diagrama de flujo de producción mostrado en la figura 1. Para este caso de estudio, las conexiones son las que se enlistan a continuación:

- del Almacen1 al ItemToFluid1
- del ItemToFluid1 al FluidPipe1
- del FluidPipe1 al FluidTank1
- del FluidTank1 al FluidPipe3
- del FluidPipe3 al FluidMixer
- del FluidMixer al FluidProcessor
- del FluidProcessor al FluidToItem
- del FluidToItem al Conveyor2
- del Conveyor2 al Combiner donde se espera la llegada de frascos
- del Combiner al Conveyor3 y al Sink
- después se hace una línea paralela del Almacen2 al FluidMixer
- conectar también del Almacen3 al Conveyor1 y del Conveyor1 al Combiner
- el Almacen1 y el Almacen2 llaman a un operador distinto para transportar los flowitem hasta el ItemToFluid, por lo que los Operators deberán tener una conexión de puerto central tanto del Almacen1 como del Almacen2, respectivamente.

4.6. Configuración de los objetos

Configuración de los Sources. Dado que el valor predeterminado del tiempo de llegadas del Almacen1 y del Almacen2 es funcional para este modelo, debe dejarse tal y como está. Únicamente se indica a cada uno de los dos Sources, la instrucción que permite llamar a cada Operator según corresponda para que se lleve los Flowitems creados hasta los dos objetos ItemToFluids. Después se da doble clic en el Almacen1 para abrir la ventana propiedades de ese objeto y se da clic en la pestaña Flow marcando Use Transport. Hacer clic en Aceptar para aplicar los cambios y cerrar la ventana de propiedades. Repetir este paso para el Almacen2. Respecto al Almacen3 sólo es necesario que se elija de la lista de selección de opciones llamada FlowItem Class la opción de Cylinder para representar a los frascos, finalmente en este mismo Source se debe configurar el campo Inter-Arrivaltime a 1.5 segundos. Nota, si desea cambiar el color de los Flowitems puede usar el comando Set Color de la opción OnExit de la pestaña Triggers (véase figura 3).

Configuración del color de los objetos. Siempre que un objeto es creado se personaliza con diferentes colores dependiendo de su clase, pero a menudo es útil asignarles un color correspondiente con el material que se esté procesando. Dado que en este modelo hay dos

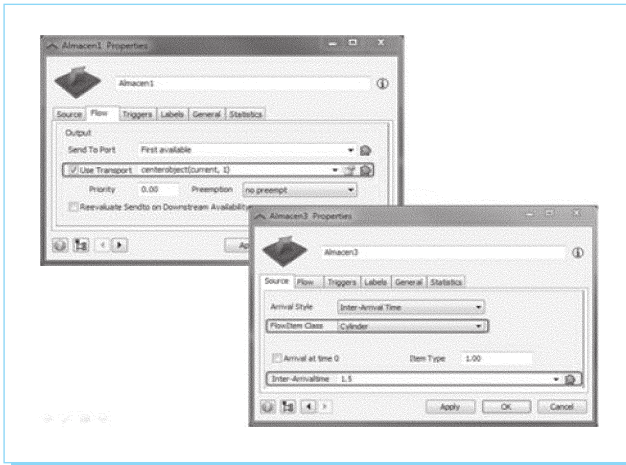


Fig. 3. Ventana de propiedades del Almacen1 y Almacen2.

líneas de procesamiento compuestas cada una por un ItemToFluid, FluidPipe, FluidTank y un FluidPipe es necesario identificar cada línea con un color distinto. Abrir la ventana de propiedades de cada uno de los cuatro objetos y cambiar el color a un blanco para la línea de leche y a los objetos de la línea de fruta asignarles un verde. Esto hará que los objetos cambien de color cuando estén en funcionamiento, siempre y cuando se tenga producto en ellos. Para ejemplificar lo anterior a continuación se muestran tres pasos:

1. Hacer doble clic en ItemToFluid1 para abrir la ventana de propiedades de ese objeto y dar clic en la pestaña general.
2. Hacer clic en el botón situado junto a un costado del campo de Color. Una tabla de colores aparecerá. Seleccionar el color blanco con un clic.
3. Hacer clic en Apply y Ok para aceptar los cambios y cerrar la ventana de propiedades.

Repetir estos pasos para el resto de los objetos mencionados anteriormente. Los objetos de una línea de procesamiento deberán ser de color blanco y los de la otra línea de color verde (véase figura 4).

Configuración de los ItemToFluids. Para crear la proporción correcta de material fluido por cada Flowitem que entre al objeto (ItemToFluids), es necesario configurar los objetos ItemToFluids:

1. Hacer doble clic en el ItemToFluid1 para abrir su ventana de propiedades. En la pestaña ItemToFluid cambiar el valor de la opción Discrete Units per Flowitem por 25. Esto indicará al ItemToFluid1 que deberá crear 25 litros de leche por cada Flowitem (cubetas de leche pasteurizada) que entre.
2. Cambiar el Maximum Object Rate y la Maximum Port Rate por un valor de 2.
3. En la opción Maximum Content asignar un valor de 100.
4. Hacer clic en Apply y Ok para aplicar los cambios y cerrar la ventana de propiedades (véase figura 5).

Repetir estos pasos para el ItemToFluid2, además cambiar su

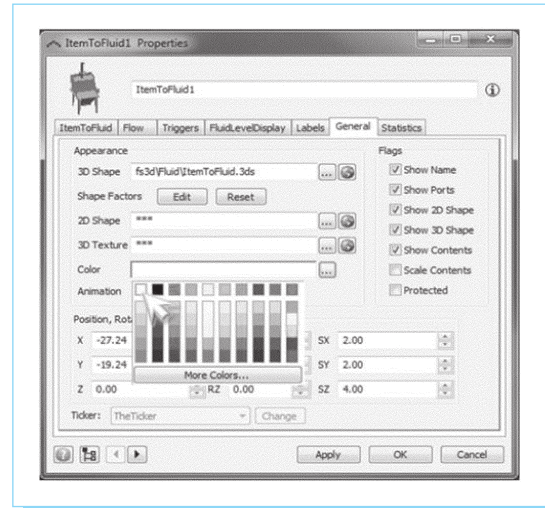


Fig. 4. Ventana de propiedades del ItemToFluid1.

capacidad máxima a 20 litros e indicar la necesidad de crear cinco litros de fruta por cada Flowitem (cubetas de preparado de fruta) que entre al ItemToFluid2.

Configuración de los FluidPipes. Las tuberías que conducen el material fluido de un objeto a otro son llamadas FluidPipes y necesitan ser configuradas. Una tubería no permite al modelador especificar la tasa o velocidad a la que entran o sale el material fluido. La velocidad de salida se basa en la misma velocidad con la que el material fluido fue recibido, para configurar se deben realizar los siguientes pasos:

1. Hacer doble clic FluidPipe1 para abrir su ventana de propiedades.

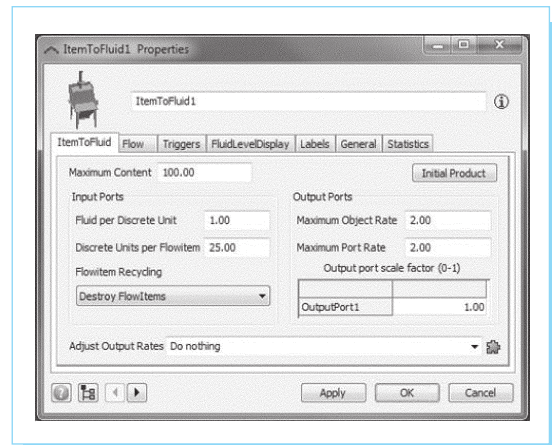


Fig. 5. Configuración de proporciones, velocidad y capacidad del objeto ItemToFluid1.

2. En la pestaña Pipe, cambiar la Maximum Flow Rate a 12 y establecer un Maximum Content de 10 litros. Esto asegurará que el material fluido (leche) pase a una velocidad de 10 litros por segundo.
3. Posteriormente, para cambiar las dimensiones, añadir secciones y definir la dirección de la tubería, hacer clic en la pestaña Layout. La tubería dentro del modelo deberá iniciar desde ItemToFluid1 y terminar hasta un punto cercano al FluidTank1. Manipular estos valores como se crea conveniente. Cambiar el diseño o estructura no afecta el comportamiento de las tuberías.
4. Hacer clic en el botón Apply y Ok para aplicar los cambios y cerrar la ventana de propiedades.

Repetir este paso para la segunda tubería FluidPipe2, pero con un contenido máximo de 20 y una velocidad de flujo de 2 litros por segundo. No olvidar modificar el diseño para representar el envío de material fluido del ItemToFluid2 hasta el FluidTank2 (véase figura 6).

Configuración de los FluidTanks. El FluidPipe1 ahora tienen una tasa de salida máxima de 10 litros por segundo, pero el FluidTank1 tiene una tasa máxima de entrada 1. Si estos valores se dejan como están, la tasa de los FluidTank1 se utilizará durante la ejecución del modelo (ya que se toma el menor de los valores) y el FluidPipe no será capaz de enviar material corriente abajo tan rápido como se ha especificado. Así que, la tasa en los FluidTanks necesita ser cambiada. Cualquier FluidTank permite al modelador establecer tres niveles que actúan como disparadores para liberar material cuando el contenido del FluidTank alcanza un nivel deseado. Estos valores se denominan Marks (marcas) y pueden editarse en la pestaña de Marks contenida en la ventana de propiedades del objeto. En este modelo, los tanques deben mantener sus puertos de salida cerrados hasta que hayan recibido una cierta cantidad de material. Al alcanzar la marca especificada, los puertos de salida se abrirán y se quedarán abiertos hasta que el tanque se vacíe o hasta que llegue a una marca inferior definida. Los tanques

siempre van a mantener sus puertos de entrada abierta. A continuación se mencionan los pasos a seguir para la configuración de los FluidTanks:

1. Hacer doble clic en FluidTank1 y en la pestaña Tank, cambiar el Maximum Object Rate y la Maximum Port Rate por el valor de 10, tanto en la entrada como en la salida.
2. Dentro de la opción Maximum Content asignar un valor de 100.
3. Hacer clic en la ficha Marks y cambiar el valor de Low Mark (marca inferior) por el de 1, y el de High Mark (marca superior) a 85. La Mid Mark (marca media) se deja en 0. Si una marca tiene el valor 0, el detonante de esa marca nunca se disparará (véase figura 7).
4. En la tabla de Marks, dar clic en la cruz verde para agregar la activación de la función Passing Low Mark y seleccionar la opción Open or Close Ports. Dentro de esta opción, dar doble clic sobre el texto azul y una lista de opciones disponibles aparecerá. Configurar la instrucción de tal forma que esta sea semejante a lo que se muestra en la imagen superior de la Figura 8.
5. Agregar además la opción Open or Close Ports para ejecutar una función Passing High Mark semejante a los parámetros mostrados en la imagen inferior de la misma figura 8.
6. Para finalizar dar clic en Apply y Ok para aplicar los cambios y cerrar la ventana de propiedades.

Repetir los pasos anteriores para el FluidTank2, pero indicar una Capacidad Máxima de 20. Además, la Tasa o Velocidad Máxima del Objeto y Puerto deberá ser de 2. No olvidar configurar una Marca Inferior de 1 y una Superior de 18 con sus respectivas instrucciones.

Orientación y tamaño de las tuberías restantes. Para seguir avanzando en el proceso de diseño del modelo, es necesario ahora orientar los siguientes dos tubos (FluidPipe3 y FluidPipe4) hacia el mezclador llamado Mixer. Recordar que el comportamiento de los tubos no se afecta si se cambia su estructura. Utilizar la pestaña de diseño Layout para configurar las tuberías de tal modo que se

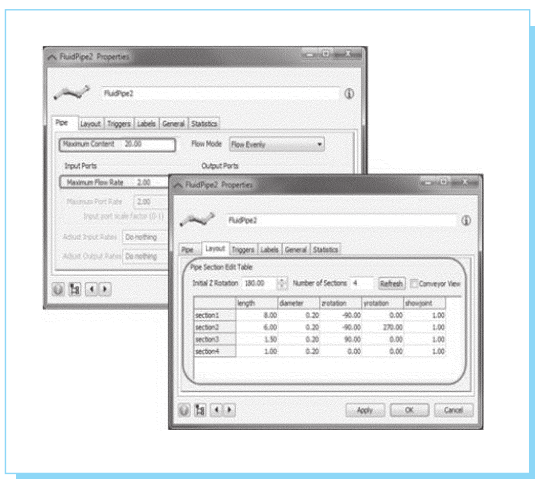


Fig. 6. Configuración de la tubería FluidPipe2.

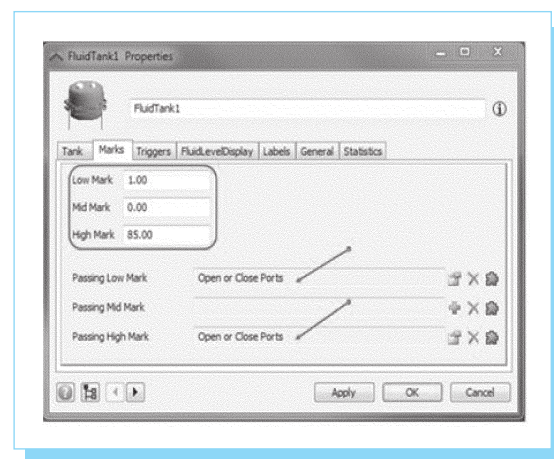


Fig. 7. Configuración del tanque FluidTank1.

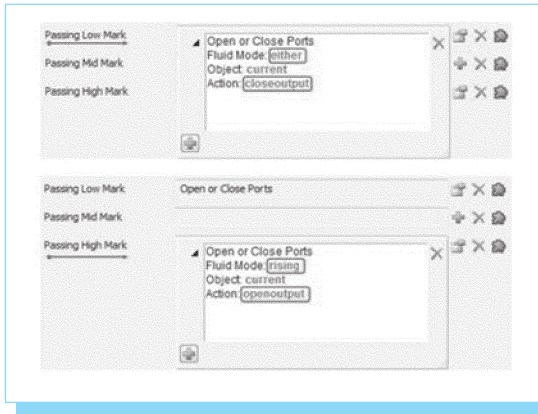


Fig. 8. Configuración de las operaciones Mark del tanque FluidTank1.

visualicen como se desea:

1. Hacer doble clic en FluidPipe3 para abrir su ventana de propiedades. En la pestaña Pipe, cambiar el Contenido Máximo a 5 y la Tasa de Velocidad Máxima de Flujo ajustarla a 10. Con esto se asegura que el material saliente de los tanques toma sólo un poco de tiempo para llegar hasta el mezclador.
 2. Modificar el diseño de la tubería de tal forma que se visualice una conexión entre el FluidTank1 y el FluidMixer.
- Repetir estos paso para el FluidPipe4, pero con un Maximum Content de 5 y un Maximum Flow Rate de 2.

Configuración de la entrada y mezcla de material fluido dentro del FluidMixer. Un mezclador de fluidos es capaz procesar dos o más materiales para combinarlos en uno nuevo. Esto se puede hacer al manipular la tabla contenida en la pestaña Steps ubicada dentro de la ventana de propiedades del FluidMixer. La tabla Mixer Steps, se utiliza para definir una serie de pasos que el mezclador realiza en cada lote que procesa. En este modelo, es necesario programar dos pasos dentro de la tabla Mixer Steps. El mezclador deberá jalar 10 litros del primer material (leche pasteurizada) por el puerto de entrada 1 durante el Paso 1. Después, se deberán jalar tres litros del segundo material (preparado de fruta) desde el puerto de entrada 2 durante el paso 2.

1. Hacer doble clic en FluidMixer para abrir su ventana de propiedades y hacer clic en la pestaña Steps.
2. Dentro del campo Number of Steps asignar un 2 para agregar dos pasos en la tabla y dar un clic en Update para actualizar la tabla. La descripción de los pasos no es importante, estos pueden ser llamados se considere conveniente. El tiempo de retardo (*delay*) de cada paso es ejecutado después de que todo el material de este paso es recibido, y antes de que el mezclador comience a recibir el material del siguiente paso. Ajustar el tiempo de retardo para el Paso 1 a 5 segundos y el tiempo de retardo para el Paso 2 a 0 (véase figura 9).
3. Posteriormente, dentro de la tabla Mixer Recipe, cambiar el número de ingredientes a 2 y dar un clic en Update para actualizar la

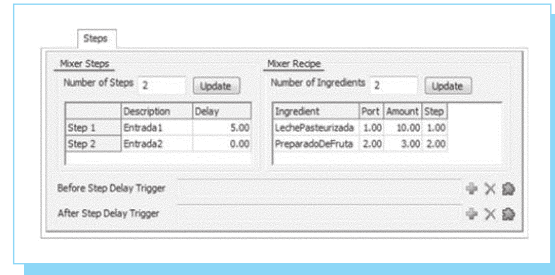


Fig. 9. Configuración de la entrada y mezcla de ingredientes dentro del FluidMixer.

tabla. Esta acción ha creado dos listas de ingredientes. Una vez más, la descripción de cada ingrediente puede ser definida como se desee, esto no afecta al modelo. En este caso, los llamaremos LechePasteurizada y PreparadoDeFruta.

Dentro de LechePasteurizada, ajuste el Port a 1, Amount a 10, y Step a 1.

Dentro de PreparadoDeFruta, ajuste el Port a 2, Amount a 3, y Step a 2.

La pestaña FluidLevelDisplay es una herramienta útil para visualizar un indicador de nivel durante la ejecución del modelo. En dicho indicador será mostrada la cantidad recibida de cada material correspondiente con la tabla Mixer Recipe en cualquier punto del tiempo simulado. Esta herramienta se encuentra oculta, pero se puede visualizar si hace clic en la pestaña FluidLevelDisplay y asigna el valor de 1 al campo "Y". De la misma forma puede manipular el resto de los campos hasta conseguir un efecto más realista (véase figura 10).

Configuración del FluidProcessor. Los valores con los cuales opera el Procesador de Fluido son adecuados para este modelo, por lo que no es necesario modificarlos. El material fluido es recibido por el puerto de entrada 1 y procesado por una cierta cantidad de tiempo para salir después por el puerto de salida 2. La cantidad de tiempo invertida en el proceso, se basa en el Maximum Content y en la Maximum Output Rate que el modelador define dentro de la

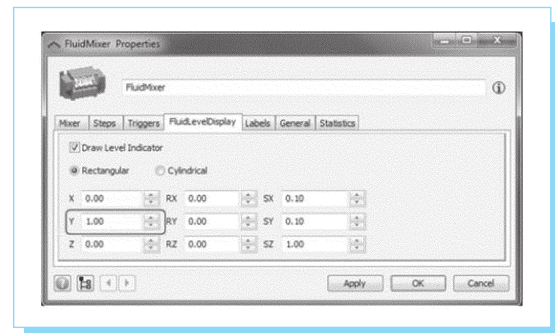


Fig. 10. Configuración del indicador de nivel del FluidMixer.

ventana de propiedades del FluidProcessor en la pestaña FluidProcessor. Se pueden modificar estos valores si desea ver cómo la operación del objeto afecta al modelo. Para hacer visible la barra indicadora de nivel de este objeto dentro de la pestaña FluidLevelDisplay repita la acción realizada en el objeto anterior.

Configuración del FluidToItem. El objeto que transforma el material fluido en Flowitems se llama FluidToItem. Este objeto multiplica el material fluido por las unidades discretas deseadas para determinar la cantidad de materia que deberá constituir una sola pieza o Flowitem. En este modelo, 0.5 litros de material fluido se convertirá en 1 Flowitem y este nuevo Flowitem será llamado Postre.

1. Hacer doble clic en el FluidToItem para abrir su ventana propiedades. En la pestaña FluidToItem, buscar el campo llamado Fluid per Discrete Unit y asignar un valor de 0.5 para indicar que por cada medio litro de material fluido se deberá crear un Flowitem o porción de Postre.
2. En esta misma pestaña cambie el Contenido Máximo a 0.5. Esto le dice al objeto que en todo momento sólo puede recoger material fluido suficiente para 1 Flowitem. Si este valor es mayor, el FluidToItem formará una línea de espera y con ello se creará demasiado espacio de almacenamiento en el modelo.
3. Para finalizar, dentro del campo Flowitem Name, escribir Postre. Así, todos los nuevos Flowitems serán llamados Postre.
4. Dar Apply y Ok para guardar los cambios.

Configuraciones finales. Una vez terminado con el objeto FluidToItem, se programan los siguientes objetos:

- Para este modelo, los Conveyors se mantienen con sus propiedades de inicio pero deberán ser indicadas sus dimensiones y dirección hasta lograr el diseño deseado.
- Al Combiner llegan tanto frascos como Flowitems Postres. Para que la velocidad de envase sea de 3 600 frascos por hora, dentro del campo Process Time de la ventana de propiedades del Combiner, colocar un 1 para indicar que sea envasado un frasco por segundo. En la pestaña Combiner elegir el modo de combinar Join para visualizar en la salida sólo un producto. Y finalmente, dentro de la pestaña Triggers, puede cambiar el color de los frascos de salida a un color azul.

4.7. Ejecución del modelo

Para ejecutar el modelo, definir el tiempo de simulación (por ejemplo, 576 000 segundos) y después dar Reset y Run. Una vez que el modelo esté corriendo, puede observarse como las tuberías parpadean. Si una tubería se muestra de color gris, significa que está vacía. Si se trata de un color opaco, el material fluye a través de la tubería. Y si se trata de un color vivo o brillante, el material está bloqueado. Además, puede verse cómo la barra indicadora de los objetos baja y sube de nivel conforme entra y sale el material fluido. Obsérvese también cómo los Flowitems son transformados en líquido y en una etapa posterior el líquido es transformado en Flowitems (véase la figura 11).

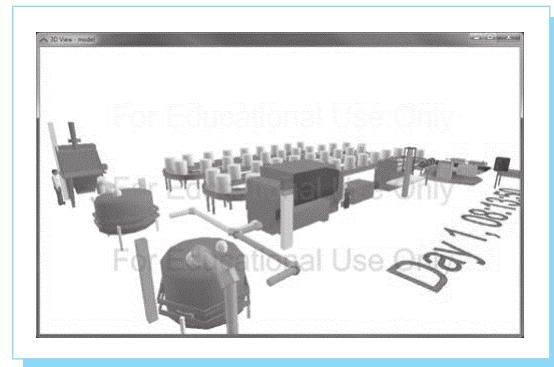


Fig. 11. Presentación del modelo final.

5. Resultados

El desempeño del sistema se describe con estadísticas. Las estadísticas son la acumulación de valores en una variable por un tiempo determinado, inician en 0 cada vez que se restaura (reset) el modelo y se acumulan valores hasta el momento en que se detiene la simulación. Además, estos resultados pueden ser vistos como números, porcentajes o gráficos y pueden programarse variables de interés para el usuario o simplemente consultarse las ya definidas.

Una vez que se ha ejecutado el modelo durante un período de tiempo predeterminado, se pueden analizar los resultados de la ejecución en muchas y muy variadas formas.

1. Algunas estadísticas que pueden ser observadas de forma rápida, se encuentran disponibles en los objetos que se muestran en el área de trabajo.
2. Además, también es posible agregar recursos especiales de la librería de objetos. Por ejemplo, el Recorder permite añadir gráficas dinámicas y variables, las cuales se animan mientras el modelo se está simulando.
3. En el menú Statistics > Reports and Statistics se puede tener acceso a un informe completo, a un resumen del informe o quizá a un informe del estado en el que se encuentra el modelo.

Las estadísticas pueden ser muchas y muy variadas y dependerán de cómo sean solicitadas. En este ejemplo, después de revisar el reporte estadístico se puede observar cómo al final del tiempo simulado se envasaron 383,929 frascos y se podría saber cuánto tiempo estuvieron en operación, vacíos, bloqueados, en estado ocioso, descompuestos, entre otros (véase figura 12).

Finalmente, recordar que para hacer una inferencia válida, los modelos de simulación se estabilizan haciendo ejecuciones con periodos de tiempo largos. Además, es importante tener en cuenta la importancia de ejecutar el número necesario de réplicas, puesto que las ejecuciones múltiples contribuyen también a mejorar la validez de la inferencia. Un mayor número de réplicas implica un número más elevado de muestras aleatorias independientes con distribución estadística diversas.

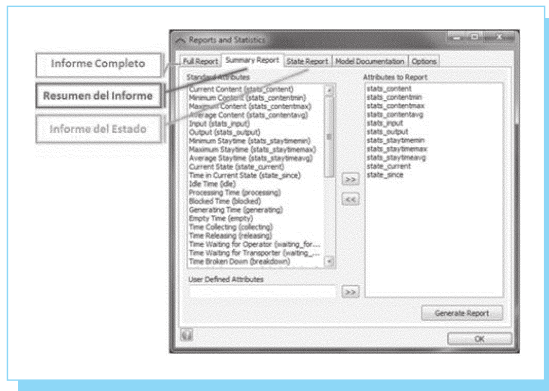


Fig. 12. Informe de las estadísticas de operación de un modelo de simulación.

6. Conclusiones

Después de entender la terminología general de la simulación y de haber construido su propio modelo en FlexSim el modelador deberá tener una idea clara de cómo operan los objetos fluidos en FlexSim. Además, como resultados de esta investigación, el lector comprobará que la simulación de un modelo desarrollado con FlexSim permite tomar mejores decisiones en la operación de los sistemas industriales, los cuales buscan mejorar su rentabilidad. Finalmente, se espera que este artículo refleje la importancia de la simulación tanto a investigadores, como a catedráticos y alumnos hacia el involucramiento en la construcción, análisis e interpretación de modelos más grandes, complejos y realistas, basados en la simulación con FlexSim.

En este artículo se ha puesto énfasis en el procedimiento para construir un modelo de simulación con la ayuda de la herramienta FlexSim. Sin embargo, debido a que muchos temas del proceso de simulación no se cubren en su totalidad pero son de gran importancia, es necesario que el lector interesado en el tema consulte [1, 2, 3, 5, 7, 14, 19].

Referencias

- [1] J. Banks, J. S. Carson, B. L. Nelson, y D. M. Nicol. *Discrete-Event System Simulation*, 4ª ed., USA: Prentice-Hall, 2005.
- [2] D. W. Kelton, R. P. Sadowski, y D. T. Sturrock. *Simulación con Software Arena*, 4ª ed., México: McGraw-Hill, 2008, pp. 1-25.
- [3] M. Beaverstock, A. G. Greenwood, E. Lavery, y W. Nordgren. *Applied Simulation Modeling and Analysis using FlexSim*. 3ª ed., FlexSim Software Products, Inc., Orem USA, 2012.
- [4] J. Acosta Flores. *Ingeniería de sistemas: un enfoque interdisciplinario*. 2ª ed., Centro de Investigaciones Interdisciplinarias en Ciencias y Humanidades de la UNAM, México: AlfaOmega, 2007, pp. 1-26.
- [5] Law Averill M. & Associates Inc. *Simulation Modeling and Analysis*, 4ª ed., Tucson, Arizona: U.S.A. McGraw-Hill, 2007, pp. 1-273.
- [6] W. L. Winston. *Investigación de operaciones aplicación y algoritmos*, 4ª ed., México, Thomson Learning, 2005, pp. 1-273, 1145-1158.
- [7] J. Von Neumann. *John Von Neumann y los orígenes de la computación moderna*, William Aspray-Gedisa, 1992.
- [8] R. E. Nance. *A History of Discrete Event Simulation Programming Language*, Department of Computer Science, Virginia Polytechnic Institute and State University, Blacksburg, Virginia, 1993, pp. 4-39.
- [9] K. D. Tocher, y D.G. Owen. *The Automatic Programming of Simulations*, Proceedings of the Second International Conference on Operational Research, 1960, pp.50-68.
- [10] M. Barceló G.. *Una historia de la informática*, Rambla del Poblenou, Barcelona, UOC, 2008, pp. 77-80.
- [11] R. L. Wexelblatt. *History of Programming Languages*, Academic Press, Proceedings of the ACM SIGPLAN History of Programming Languages Conference, 1-3 junio, 1978.
- [12] H. M. Marcowitz, B. Hausner, y H. W. Karr., *SIMSCRIPT: A Simulation Programming Language*, The RAND Corporation, Prentice Hall, 1963.
- [13] Esso, (1963). *C.S.L.: Reference Manual/Control and Simulation Language*, Esso Petroleum Company, Ltd. & IBM United Kingdom, 1963.
- [14] B. Zeigler, T. Gon Kim, y H. Praehofer. *Theory of Modeling and Simulation*, 2ª ed., New York: Academic Press, 2000.
- [15] R. M. Bryant. *SIMPAS - a Simulation Language Based on PASCAL*, Proceedings of the 1980 Winter Simulation Conference, T.I. Ören, C.M. Shub, and P.F. Roth, eds., 1980, pp. 25-40.
- [16] R. M. Bryant. *A Tutorial on Simulation Programming with SIMPAS*, In Proceedings of the 1981 Winter Simulation Conference, T.I. Ören, C.M. Delfosse, and C.M. Shub (eds.), pp. 363-377.
- [17] R. Lakshmanan, *Design and Implementation of a PASCAL Based Interactive Network Simulation Language for Microcomputers*, unpublished Ph.D. dissertation, Oakland University, Rochester, Michigan, 1983.
- [18] FlexSim Software Products, Inc. *Improve performance. Save money, Who is using FlexSim?*, Consultado en enero 2012 en <http://www.flexsim.com/>.
- [19] R. Coss Bú. *Simulación: un enfoque práctico*, (20ª ed.), Noriega/Departamento de Ingeniería Industrial, Instituto Tecnológico y de Estudios Superiores de Monterrey/Limusa, Médico DF, 2003, pp. 11-18.