

Simulación en la Ingeniería

Simulation in Engineering

MARTÍNEZ-MARÍN, Francisco Alejandro†*, MEDINA-LOZANO, Alejandra, CANTÚ-MUNGUÍA, Irma Adriana y MARTÍNEZ-MENDOZA, María Lizbeth

Instituto Tecnológico Mario Molina Pasquel y Henríquez, Campus Vallarta

ID 1^{er} Autor: *Francisco Alejandro, Martínez-Marín* / ORC ID: 0000-0002-3712-5182

ID 1^{er} Coautor: *Alejandra, Medina-Lozano* / ORC ID: 0000-0002-4458-244X

ID 2^{do} Coautor: *Irma Adriana, Cantú-Munguía* / ORC ID: 0000-0002-1904-421X

ID 3^{er} Coautor: *María Lizbeth, Martínez-Mendoza* / ORC ID: 000-0002-1840-9221

DOI: 10.35429/JSL.2019.18.6.1.6

Recibido 26 Febrero, 2019; Aceptado 30 Marzo, 2019

Resumen

La simulación es un método por el cual se puede probar o experimentar o predecir resultados en determinado proceso, escenario u objeto sin el riesgo de consecuencias reales. El objetivo de esta investigación es demostrar como la simulación es una de las más grandes herramientas en la ingeniería, la cual se utiliza para representar un proceso mediante otro y lo hace mucho más simple y entendible. Esta simulación, es en algunos casos, casi indispensable. En el trabajo de simulación se plantea una metodología para el diseño, desarrollo y evaluación. El mismo se basa en la sinergia de dos campos del saber aparentemente disímiles: la ingeniería de software por un lado y las teorías de aprendizaje modernas por el otro, pero que convergen en la generación de un producto deseable: el software de simulación. Esta metodología se basa en la aplicación de reglas existentes en ambos campos. Como resultado se tiene el empleo del software de simulación, como una técnica numérica para conducir experimentos en una computadora digital. Estos experimentos comprenden ciertos tipos de relaciones matemáticas y lógicas, las cuales son necesarias para describir el comportamiento y la estructura de sistemas complejos del mundo real por largos periodos de tiempo.

Abstract

Simulation is a method by which you can test, experiment or predict results in a certain process, scenario or object without the risk of real consequences. The objective of this research is to demonstrate how simulation is one of the greatest tools in engineering, which is used to represent a process through another and makes it much simpler and more understandable. This simulation is, in some cases, almost indispensable. In the simulation work, a methodology for design, development and evaluation is proposed. It is based on the synergy of two apparently dissimilar fields of knowledge: software engineering on the one hand and modern learning theories on the other, but which converge on the generation of a desirable product: simulation software. This methodology is based on the application of existing rules in both fields. As a result we have the use of simulation software, as a numerical technique to conduct experiments in a digital computer. These experiments involve certain types of mathematical and logical relationships, which are necessary to describe the behavior and structure of complex real-world systems over long periods of time.

Simulación, Programa, Variables

Simulation, Program, Variables

Citación: MARTÍNEZ-MARÍN, Francisco Alejandro, MEDINA-LOZANO, Alejandra, CANTÚ-MUNGUÍA, Irma Adriana y MARTÍNEZ-MENDOZA, María Lizbeth. Simulación en la Ingeniería. Revista de Simulación y Laboratorio. 2019, 6-18: 1-6

†Investigador contribuyendo como primer Autor

Introducción

Durante estos últimos años la Física Experimental en el área educativa a nivel universitario como en la enseñanza media se ha deteriorado fundamentalmente por la falta de recursos económicos, ya que, sus equipos son de alto costo y necesitan de manutención y personal bien preparado. La computación ofrece la alternativa a través de simulaciones de experimentos permitiendo mejorar en parte esta situación. Construir un experimento en un computador es más barato, no se corre el riesgo de destrucción del equipo y puede repetirse el experimento cuantas veces sea necesario.

La simulación es un método por el cual se puede probar o experimentar o predecir resultados en determinado proceso, escenario u objeto sin el riesgo de consecuencias reales. Luego permite comparar diferentes soluciones ante un problema real, probarlas y ver cuál es la mejor, posteriormente, aplicar esa solución que funcionó adecuadamente en la simulación, en el mundo físico y esperar por las consecuencias que el modelo de simulación mostró, sean las mismas a la realidad. Por ese motivo cuando se planea una simulación, ésta debe ser lo más realista posible o acercarse lo suficiente a la realidad.

La expresión: “la solución funcionaba correctamente en el modelo, no entiendo la razón por la cual no funciona en la realidad”, significa que la simulación no fue realizada correctamente, es decir, el modelo matemático de la situación real tiene fallas como no deducir correctamente como las diferentes variables afectan el todo o en el peor de los casos, olvidar incluir variables fundamentales.

Una de las lecciones que primero se aprenden al diseñar modelos de simulación, es darse cuenta que el mundo real es indeterminista, esto significa la existencia de variables fuera de nuestro control, a lo sumo estimar su comportamiento; estas variables son conocidas como variables aleatorias, y funcionan con fórmulas donde el azar es el protagonista.

Uno de los temas más llamativos es mostrar que haciendo uso de modelos de simulación, la posibilidad de solucionar problemas complejos, por ejemplo:

Encontrar el área bajo la curva dibujada por una función algebraica requiere el resolver integrales, una tarea relativamente difícil, sin embargo, usando el método Montecarlo, encontrar el área bajo cualquier curva puede ser fácil, donde el resultado obtenido es una aproximación.

Objetivo

Aplicación de software de simulación para la resolución de ejercicios de dinámica por parte de los alumnos de alumnos del Instituto Tecnológico José Mario Pasquel y Henriquez, campus Puerto Vallarta, con la finalidad de facilitar el aprendizaje de la Dinámica.

Método

En el trabajo de simulación se plantea una metodología para el diseño, desarrollo y evaluación del programa de simulación. El mismo se basa en la sinergia de dos campos del saber aparentemente disímiles: la ingeniería de software por un lado y las teorías de aprendizaje modernas por el otro, pero que convergen en la generación de un producto deseable: el software de simulación. Esta metodología se basa en la aplicación de reglas existentes en ambos campos.

Los alumnos objeto de esta investigación, son estudiantes de Dinámica del tercer semestre de Ingeniería electromecánica. Los alumnos tenían un promedio de 20 años de edad. Se analizó el uso de un software de simulación matemática, (Matlab), puede ser un recurso didáctico para facilitar el aprendizaje de la dinámica. Ya que Generalmente los estudiantes se sienten atraídos e interesados por todo el software educativo, ya que los programas suelen incluir elementos para captar la atención de los alumnos, mantener su interés y, cuando sea necesario, focalizarlo hacia los aspectos más importantes en las actividades. Por lo tanto la función motivadora es una de las más características en este tipo de materiales didácticos, y resulta extremadamente útil para los profesores.

Los programas no directivos, especialmente las bases de datos, simuladores tales como programas constructores, ofrecen a los estudiantes interesantes entornos donde investigar: buscar determinadas informaciones, cambiar los valores de las variables en un sistema, entre otros.

Además, estos programas como herramienta, pueden proporcionar a los profesores y estudiantes instrumentos de gran utilidad para el desarrollo de trabajos de investigación que se realicen básicamente al margen de los ordenadores.

Resultados

El empleo del software de simulación, es una técnica numérica para conducir experimentos en una computadora digital. Estos experimentos comprenden ciertos tipos de relaciones matemáticas y lógicas, las cuales son necesarias para describir el comportamiento y la estructura de sistemas complejos del mundo real por largos periodos de tiempo.

Matlab es un programa en cálculo matemático muy flexible y potente con posibilidades gráficas para la presentación de los datos, con aplicaciones en muchos campos de la ciencia y la investigación como herramienta de cálculo matemático. En este documento se presenta un solo ejemplo de entre las inmensas posibilidades de Matlab. El consultar las ayudas de este software u otros documentos será de gran utilidad para obtener un conocimiento más amplio. Se debe utilizar para modificar el ejemplo aquí incluido para adquirir las destrezas básicas que permitan utilizar el programa como herramienta para determinada asignatura.

La simulación es una de las más grandes herramientas en la ingeniería, la cual se utiliza para representar un proceso mediante otro y lo hace mucho más simple y entendible. Esta simulación es en algunos casos casi indispensable, como nos daremos cuenta a continuación. En otros casos no lo es tanto, pero sin este procedimiento se hace más complicado.

Ejemplo:

“Una pelota se lanza verticalmente hacia arriba desde una altura de 12 metros en el pozo de un elevador con una velocidad inicial de 18 m/s. En el mismo instante un elevador de plataforma abierta pasa por el nivel de 5 m, moviéndose hacia arriba con una velocidad constante de 2 m/s. Determine a) cuándo y dónde golpea al elevador, b) la velocidad relativa de la pelota con respecto al elevador cuando ésta lo golpea”. (Beer, 2010).

Las Figs. 1, 2, 3 muestran la posición de la pelota y el elevador en los diferentes momentos.

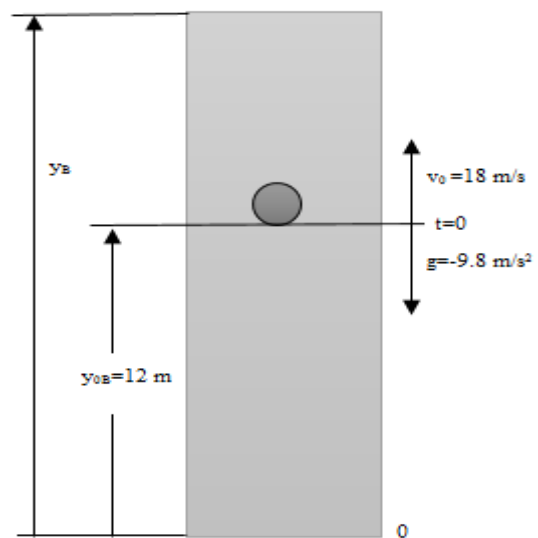


Figura 1 Pelota en condiciones iniciales
Fuente: Elaboración Propia

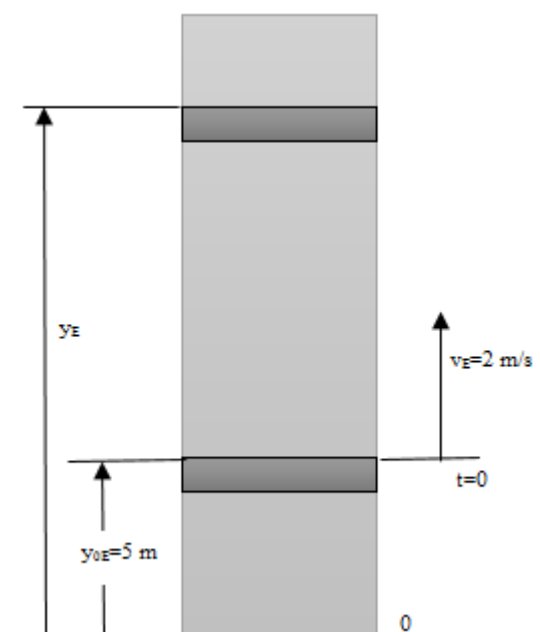


Figura 2 Elevador en condiciones iniciales.
Fuente: Elaboración Propia

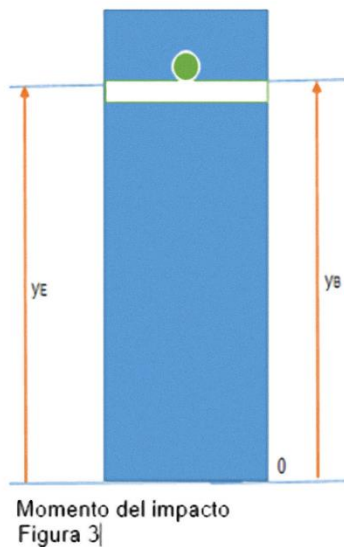


Figura 3 Momento del impacto
Fuente: Elaboración Propia

Movimiento de la pelota.
Movimiento rectilíneo uniformemente acelerado

$$v_b = v_{0b} + at$$

$$v_b = 18 - 9.81t \quad (1)$$

$$y_b = y_{0b} + v_{0b}t + \frac{1}{2}at^2$$

$$y_b = 12 + 18t - 4.9t^2 \quad (2)$$

Movimiento del elevador.
Movimiento rectilíneo uniforme (velocidad constante)

$$v_e = +2 \text{ m/s} \quad (3)$$

$$y_e = y_{0e} + v_e t$$

$$y_e = 5 + 2t \quad (4)$$

La pelota golpea al elevador cuando las alturas son iguales.

$$y_e = y_b$$

Igualando las ecuaciones 2 y 4 y resolviendo.

$$5 + 2t = 12 + 18t - 4.90t^2 \quad (5)$$

$$t = 3.65 \text{ s}$$

Posición del elevador.

$$y_e = 5 + 2(3.65) = 12.3 \text{ m} \quad (6)$$

Elevación desde el suelo = 12.30 m

La velocidad relativa de la pelota con respecto al elevador es.

$$v_{b/e} = v_b - v_e$$

$$v_{b/e} = (18 - 9.81t) - 2 \quad (7)$$

Para un tiempo $t=3.65 \text{ s}$

$$v_{b/e} = 16 - 9.81(3.65) = -19.81 \text{ m/s} \quad (8)$$

Nomenclatura

a	Aceleración (m/s ²)
t	Tiempo (s)
y _e	Posición del elevador (m)
y _{0e}	Posición inicial del elevador (m)
y _b	Posición de la pelota (m)
y _{0b}	Posición inicial de la pelota (m)
v _e	Velocidad del elevador (m/s)
v _b	Velocidad de la pelota (m/s)
v _{0b}	Velocidad inicial de la pelota (m/s)
v _(b/e)	Velocidad relativa de la pelota respecto al elevador (m/s)

Programa en Matlab

```
clear
close all
clc
%Programa para resolver problema.
%Datos.

xx=[];
g=9.8;y0b=12;v0b=18;y0e=5;ve=2;vob=18;ye=5;yb=12;
Time=0;
delta=0.01;
while (ye<=yb)
    yb=y0b+v0b*time-(g*(time)^2)/2;
    ye=y0e+ve*time;
    vb=vob-g*time;
    xx=[xx;time,yb,ye,vb,ve];
    time=time+delta; end
vb_e=vb-ve; %Velocidad relativa de la pelota
respecto al elevador.
figure %Gráficas de posición de la pelota y el
elevador en el tiempo.
plot(xx(:,1),xx(:,2),'k')
hold on
plot(xx(:,1),xx(:,3),'k--')
title('Grafica 01')
xlabel ('Gráficas de posición de la pelota y el
elevador en el tiempo. ')
legend ('Posición de la pelota.','Posición del
elevador.')
grid
figure %Gráficas de la velocidad de la pelota y el
elevador en el tiempo.
plot(xx(:,1),xx(:,4),'k')
hold on
plot(xx(:,1),xx(:,5),'k--')
title('Grafica 02')
xlabel ('Gráficas de la velocidad de la pelota y el
elevador en el tiempo.')
legend ('Velocidad de la pelota.', 'Velocidad del
elevador.')
grid
```

```

%Resultados.
fprintf('El tiempo de choque se da en:')
disp(time)
fprintf('La altura de la pelota desde el suelo es:')
disp(yb)
fprintf('La velocidad relativa de la pelota respecto al
elevador es:')
disp(vb_e)

```

Solución

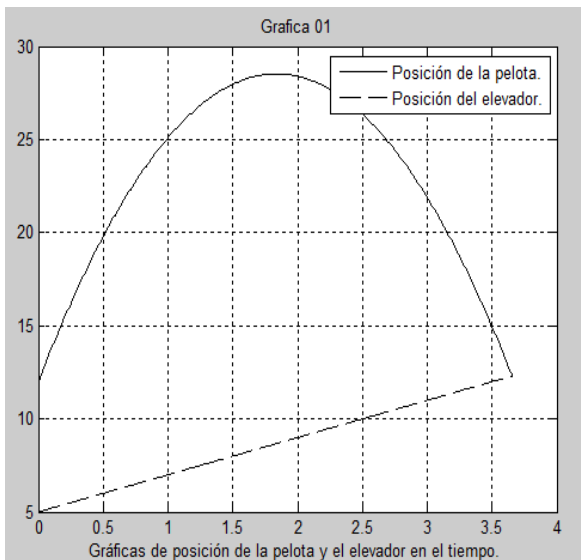


Figura 4 Gráfica de posición de la pelota y elevador
Fuente: Elaboración Propia

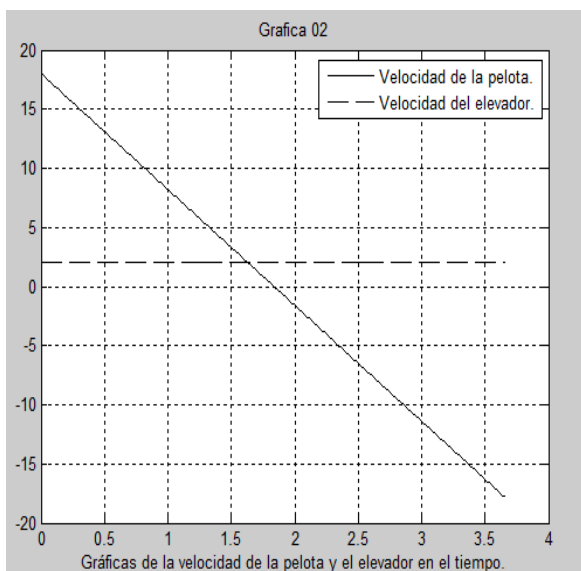


Figura 5 Gráfica de velocidad de la pelota y elevador
Fuente: Elaboración Propia

```

Command Window
El tiempo de choque se da en: 3.6700
La altura de la pelota desde el suelo es: 12.2416
La velocidad relativa de la pelota respecto al elevador es: -19.8680
fx >>

```

Figura 6

Figura 6
Fuente: Elaboración Propia

Conclusiones

Las Figs. 4, 5 muestran las gráficas obtenidas en los resultados, en ningún momento representan la trayectoria seguida por la pelota y elevador en el tiempo, la primera permite apreciar el punto de choque entre el elevador y la pelota, la segunda los valores positivos y negativos de la velocidad de la pelota y su interpretación física y también la velocidad constante del elevador, la Fig. 6 muestra los resultados obtenidos al correr el programa, la construcción del mismo exige al alumno una total comprensión del problema a resolver y mejora la habilidad en interpretar gráficos y resultados.

Mostrando el utilizar la programación como una herramienta adicional para la docencia, mejoran la atención del estudiante y, también, la productividad del maestro. Pero el usar lenguaje de programación no se limita a Matlab, ahora ya se pueden incorporar en el salón de clases elementos tales como tarjetas de adquisición de datos, sensores, actuadores y circuitos de electrónica. Éstas tecnologías a la par con técnicas dinámicas en enseñanza permiten activar procesos cognitivos en el alumno, propiciando un aprendizaje significativo, debido a que cuando se construye un laboratorio simulado, es posible aumentar el conocimiento mediante la aplicación de procesos similares al método científico: formulando hipótesis sobre un fenómeno y poniendo a prueba estas hipótesis mediante experimentos, toda esta experiencia será el cimiento para los futuros desarrolladores de ciencia y tecnología.

La experiencia descrita ha sido muy productiva. Sin embargo se podría mejorar en algunos aspectos, tales como:

Impartir cursos de Matlab y manejo de tarjetas Arduino dentro del curso propedéutico atendido por alumnos de primer ingreso, así no sería necesario ocupar el tiempo establecido para cubrir el contenido de la asignatura.

Tener un abanico más amplio de ejemplos en problemas reales en los cuales se puedan desarrollar proyectos que involucren el cálculo y la tecnología. De esta manera el grado de interés sería mayor y podrían ajustarse un poco más a los intereses particulares de más alumnos.

Elaborar un guion de trabajo de todo el curso, y que el alumno lleve una bitácora de las actividades realizadas para que sea consciente de sus logros. Sería una manera de introducirlos en la metodología de la investigación. Y la estrategia de enseñanza aprendizaje basado en proyectos en asignaturas del corte herramental o básico, tales como Cálculo Vectorial, Probabilidad, Estadística Inferencial, etc., pues se ha constatado que utilizando los ejemplos adecuados se consiguen óptimos resultados.

Referencias

Bedfor (2008). Engineering Mechanics: Dynamics, USA: Prentice Hall.

Beer J. (2010), Mecánica vectorial para ingenieros. Dinámica, Mc Graw Hill, p. 620. México D.F.

Hibbler, R. (2010). Engineering Mechanics: Combined Statics and Dynamics, 12º Edición ed., E.U.: Prentice Hall.

Jalón, R. B. (2001). Aprenda Matlab 6.1 como si estuviera en primero, Madrid.

Soutas, R. (2009). Mecánica para Ingenieros: Dinámica., 1 Edición ed., México: Cengage Learning,