

SIMULACIÓN DE LA DINÁMICA VERTICAL DE UN CUARTO DE VEHÍCULO MEDIANTE MATLAB, SIMULINK Y SIMSCAPE

SIMULATION OF THE VERTICAL DYNAMICS OF A QUARTER VEHICLE BY MATLAB, SIMULINK AND SIMSCAPE

Jorge A. Reula^a, Alejandro O. Peralta^a y Elvio Heidenreich^a

^a*Departamento de Ingeniería Mecánica, Facultad. Ingeniería del Ejército, Cabildo 15, Buenos Aires, Argentina, jorgereula9@gmail.com, www.ingenieriaest.iese.edu.ar*

Palabras clave: Dinámica Vertical, Sistemas Mecánicos Multicuerpos Rígidos y Simulaciones Numéricas.

Resumen. El Laboratorio de Simulación y Cálculo Numérico (SiCaNLab) de la Facultad de Ingeniería del Ejército ha decidido resolver la dinámica vertical de un modelo de un cuarto de vehículo mediante el SOFTWARE Matlab y sus dos Toolbox Simulink y Simscape, analizando las ventajas de cada herramienta para el uso académico. Para ello, fue desarrollado un modelo empleando el Toolbox Simulink(C) y otro modelo empleando el Toolbox Simscape(C), comparando a los mismos con la resolución de las ecuaciones diferenciales ordinarias (ODE) por medio de el algoritmo de Runge Kutta de 4to orden programando en Matlab(C). Finalmente se contrastará la precisión de los modelos.

Con este trabajo se pretende validar los modelos de la suspensión de un cuarto de vehículo para uso en futuros estudios en el área de dinámica vehicular o de sistemas multicuerpos.

Keywords: Vertical Dynamics, Mechanical Systems Rigid Multibodies and Numerical Simulations.

Abstract. The Simulation and Numerical Calculation Laboratory (SiCaNLab) of the Faculty of Engineering of the Army has decided to solve the vertical dynamics of a model of a vehicle room by means of the Matlab SOFTWARE and its two Simbox and Simscape Toolbox, analyzing the advantages of each tool for academic use For this, a model was developed using the Simulink(C) Toolbox and another model using the Simscape(C) Toolbox, comparing them to the resolution of ordinary differential equations (ODE) by means of the 4th order Runge Kutta algorithm programming in Matlab(C). Finally, the accuracy of the models will be checked.

This work is intended to validate the suspension models of a vehicle room for use in future studies in the area of vehicle dynamics or multi-body systems.

1. INTRODUCCIÓN

El presente trabajo tiene como objetivo establecer la importancia de la utilización de SOFTWARE en la educación para el modelado y resolución de fenómenos físicos. Para alcanzar este objetivo se utiliza el SOFTWARE Matlab y sus respectivos Toolbox Simulink y Simscape, los cuales poseen diferentes características y niveles de complejidad. Se abarca una comparación de estos métodos de resolución y las posibilidades de su utilización por parte de los estudiantes. Para este trabajo se há tomado como base el artículo (Sharma et al., 2013). Estos autores realizan un programa en Matlab con la finalidad de analizar el sobre impulso y tiempo de establecimiento de un modelo de automóvil de dos grados de libertad.

Actualmente hay un marcado incremento del interés en el área de las simulaciones debido a las ventajas que trae respecto a los modelos físicos o prototipos. Entre ellas podemos nombrar:

1. Análisis del comportamiento de los sistemas en diferentes condiciones sin poner en peligro vidas ni medios.
2. Reducción de costos, tiempos y modificación de los parámetros de los prototipos permitiendo infinidad de pruebas.
3. Obtención de las respuestas temporales de los sistemas y la variación de las mismas en función de las diferentes condiciones establecidas.
4. Diseño y aplicación de parámetros de control.

En el ámbito Universitario podemos establecer como ventajas:

1. Estimulación del alumno a adquirir nuevos conocimientos.
2. Facilidad para comprender la influencia de la variación de parámetros en los diferentes modelos agilizando la repetición de cálculos, en este caso en particular en la variación de la dinámica vertical vehicular.
3. Visualización de interacción de componentes físicos.

Es de gran interés para los alumnos el empleo de estas herramientas para profundizar los conocimientos adquiridos en clases y, a futuro como profesionales, para la aplicación y obtención de posibles soluciones a los problemas de una organización.

Como hipótesis del modelo se estableció:

1. Se considera al vehículo y a sus componentes rígidos.
2. Se considera la rigidez y absorción del neumático por separado.
3. Se considera los valores característicos del amortiguador y resorte constantes.

En este trabajo se procede a resolver el modelado del sistema de suspensión de un cuarto de vehículo Fig. (1).

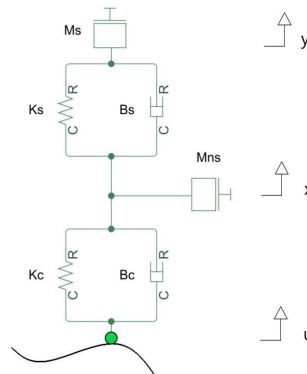


Figura 1: Esquema del problema físico

Parámetro	Símbolo	Unidad	Valor
Masa suspendida.	Ms	kg	275
Constante de amortiguación del amortiguador.	Bs	$N \cdot s/m$	1120
Constante elástica del resorte.	Ks	N/m	150000
Masa no suspendida.	Mns	kg	27
Constante de amortiguación de la cubierta.	Bc	$N \cdot s/m$	310000
Constante elástica de la cubierta.	Kc	N/m	3100
Perturbación del sistema.	u	m	0.1

Tabla 1: Constantes características del modelo.

2. MODELO MATEMÁTICO

El objeto de estudio es el planteo de las ecuaciones, establecer los métodos de resolución por simulación y la comparación de los mismos en cuanto a la utilidad académica. Las condiciones iniciales son presentadas mediante la Ec. (1) y la Ec. (2) que representan las posiciones y velocidades iniciales respectivamente.

$$y_0 = 0; x_0 = 0; u_0 = 0; \quad (1)$$

$$\dot{y}_0 = 0; \dot{x}_0 = 0; \dot{u}_0 = 0; \quad (2)$$

A continuación se transcriben las ODE de segundo orden acopladas, las cuales modelan el comportamiento temporal de un cuarto de vehículo. La Ec. (3) representa el movimiento vertical de la masa suspendida y la Ec. (4) representa el movimiento de las masas no suspendidas.

$$m_s \cdot \ddot{y} + b_s \cdot (\dot{y} - \dot{x}) + k_s \cdot (y - x) = 0 \quad (3)$$

$$m_{ns} \cdot \ddot{x} - b_s \cdot (\dot{y} - \dot{x}) - k_s \cdot (y - x) + b_c \cdot (\dot{x}) + k_c \cdot (x - u) = 0 \quad (4)$$

3. SOFTWARE DE RESOLUCIÓN

Para la resolución del presente problema mediante Matlab y sus Toolbox se procede a la configuración de los mismos.

Parámetro.	Matlab.	Simulink.	Simscape.
Tiempo de simulación.	4 [s]		
Método de resolución.	ODE 45	ODE 4	ODE 4
Paso de simulación.	fijo (0.001 [s])		
Tiempo de Resolución.	0.050213 [s]	0.47 [s]	0.88 [s]
Tolerancia relativa .	1.10^{-3}	1.10^{-3}	1.10^{-3}

Tabla 2: Configuración de los SOFTWARE.

Ademas se obtuvo la solución analítica de el sistema mediante la función transferencia del mismo Eq. (5).

$$\frac{Y(S)}{U(S)} = \frac{69440000s + 930000000}{1485s^4 + 238148s^3 + 26804400s^2 + 162440000s + 930000000} \quad (5)$$

Obteniéndose la respuesta ante un escalón de 0.1 [m] Fig. (2).

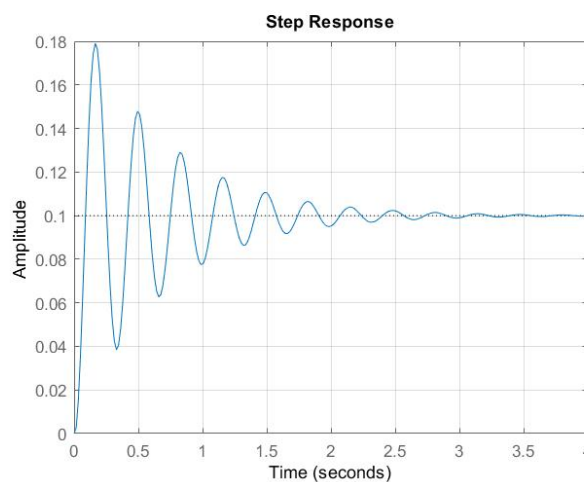


Figura 2: Solucion Analitica.

Para resolver estas ODE podemos optar por tres formas distintas:

3.1. Entorno Matlab.

Es un modo de resolución meramente matemático, por el cual mediante el comando ODE contiene una o más derivadas de una variable dependiente de otra, que generalmente es el tiempo 't', donde debemos aplicar las CI, el intervalo de tiempo a estudiar y las constantes del sistema. Se puede utilizar cualquier cantidad de ODE acopladas para resolver un sistema de ecuaciones, solo limitado por la capacidad de memoria de la computadora (Chapra y Canale, 2007).

$$\begin{pmatrix} \dot{y}_1 \\ \dots \\ \dot{y}_n \end{pmatrix} = \begin{pmatrix} f_1(t, y_1, \dots, y_n) \\ \dots \\ f_2(t, y_1, \dots, y_n) \end{pmatrix} \quad (6)$$

Si utilizamos una ODE de orden superior se realiza cambio de variables, para disminuir el orden de la misma. También se debe considerar la determinación de la ODE, según las ecuaciones diferenciales ordinarias sean rígidas o no.

Para este método de resolución se procedió a realizar los siguientes pasos:

1. Se definen las constantes del modelo. Tanto en Matlab, Simulink y Simscape, se establecen las constantes características del modelo físico, en las cuales se tienen que tener en cuenta que las mismas deben ser definidas en el Sistema Internacional del Medidas MKS.
2. Se establecen las condiciones iniciales. Las mismas se establecen en el programa inicial en forma de un vector la Ec. (7), cuya longitud dependerá de los grados de libertad del modelo físico. Este vector es introducido al programa solución con el comando ODE y allí, estas son seleccionadas y enunciadas en el sistema de ecuaciones diferenciales a resolver la Ec. (8).

$$CI = [0, \quad 0, \quad 0, \quad 0] \quad (7)$$

$$\ddot{y} = \frac{-b_s \cdot (\dot{y} - \dot{x}) - k_s \cdot (y - x)}{m_s} \quad (8)$$

3. Se utilizó el comando de resolución ODE de Matlab. Este comando la Ec.(9) envía las condiciones iniciales, según la Ec. (1) y la Ec. (2), conjuntamente con el tiempo de simulación y las condiciones odeset a la función solución de la Ec. (10) con la finalidad de obtener la solución del modelo. El comando odeset, si bien en muchas ocasiones se deja libre [], posee una gran variedad de herramientas las cuales influyen en la exactitud de la solución. Entre algunas opciones podemos nombrar: máximo paso, tolerancias de los errores tantos relativos como absolutos, paso inicial y refinamiento de las salidas. También hay que destacar que muchas de estas opciones deben ser programadas en un script separado lo cual resulta laborioso y da una mayor complejidad a esta forma de simular un fenómeno físico.

$$[t, Y] = ode45(@funSol, tspan, CI, odeset, Ctemodelo) \quad (9)$$

$$function [Y] = funSol(tspan, CI, Ctemodelo) \quad (10)$$

4. Se programa la resolución de las ecuaciones diferenciales, según las Ec. (11 - 14), las cuales están acopladas. Debido a que las ecuaciones son de segundo orden se debe utilizar el método de cambios de variables Eq. (6) para realizar la integración numérica y obtener, para nuestro caso, la posición, velocidad y aceleración de las masas, retornando estos valores a la función ODE, conjuntamente con los valores de tiempo donde fue evaluado nuestro modelo.

Además hay que tener en cuenta que para obtener la aceleración de alguna de las masas se debe realizar la derivación numéricamente. En este caso se utilizó el método de series de TAYLOR de primer orden, diferencias finitas centradas Eq. (15).

$$Y(1, 1) = CI(2) \quad (11)$$

$$Y(2, 1) = \frac{(-b_s \cdot (CI(2) - CI(4)) - k_s \cdot (CI(1) - CI(3)))}{m_s} \quad (12)$$

$$Y(3, 1) = CI(4) \quad (13)$$

$$Y(4, 1) = \frac{-b_s \cdot (CI(2) - CI(4)) - k_s \cdot (CI(1) - CI(3)) - b_c \cdot (CI(4)) - k_c \cdot (CI(3) - u)}{m_{ns}} \quad (14)$$

$$a_i = \frac{v_{i+1} - v_{i-1}}{t_{i+1} - t_{i-1}} \quad (15)$$

5. Se procedió a la gráfica de las respuestas. Para este paso Matlab posee una gran variedad de comandos para realizar gráficos, en este caso se procede a utilizar el comando PLOT el cual debe recibir como mínimo un par de vectores de la misma longitud. Para la edición/modificación de la vista de las gráficas se debe conocer una numerosa cantidad de comandos, los cuales son incluidos en el comando PLOT o en las líneas posteriores a la declaración de este.

3.2. Entorno Simulink.

Se basa en una plantilla que permite a través de diferentes bloques pre-diseñados realizar una secuencia lógica de señales y combinar las ecuaciones para obtener la respuesta temporal del fenómeno. Se realizan subsistemas los cuales se pueden evaluar independientemente de los otros, esto permite detectar errores fácilmente y solucionarlos.

Para este método de resolución se procedió a realizar los siguientes pasos:

1. Se procede a establecer las constantes en Matlab y correr el programa o utilizar el comando MASK de Simulink con el fin de correr la simulación directamente desde este último programa, lo cual también puede realizarse para Simscape.
2. Conexión de las ecuaciones siguiendo el orden de las mismas. Mediante las EDO se establecen los diferentes bloques que representan las constantes del modelo físico y operaciones matemáticas Fig. (3). Estos dan la posibilidad de variar sus parámetros cambiando

valores en su interior, lo cual da como ventaja evitar la programación de entradas al sistema y establecer las condiciones iniciales en cada bloque de integración, lo cual es familiar y sencillo para el estudiante. Las flechas conectan estos bloques y representan para nuestro caso las posición, velocidad y aceleración de las masas, las mismas pueden ser ramificadas para obtener un incremento de conexiones con otros bloques, teniendo la posibilidad de colocar el nombre a cada una, lo cual permite la fácil identificación de la misma en el diagrama y da la capacidad de automáticamente ver el nombre de la respuesta que se está graficando con el bloque SCOPE.

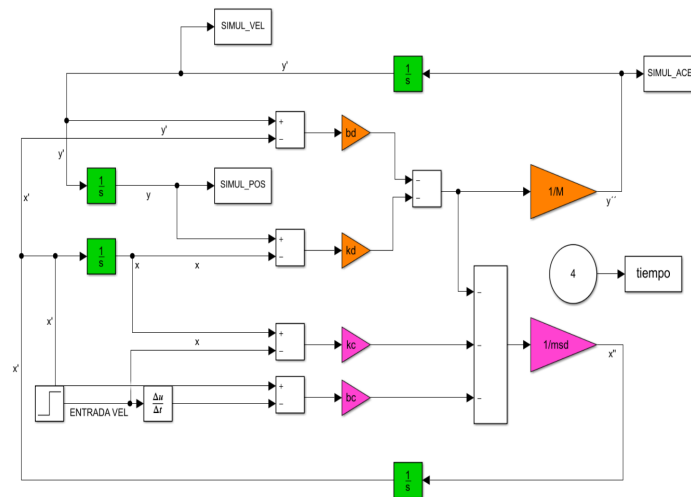


Figura 3: Modelo en Simulink de un cuarto de vehículo.

- Se procedió a las gráficas de las respuestas. Para esto simplemente se usa el bloque SCOPE, el cual se puede conectar en cualquier punto del diagrama de bloques. En nuestro caso se lo conecta en las flechas que representan los desplazamientos de las masas. Lo que permite tener la gráfica de las mismas y mediante opciones, las cuales se acceden mediante un cuadro de opciones, se puede editar los parámetros de visualización fácilmente. Sin embargo para poder obtener una comparativa entre los tres métodos seleccionados de simulación, se importó las soluciones mediante el bloque TO WORKSPACE a Matlab.

3.3. Entorno Simscape.

Permite crear modelos físicos dentro de Simulink utilizando flujos de potencia. Se construye modelos basados en conexiones físicas, empleado librerías pre diseñadas que se integran directamente. Se pueden crear y personalizar componentes mediante códigos C en Matlab (Miller y Wendlandt, 2010).

Para este método de resolución se procedió a realizar los siguientes pasos:

- Configuración y conexión de los componentes físicos del modelo. Para este caso se selecciona la masa suspendida correspondiente a un cuarto de vehículo, un resorte y amortiguador los cuales representan al sistema de suspensión. Además se posee la masa no suspendida correspondiente a la rueda, un resorte y amortiguador que representan la elasticidad y amortiguación que brinda la cubierta. Como se puede ver en el modelo Fig. (4), simplemente se selecciona los componentes,

se diagrama el modelo de igual manera que se encuentra dispuesto y conectado en el vehículo.

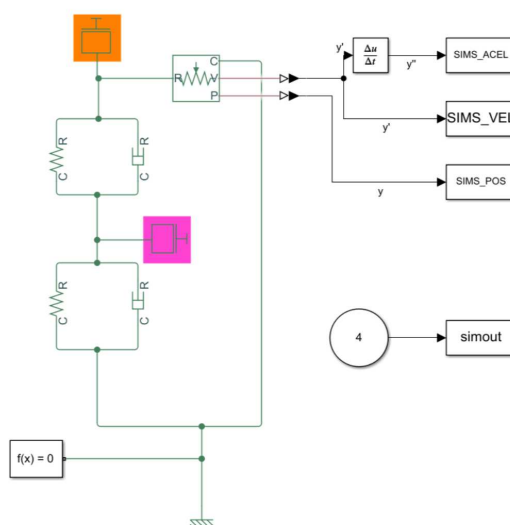


Figura 4: Modelo en Simscape de un cuarto de vehículo.

2. Modificación de las características de los bloques físicos. Los cambios de la constante característica de cada componente se lo hace introduciendo el valor en cada bloque físico y además cada uno de ellos tiene la posibilidad de asignarles condiciones iniciales con un orden de prioridad con diferentes unidades de medida. Cabe aclarar que cada bloque funciona como una caja negra ya que en su interior realiza cálculos, los cuales relacionan la entrada con la salida y el usuario desconoce.
3. Bloque físico referencia. Marcará un punto fijo a un sistema de referencia para el modelo físico, sensores y fuentes ideales.
4. Bloque solución. Este bloque permite establecer parámetros de exactitud de la solución del modelo físico, como ser: tolerancia, tiempo de muestreo, método de integración numérica, determinación de la solución en el dominio del tiempo o de la frecuencia entre otras opciones.
5. Entradas y salidas del modelo físico. En el caso de entradas al sistema primero hay que considerar las fuentes. Las fuentes son ideales y podemos mencionar fuentes ideales de velocidad y fuerza, estas poseen tres puertos: una referencia a tierra, una señal de entrada y la salida. En cuanto a la señal de entrada esta puede ser física o matemática, en este último caso es necesario utilizar el bloque convertidor Simulink TO PHYSICAL. Para el caso de la salida se realiza la obtención de información mediante sensores, los cuales pueden ser de fuerza, posición o velocidad. Estos bloques poseen tres terminales que obtienen la variación de la magnitud física a medir y un terminal de salida. Esta última se puede graficar a través de Matlab, que es lo que se realizó en este caso u optar por el bloque SCOPE, para cualquiera de los dos casos hay que utilizar un bloque convertidor PHYSICAL TO Simulink.

4. RESULTADOS

En cuanto a las tres formas de resolución podemos realizar una comparación de simplicidad de aplicación, asignándole a cada característica un valor de 1 a 3, según se puede observar en la Tab. (3).

Características	Matlab.	Simulink.	Simscape.
Norma de vectores con respecto a Matlab.	–	$1,6594,10^{-5}$.	$1,6594,10^{-5}$.
Tiempo de resolución.	0.050213 [s]	0.47 [s]	0.88 [s]
Simplicidad de ingreso de constantes.	1	2	3
Multiplicidad de forma de ingreso de constantes.	1	2	2
Simplicidad de modificación de constantes.	1	3	2
Ingreso de condiciones iniciales.	1	3	2
Aplicación de las condiciones iniciales.	1	3	2
Aplicación del método de resolución.	1	2	3
Implementación de Ecuaciones / Componentes.	1	2	3
Simplicidad de conexión de componentes.	1	2	2
Simplicidad de implementación de gráficos.	1	3	2
Simplicidad de modificación de gráficos.	1	3	2
Multiplicidad de opciones de gráficos.	3	1	1

Tabla 3: Comparación de los tres métodos.

Se procedió a comparar la norma de vectores, con respecto a Matlab, de las respuestas temporales de la posición, entre Matlab - Simulink y Matlab - Simscape, siendo en los dos casos un valor de $1,6594,10^{-5}$.

Se obtuvo la respuesta temporal de los modelos considerando como entrada un desplazamiento de amplitud 0.1 [m], las cuales son presentadas en las Fig. (5), Fig. (6) y (7).

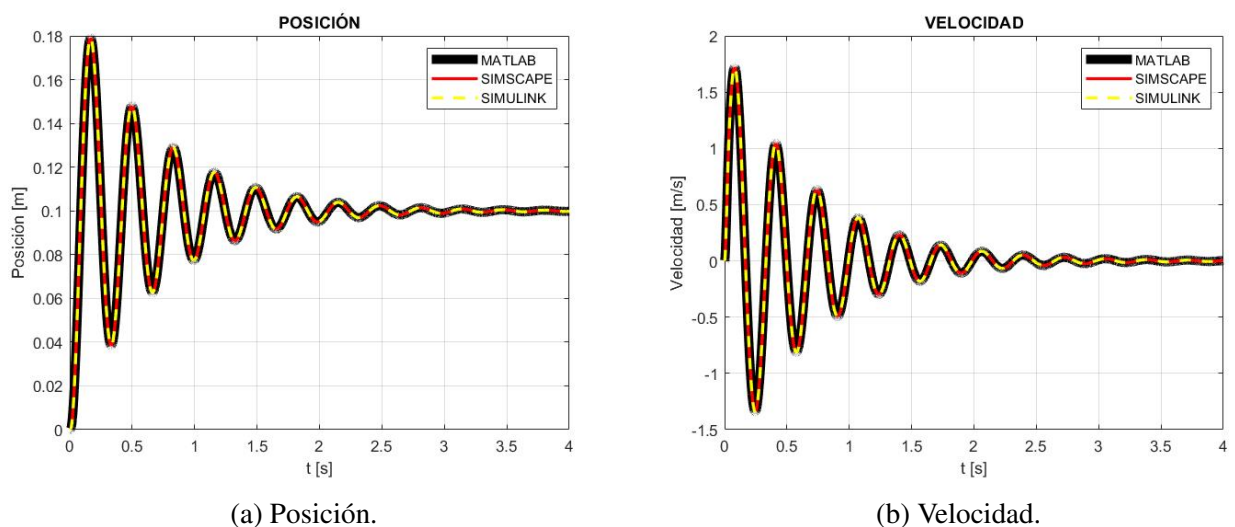


Figura 5: Respuesta temporal posición y velocidad de la masa suspendida (Ms).

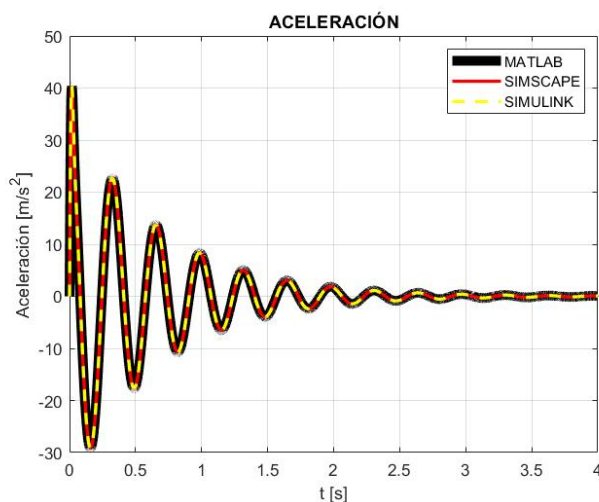


Figura 6: Respuesta temporal de la aceleración de la masa suspendida (Ms)

5. CONCLUSIONES.

En cuanto a las respuestas temporales, se puede ver claramente que la posición, Fig. (5), la velocidad, Fig. (6) y la aceleración en la Fig. (7) de la masa suspendida, son iguales para los tres métodos aplicados, para confirmar esto se obtiene la diferencia de la norma, la cual consta en la tab. (3) permitiendo afirmar que cualquiera de ellos es óptimo para la resolución de este tipo de problema, con un error prácticamente despreciable.

En cuanto a los tiempos de simulación podemos concluir que para una PC de 16GB de memoria RAM y con un procesador INTEL i5 de séptima generación, se obtuvo en Matlab un tiempo de procesamiento de aproximadamente diez veces menor con respecto a Simulink y de aproximadamente veinte veces menor con respecto a Simscape según lo visto Tab. (3).

Como se puede apreciar en Tab. (3), los Toolbox empleados: Simscape y Simulink son de mayor simplicidad que utilizando Matlab para el uso de estudiantes y profesionales con la finalidad de introducirlos en el campo de la simulación. Sin embargo, se hace necesario resaltar que a medida que se eliminan hipótesis y se simplifican los modelos, estos adquieren tal complejidad que hace que sea inevitable la combinación de las tres herramientas aquí utilizadas.

6. REFERENCIAS.

- Chapra S.C. y Canale R.P. *Métodos numéricos para ingenieros*. McGraw-Hill, 2007.
- Jazar R.N. *Vehicle roll dynamics*. En *Vehicle Dynamics: Theory and Application*, páginas 665–725. Springer, 2008.
- Miller S. y Wendlandt J. *Real-time simulation of physical systems using simscape*. MATLAB. News and Notes, páginas 1–13, 2010.
- Sharma P., Saluja N., Saini D., y Saini P. *Analysis of automotive passive suspension system with matlab program generation*. International Journal of Advancements in Technology, 4 (2), 2013.