MODELADO Y SIMULACION DE SISTEMAS

Javier HOLGADO
Universidad de Cádiz

1. INTRODUCCIÓN

Cuando se analizan los términos *Modelado* y *Simulación*, inmediatamente nos viene la imagen de un mundo ficticio donde se pueden realizar gran cantidad de operaciones, que poco o nada tienen que ver con la realidad. Esto va dejando de ser cierto a una velocidad increíble, y cada vez más, la simulación está siendo la antesala del mundo real.

Si el terreno comercial y publicitario nos vende la simulación como una especie de juego en el mundo de la multimedia y la realidad virtual, el hecho es que detrás de todo ello, hay un extraordinario trabajo que muchos científicos y técnicos que desarrollan su actividad en los campos tecnológicos, han realizado y realizan día a día para acercar los estudios teóricos al campo real a través del desarrollo de simulaciones.

La simulación es el paso previo (paso obligado por supuesto, en numerosas ocasiones) para conocer el funcionamiento de cualquier sistema y efectuar las modificaciones precisas para que evolucione según el diseño que se pretende realizar. Este tipo de trabajo se realiza con los computadores, evitando así el elevado coste que supondría la construcción física de dicho sistema, máxime si se llega a la conclusión final de que la alternativa a estudiar es inviable.

Para que el resultado de una simulación sea lo más ajustado posible a la realidad, el modelo debe ceñirse al sistema con la mayor exactitud. Este tipo de aproximaciones se realiza mediante ecuaciones diferenciales o ecuaciones en diferencia, lo que conduce a complejos análisis matemáticos, que han determinado que tanto el modelado como la simulación sean actualmente el eje central de prácticamente todas las

ramas de las Ciencias y la Ingeniería. El conocimiento que podemos así obtener de los sistemas físicos, nos permite actuar de forma más precisa, ya que se obtiene información sobre la evolución de determinadas situaciones.

Aunque parezca que la simulación es un camino glorioso, no hay que olvidar los defectos que tiene, comenzando por la excesiva facilidad con que se suelen analizar muchas situaciones por personas carentes de la adecuada experiencia y finalizando con la alta capacidad de abstracción, asociada con un elevado grado de satisfacción al fabricar una nueva versión del mundo dentro del computador, donde no existe la polución. Este es un error muy grave y cometido en exceso. El objetivo de la simulación no debe ser olvidarse del mundo real, sino todo lo contrario, para poder sacar conclusiones, efectuar análisis y tomar decisiones cuando sea preciso.

2. MODELADO DE SISTEMAS FISICOS

Con este trabajo se intenta hacer una breve descripción de algunos campos de investigación, en los que los actuales lenguajes de simulación CSSL (Computer System Simulation Languages) centran sus desarrollos y aplicaciones. Circuitos eléctricos y electrónicos, sistemas mecánicos, ecuaciones diferenciales, máquinas eléctricas, análisis de problemas complejos de termodinámica, biomedicina, química, etc., pueden ser analizados desde un símil realístico más o menos idealizado. Pueden ser sistemas continuos, discretos, mixtos, tanto desde el punto de vista unidimensional como de aplicaciones multivariables.

Análisis gráficos bidimensionales o tridimensionales, son cada vez más frecuentes en el estudio del comportamiento de los sistemas físicos, dando perspectivas de análisis totalmente diferentes a los estudios clásicos.

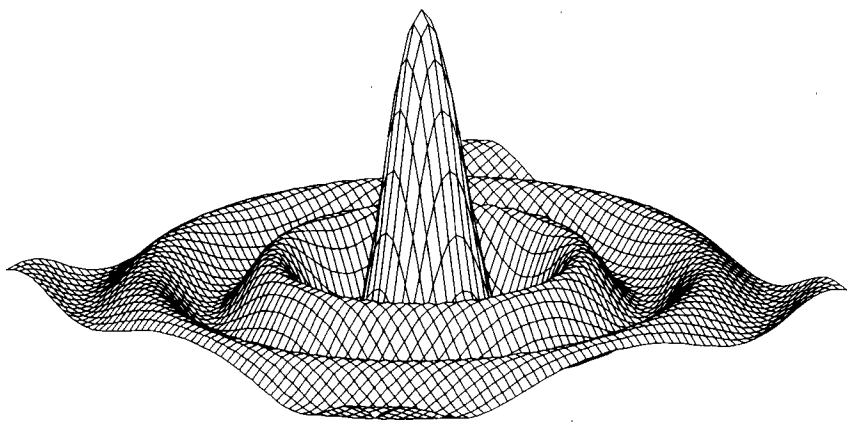

Fig. 1. Superficie tridimensional de una función senoidal

El software para la simulación de sistemas continuos fue estandarizado en 1967 (Augustin et alii, 1967).El comité estándar que lo realizó estaba constituido por Ingenieros de control, lo cuál indicaría que los lenguajes de simulación para su aplicación a sistemas continuos estaría adecuadamente confeccionado para el modelado y simulación de sistemas. Sin embargo, esto no es totalmente cierto.

El problema hasta ahora ha radicado en que mientras que se ha habilitado un software específico para simular numerosas aplicaciones con circuitos electrónicos y sistemas mecánicos, no existía el material adecuado para aplicaciones de control y por tanto, para su posterior aplicación industrial. La razón de esta circunstancia es el hecho de que los sistemas de control no sólo contienen un controlador, sino también una planta. Ello obliga a simular elementos de diferente naturaleza y complejidad.

Los "trucos" que se han empleado (y aún se emplean) para solventar algunos problemas surgidos en la simulación, como el análisis de discontinuidades, el uso de modelos dinámicos inversos, con la linealización de realimentación, la elevación del índice de un sistema al existir elementos que almacenan energía, etc., han obligado a redefinir los objetivos de los lenguajes de simulación (CSSL) y estructurar nuevamente su concepción y enfoque, de tal manera que se solventen dichos problemas para poder aplicar a sistemas más complejos.

Otro objetivo importante es la posterior aplicación de la simulación al control en tiempo real de algunos sistemas. Hasta ahora se dividía el análisis de un sistema en dos partes claramente diferenciadas, la correspondiente al análisis idealizado en el ordenador y la más complicada operación de buscar su trasvase al mundo real. El nuevo enfoque debe permitir que el paso del modelo simulado al control en tiempo real sea efectivo y con el mínimo coste y esfuerzo. El mismo algoritmo de control diseñado para el simulador, deberá ser ejecutado en el sistema real. Ello muestra claramente la especial complementariedad entre el modelado y la simulación y el control en tiempo real. Actualmente existen algunos lenguajes de simulación que empiezan sutilmente a ofrecer tal potencialidad (Otter et alii, 1993), lo cuál abre nuevas vías a la investigación y a las aplicaciones industriales.

Un complemento muy importante a ambas líneas de trabajo, es el diseño de controladores borrosos. Las nuevas perspectivas que dichos controladores han abierto desde las primeras introducciones borrosas de Zadeh en 1967 y las posteriores aplicaciones prácticas de Mandani, permiten que los complejos sistemas no lineales puedan ser estabilizados mediante razonamiento cualitativo, cubriendo la laguna de diseño en los controladores convencionales.

Entre los numerosos lenguajes de modelado y simulación, que como se ha indicado anteriormente, coexisten desde 1967, destacan con un gran peso específico MATLAB y MATHEMATICA, los casi estándares del cálculo numérico y SIMULINK, lenguaje particular de simulación para MATLAB. Sin embargo, los auténticos programas de modelado y simulación orientado a objetos, que resuelvan sistemas de ecuaciones diferenciales DAE y ODE, tienen otro enfoque diferente del propuesto por dichos programas de cálculo numérico. Paquetes y lenguajes como ACSL, DYMOLA, DSBLOCK, OMOLA, DESIRE, CTRL-C, DARE-P, etc., ofrecen tanto la posibilidad de desarrollar los cálculos numéricos necesarios, como la resolución de

sistemas de ecuaciones diferenciales, introduciendo diferentes algoritmos de integración, etc., a base de compiladores en FORTRAN, C, DSBLOCK, etc.

Lo realmente importante en gran parte de los casos, es la posibilidad de trasvasar al mundo real los conocimientos adquiridos en el modelado y la simulación, y sobre todo, la apertura del simulador a numerosos campos de aplicación, no sólo a temas específicos de electrónica o mecánica. Prueba de ello son las situaciones que se analizan en este artículo.

3. EJEMPLOS

Se analizan a continuación algunas aplicaciones específicas en el campo del modelado, la simulación y su aplicación al campo real, ofreciendo un amplio abanico de posibilidades, que demuestran el enorme alcance de este campo tecnológico. Se han empleado básicamente los lenguajes y programas ACSL, DYMOLA, FORTRAN y C++.

Ejemplo 1. Sea una ecuación diferencial genérica como la siguiente:

$$\frac{d^4y}{dt^4} + 4\frac{d^3y}{dt^3} + 6\frac{d^2y}{dt^2} + 4\frac{dy}{dt} + y = t^3 e^{-t}$$

El modelado de este sistema en un lenguaje como Dymola, conduce al siguiente desarrollo, donde **y** es la salida:

```
model ecdif20
  output y
  local a, z1, z2, z3
  a=(Time**3)*exp(-Time)
  der(z3)=-4*z3-6*z2-4*z1-y+a
  der(y)=z1
  der(z1)=z2
  der(z2)=z3
end
```

Su modelo en lenguaje ACSL y posteriormente su compilación en FORTRAN, permiten realizar la simulación completa del mismo:

```
! ACSL model generated by Dymola.
PROGRAM ecdif20
```

```
 VARIABLE Time, StartTime=0.0
 CONSTANT StopTime=1.0
 DYNAMIC
  DERIVATIVE der
   PROCEDURAL
   a = Time**3.0*exp(- Time)
   derz3 = a - (4.0*z3 + 6.0*z2 + 4.0*z1 + y)
! ELIMINATED STATE DERIVATIVES AND OUTPUTS
   dery = z1
   derz1 = z2
   derz2 = z3
 EXIT .. CONTINUE
   END ! of PROCEDURAL
   CONSTANT initz3=0
   z3 = INTEG(derz3, initz3)
   CONSTANT inity=0
   y = INTEG(dery, inity)
   CONSTANT initz1=0
   z1 = INTEG(derz1, initz1)
   CONSTANT initz2=0
   z2 = INTEG(derz2, initz2)
  END ! of DERIVATIVE
  TERMT (Time .GE. StopTime)
 END ! of DYNAMIC
END ! of PROGRAM
```

El valor de la salida simulada para un tiempo máximo $t^{max}=30^{seg}$, conduce a los siguientes valores numéricos y gráficos:

```
ACSL Runtime Exec Version 6  Level 10FX1  Aug  3 21:06:28 1995
 set stoptime=30
 output time, y
 start
      TIME 0.00000000      Y 0.00000000
      TIME 1.50000000      Y 0.00453856
      TIME 3.00000000      Y 0.12962400
      TIME 4.50000000      Y 0.49417800
      TIME 6.00000000      Y 0.82606200
      TIME 7.50000000      Y 0.87890300
      TIME 9.00000000      Y 0.70269700
      TIME 10.5000000      Y 0.46126800
      TIME 12.0000000      Y 0.26209300
      TIME 13.5000000      Y 0.13337700
      TIME 15.0000000      Y 0.06222180
      TIME 16.5000000      Y 0.02705510
```

```
TIME 18.0000000      Y 0.01110010
TIME 19.5000000      Y 0.00433731
TIME 21.0000000      Y 0.00162582
TIME 22.5000000      Y 5.8799E-04
TIME 24.0000000      Y 2.0613E-04
TIME 25.5000000      Y 7.0306E-05
TIME 27.0000000      Y 2.3405E-05
TIME 28.5000000      Y 7.6250E-06
TIME 30.0000000      Y 2.4363E-06
quit
```

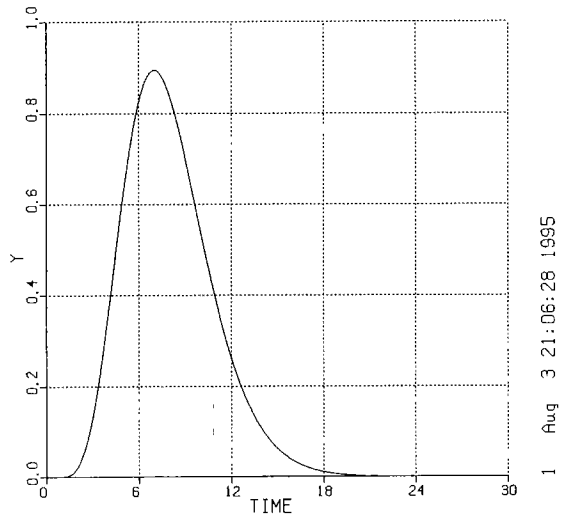

Fig. 2. Respuesta temporal de la ecuación diferencial

Ejemplo 2. Un circuito electrónico bastante conocido, es el del rectificador de media onda, con carga resistiva-capacitiva y señal de entrada senoidal:

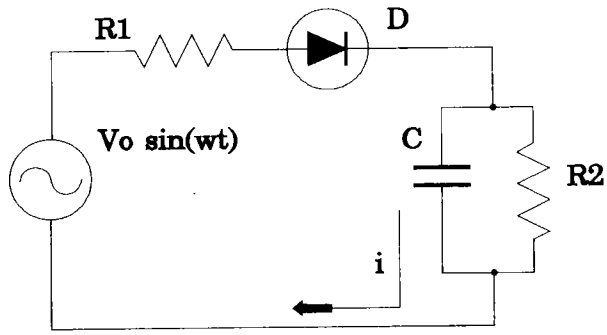

Fig. 3. Circuito rectificador de media onda

Su modelo en Dymola, vendría expresado a través de:

```
model circ4
  submodel (power) Uo
  submodel (resistor) R1(R=10.0), R2(R=30.0)
  submodel (capacitor) C1(C=0.001)
  submodel (diode) D1
  submodel common
  output y
  parameter f=50.0
  connect common - \Uo - R1 - D1 - (C1//R2) - common
  Uo.Vo=sin(2.0*3.14159*f*Time)
  y=C1.v
end
```

Las ondas más representativas del circuito, se representan en la figura 4, donde $\mathbf{V_0}$ es la tensión senoidal de entrada, $\mathbf{Y_1}$ la corriente suministrada por dicha fuente de alimentación (media onda) e $\mathbf{Y_2}$ la carga del condensador (el clásico diente de sierra).

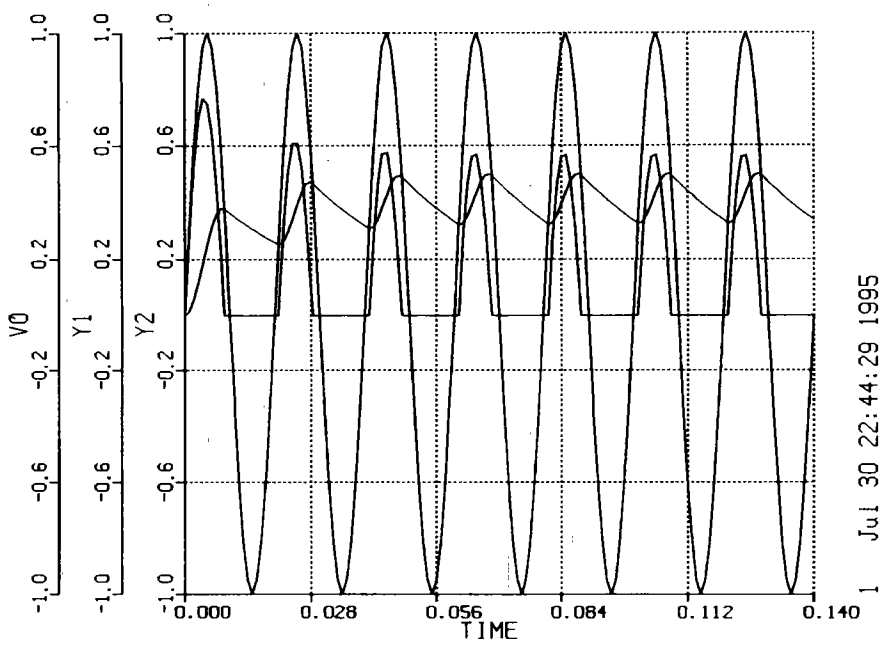

Fig. 4. Ondas de tensión y corriente del rectificador

Ejemplo 3. Sistema mecánico al que se le aplica una fuerza **F** exterior, existiendo rozamiento entre el bloque y el suelo, según el esquema general y desgloses individuales de la figura 5.

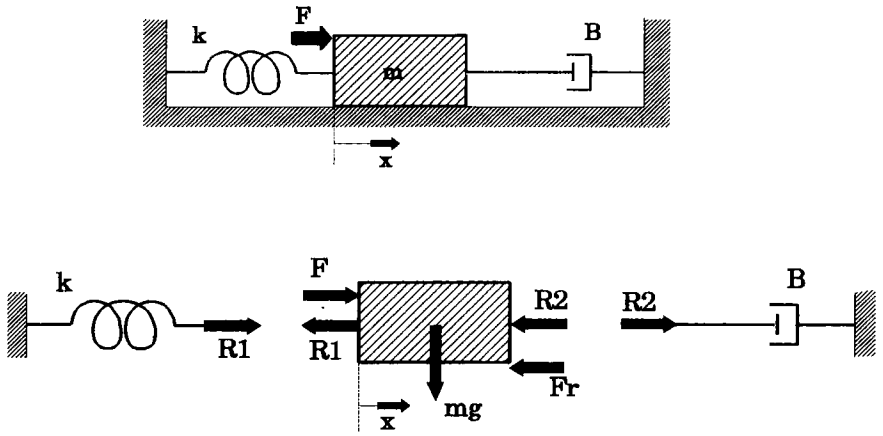

Fig. 5. Sistema mecánico resorte/masa/rozamiento

Su expresión en forma de espacio de estado conduce a:

$$R_1 = kx$$

$$R_2 = B\dot{x}$$

$$m\ddot{x} = F - F_r - R_1 - R_2$$

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -\frac{k}{m} & -\frac{B}{m} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{1}{m} \end{bmatrix} F + \begin{bmatrix} 0 \\ -\mu g \end{bmatrix}$$

$$y = [1 \quad 0] \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

La programación en ACSL de este sistema de ecuaciones, determina una estructura como la siguiente:

```
PROGRAM meca2
 VARIABLE Time, StartTime=0.0
 CONSTANT StopTime=1.0
 INITIAL
 CONSTANT F=0, k=2.0
 CONSTANT B=0.01, mu=0.01
 CONSTANT m=0.5, g=9.8
 END ! of INITIAL
 DYNAMIC
  DERIVATIVE der
   PROCEDURAL
! SORTED AND SOLVED EQUATIONS
!  meca2.
    R1 = k*x
    R2 = B*x1
    if (x1 .LT. 0.0) then
     Q101 = 1.0
    else
     Q101 = - 1.0
    end if
    sigmu = Q101
    FR = mu*m*g*sigmu
    derx1 = (F + FR - R1 - R2)/m
! END OF SORTED AND SOLVED EQUATIONS
! ELIMINATED STATE DERIVATIVES AND OUTPUTS
    derx = x1
 EXIT .. CONTINUE
    END ! of PROCEDURAL
    CONSTANT initx1=0
    x1 = INTEG(derx1, initx1)
    CONSTANT initx=0
    x = INTEG(derx, initx)
  END ! of DERIVATIVE
  TERMT (Time .GE. StopTime)
 END ! of DYNAMIC
END ! of PROGRAM
```

El resultado de la simulación se muestra en la figura 6, donde se comprueba que el efecto oscilatorio del resorte queda amortiguado por efecto del rozamiento entre el bloque y el suelo. Si no existiera dicho rozamiento, el sistema sería ideal y se mantendría oscilando ininterrumpidamente.

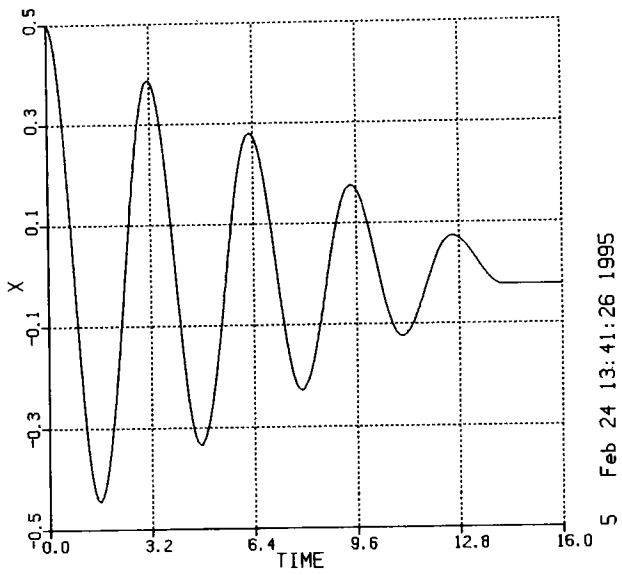

Fig. 6. Trayectoria del bloque (posición)

Ejemplo 4. Un motor de corriente continua, al que se le acopla una carga en forma de rozamiento viscoso e inercia.

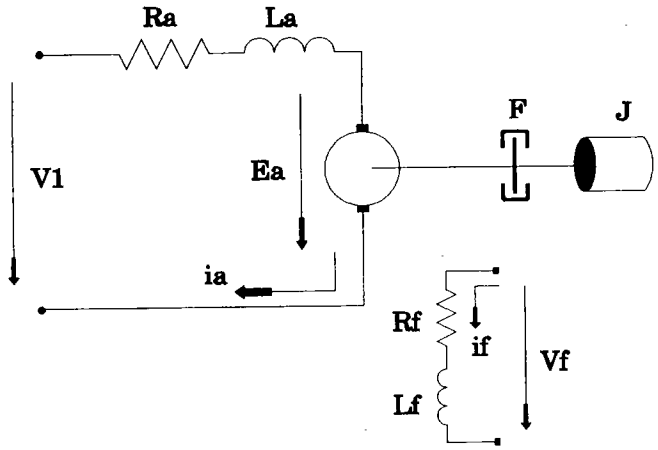

Fig. 7. Motor de corriente continua

Designando como i_f la intensidad de campo constante, **T** el par motor total y T_L el par de pérdidas en la carga:

$$T = J\ddot{\theta} + F\dot{\theta} + T_L \quad (T_L \simeq 0)$$

$$T - T_L = J\ddot{\theta} + F\dot{\theta}$$

$$T = ki_a = k_1 k_2 i_f i_a$$

$$L_a \frac{di_a}{dt} + i_a R_a = V_1 - E_a = V_1 - k_\omega \dot{\theta}$$

$$\frac{\theta}{T - T_L} = \frac{1}{Js^2 + Fs}$$

$$i_a = \frac{1}{R_a + sL_a}\left(V_1 - sk_\omega \theta\right)$$

$$Si\ L_a = 0 \quad \rightarrow \quad i_a = \frac{1}{R_a}\left(V_1 - k_\omega \dot{\theta}\right)$$

$$Si\ T_L \simeq 0 \quad \rightarrow \quad \frac{\theta}{V_I} = \frac{k/s}{JL_a s^2 + s(JR_a + FL_a) + (FR_a + kk_\omega)}$$

$$Si\ T_L \simeq 0 \quad L_a \simeq 0 \quad \rightarrow \quad \frac{\theta}{V_I} = \frac{k/s}{sJR_a + (FR_a + kk_\omega)}$$

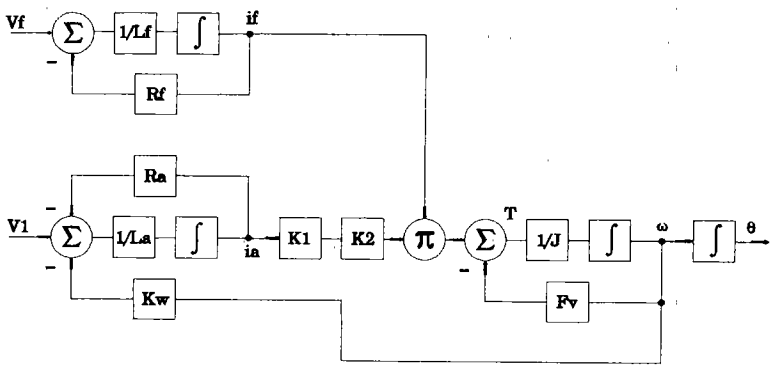

Fig. 8. Diagrama elemental de bloques

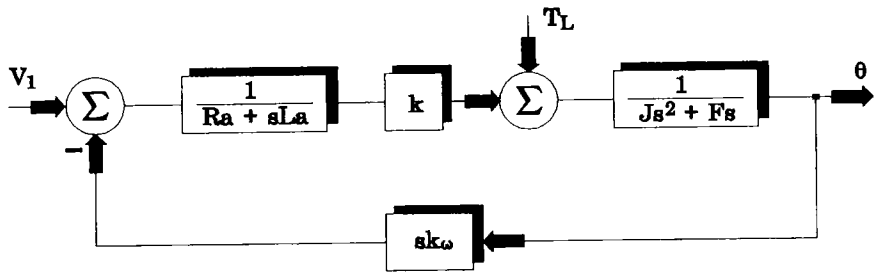

Fig. 9. Diagrama de bloques (Sistema realimentado)

Su modelado en ACSL es el siguiente:

```
PROGRAM motor1
 VARIABLE Time, StartTime=0.0
 CONSTANT StopTime=1.0
 INITIAL
 CONSTANT Ra=10, La=0.001
 CONSTANT Rf=10, Lf=0.001
 CONSTANT k=0.5, J=0.02
 CONSTANT F=0.0001, kom=0.5
 CONSTANT v1=15, vf=15
 END ! of INITIAL
 DYNAMIC
  DERIVATIVE der
   PROCEDURAL
! SORTED AND SOLVED EQUATIONS
!  motor1.
    ea = kom*om
    T = k*if*ia
    deria = (v1 - Ra*ia - ea)/La
    derom = (T - F*om)/J
    derif = (vf - Rf*if)/Lf
! END OF SORTED AND SOLVED EQUATIONS
! ELIMINATED STATE DERIVATIVES AND OUTPUTS
    derth = om
 EXIT .. CONTINUE
    END ! of PROCEDURAL
    CONSTANT initia=0
    ia = INTEG(deria, initia)
    CONSTANT initom=0
    om = INTEG(derom, initom)
```

```
    CONSTANT initif=0
    if = INTEG(derif, initif)
    CONSTANT initth=0
    th = INTEG(derth, initth)
  END ! of DERIVATIVE
  TERMT (Time .GE. StopTime)
 END ! of DYNAMIC
END ! of PROGRAM
```

La simulación determina una velocidad angular **OM** y un par de salida total **T**, según se describe en la figura 10.

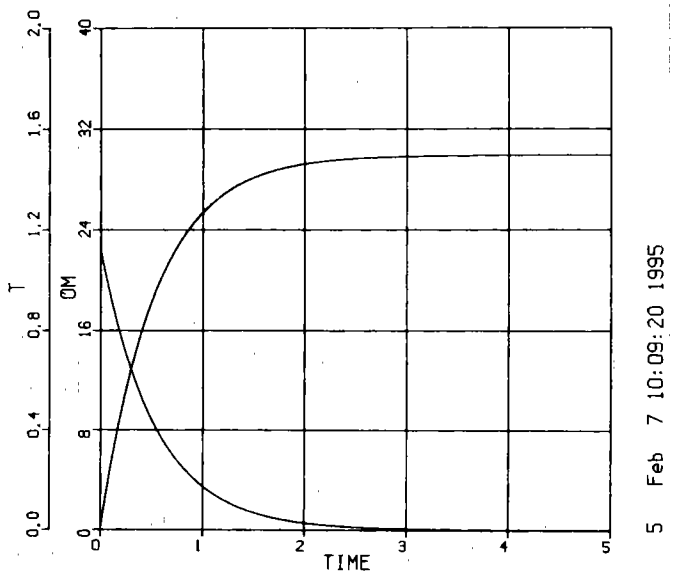

Fig. 10. Salida del motor de corriente continua

Ejemplo 5. Péndulo invertido colocado sobre un vehículo móvil.

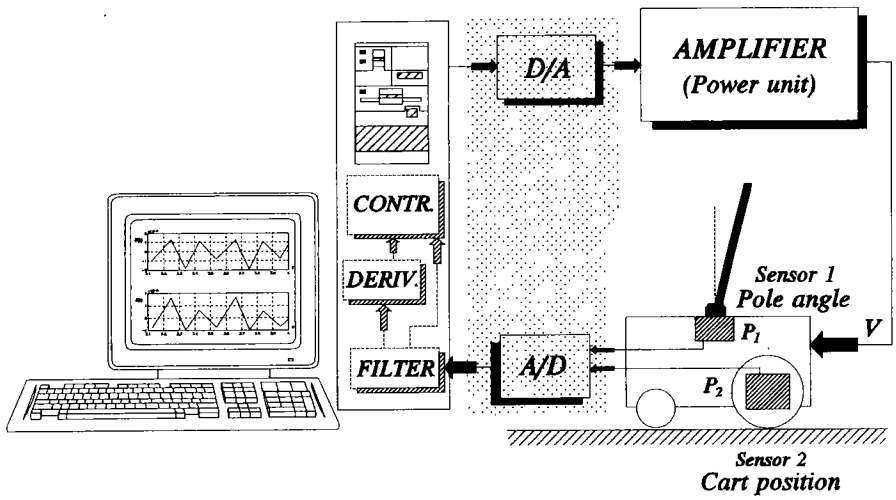

Fig. 11. Péndulo invertido

El modelado y la simulación de este sistema, ha sido un ejemplo típico de trasvase de conocimientos al mundo real. Una vez obtenidos los resultados de la simulación, se diseñó el controlador real necesario para que se pudiera estabilizar un equipo de laboratorio.

Al ser un sistema totalmente inestable, en el momento en que la varilla inicia el movimiento de caída, el potenciómetro $\mathbf{P_1}$ suministra información al controlador del sistema a través del convertidor analógico-digital, siendo dicha señal filtrada para eliminar perturbaciones. El controlador actúa con tensión **V** sobre el motor del vehículo, desplazándose en el mismo sentido que la caída de la varilla y contrarrestando así dicho desplazamiento respecto de la vertical. El vehículo, por tanto, estará siempre en continuo movimiento a derecha e izquierda para que la varilla se mantenga vertical.

Uno de los trazados obtenidos durante la simulación, que han servido para el posterior control en tiempo real, se muestra en la figura 12.

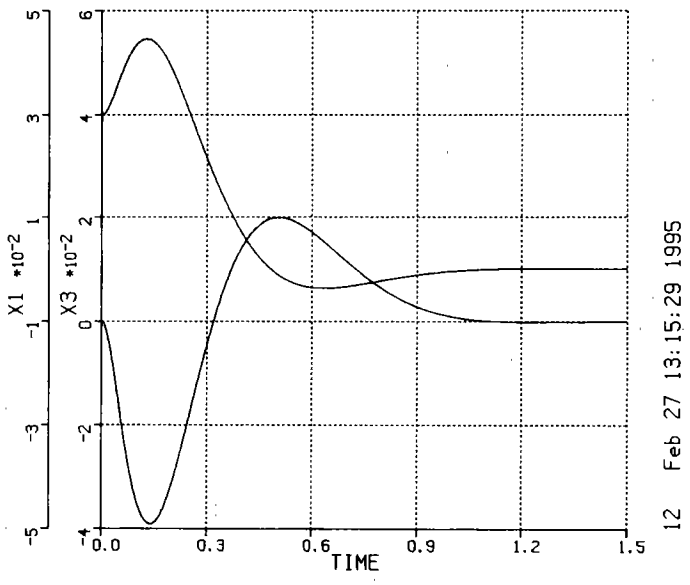

Fig. 12. Trayectoria del péndulo y velocidad angular

Ejemplo 6. Bidón que se lanza al océano desde un barco. Este puede ser un prototipo de situaciones genéricas que se desea modelar y simular. Cuando desde un barco se lanzan bidones al océano, máxime si se manipulan sustancias tóxicas o peligrosas, conviene analizar y simular con detalle el movimiento de los mismos, así como la fuerza con que golpeará el fondo marino, para estudiar una posible rotura.

Un programa de modelado en ACSL, teniendo en cuenta el empuje que ejerce el agua sobre el bidón, así como el efecto del rozamiento, podría ser de la siguiente forma:

```
PROGRAM bidon
 VARIABLE Time, StartTime=0.0
 CONSTANT StopTime=1.0
 INITIAL
 CONSTANT k=0.08, m1=16.38
 CONSTANT g=32.2, v=7.35
 CONSTANT m2=63.99
 END ! of INITIAL
 DYNAMIC
  DERIVATIVE der
   PROCEDURAL
   e = k*vy
```

```
    r = m2*v
    dervy = (m1*g - r - e)/m1
! ELIMINATED STATE DERIVATIVES AND OUTPUTS
    dery = vy
 EXIT .. CONTINUE
    END ! of PROCEDURAL
    CONSTANT initvy=0
    vy = INTEG(dervy, initvy)
    CONSTANT inity=0
    y = INTEG(dery, inity)
   END ! of DERIVATIVE
   TERMT (Time .GE. StopTime)
  END ! of DYNAMIC
END ! of PROGRAM
```

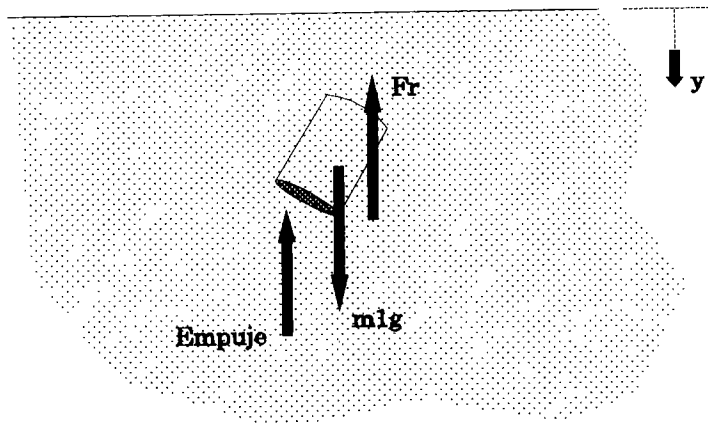

Fig. 13. Bidón lanzado al océano

Su simulación con un compilador FORTRAN, proporciona los resultados de la figura 13, de los que se extraen algunos datos parciales sobre la posición vertical (en metros) y la velocidad del bidón (en m/seg):

```
ACSL Runtime Exec Version 6  Level 10FX1  Aug  5 17:56:10 1995
 set stoptime=15
 output time, y, vy
 start
      TIME 0.00000000      Y 0.00000000      VY 0.00000000
      TIME 0.60000000      Y 0.62696400      VY 2.08886000
```

```
TIME 1.20000000    Y 2.50541000    VY 4.17161000
TIME 1.80000000    Y 5.63167000    VY 6.24826000
TIME 2.40000000    Y 10.0021000    VY 8.31884000
TIME 3.00000000    Y 15.6131000    VY 10.3834000
TIME 3.60000000    Y 22.4609000    VY 12.4418000
TIME 4.20000000    Y 30.5421000    VY 14.4943000
TIME 4.80000000    Y 39.8529000    VY 16.5407000
TIME 5.40000000    Y 50.3897000    VY 18.5812000
TIME 6.00000000    Y 62.1491000    VY 20.6157000
TIME 6.60000000    Y 75.1274000    VY 22.6442000
TIME 7.20000000    Y 89.3210000    VY 24.6668000
TIME 7.80000000    Y 104.726000    VY 26.6835000
TIME 8.40000000    Y 121.340000    VY 28.6943000
TIME 9.00000000    Y 139.158000    VY 30.6992000
TIME 9.60000000    Y 158.178000    VY 32.6982000
TIME 10.2000000    Y 178.395000    VY 34.6914000
TIME 10.8000000    Y 199.806000    VY 36.6787000
TIME 11.4000000    Y 222.408000    VY 38.6603000
TIME 12.0000000    Y 246.197000    VY 40.6360000
quit
```

Con estos resultados se puede decidir si la velocidad que alcanzarían los bidones según la profundidad del mar, es aceptable desde el punto de vista de la resistencia física al impacto.

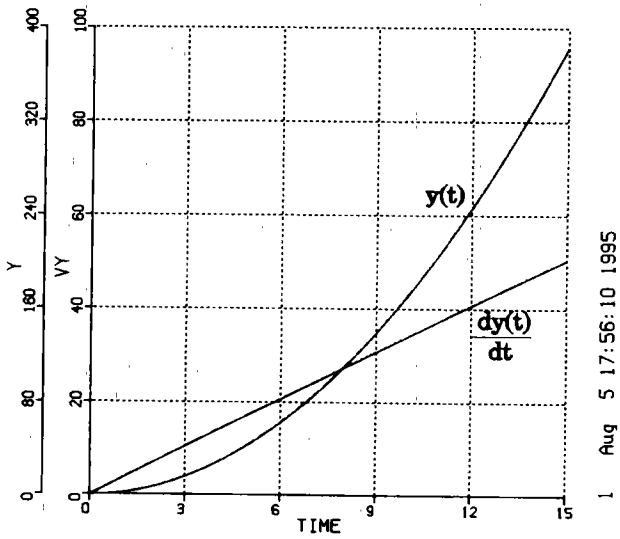

Fig. 14. Relación entre posición (Y) y velocidad (VY)

4. CONCLUSIONES

El proceso de modelado tiene dos vertientes sumamente interesantes como se ha podido demostrar. De una parte, la continua mejora de los elementos de simulación, tan ampliamente utilizados hoy día, permitirán resolver las incertidumbres y problemas que existen actualmente para solventar numerosos estudios. Por otro lado, la posibilidad de obtener aplicaciones para el contexto industrial, ya que de la simulación se pasará al control real, salvando los obstáculos que existen entre el mundo simulado y el real.

El creciente interés en esta tecnología y el interés general en resolver los problemas de aplicación que habitualmente se encuentran, hacen que existan numerosos objetivos a cubrir, con un enorme potencial, tales como el diseño de los algoritmos necesarios para los lenguajes de simulación, el estudio de la robustez que proporcionan los controladores, especialmente los borrosos, la adaptación de los resultados anteriores al diseño de controladores en tiempo real y su aplicación a casos concretos de sistemas físicos, que harán su traslado y aplicación a otros sistemas y al mundo real, enormemente útiles para resolver problemas de estabilidad.

5. REFERENCIAS

M. ANDERSON, 1992. Discrete event modeling and simulation in Omola. IEEE Computer Aided control system design, USA, 262-268.

D.C. AUGUSTIN, M.S. FINEBERG, ET ALII, 1967. The SCi Continuous System Simulation Language (CSSL). Journal of Simulation, 9: 281-303.

F.E. CELLIER, 1991. Continuous system modeling. Springer Verlag, New York.

H. ELMQVIST ET ALII, 1993. Object-oriented modeling of hybrid systems. European simulation conference, The Netherlands, 31-39.

J. HOLGADO, 1995. Controladores en lógica borrosa. Servicio de Publicaciones de la UCA.

J. HOLGADO, 1995. Estudios monográficos en automática avanzada. Servicio de Publicaciones de la UCA.

J. HOLGADO, J. ARACIL, 1994. Increasing stability by rule modification procedure. IPMU Conference, Francia.

J. HOLGADO, J. ARACIL, A. OLLERO, 1994. Statistical study on stability indices of fuzzy control systems. IEEE World Congress on Computational Intelligence, USA.

MITCHELL AND GAUTIER ASSOC. 1991. Advanced continuous simulation language. Concord, MA, USA.

M. OTTER, 1992. DSBlock: A neutral description of dynamic systems. Institute for robotics and system dynamics, Wessling, Germany, TR-R81-92.

R.C. SMITH ET ALII, 1990. Dynamic analysis and design system. Multibody systems handbook, Springer Verlag, Berlin.