MÉTODOS DE SIMULACIÓN Y MODELADO

Solución al Trabajo Práctico - Enero de 2023

EJERCICIO 1

Lea el artículo citado a continuación, que puede descargar de la página web de la asignatura, y conteste a las preguntas.

> Åström, K.J., Elmqvist, H., Mattsson, S.E. *Evolution of continuous-time modeling and simulation*. The 12^{th} European Simulation Multiconference, ESM'98, June 16–19, 1998, Manchester, UK.

1. ¿Qué analogías pueden establecerse entre el modelado basado en diagramas de bloques y el paradigma de la simulación analógica?
2. ¿Qué es el paradigma de modelado físico? ¿Qué tipo de modelos matemáticos se obtienen de aplicar el paradigma del modelado físico?
3. ¿Qué diferencias hay entre el paradigma de la simulación analógica y el paradigma del modelado físico?
4. Observe la Figura 1.1 y comente su contenido basándose en el artículo.

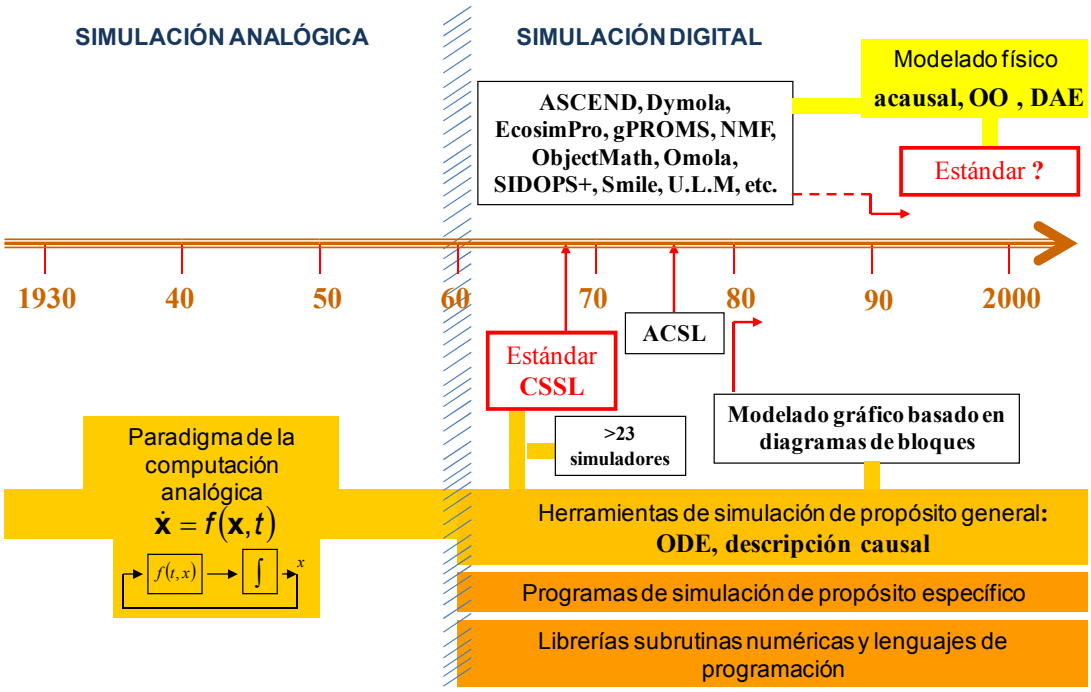


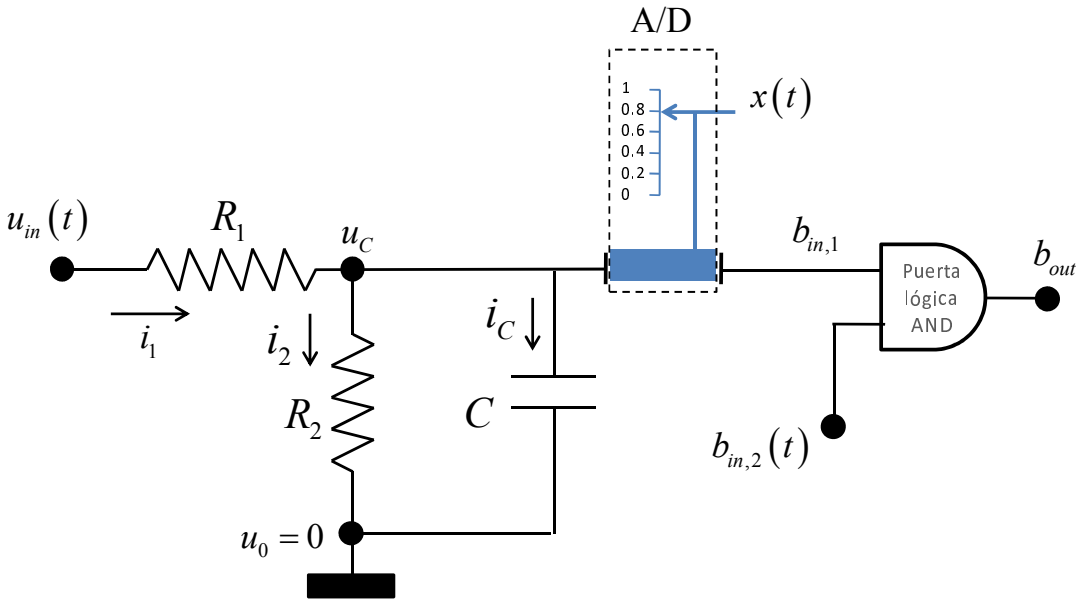
Figura 1.1: Evolución del modelado y simulación de tiempo continuo.

Solución al Ejercicio 1

En este ejercicio el alumno debe contestar, con sus propias palabras, a las cuestiones planteadas.

EJERCICIO 2

El circuito mostrado en la figura está compuesto por dos resistencias de valor constante R_1 y R_2; un condensador de valor constante C; un conversor analógico digital (A/D) que podemos regular moviendo la posición x de su cursor; y una puerta lógica AND.



En la figura se muestra el nombre asignado al voltaje en los nodos de la parte analógica del circuito (u_{in}, u_C), y a la corriente que circula a través de las resistencias (i_1, i_2) y el condensador (i_C). Suponemos que la corriente que entra al conversor A/D es cero. Es decir, $i_1 = i_2 + i_C$. El voltaje u_{in} es una función conocida del tiempo.

El conversor A/D está regulado mediante un cursor. La posición x del cursor, que puede tomar valores en el intervalo $[0, 1]$, es una función conocida del tiempo.

El funcionamiento del conversor A/D se modela de la forma siguiente. La salida del conversor A/D es una señal Booleana que depende del voltaje en su entrada, de la posición del cursor y del parámetro $U_{A/D}$. Llamando u al voltaje de entrada al conversor, x a la posición del cursor, y b a su señal de salida, la relación constitutiva del conversor es la siguiente:

$$b = \begin{cases} true & \text{si } \frac{u}{U_{A/D}} > x \\ false & \text{en caso contrario} \end{cases}$$

Tal como se muestra en la figura, en el circuito que vamos a simular hemos llamado u_c al voltaje de entrada al conversor y $b_{in,1}$ a su señal de salida. La salida del conversor es una de las entradas de la puerta lógica AND.

La puerta lógica AND tiene dos entradas ($b_{in,1}$, $b_{in,2}$), y una salida (b_{out}). Las tres son señales Booleanas. La señal de entrada $b_{in,2}$ es una función conocida del tiempo.

Se pretende simular el circuito mostrado en la figura durante 240 s, para los siguientes valores de u_{in} (expresado en voltios), $b_{in,2}$ y x:

$$
\begin{aligned}
u_{in} &= 25 + 20 \cdot \cos\left(\frac{\pi}{30} \cdot t\right) + 5 \cdot \cos\left(\frac{\pi}{2} \cdot t + \frac{\pi}{4}\right) \\
b_{in,2} &= \begin{cases} true & \text{si } t \in [30, 90] \cup [150, 210] \\ false & \text{en caso contrario} \end{cases} \\
x &= \begin{cases} 0.2 & \text{si } t < 80 \\ 0.5 & \text{si } 80 \leq t < 160 \\ 0.8 & \text{si } t \geq 160 \end{cases}
\end{aligned}
$$

donde el tiempo (t) está expresado en segundos y los parámetros valen: $U_{A/D} = 18\ V$, $R_1 = R_2 = 1000\ \Omega$, $C = 0.05$ F.

1. Escriba las ecuaciones del modelo del sistema. El modelo debe describir la evolución de las variables u_{in}, u_c, i_1, i_2, i_C, x, $b_{in,1}$, $b_{in,2}$, b_{out}.

2. Asigne la causalidad computacional. Indique cuántos grados de libertad tiene el modelo.

3. Escriba el diagrama de flujo del algoritmo para la simulación de este modelo. Emplee el método de integración de Euler explícito. La condición de finalización de la simulación es que el tiempo alcance el valor 240 s.

4. Programe el algoritmo anterior en lenguaje R y ejecute la simulación. Explique cómo ha escogido el tamaño del paso de integración. Represente gráficamente frente al tiempo las variables siguientes: u_{in}, u_c, i_1, i_2, i_C, x, $b_{in,1}$, $b_{in,2}$, b_{out}.

Solución al Ejercicio 2

Las ecuaciones del modelo se muestran a continuación. No se muestran las asignaciones de valor a las constantes y parámetros.

$$
\begin{aligned}
u_{in} &= 25 + 20 \cdot \cos\left(\frac{\pi}{30} \cdot t\right) + 5 \cdot \cos\left(\frac{\pi}{2} \cdot t + \frac{\pi}{4}\right) \\
b_{in,2} &= \begin{cases} true & \text{si } t \in [30, 90] \cup [150, 210] \\ false & \text{en caso contrario} \end{cases} \\
x &= \begin{cases} 0.2 & \text{si } t < 80 \\ 0.5 & \text{si } 80 \leq t < 160 \\ 0.8 & \text{si } t \geq 160 \end{cases} \\
u_{in} - u_c &= i_1 \cdot R_1 \\
u_c &= i_2 \cdot R_2 \\
C \cdot \frac{du_c}{dt} &= i_C \\
i_1 &= i_2 + i_C \\
b_{in,1} &= \begin{cases} true & \text{si } \frac{u_c}{U_{A/D}} > x \\ false & \text{en caso contrario} \end{cases} \\
b_{out} &= b_{in,1} \text{ and } b_{in,2}
\end{aligned}
$$

Asumiendo que la variable que aparece derivada pueden ser seleccionada como variable de estado, las variables del modelo pueden clasificarse de la forma siguiente:

– Parámetros y constantes: $U_{A/D}$, R_1, R_2, C
– Variables de estado: u_c
– Variables algebraicas: u_{in}, i_1, i_2, i_C, x, $b_{in,1}$, $b_{in,2}$, b_{out}

Para asignar la causalidad computacional al modelo, se sustituye la derivada de la variable de estado por una variable muda:

$$\frac{du_c}{dt} \rightarrow deru_c$$

Omitiendo las ecuaciones en las que se asigna valor a los parámetros y las constantes, se obtiene el modelo siguiente:

$$u_{in} = 25 + 20 \cdot \cos\left(\frac{\pi}{30} \cdot t\right) + 5 \cdot \cos\left(\frac{\pi}{2} \cdot t + \frac{\pi}{4}\right) \quad (1.1)$$

$$b_{in,2} = \begin{cases} true & \text{si } t \in [30, 90] \cup [150, 210] \\ false & \text{en caso contrario} \end{cases} \quad (1.2)$$

$$x = \begin{cases} 0.2 & \text{si } t < 80 \\ 0.5 & \text{si } 80 \leq t < 160 \\ 0.8 & \text{si } t \geq 160 \end{cases} \quad (1.3)$$

$$u_{in} - u_c = i_1 \cdot R_1 \quad (1.4)$$

$$u_c = i_2 \cdot R_2 \quad (1.5)$$

$$C \cdot deru_c = i_C \quad (1.6)$$

$$i_1 = i_2 + i_C \quad (1.7)$$

$$b_{in,1} = \begin{cases} true & \text{si } \frac{u_c}{U_{A/D}} > x \\ false & \text{en caso contrario} \end{cases} \quad (1.8)$$

$$b_{out} = b_{in,1} \text{ and } b_{in,2} \quad (1.9)$$

Las variables del modelo pueden clasificarse en conocidas (el tiempo, los parámetros, las constantes y las variables de estado) y desconocidas (las variables algebraicas y las derivadas de las variables de estado):

– Conocidas: t
$U_{A/D}$, R_1, R_2, C
u_c

– Desconocidas: u_{in}, i_1, i_2, i_C, x, $b_{in,1}$, $b_{in,2}$, b_{out}
$deru_c$

Con el fin de analizar si el modelo es *estructuralmente singular*, se comprueba que:

1. El número de ecuaciones y de variables desconocidas (incógnitas) es el mismo. Este modelo tiene 9 ecuaciones y 9 incógnitas.

2. Cada incógnita puede emparejarse con una ecuación en que aparezca y con la cual no se haya emparejado ya otra incógnita.

u_{in}	$\rightarrow$	Ec. (1.1),	$b_{in,2}$	$\rightarrow$	Ec. (1.2),	x	$\rightarrow$	Ec. (1.3),
i_1	$\rightarrow$	Ec. (1.4),	i_2	$\rightarrow$	Ec. (1.5),	$deru_c$	$\rightarrow$	Ec. (1.6),
i_C	$\rightarrow$	Ec. (1.7),	$b_{in,1}$	$\rightarrow$	Ec. (1.8),	b_{out}	$\rightarrow$	Ec. (1.9)

Aplicando el algoritmo de la partición, se obtiene el siguiente modelo ordenado y resuelto (existen otras posibles ordenaciones de las ecuaciones que son igualmente válidas):

$$
\begin{aligned}
u_c \quad & \text{Variable de estado} \\
[u_{in}] &= 25 + 20 \cdot \cos\left(\frac{\pi}{30} \cdot t\right) + 5 \cdot \cos\left(\frac{\pi}{2} \cdot t + \frac{\pi}{4}\right) \\
[b_{in,2}] &= \begin{cases} true & \text{si } t \in [30, 90] \cup [150, 210] \\ false & \text{en caso contrario} \end{cases} \\
[x] &= \begin{cases} 0.2 & \text{si } t < 80 \\ 0.5 & \text{si } 80 \leq t < 160 \\ 0.8 & \text{si } t \geq 160 \end{cases} \\
[b_{in,1}] &= \begin{cases} true & \text{si } \frac{u_c}{U_{A/D}} > x \\ false & \text{en caso contrario} \end{cases} \\
[i_1] &= \frac{u_{in} - u_c}{R_1} \\
[i_2] &= \frac{u_c}{R_2} \\
[i_C] &= i_1 - i_2 \\
[b_{out}] &= b_{in,1} \text{ and } b_{in,2} \\
[deru_c] &= \frac{i_C}{C}
\end{aligned}
$$

El modelo posee una variable de estado, por lo que tiene un grado de libertad. En la Figura 1.2 se muestra el diagrama de flujo para la simulación del modelo, empleando el método de integración de Euler explícito y considerando como condición de finalización que el tiempo simulado alcance el valor 240 s. El valor inicial de la variable de estado se ha escogido arbitrariamente igual a cero.

Se ha seleccionado un tamaño del paso igual a 0.01 s. Para ello, se ha repetido la simulación empleando diferentes tamaños del paso, encontrándose que el error cometido escogiendo 0.01 s es admisible para el propósito de este estudio.

Se muestra a continuación el código R que implementa el algoritmo mostrado en la Figura 1.2, con un intervalo de comunicación de 0.1 s. En las Figuras 1.3 y 1.4 se muestran las gráficas obtenida al ejecutar el código R.

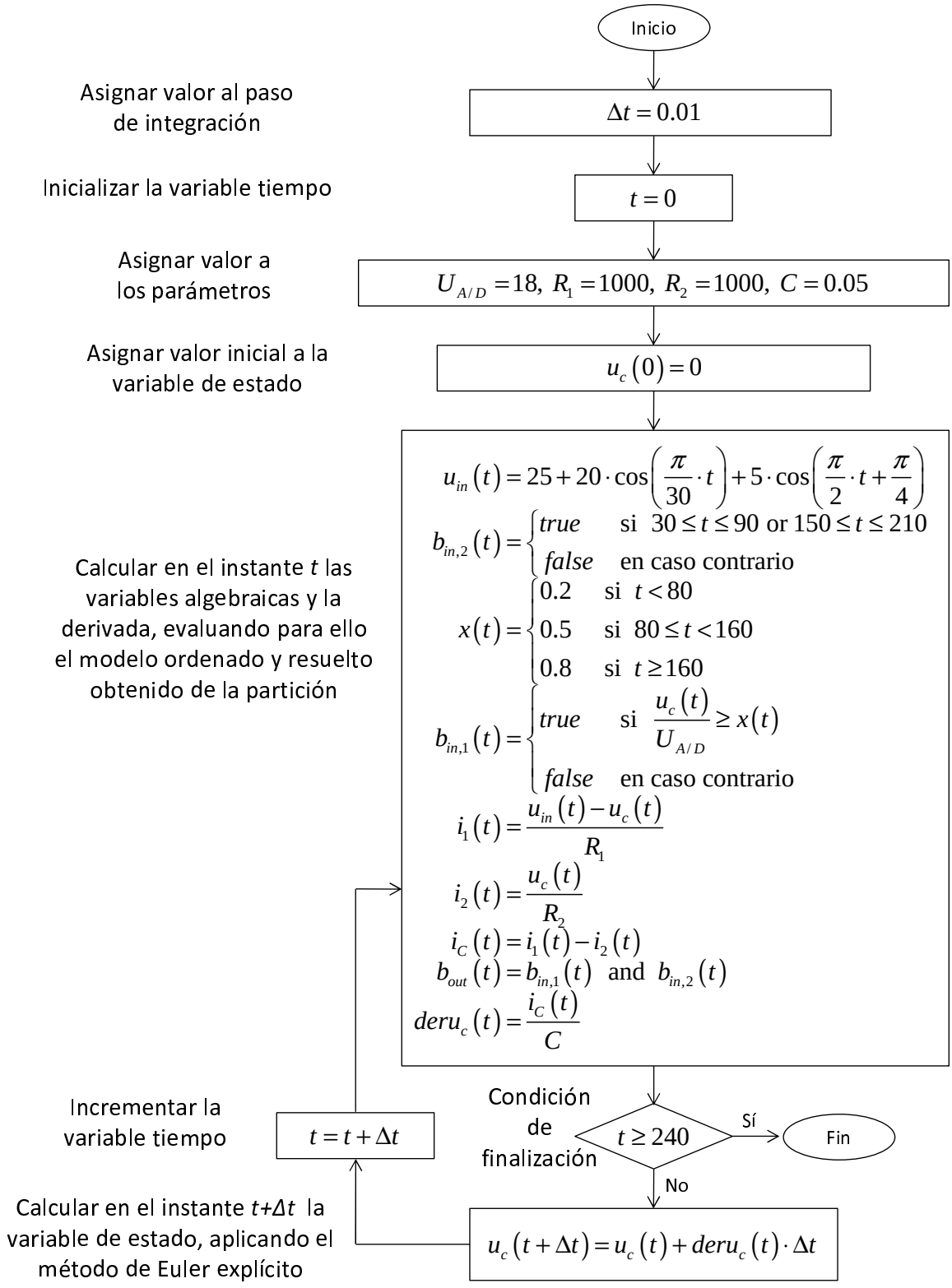


Figura 1.2: Diagrama de flujo de la simulación del modelo.

Código R del algoritmo de la simulación

```
t_fin     <- 240  # Valor final del tiempo
inc_t     <- 0.01 # Paso de integracion
N_com     <- 10   # Intervalo de comunicacion: N_com*inc_t = 0.1 s
Nstep_fin <- round(t_fin / inc_t) # Numero de pasos de integracion
# Constantes
PI    <- acos(-1)
# Parametros
U_AD  <- 18
R1    <- 1000
R2    <- 1000
C     <- 0.05
# Inicializacion del tiempo
t    <- 0
# Valor inicial del estado
uc   <- 0
# Inicializacion resultados salida
t_plot       <- numeric(0)
uin_plot     <- numeric(0)
uc_plot      <- numeric(0)
i1_plot      <- numeric(0)
i2_plot      <- numeric(0)
iC_plot      <- numeric(0)
x_plot       <- numeric(0)
bin1_plot    <- numeric(0)
bin2_plot    <- numeric(0)
bout_plot    <- numeric(0)
# Bucle de la simulacion
termina <- FALSE
n_step  <- 0
while ( !termina ) {
   termina <- n_step == Nstep_fin
   # Calculo variables algebraicas en t
   uin   <- 25 + 20*cos(PI*t/30) + 5*cos(PI*t/2+PI/4)
   bin2  <- (t >= 30 && t<=90) ||  (t >= 150 && t<=210)
   x     <- if (t < 80) 0.2 else if ( t < 160 ) 0.5 else 0.8
   bin1  <- uc / U_AD >= x
   i1    <- (uin - uc) / R1
   i2    <- uc / R2
   iC    <- i1 - i2
   bout  <- bin1 && bin2
   deruc <- iC / C
   # Almacenar resultados de salida
   if ( termina | n_step %% N_com == 0 ) {
      t_plot     <- c(t_plot, t)
      uin_plot   <- c(uin_plot, uin)
      uc_plot    <- c(uc_plot, uc)
      i1_plot    <- c(i1_plot, i1)
      i2_plot    <- c(i2_plot, i2)
      iC_plot    <- c(iC_plot, iC)
      x_plot     <- c(x_plot, x)
      bin1_plot  <- c(bin1_plot, bin1)
      bin2_plot  <- c(bin2_plot, bin2)
      bout_plot  <- c(bout_plot, bout)
   }
   n_step <- n_step + 1
   # Valor en t+inc_t de la variable de estado
   uc <- uc + inc_t*deruc
   # Actualizacion del valor del tiempo
   t <- t + inc_t
}
```

```
# Representacion grafica
# Para poder usar minor.tick es necesario instalar el paquete Hmisc
# Configuracion de la representacion grafica

plot(t_plot, uin_plot, ylim=c(0,52), xlab = "t [s]",
     ylab = "uin (azul), uc (rojo) [V]", type="l", col="blue", lwd=1.5,
     panel.first = abline(v = seq(0,t_fin,10), h = seq(0,52,2),
                          lwd = 0.5, lty = 3, col="grey") )
points(t_plot, uc_plot, type="l", col="red", lwd=1.5)
minor.tick(nx = 5, ny = 5, tick.ratio = 0.5)

dev.new()
plot(t_plot, i1_plot, ylim=c(-0.03,0.05), xlab = "t [s]",
     ylab = "i1 (azul), i2 (rojo), iC (verde) [A]",
     type="l", col="blue", lwd=1.5,
     panel.first = abline(v = seq(0,t_fin,10), h = seq(-0.03,0.05,0.002),
                          lwd = 0.5, lty = 3, col="grey") )
points(t_plot, i2_plot, type="l", col="red", lwd=1.5)
points(t_plot, iC_plot, type="l", col="green", lwd=1.5)
minor.tick(nx = 5, ny = 10, tick.ratio = 0.5)

dev.new()
plot(t_plot, x_plot, ylim=c(0,1), xlab = "t [s]",
     ylab = "x", type="l", col="blue", lwd=1.5,
     panel.first = abline(v = seq(0,t_fin,10), h = seq(0,1,0.1),
                          lwd = 0.5, lty = 3, col="grey") )
minor.tick(nx = 5, ny = 5, tick.ratio = 0.5)

dev.new()
par( mfrow = c(3,1) )
plot(t_plot, bin1_plot, ylim=c(0,1), xlab = "t [s]",
     ylab = "bin1", type="l", col="blue", lwd=1.5,
     panel.first = abline(v = seq(0,t_fin,10), h = seq(0,1,1),
                          lwd = 0.5, lty = 3, col="grey") )
plot(t_plot, bin2_plot, ylim=c(0,1), xlab = "t [s]",
     ylab = "bin2", type="l", col="blue", lwd=1.5,
     panel.first = abline(v = seq(0,t_fin,10), h = seq(0,1,1),
                          lwd = 0.5, lty = 3, col="grey") )
plot(t_plot, bout_plot, ylim=c(0,1), xlab = "t [s]",
     ylab = "bout", type="l", col="blue", lwd=1.5,
     panel.first = abline(v = seq(0,t_fin,10), h = seq(0,1,1),
                          lwd = 0.5, lty = 3, col="grey") )
```

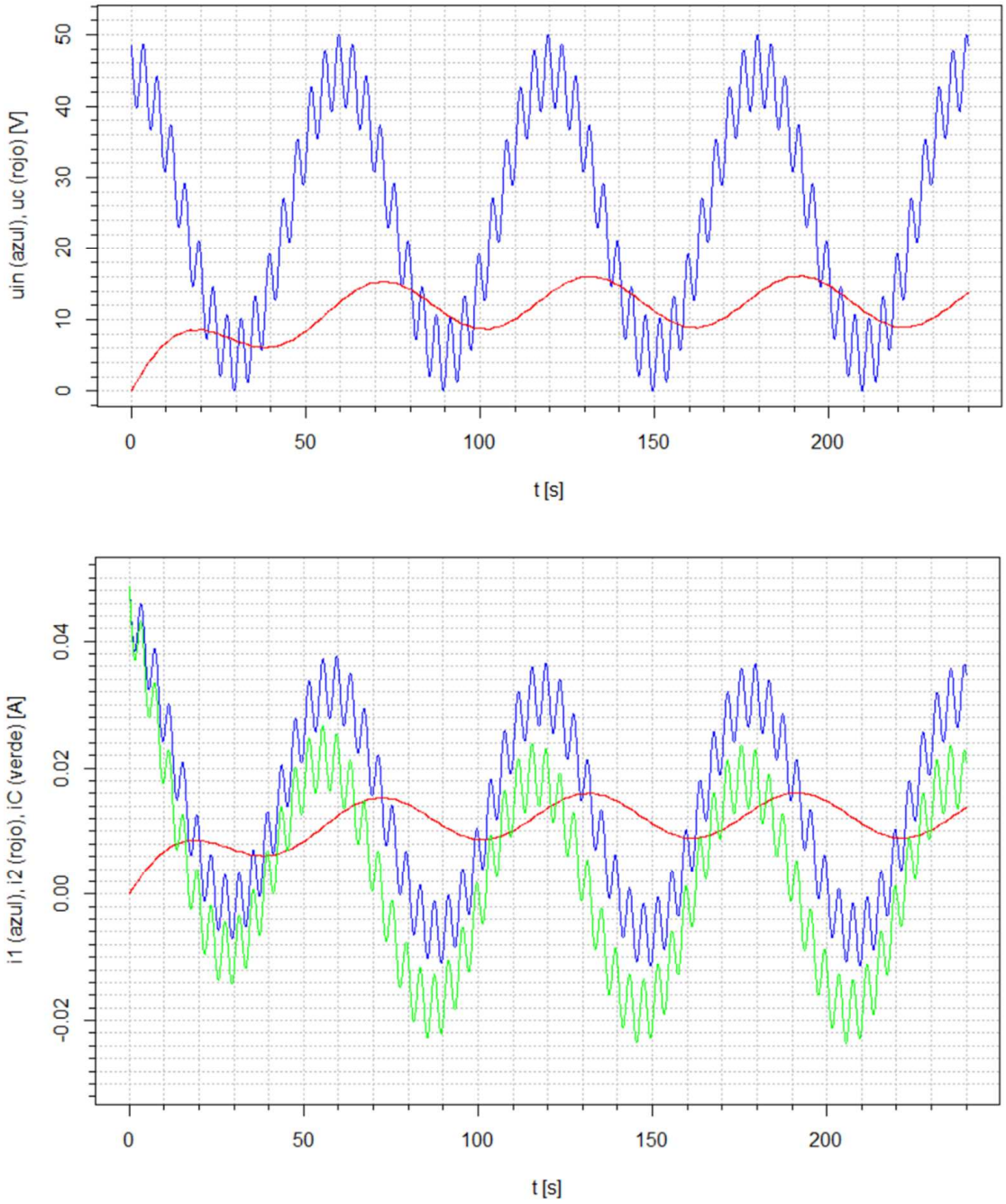


Figura 1.3: Resultado de la ejecución del código R.

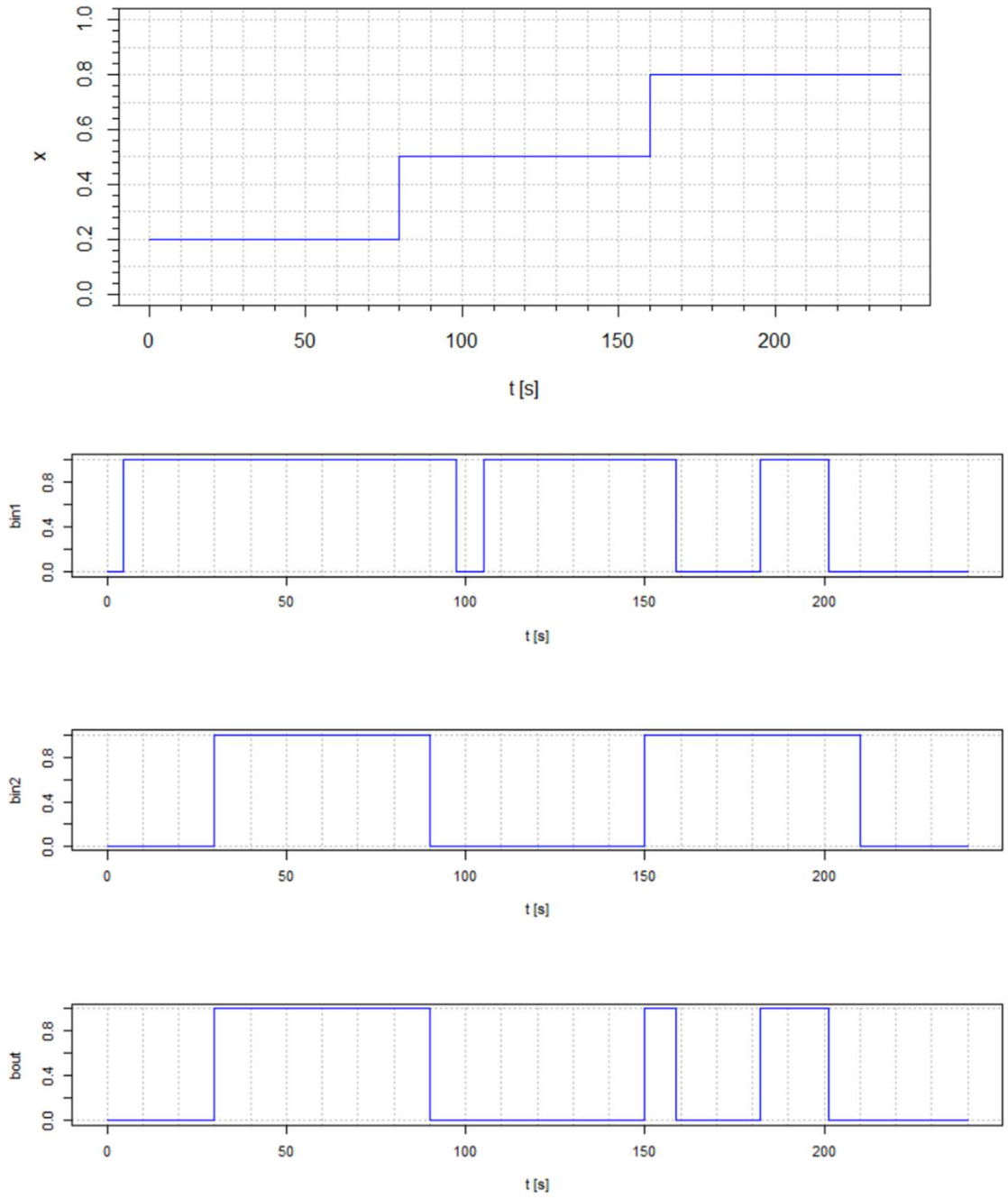


Figura 1.4: Resultado de la ejecución del código R.

EJERCICIO 3

Describa en lenguaje Modelica el sistema del ejercicio anterior, de las dos maneras siguientes:

1. Como un modelo atómico, que viene descrito por las ecuaciones que usted ha planteado al contestar a la pregunta anterior.
2. Programe una librería que contenga los componentes necesarios para componer el sistema descrito en el Ejercicio 2. A continuación, defina dicho sistema como un modelo compuesto, instanciando y conectando componentes de la librería que ha creado.

Asigne a los parámetros los valores indicados en el ejercicio anterior y simule los dos modelos anteriores durante 240 s.

Represente gráficamente frente al tiempo las variables siguientes: u_{in}, u_c, i_1, i_2, i_C, x, $b_{in,1}$, $b_{in,2}$, b_{out}.

Solución al Ejercicio 3

El modelo atómico en lenguaje Modelica se muestra en Código 1.1. El resultado de la simulación está representado en las Figuras 1.5 y 1.6.

En el Código 1.2 – 1.4 se muestra una posible forma de modelar los componentes, organizándolos en una librería, y de componer el sistema mediante dichos componentes.

```
model circ
  import SI = Modelica.SIunits;
  SI.Voltage u_in,u_C(start=0,fixed=true);
  SI.Current i_1,i_2,i_C;
  Real x;
  Boolean b_in1,b_in2,b_out;
  constant Real pi = Modelica.Constants.pi;
  parameter SI.Resistance R1 = 1000,R2 = 1000;
  parameter SI.Capacitance C = 0.05;
  parameter SI.Voltage U_AD = 18;
equation
  u_in = 25 + 20*cos(pi/30*time) + 5*cos(pi/2*time+pi/4);
  b_in2 = time >= 30 and time <= 90 or time >=150 and time <= 210;
  x = if time < 80 then 0.2 elseif time < 160 then 0.5 else 0.8;
  u_in - u_C = R1 * i_1;
  u_C = R2 * i_2;
  C * der(u_C) = i_C;
  b_in1 = u_C / U_AD > x;
  b_out = b_in1 and b_in2;
  i_1 = i_2 + i_C;
  annotation (experiment(
      StopTime=240,
      __Dymola_Algorithm="Dassl"));
end circ;
```

Código 1.1: Ejercicio 3.1: modelo atómico del sistema.

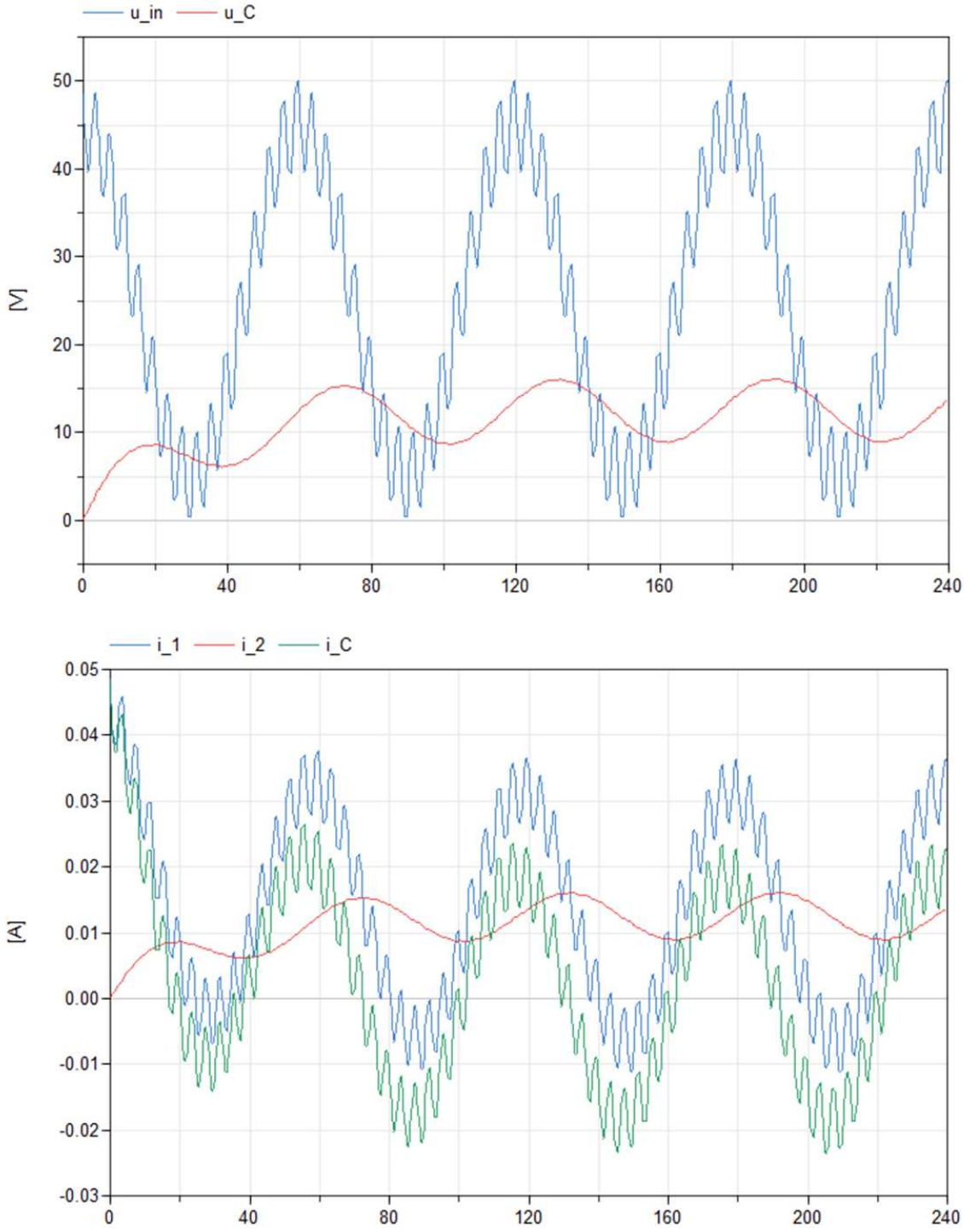


Figura 1.5: Resultado de la simulación usando Dymola.

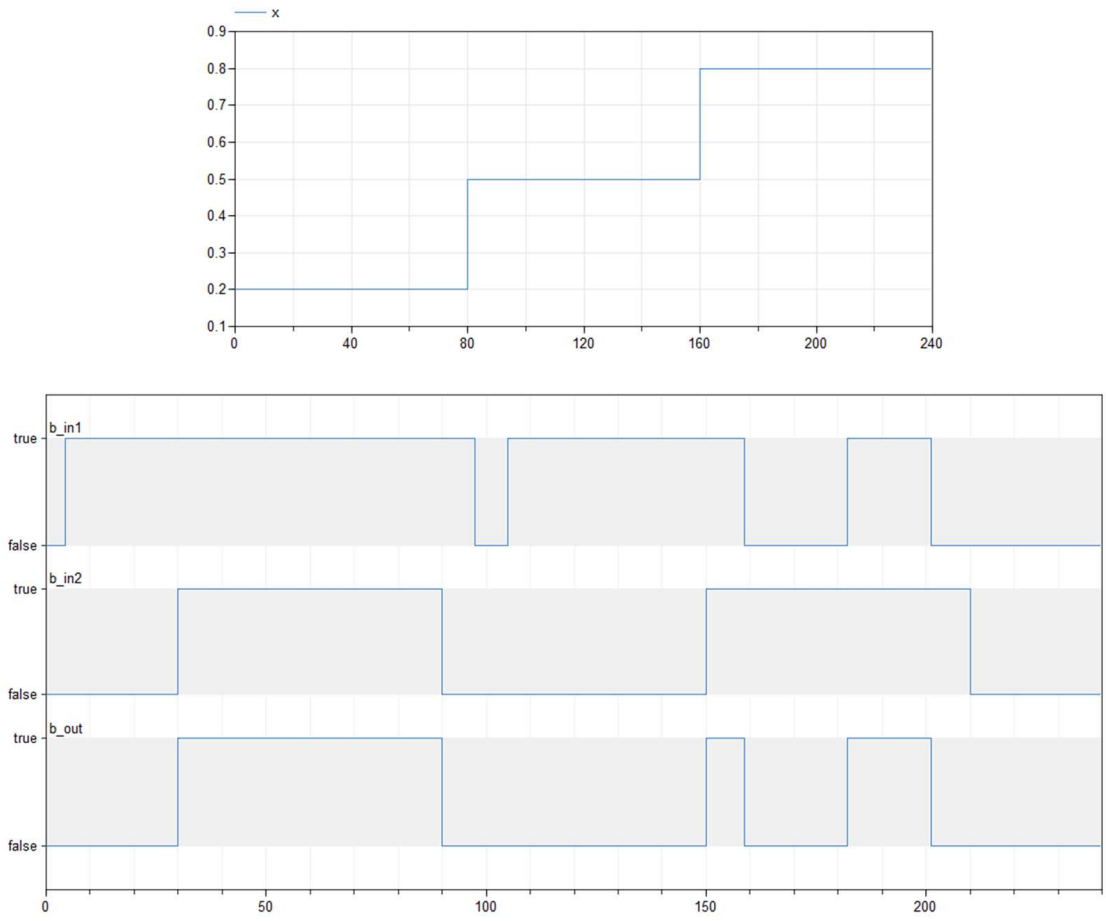


Figura 1.6: Resultado de la simulación usando Dymola.

```
package LibElectrica

  import SI = Modelica.SIunits;
  import Modelica.Constants;

  package Interfaces

    connector Pin
      SI.Voltage u;
      flow SI.Current i;
    end Pin;

    connector SignalRealIn
       input Real x;
    end SignalRealIn;

    connector SignalRealOut
       output Real x;
    end SignalRealOut;

    connector PinLogicoIn
      input Boolean b;
    end PinLogicoIn;

    connector PinLogicoOut
      output Boolean b;
    end PinLogicoOut;

  end Interfaces;

  package Fuentes

    model Voltaje_uin
      Interfaces.Pin p;
    equation
      p.u = 25 +
            20 * cos(Constants.pi*time/30) +
             5 * cos(Constants.pi*time/2 + Constants.pi/4);
    end Voltaje_uin;

    model SegnalLogica_bin2
       Interfaces.PinLogicoOut bout;
    equation
       bout.b = time >= 30 and time <= 90 or
                time >=150 and time <= 210;
    end SegnalLogica_bin2;

    model SegnalReal_x
       Interfaces.SignalRealOut sout;
    equation
       sout.x = if time < 80 then 0.2
                elseif time < 160 then 0.5 else 0.8;
    end SegnalReal_x;

  end Fuentes;
```

Código 1.2: Ejercicio 3.2: librería (1/3).

```
package Componentes

  model puertaAND2
    Interfaces.PinLogicoIn in1,in2;
    Interfaces.PinLogicoOut out;
  equation
    out.b = in1.b and in2.b;
  end puertaAND2;

  model Resistencia "Resistencia ideal"
    Interfaces.Pin p,n;
    parameter SI.Resistance R "Resistencia";
  protected
    SI.Voltage u "Voltaje entre pines (= p.u - n.u)";
  equation
    u = p.u - n.u;
    p.i = -n.i;
    u = R * p.i;
  end Resistencia;

  model Condensador "Condensador ideal"
    Interfaces.Pin p,n;
    parameter SI.Capacitance C "Capacidad";
  protected
    SI.Voltage u "Voltaje entre pines (= p.u - n.u)";
  equation
    u = p.u - n.u;
    p.i = -n.i;
    C * der(u) = p.i;
  end Condensador;

  model Tierra "Tierra"
    Interfaces.Pin p;
  equation
    p.u = 0;
  end Tierra;

  model ConversorAD
    Interfaces.SignalRealIn sxin;
    Interfaces.Pin p;
    Interfaces.PinLogicoOut bout;
    parameter SI.Voltage U_AD;
  equation
    bout.b = p.u / U_AD > sxin.x;
    p.i = 0;
  end ConversorAD;

end Componentes;
```

Código 1.3: Ejercicio 3.2: librería (2/3).

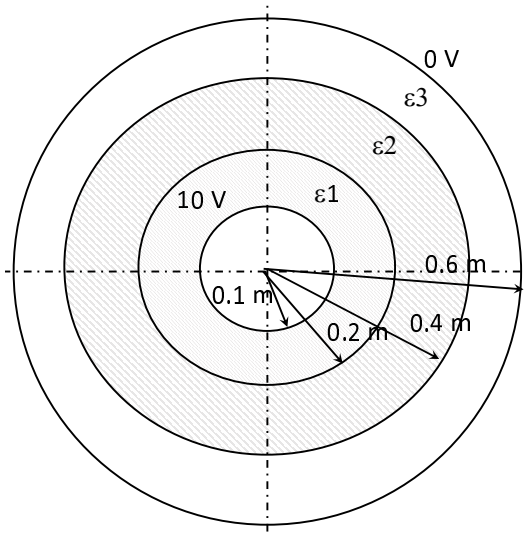
```
  package Ejemplos

  model circ
    Componentes.Resistencia R1(R=1000),R2(R=1000);
    Componentes.Condensador C(C=0.05);
    Componentes.Tierra ground;
    Componentes.ConversorAD ConvAD(U_AD=18);
    Componentes.puertaAND2 AND2;
    Fuentes.Voltaje_uin Vuin;
    Fuentes.SegnalLogica_bin2 SLbin2;
    Fuentes.SegnalReal_x SRx;
  equation
    connect(Vuin.p,R1.p);
    connect(R1.n,R2.p);
    connect(R1.n,C.p);
    connect(R1.n,ConvAD.p);
    connect(R2.n,ground.p);
    connect(C.n,ground.p);
    connect(ConvAD.sxin,SRx.sout);
    connect(ConvAD.bout,AND2.in1);
    connect(SLbin2.bout,AND2.in2);
    annotation (experiment(
      StopTime=240,
      __Dymola_Algorithm="Dassl"));
  end circ;

  end Ejemplos;
end LibElectrica;
```

Código 1.4: Ejercicio 3.2: librería (3/3).

EJERCICIO 4

Escriba un *script* en FlexPDE para obtener las líneas equipotenciales y los vectores correspondientes al campo eléctrico del sistema descrito a continuación.

El sistema es un cable coaxial que vamos a simular en dos dimensiones, en una sección transversal al cable. En la siguiente figura se muestra dicha sección transversal. Está compuesta por un conductor de sección circular y radio 0.1 m que se encuentra a 10 V, rodeado por tres medios de permitividad ε_1, ε_2 y ε_3 dispuestos concéntricamente, con espesores 0.1, 0.2 y 0.2 m respectivamente. La superficie exterior es un conductor que se encuentra a 0 V.



La permitividad de los medios es:

$$\begin{aligned} \varepsilon_1 &= 2.3 \cdot \varepsilon_0 \\ \varepsilon_2 &= 3 \cdot \varepsilon_0 \\ \varepsilon_3 &= 4 \cdot \varepsilon_0 \end{aligned}$$

donde $\varepsilon_0 = 8.85 \cdot 10^{-12}$ F/m es la permitividad del vacío.

Solución al Ejercicio 4

El *script* se muestra en el Código 1.5, el gráfico del potencial en la Figura 1.7 y el gráfico vectorial del campo eléctrico en la Figura 1.8.

```
TITLE
   'Cable'
SELECT
   errlim=3e-4      spectral_colors
VARIABLES
   U
DEFINITIONS
   r1 = 0.1  r2=0.2 r3 =0.4  r4=0.6
   eps0 = 8.85e-12  eps3=4 eps2=3 eps1=2.3 eps=eps0
   Ex=-dx(U)     Ey=-dy(U)     E=-grad(U)     Em=magnitude(E)
   Dex = eps*Ex Dey=eps*Ey D=eps*E
EQUATIONS
   div( D)=0
BOUNDARIES
region 'dominio'
   eps = eps3*eps0
   start 'exterior' (0,-r4) value(U)=0  {Exterior}
   arc( center=0,0)  angle=360
region 'dielectrico2'
   eps = eps2*eps0
   start 'dielectrico2' (0,-r3) natural(U)=0  {Dielectrico3}
   arc( center=0,0)  angle=360
region 'dielectrico1'
   eps = eps1*eps0
   start 'dielectrico1' (0,-r2) natural(U)=0  {Dielectrico2}
   arc( center=0,0)  angle=360
region 'interior'
     start 'interior' (0,-r1) value(U)=10         {Interior}
   arc( center=0,0) angle=360

PLOTS
   contour(U)  vector( E) norm
END
```

Código 1.5: Modelo del cable coaxial.

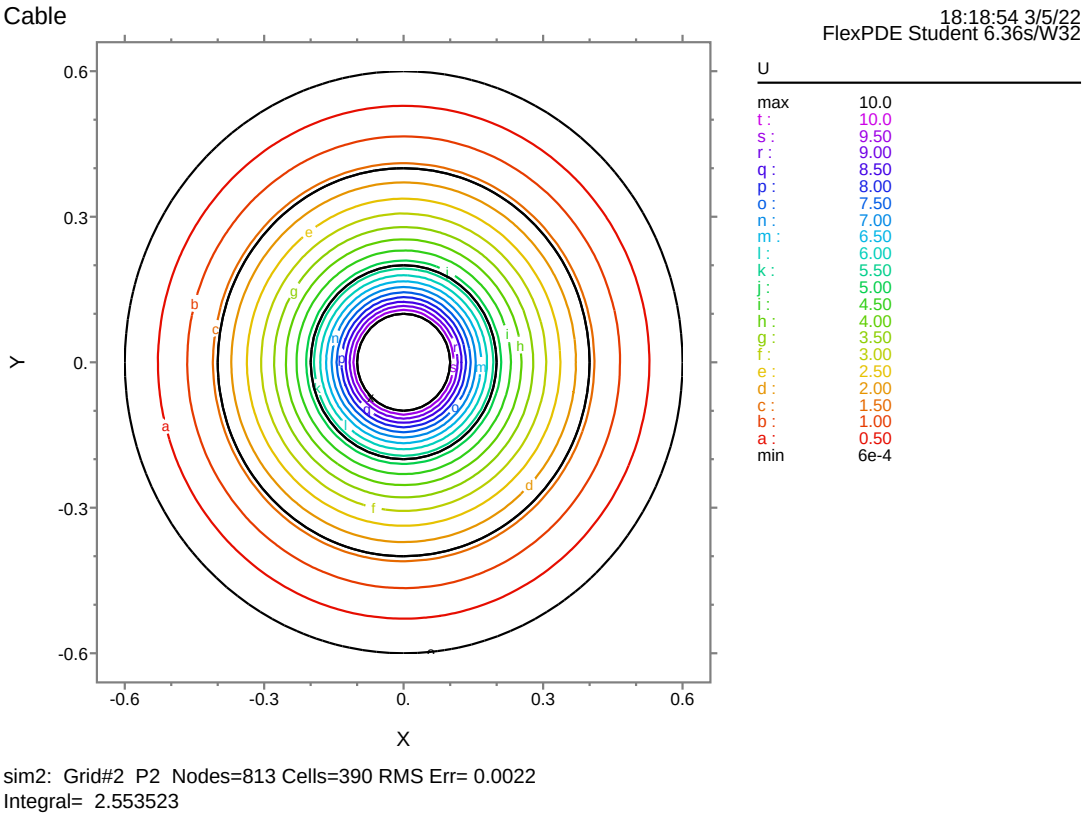


Figura 1.7: Curvas equipotenciales.

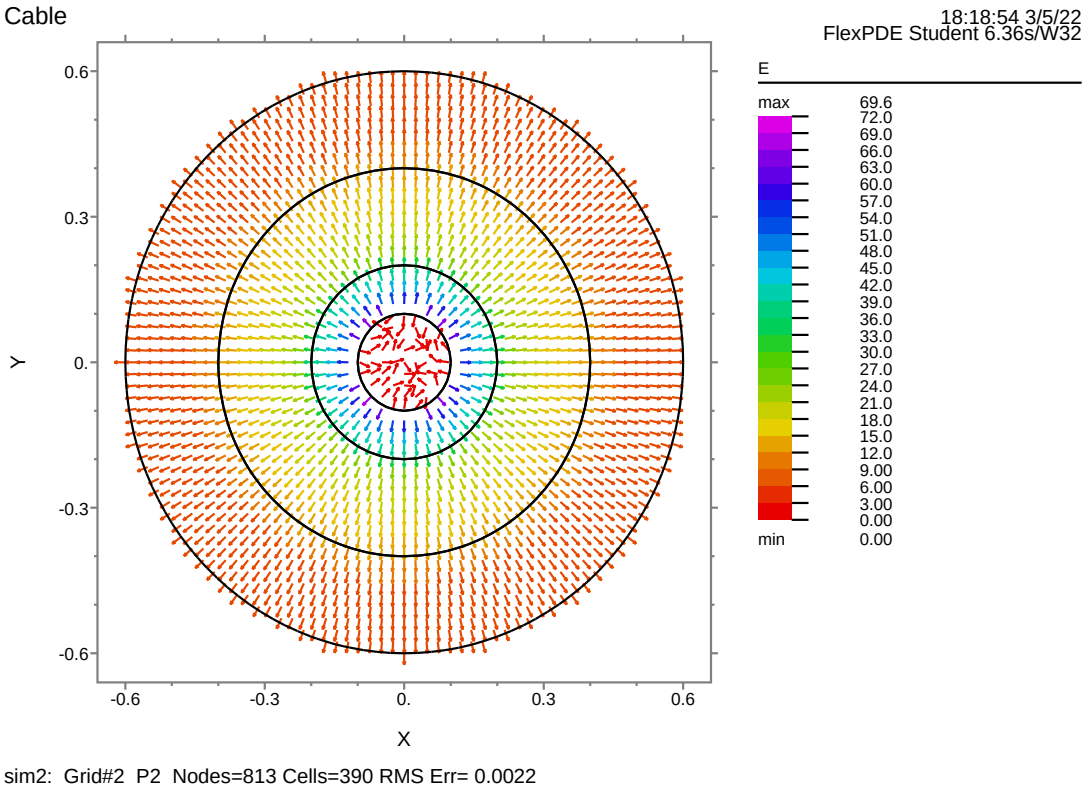


Figura 1.8: Campo eléctrico