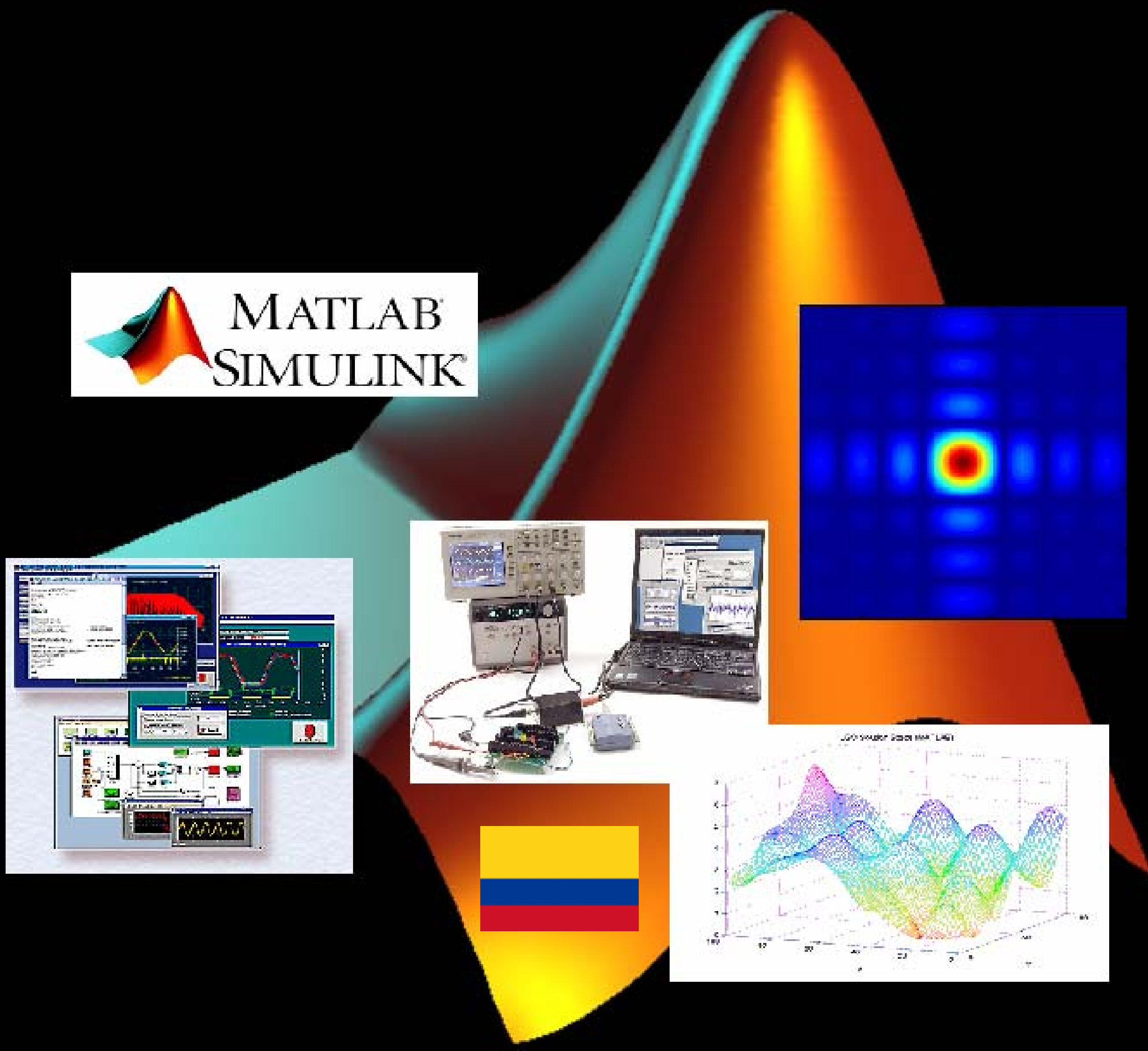
DISEÑO DE UN CONTROLADOR PID ANALOGO PARA UN CIRCUITO RC DE SEGUNDO ORDEN MEDIANTE LA SISOTOOL DE MATLAB



POR:
EDWIN GONZALEZ QUERUBIN
MORGAN GARAVITO VASQUEZ
INGENIERIA MECATRONICA
2007

DISEÑO DE UN CONTROLADOR PID ANALOGO PARA UN CIRCUITO RC DE SEGUNDO ORDEN MEDIANTE LA SISOTOOL DE MATLAB

POR:
EDWIN GONZÁLEZ QUERUBÍN
MORGAN GARAVITO VÁSQUEZ

FACULTAD DE INGENIERÍA MECATRÓNICA

UNIVERSIDAD SANTO TOMAS
BUCARAMANGA – COLOMBIA
2007

Introducción

El diseño de sistemas de control se puede realizar, ya sea en el dominio del tiempo o en el de la frecuencia. A menudo se emplean especificaciones de diseño para describir que debe hacer el sistema y como hacerlo. Siendo estas únicas para cada diseño.

Por lo tanto el diseño de sistemas de control involucra tres pasos:

- Determinar que debe hacer el sistema y como hacerlo.
- Determinar la configuración del compensador.
- Determinar los valores de los parámetros del controlador para alcanzar los objetivos de diseño.

Para alcanzar estos objetivos nos basaremos en cálculos matemáticos realizados a mano y con la ayuda del software **MATLAB** y la herramienta **Sisotool** para determinar la ubicación de los polos dominantes de nuestro sistema y observar si cumple o no con nuestras especificaciones de diseño. Así mismo se contará con la ayuda de **Circuit Maker** y **Simulink**, los cuales nos servirán para la verificación de los datos obtenidos y así poder constatar el funcionamiento de nuestro diseño.

OBJETIVOS

- Obtener la función de transferencia de la planta o circuito rc.
- Obtener la función de transferencia del controlador basado en el circuito con amplificadores operacionales (Proporcional - Integral - Derivativo).
- Obtener mediante la sisotool de Matlab la función de transferencia del controlador con los parámetros de diseño establecidos.
- Por medio de los resultados obtenidos con la sisotool de Matlab y el controlador hallado manualmente, calcular y seleccionar los elementos de circuito para el controlador PID.
- Simular el sistema de control de lazo cerrado compensado y no compensado ante una entrada escalón.
- Con base a los elementos de circuito calculados, simular mediante Circuit Maker el circuito.
- Ensamble y pruebas del controlador.

CIRCUITO RC O PLANTA

Como se puede observar en la figura 1, la planta es un sistema de segundo orden, ya que contiene dos elementos almacenadores de energía que son los condensadores Ca y Cb.

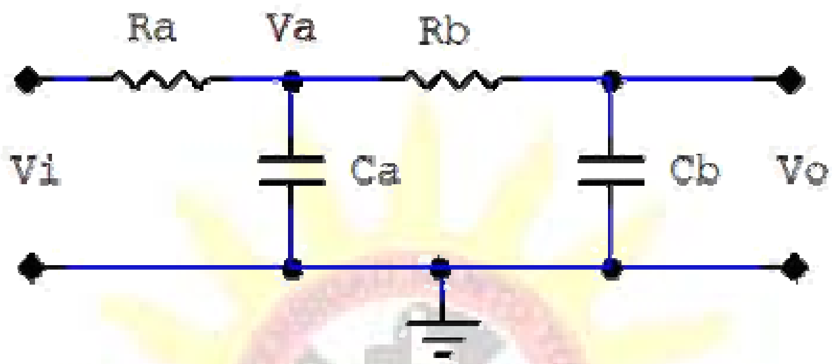


Figura 1

Función de transferencia de la planta:

1) $\frac{Vi - Va}{Ra} = \frac{Va - 0}{Ca} + \frac{Va - Vo}{Rb}$ **1.1)** $\frac{Vi}{Ra} = Va(\frac{1}{Ra} + \frac{1}{Ca} + \frac{1}{Rb}) - \frac{Vo}{Rb}$

2) $\frac{Va - Vo}{Rb} = \frac{Vo - 0}{Cb}$ $\frac{Va}{Rb} = Vo(\frac{1}{Rb} + \frac{1}{Cb})$

2.2) $Va = Vo(1 + \frac{Rb}{Cb})$

1.1) y **2.2)** $\frac{Vi}{Ra} = Vo(1 + \frac{Rb}{Cb})(\frac{1}{Ra} + \frac{1}{Ca} + \frac{1}{Rb}) - \frac{Vo}{Rb}$

Edwin González Querubín kaliman83@hotmail.com
Morgan Garavito Vásquez mor6an1@hotmail.com

$$\frac{Vo}{Vi}=\frac{1}{Ra\left[(1+\frac{Rb}{Cb})(\frac{1}{Ra}+\frac{1}{Ca}+\frac{1}{Rb})-\frac{1}{Rb}\right]}$$

$$=\frac{1}{Ra\left[\frac{1}{Ra}+\frac{1}{Ca}+\frac{1}{Rb}+\frac{Rb}{RaCb}+\frac{Rb}{CaCb}+\frac{1}{Cb}-\frac{1}{Rb}\right]}$$

$$\frac{Vo}{Vi}=\frac{1}{1+\frac{Ra}{Ca}+\frac{Rb}{Cb}+\frac{RaRb}{CaCb}+\frac{Ra}{Cb}}\quad=\frac{1}{\frac{RaRb}{CaCb}+\frac{1}{Cb}(Ra+Rb)+\frac{Ra}{Ca}+1}$$

$$\frac{Vo}{Vi}(s)=\frac{1}{RaRbCaCbS\text{^}2+(Ra+Rb)CbS+RaCaS+1}$$

$$\frac{Vo}{Vi}(s)=\frac{1}{RaRbCaCbS\text{^}2+\left[(Ra+Rb)Cb+RaCa\right]S+1}$$

3) $$G(s)=\frac{1}{RaRbCaCbS\text{^}2+(RaCa+RaCb+RbCb)S+1}$$

Edwin González Querubín kaliman83@hotmail.com
Morgan Garavito Vásquez mor6an1@hotmail.com

CALCULO DE LAS ETAPAS DEL CONTROLADOR PID

En la figura 2 se puede observar el circuito equivalente del controlador PID con cada una de sus etapas, proporcional, integral y derivativa, respectivamente.

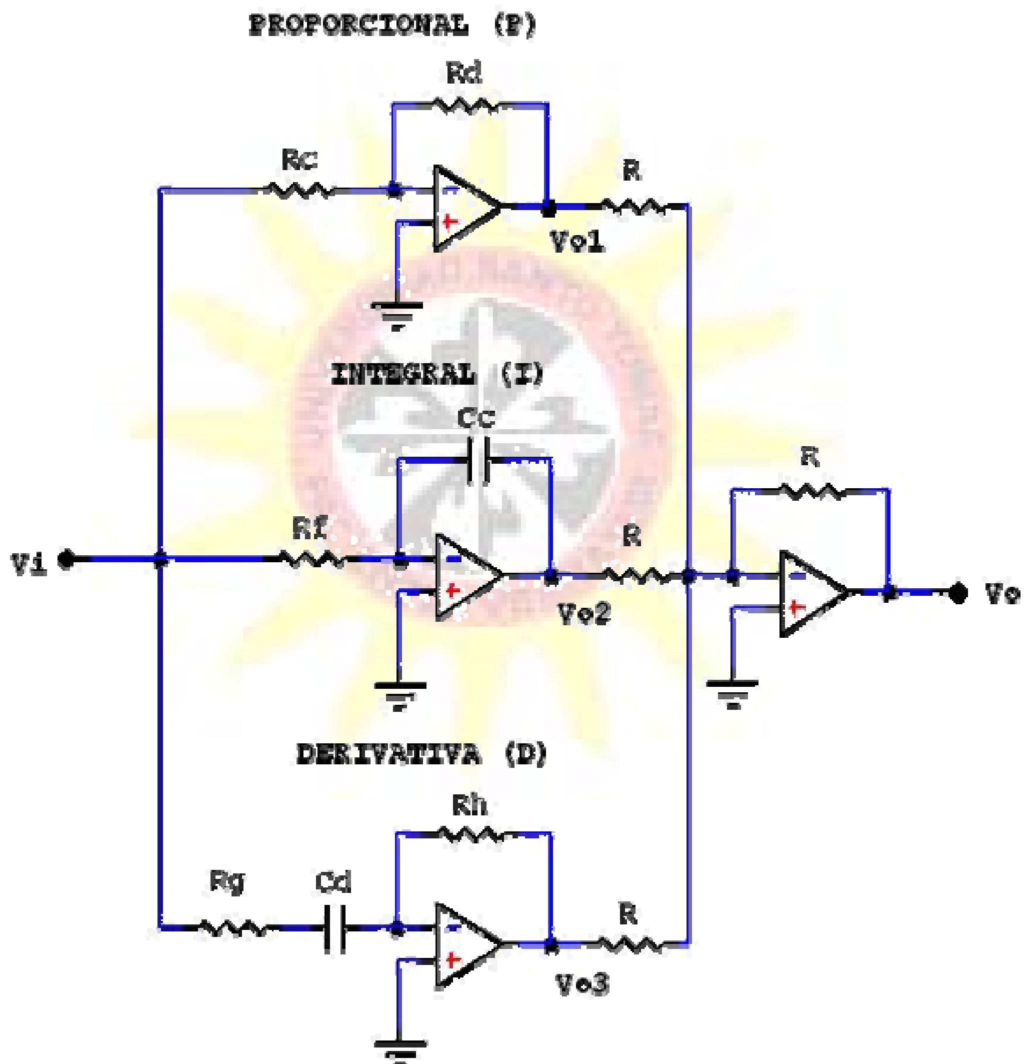


Figura 2

Edwin González Querubín kaliman83@hotmail.com
Morgan Garavito Vásquez mor6an1@hotmail.com

Parte proporcional (P)

El controlador proporcional es esencialmente un controlador anticipativo, así mismo, este tendrá efecto sobre el error en estado estable sólo si el error varía con respecto al tiempo, pero tiene una gran desventaja, atenúa el ruido en frecuencias altas. El diseño de este tipo de controlador afecta el desempeño de un sistema de control de las siguientes maneras:

- Mejora el amortiguamiento y reduce el sobrepaso máximo.
- Reduce el tiempo de asentamiento y levantamiento.
- Mejora el margen de ganancia, el margen de fase.
- En la implementación de un circuito, puede necesitar de un capacitor muy grande.

La parte proporcional no considera el tiempo, por tanto la mejor manera de solucionar el error permanente y hacer que el sistema contenga alguna componente que tenga en cuenta la variación con respecto al tiempo es incluyendo y configurando las acciones integral y derivativa.

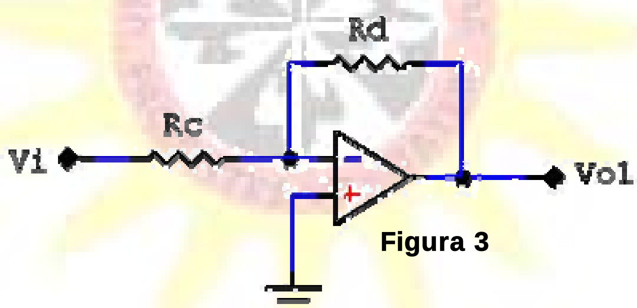


Figura 3

$$\frac{Vi-0}{Rc}=\frac{0-Vo1}{Rd} \qquad \frac{Vi}{Rc}=\frac{-Vo1}{Rd} \qquad \frac{Vo1}{Vi}=\frac{-Rd}{Rc} \qquad \textbf{4)}\ Vo1(s)=\frac{-RdVi(s)}{Rc}$$

Parte integral (I)

El modo de control integral tiene como propósito disminuir y eliminar el error en estado estacionario provocado por el modo proporcional.

El error es integrado, lo cual tiene la función de promediarlo o sumarlo por un periodo de tiempo determinado, luego es multiplicado por una constante **I**.

I representa la constante de integración. Posteriormente, la respuesta integral es adicionada al modo Proporcional para formar el control P + **I** con el propósito de obtener una respuesta estable del sistema sin error estacionario, debido a que al incorporar un polo en lazo abierto en el origen, se desplaza el lugar geométrico de las raíces del sistema hacia el semiplano derecho de s. Por esta razón, en la práctica la acción integral suele acompañarse por otras acciones de control.

El controlador integral es un circuito electrónico que genera una salida proporcional a la señal de entrada. La figura 6 muestra el circuito de un controlador integral, el capacitor C esta conectado entre la entrada inversora y la salida. De esta forma, la tensión en las terminales del capacitor es además la tensión de salida.

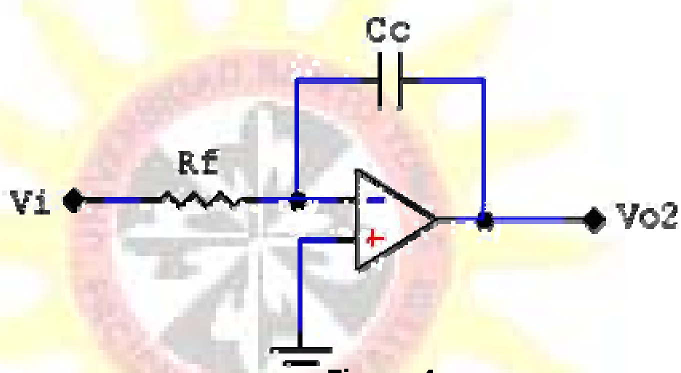


Figura 4

$$\frac{Vi-0}{Rf}=\frac{0-Vo2}{Cc} \qquad \frac{Vi}{Rf}=\frac{-Vo2}{Cc} \qquad \frac{Vi(s)}{Rf}=-CcVo2(s)S \qquad \textbf{5)} \quad Vo2(s)=\frac{-Vi(s)}{RfCcS}$$

Parte derivativa (D)

La acción derivativa se manifiesta cuando hay un cambio en el valor absoluto del error; (si el error es constante, solamente actúan los modos proporcional e integral). El error es la desviación existente entre el punto de medida y el valor de consigna, o "set point".

La función de la acción derivativa, es mantener el error al mínimo corrigiéndolo proporcionalmente con la velocidad misma que se produce, de esta manera evita que el error se incremente.

Se deriva con respecto al tiempo y se multiplica por una constante D y luego se suma a las señales anteriores (P+I). Es importante gobernar la respuesta de control a los cambios en el sistema, ya que una mayor acción derivativa corresponde a un cambio más rápido y el controlador puede responder adecuadamente.

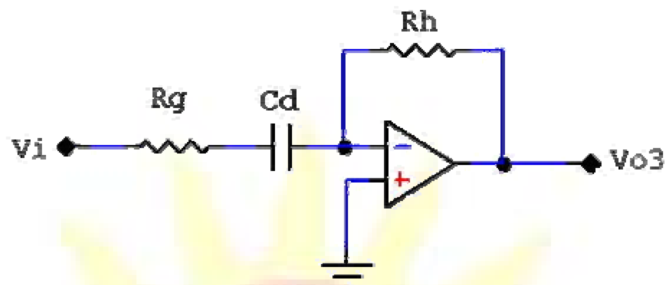


Figura 5

$$\frac{Vi-0}{(Rg+Cd)}=\frac{0-Vo3}{Rh} \qquad \frac{Vi}{Rg+Cd}=\frac{-Vo3}{Rh} \qquad Vo3=\frac{-RhVi}{Rg+Cd} \qquad Vo3(s)=\frac{-RhVi(s)}{Rg+\frac{1}{CdS}}$$

6) $$Vo3(s)=\frac{-RhCdVi(s)S}{1+RgCdS}$$

FUNCIÓN DE TRANSFERENCIA DEL CONTROLADOR PID

En base al circuito de la figura 2 y a los valores hallados en cada una de las etapas, procedemos a obtener $\frac{Vo}{Vi}$, el cual va a ser la función de transferencia del PID análogo, C(s).

$$\frac{Vo1(s)-0}{R}+\frac{Vo2(s)-0}{R}+\frac{Vo3(s)-0}{R}=\frac{0-Vo(s)}{R}$$

7) $Vo1(s)+Vo2(s)+Vo3(s)=-Vo(s)$

Reemplazando la ecuación **4**, **5** y **6** en la **7**:

$$\frac{-RdVi(s)}{Rc}-\frac{Vi(s)}{RfCcS}-\frac{RhCdVi(s)S}{1+RgCdS}=-Vo(s)$$

$$\frac{Vo}{Vi}(s)=\frac{Rd}{Rc}+\frac{1}{RfCcS}+\frac{RhCdS}{1+RgCdS}$$

$$\frac{Vo}{Vi}(s)=\frac{Rd(RfCcS)(1+RgCdS)+Rc(1+RgCdS)+RhCdS(Rc)(RfCcS)}{Rc(RfCcS)(1+RgCdS)}$$

$$\frac{Vo}{Vi}(s)=\frac{RdRfCcS+RdRfRgCcCdS\text{^}2+Rc+RcRgCdS+RcRfRhCcCdS\wedge 2}{RcRfCcS(1+RgCdS)}$$

$$\frac{Vo}{Vi}(s)=\frac{(RdRfRgCcCd+RcRfRhCcCd)S\text{^}2+(RdRfCc+RcRgCd)S+Rc}{RcRfCcRgCdS(S+\frac{1}{RgCd})}$$

$$\frac{Vo}{Vi}(s)=\frac{(\frac{RdRfRgCcCd}{RcRfRgCcCd}+\frac{RcRfRhCcCd}{RcRfRgCcCd})S\text{^}2+(\frac{RdRfCc}{RcRfRgCcCd}+\frac{RcRgCd}{RcRfRgCcCd})S+\frac{Rc}{RcRfRgCcCd}}{S(S+\frac{1}{RgCd})}$$

$$\frac{Vo}{Vi}(s)=\frac{(\frac{Rd}{Rc}+\frac{Rh}{Rg})S\text{^}2+(\frac{Rd}{RcRgCd}+\frac{1}{RfCc})S+\frac{1}{RfRgCcCd}}{S(S+\frac{1}{RgCd})}$$

8) $C(s)=\dfrac{(\frac{Rd}{Rc}+\frac{Rh}{Rg})S^{\wedge}2+(\frac{Rd}{RcRgCd}+\frac{1}{RfCc})S+\frac{1}{RfRgCcCd}}{S(S+\frac{1}{RgCd})}$

CALCULO DE LOS PARAMETROS DEL COMPENSADOR MEDIANTE MATLAB Y LA SISOTOOL

Nota: Todos los códigos de Matlab que se utilizarán durante este trabajo, se ejecutarán de forma sucesiva. Por lo que el código en el que se sitúe necesitará de los anteriores.

```
%============================================================
%            Funcion de transferencia de la Planta
%============================================================
%  Vo                          1
%  -- = -----------------------------------------------------
%  Vi   [(Ra*Rb*Ca*Cb)]*S^2 + [(Ra*Ca)+(Ra*Cb)+(Rb*Cb)]*S +1
%============================================================
%Valores asignados a los componentes de la planta.
Ra=1000;
Rb=10000;
Ca=0.2e-6;
Cb=0.1e-9;
%Numerador de la planta.
n1=1;
%Denominador de la planta.
d1=[Ra*Rb*Ca*Cb, Ra*Ca+Ra*Cb+Rb*Cb, 1];
%Funcion de transferencia de la planta.
Planta=tf(n1,d1)
```

En el anterior código se implementó la ecuación **3),** que es la función de transferencia de la planta, asumiendo valores para cada uno de los elementos:

Ra=1kΩ
Rb=10kΩ
Ca=0.2μf
Cb=0.1nf

Con estos valores y con la ayuda de Matlab obtenemos los siguientes resultados:

Función de transferencia de la planta:

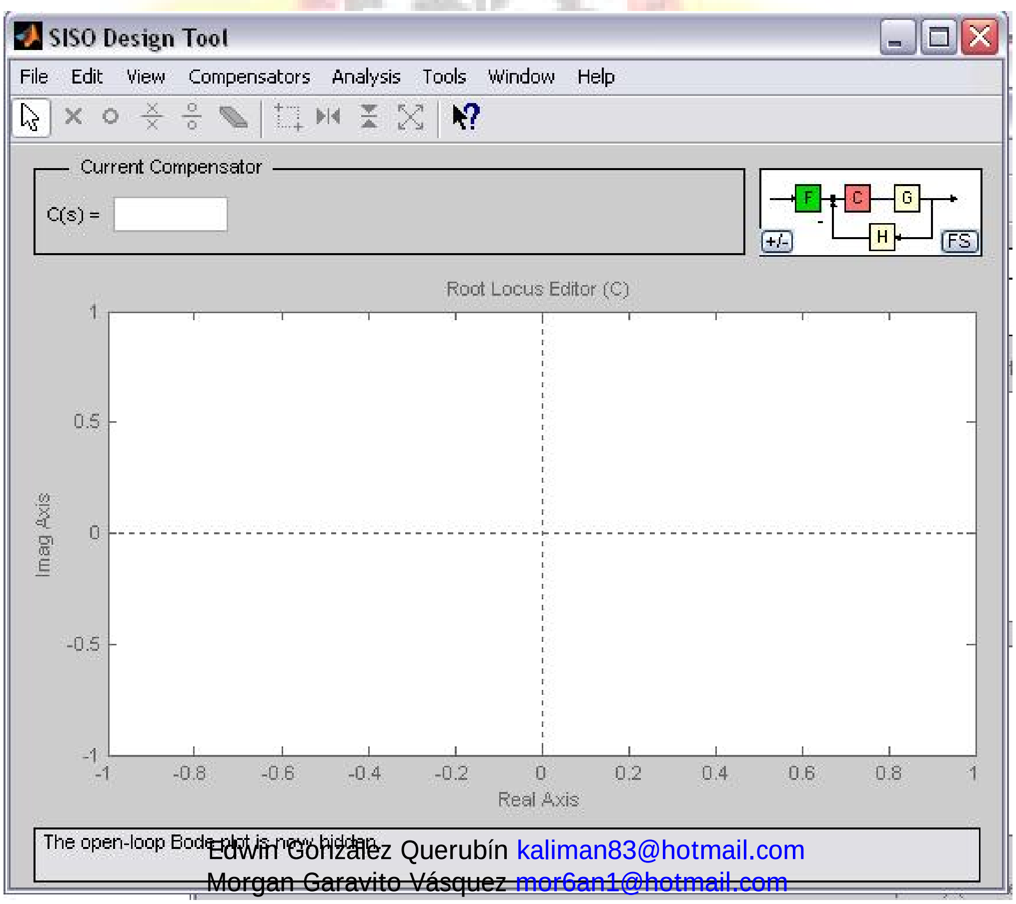
Transfer function:

$$\frac{1}{2e\text{-}010\ s^2 + 0.0002011\ s + 1}$$

IMPLEMENTACION DE LA SISOTOOL DE MATLAB

Después de ejecutar el programa anterior, procedemos a hacer uso de la toolbox de Matlab SISOTOOL.

- Para ingresar a la sisotool de Matlab ejecutamos el comando **<u>sisotool</u>** en la ventana de comandos, después de esto se observa una ventana. Ver figura 6.



Edwin González Querubín kaliman83@hotmail.com
Morgan Garavito Vásquez mor6an1@hotmail.com

Figura 6

- Ubicados en la ventana de la sisotool, nos dirigimos a la pestaña **File - Import** en donde podremos seleccionar de un menú, la función de transferencia del circuito rc o planta que hemos creado en Matlab, lo seleccionamos y luego lo asignamos a **plant**, según la figura 7.
  Por ultimo damos click en OK.

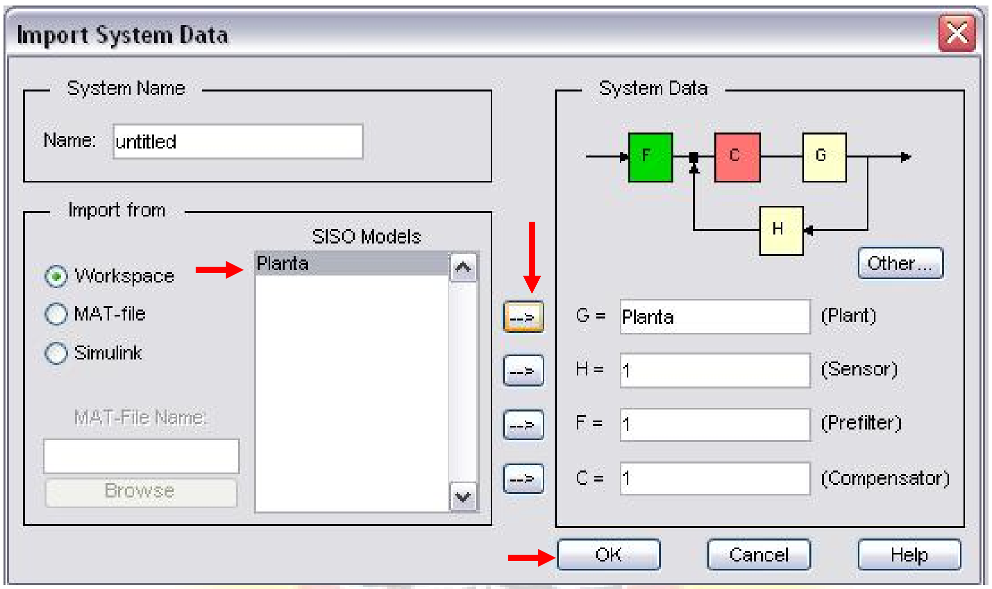
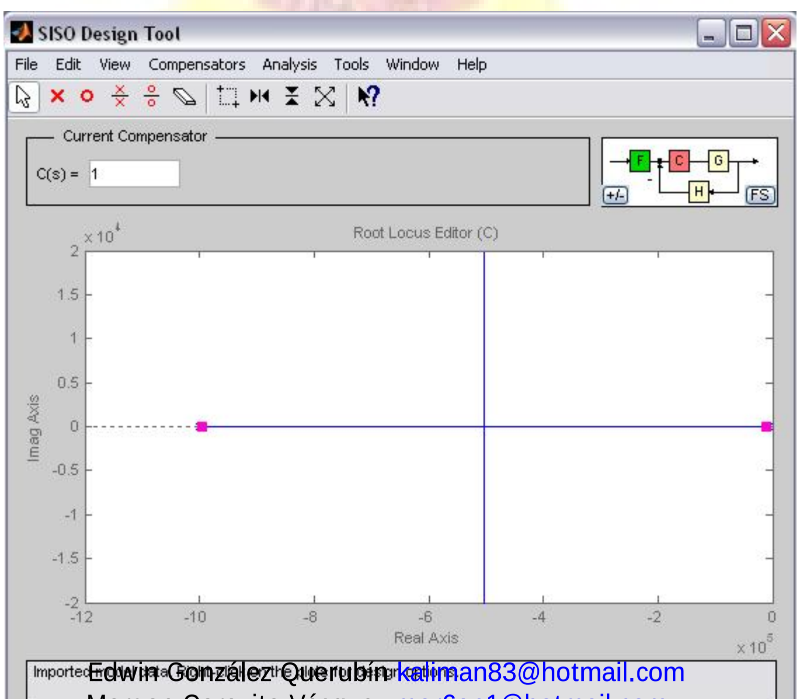


Figura 7

- En la figura 8 se observa el lugar de las raíces.



Edwin González Querubín kaliman83@hotmail.com
Morgan Garavito Vásquez mor6an1@hotmail.com

Figura 8

- Para ver el comportamiento del sistema ante una entrada escalón, seleccionamos la pestaña **Analysis** – **Response to Step command.** Esta nos muestra la respuesta al escalón del sistema en lazo cerrado como se ve en la figura 9 (gráfica azul).

En base a esta gráfica podemos observar que el sistema se estabiliza en:
ts = 0.00039s.

Teniendo en cuenta este tiempo de asentamiento (ts), seleccionamos un tiempo de establecimiento de ts=0.0001seg y un overshoot=20% para el diseño del controlador.

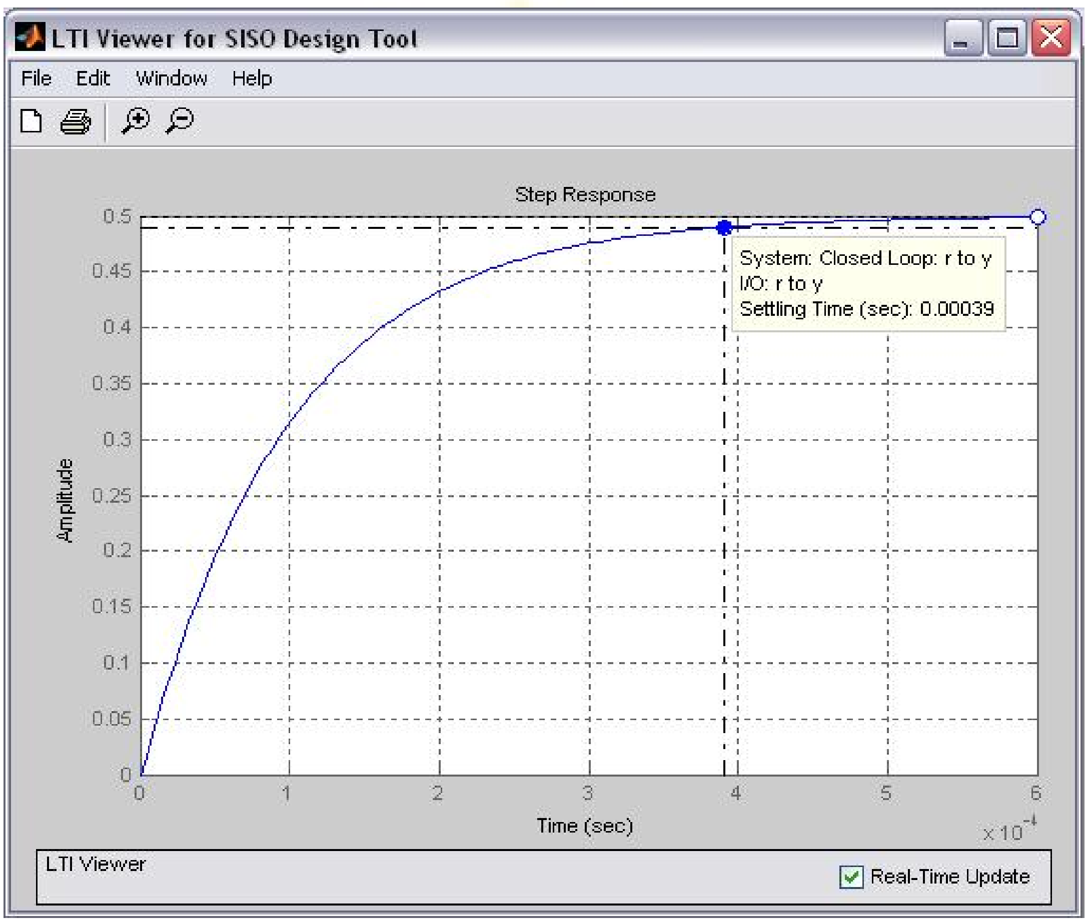


Figura 9

Edwin González Querubín kaliman83@hotmail.com
Morgan Garavito Vásquez mor6an1@hotmail.com

- A continuación procedemos a asignar el tiempo de establecimiento y el porcentaje de overshoot que se planteó anteriormente. Esto se logra haciendo un click con el botón derecho del Mouse sobre la región del lugar de las raíces y siguiendo la secuencia **Desing Constraints - New**. Esto se puede observar en la figura 10.

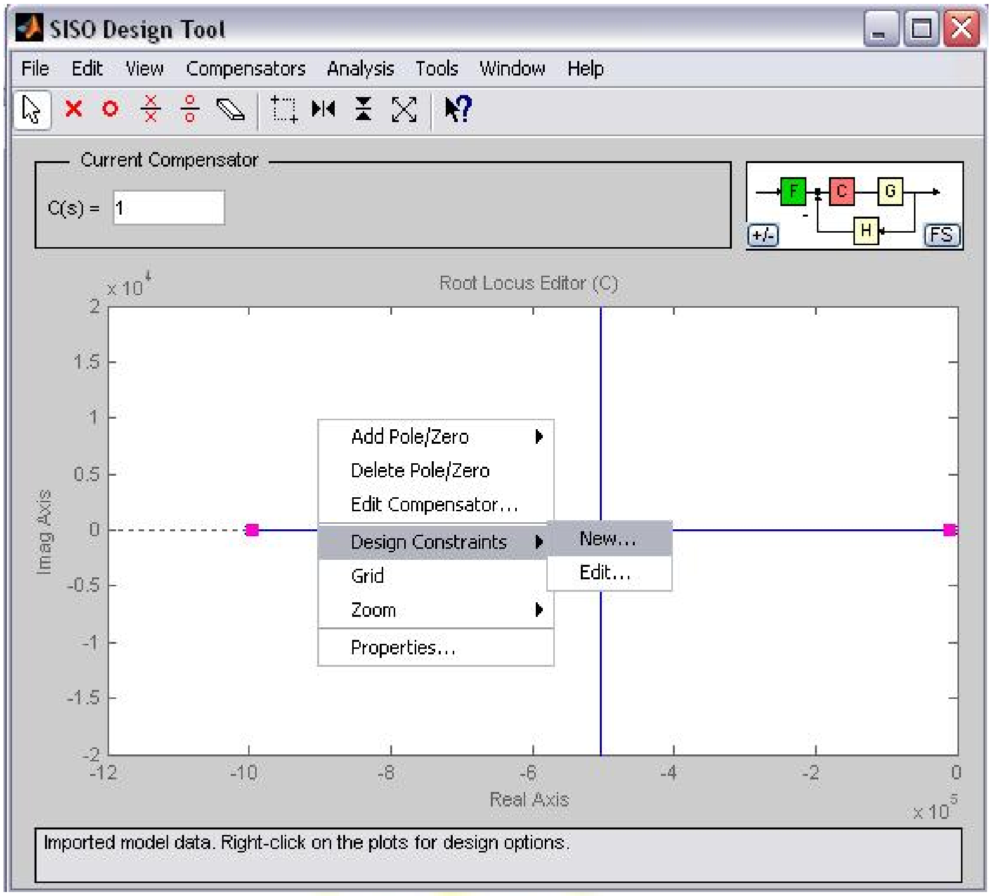


Figura 10

Seleccionamos en el menú desplegable el **Setting time** que es el tiempo de establecimiento y procedemos a asignarlo, por ultimo damos click en OK.

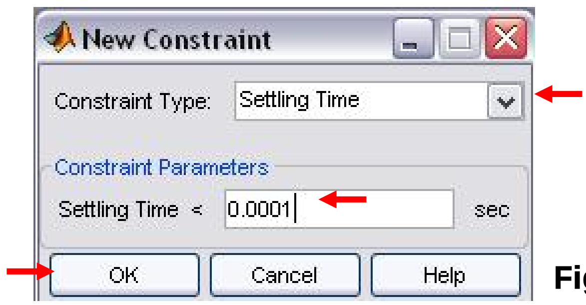


Figura 11

Para asignar el overshoot, repetimos los dos pasos anteriores, pero esta vez seleccionamos en el menú desplegable la opción **Percent overshoot**.

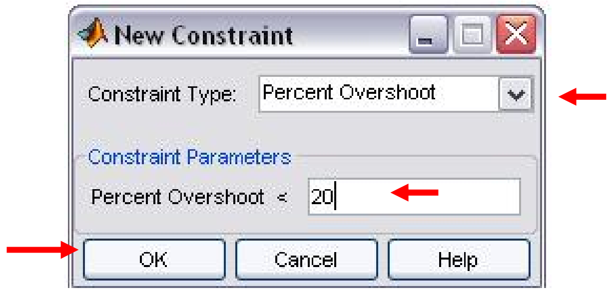


Figura 12

- Cumplidos los pasos anteriores, se puede observar en la ventana del lugar de las raíces el overshoot y el tiempo de establecimiento.

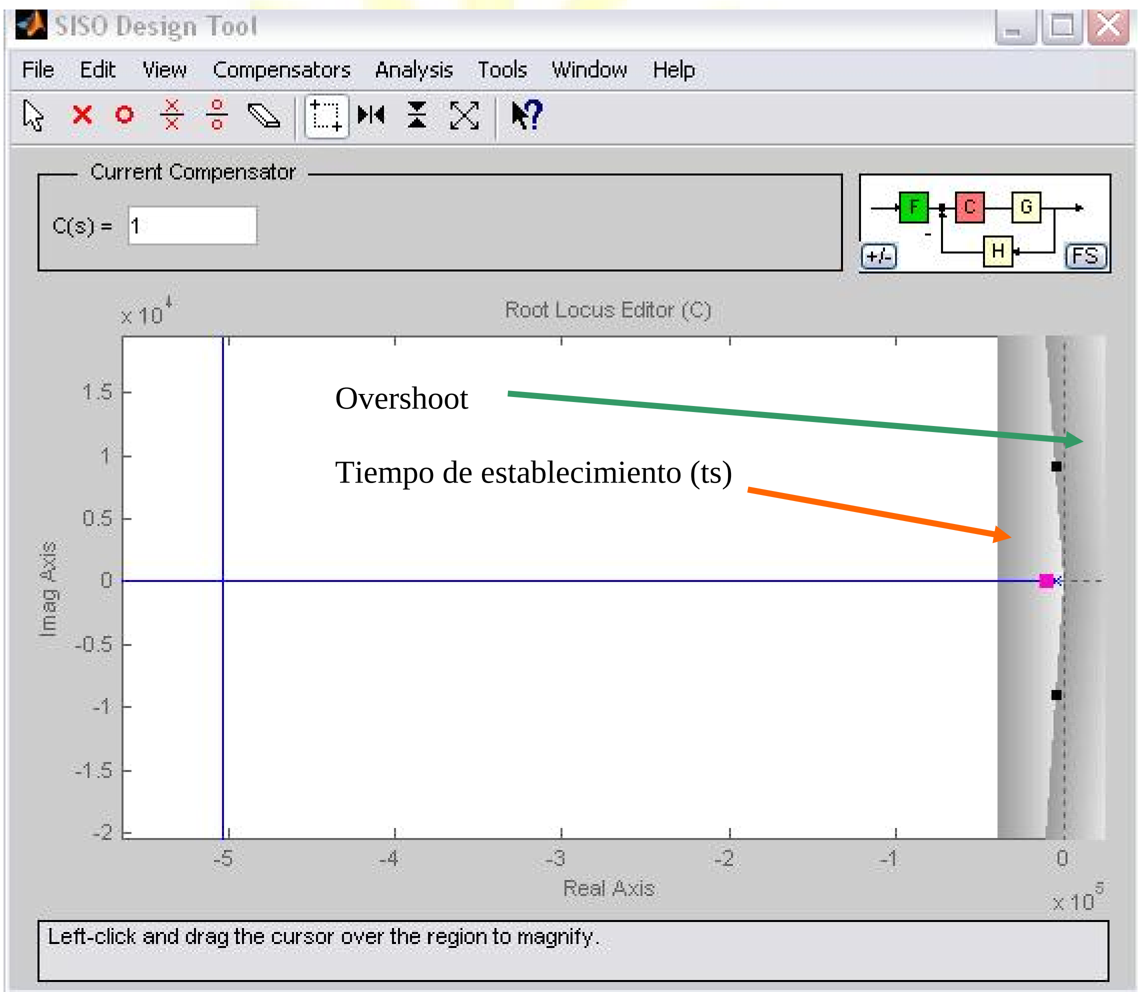


Figura 13

- En la barra de menú damos click en **Compensators** - **Format, Options** y elegimos la opción **Zero/Pole/Gain** luego clic en **Apply, Ok**, esto con el fin de visualizar el controlador en formato de ganancia, polos y ceros.

- En la figura 13, nos dirigimos a la barra de menú y seleccionamos la opción **Compensators – Edit – C**, como se observa en la figura 14.

Debido a que el compensador PID consta de dos polos y dos ceros, donde uno de sus polos esta ubicado en el origen, realizamos los siguiente pasos.

En la figura 14 damos doble-click en la opción **Add Real Zero** y asignamos dos ceros arbitrariamente, después de esto damos doble-click en **Add Real Pole** asignando un polo en el origen y otro de forma arbitraria, asignamos una ganancia del controlador en la casilla **Gain;** por ultimo damos click en **Apply** y **OK**. (Los polos ubicados en la parte real positiva hacen inestables el comportamiento del sistema). Ver figura 14.

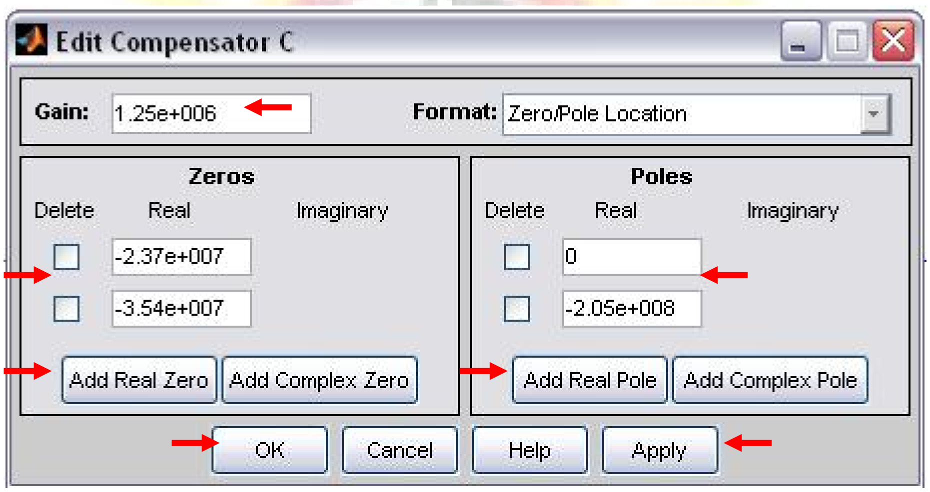


Figura 14

- En la figura 15, se puede observar en el lugar de las raíces los polos y ceros asignados al controlador C(s), además de esto en la parte superior izquierda se puede observar la ecuación del controlador asi como la ganancia de este.

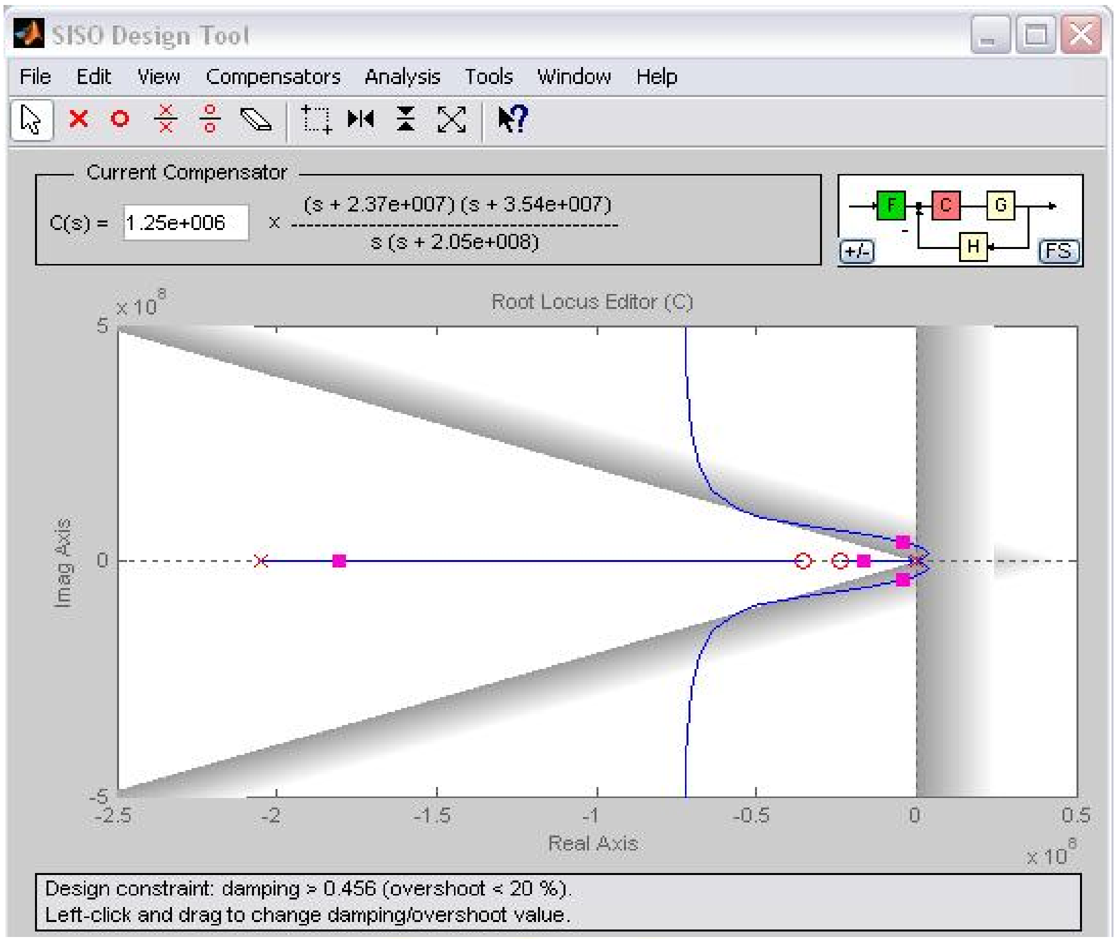


Figura 15

Dando click en la opción **Analisys – Response to step command** podemos observar la respuesta al escalón del sistema de lazo cerrado con los parámetros del compensador anteriormente seleccionados. Se puede ver que el sobre paso es aproximadamente el 80% y ts de 9,07e-7 segundos, por lo que el controlador aún no cumple con las condiciones de diseño antes asignadas.

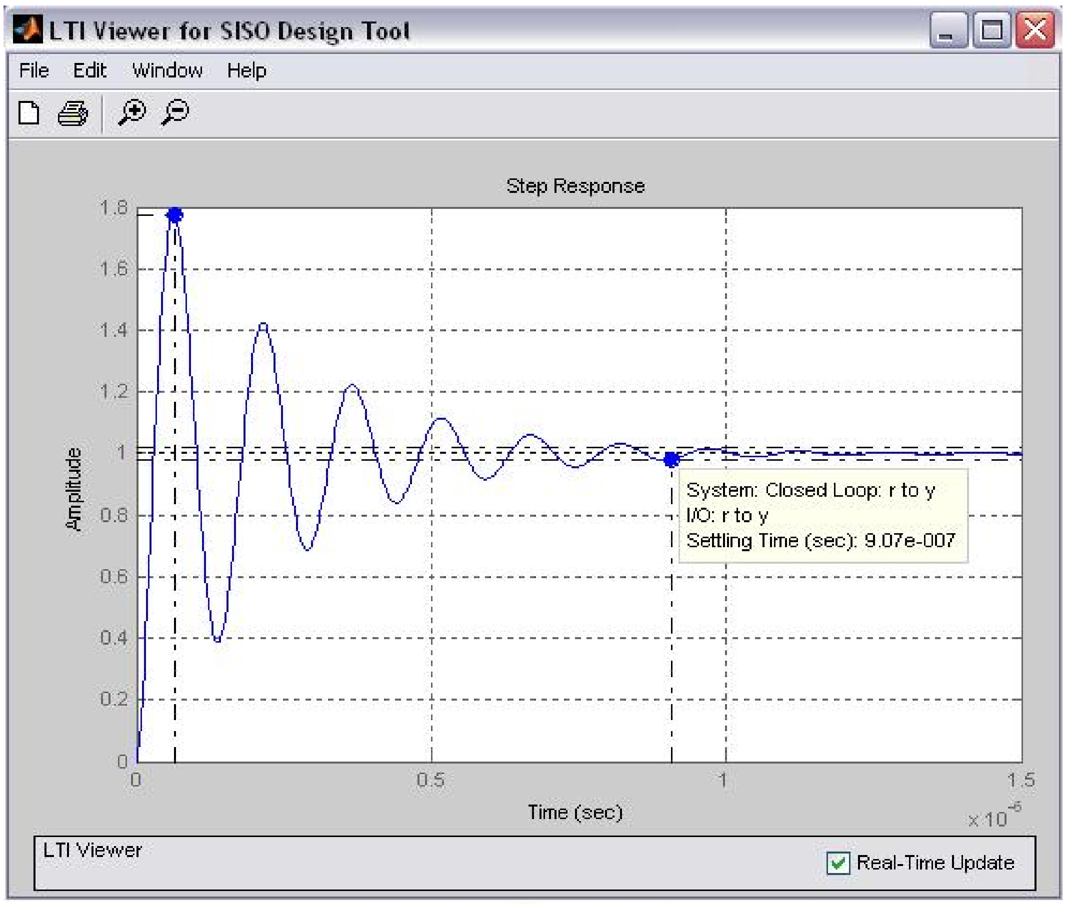


Figura 16

- La idea de ubicar los dos ceros y un polo en posiciones arbitrarias, se debe a que por medio de la sisotool que ofrece un entorno gráfico mediante el cual seleccionando estos ceros y polos y arrastrándolos a través del eje real, podemos modificar la línea de acción de los polos dominantes del sistema.

El objetivo al que se quiere llegar para obtener el controlador deseado, es el de posicionar los ceros y el polo, ubicados arbitrariamente, de tal forma que la línea de acción pase justamente en el lugar donde se cruzan el overshoot y el tiempo de establecimiento. De esta forma podremos arrastrar los polos dominantes a ese punto, en el cual el controlador cumple con las condiciones de diseño establecidas.

En la figura 17 se puede observar los parámetros mencionados anteriormente, aunque el controlador no es viable debido a que la línea de acción de los polos dominantes no pasa por el punto entre el overshoot y ts.

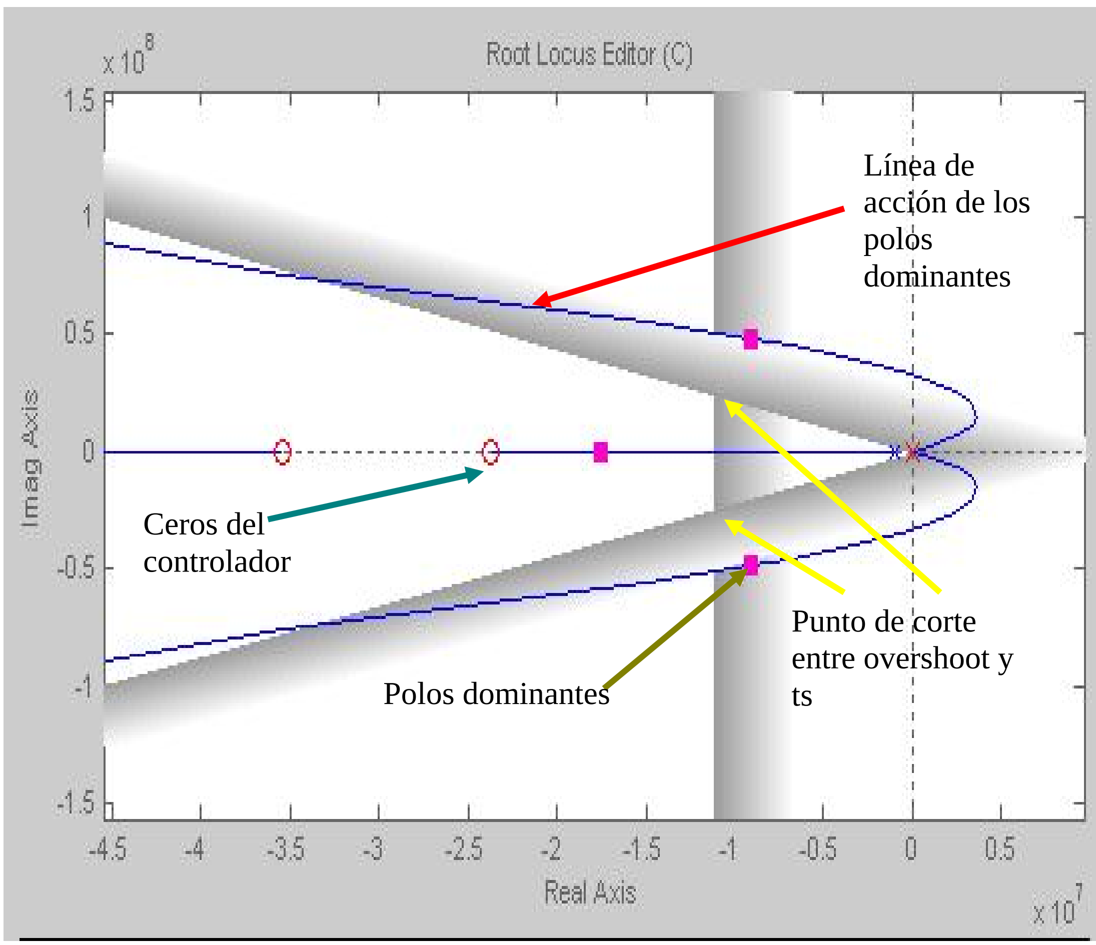


Figura 17

- Después de varias pruebas de ubicación del polo y ceros del controlador logramos hacer coincidir la línea de acción de los polos dominantes con el punto de corte entre el overshoot y el ts.

Inmediatamente ubicamos los polos dominantes en dicho punto, como se observa en la figura 18 y 19.

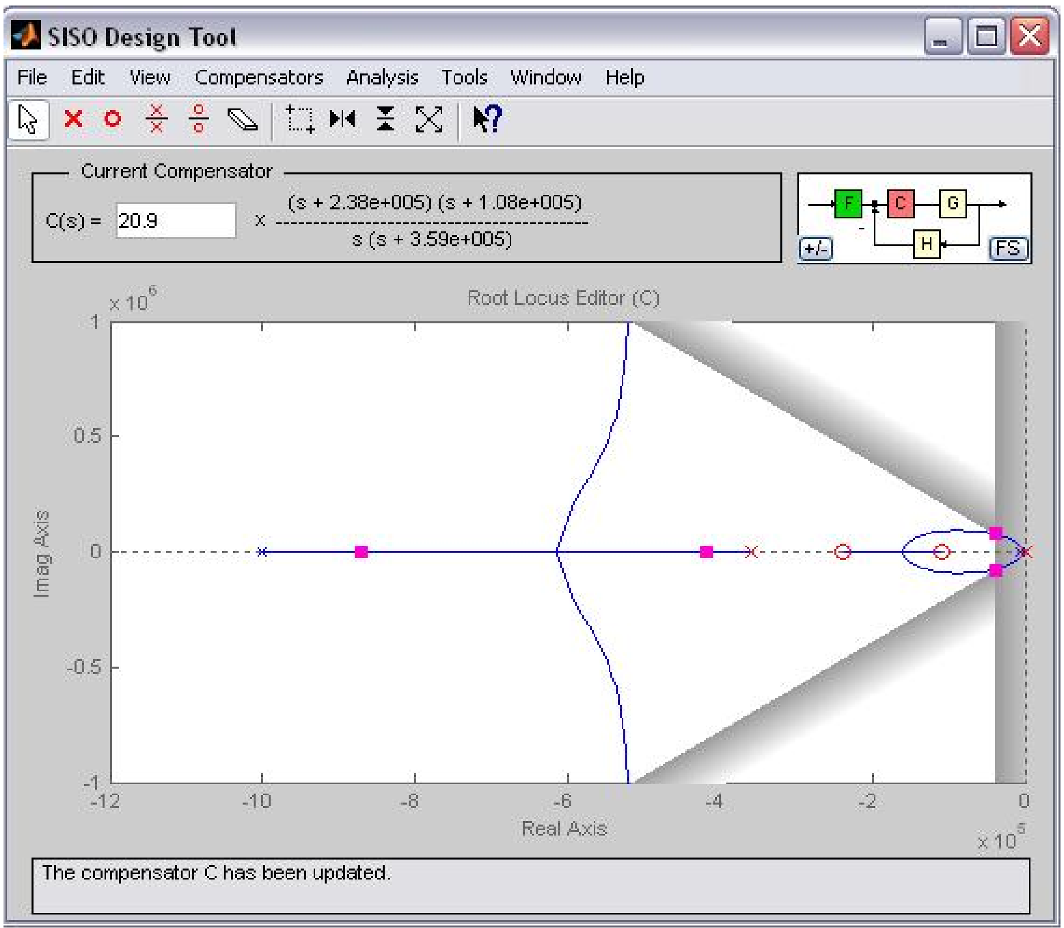


Figura 18

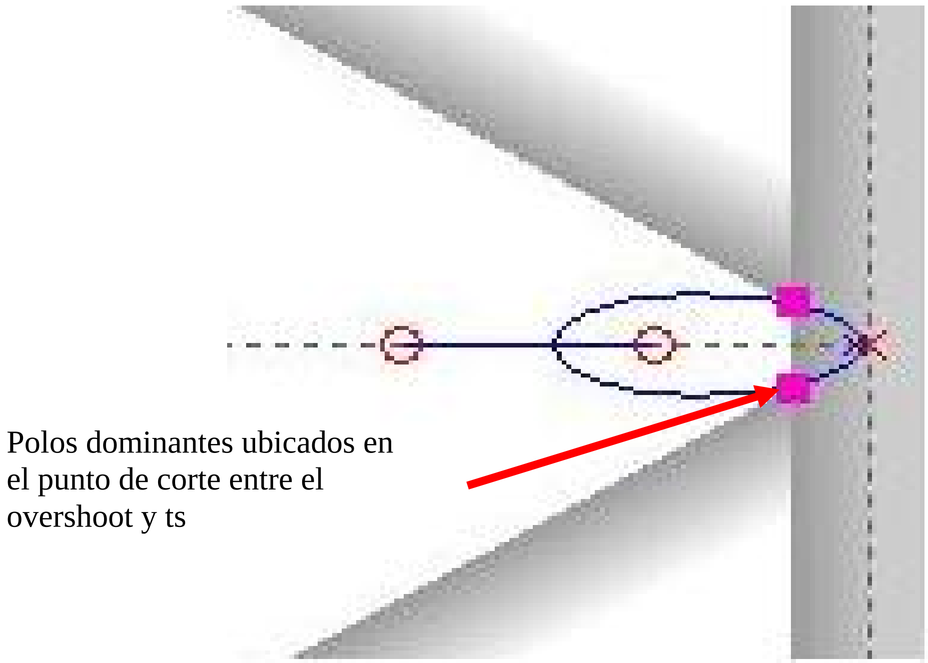


Figura 19 (Detalle de la figura 18)

Como podemos observar en la ecuación del controlador, fue necesario ubicar los dos ceros arbitrarios en: **-2.38e005** y **-1.08e005** y el polo en: **-3.59e005**, por lo cual, al ubicar los polos dominantes en el punto de corte del overshoot y ts, se obtuvo una ganancia de controlador de **20.9**.

- Nos dirigimos nuevamente a la pestaña **Analysis – Response to Step command** para visualizar la respuesta al escalón del sistema de lazo cerrado con el controlador anteriormente obtenido.

En las figuras 20 y 21 podemos observar que el sobrepaso es del 30.8% y el tiempo de establecimiento ts de 8.85e-5seg, que son significativamente cercanos al 20% del overshoot y a los 0.0001seg. de tiempo de establecimiento propuestos para el controlador. Por lo tanto, optamos por implementar este controlador en el circuito rc o planta.

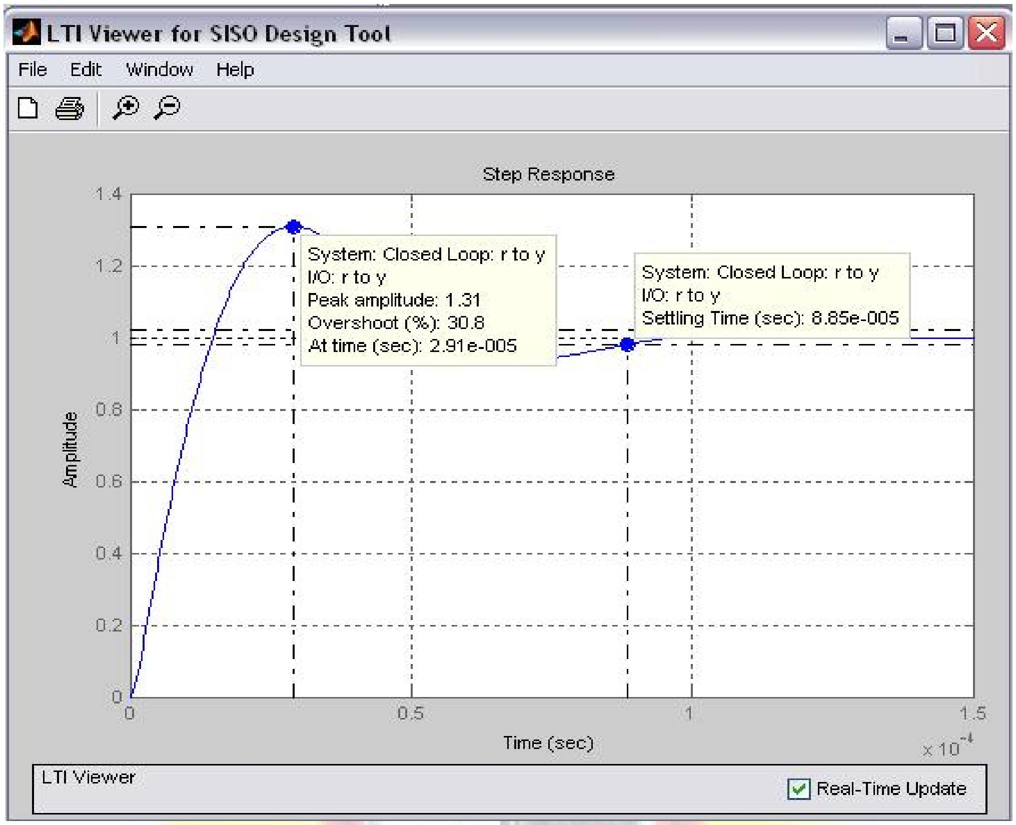


Figura 20

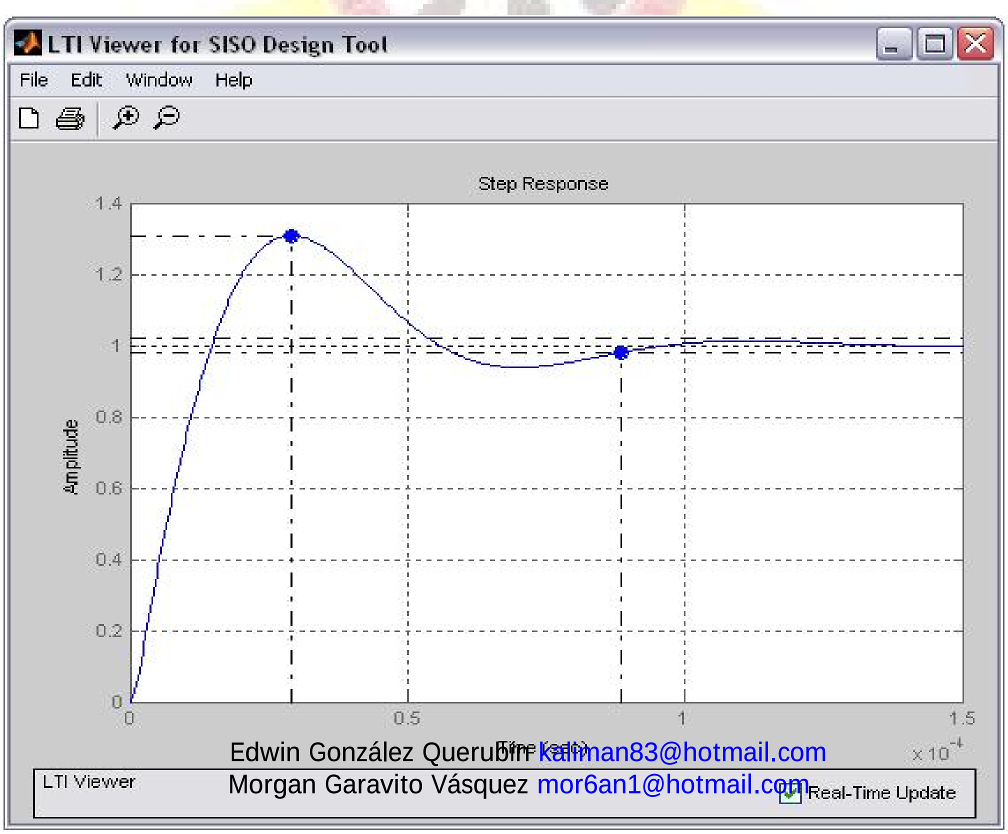


Figura 21

- Para obtener los valores de los elementos del circuito PID mediante Matlab, debemos exportar el compensador obtenido anteriormente al workspace de Matlab de la siguiente manera.

En la sisotool, damos click en **File – Export**, inmediatamente se abre la ventana que se observa en la figura 22, seleccionamos Compensator **C** y damos click en **Export to workspace**.

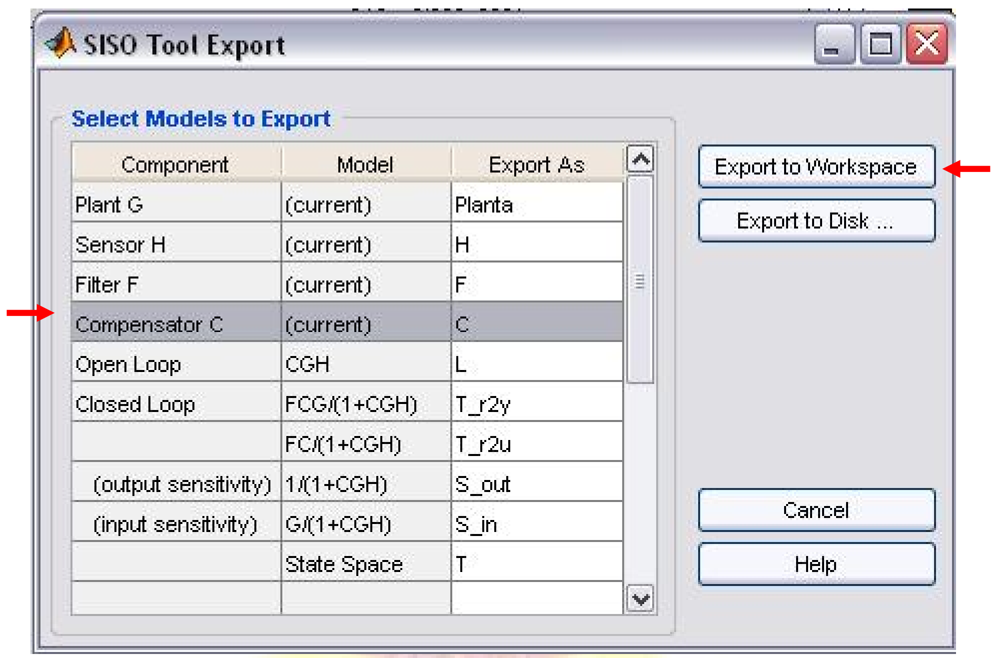


Figura 22

- Hecho el paso anterior, nos ubicamos en el workspace de Matlab y visualizamos el controlador escribiendo en la ventana de comandos **C** y presionamos la tecla Enter.

Como se podrá ver el formato de la ecuación del controlador esta en zpk.

Zero/pole/gain:

$$\frac{20.8725\ (s+2.378e005)\ (s+1.08e005)}{s\ (s+3.594e005)}$$

- Para visualizar el controlador en términos de potencias escribimos en el workspace tf(C).

Transfer function:

```
20.87 s^2 + 7.218e006 s + 5.361e011
-------------------------------------------------
              s^2 + 359400 s
```

- Remitiéndonos a la ecuación del circuito PID obtenida, podemos observar que es similar a la ecuación del PID obtenido mediante la sisotool. Esto se debe a que ambos controladores se caracterizan por tener dos ceros y dos polos, uno de ellos ubicado en el origen.

$$C(s)=\frac{\left(\frac{Rd}{Rc}+\frac{Rh}{Rg}\right)S^{\wedge}2+\left(\frac{Rd}{RcRgCd}+\frac{1}{RfCc}\right)S+\frac{1}{RfRgCcCd}}{S\left(S+\frac{1}{RgCd}\right)}=\frac{20.87\ s^{\wedge}2+7.218e006\ s+5.361e011}{s(s+359400)}$$

Por ahora, expresaremos el controlador obtenido mediante la sisotool de la siguiente forma:

$$C(s)=\frac{A1S^{\wedge}2+B1S+C1}{S(S+D1)}$$

Para hallar los valores de los elementos del circuito PID, aplicamos el método de igualación de términos de ambos controladores.

9) $A1=\frac{Rd}{Rc}+\frac{Rh}{Rg}$ **10)** $B1=\frac{Rd}{RcRgCd}+\frac{1}{RfCc}$ **11)** $C1=\frac{1}{RfRgCcCd}$

12) $D1=\frac{1}{RgCd}$

Debido a que contamos con un sistema de 4 ecuaciones y 7 incógnitas, es necesario asumir un valor de componente por etapa del circuito del PID, para así calcular los demás parámetros del circuito.

En este caso asumimos, valores de condensadores y resistencia para Cc, Cd y Rc:

Cc=0.1nf	Componente de la etapa integral
Cd=0.1nf	Componente de la etapa derivativa
Rc=220Ω	Componente de la etapa proporcional

- Despejamos Rg de la ecuación **12).**

13) $Rg = \frac{1}{D1Cd}$

- Despejamos Rf de la ecuación **11)** y reemplazamos en esta, la ecuación **13)**.

$$Rf = \frac{1}{RgCcCdC1} \qquad Rf = \frac{D1Cd}{CcCdC1} \qquad \textbf{14)}\ Rf = \frac{D1}{CcC1}$$

- Despejamos Rd de la ecuación **10)**, y reemplazamos en esta **13)** y **14)**.

$$Rd = (B1 - \frac{1}{RfCc})RcRgCd \qquad Rd = (B1 - \frac{CcC1}{D1Cc})\frac{RcCd}{D1Cd} \qquad Rd = \frac{B1Rc}{D1} - \frac{C1Rc}{D1\text{^}2}$$

15) $Rd = (B1 - \frac{C1}{D1})\frac{Rc}{D1}$

- Despejamos Rh de la ecuación **9)**, y reemplazamos en esta la ecuación **15)** y **13)**.

$$Rh = (A1 - \frac{Rd}{Rc})Rg \qquad Rh = (A1 - \frac{(B1 - \frac{C1}{D1})\frac{Rc}{D1}}{Rc})\frac{1}{D1Cd}$$

$$Rh = (A1 - (\frac{B1}{D1} - \frac{C1}{D1\wedge 2}))\frac{1}{D1Cd}$$

16) $Rh = (A1 + \frac{C1}{D1\wedge 2} - \frac{B1}{D1})\frac{1}{D1Cd}$

CALCULO DE LOS VALORES DE LOS ELEMENTOS DEL CIRCUITO PID

- Para extraer los parámetros A1 B1 C1 y D1 del controlador hallado en la sisotool, se implementó el siguiente código:

```
%============================================================
%Extracción de los parámetros del controlador C(s) obtenido
%mediante la sisotool
%============================================================
%       A1*S^2+B1*S+C1
% C(s)= --------------
%          S*(S+D1)
%Obtenemos el numerador y el denominador del compensador
[n2,d2]=tfdata(C,'v');
%Extraemos los parámetros de este
A1=n2(1,1)
B1=n2(1,2)
C1=n2(1,3)
D1=d2(1,2)
```

Lo que nos dá por resultado:

A1= 20.8725 B1 = 7.2177e+006 C1 = 5.3606e+011 D1 = 359400

- En las ecuaciones **15)** $Rd = (B1 - \frac{C1}{D1})\frac{Rc}{D1}$ y **16)** $Rh = (A1 + \frac{C1}{D1^2} - \frac{B1}{D1})\frac{1}{D1Cd}$

En las partes $(B1 - \frac{C1}{D1})$ y $(A1 + \frac{C1}{D1^2} - \frac{B1}{D1})$ observamos que el controlador hallado en la sisotool, debe cumplir con dos condiciones para que las resistencias Rd y Rh nos den valores reales o que se puedan conseguir en el mercado; ya que hasta el momento no se consiguen resistencias negativas.

Condición 1 para que Rd sea positiva: $B1 > \frac{C1}{D1}$

Condición 2 para que Rh sea positiva: $A1 + \frac{C1}{D1^2} > \frac{B1}{D1}$

Para comprobar esto utilizaremos el siguiente código:

```
%==============================================================
%Condiciones que se deben cumplir en el compensador para que
%los valores de las resistencias Rd y Rh sean reales
%==============================================================
fprintf('Las condiciones deben ser positivas\n')
Condicion_1= B1-C1/D1
Condicion_2= A1+C1/(D1^2)-B1/D1
```

Al correr el programa nos muestra por resultado:

```
Condicion_1 =
          5.7262e+006


Condicion_2 =
          4.9399
```

Ambos resultados fueron positivos, por lo tanto vamos por buen camino.

En caso de que las condiciones o alguna de estas no se cumplan, debemos regresar a la sisotool y modificar el controlador ubicando los polos y ceros arbitrarios en otras posiciones, hasta que se obtenga un controlador con parámetros que satisfagan los parámetros de diseño y además las condiciones mencionadas anteriormente.

- Cumplido el paso anterior implementamos las ecuaciones **13), 14), 15) y 16)** en el siguiente código, para obtener los valores de los elementos del circuito.

Recordemos que anteriormente asignamos valores a unos componentes del circuito:

Cc=0.1nf	Componente de la etapa integral
Cd=0.1nf	Componente de la etapa derivativa
Rc=220Ω	Componente de la etapa proporcional

Además de esto, expresaremos la función de transferencia del circuito PID de la siguiente forma:

$$C(s) = \frac{A2*S^2+B2*S+C2}{S*(S+D2)}$$

```
%==============================================================
%Función del Compensador PID con operacionales
%==============================================================
%      (Rd/Rc+Rh/Rg)S^2 + (Rd/(RcRgCd)+1/(RfCc))S + 1/(RfRgCcCd)
% C(s) = ----------------------------------------------------
%                             S*(S+1/(RgCd))
%
%Expresamos de esta forma la función del PID con
%operacionales
%       A2*S^2+B2*S+C2
% C(s) = -----------------
%          S*(S+D2)
%Asignamos un valor, solo a un componente de cada parte del
%circuito
```

```
%(Parte proporcional, Parte integral y Parte derivativa), con
%el fin de  obtener los valores de los demás componentes
%aplicando igualación de términos de ambos compensadores
Cc=0.1e-9;
Cd=0.1e-9;
Rc=220;
%A2=A1; B2=B1; C2=C1; D2=D1 los parámetros del PID con
%operacionales son iguales a los parametros del PID hallados
en la sisotool
%Cálculo de los demas componentes del circuito con
%operacionales
fprintf('Resistencias calculadas:\n')
Rd=(B1-C1/D1)*Rc/D1
Rf=D1/(C1*Cc)
Rg=1/(D1*Cd)
Rh=(A1+C1/(D1^2)-B1/D1)/(D1*Cd)
```

Lo que nos da por respuesta:

Rd = 3.5052e+003   Rf = 6.7045e+003   Rg = 2.7824e+004   Rh = 1.3745e+005

osea:

Rd = 3.5052kΩ
Rf = 6.7045kΩ
Rg = 27.824kΩ
Rh = 137.45kΩ

- Reemplazando estos valores en la función de transferencia del circuito PID usando Matlab, con el fin de comparar ambos controladores y así cerciorarnos que no se cometieran errores en despejes y operaciones algebraicas, ya que ambos controladores deben tener parámetros iguales.

```
%=============================================================
%Cálculo de los parametros del compensador, esto lo hacemos
%con el fin de comparar el compensador de la sisotool con el
%compensador con operacionales (deben ser iguales).
A2=Rd/Rc+Rh/Rg;
B2=Rd/(Rc*Rg*Cd)+1/(Rf*Cc);
C2=1/(Rf*Rg*Cc*Cd);
D2=1/(Rg*Cd);
n3=[A2 B2 C2];
d3=[1 D2 0];
Compensador=tf(n3,d3)
%Función de transferencia del controlador hallado en la
%sisotool en términos de potencias
tf(n2,d2)
```

Nos da por respuesta:

```
Transfer function:

 20.87 s^2 + 7.218e006 s + 5.361e011
-------------------------------------------------
            s^2 + 359400 s


Transfer function:

 20.87 s^2 + 7.218e006 s + 5.361e011
-------------------------------------------------
            s^2 + 359400 s
```

Comparando ambos compensadores, nos damos cuenta que son iguales, lo que indica que no hubo error alguno en los despejes y operaciones algebraicas.

ANALISIS Y SIMULACION DE LAS DIFERENTES PARTES DEL SISTEMA DE LAZO CERRADO

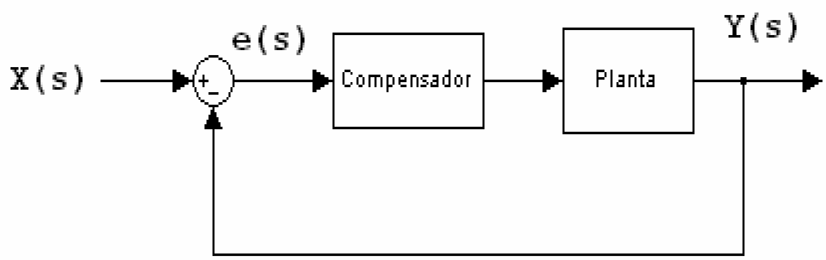


Figura 23

En la figura 23 se observa el diagrama de bloques del sistema de lazo cerrado compensado, donde **X(s)** es la entrada, **e(s)** es el error y **Y(s)** es la salida del sistema.

- Para simular las diferentes respuestas del sistema, creamos un tren de pulsos que va a ser la entrada del sistema.

```
%==========================================================
%Generamos una onda cuadrada, que va a ser la señal de
%entrada, con un periodo de 1/600, duración de 5e-3 seg y con
%muestras cada (1e-3)/100 segundos.
[u,t] = gensig('square',1/600,5e-3,(1e-3)/100);
```

- La ecuación **Y(s)/X(s)**, que denominaremos **src** (sistema retroalimentado compensado) es:

```
%Ecuación del sistema retroalimentado compensado
src=Compensador*Planta/(1+Compensador*Planta);
```

- Al excluir el compensador de la anterior ecuación, obtenemos la función de transferencia del sistema de lazo cerrado no compensada. Por lo tanto, a esta ecuación la vamos a llamar **srnc** (sistema retroalimentado no compensado).

Edwin González Querubín kaliman83@hotmail.com
Morgan Garavito Vásquez mor6an1@hotmail.com

```
%Ecuación del sistema retroalimentado no compensado
srnc=(Planta/(1+Planta));
```

- La ecuación del error del sistema, que es e**(s),** la denominaremos **s_error** (señal de error):

```
%Señal de error
s_error=1-src;
```

- El siguiente código se implementó para graficar las señales `[u,t]`, `srnc`, `src` y `s_error`.

```
%Graficamos todas las señales
subplot(2,2,1),plot(t,u);
axis([0 5e-3 -0.2 1.2]);
title('Señal de entrada'),grid;
subplot(2,2,2),lsim(srnc,u,t);
axis([0 5e-3 -0.2 1.2]);
title('Salida sin compensar'),grid;
subplot(2,2,3),lsim(src,u,t);
axis([0 5e-3 -0.5 1.5]);
title('Salida compensada'),grid;
subplot(2,2,4),lsim(s_error,u,t);
axis([0 5e-3 -1.2 1.2]);
title('Señal de error'),grid;
```

Figura 24(Señal de entrada)

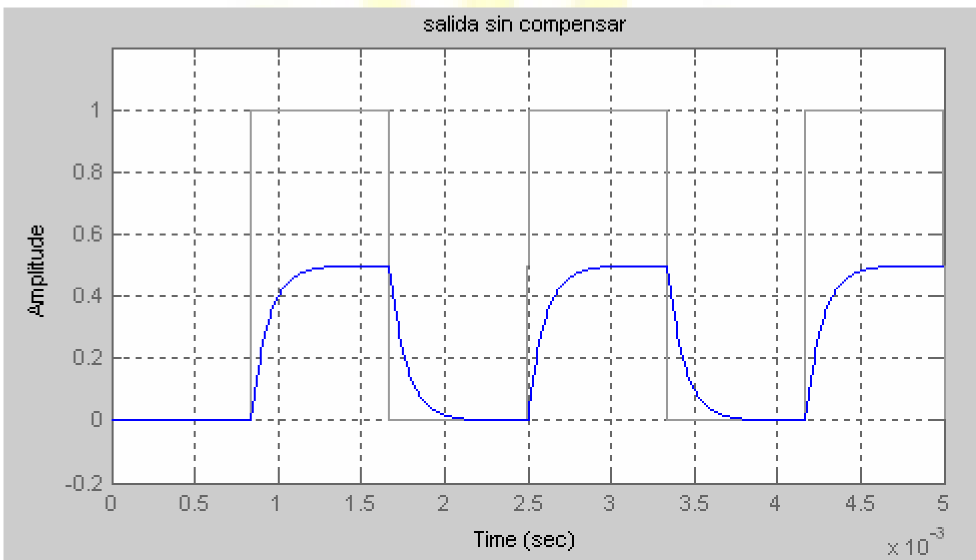


Figura 25(Salida sin compensar - srnc)

Edwin González Querubín kaliman83@hotmail.com
Morgan Garavito Vásquez mor6an1@hotmail.com

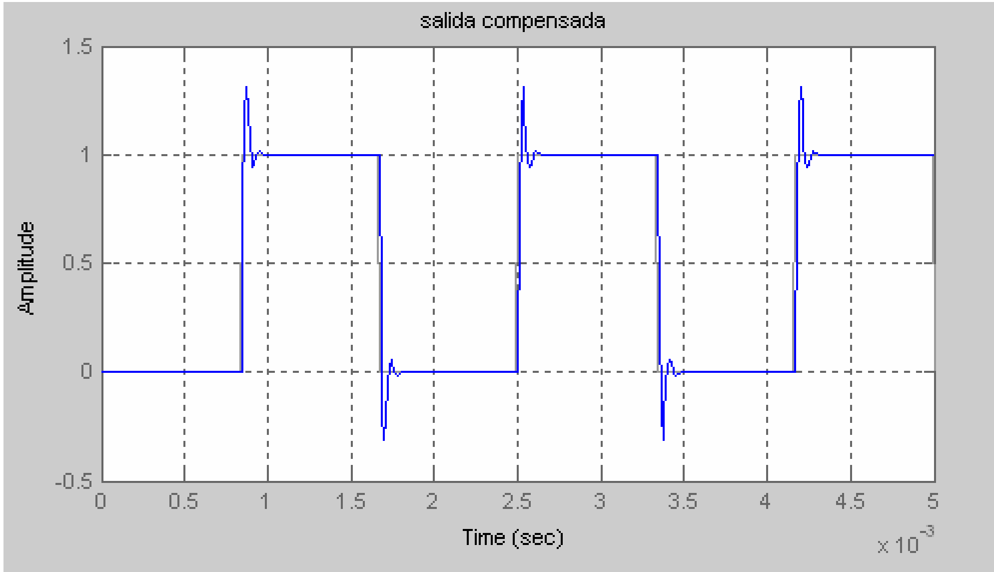


Figura 26(Salida compensada - src)

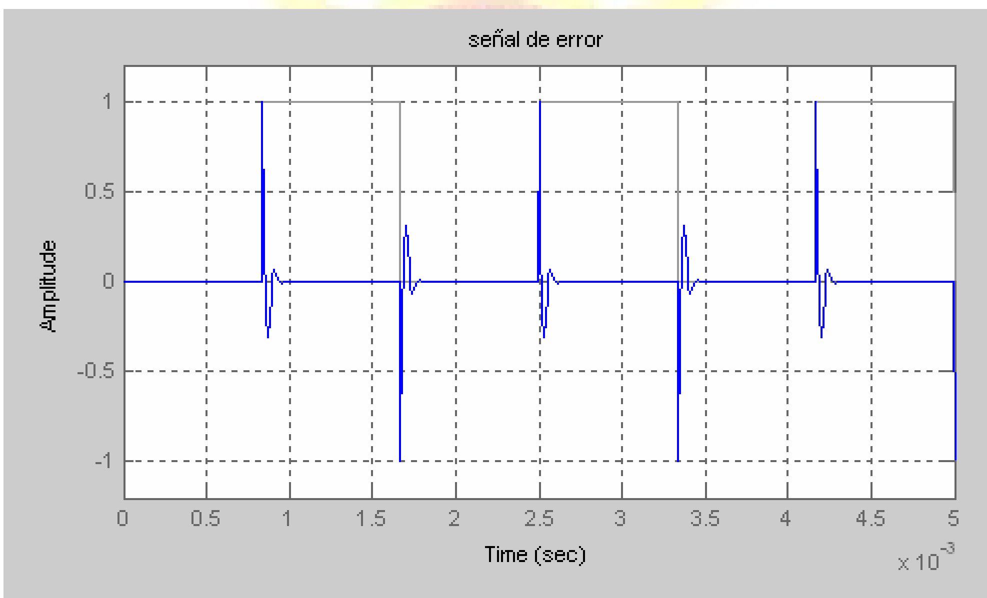


Figura 27(Señal de error – s_error)

Edwin González Querubín kaliman83@hotmail.com
Morgan Garavito Vásquez mor6an1@hotmail.com

Como vemos en las graficas anteriores, la salida compensada es muy similar a la señal de entrada, por lo que el compensador calculado cumple satisfactoriamente los parámetros asignados.

SIMULACION DEL SISTEMA MEDIANTE SIMULINK

Una forma de simular el sistema es por medio de **Simulink**, que nos expresa este mediante diagramas de bloques.

Para efectuar este análisis, obtenido el controlador en la sisotool, nos dirigimos a la opción **Tools – Draw Simulink Diagram.**

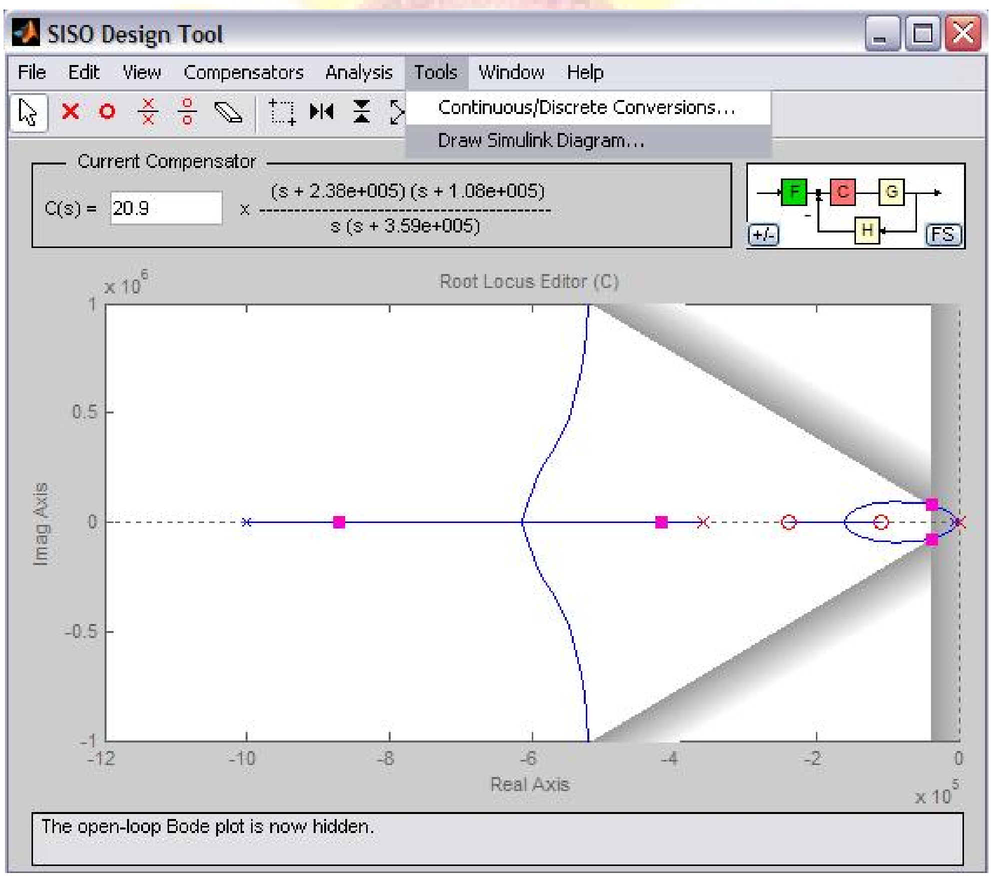


Figura 28

Edwin González Querubín kaliman83@hotmail.com
Morgan Garavito Vásquez mor6an1@hotmail.com

Una vez hecho el paso anterior, se despliega la ventana de Simulink y expresa el sistema en diagramas de bloques como se ve en la figura 29. Este en el bloque **C**, carga automáticamente el controlador diseñado en la sisotool y en el bloque **planta,** la función de transferencia de la planta.

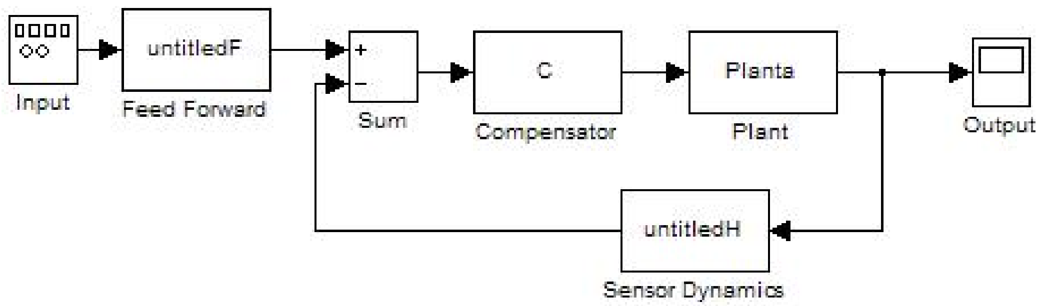


Figura 29

La entrada en el bloque **input,** debemos cargarla dando doble click sobre este mismo, seleccionando el tipo de onda y la frecuencia, en este caso, cuadrada y con una frecuencia de 600Hz, respectivamente. Ver figura 30.

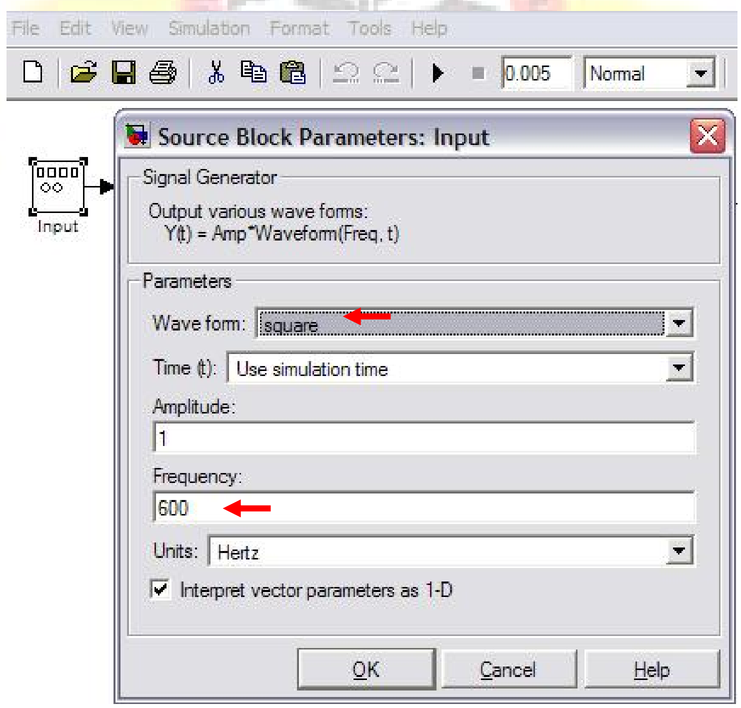


Figura 30

Edwin González Querubín kaliman83@hotmail.com
Morgan Garavito Vásquez mor6an1@hotmail.com

El diagrama de bloques de la figura 29, fue modificado con el objetivo de poder observar las diferentes señales del sistema, como señal de entrada, sistema no compensado, sistema compensado y señal de error, como se observa en la figura 31.

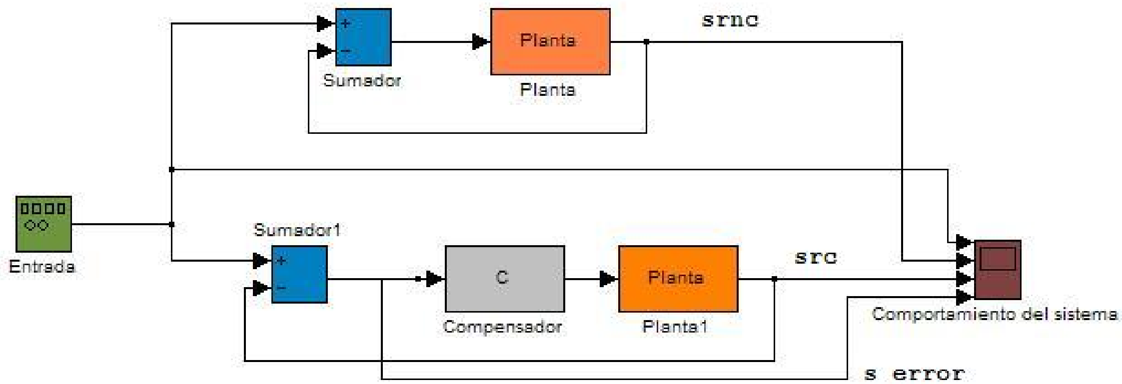


Figure 31

Los bloques untittledF y untittledH, fueron omitidos y además el bloque Output fue modificado para visualizar 4 señales diferentes.

Esto se logra al dar doble click sobre el bloque llamado Output. Una vez hecho esto procedemos a agregar la cantidad de señales que queremos visualizar, en este caso son 4.

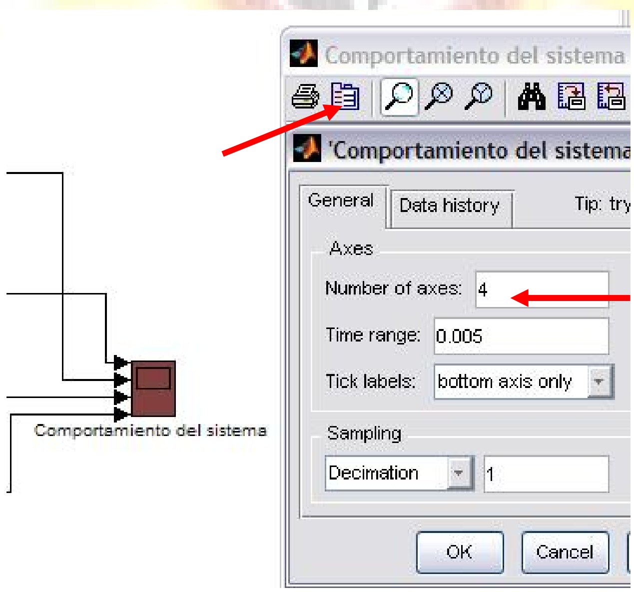
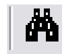


Figure 32

Ahora asignamos el tiempo de duración de la simulación como se observa en la figura 33, en este caso el tiempo es de 0.005seg. Por último nos queda ejecutar el programa dando click en **Simulation – Start** (Ver figura 34). Para visualizar las señales del sistema, damos doble click al bloque llamado Output y en esta misma ventana click en el icono Autoscale. 

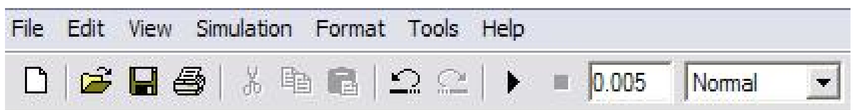


Figura 33

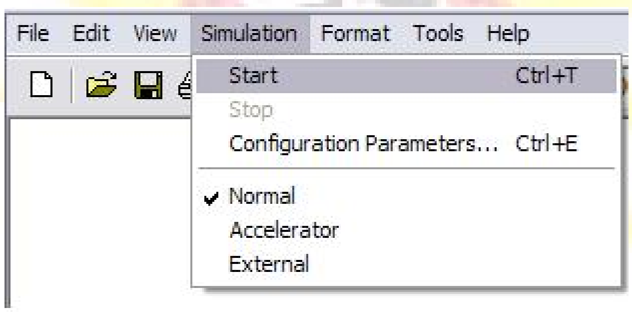


Figura 34

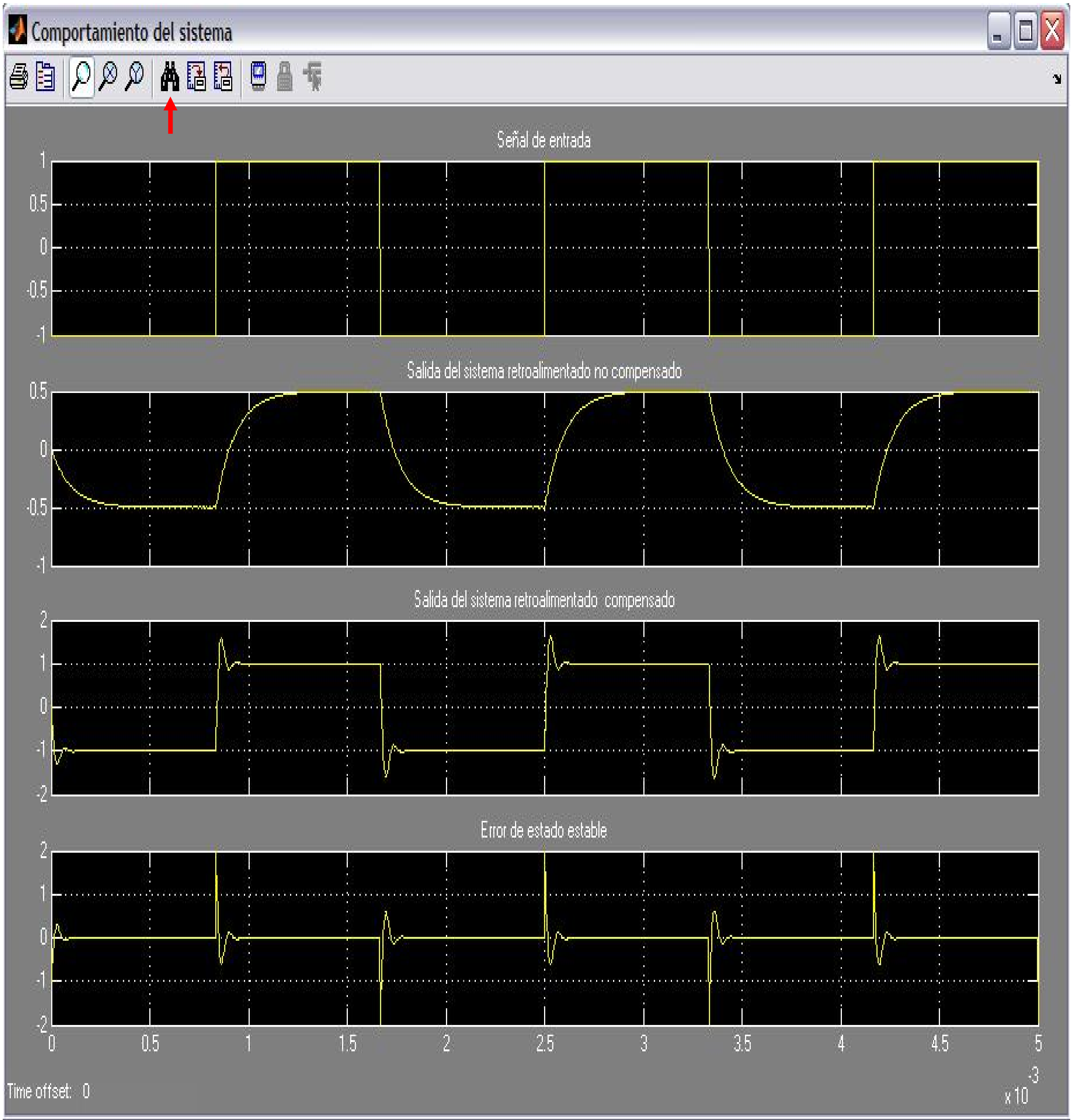


Figura 34 (Simulink)

Edwin González Querubín kaliman83@hotmail.com
Morgan Garavito Vásquez mor6an1@hotmail.com

MONTAJE Y SIMULACION EN CIRCUITMAKER

En la figura 35 se observa el circuito de lazo cerrado, donde se acopló el circuito PID de la figura 2 junto con el de la planta, figura 1. Los valores de sus componentes son los calculados, los asumidos y la resistencia **R,** que tiene el valor de 1MΩ.

Planta

Ra=1kΩ
Rb=10kΩ
Ca=0.2μf
Cb=0.1nf

Componente de la etapa proporcional

Rc=220Ω
Rd=3.5052kΩ

Componente de la etapa integral

Rf=6.7045kΩ
Cc=0.1nf

Componente de la etapa derivativa

Rg=27.824kΩ
Rh=137.45kΩ
Cd=0.1nf

Resto de componentes

R=1MΩ
Amplificadores operacionales= LM741CN

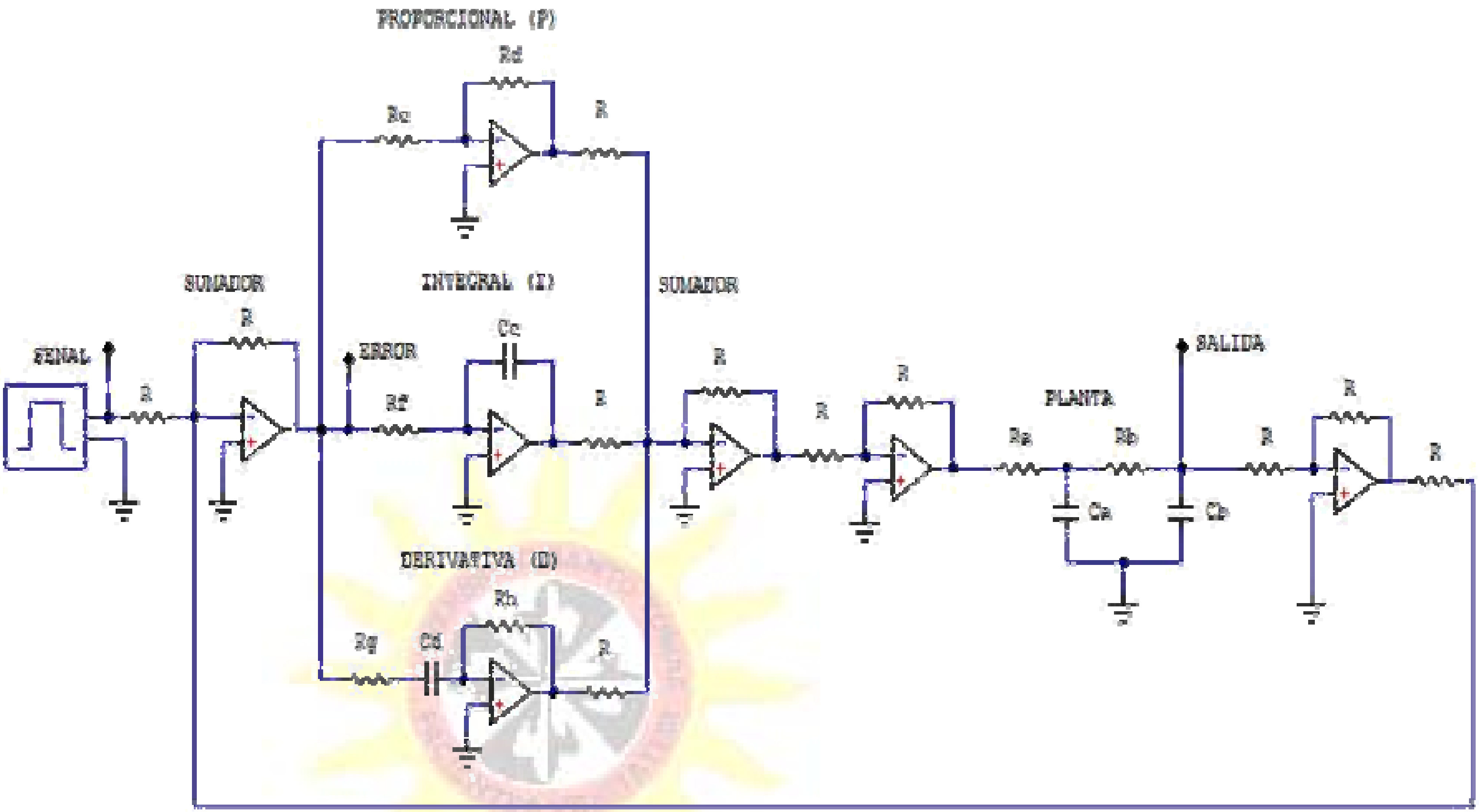


Figura 35(CircuitMaker)

Edwin González Querubín kaliman83@hotmail.com
Morgan Garavito Vásquez mor6an1@hotmail.com

Graficas de la simulación

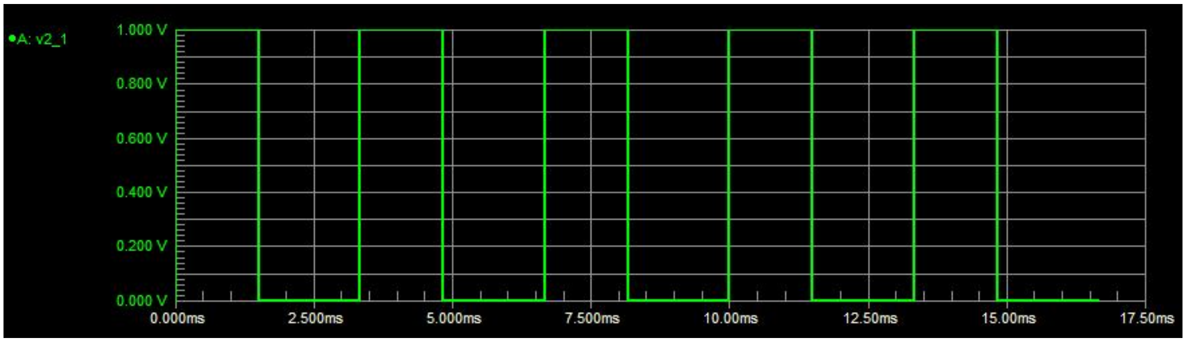


Figura 36 (SENAL)

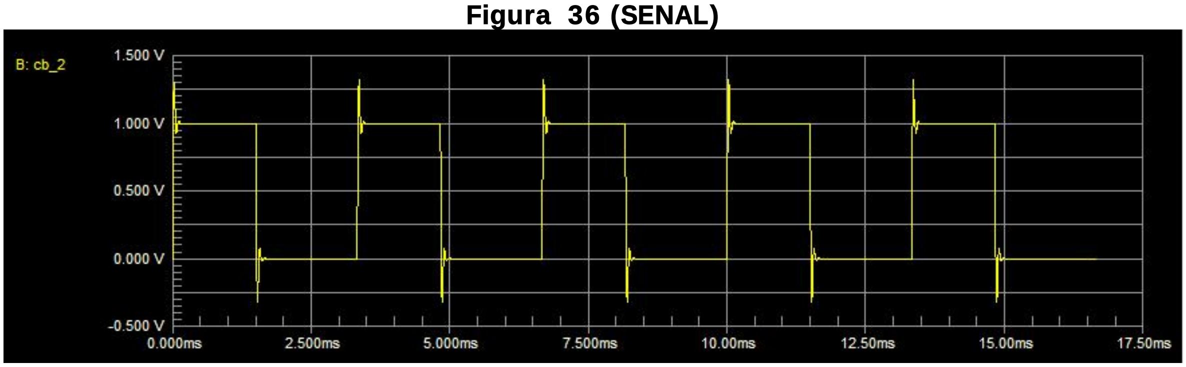


Figura 37 (SALIDA)

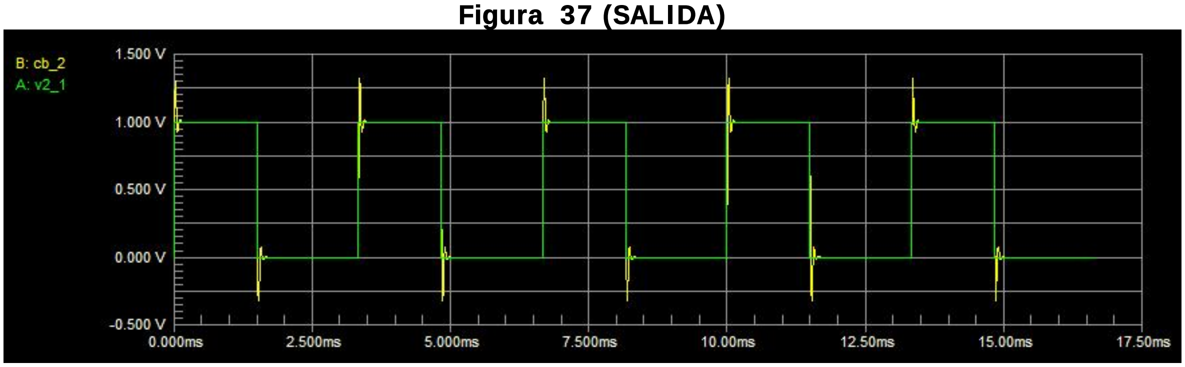


Figura 38 (SENAL y SALIDA superpuestas)

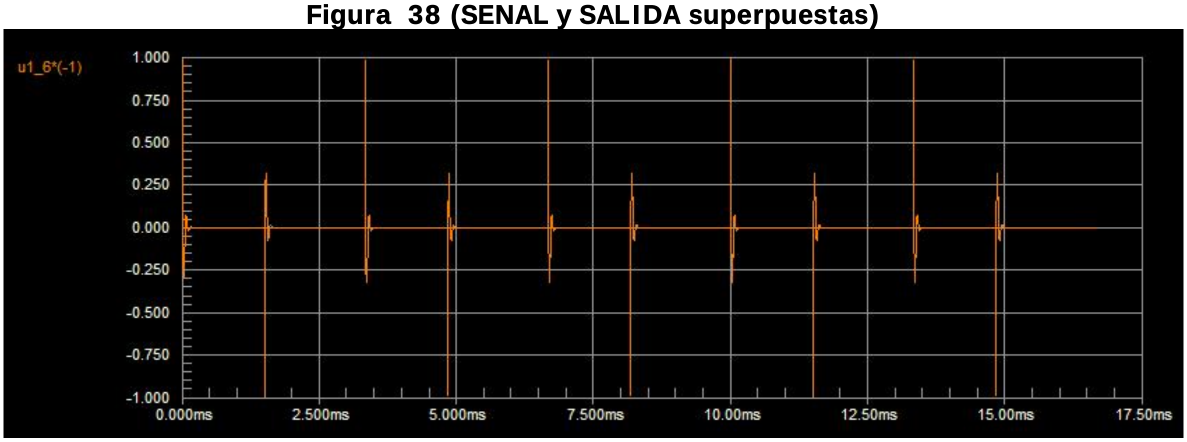


Figura 39 (ERROR*(-1))

CONCLUSIONES

- El diseño de controladores mediante la sisotool, nos evita hacer tediosos cálculos a mano, ya que por medio de su entorno gráfico nos hace las cosas más fáciles, rápidas y además podemos observar el comportamiento del sistema con el controlador actual en forma real.

- Los valores de overshoot y tiempo de establecimiento obtenidos por medio de la sisotool, fueron valores relativamente cercanos a los establecidos para las condiciones de diseño.

- A veces es necesario cambiar los parámetros de diseño del controlador, ya que muchas veces no se puede cumplir con estos, o a veces se requiere de gran cantidad de energía para que se cumplan estas pautas. Idealmente todo se puede hacer, pero en algunos casos debemos tener conocimiento de lo que la tecnología actual nos puede ofrecer.

- Se obtuvieron diferentes tipos de controladores que cumplian con el overshoot y ts (tiempo de establecimiento) asignados, pero sólo unos cuantos cumplían con las condiciones para que las resistencias del circuito PID fueran valores reales.

- Al simular el sistema de lazo cerrado compensado en Matlab, Simulink y Circuit Maker, se pudo observar que la respuesta de salida coincidían en los tres análisis, ya que las gráficas son similares como se observó durante el desarrollo del proyecto.

- Se podría decir que el diseño de compensadores es una labor complicada para un diseñador inexperto, ya que se debe tener mucha paciencia y cuidado a la hora de calcular los valores que este diseño necesite.

BIBLIOGRAFIA

Ingeniería de Control Moderna
Katsuhiko Ogata
Tercera Edición. Prentice-Hall, 1.998.

Sistemas de Control Automático
Kuo, Benjamin C.
Prentice Hall Hispanoamericana S.A

Datasheet
LM741CN

ANEXOS

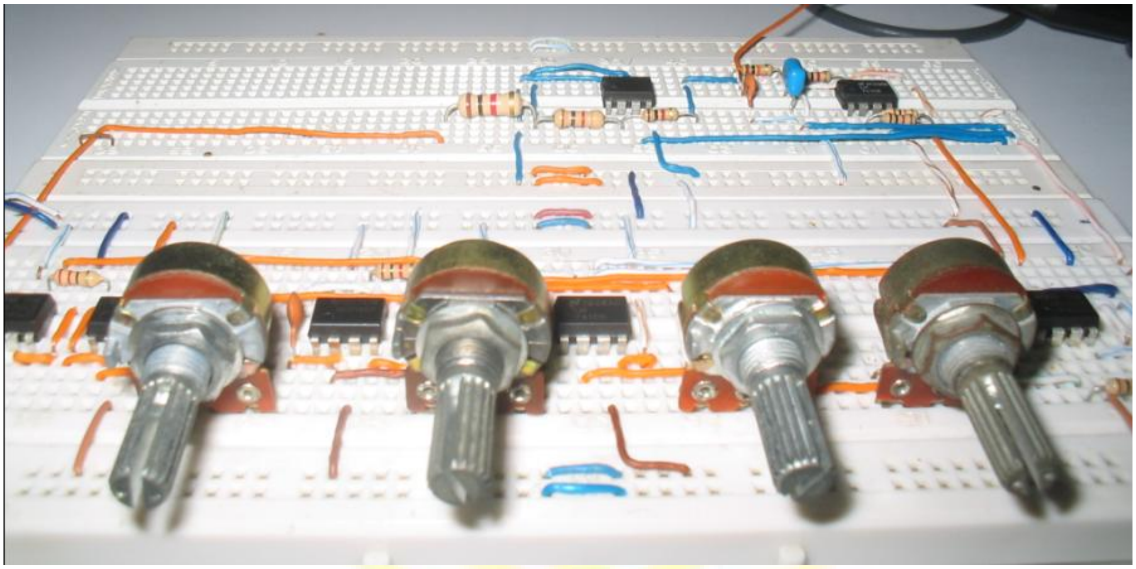
ELEMENTOS UTILIZADOS

Generador de señales

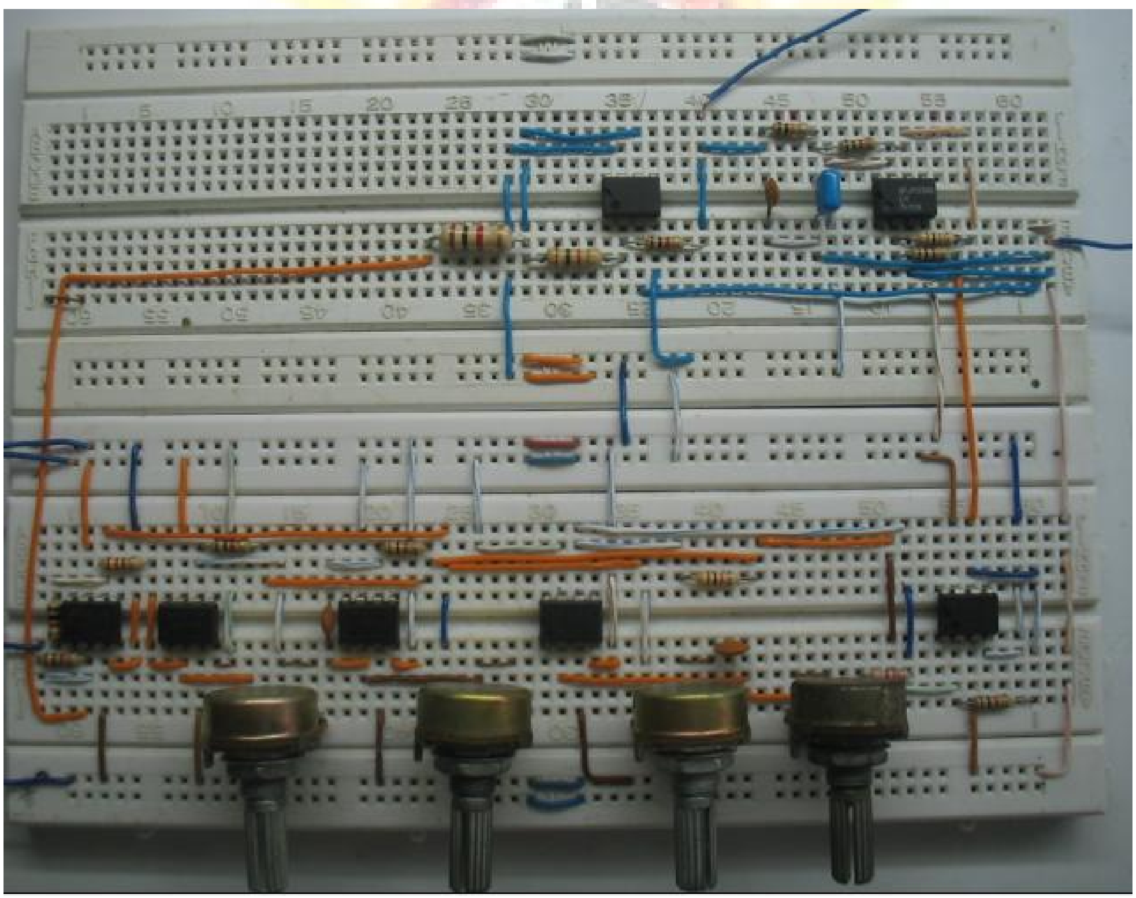
Fuente de alimentación dual

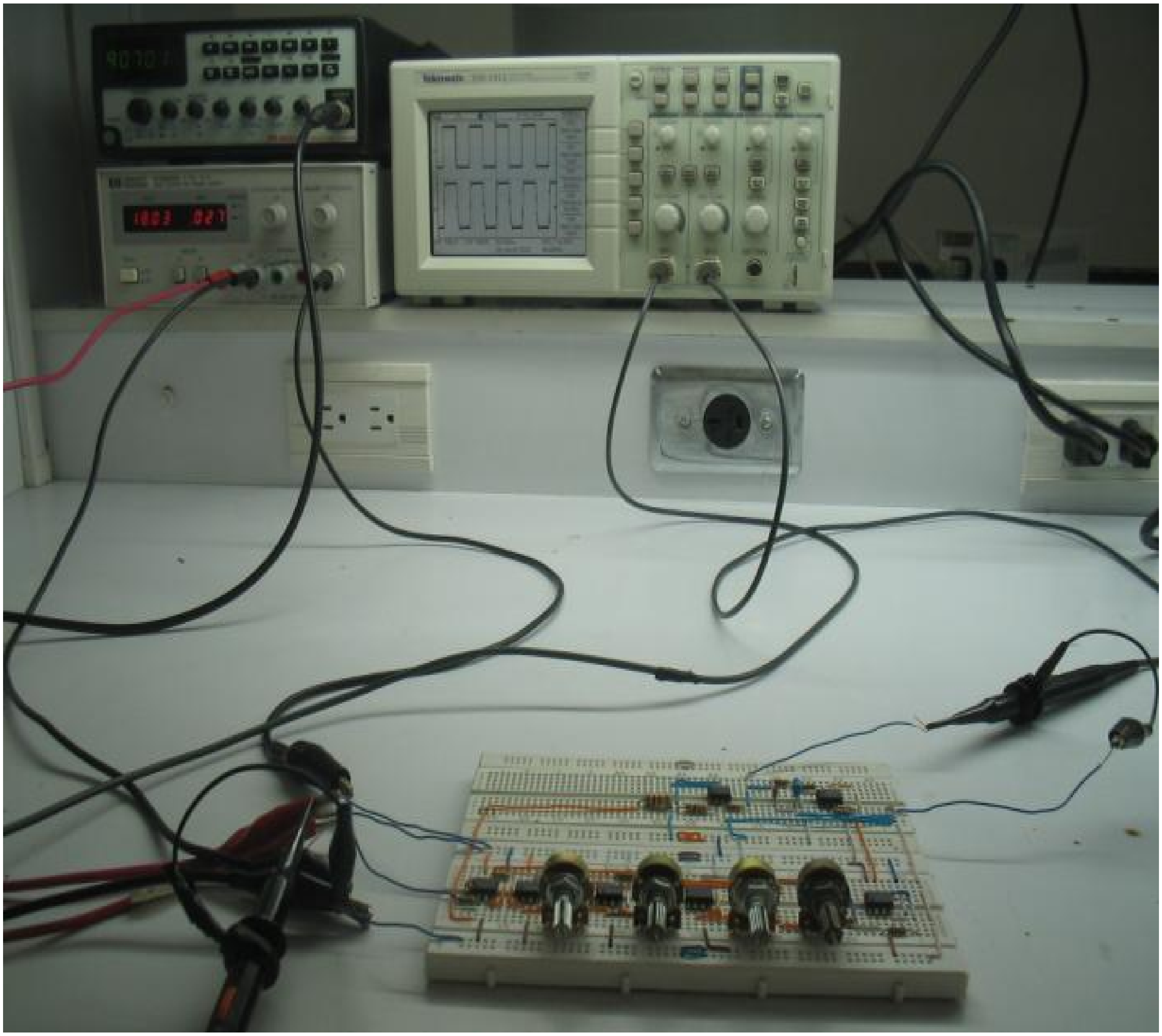
Osciloscopio

Edwin González Querubín kaliman83@hotmail.com
Morgan Garavito Vásquez mor6an1@hotmail.com



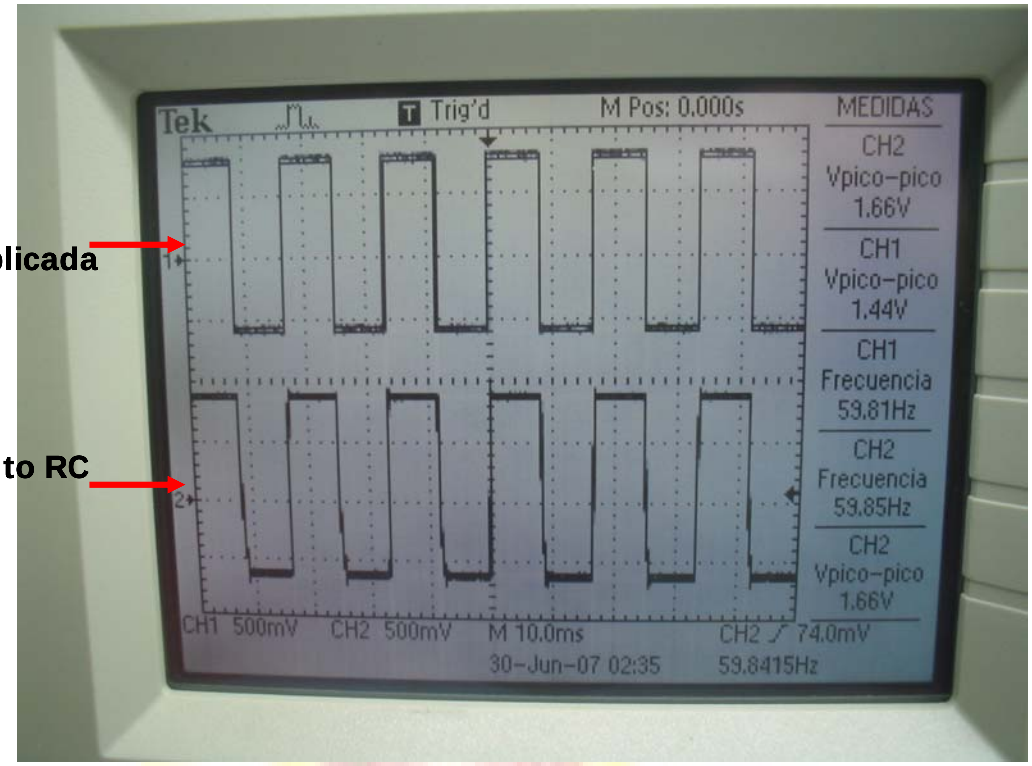
Montaje real de la figura 35





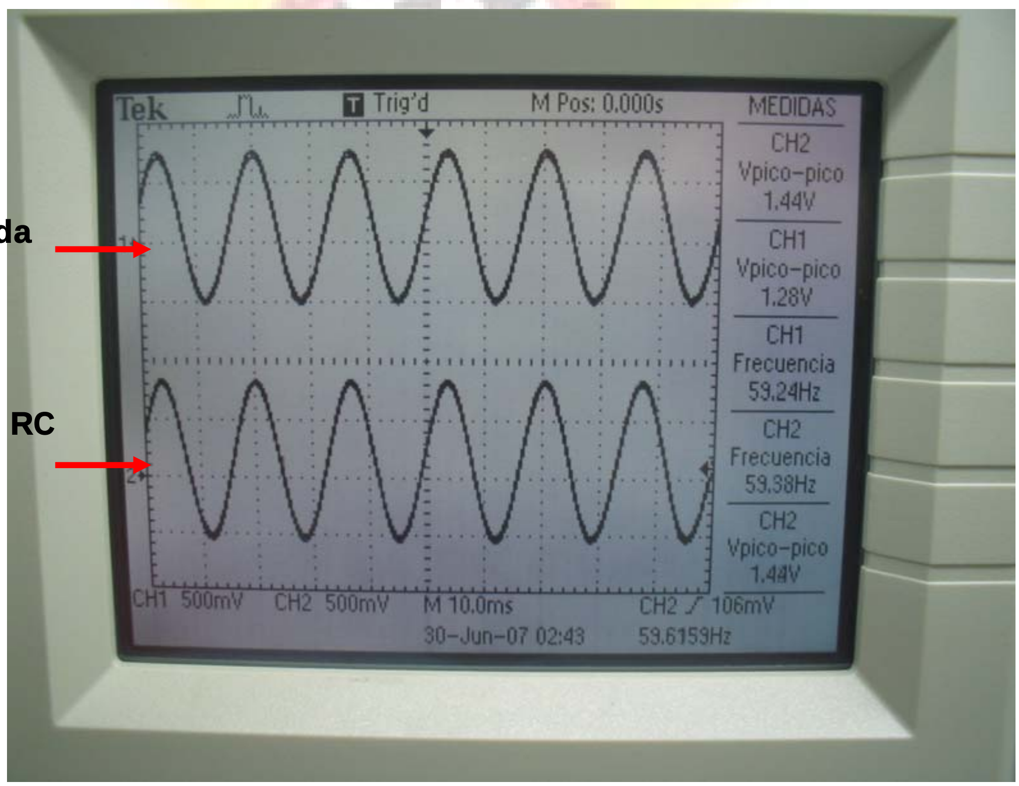
Entrada aplicada

Salida del circuito RC



Entrada aplicada

Salida del circuito RC



DUDAS O COMENTARIOS

Edwin González Querubín: → kaliman83@hotmail.com
Morgan Garavito Vásquez: → mor6an1@hotmail.com