

Control Moderno

Profesor: Fausto Andrés Escobar Revelo

10 de octubre de 2023

Nombre de laboratorio	Identificación de sistemas	
Tema	Identificación y control sistema SISO	
Sub tema	Curva de reacción	
Objetivo general	Aplicar los conceptos identificación de sistemas	
Hora	Início: 15:00	Término: 18:00
Tiempo de duración: 3 horas		

1. Objetivo

El objetivo de la presente práctica es obtener el modelo matemático de la planta ilustrado en la Figura 1 a partir de las señales $U(t)$ y $Y(t)$. Una vez obtenido este modelo se diseñará e implementará un controlador que permita que las señales de salida $Y(t)$ sigan señales de referencia tipo escalón. La consigna para cada salida deberá poderse fijar independientemente.

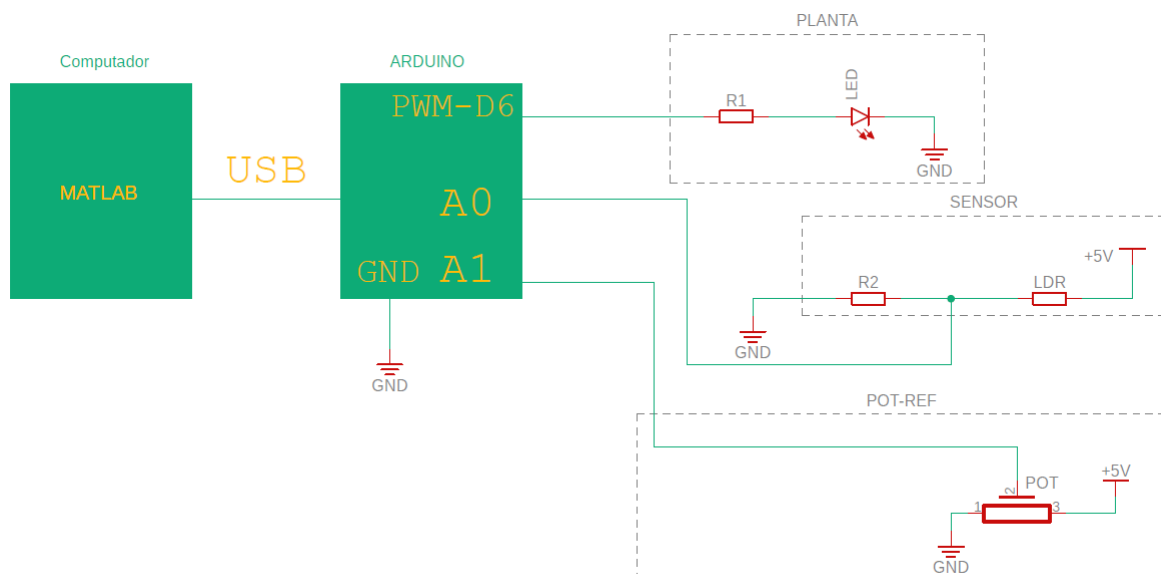


Figura 1: Diagrama de conexiones de planta de Luz.

Para realizar esta práctica se requieren los siguientes componentes (i) el circuito presentado en Figura 1. (ii) una tarjeta Arduino con conversores analógico-digital y salidas PWM (Pulse Width Modulation) (iii) un computador con Matlab 2017 o superior (iv) Archivos de Identificación que contiene las Carpetas: *MatlabFilesIdentification* y *SISO*.

- Sensor LDR o fotocelda (No comprar el módulo solo el sensor).
- Resistencias: $330\ \Omega$, $1k\Omega$ y $10k\Omega$.



- Un diodo led.
- Un potenciómetro de cualquier valor entre ($1k\Omega$ a $10k\Omega$).
- Matlab 2017^a o superior, tarjeta Arduino y Laptop.
- Caja completamente sellada de plástico donde quepa el circuito.
- Tester multímetro digital.

Repositorio: <https://github.com/xXThanatosXx/ControlModerno.git>

La presente práctica consta de seis partes: 1. Configuración del puerto de comunicación. 2. Generación de datos de entrada-salida. 3. Estimación de los parámetros del modelo del sistema de la Figura 1. 4. Validación del modelo. 5. Diseño e implementación del controlador. 6. Desarrollo del informe.

2. Configuración del puerto de comunicación

A continuación se describen los pasos para garantizar que Matlab y la tarjeta Arduino utilicen el mismo puerto serial.

- Asegúrese de que la tarjeta Arduino esté conectada a uno de los puertos USB del computador.
- Cargue el entorno Arduino y vaya al ítem Herramientas de la barra de menú principal. Una vez desplegada la lista de opciones, seleccione la opción Tarjeta y escoja el dispositivo Arduino Mega 2560 o Mega ADK.
- En el mismo menú Herramientas del paso precedente, seleccione la opción Puerto Serial y determine que puerto está utilizando la tarjeta Arduino. En la Figura 2 se muestra un ejemplo en el cual la comunicación se realiza a través del COM7.
- En el programa Matlab **ScriptIdenLDR**, modifique el argumento de la función `port = configureSerialPort()` la sentencia `com = 'COM7'`; para que coincida con el puerto seleccionado en el entorno Arduino.



Figura 2: Puerto seleccionado para realizar la comunicación entre la tarjeta Arduino y el PC.



3. Generación de datos de entrada-salida.

En la Figura 3 se presentan las señales de entrada al circuito de la Figura 1. Durante los primeros 5 segundos dichas señales toman un valor constante del 20 % ó 1 voltio. Este intervalo de tiempo es suficiente para que la planta alcance el estado estacionario. La amplitud de las extra-señales se actualiza cada 50 milisegundos.

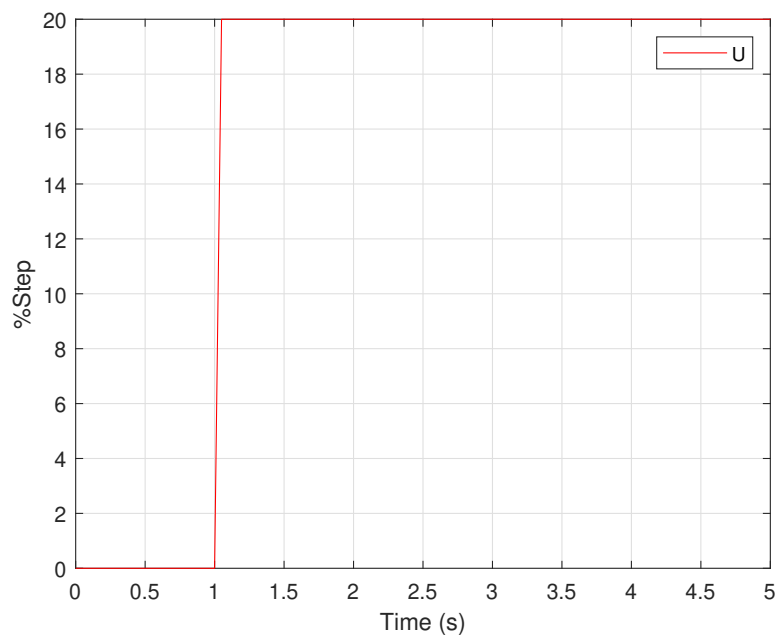


Figura 3: Señal de entrada utilizada para generar los datos de entrada-salida

- Cargue Matlab y posicione en la carpeta *MatlabFilesIdentification*
- Utilizando el comando `dir` de Matlab verifique que los programas *ScriptIdenLDR.m* y *DiscreTime.mlx* se encuentren en carpeta actual.
- Realice la conexión presentada en la Figura 1.
- En el entorno Arduino ejecute el programa *IdentificacionLDRv2.ino* que se encuentra en la carpeta System SISO.
- Posteriormente pase a Matlab y lance el programa *ScriptIdenLDR.m* para que realice la adquisición de los datos de entrada-salida. Los datos obtenidos se despliegan gráficamente y se almacenan en el archivo *DataLDR.mat*.

4. Estimación de los parámetros del modelo

En esta tercera parte de la práctica se utilizará el archivo *DataLDR.mat* generado en la sección anterior con el fin de obtener un modelo en espacio de estados del circuito eléctrico de la Figura 1. Dicho modelo se supondrá de la forma:



$$\begin{aligned}x(kh + h) &= Ax(kh) + Bu(kh) + Ke(kh) \\y(kh) &= Cx(kh) + Du(kh) + e(kh)\end{aligned}\tag{1}$$

Para obtener las matrices A, B, C, D escriba el siguiente programa en Matlab archivo *DiscreteTime.mlx*:

```
1 %% Cargar variables capturadas
2 load('DataLDR.mat')
3 % Escalon 20%
4 Input = U;
5 % Respuesta de sistema
6 Output = Y;
7 % Tiempo de muestreo
8 Ts = 0.05;
9 % Parameter estimation
10 IdData = iddata ([Y],[U ],Ts);
11 Model = n4sid (IdData,1);
12 compare(IdData,Model);
13 grid on
14 Model
15 %%Guardar planta
16 save('Model')
```

Como resultado se obtiene:

```
>> Model

Model =
Discrete-time identified state-space model:
    x(t+Ts) = A x(t) + B u(t) + K e(t)
    y(t) = C x(t) + D u(t) + e(t)

A =
      x1
x1  0.9091

B =
      u1
x1  0.002196

C =
      x1
y1  131.7

D =
      u1
y1  0

K =
      y1
x1  0.005191

Sample time: 0.05 seconds
```

Figura 4: Sistema discreto en espacio de estados.

Tenga presente que los coeficientes del modelo obtenido dependen de los datos adquiridos.



En consecuencia, el modelo que Usted obtendrá será ligeramente diferente del aquí presentado. En la Figura 5 se presenta el nivel de ajuste en los parámetros y el modelo.

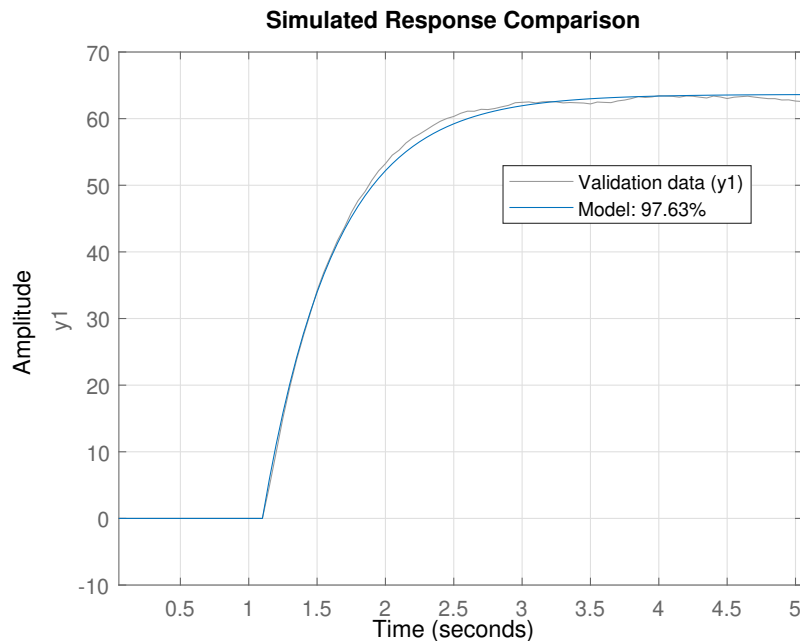


Figura 5: Ajuste entre los datos medidos y el modelo en espacio de estados obtenido.

5. Validación del modelo

En esta sección de la guía, el objetivo es comprobar la identificación del sistema objeto de estudio.

- Utilizando los comandos `tf` y `ss`, transforme el modelo obtenido en la sección precedente en una matriz de funciones de transferencia y almacénelo en la variable `G`.

$$\mathbf{G} = \text{tf}(\text{ss}(\text{Model.A}, \text{Model.B}, \text{Model.C}, \text{Model.D}, \text{Ts})) \quad (2)$$

- Cree un archivo de Simulink y compare la respuesta de la función de transferencia y espacios de estados del sistema analizado anteriormente encontrados `G` y matrices `(A,B,C,D)` 6. Los bloques Discrete State-Space y Discrete Transfer Fcn los encuentran en Library browser en la pestaña Discrete, recuerde poner el valor de `Sample Time = 0,05` en cada bloque.

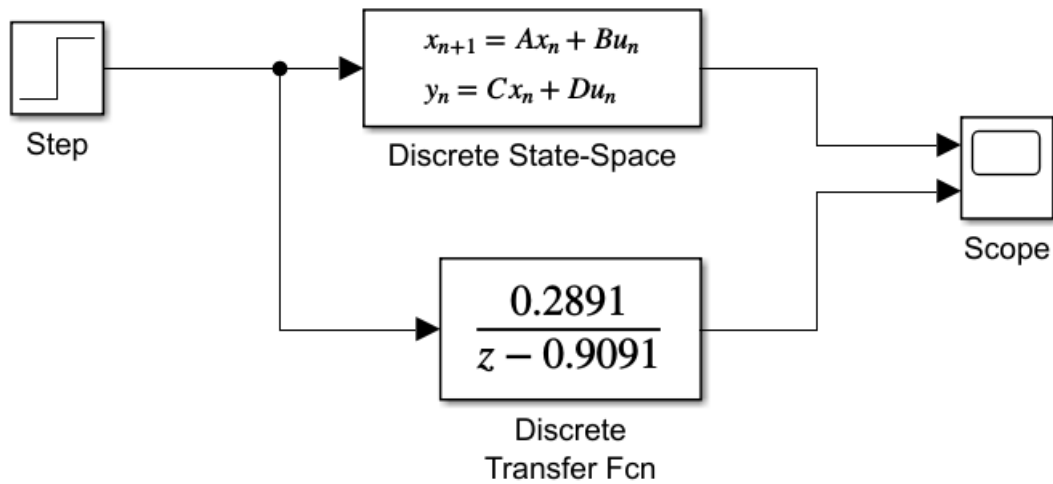


Figura 6: Esquema Simulink.

- Configure el bloque Step con los siguientes parámetros: *StepTime* : 1, *Initialvalue* = 0, *Finalvalue* = 20, *SampleTime* = 0,05.

Comparando los modelos en espacio de estados y función de transferencia discreta se obtiene:

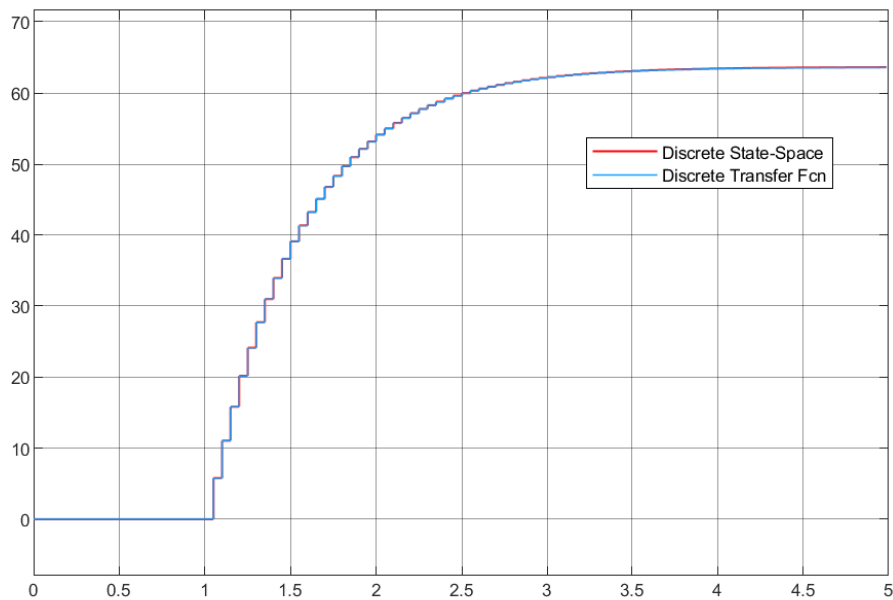


Figura 7: Respuesta de los modelos identificados a una señal escalón



6. Desarrollo del informe

- 20 % Dibuje el diagrama de bloques del sistema identificado.
- 20 % Mida el cambio de voltaje en el divisor de voltaje "**Sensor**", y describa como es la respuesta en función del cambio de luminosidad que capta el sensor.
- 20 % Identifique los estados del sistema. ¿El sistema es controlable y observable?
- 20 % Calcule el periodo de muestreo según Shannon del sistema.
- 20 % Cambie el periodo de muestreo T_s a 1 segundo y 10 ms, y realice la guía nuevamente. ¿Qué cambios se produjeron al modificar T_s ?