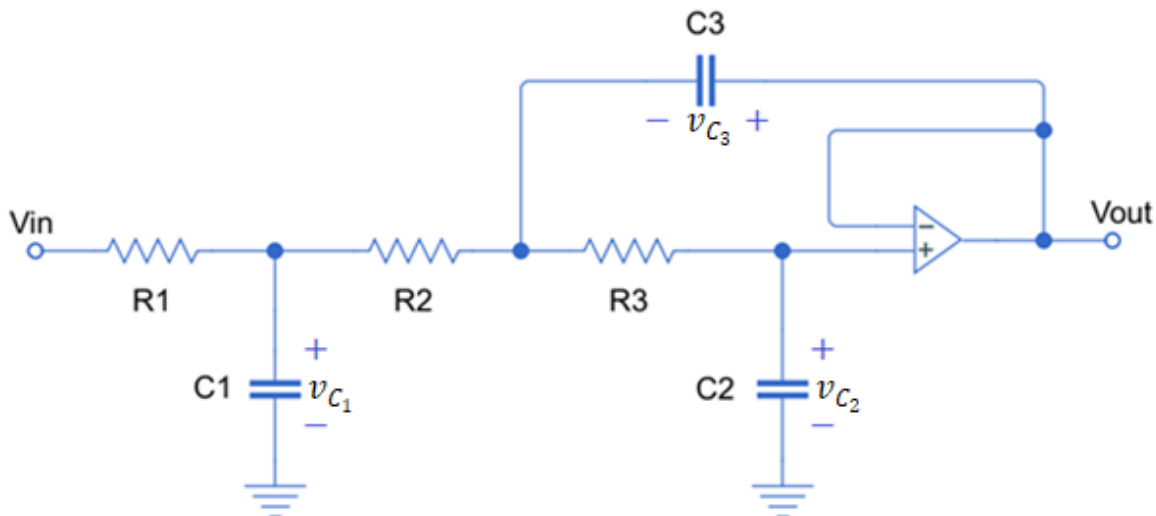


# Laboratorio 6: Control Mediante Retroalimentación de Estado

## Objetivos

- Diseñar, simular e implementar un control lineal de un circuito.
- Crear un programa para la Tiva C del controlador lineal.
- Comparar el enfoque usando variables de estado y el enfoque clásico usado en laboratorios anteriores.

**Duración:** 2 ½ sesiones



**Figura 1.** Circuito con el que se trabajará.

En este laboratorio diseñarán, simularán e implementarán un control lineal del circuito. Usarán la técnica de “*Pole Placement*” para encontrar matrices (vectores) **K** correspondientes a la ley de control lineal  $\mathbf{u} = -\mathbf{K}\mathbf{x}$ .

Como antes, usarán:  $R_1 = 1 \text{ k}\Omega$ ,  $R_2 = R_3 = 10 \text{ k}\Omega$ ,  $C_1 = 1 \text{ }\mu\text{F}$ ,  $C_2 = 0.1 \text{ }\mu\text{F}$ ,  $C_3 = 10 \text{ }\mu\text{F}$ .

Usarán como señal de entrada (referencia) una **señal cuadrada de entre 1 y 2 V, a una frecuencia de 0.667 Hz**. Para la simulación, asegúrense de que la señal **empiece en 1, no en 2**.

Se asume que las variables de estado a utilizar son los voltajes en los capacitores, en el orden y la polaridad en que se observaron en el laboratorio 5.

Apóyense en las lecturas, notas de clase y ejemplos de la semana 7 de Sistemas de Control 2, y en la *Lecture 6* de Sistemas de Control 1.

## **Primera Parte: Diseño y simulación del controlador lineal**

En esta parte usarán Matlab para encontrar matrices **K** que logren el control lineal adecuado. También graficarán la respuesta controlada usando comandos de Matlab. Además, simularán el sistema controlado en Simulink, tanto con bloques como con componentes de Simscape. Para los pasos de Matlab a continuación, se sugiere crear un *script*.

1. Consideren las matrices **A**, **B** y **C** del modelo de espacio de estados que correspondan a los voltajes de los capacitores como las variables de estado. Usen dichas matrices para generar una variable tipo **ss**. **Nota:** ya deben tener estas matrices y variable tipo **ss** (conjunto de matrices A3, B3 y C3 de los laboratorios 3 y 5).
2. Verifiquen la controlabilidad del sistema mediante el rango de la matriz de controlabilidad.
3. La salida del sistema ya controlado deberá ser similar a la requerida en el laboratorio 1. Las especificaciones de diseño son:

$$e_{ss} = 0, \quad 5 \text{ ms} < t_r < 10 \text{ ms}, \quad t_s < 100 \text{ ms}, \quad 10\% < M_p < 20\%.$$

**Propongan polos que crean adecuados para cumplir con las especificaciones. Ayuda:** pueden tomar como referencia los polos del sistema en lazo abierto, y “mover” los polos a posiciones que mejoren las características del sistema. **Atiendan las indicaciones del instructor. Importante:** Comprobarán si los polos propuestos son adecuados en el inciso 9. Si no lo son, repetirán este y los siguientes incisos. **Deberán reportar únicamente los polos finales que hagan cumplir las especificaciones.**

4. A partir de los polos del inciso anterior, encuentren la matriz **K** de la ley de control lineal. **Deberán reportar la matriz K final. Ayuda:** usen la función **place** de Matlab.
5. Encuentren la matriz **A<sub>cl</sub> = A – BK**. Usen dicha matriz (junto con las matrices **B** y **C** originales) para encontrar una nueva variable tipo **ss**, correspondiente al sistema “mejorado” (lazo cerrado).
6. Generen una señal cuadrada **con las características indicadas en la página 2** (usen las funciones utilizadas en el laboratorio 5). La señal debe tener dos períodos. Ésta será su señal de referencia.
7. **Obtengan la señal de salida tanto para el sistema original (lazo abierto) como para el sistema en lazo cerrado.** Utilicen la función **lsim** y las variables tipo **ss** encontradas anteriormente, así como la señal cuadrada generada en el inciso anterior. **Grafiquen las señales de salida junto con la señal de entrada. ¿Notan algo extraño con la escala de la señal de salida del sistema controlado?**
8. Vayan al sitio:  
<http://ctms.engin.umich.edu/CTMS/index.php?example=Introduction&section=ControlStateSpace>  
 y lean la sección “*Introducing the Reference Input*”. Allí se explica la necesidad de escalar la señal de referencia para que la salida coincida con dicha referencia en estado estacionario (logrando que  $e_{ss} = 0$ ). También se provee un enlace para descargar el archivo **rscale.m**, el cual

puede usarse para encontrar el factor de escala adecuado para su sistema. **Nota:** `rscale` NO es una función nativa de Matlab, por lo que deben descargar el archivo `rscale.m` y salvarlo en el directorio donde estén trabajando.

9. Encuentren el factor de escala (“Nbar”), aplíqueno a la señal de referencia, y **vuelvan a obtener la salida del sistema controlado. Grafiquen la señal de salida junto con la referencia (la original, no escalada). ¿Obtuvieron una respuesta adecuada?** Usen la función `linearSystemAnalyzer` (aplicado al sistema en lazo cerrado) para verificar que se cumplan todas las especificaciones del inciso 3. **Si no se cumplen, deberán encontrar nuevos polos y repetir los incisos anteriores.**

**Importante:** Únicamente deberán reportar los polos y las gráficas  finales, es decir, aquellos resultados que cumplan con las especificaciones de diseño.

10. En Simulink, usen un bloque de modelo de espacio de estado con la matriz  $\mathbf{A}_d$  que encontraron antes (y las matrices  $\mathbf{B}$  y  $\mathbf{C}$ ). Usen como referencia la señal cuadrada descrita anteriormente, y **observen la salida junto con la referencia en un bloque Scope**. No olviden que al bloque de espacio de estados debe entrar la referencia escalada. Recuerden que en los bloques pueden usar variables que estén en el *Workspace*.
11. En Simulink, usen ahora el circuito (componentes de Simscape) e implementen el control usando las variables de estado directamente (y retroalimentación). Necesitarán bloques de amplificación, suma/resta, etc. Usen como referencia la señal cuadrada descrita anteriormente, y **observen la salida junto con la referencia en un bloque Scope**. Noten que la señal a la que se le debe restar  $\mathbf{Kx}$  es la referencia escalada:  $u = \text{Nbar} \cdot \text{ref} - \mathbf{Kx}$ .

**Nota:** en este sistema sí es posible medir y usar las variables de estado directamente. Cuando eso no es posible, se pueden usar observadores (eso será el énfasis del Laboratorio 7).

12. **Repitan el paso 3**, pero buscando satisfacer las siguientes especificaciones de diseño:

$$e_{ss} = 0, \quad M_p = 0\%, \quad t_s < 25 \text{ ms.}$$

**Sugerencia:** consideren polos estrictamente reales.

13. **Repitan los pasos 4 – 7, 9 – 11**, usando los polos del inciso anterior.

**En el reporte deberán incluir únicamente los resultados finales de los incisos anteriores. Es decir, sólo los polos, la matriz  $\mathbf{K}$  y las gráficas de cuando se hayan cumplido las especificaciones de diseño.**

## **Segunda Parte: Implementación de los Controladores en la Tiva C**

En esta parte **crearán un programa para la Tiva C que implemente los controles lineales para el circuito físico**. Noten que, esencialmente, el programa debe permitir la implementación física de lo hecho en el inciso 11 de la Primera Parte. La Tiva C tiene suficientes entradas analógicas, y los cálculos y ampliaciones pueden hacerse por *software*.

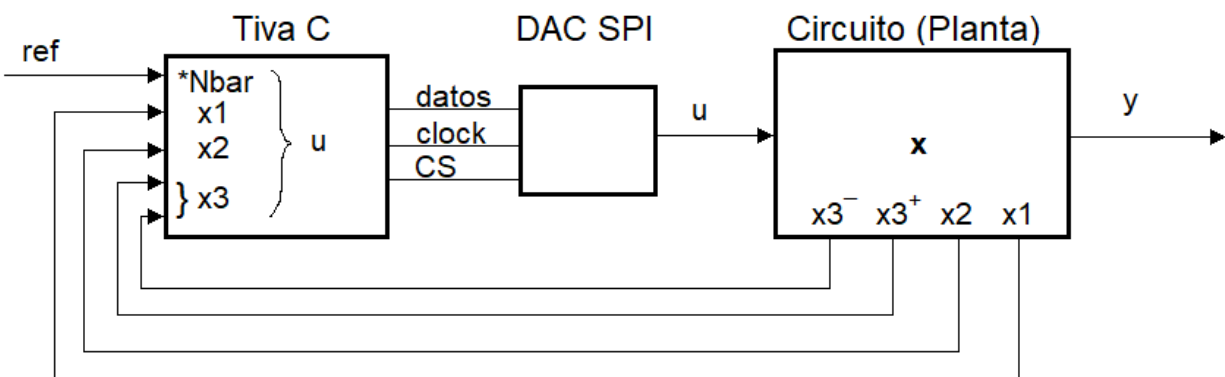
El programa debe tener código/variables correspondientes a las dos matrices  $\mathbf{K}$  que encontraron en la Segunda Parte (pueden codificar más casos, si lo desean). Pueden comentar/descomentar lo necesario para dejar la ley de control deseada. También pueden usar directivas del pre-procesador para seleccionar el caso (`#if`, etc.), o cambiar de caso en el modo “Debug” de *Code Composer Studio*. Noten que el único cambio entre los casos serán los valores de los elementos de  $\mathbf{K}$ . El algoritmo de control es el mismo.

**Nota 1:** Recuerden que el voltaje en el capacitor C3 es la diferencia entre sus terminales. La diferencia la pueden calcular por *software*.

**Nota 2:** a diferencia de lo hecho en los laboratorios 1 y 2, ahora no hemos considerado períodos de muestreo ni discretizaciones para el diseño de los controladores. ¿Qué período de muestreo creen que sería adecuado usar en sus programas?

**Nota 3:** al igual que en los programas del laboratorio 2, deberán tomar en cuenta cotas para los valores calculados de control, mapeos y conversiones de variables, y la comunicación con el DAC externo.

1. Consideren la imagen mostrada en la Figura 2. Hagan las conexiones correspondientes en su protoboard.



**Figura 2.** Conexión en lazo cerrado, retroalimentación de estado.

2. **Para cada caso de  $\mathbf{K}$** , conecten al sistema como referencia la señal cuadrada descrita anteriormente (entre 1 y 2 V, 0.667 Hz). **Observen en el osciloscopio tanto la referencia como la salida controlada.** Al igual que en la Primera Parte, ajusten la escala temporal para mostrar dos períodos de la señal cuadrada.

## **Evaluación:**

A más tardar al inicio de la primera sesión del laboratorio 7, cada grupo deberá **demostrar el funcionamiento de sus controladores (Segunda Parte)**. Pueden hacer la demostración durante las sesiones de atención a estudiantes.

**Deberán subir a Canvas un reporte (.pdf) y su programa de la Tiva C (.c), a más tardar el domingo 2 de abril.** Una entrega por grupo.

El reporte deberá incluir:

- 1) **Identificación:** sus nombres, carnés, nombre del curso, sección de laboratorio (11, 12 o 21), número y título del laboratorio, fecha.
- 2) **Resumen y discusión.** Incluyan los objetivos y lo más destacado de sus resultados obtenidos. Discutan sobre las similitudes y diferencias entre lo hecho en los laboratorios 1 y 2 (control clásico) y el enfoque por retroalimentación de estados de este laboratorio (máximo una página).
- 3) Una sección de **Resultados**. No olviden contestar las preguntas planteadas en la guía. Como siempre, **se verificará que esté todo lo indicado en color azul**. Asegúrense de numerar y titular todas las imágenes. **Recuerden organizar los resultados según las Partes y los incisos de la guía, como se explicó en la guía del laboratorio 1.**

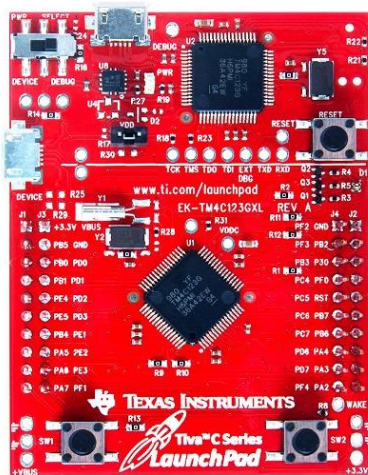
Suban únicamente la versión final de su programa. No es necesario que suban el archivo “*startup*”. Comenten su código e incluyan sus nombres, carnés, nombres del curso y número de laboratorio en la parte superior.

Asistencia y trabajo en el lab:	20%
Demostración:	20%
Reporte (.pdf):	50%
Código (.c):	10%

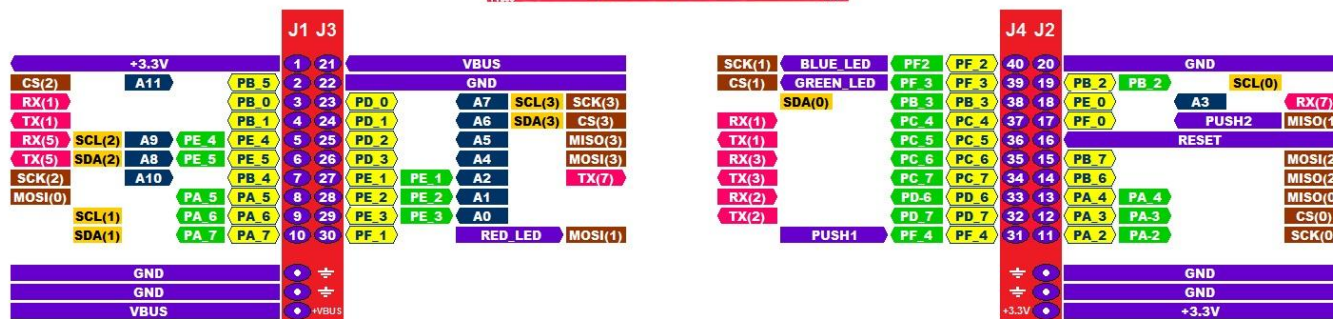
## Anexo



## Pinout Diagram Ver 1.0



Hardware
digitalRead() and digitalWrite() PORTS
analogRead()
analogWrite()
I <sup>2</sup> C (TWI)
SPI
Hardware Serial



**Arduining.com**

**Figura A1.** *Pinout* del Tiva C TMC123G