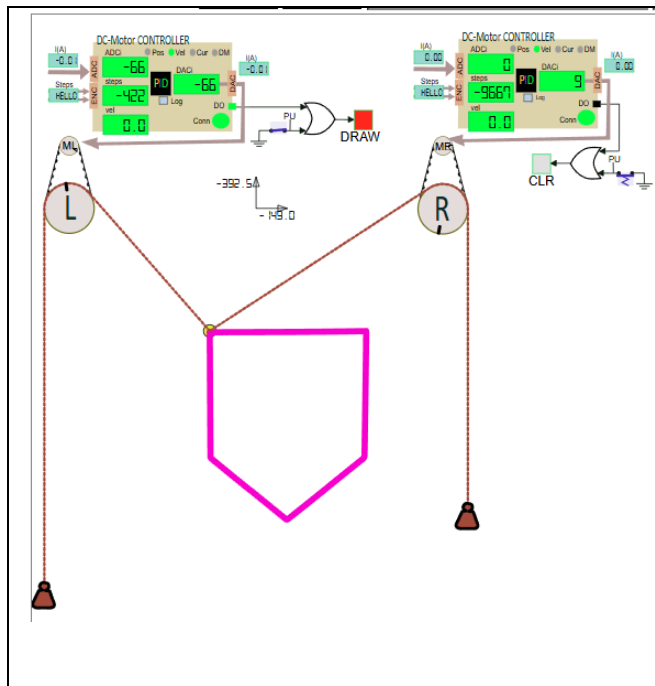


Trabajo Implementación Sistemas de Control (enero 2025)

Se dispone de un plotter vertical simulado, con un funcionamiento similar al descrito en <https://www.hackster.io/mertarduino/make-arduino-xy-plotter-drawing-robot-polargraph-b4fe36>.

El sistema está formado por:



- Dos motores ML y MR, con encóders de posición de 64x4 pulsos por vuelta, conectados mediante transmisión 1:80 a sendas poleas de radio 80mm que están separadas 1200mm en el eje horizontal.
- Una pluma trazadora con masa=0.5Kg, unida mediante cadenas que pasan a través de las poleas a sendos contrapesos de 0.5 Kg cada uno.
- Una controladora para cada motor, con comunicación TCP/IP, que permite dos modos de funcionamiento:
 - o Cadena abierta: la controladora inyecta la corriente deseada al motor para realizar control de par.
 - o Cadena cerrada: se indica a la controladora la velocidad deseada y los parámetros de control PID (K_p , K_d , K_i), y la controladora realiza el control de velocidad.

Para este sistema, se desea realizar el control de posición simultáneo de ambos motores, de manera que realicen el trazado de una forma geométrica (ejemplo un rectángulo). Para valorar la calidad del control se tendrán en cuenta tanto la precisión del trazado realizado como la rapidez en ejecutarlo.

Se dispondrá para la realización del trabajo de los siguientes elementos:

- El simulador en sí mismo, un programa que se ejecuta en Microsoft Windows 64bit y dispone de un interfaz web para ver la evolución del sistema. El simulador incluye el manual de uso, los datos de cada uno de los componentes, el modelo cinemático y dinámico utilizado, etc.
- Matlab/Simulink, versión 2023. Se pueden descargar e instalar para uso académico mediante el email de la Universidad de Oviedo. Las opciones a instalar, así como los archivos con modelos y scripts que facilitan la comunicación con el simulador, se encuentran también en el manual del simulador.
- Qt-Creator con compiladores mingw64.

Se realizará una primera parte del trabajo desde el entorno Matlab/Simulink, con el fin de definir el diseño y sintonización del control que proporcionen un trazado con un nivel de precisión aceptable.

La segunda parte del trabajo consiste en implementar el control desarrollado en entorno C/C++, de manera que se pueda disponer de una aplicación independiente que pueda controlar el sistema e interactuar con él.

La calificación de ambas partes se realizará por separado:

- Sintonización del control: 3.6 puntos en total (profesor encargado: Fernando Briz)
- Implementación C/C++: 6.4 puntos en total (profesor encargado: Ignacio Alvarez)

Sintonización del control

Tareas

- Sintonizar los reguladores en cascada de corriente, velocidad, posición para que el control de cada motor responda de la manera adecuada, según el modelo del motor que se encuentra en la documentación del simulador, y el par de carga calculado para la posición XY.

- Los ensayos se realizarán y documentarán al menos en las posiciones siguientes:

P1: X: 0 Y: -100

P2: X: 450 Y: -1200

Entregables (enviar a fbriz@uniovi.es)

Comprimido en un archivo (Matlab.zip):

1. Script(s) de Matlab y modelos de Simulink con el diseño de control.
2. Documento pdf explicando el diseño, los cálculos realizados y los resultados obtenidos en los puntos de control.

Criterios de calificación (3.6 puntos)

- El modelo de simulación funciona y está documentado de forma adecuada:
 - Con carencias evidentes en la sintonización de los reguladores, saturaciones, Hasta **1.8 puntos**
 - La respuesta dinámica, saturaciones... , son correctas: Hasta **2.7 puntos**
- Se modela adicionalmente el comportamiento del par de carga en función de la posición XY. En este caso, es necesario simular los dos ejes de forma simultánea: Hasta **0.9 puntos**

(La calificación mínima en esta tarea para superar la asignatura es de 1.8 puntos)

(La coincidencia en planteamientos con los trabajos de otros compañeros restará calificación, hasta el punto de poder invalidar el trabajo).

Implementación del control

Tareas y criterios de calificación (6.4 puntos)

- Realizar un programa básico en modo Qt-Gui para comunicarse con los motores vía TCP/IP, que implemente el control desarrollado en el apartado anterior para una trayectoria rectilínea (PL6), con las características siguientes:
 - Ejecutar el control que se ha diseñado en el apartado anterior bajo los eventos de un QTimer con paso 100ms.
 - Guardar los resultados del control en un archivo de texto (**formato del archivo en anexos**)
- Ampliar el programa de PL6 para realizar la trayectoria deseada, definida por un archivo XML según lo especificado en anexos.

Entregables (enviar a ialvarez@uniovi.es)

Comprimido en un archivo .zip:

Archivos .cpp , .h y .pro de la versión más avanzada que se haya desarrollado.

Archivos de resultados .txt para los rectángulos R1 y R2

Criterios de calificación (6.4 puntos)

- Programa básico modo Qt-GUI para una línea recta, con opciones para diferentes tipos de ensayos (contenido de PL6, con el formato de archivo de salida indicado en anexos): hasta **4 puntos**.
- Añadir opciones para trazado de una trayectoria cualquiera a partir de archivo XML: hasta **1.2 puntos**.
- Añadir la opción para la generación de una trayectoria circular, a partir de nuevos controles en que el usuario pueda establecer el centro y el radio de la circunferencia a dibujar: hasta **1.2 puntos**.

(La coincidencia del código con los trabajos de otros compañeros restará calificación, hasta el punto de poder invalidar el trabajo).

(La calificación mínima en esta tarea para superar la asignatura es de 3.2 puntos)

Fechas clave

EV1: PL6 de C++ y modelo básico Matlab/Simulink: 08/01/2025 , 23:59:59

EV2^(*): Examen de evaluación (sólo alumnos que no entreguen EV1 a tiempo): 10/01/2025, 09:00, Aula MIM2

EV3: Trabajo final según especificaciones: 31/01/2025, 23:59:59

EV4^(*): Examen y trabajo final (alumnos que no superen EV1 a EV3): 28/05/2025, 16:00, Aula MIM2

EV5^(*): Examen y trabajo final (alumnos que no superen EV1 a EV3): 02/07/2025, 16:00, Aula MIM2

(*) Según la normativa de la Universidad de Oviedo, los alumnos sólo pueden presentarse a 2 de las convocatorias en el mismo curso académico.

Anexos

Formato del archivo txt de salida

Para cada punto de la trayectoria (cada evento del timer), se añade una línea con el contenido siguiente (subíndice t indica target, subíndice c indica current) :

	Target (según cálculos del programa)									Current (obtenido de las controladoras)									
Col→	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
	t	X _t	Y _t	Vx _t	Vy _t	L1 _t	L2 _t	V1 _t	V2 _t	X _c	Y _c	Vx _c	Vy _c	L1 _c	L2 _c	V1 _c	V2 _c	I1 _c	I2 _c

Tiempo t en segundos, todas las posiciones/longitudes en mm, velocidades en mm/s, y corrientes en A.

Formato del archivo xml de entrada (datos de los puntos y del control)

<pre> <Line> <Index>1</Index> <Pen>true/false</Pen> <Clear>true/false</Clear> <Vmax>valor Vmax en mm/s</Vmax> <Amax>valor Amax en mm/s²</Vmax> <Dmax>valor Dmax en mm/s²</Vmax> <To> <x_mm>valor de x en mm</x_mm> <y_mm>valor de y en mm</y_mm> </To> </Line> <Line> <Index>2</Index> <Pen>true/false</Pen> <Clear>true/false</Clear> <To> <x_mm>valor de x en mm</x_mm> <y_mm>valor de y en mm</y_mm> </To> </Line> ... </pre>	<p>(Si no hay campo Pen, se deja como estaba) (Si no hay campo Clear, se deja como estaba) (Si no hay campo Vmax, se deja como estaba) (Si no hay campo Amax, se deja como estaba) (Si no hay campo Dmax, se deja como estaba)</p>
---	--