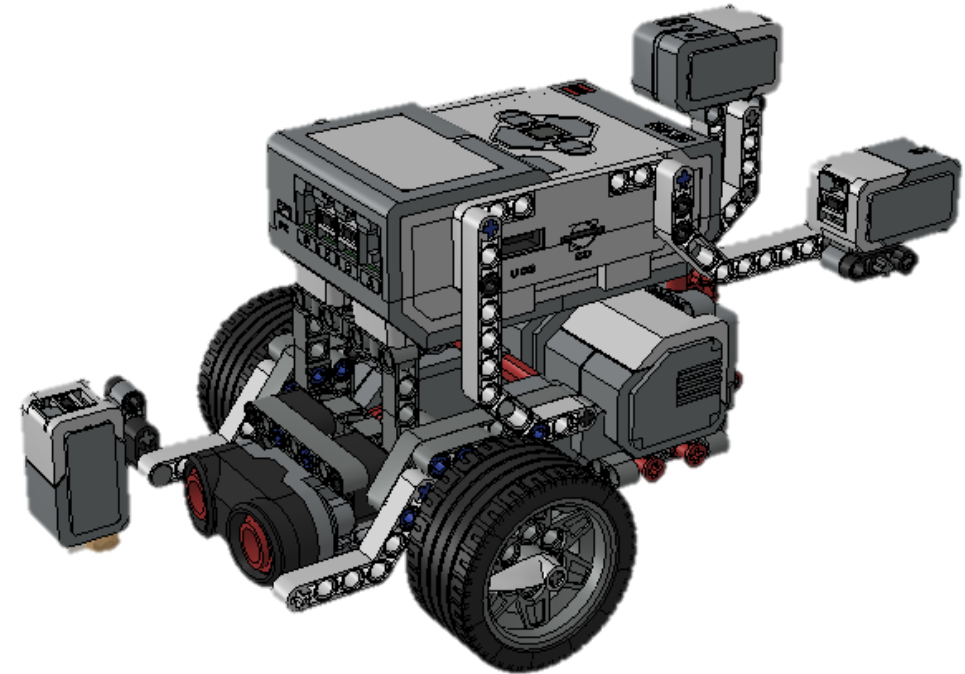


Práctica 6. Simulación de seguimiento de trayectoria con un robot diferencial en Simulink e implementación en Matlab/CoppeliaSim

Curso 2020/2021
Grado en Ingeniería Informática



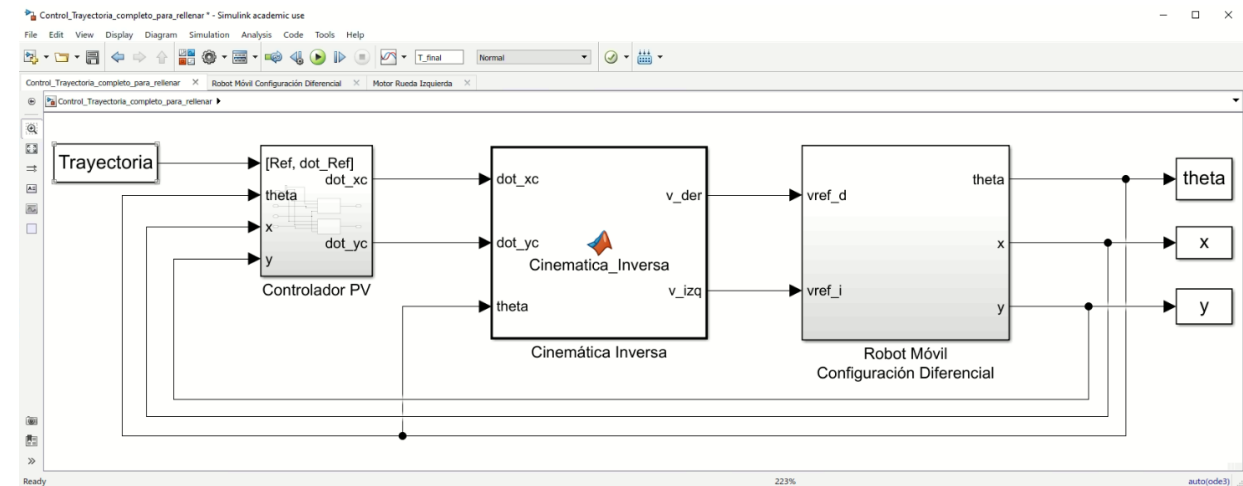
Aprender a simular el seguimiento de trayectorias de un robot móvil en configuración diferencial con Simulink.

Para ello cual utilizaremos y integraremos:

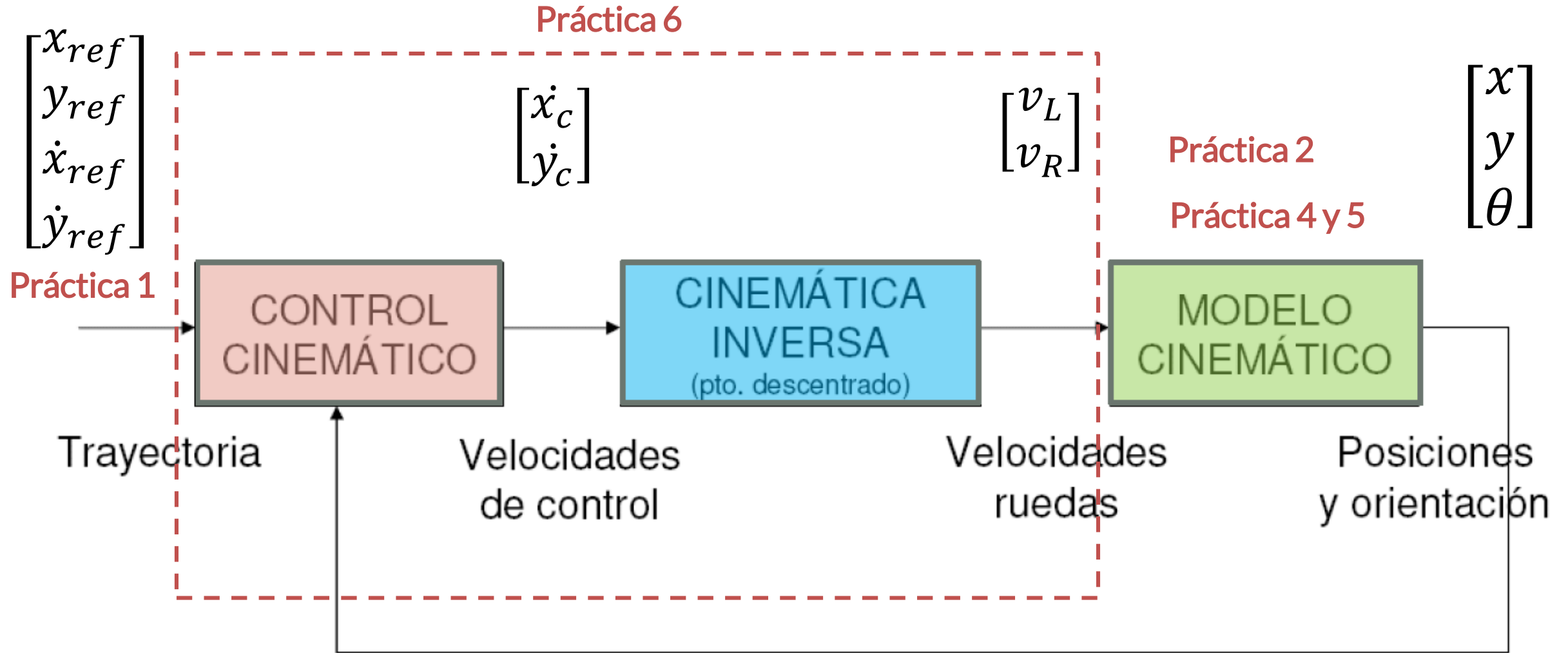
- Las funciones de generación de trayectorias creadas en la Práctica 1
- El modelado cinemático y la simulación de un robot móvil con configuración diferencial de la Práctica 2
- La identificación y control del motor de las ruedas del Lego EV3 realizados en las Prácticas 4 y 5

Implementar el algoritmo de seguimiento de trayectoria en Matlab/CoppeliaSim

Simulación de seguimiento de trayectoria con un robot diferencial en Simulink



Seguimiento de trayectorias



Ejercicio 1: Integración del control PID de velocidad angular del motor del Lego EV3 (Práctica 5) en el modelo cinemático de un robot con configuración diferencial (creado en la Practica 2).

Ejercicio 2: Implementación de la cinemática inversa de punto descentralizado (\dot{x}_c, \dot{y}_c) como bloque generador de las referencias de velocidad de las ruedas del robot (v_L, v_R) .

Ejercicio 3: Implementación del control de seguimiento de trayectoria tipo proporcional con velocidad (PV) para generar las derivadas de la posición del punto descentralizado (\dot{x}_c, \dot{y}_c) a partir de la configuración actual del robot (x, y, θ) y de la trayectoria de la referencia (x_{ref}, y_{ref}) y sus derivadas $(\dot{x}_{ref}, \dot{y}_{ref})$.

Ejercicio 4: Generación de la trayectoria de la referencia (x_{ref}, y_{ref}) y sus derivadas $(\dot{x}_{ref}, \dot{y}_{ref})$ utilizando las funciones de la practica 1, cambiando el tiempo real.

Ejercicio 5: Seguimiento de trayectoria de los puntos propuestos:

Realizar un script que defina los puntos de partida y llame a la función correspondiente para obtener las curvas de aproximación. Luego llame a la ejecución del simulink.

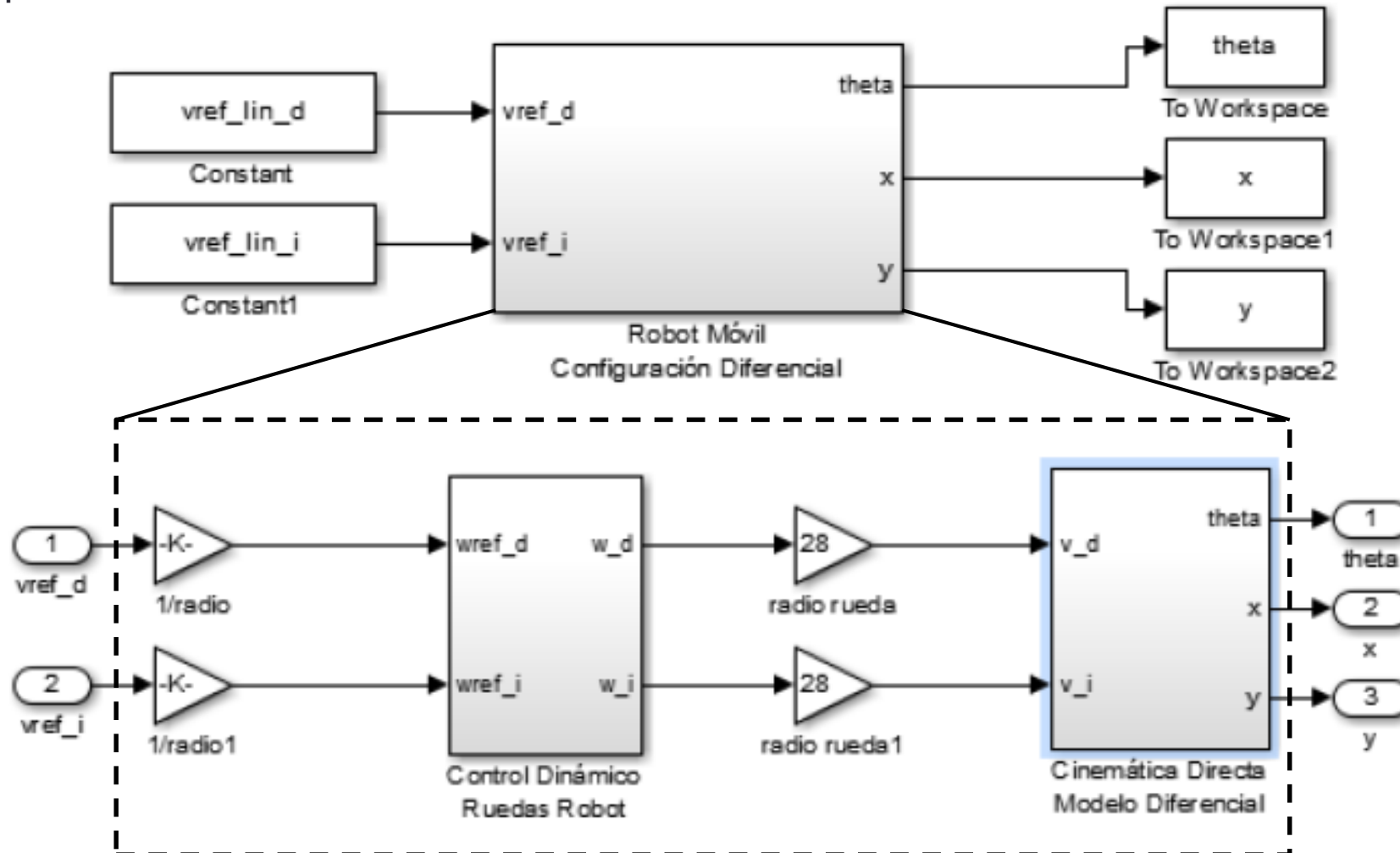
Calcula el Error (como función del tiempo) y el ITAE de la trayectoria obtenida. Incluye la representación gráfica de los puntos de partida y las curvas generadas.

Probarlo con los siguientes conjuntos de puntos:

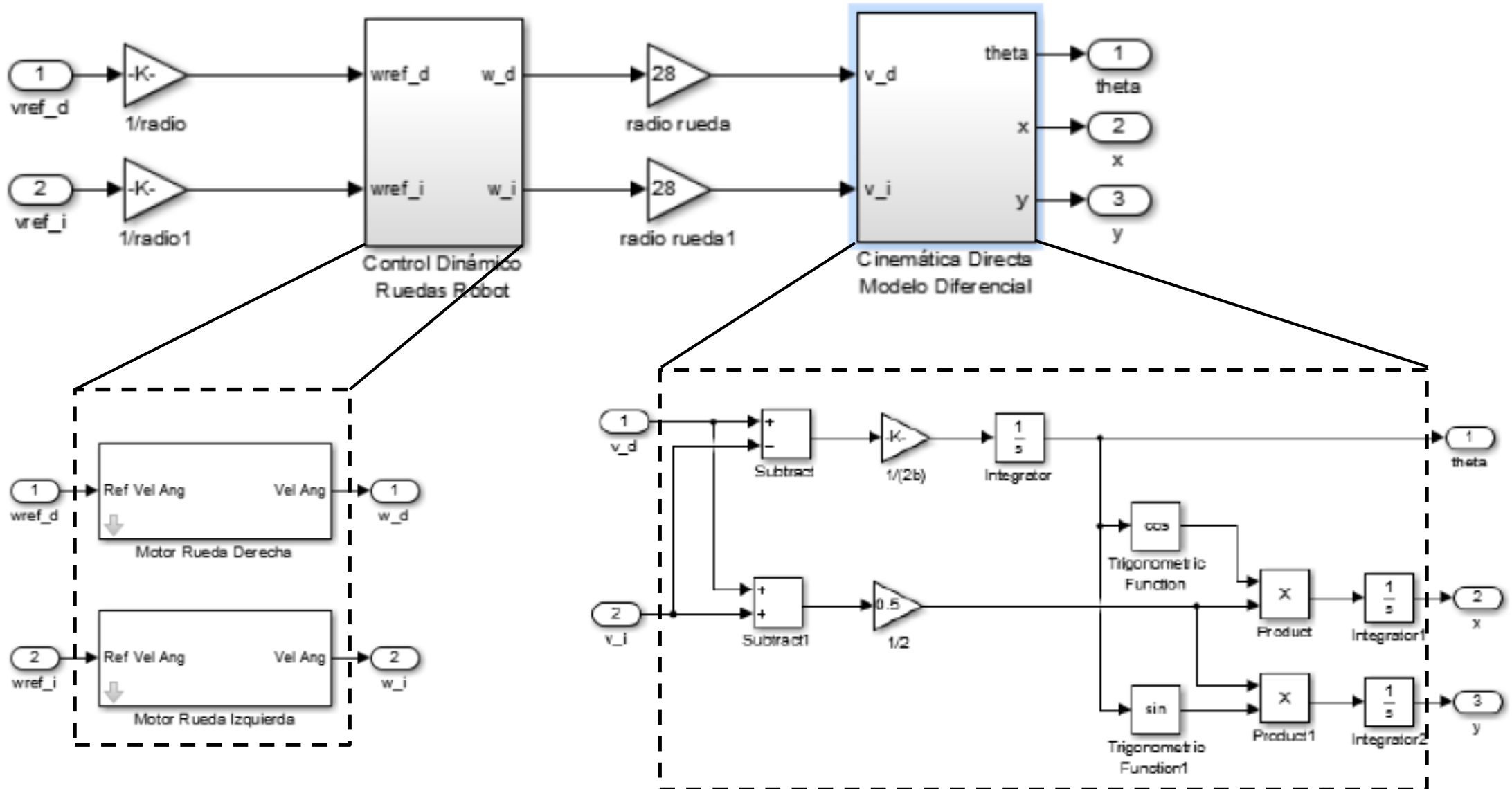
- a) `PtosControlA [0 0; 1000 500;0 1000]; %mm`
`Total_t = 4; %Segundos`
- b) `PtosControlB=[200 -200; 800 1000; 1500 500; 0 600]; %mm`
`Total_t = 20; %Segundos`
- c) `PtosControlC= Limitado por tu imaginación!!!`

Trabajo a realizar

Comenzamos con la practica 2



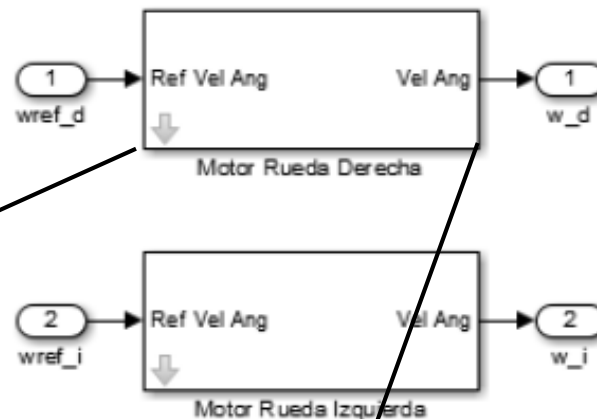
Trabajo a realizar



Trabajo a realizar



departamento de ingeniería
de sistemas y automática



Block Parameters: Motor Rueda Derecha

Controlador y modelo de motor rueda Lego EV3 (mask)

Este bloque incluye un bloque de funcion de controlador tipo PID discreto que hay que implementar (en ecuaciones en diferencias). Ademas incluye el modelo del motor de una rueda de Lego EV3, al que se le pueden cambiar los valores para personalizarlo.

Parámetros

Valor ganancia motor
0.2

Valor tiempo motor
0.06

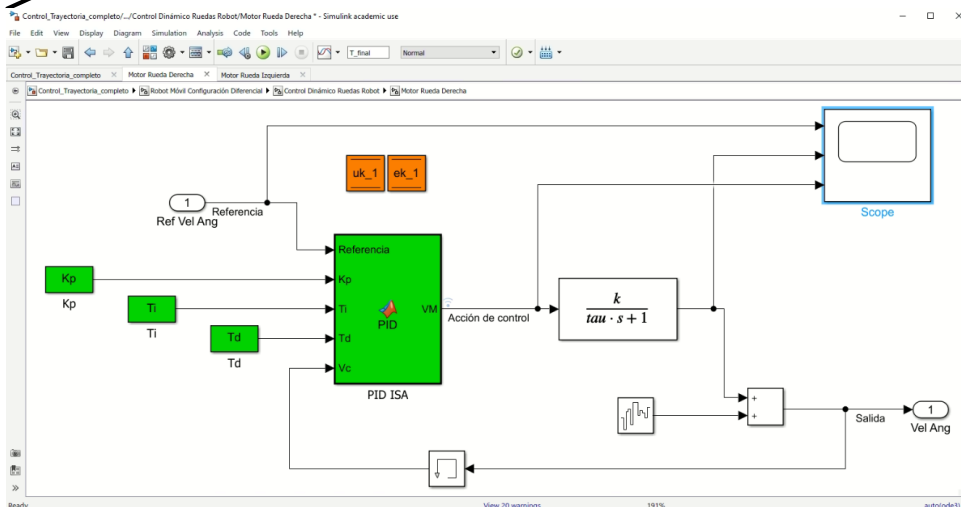
☒ Ruido

Valor Kp 8

Valor TI 0.07

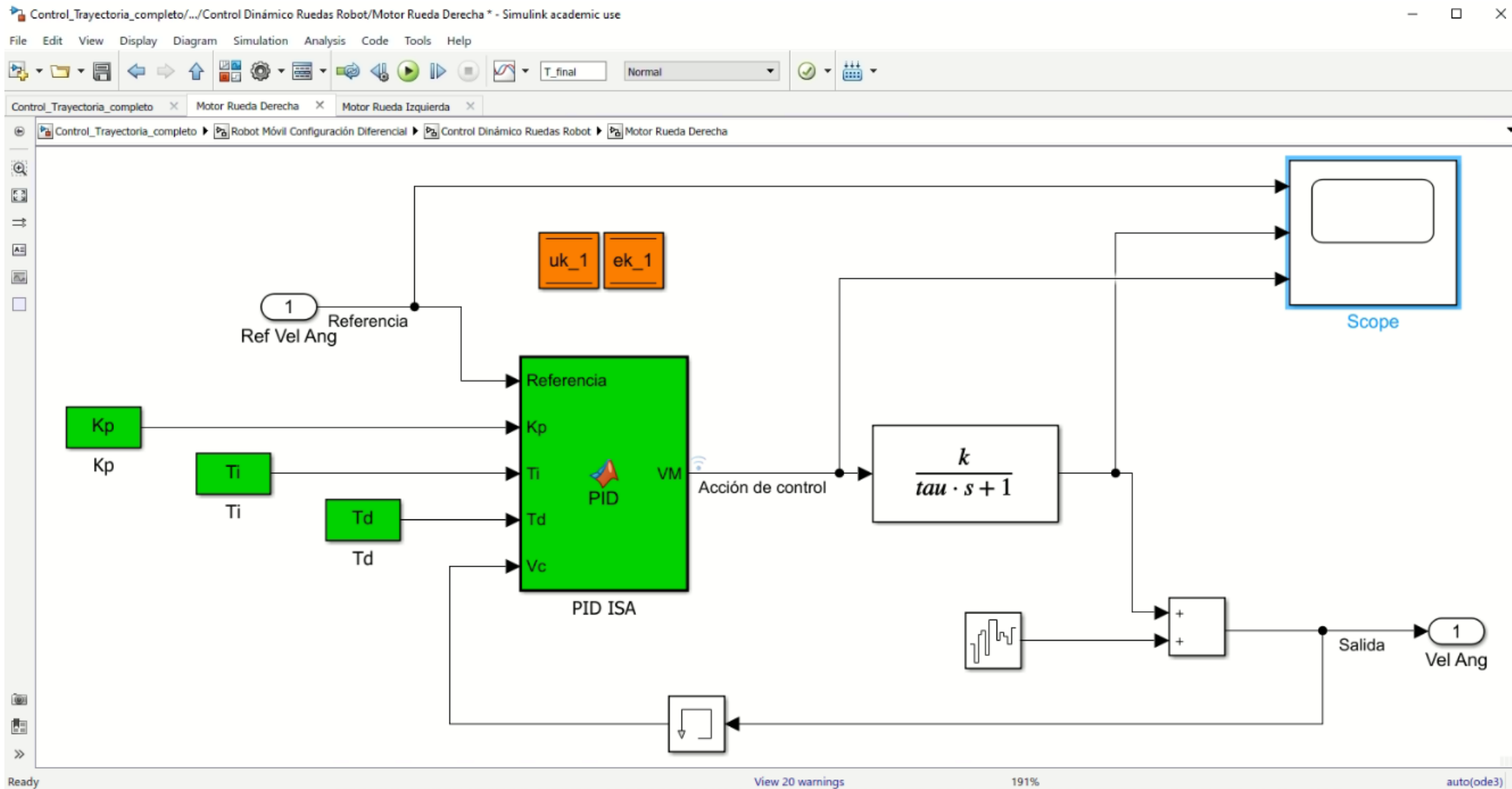
Valor Td .004

OK Cancel Help Apply



Completar con los parámetros obtenidos en la practicas 4 y 5

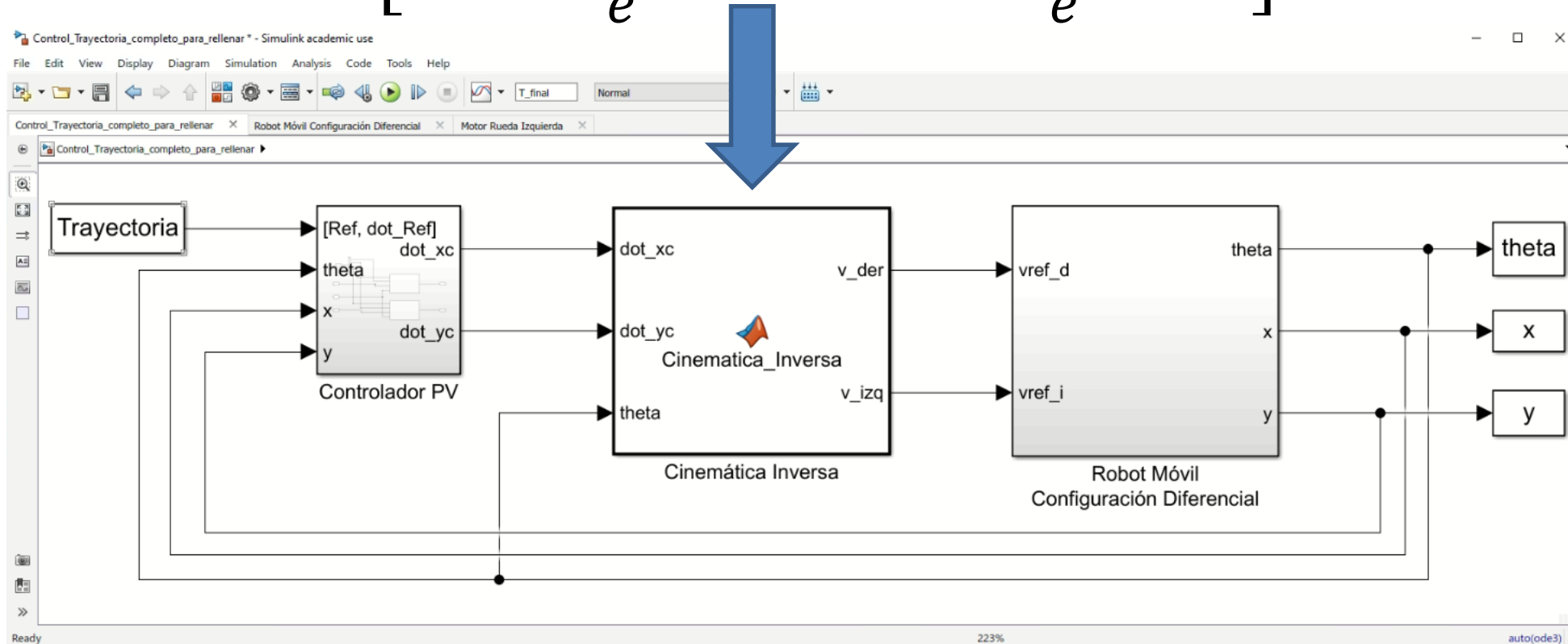
Trabajo a realizar



Trabajo a realizar

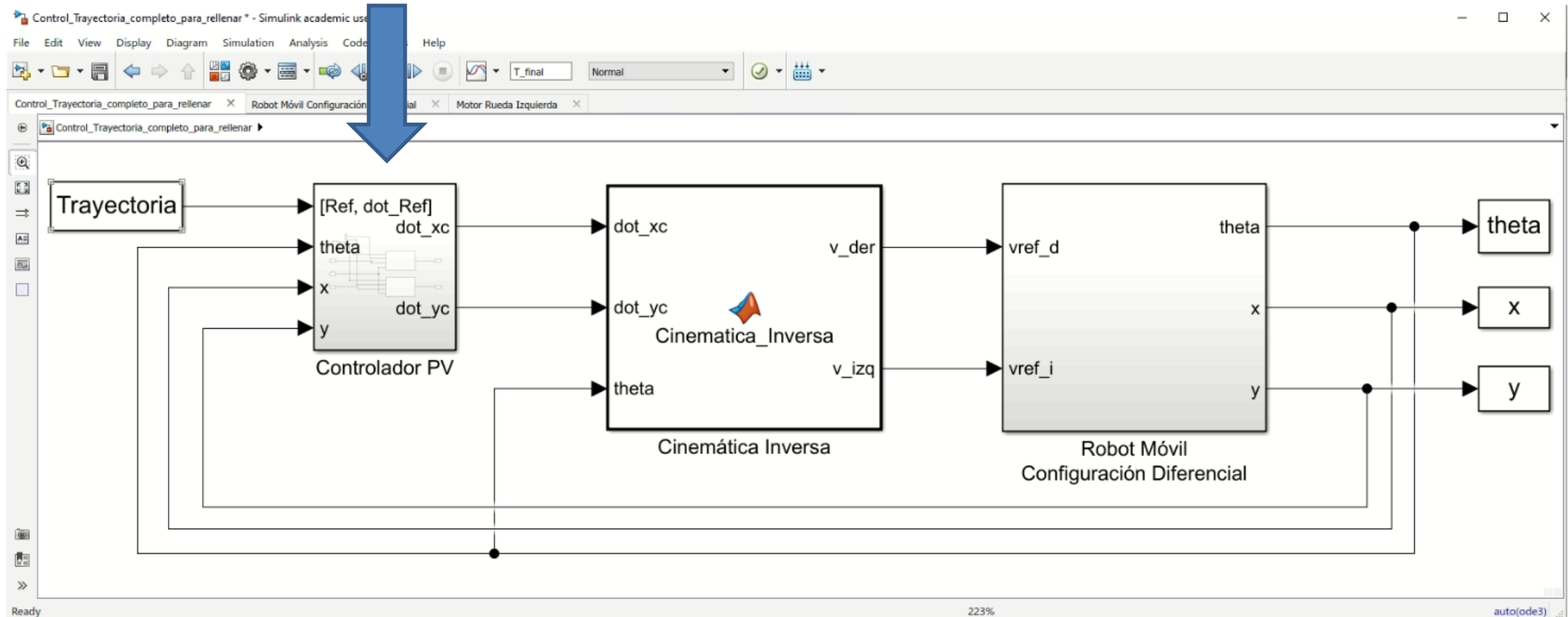
$$\begin{bmatrix} v_L \\ v_R \end{bmatrix} = \frac{1}{2} \begin{bmatrix} \cos \theta + \frac{b}{e} \cdot \sin \theta & \sin \theta - \frac{b}{e} \cdot \cos \theta \\ \cos \theta - \frac{b}{e} \cdot \sin \theta & \sin \theta + \frac{b}{e} \cdot \cos \theta \end{bmatrix} \begin{bmatrix} \dot{x}_c \\ \dot{y}_c \end{bmatrix}$$

$e = 100;$
 $b = 56;$



Trabajo a realizar

$$\begin{bmatrix} \dot{x}_c \\ \dot{y}_c \end{bmatrix} = \begin{bmatrix} \dot{x}_{ref} \\ \dot{y}_{ref} \end{bmatrix} + \begin{bmatrix} K_x & 0 \\ 0 & K_y \end{bmatrix} \begin{bmatrix} x_{ref} - (x + e \cdot \cos \theta) \\ y_{ref} - (y + e \cdot \sin \theta) \end{bmatrix}$$



Para calcular las derivadas de la referencia con respecto al tiempo:

```
diff_path = (path(2:end,:) - path(1:end-1,:))./Ts;  
diff_path = [diff_path; diff_path(end,:) ] ;  
t = ( 0:Ts:(length(diff_path)-1)*Ts )';
```

Forma de pasar (From workspace) al simulink en una sola variable y las condiciones iniciales del robot:

```
Trayectoria = [t, path, diff_path];
```

```
%Condiciones iniciales
```

```
e = 100;
```

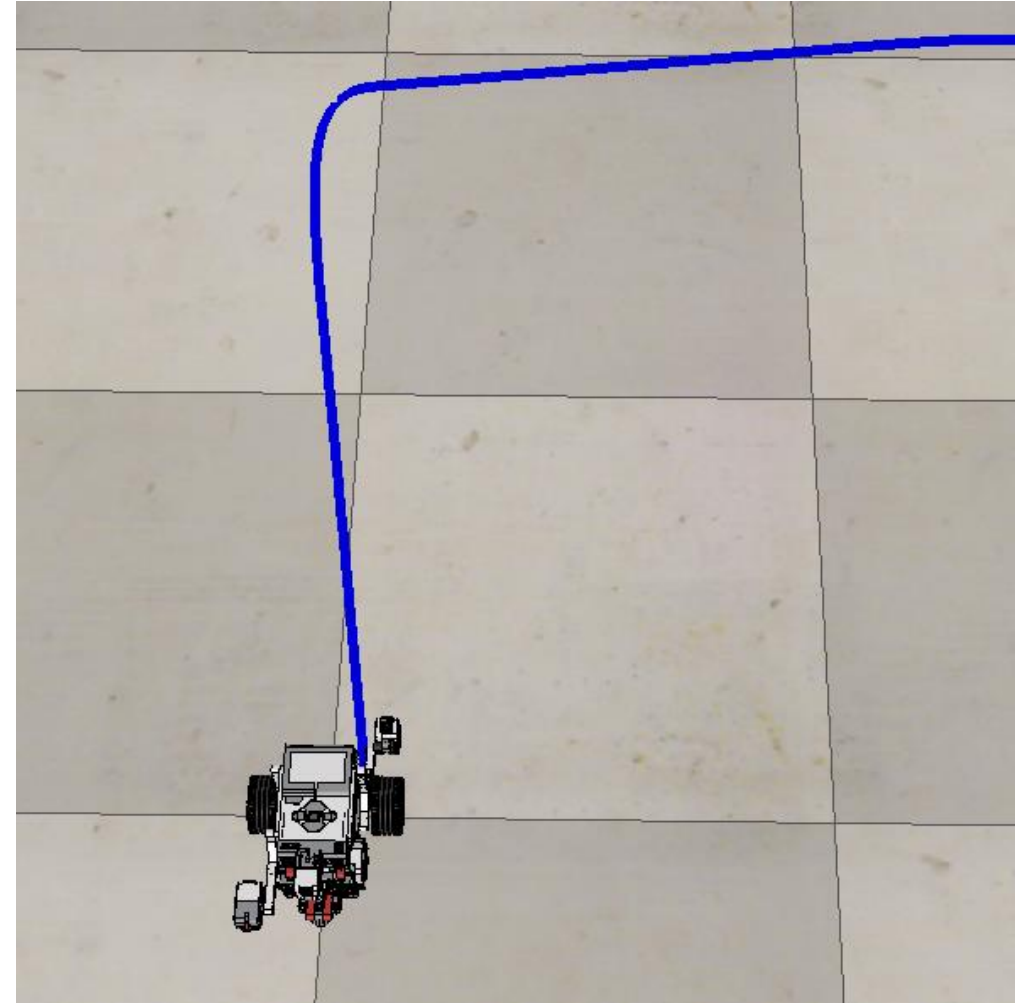
```
theta_init = pi/4;
```

```
% Para empezar deslocalizado del punto descentralizado (cuando la trayectoria a seguir empieza en 0,0)
```

```
x_init = -e*cos(theta_init);
```

```
y_init = -e*sin(theta_init);
```

Implementación del seguimiento de trayectoria en Matlab/CoppeliaSim



1. Implementar al algoritmo del punto descentrado en Matlab siguiendo el esqueleto del archivo `SeguimientoTrayectoria.m` completando las líneas correspondiente.
2. Simular en CoppeliaSim utilizando la escena `Trayectoria.ttt` y el código de cominzo `StartSeguimientoTrayectoria.m`
3. Graficar en la misma gráfica la trayectoria deseada del robot y la obtenida por odometría.
4. Calcular el error cuadrático medio del seguimiento de trayectoria obtenido.

Entregable (hasta el 18 de Abril)



departamento de ingeniería
de sistemas y automática

Entregable (entrega hasta el 18 de Abril)

Acompañando a la entrega de la memoria en la Tarea de poliformaT:

Subir a dicho directorio los ficheros desarrollados en la sesión de prácticas

Funciones, scripts y ficheros con datos obtenidos (ficheros de Matlab *.m y de Simulink*.slx)

Fichero con los nombres de los componentes del grupo

Memoria de la práctica 7 (entrega hasta el 18 de Abril)

Una memoria por grupo de prácticas en el que se incluya:

Explicación del trabajo realizado (no muy extensa)

Desarrollo de los ejercicios con comentarios, explicaciones del código y pruebas realizadas

Gráficas con simulaciones (leyendas, colores, nombres de los ejes,...)

Comentarios y conclusiones