

---

**Instituto Tecnológico de Costa Rica****Escuela de Ingeniería Electrónica****Trabajo Final de Graduación****Proyecto:** Método basado en aprendizaje reforzado para el control automático de una planta no lineal.**Estudiante:** Oscar Andrés Rojas Fonseca

I Semestre 2024

---

**Firma del asesor**

---

**Bitácora de trabajo**

Fecha	Actividad	Anotaciones	Horas dedicadas
05/03/2024	1. Continuación de pruebas para solucionar el error de OpenGL.	a) Luego de varias actualizaciones de drivers e instalación de los drivers de Nvidia necesarios el error de OpenGL persistió. b) Se optó por utilizar la herramienta <i>Jupyter Notebook</i> que funcionó sin errores.	5 horas
06/03/2024	2. Estudio del código de ejemplo <i>dpg.py</i> y <i>q-learning.py</i> del repositorio MPCRL [1].	a) Se identificaron las 2 clases principales del código ejemplo, correspondientes a un entorno definido en el momento <i>LtiSystem</i> y una clase de administración de la anterior mediante MPCRL llamada <i>LinearMpc</i> . b) La clase <i>LinearMpc</i> se encarga de hacer la distinción, dentro de sus funciones, entre el método <i>DPG</i> y <i>Q – Learning</i> .	6 horas

07/03/2024	<b>3.</b> Pruebas de definición de clases adaptadas en ejemplos de MPCRL a entornos Gymnasium <i>CartPole</i> y <i>Pendulum</i> [2].	a) La definición del entorno propio del ejemplo es un poco diferente comparado con la respectiva al <i>CartPole</i> de Gymnasium [2].	5 horas
07/03/2024	<b>4.</b> Ordenamiento del material agregado al documento final del proyecto (tesis).	a) Se copió información anteriormente adquirida en marco teórico. b) Supresión de líneas previas de la plantilla.	3 horas
08/03/2024	<b>5.</b> Estudio de las clases adaptadas <i>CartPoleSystem</i> y <i>CartPoleMpc</i> .	a) Al analizar las estructuras generadas se obtienen errores y parámetros que no corresponde. Es necesario replantear las clases.	4 horas
Total de horas de trabajo:			23 horas

## Contenidos de actividades

La estructura de clases utilizada en los ejemplos responde en primera instancia a una clase llamada en *LtiSystem*, encargada de definir las características del entorno basado en Gymnasium (*step*, *action* y *get\_state\_cost*) que se utiliza para los ejemplos, esto va desde las dimensiones del sistema de espacio de estados hasta los contenidos de sus matrices  $A$  y  $B$  [1].

Seguidamente se cuenta con la clase *LinearMpc*, la cual cuenta con los ajustes referentes a los parámetros del entrenamiento del modelo propio del ejemplo como  $V_0$ , sin embargo, la definición de dichos parámetros para el nuevo entorno *CartPole* puede recibir ciertas variaciones. Además, esta clase se encarga de la dinámica deseada del entorno y su comunicación para el entrenamiento siguiente [1].

Estas características son las que se deben adaptar en la nueva versión del código, primero con los entornos virtuales de Gymnasium y seguidamente, en el control del modelo imitador del PAMH objetivo.

# Referencias

- [1] F. Airaldi, A. Bietti, A. Casagrande, and A. Bemporad, “Learning model predictive control with policy gradients,” *IEEE Transactions on Automatic Control*, 2023.
- [2] T. F. Foundation, “Gymnasium documentation,” [https://gymnasium.farama.org/environments/classic\\_control/](https://gymnasium.farama.org/environments/classic_control/), 2024.