
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
19/03/2024	2. Continuación de pruebas para la definición de las clases referentes a <i>CartPole</i> .	a) Se definió una nueva versión de la clase <i>CartPoleSystem</i> utilizando las funciones <i>_init_()</i> , <i>reset()</i> , <i>step()</i> , <i>render()</i> , <i>close()</i> y <i>get_stage_cost()</i> .	6 horas
20/03/2024	3. Reunión de seguimiento con el asesor del proyecto.	a) Revisión de avance en el código y resaltado de errores de forma. b) Dada la complejidad de la adaptación de los ambientes <i>CartPole</i> al <i>MPC</i> [1], se decidió revisar otras fuentes, encontrando un ejemplo de implementación <i>CartPole</i> con <i>DQN</i> [2].	2 horas
20/03/2024	4. Pruebas de montaje del código <i>DQN</i> mencionado.	a) Creación de nuevo <i>env</i> para el funcionamiento del código en [2]. b) Pruebas de funcionamiento del código original (exitosa). c) Pruebas de implementación del código con su versión CUDA para mejor desempeño en SO Windows.	4 horas

21/03/2024	5. Continuación de pruebas para la definición de las clases referentes a CartPole.	a) Los parámetros del ejemplo <i>dpg.py</i> son diferentes al caso del <i>CartPole</i> por el objetivo de aprendizaje del modelo.	6 horas
22/03/2024	6. Continuación de pruebas para la definición de las clases referentes a CartPole.	a) Los parámetros del ejemplo <i>dpg.py</i> son diferentes al caso del <i>CartPole</i> por el objetivo de aprendizaje del modelo.	6 horas
Total de horas de trabajo:			23 horas

Contenidos de actividades

AAAA [1]

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] A. Paszke and M. Towers, “Reinforcement learning (dqn) tutorial,” *PyTorch*.