

---

**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
15/04/2024	1. Redefinición de la conversión del código para valores discretos ( <i>CartPole</i> ) a valores continuos ( <i>Pendulum</i> ).	a) El error en <i>select_action()</i> se corrigió pero desconfiguró parte de la función <i>optimize_model()</i> . b) Corrección del error en <i>optimize_model()</i> . c) Persisten los problemas de indexado y proceso.	6 horas
15/04/2024	2. Pruebas de entrenamiento del modelo ( <i>Pendulum</i> ).	a) Se entrenaron cuatro modelos diferentes a 600 episodios para comparar el efecto de cuatro propuestas de redes neuronales artificiales (ANN).	6 horas
16/04/2024	2. Búsqueda de la teoría de los métodos <i>PPO</i> y <i>actor – critic</i> dada la necesidad del manejo del <i>action space</i> con valores continuos.	a) SASASASAS.	4 horas
17/04/2024	3. Reunión de seguimiento con el asesor del proyecto.	a) Revisión de avance en el código y errores de forma. b) Se acordó continuar con el interés en los métodos como <i>PPO</i> como opción para el control.	2 horas
17/04/2024	2. Búsqueda de métodos para el manejo de valores continuos en <i>DQN</i> .	a) Aparece el <i>DDPG</i> . b) Opción de discretizar el <i>action space</i> .	4 horas
17/04/2024	2. Prueba de discretización del <i>action space</i> del env <i>Pendulum</i> .	a) Se logró adaptar el código del <i>PendulumDQN</i> a una versión discretizada <i>PendulumDQN<sub>d</sub>iscrete</i> , depende principalmente de la resolución seleccionada ( $n_{actions}$ ). b) Pruebas de entrenamiento de hasta 100 episodios.	4 horas

19/04/2024	4. Corrección de potenciales errores en el código <i>PendulumDQN</i> señalados por asesor.	a) Replanteo de función de recompensa <i>calculate_reward()</i> para evitar salto. b) Adición de lógica para guardado de <i>checkpoints</i> al entrenamiento y corrección del guardado del modelo.	6 horas
20/04/2024	4. Montaje y primera prueba del código <i>PPO</i> para <i>Pendulum</i> .	a) Revisión del error por cambio de versión <i>Gym</i> a <i>Gymnasium</i> .	6 horas
21/04/2024	4. Prueba de entrenamiento con versión base del <i>PPO</i> para <i>Pendulum</i> .	a) Entrenamiento del modelo con <i>render("human")</i> de 200k episodios. Mal desempeño. b) Entrenamiento del modelo con <i>render("rgb_array")</i> de 200M de episodios. En proceso.	6 horas
Total de horas de trabajo:			21 horas

## Contenidos de actividades

AAA [1].

## Referencias

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