

---

**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
12/02/2024	<b>1.</b> Búsqueda de repositorios en línea sobre RL.	a) Se encontró la librería <i>RLtools</i> en C++. . b) Creación del env para trabajar con C++. c) Ejemplo de implementación de <i>RLtools</i> .	4 horas
13/02/2024	<b>2.</b> Búsqueda de ejemplos de uso del modelo <i>Mamba</i> .	a) Se encontraron ejemplos de funcionamiento con <i>simple parallel scan implementation</i> . b) Estudio de la comparación entre el entrenamiento Mamba con <i>CUDA</i> , <i>mamba.py</i> y <i>RNN</i> . c) El repositorio <i>mamba.py</i> encontrado ejemplifica un uso simple de MAMBA.	5 horas
13/02/2024	<b>3.</b> Trabajo en la tesis del proyecto.	a) Se adaptó la plantilla para el proyecto. b) Introducción de línea guía de ideas (anteproyecto). c) Introducción de resumen de la bitácora semana 1.	3 horas

15/02/2024	4. Revisión del funcionamiento del código <i>RNAM_Synthetic.py</i> .	a) Primer proceso de entrenamiento de la versión base. b) Se verificó el registro con lo expuesto en la tesis de Jorge Brenes. c) Se probó el código de <i>RNAM_Real.py</i> sin éxito por la falta del directorio <i>../Datos_Recolectados/...</i>	6 horas
16/02/2024	5. Pruebas de variación de hiperparámetros al entrenamiento.	a) Cambios en el valor del <i>learning_rate</i> .	3 horas
Total de horas de trabajo:			21 horas

# Contenidos de actividades

## Resumen de repositorios encontrados

El ejemplo más completo corresponde a la librería encontrada *RLtools* [1], la cual cuenta con una descripción muy completa de lo que puede ofrecer al tabular y compartir capturas del entrenamiento de modelos de RL mediante algoritmos como *Twin Delayed DDPG (TD3)*, *Proximal Policy Optimization (PPO)* y *Soft Actor-Critic (SAC)*.

Se muestran ejemplos de implementación con casos como *Pendulum*, *Racing car*, *MuJoCo Ant-man* and *Acrobot* [1], similares a las capturas de la Figura 1.

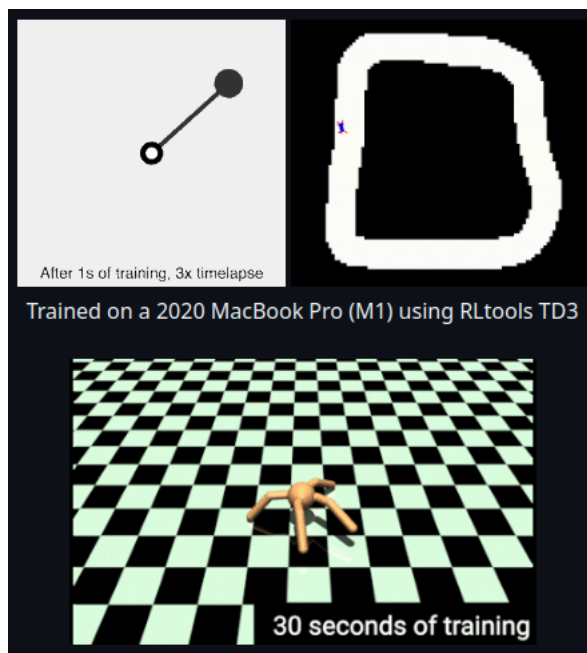


Figure 1: Capturas de implementación con RLtools [1].

De igual forma se encontró el repositorio de la librería *mpcrl* en python 2, para el entrenamiento de RL basado en modelos. La descripción del *README.md* es completa y se acompaña con ejemplos de funcionamiento y la captura facilitada mostrada en la Figura 2.

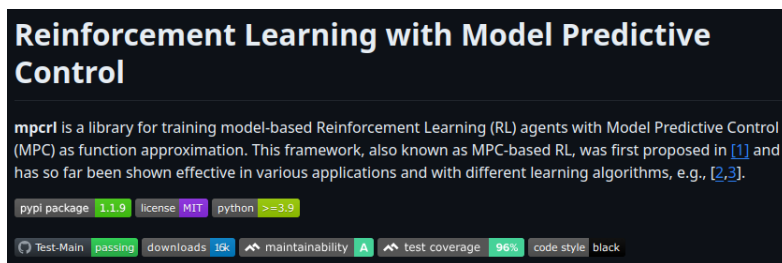


Figure 2: Capturas de descripción del modelo [1].

Para la búsqueda realizada para los repositorios de Mamba, se encontró el caso de [2], donde si se cuenta con implementación de Mamba, pero se trata de una ejemplificación de cálculos sencillos mediante PyTorch y en diferentes métodos para implementación.



Figure 3: Capturas de la descripción del modelo Mamba [2].

## Referencias

- [1] J. Eschmann, D. Albani, and G. Loianno, “Rltools: A fast, portable deep reinforcement learning library for continuous control,” 2023.
- [2] AlxndrRL, “mamba.py: a simple and efficient mamba implementation,” 2023.