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**
20/02/2024	**2**. Estudio del funcionamiento de RLtools (C++) junto con Python.	*a*) Revisión de opciones disponibles para el manejo de sistemas en ambos lenguajes de programación. *b*) Dadas las características, se prefiere el *C++ wrapper*.	5 horas
21/02/2024	**3**. Estudio de la comunicación entre el sistema (planta) y el módulo de control al sistema (Red neuronal).	*a*) Revisión de la teoría correspondiente en [1]. *b*) Revisión de ejemplos de funcionamiento del MPC [2].	5 horas
22/02/2024	**4**. Pruebas realizadas con la librería RLtools.	*a*) Instalación de dependencias y pruebas de paquetería [3].	6 horas
23/02/2024	**5**. Prueba de entrenamiento con datos reales del PAMH.	*a*) Se ejecutó el script *RNAM_Real.py* con una primera versión de los datos recolectados. Sin exito por tiempo de ejecución muy largo.	5 horas
		Total de horas de trabajo:	21 horas

Contenidos de actividades

C++ Wrapper

Para la utilización de algoritmos definidos en $C++$ como es el caso de la libretía RLtools y aplicarlos a modelos o bases en *Python*, es necesaria la "traduccción" o adaptación para manejar los dos sistemas, donde es posible traducir todo el modelo previo a $C++$ o la implementación de un *C++ wrapper*, en este caso preferido para no afectar el modelo de entrenamiento de la red neuronal mimetizadora, anteriormente definida, y su control en tiempo continuo.

Control predictivo del modelo (MCU)

La estructura de comunicación entre el sistema y el modelo de control mediante maching learning se efectua con base en funciones de optimización, donde se generan acciones controladas en cada paso para controlar alguna característica ($\hat{\mathbf{F}}$) y se toma la primera muestra de dicha acción ($\hat{x}_{k+1}$) para variar la respuesta del controlador enviada al sistema ($\mathbf{u}_j$), esto independiente y apllicable a sistemas lineales o no lineales. El proceso se simplifica en la Fig. 1.

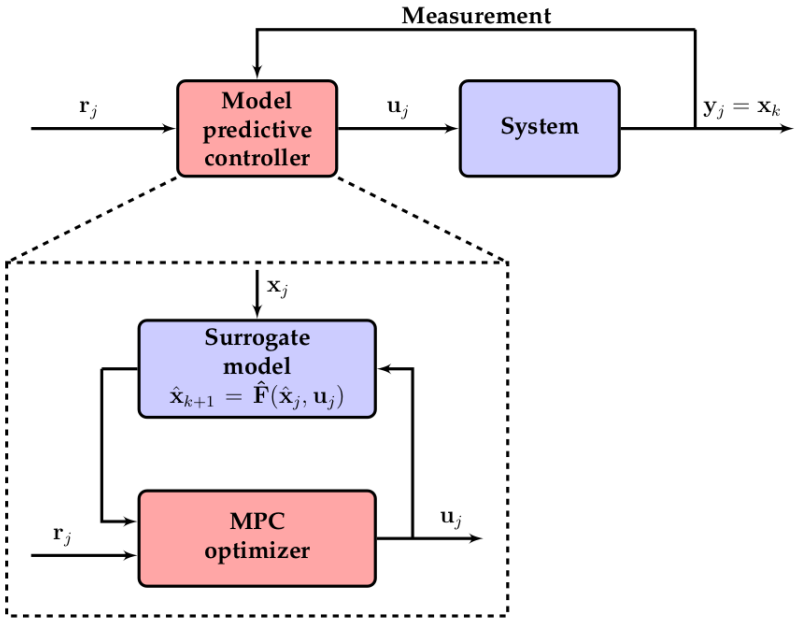

Figure 1: Esquemático del control predictivo del modelo (MPC) [1].

Pruebas de funcionamiento

Luego de instalar las dependencias necesarias para un primer entrenamiento usando *RLtools*, se logró el entrenamiento del modelo más simple de la librería, el péndulo con TD3. Ejemplo

de esto se muestra en la Fig. 2.

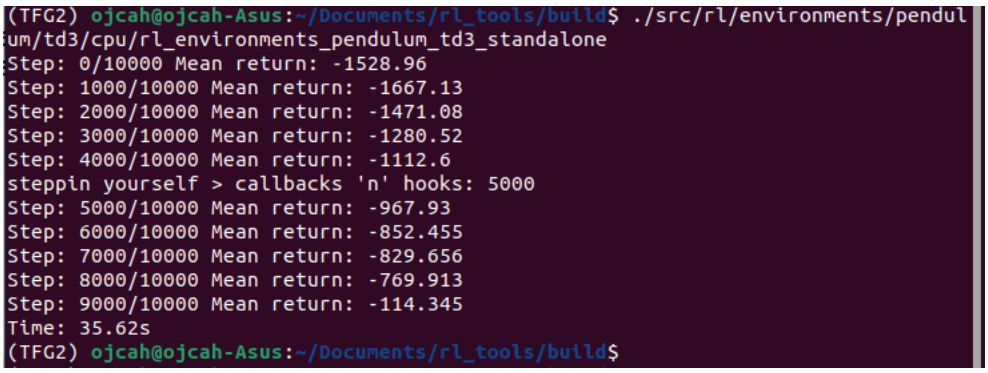

Figure 2: Entrenamiento preliminar del péndulo con RLtools.

Referencias

[1] S. L. Brunton and J. N. Kutz, *Data-Driven Science and Engineering*. Cambridge University Press, 2021.

[2] F. Airaldi, A. Bietti, A. Casagrande, and A. Bemporad, "Learning model predictive control with policy gradients," *IEEE Transactions on Automatic Control*, 2023.

[3] J. Eschmann, D. Albani, and G. Loianno, "Rltools: A fast, portable deep reinforcement learning library for continuous control," 2023.