

---

**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
05/02/2024	<b>1.</b> Estudio de la teoría de control para el péndulo amortiguado a hélice (PAMH).	a) Consulta a bibliografía de control automático: Nise (2020) y Ogata (2003). b) Revisión de material multimedia de Anibal Ruiz-Barquero referente al PAMH vía Youtube.	3 horas
06/02/2024	<b>2.</b> Estudio de la teoría de aprendizaje reforzado (RL).	a) Consulta a libros de texto como <i>Data-Driven Science and Engineering</i> (Brunton y Kutz, 2021). b) Revisión de material multimedia de Steven Brunton vía Youtube.	3 horas
06/02/2024	<b>3.</b> Revisión bibliográfica de algoritmos de aplicación de aprendizaje automático.	a) Consulta al libro <i>Reinforcement Learning: An introduction</i> (Sutton y Barto, 2020) para mayor detalle. b) Revisión de otros métodos de aprendizaje automático. c) Ejemplos de implementación de las redes neuronales recurrentes (RNN) por Patrick Loeber vía Youtube y la tesis de graduación de Jorge Brenes.	3 horas

07/02/2024	4. Revisión de repositorios en línea de métodos de aplicación de aprendizaje automático.	a) Búsqueda preliminar de repositorios generales de RL mediante Github. b) Selección de códigos con enfoques similares al control del PAMH.	5 horas
09/02/2024	5. Creación del ambiente de trabajo anaconda para montaje de la red neuronal mimetizadora (RNAS).	a) Revisión de bibliotecas utilizadas por el código base de la RNAS. b) Instalación/revisión de versiones adecuadas de <i>Python</i> , <i>ArgumentParser</i> , <i>Numpy</i> , <i>Matplotlib</i> , <i>TensorFlow</i> y <i>Weights&amp;Biases</i> .	2 horas
09/02/2024	6. Pruebas de funcionamiento de la red neuronal mimetizadora (Synthetic-PAMH.py).	a) Estudio del código de la RNAS. b) Error en el proceso por falta de cuenta y permisos del autor en W&B. c) Creación de cuenta y proyecto en W&B.	3 horas
09/02/2024	7. Estudio del funcionamiento de la herramienta Weights & Biases (W&B).	a) Revisión de material en línea sobre el uso de W&B. b) Ejemplos de implementación de W&B.	2 horas
Total de horas de trabajo:			21 horas

## Contenidos de actividades

### Resumen de teoría PAMH

El péndulo amortiguado a hélice corresponde a una planta de laboratorio compuesta de un motor BLDC a 12V controlado por torque, una masa pequeña, péndulo y soportes de aluminio de baja fricción. Un modelo simplificado del sistema se muestra en la Figura 1.

El objetivo de dicho sistema es controlar la magnitud del ángulo  $\theta_p$ , únicamente ejerciendo torque al accionar a una distancia  $l_m$  el motor con una fuerza  $F_m$  y movimiento de su masa  $m_m$ , mientras a una distancia de  $l_p$  del centro se encuentra una masa  $m_p$  que contrarresta el movimiento.

De manera que al analizar el sistema con sumatoria de torques se obtiene la constante

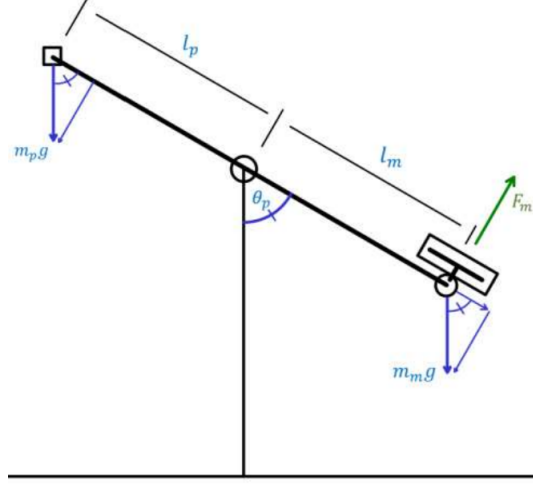


Figure 1: Modelo simplificado del PAMH.

de rosamiento central  $B$  (en caso de existir) junto con la inercia ejercida  $J_p$ . Por lo tanto, se definen las variables de estado siguientes y sus ecuaciones de estado mostradas en 1.

$$\begin{aligned} x_1 &= \theta_p & x_2 &= \dot{\theta}_p & y &= x_1 = \theta_p \\ \left\{ \begin{array}{l} \dot{x}_1 = x_2 \\ \dot{x}_2 = -\frac{B}{J_p}x_2 + (m_p l_p - m_m l_m) \frac{g}{J_p} \text{sen}(x_1) + \frac{l_m}{J_p} F_m \end{array} \right. & (1) \end{aligned}$$

## Resumen de teoría RL

AAAA

## Ambiente de trabajo anaconda para la RNAM

Luego de una revisión rápida del código perteneciente al repositorio del trabajo de Jorge Brenes, se instalaron versiones de bibliotecas necesarias para el funcionamiento de los registros, resultando en una lista simplificada como la mostrada en la Figura 2.

```

(TFG1) ojcah@ojcah-Asus:~$ conda list | grep -E 'matplotlib|python|tensorflow|keras|numpy|pandas|argparse'
gitpython                3.1.41                pypi_0    pypi
keras                     2.11.0                py310h06a4308_0
keras-preprocessing       1.1.2                 pyhd3eb1b0_0
matplotlib                3.8.0                py310h06a4308_0
matplotlib-base           3.8.0                py310h1128e8f_0
numpy                     1.26.3                py310h5f9d8c6_0
numpy-base                1.26.3                py310hb5e798b_0
pandas                    2.1.4                py310h1128e8f_0
python                    3.10.13               h7a1cb2a_0
python-dateutil           2.8.2                 pyhd3eb1b0_0
python-flatbuffers         2.0                   pyhd3eb1b0_0
python-tzdata             2023.3                pyhd3eb1b0_0
tensorflow                 2.11.0                mkl_py310hb40ee82_0
tensorflow-base           2.11.0                mkl_py310he5f8e37_0
tensorflow-estimator       2.11.0                py310h06a4308_0
(TFG1) ojcah@ojcah-Asus:~$
(TFG1) ojcah@ojcah-Asus:~$
(TFG1) ojcah@ojcah-Asus:~$

```

Figure 2: Lista simplificada de bibliotecas del *environment* TFG1.