

**Reporte proyecto final:** Recomendación de variables manipulables en molinos SAG con Deep Reinforcement Learning Offline

**Nombre**: José Luis Cádiz.

**Curso**: Seminario de robótica y sistemas autónomos - EL7021 otoño 2023.

**Profesor**: Javier Ruiz del Solar.

**Introducción**: Para abordar el proyecto en cuestión se hizo uso de la librería de Offline Reinforcement Learning [**d3rlpy**](https://takuseno.github.io/d3rlpy/),esto con el objetivo de centrar los esfuerzos en el modelamiento de los procesos que se buscan optimizar. A continuación, se muestran los resultados del modelamiento para los fenómenos **“Estabilización de TPH mediante recomendación de su nivel de carga”** y **“Reducción de cuellos de botella por alto nivel de arcilla mediante recomendación de rpm y porcentaje de sólidos”.** El algoritmo utilizado para aprender políticas optimas en ambos fenómenos fue [**Advantage Weighted Actor-Critic**](https://arxiv.org/pdf/2006.09359.pdf)**.**

Para verificar que las políticas aprendidas son consistentes, se compararan las distribuciones de lo que indican las políticas aprendidas vs la política adoptada según la data histórica, adicionalmente se analizaran algunas series de tiempo con el objetivo de chequear la consistencia de las políticas.

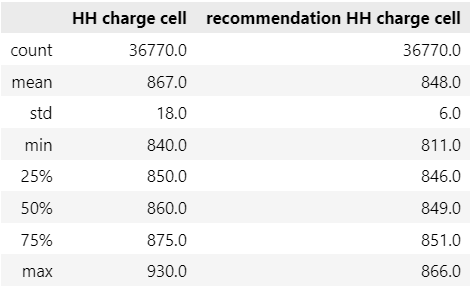
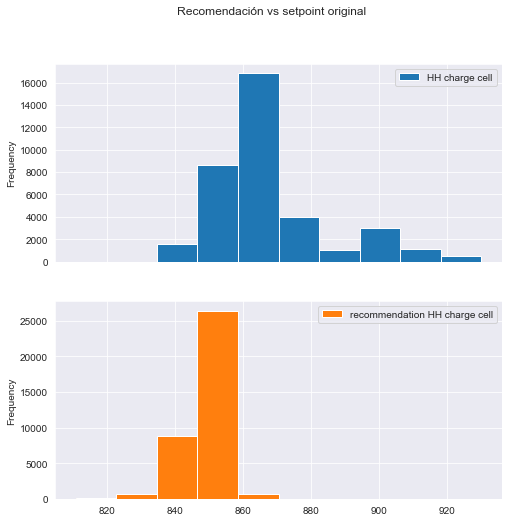
**Modelamiento**:

**Resultados**:

1. **Estabilización de TPH mediante recomendación de su nivel de carga**:

* **Visualizaciones**:

**Figura 1**: Comparación de distribución de políticas.





**Figura 2**: Estado de perdida, la recomendación sugiere bajar el limite de carga, se observa que TPH cae al no seguir recomendación.

**A graph of a stock market

Description automatically generated**

**A graph of a graph

Description automatically generatedFigura 3**: Estado de no perdida, caso en que recomendación supera la política original.

* **Análisis**:

1. **Reducción de cuellos de botella por alto nivel de arcilla mediante recomendación de rpm y porcentaje de sólidos**:

**Conclusiones:** Si bien el enfoque Offline del Reinforcement Learning permite aprender políticas optimas a partir de data histórica sin necesidad de experimentar con la planta, de igual manera surge la necesidad de poder testear las políticas aprendidas y para esto es necesario tener al menos un modelo de la planta que permita simular la incorporación de las políticas aprendidas dentro del sistema.

A partir de la necesidad de tener un modelo para poder testear las políticas aprendidas, se genera una cierta contradicción, ya que, si se tuviese un modelo de la planta, lo podríamos utilizar como ambiente y aprender políticas optimas a partir de métodos convencionales de Reinforcement Learning.

Reinforcement Learning Offline es método muy interesante para aprender políticas optimas solamente a partir de data histórica, sin embargo, es necesario diseñar metodologías para poder testear de manera robusta las políticas aprendidas a partir del mismo tipo de estructura de información con la cual se entrena el algoritmo.

También es importante mencionar que para obtener resultados satisfactorios utilizando este enfoque de aprendizaje, es muy importante entender la dinámica del sistema que se busca optimizar, ya que es de vital importancia modelar de manera correcta dichas dinámicas como un proceso de decisión de Márkov.

Se concluye que los resultados han sido parcialmente satisfactorios ya que, si bien las políticas aprendidas son consistentes con los resultados esperados a partir de conocimiento previo, no se puede decir de manera categórica que las políticas aprendidas son óptimas o seguras debido a que no es posible la experimentación.

Finalmente, a partir de todo el proceso de experimentación con la data, entrenamiento e intento de testear, se concluye que, si se llega a tener un buen modelo de simulación de la planta junto con data histórica del comportamiento de esta, debería ser posible alcanzar muy buenos resultados en el proceso de encontrar políticas optimas de una manera robusta y segura.