

Proyecto: Control Optimo y Técnicas de Aprendizaje Reforzado

◁ Entrega: 10:00 PM - 11 de julio, 2024 ▷

1. **Aprendizaje:** Considerando el caso de robot móvil de tracción diferencial y los ejemplos entregados para el mismo en la ayudantía, en este proyecto usted aprenderá a:
 - (a) Implementar un controlador basado en técnicas de **aprendizaje reforzado** (*reinforced learning* RL) como una **solución al problema de control óptimo**.
 - (b) Comparar el desempeño con respecto al control **PID discreto tradicional** los **controladores LQI discreto, y RL SAC**.
 - (c) Utilizar librerías avanzadas (en la frontera de las técnicas actuales) para: (i) el desarrollo de técnicas de control basadas en aprendizaje reforzado (librería **Stable Baselines3**), (ii) la **simulación físicamente exacta de robots y sistemas multicuerpo** (librería **MuJoCo**).
2. **Anexos:** Este proyecto se acompaña con códigos de ejemplos presentados en la ayudantía y publicados en la página web del curso https://cursos.canvas.uc.cl/courses/73150/files/folder/Ayudantias/Ayudantia_1, para una introducción rápida en estos temas.
3. **Instrucciones generales:**
 - (a) Puede realizar el proyecto de manera individual o en grupos de 2 personas.
 - (b) Utilice el **modelo del robot móvil de tracción diferencial**: “**car1.xml**” presentado en la ayudantía.
 - (c) Si no tiene un computador adecuado, puede utilizar los computadores en los laboratorios de computación de la Escuela de Ingeniería, o habilitar una cuenta en Google Colab <https://colab.research.google.com/>.
 - (d) **No es necesario usar PyTorch**, basta la funcionalidad de **Stable Baseline3**. Si desea añadir PyTorch a Colab vea <https://colab.research.google.com/github/pytorch/xla/blob/master/contrib/colab/getting-started.ipynb>. En caso se implementar soluciones en Colab, deberá dejar una versión ejecutable compartida que podamos evaluar directamente en la página Colab, además de los scripts que entregue en su informe.
 - (e) Entregables: (1) **informe escrito y códigos** subidos a Canvas **incluido el archivo del modelo del agente RL** para poder ejecutar su controlador RL y probarlo, (2) **presentación oral presencial explicando el trabajo desarrollado en la fecha de entrega (11 de julio 2024, 10 AM)**.
 - (f) Más detalles en la sección indicaciones.

1. Preguntas

Desarrolle e implemente controladores de seguimiento de trayectoria de referencia (t, x^*, y^*, θ^*) para el robot de tracción diferencial empleando un:

1. Controlador PID.
2. Controlador LQI.
3. Controlador RL (reinforced learning) con el método SAC (soft-actor critic).

Notas:

1. Defina su trayectoria de referencia como un cuadrado de 2×2 metros.
2. Optimece cada controlador de la mejor manera que pueda para idealmente lograr: (i) un seguimiento de alta exactitud (bajo error de seguimiento), (ii) bajo consumo de energía, y (iii) mínimo tiempo de ejecución de la trayectoria.

2. Indicaciones para el informe

1. El informe debe contener:
 - (a) Debe presentar un informe explicando cada pregunta por separado, indicando las partes principales de cada controlador y mostrando métricas de desempeño: error de convergencia versus tiempo, tiempo de convergencia, error en régimen permanente, robustez ante perturbaciones, variables manipuladas versus tiempo, energía y costo versus tiempo según la métrica de desempeño que usted elija. Justifique la decisión de la métrica de desempeño.
 - (b) Sección comparativa del desempeño de todos los controladores con al menos: (i) un gráfico que muestre el error de convergencia versus tiempo para cada controlador (curvas superpuestas en un mismo plot) en condiciones equivalentes, como un mismo punto de destino, para que la comparación sea válida; (ii) un gráfico que muestre las métricas de desempeño versus tiempo comparadas; (iii) un gráfico que muestre las magnitudes de las variables manipuladas versus tiempo de cada controlador; (iv) una tabla comparativa de tiempos de convergencia y errores finales en régimen permanente para cada controlador; (v) una discusión crítica sobre el desempeño, ventajas y desventajas de cada controlador, así como aquello que considera han sido sus principales aprendizajes de la actividad.
2. En la fecha de entrega deberá presentar presencialmente su implementación y resultados en 5 minutos (presentación PPT/PDF) y 5 minutos de preguntas. Si trabaja en grupo con otra persona, ambos podrán ser consultados sobre el código implementado y deben conocer los detalles, de lo contrario se restará puntaje de la nota de presentación.
3. Tanto el informe y presentación debe seguir estándares de la comunicación técnica profesional. La claridad y rigurosidad con que presente es importante, puesto que no será posible apelar la nota de presentación una vez realizada o entregado el informe.
4. Declare todas sus fuentes cuando utilice códigos o ideas que encuentre en textos, o páginas y videos Internet.

3. Información de apoyo sobre implementaciones de RL SAC

1. Se recomienda utilizar el **algoritmo SAC** implementado en la librería Stable Baselines3 <https://stable-baselines3.readthedocs.io/en/master/modules/sac.html>.
2. Una explicación resumida del algoritmo SAC se encuentra en: <https://spinningup.openai.com/en/latest/algorithms/sac.html>.
3. No es necesario usar PyTorch, pero pueda utilizar las implementaciones de algoritmos RL de PyTorch si lo desea. Un tutorial introductorio puede encontrarse en <https://pytorch.org/tutorials/>.
4. Vea el tutorial <https://towardsdatascience.com/soft-actor-critic-demystified-b8427df61665> para una explicación intuitiva de la teoría y ejemplo de aplicación del método **SAC usando PyTorch**.
5. Otro ejemplo de implementación aplicada al **problema del péndulo invertido** lo puede encontrar en: <https://colab.research.google.com/drive/1jMc2lvFICASGfVutBv9rWkke97AM30aF?usp=sharing>, cuyo resultado puede visualizarse en <https://www.youtube.com/watch?v=r1iCftTz10c>.
6. Puede encontrar más detalles sobre el método SAC en otra explicación introductoria en <https://spinningup.openai.com/en/latest/algorithms/sac.html>, y sobre el **algoritmo TD3 (Twin-Delayed DDPGO (Deep Deterministic Policy Gradient))** en <https://spinningup.openai.com/en/latest/algorithms/td3.html>. También encontrará **tutoriales sobre aprendizaje reforzado usando Gymnasium (ex-OpenAI Gym)** en <https://gymnasium.farama.org/tutorials/>, pero no existe un tutorial en Gymnasium que explique el método SAC combinando Gymnasium y Stables Baselines3 o PyTorch. Otra fuente de información son algunos videos que muestran implementaciones de SAC con PyTorch en <https://www.youtube.com/watch?v=K2qjAixgLqk>, <https://www.youtube.com/watch?v=ioidsRlf79o>. **Implementaciones con Tensorflow+Keras** se pueden encontrar en <https://www.youtube.com/watch?v=K2qjAixgLqk>.
7. La variedad de algoritmos de aprendizaje reforzado existentes es muy amplia. Un muy buen **resumen de la taxonomía** se encuentra en https://spinningup.openai.com/en/latest/spinningup/rl_intro2.html. Podrá encontrar links a los artículos originales de los algoritmos, muchos de los cuales han surgido en los últimos 7 años, por lo que realmente son un tema de frontera que probablemente seguirá evolucionando vertiginosamente en los próximos años.

Colaboración académica y conducta

Nos preocupa que no declaren como propio el trabajo de otros. En primer lugar porque necesitamos poder evaluar los aprendizajes propios de cada estudiante, independientemente del de los demás. Por otro lado, la honestidad es un valor esencial de toda persona confiable, íntegra y respetable.

Distintas instituciones y lugares pueden tener definiciones diferentes sobre formas de comportamiento colaborativo se consideran aceptables. En este curso pueden discutir ideas con otros, pero se espera que escriban sus propias soluciones de forma independiente, sin referirse a las soluciones de otros. En cuanto a códigos, sólo podrá compartir el comportamiento de entrada-salida de sus programas, pero no su implementación. Los grupos deben trabajar por separado, pero pueden compartir ideas sobre cómo probar su implementación, no la implementación misma. Por favor, recuerden que si comparten su solución con otro estudiante o grupo, aunque no hayan copiado de otros, estarían incumpliendo el Código de Honor UC.

Ejecutamos un software de detección de similitudes en todos los códigos presentados, incluyendo códigos de semestres anteriores y cualquier solución encontrada en sitios web públicos. Cualquier estudiante de este curso que viole el Código de Honor UC será reportado a la Secretaría General de la UC para iniciar un proceso de Responsabilidad Académica y Disciplinaria <https://secretariageneral.uc.cl/>. Lo correcto es actuar con honestidad en todo momento, pero bajo estrés o presión a veces se olvida. Recuerden siempre que cuidar su honor e integridad profesional valen muchísimo más que una nota o que exponerse a sanciones graves. Por esta razón, es preferible entregar menos, no copiar y ser honesto.