

Proyecto: Control Optimo y Técnicas de Aprendizaje Reforzado

◁ Entrega: 2:00 PM - 11 de julio, 2022 ▷

1. **Aprendizaje:** Considerando el caso de un brazo robótico de dos grados de libertad y los ejemplos entregados para un péndulo invertido, 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, OSC 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 OpenAI Gym), (ii) la simulación físicamente exacta de robots y sistemas multicuerpo (librería MuJoCo), (iii) la implementación de técnicas de aprendizaje de máquina (librería PyTorch para redes neuronales profundas, aprendizaje profundo e inteligencia artificial), (iv) la planificación de trayectorias y control de brazos robóticos (librería ABR Control).
2. **Anexos:** Este proyecto se acompaña con códigos de ejemplos publicados en la página web del curso, para una introducción rápida en estos temas.
3. **Indicaciones:**
 - (a) El proyecto puede realizarse de manera individual o en grupos de 2 personas.
 - (b) Utilice el modelo del brazo robótico de 2 grados de libertad (2-DOF planar): "Reacher-v2" de MuJoCo integrado a OpenAI. Ver ejemplo `gym_example_2.py` e info en: <https://www.gymnasium.ml/environments/mujoco/reacher/>.
 - (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/>, y añadir PyTorch <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.
 - (d) Entregables: (1) informe escrito y códigos subidos a Canvas, (2) presentación oral presencial explicando el trabajo desarrollado en la fecha de entrega (11 de julio 2022, 2 PM).
 - (e) Más detalles en la sección indicaciones.

1. Preguntas

Empleando las técnicas que se señalan a continuación desarrolle e implemente controladores para un brazo robótico de dos grados de libertad (brazo robótico 2-DOF planar) que lleven el extremo efector a una posición de referencia (x^*, y^*) .

1. Implemente un controlador PD (proporcional-derivativo) en el espacio de las articulaciones.
2. Implemente un controlador LQI en el espacio de las articulaciones.
3. Implemente un controlador OSC (operational-space control), que entregue torques para las articulaciones a partir de cálculos en el espacio de operación o espacio de tarea del brazo robótico.
4. Implemente un controlador RL (reinforced learning) con el método SAC (soft-actor critic).

Notas:

1. Vea la sección “Recomendaciones para cada pregunta” para consejos y ayuda que pueda orientarlo en la realización de cada pregunta.
2. Deberá utilizar el modelo de brazo 2-DOF *Reacher-v2* de MuJoCo (no otro). Se adjunta con los códigos de ejemplo del proyecto la información paso a paso para la instalación de MuJoCo, OpenAI Gym, mujoco-py.
3. El brazo 2-DOF *Reacher-v2* de MuJoCo está compuesto por dos eslabones tipo cápsula de 0.1 m de largo y radio 0.01 m, con densidad uniforme de 1000 kg/m³. Las cápsulas son cilindros con semiesferas en los extremos. Los actuadores tienen reducciones 200:1, y la gravedad por defecto es 9.81 m/s². MuJoCo emplea la convención de unidades MKS y SI. Para mayor información del modelo físico revise el archivo `xml` con definiciones del *reacher* que se encuentra en la carpeta de instalación: `C:\Anaconda3\Lib\site-packages\gym\envs\mujoco\assets\reacher.xml` y la información sobre modelación en MuJoCo <https://mujoco.readthedocs.io/en/latest/modeling.html>. La definición del entorno creado en OpenAI Gym para el modelo *Reacher* se encuentra en: <https://www.gymnasium.ml/environments/mujoco/reacher/>.

2. Indicaciones

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. Este es un curso de postgrado y en este punto de la carrera no es posible aceptar trabajos desprolijos. 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 sus fuentes cuando utilice códigos o ideas que encuentre en textos, o páginas y videos Internet.
 5. Recuerde que el proyecto es individual o en grupos de dos personas máximo.
 6. **Colaboración académica y conducta:** Nos preocupa la colaboración académica y la honestidad porque es importante que seamos capaces de evaluar sus propios aprendizajes (independientemente del de sus compañeros), así como el que no declaren como propio el trabajo de otros, ya que es una parte importante de la integridad en sus futuras carreras profesionales. Distintas instituciones y lugares pueden tener definiciones diferentes de qué 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. Deben trabajar por separado, pero pueden compartir ideas sobre cómo probar su implementación, no la implementación misma. Por favor, recuerde que si comparte su solución con otro estudiante o grupo, aunque no haya copiado de otros, está violando 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 persona que viole el Código de Honor UC será remitida 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. Recuerde siempre que cuidar su honor e integridad profesional valen más que una nota o exponerse a sanciones graves, por lo que es preferible entregar menos, no copiar y ser honesto.

3. Recomendaciones para cada pregunta

1. Control PD:
 - (a) Para desarrollar el controlador PD en el espacio de las articulaciones puede utilizar la solución exacta (algebraica) de la cinemática inversa para obtener los angulos de referencia dada la posición del target o el método numérico de Newton-Raphson. Vea las ecuaciones de cinemática en el capítulo 6 de Modern Robotics, de Kevin Lynch y Frank Park, Cambridge University Press, 2017, <http://hades.mech.northwestern.edu/images/2/25/MR-v2.pdf>, http://hades.mech.northwestern.edu/index.php/Modern_Robotics, y los códigos de ejemplo: http://hades.mech.northwestern.edu/index.php/Modern_Robotics#Software. El capítulo 11 aborda el control PID en el espacio de las articulaciones para brazo robóticos en las secciones 11.4.1 y 11.4.2.

- (b) También puede implementar la solución empleando la librería ABR para control de brazos robóticos usando el modelo `twolink` y el controlador de la clase `Joint(Controllor)`, pero note que el modelo `twolink` tiene en el archivo `xml` correspondiente parámetros distintos al brazo `Reacher`. Para mayor información consulte https://github.com/abr/abr_control, <https://morioh.com/p/55de5506da8b>. Lo más sencillo puede ser entender la implementación de la librería ABR para adaptar el código y aplicarlo directamente al modelo `Reacher` sin cargar el modelo `twolink`. Otra opción podría ser cambiar el modelo `twolink` por el `Reacher` en ABR, pero cuidado que el estado del modelo simulado en el entorno OpenAI Gym sea el que provee la información al modelo ABR.

2. Control LQI:

- (a) Utilice las ecuaciones del modelo dinámico y los apuntes del curso para la implementación de controladores LQR-LQI. Para las ecuaciones del modelo dinámico de un brazo robótico 2-DOF planar vea el capítulo 8 de *Modern Robotics*, de Kevin Lynch y Frank Park, Cambridge University Press, 2017 (mencionado anteriormente). También puede leer los tutoriales de Travis DeWolf publicados en el blog Study Wolf, <https://studywolf.wordpress.com/site-index/> en la sección “Control theory” y “LQR, iLQR, and related methods”.

3. Control OSC:

- (a) Utilice las ecuaciones del modelo dinámico como se mencionó en el punto anterior. Para información sobre el control en el espacio de la tarea (*task-space control* u OSC vea el capítulo 11.4.3 de *Modern Robotics*, de Kevin Lynch y Frank Park, Cambridge University Press, 2017. También puede consultar los tutoriales de Travis DeWolf publicados en el blog Study Wolf, <https://studywolf.wordpress.com/site-index/> en la sección “Control theory”.

4. Control RL SAC-TD3:

- (a) Para la implementación del método SAC TD3 (Twin-Delayed DDPGO (Deep Deterministic Policy Gradient)) se sugiere utilizar PyTorch <https://pytorch.org/tutorials/>.
- (b) 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.
- (c) Otro ejemplo de implementación al péndulo invertido lo puede encontrar en: <https://colab.research.google.com/drive/1jMc2lvFICASGfVutBv9rWkke97AM30aF?usp=sharing>, cuyo resultado puede visualizar en <https://www.youtube.com/watch?v=r1iCftTzl0c>.
- (d) Puede encontrar más detalles sobre el método SAC en otra explicación introductoria en <https://spinningup.openai.com/en/latest/algorithms/sac.html>, y TD3 <https://spinningup.openai.com/en/latest/algorithms/td3.html>. También encontrará tutoriales sobre aprendizaje reforzado con OpenAI Gym en <https://www.gymnasium.ml/content/tutorials/>, pero no existe uno en OpenAI Gym que explique el método SAC combinando OpenAI Gym y 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>. Otras implementaciones con Tensorflow+Keras se pueden encontrar en <https://www.youtube.com/watch?v=K2qjAixgLqk>.
- (e) 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 5 años, por lo que realmente son un tema de frontera que probablemente seguirá evolucionando vertiginosamente en los próximos años.