Continuous control of a Robot Manipulator using Deep Deterministic Policy Gradient

Maithili Shetty[1], Brunda Vishishta[1], Shrinidhi Choragi[1], Karpagavalli Subramanian[1] and Koshy George[2]

[1]Department of Electronics and Communication Engineering, PES University, Bangalore, India
[2]Department of Electrical and Electronics Engineering, SRM University – AP, Guntur District, India.

November 26, 2021

Outline

Motivation

Few applications of Robot Manipulator

- **Industrial**: welding, painting, assembly, disassembly, pick and place
- **Medical**: invasive surgeries, prescription dispensing systems

Motivation

Few applications of Robot Manipulator

- **Industrial**: welding, painting, assembly, disassembly, pick and place
- **Medical**: invasive surgeries, prescription dispensing systems

Challenges:

- Complex dynamics and uncertain operating conditions
- Parameter uncertainties, frictional forces, external disturbances, measurement noise, payload variations
- Most model-based controllers require an accurate model of the system and thereby fail to compensate

Motivation

Few applications of Robot Manipulator

- **Industrial**: welding, painting, assembly, disassembly, pick and place
- **Medical**: invasive surgeries, prescription dispensing systems

Challenges:

- Complex dynamics and uncertain operating conditions
- Parameter uncertainties, frictional forces, external disturbances, measurement noise, payload variations
- Most model-based controllers require an accurate model of the system and thereby fail to compensate

Reinforcement Learning:-

- Ability to generate optimal policies with no prior knowledge of the environment
- Deep RL addresses the issues of scalability, memory and computational complexity
- Can extend to multiple degrees-of-freedom

Related Work

- **Actor-Critic**: reference tracking in a 6-DoF robotic arm [1], vibration suppression in a two-link robot manipulator [2]

Related Work

- **Actor-Critic**: reference tracking in a 6-DoF robotic arm [1], vibration suppression in a two-link robot manipulator [2]
- Reference [3] compares the performance of several parameterized function approximators to control a robot manipulator in the presence of payload variations and external disturbances

Related Work

- **Actor-Critic**: reference tracking in a 6-DoF robotic arm [1], vibration suppression in a two-link robot manipulator [2]
- Reference [3] compares the performance of several parameterized function approximators to control a robot manipulator in the presence of payload variations and external disturbances
- Disadvantage: Most of these techniques operate with discrete action spaces which provide low scalability as the number of degrees-of-freedom increase.

Related Work

- **Actor-Critic**: reference tracking in a 6-DoF robotic arm [1], vibration suppression in a two-link robot manipulator [2]
- Reference [3] compares the performance of several parameterized function approximators to control a robot manipulator in the presence of payload variations and external disturbances
- Disadvantage: Most of these techniques operate with discrete action spaces which provide low scalability as the number of degrees-of-freedom increase.
- **DDPG**: trajectory-tracking control of a SCARA and mobile robot [4], reaching task of a 6-DoF robot manipulator in 3D space [5]

Related Work

- **Actor-Critic**: reference tracking in a 6-DoF robotic arm [1], vibration suppression in a two-link robot manipulator [2]
- Reference [3] compares the performance of several parameterized function approximators to control a robot manipulator in the presence of payload variations and external disturbances
- Disadvantage: Most of these techniques operate with discrete action spaces which provide low scalability as the number of degrees-of-freedom increase.
- **DDPG**: trajectory-tracking control of a SCARA and mobile robot [4], reaching task of a 6-DoF robot manipulator in 3D space [5]
- No performance evaluation in the presence of uncertainties and disturbances

Robot Manipulator Dynamics

The dynamics of the robot manipulator is described as follows:

$$M_{11}\ddot{\theta}_1 + M_{12}\ddot{\theta}_2 + V_1 + G_1 = \tau_1 \qquad (1)$$

$$M_{12}\ddot{\theta}_1 + M_{22}\ddot{\theta}_2 + V_2 + G_2 = \tau_2 \qquad (2)$$

In a more standard form:

$$M(\theta)\ddot{\theta} + V(\theta, \dot{\theta}) + G(\theta) = \tau \qquad (3)$$

where,

$$M = \begin{pmatrix} M_{11} & M_{12} \\ M_{12} & M_{22} \end{pmatrix}, V = \begin{pmatrix} V_1 \\ V_2 \end{pmatrix}, G = \begin{pmatrix} G_1 \\ G_2 \end{pmatrix}, \tau = \begin{pmatrix} \tau_1 \\ \tau_2 \end{pmatrix}$$

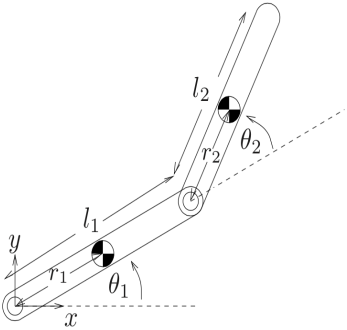


Figure: TLRM

Goal: Track a desired reference, $\theta^* = [\theta_1^*, \theta_2^*]^T$

Reinforcement Learning

- Learning takes place through agent-environment interactions
- At each time-step t, agent takes an action A_t based on the current state S_t, giving rise to a new state S_{t+1} receiving an immediate reward of r_t while following a policy π

Reinforcement Learning

- Learning takes place through agent-environment interactions
- At each time-step t, agent takes an action A_t based on the current state S_t, giving rise to a new state S_{t+1} receiving an immediate reward of r_t while following a policy π
- Cumulative reward:

$$R_\pi = r_{t+1} + \gamma r_{t+2} + \gamma^2 r_{t+3} + \cdots = \sum_{i=0}^{\infty} \gamma^i r_{t+i+1} \tag{4}$$

Reinforcement Learning

- Learning takes place through agent-environment interactions
- At each time-step t, agent takes an action A_t based on the current state S_t, giving rise to a new state S_{t+1} receiving an immediate reward of r_t while following a policy π
- Cumulative reward:

$$R_\pi = r_{t+1} + \gamma r_{t+2} + \gamma^2 r_{t+3} + \cdots = \sum_{i=0}^{\infty} \gamma^i r_{t+i+1} \tag{4}$$

- The goodness of a state is determined by its value function:

$$V_\pi(s) = \mathbb{E}_\pi\left[R_\pi | S_t = s\right] = \mathbb{E}_\pi\left[\sum_{k=0}^{\infty} \gamma^k r_{t+i+1} | S_t = s\right] \tag{5}$$

Reinforcement Learning

- Learning takes place through agent-environment interactions
- At each time-step t, agent takes an action A_t based on the current state S_t, giving rise to a new state S_{t+1} receiving an immediate reward of r_t while following a policy π
- Cumulative reward:

$$R_\pi = r_{t+1} + \gamma r_{t+2} + \gamma^2 r_{t+3} + \cdots = \sum_{i=0}^{\infty} \gamma^i r_{t+i+1} \tag{4}$$

- The goodness of a state is determined by its value function:

$$V_\pi(s) = \mathbb{E}_\pi\left[R_\pi | S_t = s\right] = \mathbb{E}_\pi\left[\sum_{k=0}^{\infty} \gamma^k r_{t+i+1} | S_t = s\right] \tag{5}$$

- Similarly, the value of taking an action a in state s while following a policy π is given by:

$$Q_\pi(s,a) = \mathbb{E}_\pi\left[R_\pi | S_t = s, A_t = a\right] = \mathbb{E}_\pi\left[\sum_{k=0}^{\infty} \gamma^k r_{t+i+1} | S_t = s, A_t = a\right] \tag{6}$$

Introduction to Deep Deterministic Policy Gradient (DDPG)

- Why DDPG?
  1. Proposed as a simple integration of deep Q-networks and actor-critic
  2. Successful in operating over continuous action spaces
  3. As the uncertainties and disturbances are deterministic in nature, a deterministic policy is sufficient to achieve our targets
- There are 4 main steps to this process:
  1. Experience replay; Finite-sized cache D
  2. Actor & critic networks: $\mu(s|\theta^{\mu})$ and $Q(s, a)$
  3. Target network updates: Added stability during training, $\theta_t = \rho\theta + (1-\rho)\theta_t$
  4. Exploration: Adding an appropriate noise signal, $\mathcal{N}$

DDPG-based Controller Specifications

- DDPG-based controller is augmented with a stabilizing PD controller
- $\Delta\tau_1$ and $\Delta\tau_2$ compensate for the uncertainties in the system

$$R = -e^T Qe \qquad (7)$$

where $e = \theta - \theta^*$ and Q is a positive definite matrix.

- Few of the DDPG parameters:

Parameter	Description	Values
(α_a, α_c)	Learning rates	$(0.001, 0.002)$
γ	Discount factor	0.99
$\Delta\tau$	Action Space	[-1,1] Nm

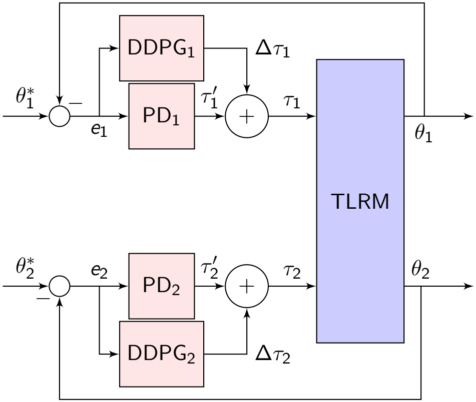


Figure: TLRM controller framework

Description of Uncertainties

- **Frictional Forces**
  1. Friction that is present between the joints of the robot manipulator can severely affect motion quality
  2. The frictional matrix can be modelled as:

$$F(\dot{\theta}) = \begin{bmatrix} \nu_1\dot{\theta}_1 + k_1 \text{sgn}(\dot{\theta}_1) \\ \nu_2\dot{\theta}_2 + k_2 \text{sgn}(\dot{\theta}_2) \end{bmatrix} \tag{8}$$

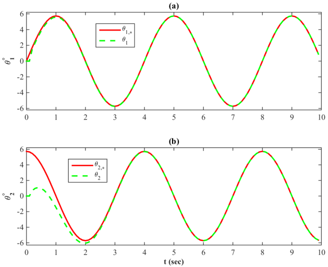
  3. The combined TLRM dynamics can be written as:

$$M(\theta)\ddot{\theta} + V(\theta, \dot{\theta}) + G(\theta) + F(\dot{\theta}) = \tau$$

- **Torque Disturbances**
  1. An external torque disturbance is introduced into the system after a duration of 5 seconds
  2. It is modelled as a sinusoidal signal with a frequency of 2π rad/sec
  3. The amplitude of the disturbance is taken to be 20% of the torque (τ) input

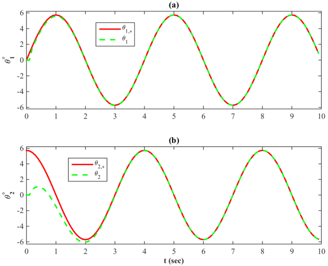
$$\tau_d = \begin{bmatrix} 0.2\tau_1 \sin 2\pi t \\ 0.2\tau_2 \sin 2\pi t \end{bmatrix} \tag{9}$$

Simulation Results with PD controller

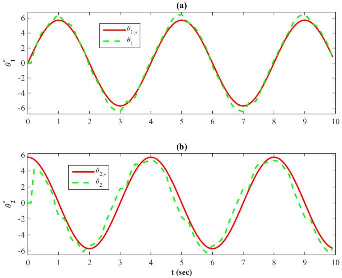


(a) No uncertainties

Simulation Results with PD controller

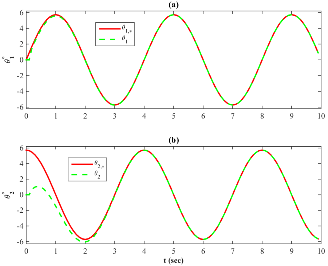


(a) No uncertainties

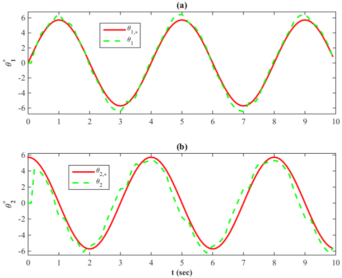


(b) With frictional forces

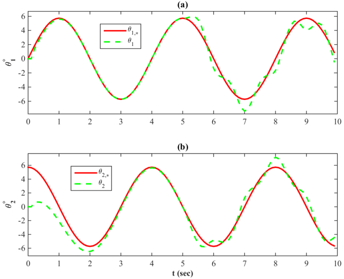
Simulation Results with PD controller



(a) No uncertainties



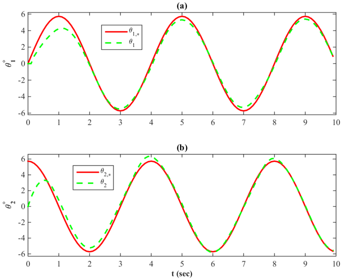
(b) With frictional forces



(c) With torque disturbances

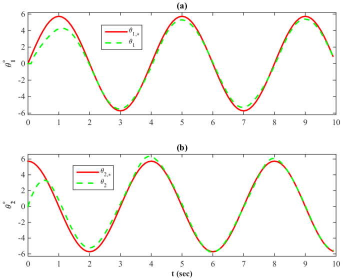
Figure: Tracking performance of PD controller

Simulation Results with DDPG controller

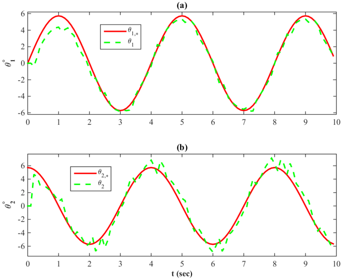


(a) No uncertainties

Simulation Results with DDPG controller

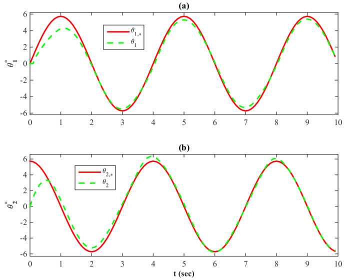


(a) No uncertainties

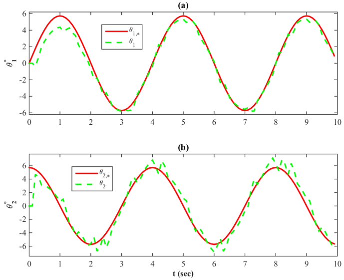


(b) With frictional forces

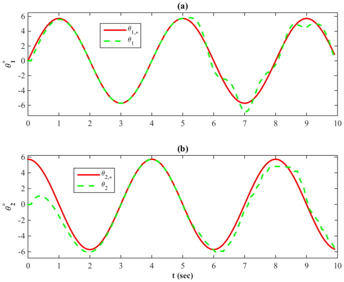
Simulation Results with DDPG controller



(a) No uncertainties



(b) With frictional forces



(c) With torque disturbances

Figure: Tracking performance of DDPG controller

Results

Table: Performance Comparison

Uncertainty	Angle	PD		DDPG	
		MSE	VAF	MSE	VAF
None	θ_1	0.399	97.493	0.581	96.726
	θ_2	1.940	89.798	0.830	94.970
Friction $F(\dot{\theta})$	θ_1	0.765	95.199	0.763	96.101
	θ_2	2.522	85.232	0.886	94.604
Torque τ_d	θ_1	0.793	95.161	0.668	95.798
	θ_2	2.744	84.963	1.855	90.635

Conclusion & Future Work

- As seen from the results, the MSE and VAF values are comparatively better for a DDPG-based controller

Conclusion & Future Work

- As seen from the results, the MSE and VAF values are comparatively better for a DDPG-based controller
- There is not requirement to constantly re-tune the parameters of the network to adapt to the changes in the system

Conclusion & Future Work

- As seen from the results, the MSE and VAF values are comparatively better for a DDPG-based controller
- There is not requirement to constantly re-tune the parameters of the network to adapt to the changes in the system
- Thereby, the DDPG-based controller better compensates for the disturbances and is hence more robust when compared to a nominal PD controller

Conclusion & Future Work

- As seen from the results, the MSE and VAF values are comparatively better for a DDPG-based controller
- There is not requirement to constantly re-tune the parameters of the network to adapt to the changes in the system
- Thereby, the DDPG-based controller better compensates for the disturbances and is hence more robust when compared to a nominal PD controller
- Future areas of focus can include:

Conclusion & Future Work

- As seen from the results, the MSE and VAF values are comparatively better for a DDPG-based controller
- There is not requirement to constantly re-tune the parameters of the network to adapt to the changes in the system
- Thereby, the DDPG-based controller better compensates for the disturbances and is hence more robust when compared to a nominal PD controller
- Future areas of focus can include:
    1. suitable formulation of the reward function to achieve better performance

Conclusion & Future Work

- As seen from the results, the MSE and VAF values are comparatively better for a DDPG-based controller
- There is not requirement to constantly re-tune the parameters of the network to adapt to the changes in the system
- Thereby, the DDPG-based controller better compensates for the disturbances and is hence more robust when compared to a nominal PD controller
- Future areas of focus can include:
  1. suitable formulation of the reward function to achieve better performance
  2. several other disturbances such as measurement noise, and wind disturbances

Conclusion & Future Work

- As seen from the results, the MSE and VAF values are comparatively better for a DDPG-based controller
- There is not requirement to constantly re-tune the parameters of the network to adapt to the changes in the system
- Thereby, the DDPG-based controller better compensates for the disturbances and is hence more robust when compared to a nominal PD controller
- Future areas of focus can include:
    1. suitable formulation of the reward function to achieve better performance
    2. several other disturbances such as measurement noise, and wind disturbances
    3. comparison with several other RL algorithms

References

[1] Y. P. Pane, S. P. Nageshrao, and R. Babuška, "Actor-critic reinforcement learning for tracking control in robotics," in *2016 IEEE 55th Conference on Decision and Control (CDC)*, 2016, pp. 5819–5826.

[2] W. He, H. Gao, C. Zhou, C. Yang, and Z. Li, "Reinforcement learning control of a flexible two-link manipulator: An experimental investigation," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 51, no. 12, pp. 7326–7336, 2021.

[3] H. Shah and M. Gopal, "Reinforcement learning control of robot manipulators in uncertain environments," in *2009 IEEE International Conference on Industrial Technology*, 2009, pp. 1–6.

[4] S. Zhang, C. Sun, Z. Feng, and G. Hu, "Trajectory-tracking control of robotic systems via deep reinforcement learning," in *2019 IEEE International Conference on Cybernetics and Intelligent Systems (CIS) and IEEE Conference on Robotics, Automation and Mechatronics (RAM)*, 2019, pp. 386–391.

[5] R. Zeng, M. Liu, J. Zhang, X. Li, Q. Zhou, and Y. Jiang, "Manipulator control method based on deep reinforcement learning," in *2020 Chinese Control And Decision Conference (CCDC)*, 2020, pp. 415–420.