Journal of Automation and Intelligence 4 (2025) 136–144

Contents lists available at ScienceDirect

Journal of Automation and Intelligence

Research article

Decentralised adaptive learning-based control of robot manipulators with unknown parameters☆

Emil Mühlbradt Sveen, Jing Zhou *, Morten Kjeld Ebbesen, Mohammad Poursina

Faculty of Engineering and Science, University of Agder, Jon Lilletuns vei 9, Grimstad, 4879, Agder, Norway

A R T I C L E I N F O

Keywords:
Adaptive dynamic programming
Optimal control
Robot manipulator
4-DoF
Unknown dynamics

A B S T R A C T

This paper studies motor joint control of a 4-degree-of-freedom (DoF) robotic manipulator using learning-based Adaptive Dynamic Programming (ADP) approach. The manipulator's dynamics are modelled as an open-loop 4-link serial kinematic chain with 4 Degrees of Freedom (DoF). Decentralised optimal controllers are designed for each link using ADP approach based on a set of cost matrices and data collected from exploration trajectories. The proposed control strategy employs an off-line, off-policy iterative approach to derive four optimal control policies, one for each joint, under exploration strategies. The objective of the controller is to control the position of each joint. Simulation and experimental results show that four independent optimal controllers are found, each under similar exploration strategies, and the proposed ADP approach successfully yields optimal linear control policies despite the presence of these complexities. The experimental results conducted on the Quanser Qarm robotic platform demonstrate the effectiveness of the proposed ADP controllers in handling significant dynamic nonlinearities, such as actuation limitations, output saturation, and filter delays.

1. Introduction

For uncertain linear systems, adaptive controller designs have been extensively studied [1,2]. Adaptive Dynamic Programming (ADP) has been employed to solve control problems for uncertain systems. Recent advances in ADP use neural networks as a tool to approximate typical nonlinear or advanced functions [3]. Learning-based ADP, as described in [4], is inspired by Reinforcement Learning(RL), and leverages some of RL's key concepts to solve the optimal controller problem. The success of ADP ranges from self-driving cars [5] and balancing robots [6], where optimal controllers were derived from an initial stabilising controller and limited exploration, demonstrating data efficiency and performance. This motivate further study of the learning-based approach of ADP for linear time-invariant systems.

Research on robotic manipulators have focused on both linear and non-linear control, despite the nonlinear dynamics typically found in these systems. Classical approaches with proportional–integral–derivative (PID) and LQR controllers still attract attention [7,8]. Linear control approaches can greatly benefit from linearisation techniques [9]. When the system is well known, model predictive control (MPC) has demonstrated high accuracy in control with different linearisation techniques [10,11]. For partially uncertain systems, adaptive approaches that consider uncertainties in the end-effector load situation still require accurate kinematic knowledge of the robot [12]. When the model of the robot serial manipulator is completely unknown, state-of-the-art reinforcement learning (RL) algorithms can learn an accurate low-level controller policy, and at the same time extend the control policy to incorporate coupling effects, and even the task decision process [13]. [14] used an ADP approach with a critic network and fuzzy logic on a 2-DoF serial manipulator with unknown dynamics to find optimal controllers. While related, significant amount of data is needed for training a neural network, thus training efficiency suffers. A solution to the data problem is to use simulators, however the sim-to-real gap can be significant [13].

For robot manipulator tracking tasks, decentralised control or independent-joint control have been investigated [15–17], where a separate actuator taking feedback only from that particular joint is responsible for the joint control. This paper is motivated by the need to enhance data efficiency in finding optimal controllers for non-linear systems. A learning-based ADP is developed for open-chain multibody systems. The proposed approach is demonstrated on a 4 DOF robot manipulator, the Quanser Arm. It demonstrates that the effectiveness and applicability of ADP for nonlinear systems can be successfully established. Related to the field of control of serial robotic manipulators, where additional controller features are needed to accurately control i.e. a tool tip position and orientation related to a work scene, this paper serves as a *first step* towards learning-based ADP for robotic control, as the study is focuses towards developing the low-level motor controllers. The method for generating joint references is not included in the scope of this work.

A common method for analysing the kinematic and dynamic framework of robotic serial manipulators is the Denavit–Hartenberg (DH) convention. If a robot is classified as a subsystem of an open multibody kinematic chain, with one degree of freedom per joint, the DH parameters can be used to describe the rotation and translation of each joint from a joint coordinate frame of reference in [18–20]. A limitation of using the DH method is the occurrence of computational singularities, and the modelling method is known to be relatively complicated [21]. In this paper, a more general approach to modelling is proposed where the robot manipulator is formulated as a multibody kinematic problem, specifically as an open-loop recursive kinematic chain. The main advantage of this approach is that the analysis is more general than the DH-convention, and the method can be expanded to systems which do not fit the DH-criteria. With proper selection of local coordinate frames, computational singularities can be avoided as long as rotation about one strategically selected axis is constrained. The multibody approach is more intuitive, as all rotational joints share the same constraint type which lead to similar equations. Therefore, the main modelling analysis in the multibody framework is to define the correct constraint vectors of each joint. For more detailed analysis, all system states, reaction forces, and moments are available through analysis of the Lagrange multipliers and the Jacobian matrix. The main contributions of this paper are as follows:

- The paper provides a comprehensive analysis of a robot manipulator modelled as a 4-DoF open-chain multibody recursive system.
- The data driven off-policy ADP algorithm is proposed to generate the decentralised optimal controllers for robot manipulator in the presence of unknown complexities, supported by experimental validations.
- The convergence of the data-driven algorithm and the stability of the robot manipulator with the proposed adaptive optimal controllers are theoretically analysed.
- Through the application on a non-linear robot manipulator from Quanser with a pragmatic approach to linearisation, while addressing controller limitations and filter delays, it is demonstrated that the feasibility and effectiveness of learning-based ADP controller.

The structure of the paper is as follows; in Section 2 a dynamic analysis of a 3D 4-link robotic manipulator with rotational joints is presented. In Section 3 an approach to using learning-based ADP for continuous time-invariant system approach is presented. The objective is to find an optimal controller for each robot joint based on a set of cost matrices and some exploration trajectories from the observable and controllable position and velocity states of each joint. In Section 4 the validity and efficiency of the presented method is demonstrated through 2 systems, on a simulated 4-DoF open kinematic chain, and on the Quanser Qarm experimental platform. The two systems have similar kinematics but different controller architectures.

2. Modelling and dynamic analysis of a 4-Dof stiff multibody open chain

The presented multibody analysis is a general representation for any 4DoF system with revolute-only joint configuration. Note that bold-face uppercase letters are used for matrices, bold-face lowercase letters for vectors, and normal-face lowercase letters are used for scalars.

Forward dynamics

In computational iterative methods, the 3D forward recursive kinematic problem can be solved for a system subjected to gravity, joint friction, and motor torque, based on [22]. A body subjected to gravity can be expressed by the Lagrangian equation in vector form:

$$\boldsymbol{M}(\theta)\ddot{\theta} + \boldsymbol{C}(\theta,\dot{\theta}) + g(\theta) = \boldsymbol{\tau} \tag{1}$$

where $\boldsymbol{M}(\theta) \in \mathbb{R}^{q\times q}$ is the mass matrix with q defined body states, $\boldsymbol{C}(\theta,\dot{\theta}) \in \mathbb{R}^{q\times q}$ is the Coriolis matrix, $g(\theta) \in \mathbb{R}^{q}$ describes the force due to gravity and joint friction, and $\boldsymbol{\tau} \in \mathbb{R}^{q}$ is, for a revolute joint, the torque applied in the joint motor. $\boldsymbol{\theta}$, $\dot{\boldsymbol{\theta}}$ and $\ddot{\boldsymbol{\theta}}$ is the positions, velocities and accelerations, respectively. Since (1) is derived from the general Euler–Lagrange equation in (2):

$$\boldsymbol{F} = \frac{\mathrm{d}}{\mathrm{d}t}\frac{\partial L}{\partial \dot{\boldsymbol{q}}} - \frac{\partial L}{\partial \boldsymbol{q}} \tag{2}$$

with $\boldsymbol{F}$ as the generalised force in the system and L as the Lagrangian function, a reformulation can be performed. From a 3D spatial frame Lagrangian derivation with a force-velocity duality of $\begin{bmatrix}\boldsymbol{f}\\ \boldsymbol{n}\end{bmatrix}$ and $\begin{bmatrix}\boldsymbol{v}\\ \boldsymbol{\omega}\end{bmatrix}$ where $\boldsymbol{f} \in \mathbb{R}^3$ is the forces and $\boldsymbol{n} \in \mathbb{R}^3$ is the moments, and $\boldsymbol{v} \in \mathbb{R}^3$ is the translational velocities and $\boldsymbol{\omega} \in \mathbb{R}^3$ is the rotational velocities, equalities for the forces and moments can be separated such that:

$$\frac{\partial L}{\partial \boldsymbol{v}} = m\boldsymbol{v} \tag{3}$$

$$\boldsymbol{f} = m\frac{\mathrm{d}}{\mathrm{d}t}\boldsymbol{v} \tag{4}$$

and

$$\frac{\partial L}{\partial \boldsymbol{\omega}} = \boldsymbol{J}\boldsymbol{\omega} \tag{5}$$

$$\boldsymbol{n} = \tilde{\boldsymbol{\omega}}\boldsymbol{J}\boldsymbol{\omega} + \boldsymbol{J}\frac{\mathrm{d}}{\mathrm{d}t}\boldsymbol{\omega} \tag{6}$$

The ˜ operator denotes the skew symmetric matrix of a vector. Then, in matrix form the spatial frame Newton-Euler equation is obtained for a body in 3D as:

$$\begin{bmatrix} m\boldsymbol{I}_{3\times3} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{J}\end{bmatrix}\begin{bmatrix}\dot{\boldsymbol{v}}\\ \dot{\boldsymbol{\omega}}\end{bmatrix} + \begin{bmatrix}\boldsymbol{0}\\ \tilde{\boldsymbol{\omega}}\boldsymbol{J}\boldsymbol{\omega}\end{bmatrix} = \begin{bmatrix}\boldsymbol{f}\\ \boldsymbol{n}\end{bmatrix} \tag{7}$$

Velocities and position coordinates can be integrated from the accelerations. From a calculation efficiency standpoint it is desirable to obtain a static inertia matrix, which can be achieved by solving the rotational accelerations and velocities in local coordinates, with the local mass moment of the inertia matrix denoted $\boldsymbol{J}'$. The rotational accelerations and moments in the local coordinates are thus: $\boldsymbol{J}'\dot{\boldsymbol{\omega}}' + \tilde{\boldsymbol{\omega}}'\boldsymbol{J}'\boldsymbol{\omega}' = \boldsymbol{n}'$. The global translational accelerations, denoted $\ddot{\boldsymbol{r}}$, of the centre of mass of each body are given by $m\ddot{\boldsymbol{r}} = \boldsymbol{f}$. Then, for an unconstrained body in 3D subjected to external and internal forces:

$$\begin{bmatrix} m\boldsymbol{I}_{3\times3} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{J}'\end{bmatrix}\begin{bmatrix}\ddot{\boldsymbol{r}}\\ \dot{\boldsymbol{\omega}}'\end{bmatrix} + \begin{bmatrix}\boldsymbol{0}\\ \tilde{\boldsymbol{\omega}}'\boldsymbol{J}'\boldsymbol{\omega}'\end{bmatrix} = \begin{bmatrix}\boldsymbol{f}\\ \boldsymbol{n}'\end{bmatrix} \tag{8}$$

For a multibody unconstrained system where (8) is defined for each i body, a system of equations written in matrix form where (8) is compressed and reformulated as:

$$\boldsymbol{M}\dot{\boldsymbol{v}} + \boldsymbol{b} = \boldsymbol{g}^{p} \tag{9}$$

Then, for a multibody system where each joint is defined as a revolute joint with constraint equations that remove 5 degrees of freedom, for each joint 3 and 2 rotational freedoms, each joint is only allowed to move about the unconstrained degree of freedom. A dynamic equation for the torque in the remaining rotational axis of freedom is subject to movement with a general joint and body-pair definition as in Fig. 1. Then, a serial kinematic chain can be defined. For each body, a fixed local coordinate system is attached to the Centre of Mass (CoG), with axis definitions as in Fig. 2. Then each set of body-pair constraints have

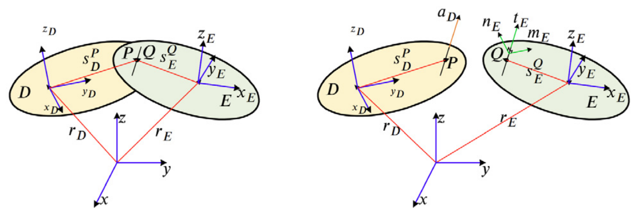

Fig. 1. Multibody parameters of a revolute joint.

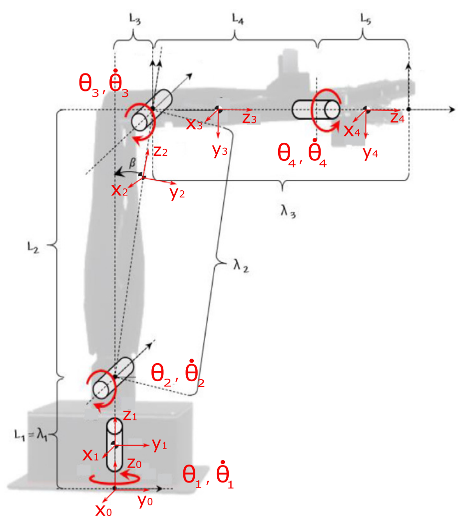

Fig. 2. Axis definitions of a 4-DoF open kinematic chain.

associated reaction forces, such that the constrained multibody system can be written as:

$$\boldsymbol{M}\dot{\boldsymbol{v}} + \boldsymbol{b} = \boldsymbol{g}^{ext} + \boldsymbol{g}^{r} \tag{10}$$

where $\boldsymbol{g}^{ext}$ is the external forces and moments acting on the system, and $\boldsymbol{g}^{r} \in \mathbb{R}^{c}$ is the reaction forces in the constraints, c is the number for constraints. $\boldsymbol{g}^{r}$ can also be expressed as the matrix product of the Jacobian matrix $\boldsymbol{\Phi}_q$, and the vector of Lagrangian multipliers $\lambda \in \mathbb{R}^{c}$; $\boldsymbol{g}^{r} = \boldsymbol{\Phi}_q^{\mathrm{T}}\lambda$. From the constraint matrix at the position level, $\boldsymbol{\Phi} = \boldsymbol{0}$ that must be satisfied along with (10). Then the second time derivative of the constraint matrix is:

$$\ddot{\boldsymbol{\Phi}} = \boldsymbol{\Phi}_q\dot{\boldsymbol{v}} + \dot{\boldsymbol{\Phi}}_q\boldsymbol{v} = \boldsymbol{0} \tag{11}$$

and defining the terms related to squared velocities $\dot{\boldsymbol{\Phi}}\boldsymbol{v} = -\boldsymbol{\gamma}$ such that:

$$\boldsymbol{\Phi}_q\dot{\boldsymbol{v}} = \boldsymbol{\gamma} \tag{12}$$

Then combining (10) with the reaction forces expressed by the Lagrangian multipliers and (12), the complete constrained multibody kinematic chain is expressed as:

$$\begin{bmatrix} \boldsymbol{M} & -\boldsymbol{\Phi}_q^{\mathrm{T}} \\ \boldsymbol{\Phi}_q & \boldsymbol{0} \end{bmatrix} \begin{bmatrix} \dot{\boldsymbol{v}} \\ \lambda \end{bmatrix} = \begin{bmatrix} \boldsymbol{g}^{ext} - \boldsymbol{b} \\ \boldsymbol{\gamma} \end{bmatrix} \tag{13}$$

where for a 4DoF system in 3D, the mass and inertia matrix $\boldsymbol{M}$ has size 24×24, the Jacobian matrix $\boldsymbol{\Phi}_q$ has size 20×24, $\dot{\boldsymbol{v}}$, $\boldsymbol{g}^{ext}$ and $\boldsymbol{b}$ of length 24, and λ and γ of length 20. The external forces, defined in global coordinates, are due to gravity g, joint viscous friction C_i, and joint motor torque input U_i.

Defining $\ddot{\boldsymbol{\Phi}}$ for the body pair in Fig. 1 with the geometric vectors defined in the local coordinates of body i, from the centre of gravity of body i to the common denoted joint position PQ. Rotations about the base joint are referenced to ground. Geometric vectors referenced to the ground can be subscripted with 0. The vectors $\boldsymbol{m}_E$, $\boldsymbol{n}_E$ and $\boldsymbol{t}_E$ are unity constraint vectors of the referenced joint. $\boldsymbol{I}$ is a identity matrix of size 3×3. $\boldsymbol{A}_i$ is the rotation matrix of body i, given as the Bryant *x-y-z* transformation matrix of the rotational coordinates of each joint, transformed from global to local representations:

$$\boldsymbol{A}_i = \begin{bmatrix} c_y c_z & -c_y s_z & s_y \\ c_x s_z + s_x s_y c_z & c_x c_z - s_x s_y s_z & -s_x c_y \\ s_x s_z - c_x s_y c_z & s_x c_z + c_x s_y c_z & c_x c_y \end{bmatrix} \tag{14}$$

where c_x is the shorthand notation for $\cos(\phi_x)$ and s_y is $\sin(\phi_y)$, etc. Then, the system can be constrained by defining a set of constraint vectors from a reference position to a joint for the position ($\boldsymbol{\Phi}$), the velocity ($\dot{\boldsymbol{\Phi}}$) and the acceleration ($\ddot{\boldsymbol{\Phi}}$). Joints in an open kinematic loop can be recursively defined in relation to each other. For a revolute joint, 3 vectors constrain the 3 translational states, and 2 vectors normal to each other constrain 2 rotational states. The constraint equations at the position level can be derived as defined in Fig. 1:

$$\boldsymbol{\Phi} = \begin{bmatrix} \boldsymbol{r}_D + \boldsymbol{A}_D \boldsymbol{s}_D^{P'} - \boldsymbol{r}_E - \boldsymbol{A}_E \boldsymbol{s}_E^{Q'} \\ (\boldsymbol{A}_D \boldsymbol{a}'_D)^{\mathrm{T}} \boldsymbol{A}_E \boldsymbol{m}'_E \\ (\boldsymbol{A}_D \boldsymbol{a}'_D)^{\mathrm{T}} \boldsymbol{A}_E \boldsymbol{n}'_E \end{bmatrix} = \boldsymbol{0} \tag{15}$$

with its first derivative equal to:

$$\dot{\boldsymbol{\Phi}} = \begin{bmatrix} \dot{\boldsymbol{r}}_D + \boldsymbol{A}_D \tilde{\boldsymbol{\omega}}'_D \boldsymbol{s}_D^{P'} - \dot{\boldsymbol{r}}_E - \boldsymbol{A}_E \tilde{\boldsymbol{\omega}}'_E \boldsymbol{s}_E^{Q'} \\ -(\boldsymbol{A}_E \boldsymbol{m}'_E)^{\mathrm{T}} \boldsymbol{A}_D \tilde{\boldsymbol{a}}'_D \boldsymbol{\omega}'_D - (\boldsymbol{A}_D \boldsymbol{a}'_D)^{\mathrm{T}} \boldsymbol{A}_E \tilde{\boldsymbol{m}}'_E \boldsymbol{\omega}'_E \\ -(\boldsymbol{A}_E \boldsymbol{n}'_E)^{\mathrm{T}} \boldsymbol{A}_D \tilde{\boldsymbol{a}}'_D \boldsymbol{\omega}'_D - (\boldsymbol{A}_D \boldsymbol{a}'_D)^{\mathrm{T}} \boldsymbol{A}_E \tilde{\boldsymbol{n}}'_E \boldsymbol{\omega}'_E \end{bmatrix} = \boldsymbol{0} \tag{16}$$

such that for the second time derivative of the position constraints we obtain $\ddot{\boldsymbol{\Theta}}$ and rearrange to define $\boldsymbol{\Theta}_q$, $\dot{\boldsymbol{v}}$ and $\boldsymbol{\gamma}$ to satisfy (12) for each body pair:

$$\begin{bmatrix} \boldsymbol{I} & -\boldsymbol{A}_D \tilde{\boldsymbol{s}}_D^{P'} & -\boldsymbol{I} & \boldsymbol{A}_E \tilde{\boldsymbol{s}}_E^{Q'} \\ \boldsymbol{0} & -(\boldsymbol{A}_E \boldsymbol{m}'_E)^{\mathrm{T}} \boldsymbol{A}_D \tilde{\boldsymbol{a}}'_D & \boldsymbol{0} & -(\boldsymbol{A}_D \boldsymbol{a}'_D)^{\mathrm{T}} \boldsymbol{A}_E \tilde{\boldsymbol{m}}'_E \\ \boldsymbol{0} & -(\boldsymbol{A}_E \boldsymbol{n}'_E)^{\mathrm{T}} \boldsymbol{A}_D \tilde{\boldsymbol{a}}'_D & \boldsymbol{0} & -(\boldsymbol{A}_D \boldsymbol{a}'_D)^{\mathrm{T}} \boldsymbol{A}_E \tilde{\boldsymbol{n}}'_E \end{bmatrix} \begin{bmatrix} \ddot{\boldsymbol{r}}_D \\ \dot{\boldsymbol{\omega}}'_D \\ \ddot{\boldsymbol{r}}_E \\ \dot{\boldsymbol{\omega}}'_E \end{bmatrix} =$$

$$\begin{bmatrix} \boldsymbol{A}_D \tilde{\boldsymbol{\omega}}'_D \tilde{\boldsymbol{s}}'^{P}_D \boldsymbol{\omega}'_D - \boldsymbol{A}_E \tilde{\boldsymbol{\omega}}'_E \tilde{\boldsymbol{s}}'^{Q}_E \boldsymbol{\omega}'_E \\ 2(\boldsymbol{A}_D \tilde{\boldsymbol{\omega}}'_D \boldsymbol{a}'_D)^{\mathrm{T}} \boldsymbol{A}_E \tilde{\boldsymbol{m}}'_E \boldsymbol{\omega}'_E + (\boldsymbol{A}_E \boldsymbol{m}'_E)^{\mathrm{T}} \boldsymbol{A}_D \tilde{\boldsymbol{\omega}}'_D \tilde{\boldsymbol{a}}'_D + (\boldsymbol{A}_D \boldsymbol{a}'_D)^{\mathrm{T}} \boldsymbol{A}_E \tilde{\boldsymbol{\omega}}'_E \tilde{\boldsymbol{m}}'_E \boldsymbol{\omega}'_E \\ 2(\boldsymbol{A}_D \tilde{\boldsymbol{\omega}}'_D \boldsymbol{a}'_D)^{\mathrm{T}} \boldsymbol{A}_E \tilde{\boldsymbol{n}}'_E \boldsymbol{\omega}'_E + (\boldsymbol{A}_E \boldsymbol{n}'_E)^{\mathrm{T}} \boldsymbol{A}_D \tilde{\boldsymbol{\omega}}'_D \tilde{\boldsymbol{a}}'_D + (\boldsymbol{A}_D \boldsymbol{a}'_D)^{\mathrm{T}} \boldsymbol{A}_E \tilde{\boldsymbol{\omega}}'_E \tilde{\boldsymbol{n}}'_E \boldsymbol{\omega}'_E \end{bmatrix} \tag{17}$$

Then from the defined matrices and vectors, the Lagrange multipliers can be solved from:

$$\lambda = (\boldsymbol{\Phi}_q \boldsymbol{M}^{-1} \boldsymbol{\Phi}_q^{\mathrm{T}})^{-1} (\boldsymbol{\gamma} - \boldsymbol{\Phi}_q \boldsymbol{M}^{-1} (\boldsymbol{g}^{ext} - \boldsymbol{b})) \tag{18}$$

The translational accelerations referenced to the global coordinate system, $\ddot{\boldsymbol{r}}_i$, and the first derivative of the rotational velocities with reference to each local coordinate system, $\dot{\boldsymbol{\omega}}_i$, can be found by

$$\begin{aligned} &\begin{bmatrix} \ddot{\boldsymbol{r}}_1 & \dot{\boldsymbol{\omega}}'_1 & \ddot{\boldsymbol{r}}_2 & \dot{\boldsymbol{\omega}}'_2 & \ddot{\boldsymbol{r}}_3 & \dot{\boldsymbol{\omega}}'_3 & \ddot{\boldsymbol{r}}_4 & \dot{\boldsymbol{\omega}}'_4 \end{bmatrix}^{\mathrm{T}} \\ &= \boldsymbol{M}^{-1} (\boldsymbol{\Phi}_q^{\mathrm{T}} \lambda + \boldsymbol{g}^{ext} - \boldsymbol{b}) \end{aligned} \tag{19}$$

Finally, angular velocities are transformed into time derivatives of the rotational coordinates with the transformation matrix $\boldsymbol{T}_i^{\omega}$, so that:

$$\dot{\theta}_i = (\boldsymbol{T}_i^{\omega})^{-1} \boldsymbol{\omega}_i \tag{20}$$

with

$$(\boldsymbol{T}^{\omega})^{-1} = \frac{1}{c_y} \begin{bmatrix} c_y & s_x s_y & -c_x s_y \\ 0 & c_x c_y & s_x c_y \\ 0 & -s_x & c_x \end{bmatrix} \tag{21}$$

The system states are first calculated as local accelerations and integrated to local velocities, then transformed to global velocities and integrated to global positions.

The described system model and the system states, $[x_i, y_i, z_i, \theta_{xi}, \theta_{yi}, \theta_{zi}]^{\mathrm{T}}$ and $[\dot{x}_i, \dot{y}_i, \dot{z}_i, \dot{\theta}_{xi}, \dot{\theta}_{yi}, \dot{\theta}_{zi}]^{\mathrm{T}}$ are unknown to the controller. To obtain closed loop control a joint sensor is modelled, so that for each joint, the position and the velocity about the unconstrained rotational joint are extracted from the model simulation, and used as an input to the controller. Then, for each decentralised joint controller i, the observable and controllable unconstrained rotational position θ_i and observable and controllable unconstrained rotational velocity $\dot{\theta}_i$, as indicated in Fig. 2, are available to the joint controller to generate the controller input K_i. The objective of the motor joint controller is to drive each joint to a reference position (see Fig. 3).

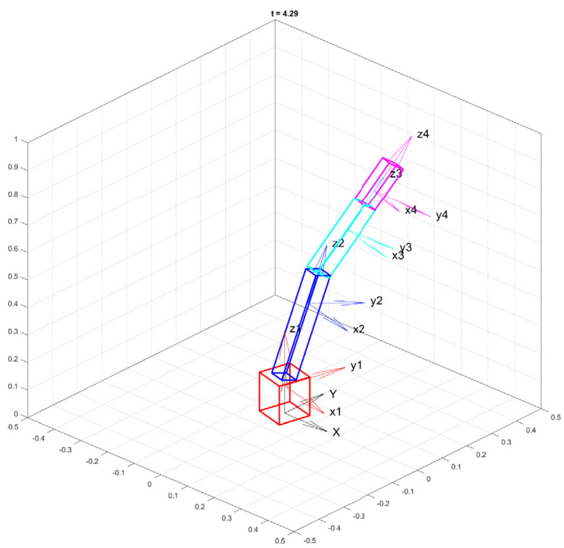

Fig. 3. Simplified animation of 4DoF 3D multibody simulation.

3. ADP based control design

In the implementation of ADP for uncertain linear time-continuous systems, an assumption of linearity of the system to be controlled is made. This assumption of linearity has previously been taken in [23, 24], even when the systems to be controlled were, in fact, significantly nonlinear. From these previous results it is therefore known that successful ADP algorithm convergence is highly dependent on bounded parameters, both in initial system conditions and in ADP algorithm parameters. This paper will not elaborate further on the boundedness of all such parameters but rather demonstrate that the off-line, off-policy ADP method presented in this chapter, based on [4,25], can be used successfully on a much larger and advanced system with the additional filters, delays, quantisation, and other challenges that come with real-time physical systems.

To match the general non-linear form in (1) with a suitable linear expression, the following assumption is made:

Assumption 1. The control problem, as a subsystem, seen from the motor controller of each joint, the subsystems can be described by some functions of only two known states, the position and velocity of the controlled joint. In state space notation where the state vector contains the known states of the system, the assumed form of each subsystem is [14]:

$$\dot{\boldsymbol{x}}_i = \begin{bmatrix} 0 & 1 \\ 0 & -\boldsymbol{M}_i^{-1}(\theta_i)\boldsymbol{C}_i(\theta_i) \end{bmatrix} \boldsymbol{x}_i + \begin{bmatrix} 0 \\ \boldsymbol{M}_i^{-1}\theta_i \end{bmatrix} [\tau_i - g(\theta_i)] \tag{22}$$

Which, if sufficiently linear, i.e. the gravity induced moments are either compensated for, or have a sufficiently small and bounded effects on the system, an approximation to a continuous linear time-invariant systems can be valid. If so, the system has the general form:

$$\dot{\boldsymbol{x}}_i = \boldsymbol{A}_i \boldsymbol{x}_i + \boldsymbol{B}_i \boldsymbol{u}_i \tag{23}$$

Remark 1. The above assumption where (23) is assumed from (1) and (13) can only be rigorously proven within a small linearised region of each joint. In practical terms, non-linearities in physical systems can often be unsuccessfully modelled, or omitted completely. By taking a worst-case approach to linearisation, the actual controller, if found by successful algorithm convergence, must be evaluated with suspicion. However, the results will be valuable for evaluating other aspects of the ADP approach, e.g. algorithm robustness, selection of exploration strategy, and failure modes.

Remark 2. To simplify notation, the subscript denoting body-joint identifier i is omitted for the rest of the presented theory. Also, according to conventional ADP practice, as the theory does not differentiate between scalar sized systems and matrix sized systems, the strict bold face notation of matrices and vectors are omitted.

The general form of each joint subsystem, expressed as continuously linear and time-invariant, and according to conventional LQR theory, iff the subsystem is stable, there exists a constant K, such that $A - BK$ is Hurwitz. An LQR-controller is formulated as the control objective;

$$u = -Kx \tag{24}$$

where the control objective is to minimise a subsystem specific performance index

$$J(x; u) = \int_0^\infty (x^{\mathrm{T}} Q x + u^{\mathrm{T}} R u) \mathrm{d}t = x^{\mathrm{T}} P + x \tag{25}$$

with solution P, given by

$$P = \int_0^\infty e^{(A-BK)^{\mathrm{T}} t} (x^{\mathrm{T}} Q x + K^{\mathrm{T}} R K) e^{(A-BK)t} \mathrm{d}t \tag{26}$$

With a valid optimal input, there is an optimal controller gain matrix, $u^* = -K^* x$, which complies with the optimal performance index.

$$J(x; u^*) = x^{\mathrm{T}} P^* x \tag{27}$$

The optimal solution to (27) is $P^* = P^{*\mathrm{T}} > 0$. This is ARE from

$$A^{\mathrm{T}} P^* + P^* A - P^* B R^{-1} B^{\mathrm{T}} P^* + Q = 0 \tag{28}$$

with a known P^*, has the solution to the optimal controller:

$$K^* = R^{-1} B^{\mathrm{T}} P^* \tag{29}$$

To solve (28), complete dynamic knowledge is required for matrices A and B. If such dynamic information is unknown or inaccurate, it is known from [4] that if an exploration input, $u' = -K_0 x + \zeta$, the solution to the performance index, P^* in (27), can be found iteratively. In this work we assume no a-priori dynamic information is available to the controller.

$$\begin{aligned} &x(t+\delta t)^{\mathrm{T}} P_j x(t+\delta t) - x(t)^{\mathrm{T}} P_j x(t) = \\ &-\int_t^{t+\delta t} [x^{\mathrm{T}}(Q + K_j^{\mathrm{T}} R K_j) x - 2(u' + K_j x)^{\mathrm{T}} R K_{j+1} x] \mathrm{d}\tau \end{aligned} \tag{30}$$

where

$$P \in \mathbb{R}^{n \times n} \to \hat{P} \in \mathbb{R}^{\frac{1}{2} n(n+1)}$$

$$\hat{P} = [p_{11}, 2p_{12}, \dots, 2p_{1n}, p_{22}, 2p_{23}, \dots, 2p_{n-1,n}, p_{nn}]^{\mathrm{T}}.$$

An initially stabilising K_j, P_j and K_{j+1}, with $P_j = P_j^{\mathrm{T}} > 0$, satisfying the ARE, then $x(t)^{\mathrm{T}} P_j x(t)$ can be found uniquely from collected data, if (23) is valid. The following equalities define an iterative method of solution in which the Kronecker product is used.

$$x^{\mathrm{T}} Q_j x = (x^{\mathrm{T}} \otimes x^{\mathrm{T}}) vec(Q_j) \tag{31}$$

and

$$\begin{aligned} (u + K_j x)^{\mathrm{T}} R K_{j+1} x = &[(x^{\mathrm{T}} \otimes x^{\mathrm{T}})(I_n \otimes K_j^{\mathrm{T}} R) + \\ &(x^{\mathrm{T}} \otimes u_0^{\mathrm{T}})(I_n \otimes R)] vec(K_{j+1}) \end{aligned} \tag{32}$$

With a positive integer l, the data matrices $\delta_{xx} \in \mathbb{R}^{l\times n^2}$, $I_{xx} \in \mathbb{R}^{l\times n^2}$ and $I_{x}u \in \mathbb{R}^{l\times mn}$, with the sampling period t are defined.

$$\begin{aligned}\delta_{xx} &= [x\otimes x|_{t_1}^{t_1+\delta t}, x\otimes x|_{t_2}^{t_2+\delta t}, \dots, x\otimes x|_{t_l}^{t_l+\delta t}]^{\mathrm{T}}\\ I_{xx} &= [\int_{t_0}^{t_1} x\otimes x\mathrm{d}\tau, \int_{t_1}^{t_2} x\otimes x\mathrm{d}\tau, \dots, \int_{t_{l-1}}^{t_l} x\otimes x\mathrm{d}\tau]^{\mathrm{T}}\\ I_{xu} &= [\int_{t_0}^{t_1} x\otimes u_0\mathrm{d}\tau, \int_{t_1}^{t_2} x\otimes u_0\mathrm{d}\tau, \dots, \int_{t_{l-1}}^{t_l} x\otimes u_0\mathrm{d}\tau]^{\mathrm{T}}\end{aligned} \tag{33}$$

For any stabilising K_j and valid (23), the linear equality must hold for each iteration step for a successful convergence.

$$\Theta_j \begin{bmatrix} \hat{P}_j \\ vecK_{j+1}\end{bmatrix} = \Xi_j \tag{34}$$

where the data matrices are constructed from the exploration phase.

$$\Theta_j = [\delta_{xx}, 2I_{xx}(I_n\otimes K_j^{\mathrm{T}}R) - 2I_{xu}(I_n\otimes R)] \tag{35}$$

$$\Xi_j = -I_{xx}vecQ_j \tag{36}$$

Full rank of Θ_j is required for a unique solution of $P_j = P_j^T$ to exist from (34). If so, for each $j = 0, 1, 2, \dots$, there must exist a sufficiently large integer $l_j > 0$ such that each sample period $t_l - t_{l-1}$ has sufficiently many data samples l_j for each iteration of j. The rank condition ensures the solution to be unique and is given as:

$$rank(\Theta_j) = \frac{n(n+1)}{2} + mn \tag{37}$$

Inspired by persistent excitation [4], the rank condition is of critical importance. But it can only be checked computationally, and for the offline approach this can only be performed after the exploration phase. Previous results [23,24], showed that for a nonlinear system, both the initial conditions and the algorithm parameters are critical for the rank condition, i.e. these parameters are bounded and convergence is not guaranteed. This presents a problem for the practicality of the implementation to nonlinear systems, as brute forcing of all solutions across a multidimensional parameter space might be required but is undesirable. However, if the rank condition holds, the off-policy ADP Algorithm can be used to find a solution of ARE, P^* that yields a near-optimal control policy K^*, evaluated by the performance index (30) and a convergence criterion ϵ, within the trajectories of the exploration phase. And assuming that (23) is valid and verifying the solution on the system, might be a sufficient and more effective alternative than brute forcing.

Off-policy ADP Algorithm

---

1. *Initialisation*
   For each joint controller K_i, Find K_{i_0} such that $A_i - B_iK_{i_0}$ is Hurwitz. Let $j_i = 0$.
2. *Online data collection*
   Apply $u_{i_0} = -K_{i_0}x_i + \zeta_i$ as the control input from $t = t_1 = 0$. Compute δ_{xx_i}, I_{xx_i} and I_{xu_i} until the rank condition (37) is satisfied. Let $j_i = 0$.
3. *Policy evaluation and improvement*
   Solve P_{i_j} with $P_{i_j} = P_{i_j}^{\mathrm{T}}$, and $K_{i_{j+1}}$.
4. *Off-policy iteration*
   Let $j_i \leftarrow j_i + 1$, and repeat (3), until $|P_{i_j} - P_{i_{j-1}}| \le \epsilon_i$ for $j_i \ge 1$, and $\epsilon_i > 0$.
5. *Exploitation*
   Use $u_i = -K_i^* x_i$ as the near-optimal control policy.

Theorem 1. *Let $K_0 \in \mathbf{R}^{m\times n}$ be any stabilising feedback gain matrix. Let P_j and K_{j+1} be a pair of matrices obtained from the Off-policy ADP Algorithm. Then, under the rank condition (37), the following properties hold:*

1. *$A - BK_j$ is Hurwitz*
2. *$P^* \le P_{j+1} \le P_j$*
3. *$\lim_{j\to\infty} K_j = K^*$, and $\lim_{j\to\infty} P_j = P^*$*

Proof. iff the linear assumption is valid, the proof in [4] can be used. The Lyapunov equation for the first iteration $j = 0$, combining (28) and (29)

$$A_0^{\mathrm{T}}P_0 + P_0A_0 + Q + K_0^{\mathrm{T}}RK_0 = 0 \tag{38}$$

with P_0 as a finite and positive definite solution, then by (26) with $\alpha = A - BK_j$

$$P_0 - P_1 = \int_o^\infty e^{\alpha_1^{\mathrm{T}}t}(K_0 - K_1)^{\mathrm{T}}R(K_0 - K_1)e^{\alpha_1 t}\mathrm{d}t \ge 0 \tag{39}$$

and

$$P_1 - P^* = \int_o^\infty e^{\alpha_1^{\mathrm{T}}t}(K_0 - K^*)^{\mathrm{T}}R(K_0 - K^*)e^{\alpha_1 t}\mathrm{d}t \ge 0 \tag{40}$$

It then follows from the above that Property 2 in Theorem 1 is valid with P^* positive definite. Then, iff P_j is decreasing for each iteration of j, i.e. converges towards a lower bound by property 3. This also implies that property 1 must hold. Therefore, the validity of the linear assumption of the nonlinear system can be verified by investigating the convergence of P_j and since (29), this is also valid for the convergence of K_j.

Remark 3. With the assumption of linearity, the algorithm above will converge to an optimal solution based on the trajectories collected during the exploration phase. Thus, for a nonlinear system the performance index (30) will evaluate the system based on Assumption 1 and the optimality criterion should only be valid in the neighbourhood of this region. For a large exploration space, the linearisation may not be strictly valid and local minima may exist.

Theorem 2. *Iff the system is sufficiently linear, with the resulting near-optimal LQR controller structure controlling each joint, the stability can be proven through the following Lyapunov. Consider the continuous, positive definite function of the joint state $V_i(x_i)$ such that $V_i(x_i) = \frac{1}{2}x_i^{\mathrm{T}}P_i^*\dot{x}_i$ with P_i^* constant and the actual joint gain B_i', then for each joint controller;*

$$\dot{V}_i(x_i) = 2x_i^{\mathrm{T}}P_i^*\dot{x}_i \tag{41}$$

$$\dot{V}_i(x_i) = 2x_i^{\mathrm{T}}P_i^*(A_ix_i - B_i'R_i^{-1}B_i^{\mathrm{T}}K_ix_i) \tag{42}$$

Where the unknown matrices A and B are used such that the ARE in (28) can be substituted:

$$\begin{aligned}\dot{V}_i(x_i) = {} & x_i^{\mathrm{T}}(-Q_i + P_i^*B_iR_i^{-1}B_i^{\mathrm{T}}P_i^*)x_i \\ & - 2x_i^{\mathrm{T}}K_iB_i'R_i^{-1}B_i^{\mathrm{T}}P_i^*x_i\end{aligned} \tag{43}$$

***Proof*.** It is required that $\dot{V}_i(x_i) > 0$ for $x \neq 0$, then

$$Q_i + P_i^*B_iR_i^{-1}B_i^{\mathrm{T}}P_i^* - 2P_i^*(B_i - B_i')R_i^{-1}B_i^{\mathrm{T}}P_i^* > 0 \tag{44}$$

Let $B' = BG$ where G is a diagonal gain matrix with unknown gains for each observable and controllable state, the above equation can be rewritten as:

$$Q - P_i^*B_i(I - 2G)R_i^{-1}B_i^{\mathrm{T}}P_i^* > 0 \tag{45}$$

This is true if, for all controllable states, $G_{i,i} > 0.5$. Therefore, the LQR joint controller provides a gain reduction of 0.5 and infinite gain amplification, for all controllable states.

4. Results

The ADP controller structure is verified in two similar 4-DoF serial kinematic systems, but with different controller structures. A simulated system with a directly computed motor torque input, and an experimental system with a PWM input. The difference in controller structure is emphasised in figure (see Fig. 4).

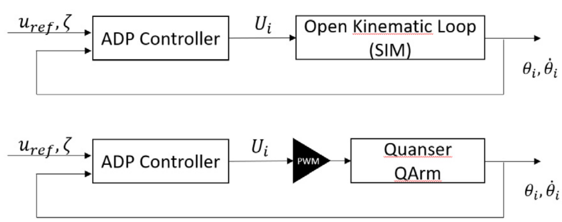

Fig. 4. Simplified controller overview with emphasis on difference between simulated and experimental system.

4.1. Simulation results

The 3D multibody analysis roughly based on the Quanser Qarm was simulated in MATLAB with both the ODE45 and a custom Newton–Raphson solver on an Intel i5-10210U CPU with 4 1.60 GHz cores. Some parameters of the QArm were known to the authors, others were estimated based on empirical experience. The values in Table 1 are not used in the controller design but used as parameters for the simulation. Attempts to fit the parameters to match the experimental results with the simulation results were not made. In particular, the simulation structure uses a direct torque input measured in [Nm], and the Quanser Qarm controller used a PWM signal, normalised between 0 and 1.

Table 1
4 DoF multibody model parameters.

Parameter	ID	Value
Lengths	L_i	$[14, 36, 5, 25, 15]$ cm
CG-Joint length	l_i	$[3.99, 10.71, 15.61, 9.98]$ cm
Body mass	m_i	$[0.791, 0.459, 0.269, 0.257]$ kgs
Body 1	J_1	$[1.556, 1.389, 1.556]10^{-3}$ kgm^2
Body 2	J_2	$[0.192, 9.61, 9.61]10^{-3}$ kgm^2
Body 3	J_3	$[2.069, 2.069, 0.268]10^{-3}$ kgm^2
Body 4	J_4	$[1.120, 1.120, 0.653]10^{-3}$ kgm^2
Friction coefficients	C_i	$[5, 3, 2, 0.2]$ Nms^{-1}

The performance objectives was set to $Q_1 = Q_2 = Q_3 = Q_4 = \begin{bmatrix} 5 & 0 \\ 0 & 0.5 \end{bmatrix}$, $R_1 = R_2 = R_3 = R_4 = [0.001]$. A convergence criterion $\epsilon = 0.001$, ODE45 evaluation step time of 0.01 s and a learning window of $l = 600$. All joint controllers successfully converged to an near-optimal controller, as can be seen in Fig. 5. The optimal controllers from the simulated system was found by the shared exploration strategy of $\zeta = 20\sin(1.5t) + 10\sin(4t)$. The joint controllers were $K_1^* = \begin{bmatrix} 3.16 & 0.11 \end{bmatrix}$, $K_2^* = \begin{bmatrix} 6.83 & 1.06 \end{bmatrix}$, $K_3^* = \begin{bmatrix} 3.32 & 0.29 \end{bmatrix}$, and $K_4^* = \begin{bmatrix} 2.58 & 0.64 \end{bmatrix}$. The step response of the learnt controllers to $\pm 0.5[rad]$ are shown in Fig. 6.

4.2. Experimental results

The ADP algorithm was implemented on the Quanser Qarm, with PWM input. The performance objectives was set to $Q_1 = Q_2 = Q_3 = Q_4 = \begin{bmatrix} 5 & 0 \\ 0 & 0.5 \end{bmatrix}$, $R_1 = R_2 = R_3 = R_4 = [0.1]$. Note that the shoulder joint ($i = 2$) and wrist joint ($i = 4$) was re-tuned to $R_2 = [0.1]$ and $R_4 = [0.75]$, due to joint chattering of the learned controller. The learning window required for each joint was $l_1 = 2100, l_2 = 2100, l_3 = 3100, l_4 = 2100$

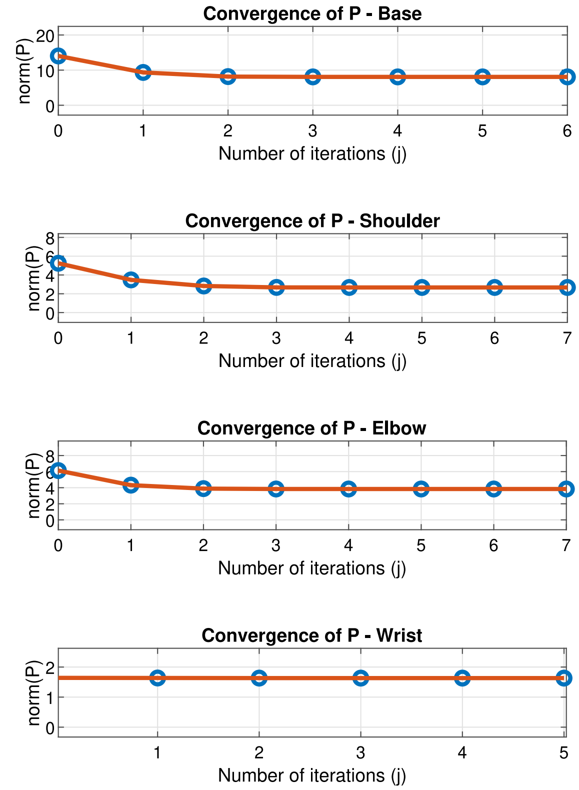

Fig. 5. Algorithm convergence of base, shoulder, elbow and wrist from simulation.

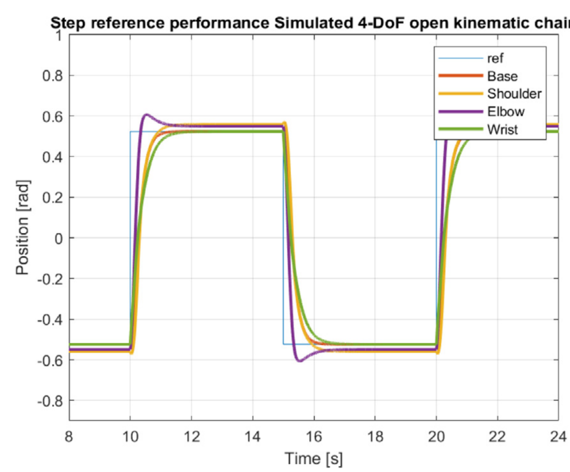

Fig. 6. Step response of simulated 4DoF kinematic chain.

with a step time of 0.01 s was significantly more than for the simulation results (see Figs. 7 and 8). The near-optimal controllers was found from a common exploration strategy of $\zeta = 0.5\sin(t)$. The learned controllers were $K_1^* = \begin{bmatrix} 2.46 & 0.50 \end{bmatrix}$, $K_2^* = \begin{bmatrix} 6.23 & 0.18 \end{bmatrix}$, $K_3^* = \begin{bmatrix} 9.48 & 0.26 \end{bmatrix}$, and $K_4^* = \begin{bmatrix} 1.18 & 0.28 \end{bmatrix}$. The step performance is shown in Table 2, evaluated as the settling time of a unity step input equally distanced around the unstable equilibrium, and the 2% steady-state error on both sides of the said equilibrium (see Fig. 9).

4.3. ADP algorithm failure modes

Failure to obtain a full rank of Θ_j will lead to no solution of the equality in (34), and a failure of the algorithm iteration will occur. This can be solved by increasing the learning window length l_j, however

Fig. 7. Quanser Qarm 4-link open kinematic chain experimental platform.

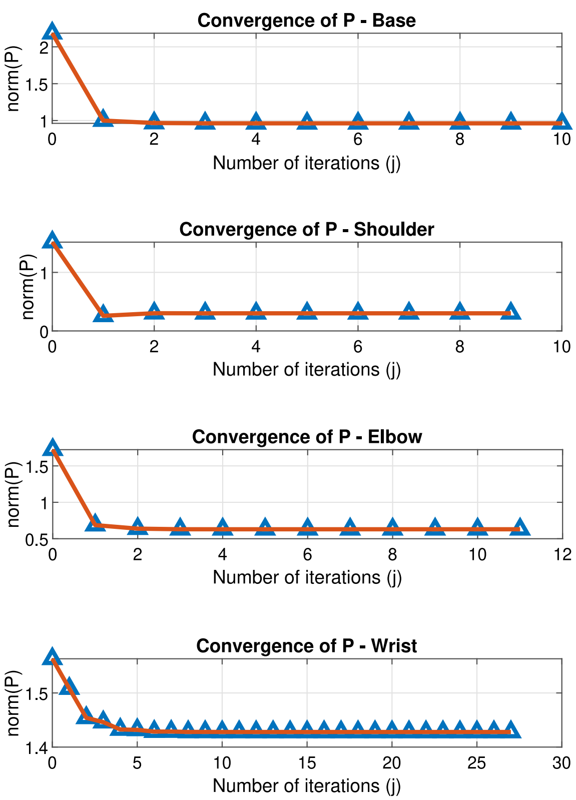

Fig. 8. Algorithm convergence of base, shoulder, elbow and wrist from Quanser Qarm.

from the results in [23], for nonlinear systems one or more parameters may be outside the bounded region. Successful convergence, but with inadmissible controller values, that is, negative controller values, was commonly found in both simulation and experimental testing.

4.4. Discussion

The proposed recursive multibody approach to modelling of the robotic manipulator was able to efficiently simulate the 4-DoF open kinematic chain. The proposed learning-based ADP controller demonstrated convergence to a near-optimal solution evaluated by Q, R and ϵ for all joints during a similar exploration strategy. Thus, the assumption of sufficient linearity must be valid, at least within the exploration trajectories, with theoretical results for stability of the learned controller, and convergence to such controller. The optimal controllers did not show any noticeable deterioration in performance outside the exploration region. Convergence was fast, only requiring 6 s of collection time, proving the efficiency of learning-based ADP. However, convergence failures were encountered during this work, which led to inapplicable controller values. Convergence was found to be especially sensitive to initial conditions and exploration noise input, often leading to failure of the rank condition in (37). Convergence failures are considered a major challenge with learning-based ADP, and is a previously known issue, that is exaggerated by significantly non-linear dynamics. The linearisation approach used in this paper is considered *worst-case*, i.e. no information about the system dynamics was available. If some knowledge about the system dynamics was known, it could be exploited in the design of e.g. a feedforward control signal, and the control problem would be more similar to previous results with linearised systems. The system gain due to PWM normalisation and quick saturation of the controller input is believed to be beneficial to performance. The main advantage of the learning-based ADP approach is to obtain the optimal LQR controller without the need to obtain the system dynamic matrix, nor the input dynamic matrix in state space form. In other aspects of controller performance, the presented work should be viewed as equivalent to traditional LQR control.

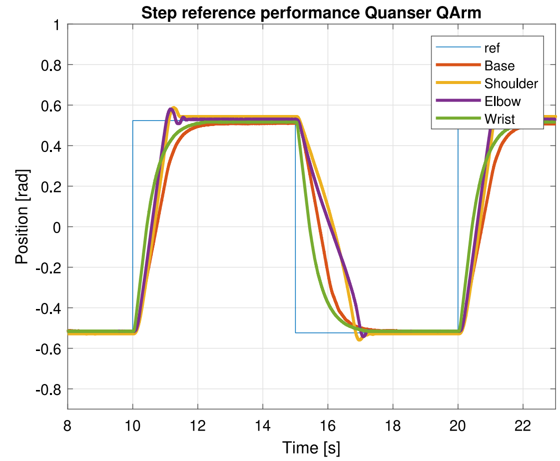

Fig. 9. Step response of joint controllers on Quanser Qarm.

Table 2
ADP controller step performance.

4DoF Simulator			
Controller, i	t_s(2%)	e_{ss}@-0.5 [rad]	e_{ss}@0.5 [rad]
1 (Base)	0.89 s	0	0
2 (Shoulder)	1.03 s	−0.033	0.033
3 (Elbow)	0.95 s	−0.024	0.024
4 (Wrist)	1.25 s	0	0
Quanser Qarm			
Controller, i	t_s	e_{ss}@-0.5 [rad]	e_{ss}@0.5 [rad]
1 (Base)	1.90 s	0.018	−0.015
2 (Shoulder)	1.38 s	0.005	0.020
3 (Elbow)	1.42 s	−0.007	0.007
4 (Wrist)	1.68 s	0.007	−0.007

5. Conclusion

This paper studies motor joint control of a 4-degree-of-freedom (DoF) robotic manipulator using learning-based Adaptive Dynamic Programming (ADP) approach. The manipulator's dynamics was modelled as an open-loop 4-link serial kinematic chain with 4 Degrees of Freedom (DoF), which is a computationally effective approach that, to the best of the author's knowledge, is not previously used in the modelling of robotic manipulators. Decentralised optimal controllers were designed for each link using the ADP approach based on a set of cost matrices and data collected from exploration trajectories. The proposed control strategy employed an off-line, off-policy iterative approach to derive four optimal control policies, one for each joint, under similar exploration strategies. The controller objective was met during both simulation and physical experiments and showed that four independent optimal controllers were found under similar exploration strategies. The proposed ADP approach successfully yields optimal linear control policies despite the presence of system complexities. The experimental results conducted on the Quanser Qarm robotic platform demonstrate the effectiveness of the proposed ADP controllers in handling significant dynamic nonlinearities, such as actuation limitations, output saturation, and filter delays. The presented controller performance, with advantages, and limitations are equivalent to obtaining an LQR controller for the system, but the results show that the learning-based ADP approach makes it possible to obtain an LQR-equivalent controller without the need to identify the system dynamics and input dynamics. As this paper focuses on the development of the low-level controller architecture, a kinematic solver based on inverse kinematic information of the system is required to generate the appropriate joint reference signals to achieve a tool tip position and orientation, relative to a global coordinate frame. A kinematic solver will also compensate for joint interconnections and coupling effects. In typical cases, the kinematic model is developed by analysis or by using available kinematic solvers. Information from such a kinematic model can be exploited by learning-based ADP. Related work also shows that a linearisation approximation of the system will give the ADP controller increased accuracy and robustness of convergence. The future work related to learning-based ADP and open kinematic chains is to find a good balance between modelling accuracy, algorithm convergence robustness, and controller accuracy. The level of *approximation* accuracy of the kinematic and dynamic parameters will be evaluated in such work. The simulation in this paper was effectively solved on a CPU with relatively low computing power, thus the modelling approach will be considered to be used as the kinematic solver on on-board real-time systems like the Nvidia Jetson or similar.

CRediT authorship contribution statement

Emil Mühlbradt Sveen: Writing – original draft. **Jing Zhou:** Writing – review & editing, Supervision, Methodology, Funding acquisition. **Morten Kjeld Ebbesen:** Writing – review & editing. **Mohammad Poursina:** Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgment

This work was supported by the DEEPCOBOT project under Grant 306640/O70 funded by the Research Council of Norway.

Data availability

No data was used for the research described in the article.

References

[1] P. Ioannou, J. Sun, Robust adaptive control: a unified approach 2 (1989) 1876–1881.
[2] G. Tao, Adaptive control design and analysis, 2003.
[3] D. Liu, X. Shan, B. Zhao, B. Luo, Q. Wei, Adaptive dynamic programming for control: A survey and recent advances, IEEE Trans. Syst. Man, Cybernetics: Syst. 51 (1) (2021) 142–160.
[4] Y. Jiang, Z.-P. Jiang, Robust adaptive dynamic programming, in: Robust Adaptive Dynamic Programming, 2016, pp. 85–111.
[5] M. Huang, M. Zhao, P. Parikh, Y. Wang, K. Ozbay, Z.-P. Jiang, Reinforcement learning for vision-based lateral control of a self-driving car, in: 2019 IEEE 15th International Conference on Control and Automation, ICCA, 2019, pp. 1126–1131.
[6] L. Cui, S. Wang, J. Zhang, D. Zhang, J. Lai, Y. Zheng, Z. Zhang, Z.-P. Jiang, Learning-based balance control of wheel-legged robots, IEEE Robot. Autom. Lett. 6 (4) (2021) 7667–7674.
[7] M. Azeez, A. Abdelhaleem, S. Elnaggar, K. Moustafa, K. Atia, Optimization of PID trajectory tracking controller for a 3-DOF robotic manipulator using enhanced artificial bee colony algorithm, Sci. Rep. 13 (2023) 11164.
[8] A. Asghar, M. Iqbal, A. Khaliq, S. Rehman, J. Iqbal, Performance comparison of structured H-infinity based looptune and lqr for a 4-DOF robotic manipulator, PLOS ONE 17 (2022) e0266728.
[9] D. Kostic, B. Jager, M. Steinbuch, R. Hensen, Modeling and identification for high-performance robot control: An RRR-robotic arm case study, IEEE Trans. Control Syst. Technol. 12 (2004) 904–919.
[10] Y. Chen, X. Luo, B. Han, Q. Luo, L. Qiao, Model predictive control with integral compensation for motion control of robot manipulator in joint and task spaces, IEEE Access 8 (2020) 107063–107075.
[11] J.S. Terry, L. Rupert, M.D. Killpack, Comparison of linearized dynamic robot manipulator models for model predictive control, in: 2017 IEEE-RAS 17th International Conference on Humanoid Robotics (Humanoids), 2017, pp. 205–212.
[12] W. Sihan, Z. Kaiqiang, H. Guido, An adaptive controller for robotic manipulators with unknown kinematics and dynamics, IFAC- Pap. 53 (2) (2020) 8796–8801, 21st IFAC World Congress.
[13] J. Ibarz, J. Tan, C. Finn, M. Kalakrishnan, P. Pastor, S. Levine, How to train your robot with deep reinforcement learning: lessons we have learned, Int. J. Robot. Res. 40 (2021) 027836492098785.
[14] D. Samrat, B. Laxmidhar, Near-optimal control of serial manipulator with unknown dynamic model, IFAC Proc. Vol. 47 (1) (2014) 1131–1137, 3rd International Conference on Advances in Control and Optimization of Dynamical Systems (2014).
[15] Su Hau Hsu, Li Chen Fu, A fully adaptive decentralized control of robot manipulators, Automatica 42 (2006) 1761–1767.
[16] H. AzizZadeh, M.B. Menhaj, H.A. Talebi, Decentralized force and motion control of multiple cooperative manipulators, Automatika 62 (2021) 98–108.
[17] X. Gao, G. Chen, Z. Huang, Z. Liang, J. Fei, F. Sun, A decentralized adaptive sliding mode control method for reconfigurable modular robots considering joint friction, in: 35th Chinese Control and Decision Conference, CCDC, 2023, pp. 3832–3837.
[18] Z.T. Han, M.H. Mohd Zaman, M.F. Ibrahim, A. M Moubark, Kinematic analysis for trajectory planning of open-source 4-dof robot arm, Int. J. Adv. Comput. Sci. Appl. 12 (6) (2021) 768–775.
[19] A. Borkar, P. Kowe, R. Pal, U. Walke, P. Ukey, R. Wasnik, A. Buchunde, V. Gupta, Kinematic Design of a Serial Robotic Manipulator (5 dof) System (Ph.D. thesis), 2021.
[20] S. Karupusamy, S. Maruthachalam, B. Veerasamy, Kinematic modeling and performance analysis of a 5-dof robot for welding applications, Machines 12 (2024) 378.
[21] Y.-B. Li, T.-J. Li, H.-F. Zhu, D. Yang, Y.-J. Chen, W.-M. Zhang, J.-M. Liu, T. Zhang, Comparative analysis of the kinematics solution based on the DH method and screw theory, Math. Probl. Eng. 2021 (2021) 1–13.
[22] P.E. Nikravesh, Computer-Aided Analysis of Mechanical Systems, Prentice-Hall, 1988.
[23] E.M. Sveen, J. Zhou, Adaptive learning based motor control of an unknown robot manipulator, in: IECON 2023- 49th Annual Conference of the IEEE Industrial Electronics Society, 2023, pp. 1–6.
[24] E.M. Sveen, J. Zhou, Applications of linear adaptive dynamic programming (ADP) to a non-linear four-bar mechanism, in: 2023 11th International Conference on Control, Mechatronics and Automation, ICCMA, 2023, pp. 177–182.
[25] Z.-P. Jiang, T. Bian, W. Gao, Learning-based control: A tutorial and some recent results, 2020.

Emil Mühlbradt Sveen is a Ph.D. -fellow at the Faculty of Engineering and Science, University of Agder, Norway. He received a Bachelor of Mechatronics from University of Agder in 2016 and a Master of Mechatronics from the University of Agder in 2018. Aside from the academic interests, he is also involved in entrepreneurial work, among being the inventor of US2024077141A1.

Jing Zhou is currently a professor in the Faculty of Engineering and Science, University of Agder, Norway, where she serves as the research director of the Priority Research Center of Mechatronics and the head of Robotics and Automation group. She received the Ph.D. degree from the School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore, in 2006. She was a Senior Research Scientist with the International Research Institute of Stavanger from 2009 to 2016 and a Postdoctoral Fellow with the Norwegian University of Science and Technology from 2007 to 2009. Dr Zhou's research interests include adaptive control, nonlinear control, non-smooth nonlinearities, network systems, cyber–physical systems, with applications in robotics, offshore mechatronics, drilling and well systems, cranes, and vessels. She is a Fellow of the Norwegian Academy of Technological Sciences (NTVA). She has served as an Associate Editor for several international journals, including IEEE Transactions on Automatic Control, IEEE Transactions on Industrial Electronics, and Systems & Control Letters.

Morten Kjeld Ebbesen is with the Department of Engineering Sciences, University of Agder, Norway, as an associate professor in the Mechatronics group. He received his M.Sc. (2003) and Ph.D. (2009) in mechanical engineering from the University of Aalborg, Denmark. His interests are dynamics, flexible multi-body systems, time-domain simulation, fluid power, and optimisation.

Mohammad Poursina is an Associate Professor in the Department of Engineering Sciences at the University of Agder. He received his Ph.D. from Rensselaer Polytechnic Institute (RPI) in 2011. He served as a visiting scholar at Sibmios NIH Center for Biomedical Computation at Stanford University and also served as a Research Specialist at RPI for two years. In 2014, he joined the Department of Aerospace and Mechanical Engineering at the University of Arizona as an Assistant Professor. In 2019, he joined the University of Agder. He is the Associate Editor of ASME Journal of Computational and Nonlinear Dynamics. He is also a member of the editorial board of Journal of Multibody System Dynamics. His main research area is the design, modelling, simulation, and control of complex multibody systems with different applications including adaptive multiscale molecular dynamics simulations, robotics, robotic rehabilitations, contact and impact, hybrid robots, and uncertainty analysis.