Near optimal tracking control of a class of non-linear systems and an experimental comparison

Farshid Asadi1, Ali Heydari2
1Mechanical and Aerospace Engineering, Rutgers University, Piscataway, NJ 08854 USA
2Mechanical Engineering, University of New Mexico, Albuquerque, NM 87131, USA
E-mail: farshidasadi47@yahoo.com

Abstract: In this study, near optimal tracking of a class of non-linear systems is addressed. Adaptive (approximate) dynamic programming (ADP) approach is used to calculate the optimal control in closed form. ADP has been widely used to resolve optimal regulation and tracking problems of non-linear control systems. Despite advances in the so called supervised and unsupervised ADP techniques for optimal tracking, they have a main drawback. That is, the optimal controller needs to be recalculated for every particular reference trajectory. The main goal of this work is to address this issue for a class of non-linear systems. Finally, this approach is applied on a Delta robot and the performance of the method is analysed experimentally.

1 Introduction

Trajectory tracking of non-linear systems is a classic problem in control theory. Optimal control, as one of the approaches to solve this problem, has attracted some efforts throughout past several decades. The interested reader can refer to [1–6] for a concise introduction to more common non-linear optimal control techniques. Since analytic solutions of non-linear optimal control problems is not available, except for simple cases, seeking approximate solutions is a common practice. Adaptive (approximate) dynamic programming (ADP) technique [2], that is approximating cost function with a neural network(NN) and learning the optimal cost function in a backward manner (dynamic programming), is one of the widely used techniques among researchers. Optimal tracking problem have been studied for both continuous time and discrete time systems, but regardless of this, the pursued solutions can be categorised into two general frameworks: linear quadratic regulator (LQR) extensions and ADP-based approaches.

In the first approach, the non-linear plant is modelled as a linear system with time-varying matrices, and then techniques from linear optimal control theory are used. For instance, in [7], a feedback linearisation is done on the non-linear plant and then, a linear optimal problem is defined for the resultant feedback linearised system. In this approach, the object function is not directly related to the physical system and may not have physical realisation. In [8, 9], state-dependent Riccati equation approach is used for input affine non-linear systems. The main drawback of this method is that the proper choice of state-dependent quasilinear form plays an important role in the algorithm [5]. Also, in [10], a general non-linear system is considered and error dynamics is estimated adaptively as a linear system. The optimal control then, is calculated based on the linear estimation.

In the ADP-based approaches, the total cost is approximated with a function approximator of appropriate form and then, this approximation is used in order to calculate optimal cost and optimal control. ADP-based approaches can be categorised into two branches based on the objective function that they used. In the first one, the objective function for tracking is defined based on the error and the transient control of the system. Optimising this cost function leads to optimality but the resulted controller is not locally asymptotically stable in general, this will be discussed later. For instance, in [11], the general non-linear system is decomposed based on its linearisation and residual terms, then the optimal control is calculated as a combination of linear and residual part. In [12], a finite horizon continuous time optimal tracking is considered and then, the optimal control is calculated by direct implementation of ADP. A discrete time version of finite horizon approximate optimal tracking can also be found in [13], in this work the controller can accept different initial conditions of the same reference trajectory dynamics. Also, in [14–16], states and reference trajectory are augmented in a new variable and the optimal problem is solved as a regulation. In these three works reinforcement learning is used to calculate the optimal control online. Moreover in [17], a discrete time optimal tracking controller is considered for a switching system and is solved by direct implementation of ADP. Another ADP-based reinforcement learning is used in [18] for tracking control of a class of discrete time non-linear systems with unknown dynamics, however in this approach it is assumed that the input transition matrix is positive definite. For problems with bounded input, Lyshevski [19] has proposed an approach by adding a non-quadratic functional to the total cost. This approach is not done with ADP for tracking problems. However, in [20] it is used along with ADP for a regulation problem.

In the second category of ADP-based approaches, the control input is decomposed into a steady state (that makes the error dynamic stationary at origin) and a transient part. Then, the control objective is defined based on the error and the transient control. This approach can be found in [21, 22]. In [22], reference trajectory and error are augmented into new states and then the ADP is used to solve the resulted augmented regulation optimal problem. In [23], an optimal control problem is defined for the transient control and then the total cost is calculated in an online manner by ADP. In this method knowledge of system dynamics is not necessary. In [24], reinforcement learning is used to solve the optimal tracking control of a non-linear system with unknown dynamics online. In these two works ([23, 24]) the optimal problem is solved for an augmented system, as in [22].

Despite advances in the mentioned works, ADP-based methods all share a common drawback. That is, the controller needs to be re-calculated for each particular reference trajectory. This issue also exists in LQR-based techniques.

This work is dedicated to solve the mentioned problem of ADP-based approaches for a class of non-linear systems. The proposed method uses the idea of control decomposition to eliminate trajectory dynamics from error dynamics, without eliminating systems dynamic matrices. The optimization is done based on using transient control in the objective function, which is called modified total cost, in here. Effects of this decomposition on
optimal and asymptotic stability of the closed loop system will be discussed, which, to the best of our knowledge, has not been done in the related literature yet. Furthermore, it will be shown that by optimising expectation of modified total cost, instead of its exact value, there is no need to know the reference trajectory in the training stage. This change will lead to the main contribution of this paper, that is a near optimal asymptotically stabilising tracking controller that does general tracking for a class of non-linear systems. Finally, it will be shown that using optimal control based on expected value of modified total cost (instead of its exact value), does not hurt asymptotic stability of the closed loop system. The proposed controller is near optimal in three aspects. First, because of the form of steady state control that is used. Second, it approximates some standard non-linear control techniques. Third, it approximates this objective function which is the core of ADP.

In what follows, first the problem is defined and the proposed method is explained. Then theoretical support for optimality, convergence, and asymptotic stability of the approach is presented. Finally, the method is implemented experimentally on a Delta parallel manipulator and its performance is shown in comparison to some standard non-linear control techniques.

2 Problem statement and resolution

Let us define a tracking problem for a non-linear system of the following form

$$\dot{x} = f(X) + g(X)u, \quad (1)$$

where $x(t) \in \mathbb{R}^{n \times 1}$ is an $m$-vector (All vectors are column vectors) of output of interest, $X = [x_1, ..., x_{n-1}, \dot{x}_{n-1}] \in \mathbb{R}^{n \times 1}$ is $n$-vector of states, and $u(t) \in \mathbb{R}^{m \times 1}$ is $m$-vector of control input. Also, note that $n = m \times p$. Furthermore, $f(\cdot) : \mathbb{R}^{n \times 1} \rightarrow \mathbb{R}^{n \times 1}$ and $g(\cdot) : \mathbb{R}^{n \times 1} \rightarrow \mathbb{R}^{m \times 1}$ are functions representing dynamics of the system. Moreover $f(\cdot)$ and $g(\cdot)$ and their Jacobians are assumed to be continuous. Also, $\dot{x}^i$ denotes the $i$th time derivative of $x$. The system is supposed to follow a particular reference trajectory (not known a priori), that is $X_d(t) = [x_d, ..., x_{d-n}, \dot{x}_{d-n}] \in D \subset \mathbb{R}^{n \times 1}$, with zero tracking error. Furthermore, assume that $X_d$ exists and is continuous. Tracking error is defined as $E[a, ..., e_{m-1}^i] \in \mathbb{R}^{m \times 1}$ where $e = x - X_d$. As mentioned in the introduction, in some of the related literature, the optimal controller is designed to minimise the following cost function

$$J = \int_0^\infty \exp(-\rho t) (E'QE + u'Ru)dt, \quad (2)$$

where $Q \in \mathbb{R}^{n \times n}$, $R \in \mathbb{R}^{m \times m}$, and $\rho \in \mathbb{R}_{>0}$ are semi-positive definite error penalising matrix, positive definite control penalising matrix, and discount factor, respectively. The error dynamics of this tracking problem can be written in the following form

$$e^{(i)} = f(E + X_d) + g(E + X_d)u - \dot{x}_d^i. \quad (3)$$

This error dynamics can be also stated in state space form as

$$E = F(E + X_d) + G(E + X_d)u - X_d, \quad (4)$$

where one has

$$F(E + X_d) = \begin{bmatrix}
0 & e + x_d \\
\vdots & \vdots \\
0 & e^{p-1} + x_{d-1}^{p-1} \\
f(E + X_d)
\end{bmatrix}$$

Note that the above dynamics is non-autonomous (The dynamic system has a direct dependence on time through $X_d$ and its time derivative) and non-stationary (the equilibrium point of the dynamic system does not lie at origin) at origin with respect to its states $E$.

Even though error dynamics and objective function in the form of (3) and (2) are commonly used, there are three disadvantages with formulating the problem in this way. First, the optimal tracking controller is not locally asymptotically stabilising in general (see Appendix 1). The reason is that the reference trajectory, generally is not an invariant set of the system dynamics, which is needed for optimal control to be asymptotically stabilising. This shows itself as a steady-state error (This steady-state error is because of the analytical construction of the controller, not from disturbance and/or uncertainties) second, the presence of discounting factor $\rho$ means that, just a limited part of horizon is important to the controller. This will lead to a higher steady-state error and worsens the effects of the first problem. Third, the resulted optimal control can only follow (Assuming that the application tolerates the first and the second mentioned problems) the reference trajectory that is solved for. This means that for new reference trajectories, the problem should be resolved. These issues have motivated some authors to use a modified objective function and to revisit components of the error dynamics by decomposing the control input into steady and transient parts. However, the third issue is not solved in any of the ADP-related literature yet, to the best of the authors’ knowledge. Furthermore, interpretation of such control decomposition with respect to optimality is not done in the referenced works, to the best of our knowledge.

In a tracking problem, the evolution of the system can be categorised in two phases: the transient and the steady state. This gives an idea of decomposing the control to a steady-state control plus a correction term, when it is possible. For a system in the form of (1), the steady-state control, that is $u_e$, can be defined to satisfy the following equation

$$\dot{x}_d^i = f(X_d) + g(E + X_d)u_e, \quad (5)$$

this form of steady-state control is used in [23] for a discrete-time system, and its main advantage over other forms in literature is that it eliminates trajectory dynamics from error dynamics of (3). A controllable plant is assumed, therefore $g(E + X_d) = g(X)$ is invertible. Then $u_e$ can be calculated as

$$u_e = g^{-1}(E + X_d)(\dot{x}_d^i - f(X_d)). \quad (6)$$

At any instance, if error equals zero, that is $E = 0$, then applying $u_e$ leads to perfect tracking. The total control is the sum of steady-state control, that is $u_e$, and a corrective term, that is $\Delta u$, so it is defined as

$$u = u_e + \Delta u. \quad (7)$$

By substituting (7) and (6) in (3), the error dynamics equation, (3) can be rewritten as

$$e^{(i)} = f(E + X_d) - f(X_d) + g(E + X_d)\Delta u. \quad (8)$$

Furthermore, this equation can be written in state space form as

$$E = F(E + X_d) - F(X_d) + G(E + X_d)\Delta u. \quad (9)$$

The above error dynamics is stationary at origin. Now, one can define modified total cost based on the corrective term $\Delta u$ instead of total control $u$ as
where optimal transient control will be calculated by optimising the above total cost. This cost function is commonly used in this category of solutions to the optimal tracking problem. Decomposing the control and redefining the total cost resolves the first mentioned problem. Assuming that the system is controllable, the above modified total cost is bounded. The reason is that $\Delta u$ vanishes as the transient phase finishes. This cost function minimises the error and the corrective control term, so it brings the system to the steady-state tracking phase asymptotically. The reason is that, by imposing the steady-state control and optimising modified total cost, the optimal problem is converted to an optimal regulation problem which is asymptotically stable (see for example [26, pp. 284–291] and taking time derivative of (11) and using the error equation for (10) it can be seen (see Section 3) that knowledge of dynamics from (9), the non-optimal HJB equation can be derived once the expectation of modified value function is calculated, it can be used to track every trajectory in the ROI. Therefore, the third issue with existing ADP-based methods is also solved for non-linear systems with the dynamics given by (1).

To calculate expected modified value function, and consequently optimal transient control, in a closed form, ADP [2] is used here. To do this, value functions in (14) and (15) are approximated with a linear (in weight) NN as

$$V_t(E) = W_t^\top \phi(E),$$

(17)

$$\bar{V}(E) = \bar{W}^\top \phi(E),$$

(18)

where $\phi(E)$ is the vector of bias functions.

The procedure for training the NN with integral value iteration can be summarised as:

(a) Initialise some random values for $X$ and $X_d$ in the ROI.
(b) Calculate errors $E$, based on values generated in step a.
(c) Initialise (15) with $\bar{V}(E) = 0$.
(d) For every $E$ and $X_d$ pair and some small constant value of $T$, repeat the following stages:

(i) Calculate $\Delta u^i_t, \Delta u^i(E, X_d)$ from (13).
(ii) Calculate $\bar{V}_{i+1}(E)$ from (15).
(e) Use values from step a to update weights from the least square method [33, pp. 302–305].
(f) Calculate $\delta = \max|W_{i+1} - W_{i}|$ and do one of the following stages:

(i) If $\delta < \text{threshold}$, terminate the iterative procedure and set $W = W_{i+1}$ and go to step g.
(ii) If $\delta \geq \text{threshold}$ then go to step d.
(g) The optimal corrective term can be calculated as

$$\Delta u^i_t = \frac{1}{2} R^{-1} G^\top W^\top \phi(E),$$

(19)

$$\Delta u^i = \frac{1}{2} R^{-1} G^\top W^\top \phi(E),$$

(20)

where $\phi(E)$ is the vector of bias functions.

in the following form and apply Bellman principle of optimality [32] as

$$\nabla V(E(t)) = \mathbb{E}_{X_d \in \text{ROI}} \left\{ \int_{t}^{t+\Delta T} \left( E'(t)Q + \Delta u^r R \Delta u \right) \right\},$$

(14)

$$\nabla \bar{V}(E(t)) = \mathbb{E}_{X_d \in \text{ROI}} \left\{ \int_{t}^{t+\Delta T} \left( E'(t)Q + \Delta u^r R \Delta u \right) \right\},$$

(15)

where $\Delta T \to 0$. This equation can be used in the so called integral value iteration to learn the expectation of modified optimal cost (in the other words, value function), from the following iterative procedure,

$$V_{i+1}(E(t)) = \mathbb{E}_{X_d \in \text{ROI}} \left\{ \int_{t}^{t+\Delta T} \left( E'(t)Q + \Delta u^r R \Delta u \right) \right\},$$

(15)

where $\Delta u^r$ is defined as

$$\Delta u^r = \frac{1}{2} R^{-1} G^\top W^\top \phi(E).$$

(16)

Note that because of using the expectation of total cost, there is no need for knowing the trajectory in training stage. This means that once the expectation of modified value function is calculated, it can be used to track every trajectory in the ROI. Therefore, the third issue with existing ADP-based methods is also solved for non-linear systems with the dynamics given by (1).

There are several ways in the literature to solve the resulted HJB equation, including PI (policy iteration) algorithm [4, 28], integral PI [29], integral VI (value iteration) [30], projection technique [31], perturbation method [31], and parametric linear programming technique [31]. Among these methods integral VI is chosen. The reason is that it gives a good understanding of underlying theory and it does not need initial admissible policy as needed in other iterative methods based on PI. One can rewrite (11)
3 Theoretical background

3.1 Optimality

In the presented approach, the total control is calculated as a combination of the steady-state control and the optimal transient control. Considering this decomposition, one may ask about the optimality of the resulted total cost with respect to the original cost function, that is (2). To make the analysis easier to grasp, we will investigate the case of a specific reference trajectory to avoid getting involved in the expected values of the total costs. Let us define a new steady-state control \( u_i \) and a discounted version of (10) as

\[
u_i = g^*(X_0|x_0^0) - f(X_0),
\]

(19)

\[W(E) = \int_0^\infty \exp(-\rho t)(E'R'E + \Delta u'R\Delta u)dr.
\]

(20)

By substituting total control based on \( u_i \), that is \( u = u_i + \Delta u \), in (2), the total cost can be rewritten as

\[J(E) = \int_0^\infty \exp(-\rho t)(E'R'E + \Delta u'R\Delta u)dr + 2\Delta u'R\mu + \mu'R\mu dr.
\]

(21)

or

\[J(E) = W(E) + \int_0^\infty \exp(-\rho t)(2\Delta u'R\mu + \mu'R\mu)dr.
\]

(22)

By minimising (21) and (22) over \( \Delta u \) and eliminating \( J'(E) \) among them one can write

\[
\min_{\Delta u} \left\{ W(E) + \int_0^\infty \exp(-\rho t)(2\Delta u'R\mu + \mu'R\mu)dr \right\} = \min_{\Delta u} \left\{ \int_0^\infty \exp(-\rho t)(E'R'E + \Delta u'R\Delta u)dr + 2\Delta u'R\mu + \mu'R\mu dr \right\}.
\]

(23)

The minimisation is on the same variable on both sides of (23), so it can be simplified as

\[
\min_{\Delta u} \left\{ W(E) \right\} = W^*(E) = \min_{\Delta u} \left\{ \int_0^\infty \exp(-\rho t)(E'R'E + \Delta u'R\Delta u)dr \right\}.
\]

(24)

The process above means that optimising \( W(E) \) is equal to optimising \( J(E) \), while imposing \( u_i \) on the system. Further, as \( \rho \to 0 \), \( W(E) \to V(E) \). This means that optimising \( V(E) \) is equivalent to optimising \( W(E) \) while imposing \( u_i \) on the system and \( \rho \to 0 \). Since imposing \( u_i \) on the error dynamics and finding optimal transient control, for a specific reference trajectory, transforms the optimal control problem to an equivalent optimal regulation problem, so it is asymptotically stable (see [25] for asymptotic stability of optimal regulation problems). Also, since \( u_i \) is the exact steady-state control, the resulted total control is optimal among asymptotically stabilising controllers, however it is not the absolute optimal control (that as discussed, see Appendix 1, is not generally asymptotically stabilising). Furthermore, because \( u_i \) (that is used in the proposed method) has a slight difference with \( u_0 \), the proposed method has a degree of sub-optimality. Therefore, the proposed method, for a specific reference trajectory, is a near optimal control among asymptotically stabilising tracking controllers.

The other point that should be investigated is about optimality of optimal transient control, that is (13), with respect to expected value of modified total cost, that is (11). In the presented approach, the optimisation problem is defined based on the expectation of modified total cost. Introducing the expectation in the equations, a legitimate question is the optimality of the selected control, that is (13). To answer this, one needs to take derivative of LHS of (12) with respect to \( \Delta u \) and solve the following equation for \( \Delta u \)

\[
\frac{\partial}{\partial \Delta u} \left\{ \mathbb{E}_{x_i \in \text{ROI}} \left\{ \nabla_E(F - F_d + G\Delta u) + \mathbb{E}[E'(Q\Delta u) + \Delta u'R\Delta u] \right\} \right\} = 0.
\]

(25)

In the above equation, since the expectation is over \( x_i \), the derivative can be exchanged with the expectation and one has

\[
\mathbb{E}_{x_i \in \text{ROI}} \left\{ \frac{\partial}{\partial \Delta u} \left\{ \nabla_E(F - F_d + G\Delta u) + \mathbb{E}[E'(Q\Delta u) + \Delta u'R\Delta u] \right\} \right\} = 0.
\]

(26)

note that if the following holds

\[
-\mathbb{E}_{x_i \in \text{ROI}} \left\{ \nabla_E(F - F_d + G\Delta u) + \mathbb{E}[E'(Q\Delta u) + \Delta u'R\Delta u] \right\} = 0,
\]

(27)

then also (26) holds, therefore the answer to the above equation, which is well known to be in the form of (13), is a solution to (25). Consequently, (13) is a minimiser to (11) and (12). Therefore, optimality of (13) with respect to the expectation of modified total cost is proved.

3.2 Convergence

Since integral value iteration of (15) and (16) is an iterative procedure, convergence of the iterations is a concern. This concern is investigated in this Section while neglecting approximation error of (17). The procedure is adopted from [30] with some changes. Let us define the following transformations, for some positive stabilising control policy, that is

\[
(\mathbb{V}^{\mu}_E(V, X_0)) = \int_t^{t+\Delta T} (E'R'E + \Delta u'R\Delta u)dr + V(E(t+\Delta T)),
\]

(28)

\[
\mathbb{V}^{\mu}_E(V) = \mathbb{E}_{x_i \in \text{ROI}} \left\{ \mathbb{V}^{\mu}_E(V, X_0) \right\}.
\]

(29)

Also let \( T(V, X_0) \) be defined as \( T^{\mu}_E(V, X_0) \), that is calculated with \( \Delta u^{\mu}_e = -\mathbf{K}G(E + X_0)V fruitful which is the minimiser of right-hand side (RHS) of (28) (and also (29), in the same way explained in Section 3.1). Furthermore, let \( \mathcal{T}(V) \) be the expected value of \( T(V, X_0) \). In this way one can write (14) and (15) in the forms of \( \mathcal{T}(V) = \mathcal{T}(\mathbb{V}) \) and \( \mathcal{T}^{\mu}(V) = \mathbb{V}(\mathcal{T}(V)) \), respectively.

For some \( V_i \) and \( V_i \), the inequality \( V_i < V_i \) holds, then \( T^{\mu}(V_i, X_0) < T^{\mu}(V, X_0) \) also holds for all \( E, X_0 \in \text{ROI} \) and \( E \neq 0 \). Therefore one can conclude that \( T^{\mu}(V) < T^{\mu}(V) \). Consequently, by this assumption, one has \( \mathcal{T}(V) \leq T^{\mu}(V) < T^{\mu}(V) \).

If \( T_i = 0 \) holds, by investigating (15) one can see that \( \mathbb{V}_i > \mathbb{V}_i \). Also, if one assumes \( V_i = \mathbb{V}_i \), then \( \mathbb{V}_i = \mathcal{T}(\mathbb{V}_i) > \mathbb{V}_i = \mathcal{T}(\mathbb{V}) \). Thus, if integral value iteration starts with \( T_0 = 0 \), then by induction one has \( \mathcal{T}_i > \mathcal{T}_i \) for all \( E, X_0 \in \text{ROI} \) and \( E \neq 0 \).

Since a controllable plant is assumed, then there exist a non-optimal (in the sense of expectation of modified total cost) stabilising control policy, that is \( \Delta u \), whose expected modified total cost, that is \( \mathbb{V}^{\mu}_E(v) \), is greater than \( \mathbb{V}_E \) and \( \mathcal{T}(\mathbb{V}) \). Also, note that one can write \( \mathcal{T}_0 = T^{\mu}_E(Z_0) \). Therefore \( \mathcal{T}_i = \mathbb{V}(\mathcal{T}_0) < \mathcal{T}_i = T^{\mu}_E(Z_0) \).
Consequently one can conclude $Z_{i+1} < Z_n$ in the same manner used in the previous paragraph.

Furthermore, since $V < Z_n$, then $V = T(V) < Z_1 = T(Z_0)$. By repeating this for $i$ times, one can conclude that $V^i < Z_n$. Since $Z_i$ is a decreasing positive sequence (elements of the sequence are positive) that is lower bounded by $V^i$, it converges to this lower-bound. Moreover, since $V_i < Z_{n-1}$ by the same reasoning $V_i < Z_n$.

Finally, assuming that $V_0 = 1$, one can combine strictly monotonic convergence of $V_i$ to $V^i$, strictly monotonic increase of $V$, and positivity of these functions to conclude that $V_i$ also converges to $V$ as $i \to \infty$ for all $E, X_i \in ROI$ and $E \neq 0$. Furthermore, one has $V_i(0) = Z_i(0) = V(0) = 0$ from their construction. As a result, convergence of the integral value iteration for the proposed method is guaranteed for all $E, X_i \in ROI$.

### 3.3 Stability

The most important aspect of a controller is its stability. We will show asymptotic stability of the proposed method through an appropriate Lyapunov function. Let us define the Hamiltonian, for any differentiable function $V_i(E): \mathbb{R}^{n+1} \to \mathbb{R}$, as

$$H_{\Delta u}(V_i) := V_i(F - F_i + G\Delta u) + E'QE + \Delta u'R\Delta u.$$  \hspace{1cm} (30)

Furthermore, let $H(V)$ be optimal value of $H_{\Delta u}(V_i)$ with respect to $\Delta u$. Moreover, let us define $Z(E)$ as the modified total cost of an admissible control (in this work, admissible controls are limited to asymptotically stabilising controllers), calculated from (10) for a specific reference trajectory. Based on Lemma 1 of [28], if the following inequality holds, for any $V_i$,

$$H(Z) \leq H(V_i),$$  \hspace{1cm} (31)

then one has

$$V_i(E) \leq Z(E).$$  \hspace{1cm} (32)

One can write optimal value of the modified total cost of a specific reference trajectory, that is calculated form (10) (here called $S'(E)$ for clarity of notation), as

$$S'(E) = V(E) + D'(E).$$  \hspace{1cm} (33)

Furthermore, since $S'$ is the solution of HJB equation, so one has

$$H(S') = S'_0(F - F_i + G\Delta u_i) + E'QE + \Delta u_i'R\Delta u_i = 0.$$  \hspace{1cm} (34)

where $\Delta u_i = \Delta u' + \Delta u_i = -\frac{1}{2}R^{-1}G(V_i + D(E))$. This equation can be rewritten as

$$D'_0(F - F_i + G(\Delta u' + \Delta u_i)) + E'QE + (\Delta u' + \Delta u_i)R(\Delta u' + \Delta u_i) = 0.$$  \hspace{1cm} (35)

For any specific reference trajectory, there could be three cases, based on $D'(E)$. First, assume that $D' = 0$. In this case $\bar{V}(E) = V(E)$. So, by substituting (33) into (34) one has

$$\bar{V}' = \bar{V}'_0(F - F_i + G\Delta u) = -E'QE - \Delta u'R\Delta u'.$$  \hspace{1cm} (36)

RHS of the above equation is negative definite (from definition of the problem). Therefore, when $D' = 0$ and the system is controlled by the proposed controller, that is $\Delta u'$ from (13), $\bar{V}'$ is negative definite.

Second, assume that $D'(E) < 0$ for $E \neq 0$. In this case, negative definiteness of the $\bar{V}'$ under the proposed controller can be proved by contradiction. Assume that $\bar{V}'$ is negative except for some $E, X_i \in ROI$ and $E \neq 0$. Therefore, one can write $\bar{V}' = \bar{V}'_0(F - F_i + G\Delta u') \geq 0$ for some $E, X_i \in ROI$ and $E \neq 0$, which leads to $\bar{H}(\bar{V}) \geq \bar{H}(\bar{V}') = 0$. In this way one can conclude that $\bar{V}' \leq \bar{V}'$ (from Lemma 1 of [28]) for some $E, X_i \in ROI$ and $E \neq 0$. This is contradictory with the assumption of $D'(E) > 0$, so one cannot have $\bar{V}' \geq 0$ for some $E, X_i \in ROI$ and $E \neq 0$. From the same reasoning, one also cannot have $\bar{V}' \leq 0$ for all $E, X_i \in ROI$ and $E \neq 0$. Therefore, $\bar{V}'$ is negative definite under the proposed controller for $D'(E) < 0$, taking into account that $\bar{V}' = 0$ for $E = 0$ from its construction.

Third, assume that $D'(E) > 0$ for $E \neq 0$. In this way, One has $H(D) > H(S') = 0$ for all $E, X_i \in ROI, E \neq 0$. The reason is that assuming $H(D') \leq H(S')$ holds for some $E, X_i \in ROI$ and $E \neq 0$, leads to $D' \geq S'$, which is contradictory to $S' = \bar{V}' + D'$ in the current case. Also, one has $H(D') = 0$ for all $E, X_i \in ROI, E = 0$, by its definition.

Note that one can write

$$D'_0(F - F_i + G\Delta u) + E'QE + \Delta u'R\Delta u = 0,$$  \hspace{1cm} (37)

By using (35), one can write

$$D'_0(F - F_i + G\Delta u) + E'QE + \Delta u'R\Delta u + \Delta u_iR\Delta u_i = 0.$$  \hspace{1cm} (38)

By substituting (37) into (38) and combining the terms one has

$$D'_0(F - F_i + G\Delta u) + E'QE + \Delta u_iR\Delta u_i + \bar{V}'_0(F - F_i + G\Delta u) = 0.$$  \hspace{1cm} (39)

furthermore, by minimising first three terms in (39), it can be written as

$$D'_0(F - F_i + G\Delta u) + E'QE + \Delta u_iR\Delta u_i + \bar{V}'_0(F - F_i + G\Delta u) \leq 0.$$  \hspace{1cm} (40)

First three terms of (40) are equal $H(D')$, by definition. By using $H(D') > 0$ for $E \neq 0$ (as per current case) in (40), one will conclude that for all $X_i, E \in ROI, E \neq 0$,

$$\bar{V}'_0(F - F_i + G\Delta u) < 0.$$  \hspace{1cm} (41)

Also one has $\bar{V}'_0(F - F_i + G\Delta u) = 0$ for $E = 0$, by definition. As a result, $\bar{V}'$ is negative definite under the control $\Delta u'$ for $D'(E) > 0$.

$\bar{V}'(E)$ is positive definite by its construction and as we proved $\bar{V}' = \bar{V}'_0(F - F_i + G\Delta u')$ is negative definite in the ROI. Therefore $\bar{V}'(E)$ is a Lyapunov function for the system under control of $\Delta u'$, that is from (13). Therefore, the closed loop system, from the proposed method, is locally asymptotically stable.

### 4 Experimental case study

To show and compare the performance of the presented approach, an experimental study is done on a developed Delta parallel robot, as depicted in Fig. 1.
Delta robot is a parallel manipulator with three transnational degrees of freedom designed by Clavel [34]. The dynamic model of this robot can be presented in the following form [35]

\[ M(w, q)\ddot{w} + C(w, q, w, \dot{q})\dot{w} + G(w, q) = J(w, q)^\top \tau, \quad (42) \]

where \( M, C, G, w, q \), and \( \tau \) are mass matrix, Coriolis matrix, gravitational vector, workspace coordinate vector, joint space coordinate vector, and motor torques, respectively. Different methods have been used to control the Delta robot [36–39]. In the present work, computed torque (CT) method, which is usually used as a benchmark in the related literature, and sliding mode control (SMC), which is suitable benchmark of robustness, are considered for comparison.

The control law based on CT can be written as [38]

\[ \tau = J^{-1}(Cw + G + M(w_d - \lambda \dot{w} - K_p \dot{w})), \quad (43) \]

where \( w_d, K_d \), and \( K_p \) are desired position, derivative gain, and proportional gain, respectively. \( \dot{w} = w - w_d \) is also position error.

The control law for SMC can also be written as [40]

\[ \tau = J^{-1}\left(\dot{C}w + G + M \left( w_d - \lambda \dot{w} - K_{sat}(\frac{S}{\varphi}) \right) \right), \quad (44) \]

where \( S = \dot{w} + \lambda \dot{w} \), \( \lambda \), \( K \), and \( \varphi \) are sliding surface, sliding surface parameter, sliding mode controller gain, and boundary layer, respectively. Also, \( sat \) denotes saturation function.

To train ADP-based controller, 500 sets of randomly generated data is used and the training is done ten times independently with least square method. Then weights of these ten trainings are averaged and used for the experiments. Based on our experience, the averaged weights present good repeatability, whereas in each individual training different weights may be achieved (even if higher number of data is used for an individual training). To make three controllers comparable, they are tuned so that they have similar rise time. Also, experiments are done with 500 Hz sampling frequency. Furthermore, no friction compensation is done in experiments. For all experiments, actuators are saturated at 5 N m. Other parameters used in the tests can be found in Appendix 2.

4.1 Results

Two scenarios are considered to compare the performance of the methods. First, the robot is supposed to draw a circle in z-plane, with \( x = 250 \cos(\pi t) \) (mm) and \( y = 250 \sin(\pi t) \) (mm). Second, the robot is supposed to go to two different locations sequentially, i.e. \([100 \ 100 \ 450]^\top \) (mm) and \([-100 \ -100 \ 600]^\top \) (mm). Moreover, to compare robustness of the controllers, both scenarios are repeated by adding a 1 kg mass as an uncertainty to the end effector. Also, all experiments started from robot’s home position at \([0 \ 0 \ 500]^\top \) (mm). Video of the tests can be found in [41].

Results related to first scenario without uncertainty are summarised in Figs. 2–5 and Table 1. Results related to second scenario without uncertainty are presented in Figs. 6–9. The difference between performance of the three methods (without the uncertain mass), as observed through Figs. 2–5 and 7–9 and Table 1, is small. However, even these small differences are considerable in the context of robotic applications, given tight tolerances and high accuracy requirements. CT does the worst, both in step response and following a circle. The performance of the proposed method is very close to that of SMC. But except for the y coordinate error of step test, SMC controller slightly does a better job.
Results of experiments with a mass added as uncertainty are shown in Figs. 10–21 and Table 2.
As it can be seen in Figs. 10–12 and 16–18 and Table 2 adding the uncertainty to the system increases steady-state errors for all three methods. While one can say the robustness of SMC is higher than other two methods, the performance of the proposed methods still remains close to that of SMC. Moreover, the CT controller falls much behind in comparison to the other two methods, as expected.

The cost comparison of the controllers is given for first scenario, with and without uncertainty in Table 3. This cost is evaluated based on considering the total cost in the objective function, similar to (2), for first five seconds of the experiments. As expected, for the case without uncertainty, the cost of the proposed approach is lower by at least 22%. In the case with uncertainty, the cost of proposed method is ~ 2% higher than that of CT method. This is logical as the superior performance of the proposed method against uncertainty, needs more control effort.

Despite advantages of the proposed optimal controller, that can be seen in experiments, like every other method it has its disadvantageous too. ADP, that is used to solve the non-linear optimal control problem in a closed form, is a numerical procedure.

### Table 1  Mean and standard deviation of steady state $e_4$, first scenario without uncertainty

|        | x     | y     | z     |
|--------|-------|-------|-------|
| CT     | 2.2119| 2.1916| 0.4873|
| ADP    | 1.6669| 1.7063| 0.2387|
| SMC    | 1.4699| 1.4020| 0.2214|

|        | x     | y     | z     |
|--------|-------|-------|-------|
| CT     | 1.1095| 1.1505| 0.2358|
| ADP    | 0.8907| 0.8686| 0.1751|
| SMC    | 0.7393| 0.7247| 0.0861|

Fig. 3 First scenario tracking error $e_x$ without uncertainty

Fig. 4 First scenario tracking error $e_y$ without uncertainty

Fig. 5 First scenario tracking error $e_z$ without uncertainty

Fig. 6 Second scenario desired trajectory

Fig. 7 Second scenario tracking error $e_x$ without uncertainty
and it can have convergence problems in practice. To be more precise, depending on the approximation error of the NN that is used, the ADP-based algorithm may not converge for all values of $Q, R$, and sampling time. Therefore, the designer should consider this in controller design stage. The other disadvantage of ADP is about choosing basis functions. Even though some works has been done about this, there is no conclusive work yet. Generally as ROI gets bigger in dimension, finding basis functions that accurately interpolate the value function, gets much harder. Consequently the convergence of the algorithm will be affected, however normalising the data might be helpful. The other issue is the so called curse of dimensionality [32] in dynamic programming. Despite mitigation of this problem by implementing ADP, and also in the proposed method through reduction of value function parameters by introducing expectation of value function (expectation of value function depends only on $E$ instead of $E$ and $X_d$), the problem still exists.
The other point to be mentioned is about chattering. This is usually considered a problem related to SMC. However, this phenomenon can also happen for the other two methods, because of control discontinuity resulted from digital implementation. In tuning all of the controllers, it was observed that there is an upper-bound on the aggressiveness of each of the controllers, that can be achieved without chattering.

5 Conclusion

In this paper, a new framework is introduced for optimal tracking problem of a class of non-linear systems. In contrast to previous works on optimal control, the presented approach can track any trajectories (of course in the ROI) after one training. Also, using expectation of total cost, number of parameters decreased. This mitigates curse of dimensionality. The presented method is then applied to a relatively complex non-linear system and its
performance is shown experimentally. The current work addressed asymptotic optimal tracking problem for non-linear systems in canonical form. Future work can focus on the same problem for

6 Acknowledgments

This research was partially supported by the United States National Science Foundation through Grant 1745212.

7 References

[1] Bryson, A.E.: ‘Applied optimal control: optimization, estimation and control’ (Routledge, 1975)
[2] Werbos, P.: ‘Adp: goals, opportunities and principles’, Handb. Learn. Approx. Dyn. Program., 2004, 2, p. 1
[3] Lewis, F.L., Vrabie, D.: ‘Reinforcement learning and adaptive dynamic programming for feedback control’, IEEE Circuits Syst. Mag., 2009, 9, (3), pp. 32–50
[4] Saridis, G.N., Lee, C.-S.G.: ‘An approximation theory of optimal control for trainable manipulators’, IEEE Trans. Syst., Man, Cybern., 1979, 9, (3), pp. 152–159
[5] Cimen, T.: ‘State-dependent riccati equation (SDRE) control: a survey’, IFAC Proc., 2008, 41, (2), pp. 3761–3775
[6] Zhang, Y., Li, S., Liao, L.: ‘Near-optimal control of nonlinear dynamical systems: a brief survey’, Ann. Rev. Control, 2019, 47, pp. 71–80
[7] Lahdhiri, T., Elmaraghi, H.A.: ‘Design of an optimal feedback linearizing-based controller for an experimental flexible-joint robot manipulator’, Optim. Control Appl. Methods, 1999, 20, (4), pp. 165–182
[8] Cimen, T., Banks, S.P.: ‘Nonlinear optimal tracking control with application to supertankers for autopilot design’, Automatica, 2004, 40, (11), pp. 1845–1863
[9] Jeyod, H.A., Ghaffari, A.: ‘Development of a novel nonlinear estimator based on state-dependent riccati equation technique for articulated vehicles’, Proc. Inst. Mech. Eng., Part K: Multi-body Dyn., 2019, 233, (3), pp. 516–530
[10] Chou, J.-H., Hsieh, C.-H., Sun, J.-H.: ‘On-line optimal tracking control of continuous-time systems’, Mechatronics, 2004, 14, (5), pp. 587–597
[11] Tang, G.-Y., Zhao, Y.-D., Zhang, B.-L.: ‘Optimal output tracking control for nonlinear systems via successive approximation approach’, Nonlinear Anal.: Theory, Methods Appl., 2007, 66, pp. 1365–1377
[12] McLain, T.W., Bailey, C.A., Beard, R.W.: ‘Synthesis and experimental testing of a nonlinear optimal tracking controller’, Proc. 1999 American Control Conf. (Cat. No. 99CH36251) IEEE, 1999, vol. 4, pp. 2847–2851
[13] Heydari, A., Balakrishnan, S.N.: ‘Fixed-final-time optimal tracking control of input-affine nonlinear systems’, Neurocomputing, 2014, 129, pp. 528–539
[14] Modares, H., Kiumarsi, B., Vanvoudakis, K.G., et al.: ‘Adaptive Sh, [infinity ] Shv tracking control of nonlinear systems using reinforcement learning’, in ‘Adaptive learning methods for nonlinear system modeling’ (Elsevier, 2018), pp. 313–333
[15] Kiumarsi, B., Lewis, F.L.: ‘Actor-critic-based optimal tracking for partially unknown nonlinear discrete-time systems’, IEEE Trans. Neural Netw. Learn. Syst., 2014, 26, (1), pp. 140–151
[16] Modares, H., Lewis, F.L.: ‘Optimal tracking control of nonlinear partially-unknown constrained-input systems using integral reinforcement learning’, Automatica, 2014, 50, (7), pp. 1780–1792
[17] Khiabani, A.G., Heydari, A.: ‘Design and implementation of an optimal switching controller for uninterruptible power-supplies inverters using adaptive dynamic programming’, IET Power Electron., 2012, 5, pp. 3068–3076
[18] Yang, Q., Jagannathan, S.: ‘Reinforcement learning controller design for affine nonlinear discrete-time systems using online approximators’, IEEE Trans. Syst., Man, Cybern., Part B (Cybern.), 2011, 41, (2), pp. 377–390
[19] Lyshchevsky, S.E.: ‘Optimal tracking control of nonlinear dynamic systems with control bounds’, Proc. 38th IEEE Conf. on Decision and Control (Cat. No. 99CH36251) IEEE, 1999, vol. 5, pp. 4810–4815
[20] Abu-Khalaf, M., Lewis, F.L.: ‘Nearly optimal control laws for nonlinear systems with saturating actuators using a neural network hjb approach’, Automatica, 2005, 41, (5), pp. 779–791
[21] Park, Y.-M., Choi, M.-S., Lee, K.Y.: ‘An optimal tracking neuro-controller for nonlinear dynamic systems’, IEEE Trans. Neural Netw. Learn. Syst., 2014, 25, (1), pp. 20–31
[22] Vamvoudakis, K.G., Lewis, F.L.: ‘Policy iteration algorithm for continuous-time systems’, Proc. Vol. 19, (54), pp. 40–48
[23] Lyshchevsky, S., Meyer, A.U.: ‘Control system analysis and design upon the lyapunov method’, Proc. of 1995 American Control Conf. (Cat. No. 95CH36251) IEEE, 1995, vol. 5, pp. 3219–3222
[24] Athans, M., Falb, P.L.: ‘Optimal control: an introduction to the theory and its applications’ (Courier Corporation, 2013)
[25] Strulovici, B., Szydlowski, M.: ‘On the smoothness of value functions and the existence of optimal strategies in diffusion models’, J. Economic Theory, 2015, 159, pp. 1016–1055
[26] Leake, R., Liu, R.-W.: ‘Contraction of suboptimal control sequences’, SIAM J. Control Optim., 1967, 5, (3), pp. 54–63
[27] Beard, R.W., Saridis, G.N., Wen, J.T.: ‘Galerkin approximations of the generalized hamilton-jacobi-bellman equation’, Automatica, 1997, 33, (12), pp. 2159–2177
[28] Bhat, T., Jiang, Z.-P.: ‘Value iteration, adaptive dynamic programming, and optimal control of nonlinear systems’, 2016 IEEE 55th Conf. on Decision and Control (CDC) IEEE, 2016, pp. 3375–3380
[29] Kompas, T., Chiu, L.: ‘A comparison of parametric approximation techniques to continuous-time stochastic dynamic programming problems’, Tech. Rep., 2010
Consider the following optimal tracking system that is defined based on exact total cost
\[ J(x_0) = \int_{0}^{\infty} \exp(-\rho t) q(x(t)^2 + u^2) dt, \]
where \( \rho > 0 \) and \( q = 1 \). Let assume that the desired trajectory is \( r(t) = 2 \). In this case if the initial condition is \( x_0 = 1 \). Then the optimal control intuitively becomes \( u^*(t) = -1 \); this can be easily verified from the standard LQT solution. Applying the optimal control, the state time history becomes \( x(t) = 1 \). Therefore, the optimal tracking control based on exact total cost is not asymptotically stabilising. This happens due to the fact that the reference trajectory is not within invariant sets of the system (this makes error dynamics non-stationary at origin), which is needed for asymptotic stability of the closed loop system under optimal control. To be more precise the optimal tracking controller based on exact total cost

8.2 Appendix 2: experimental parameters
Here, control parameters and geometrical and inertial parameters of the robot used in the experiments are given. CT parameters:
\[ K_p = 1600 \text{ proportion gain} \]
\[ K_d = 100 \text{ derivative gain} \]

Proposed controller parameters:
\[ Q = D^TD \text{ state penalising matrix} \]
where \[ D = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \]
\[ R = 0.001I_{3 \times 3} \text{ control penalising matrix} \]
\[ Q(X) = \begin{bmatrix} x_1 & x_2 & x_3 \end{bmatrix} \]
\[ \phi(x) = \begin{bmatrix} e_1 & e_2 & e_3 \end{bmatrix} \]
\[ W = \begin{bmatrix} 0.0025 & -0.1939 & 0.3330 \\ -0.2257 & 0.0026 & -0.0009 \\ 0.0008 & 0.0317 & -0.0026 \\ 0.0002 & -0.0055 & 0.0507 \\ -0.0001 & -0.0002 & -0.0002 \\ 1.8550 & 1.8911 & 1.9928 \\ 0.0012 & 0.0012 & 0.0016 \end{bmatrix}^T \]

Sliding mode parameters:
\[ K = 70 \text{ sliding mode gain} \]
\[ \lambda = 20 \text{ sliding surface parameter} \]
\[ \varphi = 0.35 \text{ boundary layer} \]

Geometrical parameters:
\[ r_b = 0.2 \text{ m fixed platform radius} \]
\[ r_a = 0.05 \text{ m moving platform radius} \]
\[ \phi_1 = \frac{\pi}{3} \text{ first leg position angle} \]
\[ \phi_2 = \frac{\pi}{3} \text{ second leg position angle} \]
\[ \phi_3 = \frac{\pi}{3} \text{ third leg position angle} \]
\[ l_0 = 0.2 \text{ m motor leg length} \]
\[ r_1 = 0.3933 \text{ motor leg COM length ratio} \]
\[ l_k = 0.52 \text{ m intermediary leg length} \]
\[ r_k = 0.5 \text{ intermediary leg COM length ratio} \]
\[ R_k = \begin{bmatrix} 0 & r_k & 0 \end{bmatrix} \]
\[ R_0 = \begin{bmatrix} 0 & r_b & 0 \end{bmatrix} \]

Inertial parameters:
\[ m_b = 1.055 \text{ kg moving platform mass} \]
\[ I_{mb} = 0.0465475 \text{ kg m}^2 \text{ motor inertia} \]
\[ m_l = 0.116 \text{ kg motor leg mass} \]
\[ m_k = 2 \times 0.05788 \text{ kg intermediary leg mass} \]
\[ l_0 = 0.3933 \text{ intermediary leg mass} \]
\[ d_k = 0.08 \text{ m intermediary leg distance from each other at every joint} \]
\[ = 5.74769459 \times 10^{-3} \text{ kg m}^2 \text{ intermediary leg pairs dominant ii} \]