Reinforcement Learning Adaptive PID Controller for an Under-actuated Robot Arm

Adel Akbarimajd1*

1Faculty of Electrical Engineering, University of Mohaghegh Ardabili, Ardabil, Iran.

Abstract: An adaptive PID controller is used to control of a two degrees of freedom under actuated manipulator. An actor-critic based reinforcement learning is employed for tuning of parameters of the adaptive PID controller. Reinforcement learning is an unsupervised scheme wherein no reference exists to which convergence of algorithm is anticipated. Thus, it is appropriate for real time applications. Controller structure and learning equations as well as update rules are provided. Simulations are performed in SIMULINK and performance of the controller is compared with NARMA-L2 controller. The results verified good performance of the controller in tracking and disturbance rejection tests.

Keywords: Robot manipulator, Under-actuated mechanism, Adaptive PID controller, Reinforcement learning

1. Introduction

Under-actuated robot manipulator is a kinematic chain wherein total degree of the freedom of the mechanism is more than actuators. Under actuated manipulators are advantageous from the minimalism viewpoint in robotics where a task is performed with less energy consuming actuations. In fact dynamic of the mechanism is exploited instead of fighting it [1][2]. Moreover, studies on under-actuated can be beneficial in building of fault tolerant mechanisms, as when some joints of a fully actuated manipulator fail, the task can be continued before need for repairing them.

Control of under-actuated manipulators is a challenging issue because of their nonlinear characteristics and the lake of global controllability. Fortunately, It was proven that these manipulators have small-time locally controllability on an open subset of their zero velocity section, which allow them to follow any path in this subset [3]. This fact makes adaptive controllers as suitable choice for under-actuated manipulators.

Among different controllers, PID controllers are the most popular ones due to their simple implementation and high reliability. Moreover, in most cases model-free methods are available for tuning of PID parameters. As a result PID controllers have been extensively used in industries. Nevertheless, in time variant systems where the controller parameters should be adjusted according to variations in system dynamics, achieving good control performance is difficult.

Designing good performance adaptive PID controllers have been a challenging issue in recent years. In adaptive PID controllers, controller parameters should be tuned according to changes in system dynamics. Different structures have been introduced for adaptive PID controller. Three main categories of such structures include conventional adaptive PID controllers [4]-[6], fuzzy adaptive PID controllers [7]-[9] and evolutionary based adaptive PID controllers [10]-[12]. Conventional adaptive controllers exhibit low performance behavior, fuzzy adaptive PID controllers require prior knowledge of the system to be adequately tuned and evolutionary based adaptive PID-controllers are not appropriate for fast dynamic systems because of their required training time.

Neural network based adaptive controllers those employ supervised learning methods (ex. [10]) can be categorized in evolutionary adaptive controllers. As mentioned earlier, training process of these controllers need a period of time to be converged, which makes these controllers unsuitable for online instant applications. Unlike supervised learning approaches, there is no reference pattern in reinforcement learning methods. Reinforcement learning, which has origin in behaviorist psychology, adopts a test and verification method where the learning agent interacts with its environment and learns from the consequence of its actions [13][14]. As there is no reference to which convergence of algorithm is anticipated, the results of reinforcement learning can be instantly utilized, hence this learning approach can be employed in online and real time applications.

*Corresponding author: akbarimajd@uma.ac.ir
2015 UTHM Publisher. All right reserved.
penerbit.uthm.edu.my/ojs/index.php/ijie
2. Relate works to Control of Under-actuated Manipulators

There are lots of works reported in the framework of control of under-actuated manipulators. Arai and his colleagues [21][22] presented a control strategy based on holding of brakes on the passive joint in a 3-DoF under-actuated manipulator. They afterwards proposed a methodology to stabilize the trajectories of passive joint [23]. In [24] a measure of the dynamic coupling between the active and the passive joints was exploited as a cost function of an optimal control strategy that was applied to control of under-actuated manipulator. For a gravity-assisted under-actuated manipulator, a nonlinear closed loop control law that is guaranteed to be stable in positioning one unactuated joint at a time was presented in [25] where a Lyapunov function is introduced to prove the convergence of that control scheme. A robust adaptive control scheme for an under-actuated free-flying space robot is devised in [26]. In [27] a feedback linearization decoupling dynamic control scheme for one-passive joint under-actuated manipulators is proposed. Fuzzy sliding mode control was employed in [28] to control a 3R under-actuated manipulator. A motion planning method for a 3R under-actuated manipulator was presented by Lynch et al [29] and subsequently kinematical controllability of under-actuated systems was illustrated [30]. A controller was designed for a class of under-actuated manipulators to render the closed-loop equilibrium at the origin globally attractive [31]. De Luca and his colleagues used nilpotent approximations to control an under-actuated 2R manipulator [32][33] and performed a simple test to show that a planar 2R manipulator does not satisfy STLC conditions; therefore it will show spinning motion while steering from a certain configuration [34]. A point-to-point control method for a 2R planar under-actuated manipulator was introduced in [35] where passive link is firstly moved into its desired position, then the second link is moved into its desired position keeping the passive link at rest. A fuzzy controller for an under-actuated manipulator was developed in [36] whose member functions are optimized using genetic algorithm. Xin et. al. devised an energy based swing-up controller that uses a new Lyapunov function based on that transformation for n-link revolute planar robot with any one of the joint being a passive joint [37]. Akbarimajd and Kia have proposed neural network based nonlinear autoregressive moving average controller to stabilize a 2-DoF passive manipulator [38].

As review of above literature shows, different approaches have been proposed in order to control passive manipulators. However, as our proposed method is a neural network based intelligent and model-free method, to evaluate performance of our method, it would be fair to compare it with a similar one. Among recent works in this area [38] is also neural network based intelligent and model-free method. Therefore we will compare the results of the controller with that work. Details will be provided in next sections.
SAM is used to generate real PID coefficients $K(t)$ based on prior PID parameters $K'(t)$ suggested by the actor and estimated signal $V(t)$. The critic receives state and immediate external reinforcement signal (say immediate reward) from the environment and produces error signal $\delta_{st}(t)$ and estimation value function $\tilde{V}(t)$. $\delta_{st}(t)$ is directly prepared for the actor and the critic and it is behaved as a basis of updating parameters of the actor and the critic. $V(t)$ is sent to SAM and is employed to modify output of the vector.

Effect of the error signal and its variations on performance of control should be simultaneously considered in designing of external reinforcement signal $r(t)$. Thus, $r(t)$ is defined as:

$$r(t) = \alpha r_e(t) + \beta r_{ec}(t)$$

(1)

where $\alpha$ and $\beta$ are weighting factors and:

$$r_e(t) = \begin{cases} 0 & |e(t)| < \varepsilon \\ -0.5 & \text{otherwise} \end{cases}$$

(2)

$$r_{ec}(t) = \begin{cases} 0 & |e(t)| < |e(t-1)| \\ -0.5 & \text{otherwise} \end{cases}$$

(3)

and $\varepsilon$ determines tolerable error band.

3.2 Actor critic learning based on RBF

RBF is a multilayered feed forward neural network. Structure of RBF is shown in Fig. 2. A RBF network is employed for implementation of learning processes of value function of the critic and the policy function of the actor. Layers of the network and their role is described as the sequel. In input layer each neuron is a system state variable $x_i$ and state vector $x(t) \in R^3$ is directly supplied to the next layer known as hidden layer. In hidden layer, the kernel function of each neuron is selected to be Gaussian function where output of $j^{th}$ neuron is given as:
\[ \Phi_j(t) = \exp \left( -\frac{\|x(t) - \mu_j(t)\|^2}{2\sigma_j^2(t)} \right), \quad j = 1, \ldots, h \quad (4) \]

where \( \mu_j \) is center vector and \( \sigma_j \) is width scalar of \( j \)th neuron and \( h \) is the size of the hidden layer. Finally output layer is composed of two parts including actor part and critic part. \( m \)th output of the actor part can be calculated as:

\[ K'_m(t) = \sum_{j=1}^{h} w_{mj}(t)\Phi_j(t), \quad m = 1, 2, 3 \quad (5) \]

Then real PID parameters are obtained as:

\[ K(t) = K'(t) + n_k(0, \sigma_V(t)) \quad (6) \]

where \( n_k \) is Gaussian noise with \( \sigma_V(t) = \frac{1}{1 + e^{2V(t)}} \).

When \( V(t) \) is large \( n_k \) is small and vice versa and this provides a good tradeoff between exploration and exploitation.

In actor-critic learning the actor learns the policy function and the critic learns the value function using TD-error \( \delta_{TD}(t) \) which by itself is calculated as:

\[ \delta_{TD}(t) = r(t) + \gamma V(t) - V(t-1) \quad (7) \]

where \( 0 < \gamma < 1 \) is discount factor. Performance index of the learning system is defined as:

\[ E(t) = \frac{1}{2} \delta_{TD}^2(t) \quad (8) \]

Weights of network \( w_{mj}(t) \), \( v_j(t) \) and \( \mu_j(t) \) are updated to minimize above index and through a gradient descent method and chain rule (for details see [39]).

### 3.3 Controller design

The Based on discussions of previous sections, stages of designing of adaptive PID controller can be illustrated as the following:

**Step 1.** Arbitrarily set initial values for parameters of the learning system including \( \alpha, \beta, \varepsilon, \gamma, \alpha_d \) and network parameters including \( a_c, \eta_c, \eta_d, \sigma_j(0), w_{mj}(0), v_j(0), \mu_j(0) \)

**Step 2.** Read real output \( y(t) \) and calculate \( e(t), \Delta e(t), \Delta \dot{e}(t) \)

**Step 3.** Receive immediate reward \( r(t) \)

**Step 4.** Obtain outcome of actor \( K'(t) \) and value function of critic \( V(t) \).

**Step 5.** Calculate real parameters of PID controller \( K(t) \) and accordingly find control signal \( u(t) \)

**Step 6.** Apply \( u(t) \) to the system and get output \( y(t+1) \) and \( r(t+1) \) for next time step.

**Step 7.** Calculate \( K'(t+1) \) and \( V(t+1) \)

**Step 8.** Calculate TD error \( \delta_{TD}(t) \)

**Step 9.** Update weights of the actor and the critic

**Step 10.** Update weights of RBF network.

**Step 11.** If the final time is not achieved, go to Step 2.

### 4. NARMA-L2 Controller

In this section we briefly introduce NARMA-L2 controller that have been used in [38] to control passive manipulator of Fig. 1. NARMA-L2 is one of the most appropriate architectures for prediction and control of time variant nonlinear systems. This control technique is based on input output linearization. Block diagram of NARMA-L2 controller is shown in Fig. 4. There are two basic steps in NARMA-L2 including identification step and controller design step.

In identification step, the following nonlinear autoregressive moving average model is adopted for the system:

**Plant**

**Model**

**Controller**

\[ f(x(k)) \]

\[ g(x(k)) \]

\[ \hat{f}(ANN 1) \]

\[ \hat{g}(ANN 2) \]

**Fig. 4. NARMA-L2 Controller**
\[
\begin{align*}
\dot{y}(k + d) &= \hat{f}(y(k), y(k-1), \ldots, y(k-n+1), \\
u(k-1), \ldots, u(k-m+1) \\
+ \hat{g}(y(k), y(k-1), \ldots, y(k-n+1), \\
u(k-1), \ldots, u(k-m+1))u(k)
\end{align*}
\]

where \(u(k)\) and \(y(k)\) are the system input and output respectively and \(d\) is the relative degree. The positive integers \(m\) and \(n\) are respectively the number of measured delayed values of inputs and outputs. \(\hat{f}\) and \(\hat{g}\) are approximated by two MLP neural networks (see Fig. 4).

In controller design step, using (9) the control rule is given by:

\[
u(k) = \frac{A(k)}{B(k)} \tag{10}
\]

where

\[
\begin{align*}
A'(k) &= y_r(k + d) - \hat{f}(y(k), \ldots, y(k-n+1), \\
u(k), \ldots, u(k-m+1)) \\
B'(k) &= \hat{g}(y(k), \ldots, y(k-n+1), \\
u(k), \ldots, u(k-m+1))
\end{align*}
\]

The control rule (10) is not realizable since input computation of \(u(k)\) requires the output signal \(y(k)\). A more practical form can be represented as the following

\[
u(k + 1) = \frac{A'(k)}{B'(k)} \tag{11}
\]

where

\[
\begin{align*}
A(k) &= y_r(k + d) - \hat{f}(y(k), \ldots, y(k-n+1), \\
u(k-1), \ldots, u(k-m+1)) \\
B(k) &= \hat{g}(y(k), \ldots, y(k-n+1), \\
u(k-1), \ldots, u(k-m+1))
\end{align*}
\]

Rule (11) is realizable for \(d>1\). For more details about this controller see for example [38].

5. Simulation Results and Discussion

In order to design proposed adaptive PID controller for under-actuated manipulator of Fig. 1, control signal \(u\) and system output \(y\) should be determined. For this manipulator, control signal would the torque applied to the base joint. Output signal should be selected according to control goal. Namely the output signal could be \(Y_{CoG}\) or \(X_{CoG}\) of the second link or a function of these coordinates. Design of a regulator for an appropriate selection of output function, renders CoG of the passive link stay in a specific area.

A model of 2R planar under-actuated manipulator was constructed using SimMechanics in SIMULINK. Links are similar with the parameters masses \(m_1=m_2=2\) kg, moments of inertia \(I_1=I_2=0.01\) N.m. and length \(l_1=l_2=0.2\) m. The output is selected as \(y\) coordinate of the CoG of the second link, i.e. \(y=Y_{CoG}\). Input is torque applied to the active joint i.e. \(u=T_1\). We also assumed that the joints are frictionless.

Using abovementioned input-output pair of signals both the proposed controller and NARMA-L2 controller are designed. The results are devised and compared at the sequel.

5.1 Tracking test

In tracking simulation, the manipulator is initialized at fully extended configuration which means joint angles are \(\theta_1=0\) and \(\theta_2=0\). In this configuration \(Y_{CoG}=0.0\). We applied a square wave reference trajectory for \(Y_{CoG}\) as it is shown in Fig. 4. The reference signal is 0.5m for first 1 second then it switches down to 0 at \(t=1\) sec. The system is simulated with both the proposed controller and NARMA-L2 controller of [38]. Fig 5 shows snapshots of simulations with the proposed controller. Fig. 6 shows output of the system \((Y_{CoG})\) with the proposed controller and NARMA-L2 of [38]. With both controllers the system can track the reference. However, the performance of the proposed controller is better that the controller of [38] in terms of transient response. The evidence is that transient time corresponding to the proposed controller and controller of [38] are \(T_s=2.7\) sec and \(T_s=5\) sec respectively. Moreover, the response of the proposed controller has no overshoot where the overshoot of NARMA-L2 is about 11%. It is noteworthy that these values are calculated after \(t=1\) sec after learning time of the controller. In the first second the controller learns the plant and its performance is not very good (however it is still better that NARMA-L2) which is usual event in adaptive controllers.

5.2 Disturbance rejection test

In this set of simulations, we also assumed that the robot is in the fully extended configuration. Simulation starts and the robot remains in this state until \(t=1\) sec. At time \(t=1\) sec a pulse disturbance with 0.15 sec in time duration and 0.25 N.m in amplitude – as shown in Fig. 7 – is added to the control signal. In fact, a disturbance torque is inserted to the active joint.
As a result the passive joint deviates from its initial position but the proposed controller and controller of [38] both restore Y\textsubscript{CoG} back to zero. Over again, time response of the proposed controller is better than NARMA-L2. In next simulation we increased the magnitude of the disturbance to 0.35N.m. From Fig. 8 it is evident that the system with NARMA-L2 controller has become unstable. The proposed controller can reject disturbances with magnitudes less 0.35N.m while this limit for controller of [38] is 0.3N.m. Actually, the proposed controller is better disturbance rejection performance than that of [38] in terms of both response time and disturbance tolerable limit.
Fig. 7. Disturbance simulation of the proposed Adaptive-PID controller and NARMA-L2 controller (top) by a disturbance with magnitude 0.25N.m (bottom). Both systems can restore the response.

Fig. 8. Disturbance simulation of the proposed Adaptive-PID controller and NARMA-L2 controller (top) by a disturbance with magnitude 0.35N.m (bottom). NARMA-L2 is unstable.

Above simulation results verify that the proposed adaptive PID controller has stabilized the mechanism and it is robust to external disturbances and its performance is better than NARMA-L2 controller.

6. Summary

An adaptive PID controller tuned by an actor-critic reinforcement learning was successfully employed in a under actuated manipulator. It was illustrated that the controller has good performance in one coordinate tracking. The controller could resist against pulse disturbances. Simulations in MATLAB/SIMULNK and comparisons with NARMA-L2 controller verified these arguments.

By extending the idea for multivariable controllers, the controller will be able to perform full coordinate tracking.

References


