2023 IEEE Symposium Series on Computational Intelligence (SSCI)
Mexico City, Mexico. December 5-8, 2023

Robot PID control using reinforcement learning

Guillermo Puriel
Departamento de Control Automático
CINVESTAV-IPN
Mexico City, Mexico

Xiaoou Li
Departamento de Computación
CINVESTAV-IPN
Mexico City, 0736, Mexico

Brisbane Ovilla-Martinez
Departamento de Computación
CINVESTAV-IPN
Mexico City, 0736, Mexico

Wen Yu
Departamento de Control Automático
CINVESTAV-IPN
Mexico City, Mexico

***Abstract*—In this paper, the robot PID control is compensated by the reinforcement learning. The controller adjustment is proposed by the stability analysis. The reinforcement learning can compensate the dynamics of the robot. This method avoids the problems due to big integral gain of classical PID control. The experimental results show the effectiveness of the proposed controller.**

***Index Terms*—PID control,reinforcement learning, robust control**

I. Introduction

In recent years, the area of machine learning has attracted the attention of many researchers who have contributed to great progress. Machine learning is a set of algorithms or computational methods to ensure that a computer system is able to improve its performance as it interacts with the environment.

In the field of artificial intelligence, computer systems that display intelligent behaviors are studied. One of the most fruitful research areas in machine learning is reinforcement learning. Reinforcement learning focuses on optimal control and robotics applications [1]. Reinforcement learning aims to maximize reward when in state x and apply action u. Robot states can be considered continuous or discrete. Furthermore, the actions chosen are seen as the torques applied to the motor that result in a change of state [3]. The control policy within the literature can be found as the actions that are sent to the motor to generate a change of state. The idea here is to receive both the positions and the speeds and apply an input torque to the motor to produce a change in the output [2]. Therefore, finding a policy that maximizes the sum of the long-term rewards is the main goal of reinforcement learning.

Reinforcement learning can be found to solve different problems in the literature such as robotics, optimal control, multi-agent systems, game theory, etc. In general, the studies focus on the knowledge of optimal solutions, and not on learning or approximation methods.

In reinforcement learning, the central idea is that an agent learns to achieve a goal through their interaction with the environment. This problem is usually handled by Markov Decision Processes (MDP). The control of the robot within the Markov decision process can be seen as the knowledge of transitions and rewards that only depend solely on the current state. Therefore, a Markov state contains all the information related to dynamics, so once the current state is known, the history of the transitions that led the agent to that position is irrelevant. Reinforcement learning can be found in deterministic or stochastic form. In a deterministic decision problem, the execution of a certain action u_k, in a certain state x_k, always brings the agent to the same state x_{k+1}. In contrast, in a stochastic environment, each transition is associated with a probability distribution over the state space X, that is, the agent can end up in different states in two different executions of the same action u_k, at x_k. The MDP can also be considered small or large, where it refers to the size of the state and the spaces of action. Finally, they can be analyzed continuously or in the form of episodes, where in an episodic task the simulation of the interaction between the agent and the environment are divided into episodes. Each episode begins in an initial state and ends in a special state called the terminal state.

Q-Learning is a very popular control algorithm in reinforcement learning that uses temporal difference off-policy, where it directly approximates the optimal state action value function, regardless of the policy followed. If all actions for all state-action pairs are updated an infinite number of times, with a decreasing value of α, then the Q-Learning algorithm converges to Q with probability 1. Two important aspects can be found in reinforcement learning: One would be the tabular representation and the second the discretization of the state space. Also within these aspects some problems appear, such as: approximations based on domain knowledge, discretization of the state space, approximation of functions.

Recently, artificial intelligence algorithms have been used to solve problems in robotics and classical control. One of them is the neural networks that are used to estimate dynamic parameters of the robot model or parameters in the control loop. The Q-Learning algorithm has also been used to find the adjustment of the control loop gains or as a compensator for robot dynamics. Finally, the Q-Learning algorithm belongs

This work was supported in part by CONACYT under the grant CONACyT-A1-S-8216 and by CINVESTAV under the grants, SEP-CINVESTAV-62 and CNR-CINVESTAV

978-0-7381-4408-5/23/$31.00 ©2023 IEEE

to artificial intelligence and has an active and important participation in machine learning.

II. Reinforcement learning in the form of sliding mode

The goal in reinforcement learning is not only to maximize the immediate reward, but also the long-term reward which is represented as

$$V_k = \tilde{q}_{k+1} + \gamma \tilde{q}_{k+2} + \gamma^2_{k+3}\tilde{q}_{k+3} + \cdots = \sum_{l=k+1}^{T} \gamma^{l-1}\tilde{q}_l \quad (1)$$

where the vector $\tilde{q} = q - q^d \in \mathbb{R}^n$ is the control error, the vector $q^d \in \mathbb{R}^n$ is the desired reference, T represents the final time, γ is a parameter, $0 \leq \gamma \leq 1$.

The control objective can be formally defined as follows: given the desired reference $q_d \in \mathbb{R}^n$ constant for all $t \geq 0$, the problem is to design a law of control

$$\tau_k = \pi\left(\tilde{q}_k, q_k\right) \quad (2)$$

such that $q_k \to q^d \in \mathbb{R}^n$, π is the control policy.

In order to include control action to the value function (1), we introduce Q-function $Q^{\pi}(q_k, \tau_k)$ that represents the action τ_k in the state q_k following the policy π,

$$Q^{\pi}(q_k, \tau_k) = E\left[\sum_{l=0}^{\infty}\gamma^l R_{k+l+1} \mid q_k = x, \tau_k = u\right] \quad (3)$$

We define π^* as the *optimal policy* such that

$$V^*(q_k) = \max_{\pi_k} V^{\pi_k}(q_k) \text{ or } Q^*(q_k, \tau_k) = \max_{\pi_k} Q^{\pi_k}(q_k, \tau_k) \quad (4)$$

where Q^* is the *optimal action-value function* .

From Bellman principle, the value function $V(x)$ and Q-function $Q^{\pi}(q_k, \tau_k)$ have the recursive properties [4]

$$\begin{aligned} &V^{\pi}(q_k) = \sum_{\tau_k}\pi(\tau_k \mid q_k) \\ &= \sum_{q_k} p(q_k \mid q_k, \tau_k)\left[r(q_k, \tau_k, q_k) + \gamma V^{\pi}(q_k)\right] \\ &Q^{\pi}(q_k, \tau_k) = \sum_{q_k} p(q_k \mid q_k, \tau_k)\left[R(q_k \mid q_k, \tau_k) + \gamma V^{\pi}(q_k)\right] \end{aligned} \quad (5)$$

So the optimal value functions can be expressed recursively as

$$Q^*(q_k, \tau_k) = E[R_{k+1} + \gamma V^*(q_k) \mid q_k = x, \tau_k = u]. \quad (6)$$

This means

$$\begin{aligned} &Q^*(q_k, \tau_k) = E\left[R_{k+1} + \gamma \max\nolimits_{\tau_k} Q^*(q_k, \tau_k) \mid q_k = x, \tau_k = u\right] \\ &= \sum_{q_k} p(q_k \mid q_k, \tau_k)\left[r(q_k, \tau_k, q_k) + \gamma V^*(q_k)\right] \\ &= \sum_{q_k} p(q_k \mid q_k, \tau_k)\left[r(q_k, \tau_k, q_k) + \gamma \max\nolimits^*_{\tau_k} Q^*(q_k, \tau_k)\right] \end{aligned} \quad (7)$$

We use the following temporal differences learning to estimate the Q-function

$$\begin{aligned} &Q^{(k+1)}(q_k, \tau_k) = Q^{(k)}(q_k, \tau_k) \\ &+\alpha\left[R^{(k)} + \gamma \max\nolimits_{\tau_k} Q^{(k)}(q_k, \tau_k) - Q^{(k)}(q_k, \tau_k)\right] \end{aligned} \quad (8)$$

where q_k, τ_k are the state and action in time step k, α is the learning rate, $0 < \alpha \leq 1$, γ is the discount factor.

The robot control (2) becomes

$$\tau_k = \beta \arg\max_{\tau_k}\left[Q^{(k+1)}(q_k, \tau_k)\right] \quad (9)$$

β is a constant ($\beta > 0$).

The reinforcement learning controller (9) can be formed into the following form

$$\tau_k = \beta(-sign(\tilde{q}_k) + \Gamma_k) \quad (10)$$

where Γ_k is the difference between the sliding mode control $sign(\tilde{q}_k)$ and the reinforcement learning control (9).

Γ_k is decided by the Q-value function $Q^{(k)}$, which is calculated by the Q-learning (8). The algorithm of the Q-learning using sliding mode form is as follows.

```
$Q - learning\ algorithm\ for\ one\ control$
  Initialize $Q^{(k)}(\tilde{q}_{s_i}, \tau_{r_j})$ random
  **Repeat** in each task
      Initialize $\tilde{q}_{s_i}$
      **Repeat** in each episode
        Action $\tau_{r_j}$ from $\tilde{q}_{s_i}$,   $\tau_k = -sign(\tilde{q}_k)$
        Control law from $Q^{(k)}$,
            $\tau_r = \beta \arg\max_{\Gamma_k} Q^{(k+1)}(\tilde{q}_{s_i}, \tau_{r_j})$
        Action $\tau_{r_j}$ to the robot
        Update $Q^{(k)}(\tilde{q}_{s_i}, \tau_{r_j})$ with the observed data $q_k$:
          $Q^{(k+1)}(\tilde{q}_{s_i}, \tau_{r_j}) \leftarrow Q^{(k)}(\tilde{q}_{s_i}, \tau_{r_j})$
          $+ \alpha \left[ \begin{array}{c} V^{(k)} + \gamma \max_{\tau_r} Q^{(k)}(\tilde{q}_{s_{i+1}}, \tau_{r_{j+1}}) \\ -Q^{(k)}(\tilde{q}_{s_{i+1}}, \tau_{r_j}) \end{array} \right]$
      **Until** $\tilde{q}_{s_i}$ be a terminal state
  **Until** the task is finished
```

III. PID control with reinforcement learning compensation

The reinforcement learning control (10) has big chattering. Now we use the classical PID control. The reinforcement learning control (10) is applied as an compensator of the PID control.

The PID control with the reinforcement learning compensation is

$$\tau = K_p\tilde{q} + K_d\dot{\tilde{q}} + K_i\int_0^t \tilde{q}(\psi)d\psi + u_r \quad (11)$$

where the design matrices $K_p, K_d, K_i \in \mathbb{R}^{n\times n}$ called respectively proportional, derivative and integral gains, are positive and symmetric definite matrices.

u_r is the continuous time version of (10),

$$u_r = \beta\left[-sign(\frac{d}{dt}\tilde{q})\right] + \Gamma)$$

In the case of the regulation $\dot{q}_d = 0$, $\frac{d}{dt}\tilde{q} = -\dot{q}$

$$u_r = \beta\left[sign(\dot{q}) + \Gamma\right]$$

We use a new additional state variable ξ, and defined $\dot{\xi} = K_i\tilde{q}$, so the PID control is

$$\begin{aligned} &\tau = K_p\tilde{q} - K_d\dot{q} + \xi + \beta\left[sign(\dot{q}) + \Gamma\right] \\ &\dot{\xi} = K_i\tilde{q} \end{aligned} \quad (12)$$

The dynamics of a rigid serial $n-$link manipulator can be defined as [12].

$$M\left(q\right)\overset{..}{q}+C(q,\overset{.}{q})\overset{.}{q}+F(\overset{.}{q})+G\left(q\right)=\tau \tag{13}$$

here $q\in\Re^{n}$ represents the position of links, $\overset{.}{q}\in\Re^{n}$ represents the velocity of links, $M(q)\in\Re^{n\times n}$ the inertial matrix, $C(q,\overset{.}{q})\in\Re^{n\times n}$ represents the centripetal and Coriolis force matrix, $G(q)\in\Re^{n}$ is the vector of gravity, $F\in\Re^{n\times n}$ is a positive definite diagonal matrix of friction terms (friction viscous), $\tau\in\Re^{n}$ is the input control vector.

The closed-loop system is

$$\begin{array}{c} M(q)\ddot{q}+C(q,\dot{q})\dot{q}+F(\dot{q})+G(q) \\ =K_{p}\tilde{q}-K_{d}\dot{q}+\xi+\beta\left[sign(\dot{q})+\Gamma\right] \end{array}$$

In state space

$$\frac{d}{dt}\left[\begin{array}{c}\xi\\ \tilde{q}\\ \dot{q}\end{array}\right]=\left[\begin{array}{c} K_{i}\tilde{q}\\ -\dot{q}\\ M(q)^{-1}\left[\begin{array}{c} K_{p}\tilde{q}-K_{d}\dot{q}+\xi+\beta\left[sign(\dot{q})+\Gamma\right]\\ -C(q,\dot{q})\dot{q}-F(\dot{q})-G(q)\end{array}\right]\end{array}\right] \tag{14}$$

The equilibrium can be transferred to the origin by the following change of variable $\tilde{\xi}=\xi-G(q_{d})$. Note that the above equation is autonomous and its only equilibrium is the origin $\left[\tilde{\xi}^{T},\tilde{q}^{T},\dot{q}^{T}\right]^{T}=0\in\mathbb{R}^{3n}$. The following theorem gives the stability analysis of the PID control with reinforcement learning compensation.

Theorem 1: Consider robot dynamic (13) controlled by the PID+RL control (12), the closed loop system (14) is semi globally asymptotically stable at the equilibrium:

$$x=\left[\xi^{T}-G(q_{d}),\tilde{q}^{T},\dot{q}^{T}\right]^{T}=0\in\mathbb{R}^{3n}$$

with the following conditions gains:

$$\begin{array}{l} \lambda_{\min}(K_{p})\geq\frac{3}{2}k_{g}\\ \lambda_{\min}(K_{d})\geq\eta+\lambda_{\max}(M)\\ \lambda_{\max}(K_{i})\leq\eta\frac{\lambda_{\min}(K_{p})}{3\lambda_{\max}(M)}\\ \lambda_{\min}(\beta)\geq\Gamma+\lambda_{\max}(B_{f1}) \end{array} \tag{15}$$

where $\eta=\sqrt{\frac{1}{3}\lambda_{\min}(K_{p})\lambda_{\min}(M)},k_{g}$ satisfy the Lipschitz condition.

Proof 1: See Appendix.

Theorem 1 tells us how to select the gains of PID control in (11), and how to select the gain of the reinforcement learning compensator (10).

IV. Experimental results

This section shows experimental work on a 2-DOF manipulator robot. The manipulator robot is similar as the forearm that had a certain symmetry with the human arm. Figure 1 shows the manipulator robot from an isometric view in its initial condition, where both the arm and the forearm are extended and resting in their stable equilibrium position. Figure **??** shows both the arm and forearm are extended at different angles from zero. The work space comprises a circle with a diameter of approximately 1 meter. Table 1 shows the robot manipulator parameters. The desired final conditions are $q_{d1}=\pi/4$, $q_{d2}=\pi/4$.

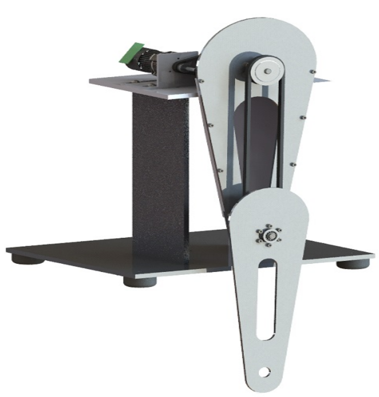

Fig. 1. 2 DOF manipulator robot arm

Table 1. 2 DOF Robot Manipulator Parameters.

Parameters	Description	Value
m_1	Arm Mass	$0.2393\ kg$
l_1	Arm Length	$0.240\ m$
l_{c1}	Center of mass Arm	$0.0684\ m$
I_1	Arm Inertia	$0.002547kgm^2$
b_1	Viscous shoulder friction	$0.0017\frac{Nm}{rad/s}$
q_1	Arm Position	$q_1\ rad$
m_2	Forearm mass	$0.1541\ kg$
l_2	Forearm Length	$0.200\ m$
l_{c2}	Center of Mass Forearm	$0.0574\ m$
I_2	Forearm Inertia	$0.001153kgm^2$
b_2	Elbow viscous friction	$0.0013\frac{Nm}{rad/s}$
q_2	Forearm position	$q_2\ rad$

Theorem 1 gives sufficient conditions for the minimal values of PID gains. From the parameters in Table 1 and (15), $\lambda_{\min}(K_{p})\geq\frac{3}{2}k_{g}$, $\lambda_{\min}(K_{p})\geq0.9156$, when the links are extended $q_{d1}=\pi/2$ and $q_{d2}=0$. We select $k_{g}=0.6104$, $\lambda_{\min}\left(K_{d}\right)\geq\eta+\lambda_{\max}(M)$, the eigenvalues of $M(q)$ are $\lambda_{1}=0.0178$, and $\lambda_{2}=0.0011$, $\lambda_{\max}(M)=0.0178$. Because $\eta=\sqrt{\frac{1}{3}\lambda_{\min}(K_{p})\lambda_{\min}(M)}$, $\lambda_{\min}(K_{p})=0.9156$, $\lambda_{\min}(M)=0.0011$, thus $\lambda_{\min}\left(K_{d}\right)\geq0.0183+0.0178=0.0361$. So

$$K_{p}=\left[\begin{array}{cc}1 & 0\\ 0 & 0.9156\end{array}\right],K_{i}=\left[\begin{array}{cc}0.3 & 0\\ 0 & 0.3\end{array}\right],K_{d}=\left[\begin{array}{cc}0.2 & 0\\ 0 & 0.1\end{array}\right]$$

The gain matrix of Q-learning is

$$\beta=\left[\begin{array}{cc}0.4 & 0\\ 0 & 0.05\end{array}\right]$$

The objective of PID+RL control is make the transient performance faster and less overshoot based on human-like learning algorithm. The initial conditions of the positions and velocities are zero. The desired joint positions are $q_{d1}=\pi/2[rad]$ and $q_{d2}=\pi/2[rad]$. The initial state error and the velocity are $\left[\begin{array}{cccc}\tilde{q}_{1} & \tilde{q}_{2} & \dot{q}_{1} & \dot{q}_{2}\end{array}\right]=\left[\begin{array}{cccc}\pi/4 & \pi/4 & 0 & 0\end{array}\right]$.

The whole control system is shown in Figure 2. The results of the control law are shown in Figure 3. We can see that the

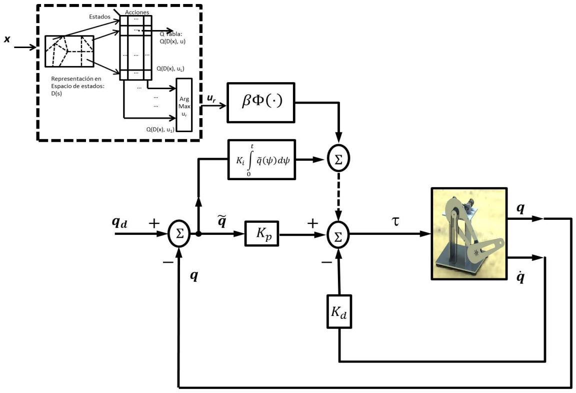

Fig. 2. The PID+QL control

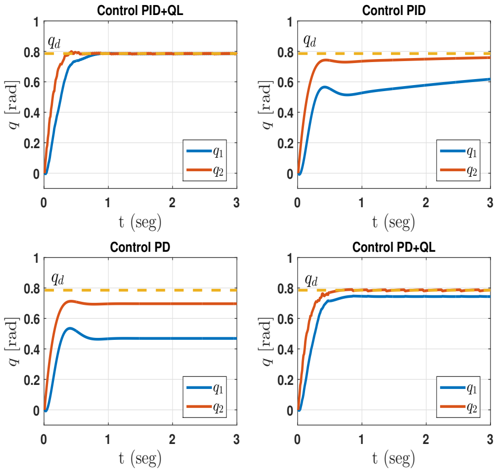

Fig. 3. Comparison between different control schemes

PID+RL control has better performance compared to the other controllers. The reinforcement learning uses 1000 episodes, and each episode has 1000 iterations. The learning time is about 120 seconds.

It is worth mentioning that the tuning of the gains is obtained directly from Theorem 1, and the minimum value of the eigenvalues is used. So the gain of the learning had active participation compensating the dynamics of the robot. The other controllers, such as the PID control, need more time to convergence the desired reference, approximately $10-15$ seconds.

To reduce the convergence time, we can only increase the proportional and derived gains. But the transient performances become bad. On the other hand, PD+RL and PID+RL do not need big PD gains. The academic criterion widely used in the robotic scientific community, to measure the performance of a control algorithm, are the integral of absolute error (IAE): $AE = \int_0^\infty \mid \tilde{q}(t) \mid dt$, and the integral of time for absolute error (ITAE): $ITAE = \int_0^\infty t \mid \tilde{q}(t) \mid dt$. The comparison results are shown in Table 2.

Table 2. Performance Index

	IAE q_1	ITAE q_1	IAE q_2	ITAE q_2
PD	1.0066	1.4293	0.3378	0.4060
PID	0.7588	0.9459	0.2070	0.1735
PID+RL	0.1855	0.0367	0.0979	0.0188
PD+RL	0.2864	0.2115	0.1088	0.0216

V. Conclusion

In order to solve the problem of robot PID control, such as bad transient performance and long convergence time, this paper combine the PID control with the reinforcement learning. The explicit tuning methods for the PID control and the reinforcement learning compensator are proposed. Stability analysis with Lyapunov method is given. The experimental results show the validation of the PID+RL control.

VI. Appendix

Proof 2 (Proof of Theorem 1): To study the stability of the origin of the state, we use the following candidate Lyapunov function

$$\begin{array}{c} V(\tilde{\xi},\tilde{q},\dot{q}) = \frac{1}{2}\dot{q}^T M(q)\dot{q} + \frac{1}{2}\tilde{q}^T K_p \tilde{q} + U(q_d - q) - k_u + \\ +\tilde{q}^T G(q_d) + \tilde{q}^T \tilde{\xi} + \frac{3}{2} G(q_d)^T K_p^{-1} G(q_d) + \frac{\alpha}{2}\tilde{\xi}^T K_i^{-1}\tilde{\xi} + \\ -\alpha \dot{q}^T M(q)\tilde{q} + \frac{\alpha}{2}\tilde{q}^T \left(K_d + B_{f1}\right)\tilde{q} + \alpha K_i^{-1}\int_0^t \Phi(\dot{q}) d\xi \end{array}$$

where $U(q_d - q)$ denotes the potential energy of the robot, $k_u = \min_q \left\{U(q_d - q)\right\}$, that is added so that $V(0) = 0$, and α is a positive constant that satisfies well-defined conditions for the candidate Lyapunov function to be positive definite. It will be shown that the Lyapunov candidate function is defined positive, $V(\tilde{\xi},\tilde{q},\dot{q}) \geq 0$. We divide the Lyapunov function into four parts $V(\tilde{\xi},\tilde{q},\dot{q}) = \sum_{i=1}^4 V(\tilde{\xi},\tilde{q},\dot{q})_i$:

$$\begin{array}{l} V(\tilde{\xi},\tilde{q},\dot{q})_1 = \frac{1}{6}\tilde{q}^T K_p \tilde{q} + \tilde{q}^T G(q_d) + \frac{3}{2} G(q_d)^T K_p^{-1} G(q_d) \\ V(\tilde{\xi},\tilde{q},\dot{q})_2 = \frac{1}{6}\tilde{q}^T K_p \tilde{q} + \tilde{q}^T \tilde{\xi} + \frac{\alpha}{2}\tilde{\xi}^T K_i^{-1}\tilde{\xi} \\ V(\tilde{\xi},\tilde{q},\dot{q})_3 = \frac{1}{6}\tilde{q}^T K_p \tilde{q} - \alpha \dot{q}^T M(q)\tilde{q} + \frac{1}{2}\dot{q}^T M(q)\dot{q} \\ V(\tilde{\xi},\tilde{q},\dot{q})_4 = U(q_d - q) - k_u + \frac{\alpha}{2}\tilde{q}^T \left(K_d + B_{f1}\right)\tilde{q} \\ +\alpha K_i^{-1}\int_0^t \Phi(\dot{q}) d\tilde{\xi} \end{array} .$$

The first term $V(\tilde{\xi},\tilde{q},\dot{q})_1$, we can easily see that if $K_p > 0$, then $V(\tilde{\xi},\tilde{q},\dot{q})_1$ is positive semi-definite.

$$V(\tilde{\xi},\tilde{q},\dot{q})_1 = \frac{1}{2}\left[\begin{array}{c} \tilde{q} \\ G(q_d) \end{array}\right]\left[\begin{array}{cc} \frac{1}{3}K_p & I \\ I & 3K_p^{-1} \end{array}\right]\left[\begin{array}{c} \tilde{q} \\ G(q_d) \end{array}\right] \geq 0.$$

The second term $V(\tilde{\xi},\tilde{q},\dot{q})_2$, we obtain the first condition of α for the function to be positive definite.

$$V(\tilde{\xi},\tilde{q},\dot{q})_2 = \frac{1}{2}\left[\begin{array}{c} \tilde{q} \\ \tilde{\xi} \end{array}\right]\left[\begin{array}{cc} \frac{1}{3}K_p & I \\ I & \alpha K_i^{-1} \end{array}\right]\left[\begin{array}{c} \tilde{q} \\ \tilde{\xi} \end{array}\right],$$

Using Sylvester's criterion, to prove that the matrix is positive definite, the determinant must be positive, so we have:

$$V(\tilde{\xi},\tilde{q},\dot{q})_2 = \frac{1}{2}\left[\begin{array}{c} \tilde{q} \\ \tilde{\xi} \end{array}\right]\left[\begin{array}{cc} \frac{1}{3}\lambda_{\min}(K_p) & 1 \\ 1 & \alpha\lambda_{\min}(K_i^{-1}) \end{array}\right]\left[\begin{array}{c} \tilde{q} \\ \tilde{\xi} \end{array}\right]$$

If

$$\begin{gathered}\tfrac{1}{3}\lambda_{\min}(K_p)\alpha\lambda_{\min}(K_i^{-1})-1\geq 0\\ \tfrac{\alpha}{3}\lambda_{\min}(K_p)\lambda_{\min}(K_i^{-1})\geq 1\\ \alpha\geq\tfrac{3}{\lambda_{\min}(K_p)\lambda_{\min}(K_i^{-1})}\end{gathered}$$

then $V(\tilde{\xi},\tilde{q},\dot{q})_2$ is positive definite. The third term $V(\tilde{\xi},\tilde{q},\dot{q})_3$, using Sylvester's criterion again

$$V(\tilde{\xi},\tilde{q},\dot{q})_3=\frac{1}{2}\left[\begin{array}{c}\tilde{q}\\ \dot{q}\end{array}\right]\left[\begin{array}{cc}\frac{1}{3}\lambda_{\min}(K_p) & -\alpha\lambda_{\max}(M)\\ -\alpha\lambda_{\max}(M) & \lambda_{\min}(M)\end{array}\right]\left[\begin{array}{c}\tilde{q}\\ \dot{q}\end{array}\right]$$

If $\frac{1}{3}\lambda_{\min}(K_p)\lambda_{\min}(M)-\alpha^2\lambda_{\max}(M)^2\geq 0$, $\frac{1}{3}\lambda_{\min}(K_p)\lambda_{\min}(M)\geq\alpha^2\lambda_{\max}(M)^2$, $\frac{\frac{1}{3}\lambda_{\min}(K_p)\lambda_{\min}(M)}{\lambda_{\max}(M)^2}\geq\alpha^2$, and $\frac{\sqrt{\frac{1}{3}\lambda_{\min}(K_p)\lambda_{\min}(M)}}{\lambda_{\max}(M)}\geq\alpha$, then $V(\tilde{\xi},\tilde{q},\dot{q})_3$ is positive definite. It is easy to see that $V(\tilde{\xi},\tilde{q},\dot{q})_4\geq 0$. So $V(\tilde{\xi},\tilde{q},\dot{q})\geq 0$. The condition for α is

$$\frac{\sqrt{\frac{1}{3}\lambda_{\min}(K_p)\lambda_{\min}(M)}}{\lambda_{\max}(M)}\geq\alpha\geq\frac{3}{\lambda_{\min}(K_p)\lambda_{\min}(K_i^{-1})},\quad(16)$$

We see that if K_p is big enough or K_i small enough, then $V(\tilde{\xi},\tilde{q},\dot{q})$ is semi-globally positive defined. The derivative in time of $V(\tilde{\xi},\tilde{q},\dot{q})$ along the closed loop system, and using

$$\frac{d}{dt}\int_0^t\Phi(\dot{q})d\tilde{\xi}=\frac{\partial\int_0^t\Phi(\dot{q})d\tilde{\xi}}{\partial\tilde{\xi}}\frac{\partial\tilde{\xi}}{\partial t}=\dot{\tilde{\xi}}^T\Phi(\dot{q})$$

We have

$$\begin{aligned}&\dot{V}(\tilde{\xi},\tilde{q},\dot{q})=\dot{q}^TM(q)\ddot{q}+\tfrac{1}{2}\dot{q}^T\dot{M}(q)\dot{q}+\tilde{q}^TK_p\dot{\tilde{q}}+\dot{q}G(q)\\ &+\dot{\tilde{q}}^TG(q_d)+\dot{\tilde{q}}^T\tilde{\xi}+\tilde{q}^T\dot{\tilde{\xi}}+\alpha\tilde{\xi}^TK_i^{-1}\dot{\tilde{\xi}}-\alpha\dot{\tilde{q}}^TM(q)\dot{q}\\ &-\alpha\tilde{q}^T\dot{M}(q)\dot{q}-\alpha\tilde{q}^TM(q)\ddot{q}+\alpha\tilde{q}^T\left(K_d+B_{f1}\right)\dot{\tilde{q}}\\ &+\alpha\dot{\tilde{\xi}}^TK_i^{-1}\Phi(\dot{q})\end{aligned}$$

Using the anti-symmetry property $\frac{1}{2}\dot{q}^T\dot{M}(q)\dot{q}-\dot{q}^TC(q,\dot{q})\dot{q}=0$, $M(\dot{q})=C(q,\dot{q})+C(q,\dot{q})^T$, we have the following:

$$\begin{aligned}&\dot{V}(\tilde{\xi},\tilde{q},\dot{q})=-\dot{q}^T\left[K_d-\alpha M(q)\right]\dot{q}-\tilde{q}^T[\alpha K_p-K_i]\tilde{q}\\ &-\alpha\tilde{q}^TC(q,\dot{q})^T\dot{q}\\ &-\alpha\tilde{q}^T\left[G(q_d)-G(q)\right]+\dot{q}^T\left[\beta(-sign(\dot{q})+\Gamma)-F(\dot{q})\right]\end{aligned}$$

Now the upper-bounds on the Lyapunov function is

$$\begin{aligned}&-\dot{q}^T\left[K_d-\alpha M(q)\right]\dot{q}\leq-\left[\ \lambda_{\min}(K_d)-\alpha\lambda_{\max}(M)\ \right]\ \|\dot{q}\|_2^2\\ &-\tilde{q}^T\left[\alpha K_p-K_i\right]\tilde{q}\leq-\left[\ \alpha\lambda_{\min}(K_p)-\lambda_{\max}(K_i)\ \right]\ \|\tilde{q}\|_2^2\end{aligned}$$

We use the properties $\|C(x,y)z\|\leq k_{C1}\|y\|\|z\|$ and $\|G(q_d)-G(q)\|\leq k_g\|x-y\|$,

$$\begin{gathered}-\alpha\tilde{q}^TC(q,\dot{q})^T\dot{q}\leq\alpha k_{C1}\|\tilde{q}\|_2\|\dot{q}\|_2^2\\ -\alpha\tilde{q}^T\left[G(q_d)-G(q)\right]\leq\alpha k_g\|\tilde{q}\|_2^2\end{gathered}$$

Using $\dot{q}^Tsign(\dot{q})=\|\dot{q}\|$,

$$\begin{gathered}-\dot{q}^T\beta sign(\dot{q})\leq-\lambda_{\min}(\beta)\|\dot{q}\|,\quad\dot{q}^T\Gamma\leq\Gamma\|\dot{q}\|\\ -\dot{q}^TF(\dot{q})\leq\lambda_{\max}(B_{f1})\|\dot{q}\|\end{gathered}$$

After taking the upper bounds of the Lyapunov function,

$$\begin{aligned}&\dot{V}(\tilde{\xi},\tilde{q},\dot{q})\leq-\left[\lambda_{\min}(K_d)-\alpha\lambda_{\max}(M)\right]\|\dot{q}\|_2^2\\ &-\left[\alpha\lambda_{\min}(K_p)-\lambda_{\max}(K_i)\right]\|\tilde{q}\|_2^2\\ &\alpha k_{C1}\|\tilde{q}\|_2\|\dot{q}\|_2^2+\alpha k_g\|\tilde{q}\|_2^2-\lambda_{\min}(\beta)\|\dot{q}\|_1\\ &+\Gamma\|\dot{q}\|_1+\lambda_{\max}(B_{f1})\|\dot{q}\|_1\end{aligned},$$

The time derivative $\dot{V}$ is

$$\begin{aligned}&\dot{V}(\tilde{\xi},\tilde{q},\dot{q})\leq-\left[\lambda_{\min}(K_d)-\alpha\lambda_{\max}(M)-\alpha k_{C1}\|\tilde{q}\|_2\right]\|\dot{q}\|_2^2\\ &-\left[\alpha\lambda_{\min}(K_p)-\lambda_{\max}(K_i)-\alpha k_g\right]\|\tilde{q}\|_2^2\\ &-\left[\lambda_{\min}(\beta)-\Gamma-\lambda_{\max}(B_{f1})\right]\|\dot{q}\|_1\end{aligned}$$

If we choose the upper bound of the position error as $\|\tilde{q}\|_2$

$$\|\tilde{q}\|_2\leq\frac{\lambda_{\max}(M)}{\alpha k_{C1}},$$

Taking the first term of $\dot{V}(\tilde{\xi},\tilde{q},\dot{q})$, the following relation is obtained

$$\lambda_{\min}(K_d)-\alpha\lambda_{\max}(M)-\alpha k_{C1}\|\tilde{q}\|_2>0$$

Because $\|\tilde{q}\|_2\leq\frac{\lambda_{\max}(M)}{\alpha k_{C1}}$ and $\alpha=\frac{\sqrt{\frac{1}{3}\lambda_{\min}(K_p)\lambda_{\min}(M)}}{\lambda_{\max}(M)}$, $\lambda_{\min}(K_d)-\alpha\lambda_{\max}(M)-\alpha k_{C1}\frac{\lambda_{\max}(M)}{\alpha k_{C1}}\geq 0$,

$$\lambda_{\min}(K_d)\geq\eta+\lambda_{\max}(M)$$

where $\eta=\sqrt{\frac{1}{3}\lambda_{\min}(K_p)\lambda_{\min}(M)}$. Now we take the second term of $\dot{V}(\tilde{\xi},\tilde{q},\dot{q})$, and using $\alpha=\frac{3}{\lambda_{\min}(K_p)\lambda_{\min}(K_i^{-1})}$, so $\alpha\lambda_{\min}(K_p)-\lambda_{\max}(K_i)-\alpha k_g\geq 0$, $\alpha\lambda_{\min}(K_p)\geq\lambda_{\max}(K_i)+\alpha k_g$, then

$$\lambda_{\min}(K_p)\geq\frac{3}{2}k_g$$

The minimum value for K_i

$$\frac{\sqrt{\frac{1}{3}\lambda_{\min}(K_p)\lambda_{\min}(M(q))}}{\lambda_{\max}(M)}\geq\alpha\geq\frac{3}{\lambda_{\min}(K_p)\lambda_{\min}(K_i^{-1})}$$

Because $\lambda_{\min}(K_i^{-1})=\frac{1}{\lambda_{\max}(K_i)}$

$$\eta\frac{\lambda_{\min}(K_p)}{3\lambda_{\max}(M)}\geq\lambda_{\max}(K_i)$$

Therefore, K_p,K_d K_i and β need (15), and then

$$\dot{V}(\tilde{\xi},\tilde{q},\dot{q})\leq 0$$

So $\tilde{\xi},\tilde{q},\dot{q}\in\mathcal{L}_\infty$ and

$$\begin{aligned}&\int\limits_t^\infty\left(\lambda_{\min}\left(A\right)\|\tilde{\xi}\|^2+\lambda_{\min}\left(A\right)\|\tilde{q}\|^2+\lambda_{\min}\left(A\right)\|\dot{q}\|^2\right)d\tau\\ &\leq V-V_\infty<\infty\end{aligned}$$

Thus, $\tilde{\xi}_t\in\mathcal{L}_2\cap\mathcal{L}_\infty$ and $\frac{d}{dt}\tilde{\xi}\in\mathcal{L}_\infty$, are also for $\tilde{q}$ and $\dot{q}$. By the Barbalat's lemma, we can conclude that

$$\lim_{t\to\infty}\tilde{\xi}=0,\quad\lim_{t\to\infty}\tilde{q}=0,\quad\lim_{t\to\infty}\dot{q}=0\qquad(17)$$

References

[1] Y.Jin, Decentralized Adaptive Fuzzy Control of Robot Manipulators, *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, Vol.28, No.1, 47-57, 1998

[2] M.W.Spong and M.Vidyasagar, *Robot Dynamics and Control,* John Wiley & Sons Inc., Canada, 1989.

[3] F.L.Lewis, D.M.Dawson, C.T.Abdallah, *Robot Manipulator Control: Theory and Practice*, 2nd Edition, Marcel Dekker Inc, New York, NY 10016, 2004

[4] R. S. Sutton and A. G. Barto, "Reinforcement learning: an introduction", The MIT Press, March 1998. ISBN 0262193981

[5] Deisenroth, M. Peter, G. Neumann, and J. Peters, “A survey on policy search for robotics”, Foundations and Trends in Robotics vol. 2, pp. 1-142, 2013.

[6] A. S. Polydoros and L. Nalpantidis, “Survey of Model-Based Reinforcement Learning: Applications on Robotics,” Journal of Intelligent & Robotic Systems, vol. 86, pp. 153-173, 2017.

[7] J. J. Kober, A. Bagnell, and J. Peters, “Reinforcement learning in robotics: A survey”, The International Journal of Robotics Research vol. 32, pp. 1238-1274, 2013.

[8] L. P. Kaelbling , M. L. Littman, and A. W. Moore, “Reinforcement learning: A survey.” Journal of artificial intelligence research vol. 4 , pp. 237-285, 1996.

[9] T. Moerland, J. Broekens, and C. M. Jonker, “Emotion in Reinforcement Learning Agents and Robots: A Survey.” arXiv preprint arXiv:1705.05172 , 2017 .

[10] M. Ghavamzadeh, S. Mannor, J. Pineau and A. Tamar. “Bayesian reinforcement learning: A survey”, Foundations and Trends in Machine Learning, vol. 8, No. 5-6, pp.359-483, 2015.

[11] R. Kelly and V. Santibáñez, “Control de Movimiento de Robots Manipuladores”, Pearson Prentice Hall, 2003

[12] M.W. Spong and M. Vidyasagar, “Robot Dynamics and Control,” John Wiley & Sons Inc.,Canada, 1989.

[13] F.L. Lewis, A. Yesildirek and K.Liu, “Multilayer Neural-Net Robot Controller with Guaranteed Tracking Performance,” IEEE Trans. on Neural Networks, vol.7, No.2, pp. 388-399, 1996.

[14] F.L. Lewis, “Neural Network Control of Robot Manipulators,” IEEE Expert, vol.11, No.2, pp. 64-75, 1996.

[15] Y. Zheng, S. Luo, and Z. Lv, “Control double inverted pendulum by reinforcement learning with double cmac network”, Pattern Recognition, 2006. ICPR 2006. 18th International Conference on. vol. 4. IEEE, 2006.

[16] S. Hosokawa and K. Nakano, “A reward allocation method for reinforcement learning in stabilizing control of T-inverted pendulum” , Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON), 2012 9th International Conference on. IEEE, 2012.

[17] S. Hosokawa, J. Kato and K. Nakano, “A reward allocation method for reinforcement learning in stabilizing control tasks” , Artificial Life and Robotics, vol. 19, No.2, pp. 109-114, 2014.

[18] S. Mahadevan, “Average reward reinforcement learning: Foundations, algorithms, and empirical results,” Machine learning, vol. 22, No.1 , pp.159-195, 1996.

[19] Y. Zheng, S. w. Luo and Z. Lv, “Active exploration planning in reinforcement learning for Inverted Pendulum system control,” Machine Learning and Cybernetics, 2006 International Conference on. IEEE, 2006.

[20] W. Linglin, L. Yongxin and Z. Xiaoke, “Design of reinforce learning control algorithm and verified in inverted pendulum.” Control Conference (CCC), 2015 34th Chinese. IEEE, 2015.

[21] M. Hehn, R. D. Andrea, “A Flying inverted pendulum,” Robotics and Automation (ICRA), 2011 IEEE International Conference on. IEEE, 2011.

[22] R. Figueroa, A. Faust, P. Cruz , L. Tapia and R. Fierro, “Reinforcement learning for balancing a Flying inverted pendulum,” Intelligent Control and Automation (WCICA), 2014 11th World Congress on. IEEE, 2014.

[23] A. Faust, P. Ruymgaart, M. Salman, R. Fierro and L. Tapia. ”Continuous action reinforcement learning for underactuated dynamical system control.” Adaptive Motion Planning Research Group Technical Report TR13-002 (2013).

[24] J. Kober, , and J. R. Peters. “Policy search for motor primitives in robotics,” Advances in neural information processing systems. 2009.

[25] Z. Huang, J. Liu, Z. Li and C. Y. Su, “Adaptive impedance control of robotic exoskeletons using reinforcement learning.” Advanced Robotics and Mechatronics (ICARM), International Conference on. IEEE, 2016.