Adaptive Control Design and Stability Analysis of RR Robotic Manipulators

Tsung-Wun Wang
Department of Mechanical Engineering
National Taiwan University
Taipei, Taiwan
b05502097@ntu.edu.tw

***Abstract*—In this paper, the adaptive control design and stability analysis of robotic manipulators is presented based on two approaches, i.e., Lyapunov stability theory and hyperstability theory. For the Lyapunov approach, two types of controlling a 2-degree-of-freedom (2-DOF) robotic manipulator are introduced, i.e., computed-torque control and adaptive control. In addition, the adaptive control is applied to the end-effector motion control and force control, and motion control (e.g., position control, trajectory tracking) will be emphasized in this paper. On the other hand, the control system developed through integrating proportional integral derivative (PID) and model reference adaptive control (MRAC) is convergent by hyperstability approach. The characteristics of the systems, developed by PID control, MRAC control and hybrid (PID+MRAC) control are compared.**

***Keywords*—Lyapunov stability; hyperstability; computed-torque control; model reference adaptive control (MRAC); robotic manipulator**

I. Introduction

The control of robotic manipulators has become important due to the development of the flexible automation. Requirements such as the high speed and high precision trajectory tracking make the modern control indispensable for versatile applications of manipulators.

Conventional controllers for robotic structures are based on independent control schemes in which each joint is controlled separately by a simple servo loop. When a robotic manipulator end-effector grasps an object to conduct work, it will change the dynamics of the robotic manipulator since the mass and initial properties of the grasped object may be unknown [1]. This classical control scheme (for example a PD control) is inadequate for precise trajectory tracking. The imposed performance for industrial applications requires the consideration of the complete dynamics of the manipulator. Moreover, in real-time applications, the ignoring parts of the robot dynamics or errors in the parameters of the robotic manipulator may cause the inefficiency of this classical control. An alternative solution to PD control is the computed torque technique. This classical method is in fact a nonlinear technique that takes account of the dynamic coupling between the robot links. The main disadvantage of this structure is the assumption of an exactly known dynamic model [2].

Industrial robotic manipulators are exposed to structured and unstructured uncertainties. Structured uncertainties are characterized by having a correct model but with parameter uncertainty (unknown loads and friction coefficients, imprecision of the manipulator link properties, etc.). Unstructured uncertainties are characterized by unmodelled dynamics. Generally speaking, two classes of strategies have been developed to maintain performance in the presence of the parameter uncertainties: robust control and adaptive control. The adaptive controllers can provide good performances in face of very large load variation. Therefore the adaptive approach is intuitively superior to robust approach in this type of application. When the dynamic model of the system is not known a priori, a control law is designed based on an estimated model. This is the basic idea behind adaptive control strategies [6].

Model reference adaptive control (MRAC) is a more advanced control technique, which was first developed by Landau [7], and it will be emphasized because the conventional controllers cannot cope with the changing load. Therefore, the computed-torque and MRAC will be first introduced, and then propose a new hybrid control, a combination of proportional–integral–derivative (PID) and MRAC controller, to observe its performance by hyperstability approach. Also, the convergent behavior and characteristics under the situation of simple PID, simple MRAC and hybrid (PID+MRAC) control will be compared.

II. Problem Formulation

A. RR Robotic Manipulator

In this semester, we have learned a lot of adaptive control strategies in class. However, the motion of equation, derived from Lagrangian technique, is more complicated than any systems we have seen. Also, there are many unknown parameters and uncertainties for a 2-degree-of-freedom (2-DOF) robotic manipulator. Therefore, we will mainly focus on this planar robotic manipulator with two revolute joints and design a control strategy to observe its performance.

B. Stability

The stability of a system is important for all of us, and the stable system is the first priority that we should follow because an unstable system is dangerous. To overcome this problem, the stability should not be proved in the way that local stability occurs around the equilibrium points; the system should be globally stable to ensure that no other dangerous cases will occur. To find out the solution, we should derive a general case to prove that the system can be stable in all cases.

III. Proposeed Approach

Since the motor output, the torque in most cases, can be set, the value of torque should be determined while the robotic manipulator is tracking the trajectory simultaneously. The controller is designed to track the variation of the torque, but before we start to simulate, the stability of the control system should be verified. Lyapunov stability is a solution to determine if a system is stable. Two theories will be introduced before the proofs are deducted.

A. *Lyapunov Stability* [4]

Let $V(x,t)$ be a non-negative function with derivative $\dot{V}$ along the trajectories of the system. If $V(x,t)$ is locally positive definite and $\dot{V}(x,t) \leq 0$ locally in x and for all t, then the origin of the system is locally stable (in the sense of Lyapunov).

B. *Barbalat's lemma* [5]

If $\lim_{t\to\infty} \int_0^t f(\tau)d\tau$ exists and is finite, and $f(t)$ is a uniformly continuous function, then $\lim_{t\to\infty} f(t) = 0$.

C. *Notation*

The symbol q denotes the actual joint angle, while q_d represents the estimated joint angle of the robotic manipulator. The error $\tilde{q}$ represents the difference between q and q_d. In addition, the regressor vector is denoted as y; the true unknown parameter is denoted as θ, while θ_d represents the estimated unknown parameter.

IV. Computed-Torque Control

A. *Lyapunov Stability*

A control scheme of a spring-mass system can be modelled as $F = -k_P(x - x_d) - k_V\dot{x}$ in [1], and the PD controller works is that it mimics the spring-mass system. The system can approach to the globally stable point, so the here the detailed proof will be derived. It is known that the PD controller can be described as follows:

$$\tau = -K_P\tilde{q} - K_D\dot{q} \tag{1}$$

Suppose the Lyapunov function is

$$V = \frac{1}{2}\dot{q}^T H(q)\dot{q} + \frac{1}{2}\tilde{q}^T K_P \tilde{q} \tag{2}$$

Where $H(q)$ represents the inertia matrix. Then,

$$\dot{V} = \dot{q}^T H(q)\ddot{q} + \dot{\tilde{q}}^T K_P \tilde{q} \tag{3}$$

Substitute $H\ddot{q}$ and τ into (3), we have

$$\dot{V} = -\dot{q}^T(K_D + D)\dot{q} \le 0 \tag{4}$$

Based on Barbalat's lemma, if $\dot{V}$ approaches to zero, then $\dot{q}$ approaches to zero. By the assumption of $\dot{q_d} = 0$ and $\ddot{q} = -H(q)^{-1}K_P\tilde{q}$, the value of $\ddot{q}$ is zeros if the error $\tilde{q}$ is zero.

V. Stability of MRAC Design

A. *Motion Control (Joint Space)*

The dynamic equation of an n-link robotic manipulator is given by

$$H(q)\ddot{q} + C(q,\dot{q})\dot{q} + D(q,\dot{q})\dot{q} + G(q) = \tau \tag{5}$$

where q is the joint angle, $H(q)$ is the inertia matrix, $C(q,\dot{q})\dot{q}$ is the centripetal and Coriolis torques, $D(q,\dot{q})\dot{q}$ is the friction, and $G(q)$ is the gravitational torque. Now suppose our controller output is defined by

$$\tau = y\theta_d - K_D S \tag{6}$$

and the following adaptive law is chosen

$$\dot{\theta_d} = -\Gamma y^T S \tag{7}$$

where s is defined as $s = \dot{\tilde{q}} - \lambda\tilde{q}$, λ is a positive constant, Γ is a symmetric positive definite matrix, and $\dot{q_r}$ is defined as $\dot{q_r} = \dot{q_d} - \lambda\tilde{q}$. Therefore,

$$\dot{s} = \ddot{q} - \ddot{q_r} \tag{8}$$

and the system described by (5) can be modelled as

$$y\tilde{\theta} = H\dot{S} + K_D S + DS + CS \tag{9}$$

where $y\tilde{\theta}$ is equal to $H\ddot{q_r} + C\dot{q_r} + D\dot{q_r} + G$. Now suppose that the unknown θ does not change with time and the Lyapunov function is chosen as

$$\mathrm{V} = \frac{1}{2}\left(S^T H S + \tilde{\theta}^T \Gamma^{-1}\tilde{\theta}\right) \tag{10}$$

Then,

$$\dot{\mathrm{V}} = -S^T(K_D + D + C)S \le 0 \tag{11}$$

Based on Barbalat's lemma, if $\dot{V} \to 0$, then $S \to 0$. Hence, it can be proved that $\tilde{q} \to 0$ and $\dot{\tilde{q}} \to 0$.

B. *Motion Control (Operational Space)*

By applying a force, F, at the end-effector, the joint torque is required to achieve the task space control:

$$\tau = J^T F \tag{12}$$

where J is a Jacobian matrix. The operational space control includes the motion control and force control, and their target is to control the motion of the end-effector and to control the force that the end-effector should apply, respectively. From (5) and (12), the end-effector equation of motion in the operational space is

$$M_x(x)\ddot{x} + V_x(x,\dot{x})\dot{x} + G_x(x) = F \tag{13}$$

Where x is the end-effector position and orientation, $M_x(x)$ is the kinetic energy matrix, $V_x(x,\dot{x})$ is the centripetal and Coriolis force, and $G_x(x)$ is the gravitational force. By using the decoupling technique as [1], the dynamics of the robotic manipulator can be estimated when the mass and initial properties is changing and unknown.

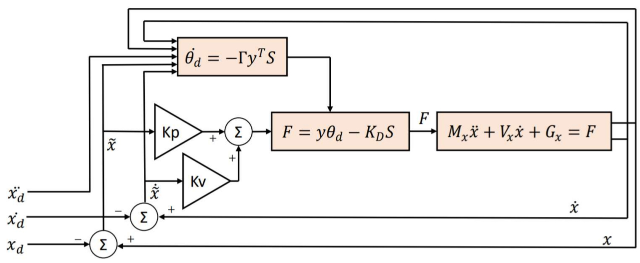
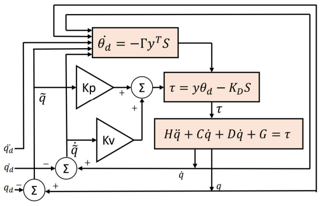
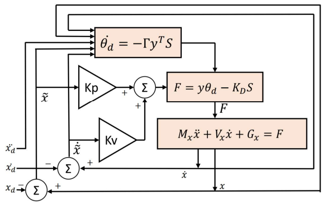
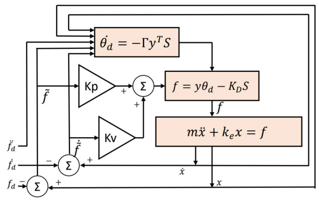


Fig. 1. Adaptive Motion Control Structure.

Suppose the Lyapunov function

$$\mathrm{V} = \frac{1}{2}\left(S^T M_x S + \tilde{\theta}^T \Gamma^{-1}\tilde{\theta}\right) \tag{14}$$

TABLE I. COMPARISON AMONG DIFFERENT CONTROL SCHEMES

	Motion Control (Joint Space)	**Motion Control (Operational Space)**	**Force Control**
Dynamic Equation	$H\ddot{q} + C\dot{q} + D\dot{q} + g = \tau$	$M\ddot{x} + V\dot{x} + G = F$	$m\ddot{x} + k_e x = f$
Unknown	$H\ddot{q} + C\dot{q} + D\dot{q} + g$	$M\ddot{x} + V\dot{x} + G$	$m\ddot{x}$
Structure			

The model in (13) is reconstructed to $y\tilde{\theta} = M\dot{S} + K_D S + VS$. Also, the same control law (6) and adaptive law (7) is used to prove the stability. Thus,

$$\dot{V} = -S^T(K_D + V_x)S \leq 0. \tag{15}$$

Based on Barbalat's lemma, if $\dot{V} \to 0$, then $S \to 0$. Hence, it can be proved that $\tilde{x} \to 0$ and $\dot{\tilde{x}} \to 0$.

C. Force Control

The dynamic equation in force control case is

$$m\ddot{x} + k_e x = f \tag{16}$$

where x is the end-effector displacement, m is the mass of the end-effector, and k_e is the gain of the virtual spring. The end-effector dynamic equation (16) is different from the one in motion control case. In the motion control case, when the robot end-effector grasps an object, the unknown parameters are in $H\ddot{q} + C\dot{q} + D\dot{q} + g$ (joint space) or $M\ddot{x} + V\dot{x} + G$ (operational space). In the force control case, the unknowns are in $m\ddot{x}$, as shown in Table I.

Now assume the Lyapunov function as follows,

$$V = \frac{1}{2}\left(S^T m S + \tilde{\theta}^T \Gamma^{-1} \tilde{\theta}\right) \tag{17}$$

By using the decoupling technique and substituting (16), (6) and (7) into (17), $\dot{V}$ can be calculated:

$$\dot{V} = -S^T K_D S \leq 0. \tag{18}$$

Based on Barbalat's lemma, if $\dot{V} \to 0$, then $S \to 0$. Thus, $\tilde{x} \to 0$ and $\dot{\tilde{x}} \to 0$.

VI. SIMULATIONS

A. Modelling of 2-DOF Robotic Manipulator

For a planar robotic manipulator with two revolute joints, the dynamic equation is deducted by Lagrangian technique.

$$\tau_1 = \left[(m_1 + m_2)L_1^{\ 2} + m_2 L_2^{\ 2} + 2m_2 L_1 L_2 c_2\right]\ddot{q}_1 + \left[m_2 L_2^{\ 2} + m_2 L_1 L_2 c_2\right]\ddot{q}_2$$
$$+2(-m_2 L_1 L_2 s_2)\dot{q}_1\dot{q}_2 + (-m_2 L_1 L_2 s_2)\dot{q}_2 \tag{19}$$

$$\tau_2 = \left[m_2 L_2^{\ 2} + m_2 L_1 L_2 c_2\right]\ddot{q}_1 + m_2 L_2^{\ 2}\ddot{q}_2 + m_2 L_1 L_2 s_2 \dot{q}_1^{\ 2} \tag{20}$$

Equation (19) and (20) can be re-parameterized, and then the unknown parameters will be estimated. The regressor vector y has a dimension of 2×9, and the length of unknown parameters θ is 9:

$$y =$$

$$\begin{bmatrix} \ddot{q}_{r1} & \ddot{q}_{r2} & \ddot{q}_{r1} + \ddot{q}_{r2}\cos(q_2)(2\ddot{q}_{r1} + \ddot{q}_{r2}) - \sin(q_2)(\dot{q}_2\dot{q}_{r1} + (\dot{q}_1\dot{q}_2)\dot{q}_{r2}) & \ddot{q}_{r1} & \ddot{q}_{r1} + \ddot{q}_{r2} & \cos(q_1) & \cos(q_1)\cos(q_1 + q_2) \\ 0 & 0 & \ddot{q}_{r1} + \ddot{q}_{r2}\cos(q_2)(2\ddot{q}_{r1} + \ddot{q}_{r2}) - \sin(q_2)(\dot{q}_2\dot{q}_{r1} + (\dot{q}_1\dot{q}_2)\dot{q}_{r2}) & 0 & \ddot{q}_{r1} + \ddot{q}_{r2} & 0 & 0 & \cos(q_1 + q_2) \end{bmatrix} \tag{21}$$

$$\theta = [m_1 L_{c1}^{\ 2}, m_2 L_1^{\ 2}, m_2 L_{c2}^{\ 2}, m_2 L_1 l_{c2}, I_1, I_2, m_1 l_{c1} g, m_2 l_1 g, m_2 l_{c2} g\,]^T \tag{22}$$

B. Computed-Torque Control

The planar robotic manipulator with two revolute joints is used in the simulation. The known parameters are (SI units): the lengths of the robotic manipulator are L_1= 1 and L_2 = 0.5 meters; the mass of the links are m_1 = 10 kg and m_2 = 2.5 kg. In the computed-torque control, let the gain K_P and K_D be 2000 and 1000, respectively. By assuming zero initial conditions of the robot manipulator, the control objective is to track the desired trajectory given by

$$q_{d1} = 0.4\sin(0.4\pi t), \ q_{d2} = -0.5\sin(0.5\pi t) \tag{23}$$

Then the tracking errors approach to zero, as shown in Fig.2.

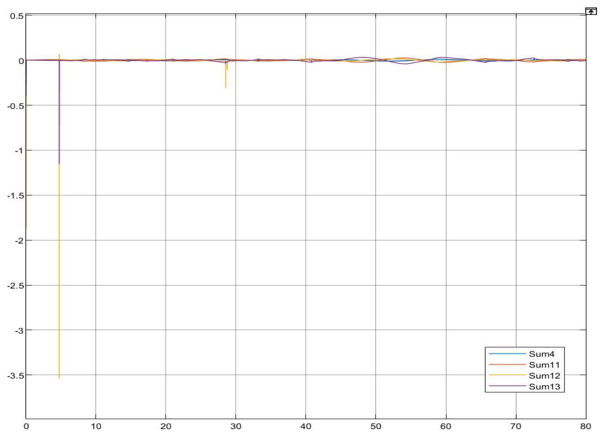


Fig. 2. Tracking errors $\tilde{q} = [\,\tilde{q}_1, \tilde{q}_2, \dot{\tilde{q}}_1, \dot{\tilde{q}}_2\,]^T$ – the computed-torque case.

C. *MRAC Design*

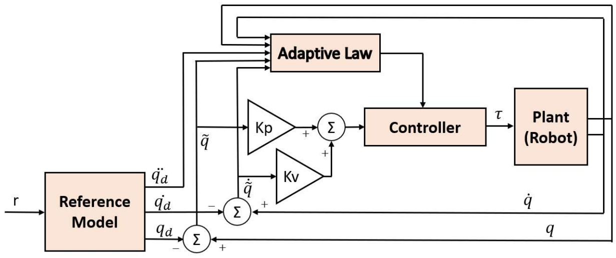


Fig. 3. Adaptive control structure – the motion control case.

The adaptive control design is followed by the above figure, and, and let the reference inputs as pulse functions, as shown in the Fig. 4. Furthermore, the reference model and the robot plant are defined by (24) and (25), respectively.

$$W_m = \frac{1}{s^2+8s+25} \tag{24}$$

$$G_p = \frac{1}{s^2+3s+25} \tag{25}$$

1) *Results*

The unknown parameters, as described in (22), converge to constants after the values of reference input change, as shown in Fig. 5.

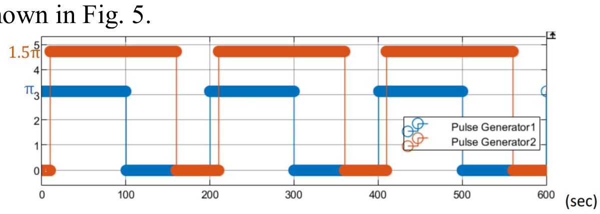


Fig. 4. Reference input r_1 and r_2.

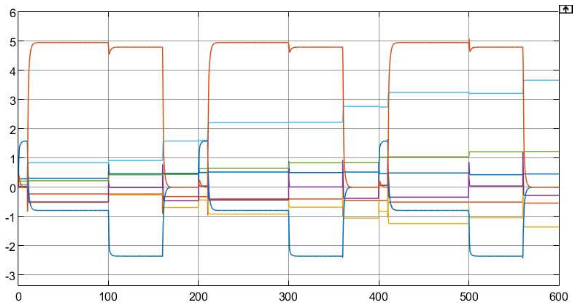


Fig. 5. Nine unknown parameters θ, defined by (22), are convergent.

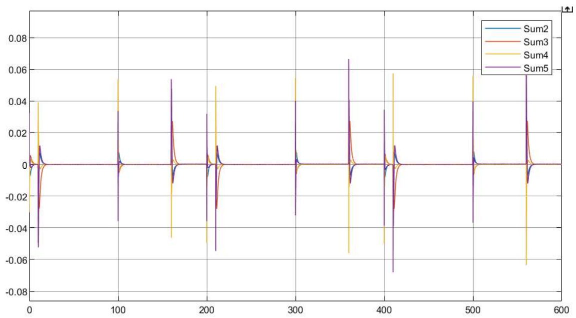


Fig. 6. Tracking errors – the MRAC case.

The adaptive control law (6) and (7) is implemented. We can see that even if the estimated parameters θ_d are used, the evolution of tracking errors remains good.

VII. Hyperstability

A. *Hyperstability*

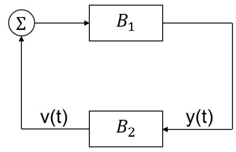


Fig. 7. Standard feedback system.

A necessary and sufficient condition for the system B_1 to be hyperstable is that the transfer matrix

$$Z(s) = D + C(sI - A)^{-1}B \tag{26}$$

should be positive real.

B. *Simulation of Hybrid Control (PID+MRAC)*

1) *Structure:*

For the hyperstability approach, a control system is developed through integrating a proportional–integral–derivative (PID) control system and a model reference adaptive control (MRAC) system, as shown in Fig. 8.

Assume the reference model and the robot system are the same as (24) and (25), respectively. The reference input is the same as the previous, as shown in Fig. 4.

2) *PID Controller:*

Four PID controllers are designed to improve the performance: two for the first joint, and the other two for the second joint.

With the help of PID tuner in Simulink, the stable point can be found with respect to the system requirement. There are four parameters need to be designed for each PID block. Here is the compensator formula:

$$K_p + K_I \frac{1}{s} + K_D \frac{N}{1+N\frac{1}{s}} \tag{27}$$

TABLE II. Parameters in (27)

Error	**Proportional K_p**	**Integral K_I**	**Derivative K_D**	**Filter Coefficient N**
$q_1 - q_{d1}$	-37.81	-78.28	-4.541	974
$q_2 - q_{d2}$	-32.93	-67.78	-3.955	921
$\dot{q}_1 - \dot{q}_{d1}$	-2.00	-315.0	0	100
$\dot{q}_2 - \dot{q}_{d2}$	-1.882	-98.42	0	100

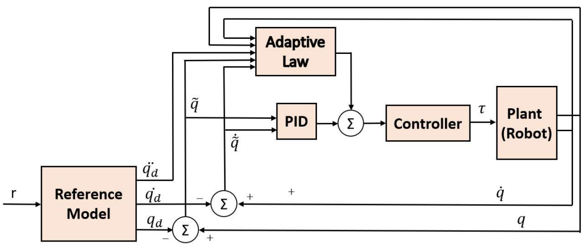


Fig. 8. Hybrid control structure (PID+MRAC).

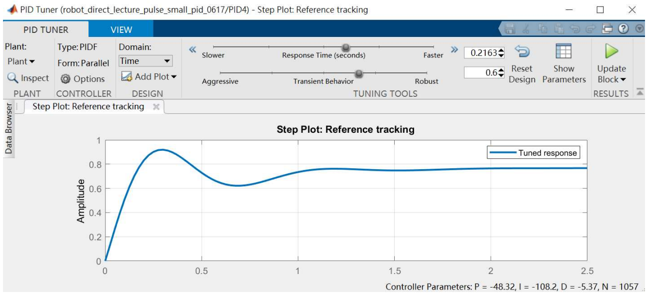


Fig. 9. PID tuner in Simulink

3) Results

a) Unknown θ

As described in (22), the unknown parameters are estimated, and all of them are convergent.

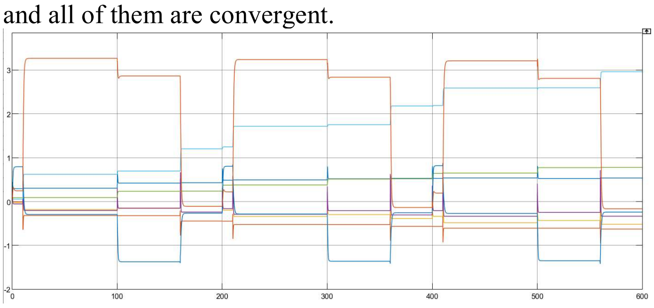


Fig. 10. Unknown parameters – the hybrid control (PID+MRAC) case.

b) Tracking errors $\tilde{q} = [\,\widetilde{q_1}, \widetilde{q_2}, \dot{\widetilde{q_1}}, \dot{\widetilde{q_2}}\,]^T$

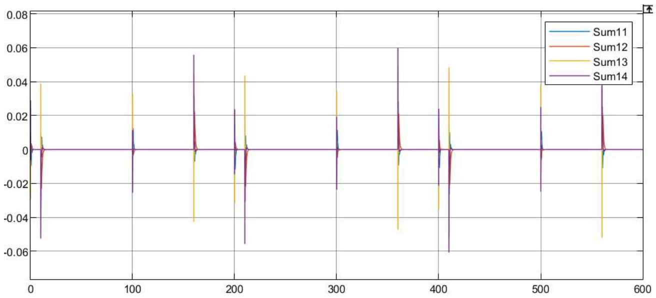


Fig. 11. Tracking errors – the hybrid control (PID+MRAC) case.

C. Comparison between MRAC control and hybrid control

Now let us mainly focus on the performance of joint 1.

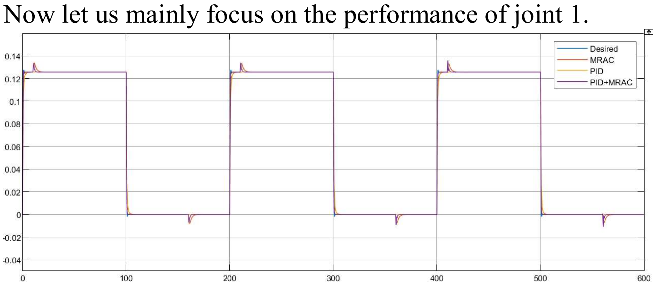


Fig. 12. The joint angle q_1 in the three control cases.

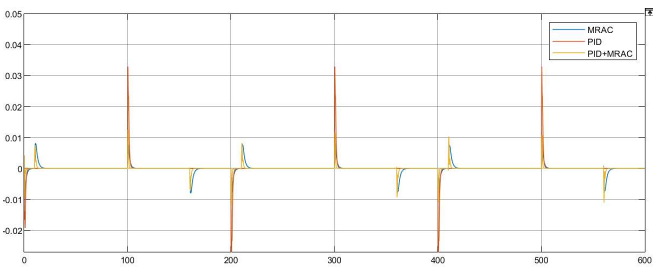


Fig. 13. The tracking error $\widetilde{q_1}$ in the three control cases.

To compare the details, Fig. 12 and Fig. 13 are enlarged, as shown below.

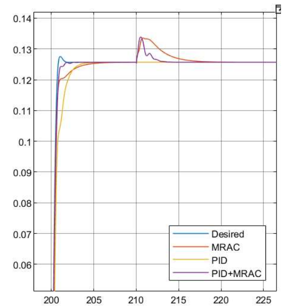


Fig. 14a The joint angle q_1

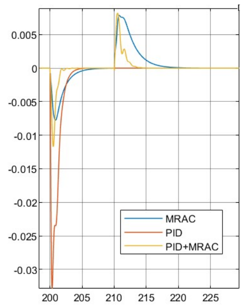


Fig. 14b Tracking error $\widetilde{q_1}$

Fig. 14. Zooming in Fig. 12 and Fig. 13. Please note that the reference input r is the pulse function, and r_1 changes the value at 200 (sec), while r_2 changes the value at 210 (sec).

At 200 (sec), the Fig. 14a shows that the hybrid control case (PID+MRAC) converges quickly than the other two cases. However, at 210 (sec), no overshoot occurs in PID cases, but hybrid control is still better than simply MRAC. In Fig. 14b, when the value of r_1 changes at 200 (sec), the error of the PID case is the largest, while the error of the MRAC case is the smallest. In the three cases, the error of hybrid control converges to zero most quickly.

D. Discussion

1) What if K_p, K_I, K_D are no longer constants?

It seems that the many constants designed in all cases are not adaptive in different environments. It has been shown that with a fixed PID controller combined with the MRAC scheme has improved the robot performance. How about the controller with adaptive K_p, K_I and K_D? Then the stability of this new scheme should be proved, and that is my future work.

2) Other possible hybrid control schemes?

For sure, there are many types of control schemes, and many of them has been proved to be stable. However, the combination of several control strategies is still questionable. The main focus is to determine a better controller such that the performance of the existing system can be improved. As control engineers, this is the problem that we try to solve.

VIII. Conclusion

With respect to the hyperstability approach, a control system was developed through integrating a PID and an MRAC, and the convergent behavior and characteristics under the situation of the PID, MRAC, and PID+MRAC were compared. The outcome indicated the new enhanced hybrid (PID+MRAC) converged faster than other simple control cases.

TABLE III. Comparison among Different Control Schemes

Control Methods	**Advantage**	**Disadvantage**
Computed-torque	For linear models	For known models
MRAC control	For unstructured models	
Hybrid control (MRAC+PID)	Fast convergence	

REFERENCES

[1] W. Bin, “Adaptive Control Design and Stability Analysis of Robotic Manipulators,” Actuators, vol. 7, pp. 89–105, December 2018.

[2] D. Popescu, D. Selisteanu, C. Ionete, M. Roman and L. Popescu, “Neural and Adaptive Control Strategies for a Rigid Link Manipulator”, Robot Manipulators Trends and Development, Agustin Jimenez and Basil M Al Hadithi, IntechOpen, March 2010.

[3] V. V. Chalam, “Adaptive control systems: techniques and applications,” New York: Marcel Dekker, April 1987.

[4] F. L. Lewis, D. M. Dawson, and C. T. Abdallah, Robot Manipulator Control: Theory and Practice, 2nd ed., CRC Press, 2003.

[5] R. M. Murray, S. S. Sastry, and Li Zexiang. A Mathematical Introduction to Robotic Manipulation, 1st ed., CRC Press, 1994.

[6] R. Ortega, & M.W. Spong, “Adaptive motion control of rigid robots: a tutorial”. Automatica, Vol. 25, pp. 877-888, November 1989.

[7] Dubowsky, S.; Desforges, D. The application of model-referenced adaptive control to robotic manipulators. J. Dyn. Syst. Meas. Control, vol. 101(3), pp. 193–200, September 1979.