

# A New Approach to Switching Robust Adaptive Control

## Author:

Quang, Nguyen C; Tordon, Michal J; Katupitiya, Jayantha

## **Publication details:**

Proceedings of the 43rd IEEE Conference on Decision and Control pp. 3247-3252 0-7803-8683-3 (ISBN)

### Event details:

43rd IEEE Conference on Decision and Control Atlantis, Paradise Island, Bahamas

# Publication Date:

2004

# **DOI:** https://doi.org/10.26190/unsworks/88

## License:

https://creativecommons.org/licenses/by-nc-nd/3.0/au/ Link to license to see what you are allowed to do with this resource.

Downloaded from http://hdl.handle.net/1959.4/10964 in https:// unsworks.unsw.edu.au on 2024-04-26

# A New Approach to Switching Robust Adaptive Control

N.C. Quang, M.J. Tordon, and J. Katupitiya School of Mechanical and Manufacturing Engineering, University of New South Wales, Australia. Email: M.Tordon@unsw.edu.au

*Abstract*—Comparison of advantages and disadvantages of control methods used in nonlinear systems is presented. To take advantage of both adaptive and robust control methods, a novel method which switches between the two is proposed. Robust control is used during transients and situations where parameters are uncertain. During steady state operations, adaptive control is used. Switching between the two methods is carried out based on the standard deviation of the estimated parameter vector. The method has been implemented on a 2 DOF articulated robot. Experimental results are presented to prove the robustness and the accuracy of the new control method.

#### I. INTRODUCTION

Systems in reality are uncertain. Two kinds of uncertainties exist; parametric uncertainties and uncertain nonlinearities. In designing modern high performance systems, these two types of uncertainties must be accounted for and the successful elimination of their effects is of practical significance. So far numerous conventional methods such as PD, PID, computed torque control (CTC) as well as advanced control methods have been tried to solve the above mentioned problems. In general, the advanced methods are superior to the conventional control methods and the two principal categories are; the adaptive control (AC) and the robust control (RC).

Adaptive control methods usually use parameter adaptation laws to update estimated model parameters of systems on-line. They use updated parameters in control laws to make systems adapt to parametric changes. Typically, in AC the transient performance is not considered. In order to make the systems perform well, nonlinearities of the systems must be known. Such controllers are described in [1]-[4].

In [5] the robustness of adaptive controllers applied to a robot manipulator is analyzed. As an improvement, the authors modified some adaptive control laws in an attempt to deal with unmodelled robot dynamics and external disturbances.

Apart from the above, many other adaptive control methods have been developed to obtain better control performance. The results obtained with the adaptive control methods cited above indicate that transient errors are large in comparison to those in the steady state. This drawback results from the fact that adaptive controllers usually deal only with the ideal case of parametric uncertainties rather than disturbances. The process of parameter estimation takes some time to converge. Especially at the start, the parameters are poorly estimated, resulting in the poor transient performance. In addition, when external disturbances are taken into account, while the robustness increases the decrease in asymptotic stability can not be avoided. The reason is that when external disturbances come into play, they affect the parameter adaptation laws. This leads to the parameter drift. As a consequence, if the external disturbances are large, they can cause systems to become unstable. These problems are well known and are described in [4]-[6].

Robust control methods, on the other hand, can successfully deal with most of the weaknesses of the adaptive control methods. Two of the popular robust control approaches are the Sliding Mode Control (SMC) [3], [7]-[9] and  $H_{\infty}$  optimal control [10], [11].

For linear systems,  $H_{\infty}$  controllers can be obtained in the state space by solving Riccati equation [10] or by using Linear Matrix Inequality (LMI) technique [11]. In nonlinear systems,  $H_{\infty}$  control problem becomes much more difficult since it seeks the solution of the very complex Hamilton-Jacobi (HJ) equation. In practice, different kinds of numerical approximation methods are suggested to solve the HJ equation.

The SMC is preferably applied because of its simplicity. Many different versions of SMC have been suggested and developed [3], [7]-[9]. In the SMC, the tracking errors are forced to zero by applying nonlinear switching control inputs. That way, systems will slide along a manifold and reach the origin of the sliding space in spite of external disturbances and uncertain dynamics. As a result, at the transient stage when large uncertainties are present, systems are robust, i.e. have good transient performance. Due to the absence of parameter adaptation, RC in general and the SMC in particular, make no discrimination between external disturbances and parametric uncertainties during the transient or the steady state periods. Thus, the quality and performance at the transient and the steady state periods are similar. Consequently, tracking accuracy in the steady state shows no improvement. In the classical SMC methods, the switching control inputs cause chattering. To overcome this problem, fixed or time varying boundary layers are incorporated in the control law, resulting in reduced switching frequency.

As can be noted, robust controllers and adaptive controllers exhibit good performance under different circumstances. Therefore, it is possible to design a methodology where AC and RC will complement each other to produce superior overall performance.

This paper presents a new method called Switching Robust and Adaptive Control (SRAC) that exploits advantages of both control methods to improve the performance of nonlinear systems with modelling uncertainties and disturbances by switching between adaptive and robust controllers. Switching criteria are established to ensure bumpless switching between RC and AC. This method has been experimentally verified and compared with a number of other conventional and advanced control methods in a real 2 DOF manipulator with unknown but fixed parameters and payload. The results presented show superior performance by way of increased adaptability and reduced tracking errors.

#### **II. THE CONTROLLER**

#### A. The concept of SRAC

The new method exploits the advantages of both adaptive and robust control methods. It uses one of the methods when it is advantageous to do so and switches to the other method when the other method becomes the preferred choice. The core issue is to determine when either robust or adaptive controller downgrades their performance so that controller switching can take place.

Other researchers have taken similar approaches [5] by treating the parametric uncertainties first and then attempting to make the system robust. This approach only guarantees asymptotic stability and does not mention the accuracy.

Another approach [12] is to design robust controller as a baseline controller that ensures the designed transient response and certain tracking accuracy. However, since in this method, the two control components are integrated into one overall control law with the parameter adaptation running concurrently with the baseline robust control law, care should be taken to avoid conflict between the two controllers. This conflict comes from the fact that the baseline robust controller requires the parametric uncertainties to be bounded, whereas the parameter adaptation may deliver unacceptably large changes of the parameters due to external disturbances. To solve this conflict, they modified the parameter adaptation laws. In their system, the baseline robust control law component is always in use even when external disturbances are very small. The permanent presence of the baseline robust controller increases the steady state tracking errors. It is better to take full advantage of AC and turn off the baseline robust control law component. However, the smooth switching from RC to AC should be appropriately treated. Usually the process of switching from AC to RC works well as RC naturally has good



Fig. 1. The SRAC design process.

robustness provided that the robust controller operates alone without the adaptive controller. On the other hand, switching from RC to AC needs more careful consideration as the robustness of AC to the external disturbances is inherently weak and is not guaranteed. The performance of the system with the AC is greatly influenced by the initial estimates of the parameters. The estimates of the parameter vector and the associated errors can be described by the following equations:

$$\tilde{a}(0) = a(0) - \hat{a}(0),$$
 (1a)

$$\tilde{a}(t) = a(t) - \hat{a}(t). \tag{1b}$$

$$\tilde{a}(t) = f(t)\tilde{a}(0). \tag{1c}$$

where:

- a(0), a(t) designate the value of actual parameter vector at time instants 0 and t respectively;
- â(0), â(t) designate the estimates of parameter vector at time instants 0 and t respectively;
- ã(0), ã(t) designate the errors in the estimates of parameter vector at time instants 0 and t respectively; and function f(t) is chosen such that f(t) → 0 when t→∞.

If the parameter guesses are poor, it will take long time for the parameters to converge. In addition, switching from RC to AC can give rise to an overshoot worse than that caused by AC or RC. It can even destabilize the system.

The method proposed in this work guarantees improved performance in comparison to performance of AC or RC operating alone all the time. The general block diagram for designing SRAC controllers in this manner is shown in Fig. 1. The approach considers using RC at the start to cope with the transient effects. The switching criteria described below will then switch to AC. This way the system will preserve the robustness against parametric uncertainties and external disturbances usually unknown at the start. The main issue to be addressed is the determination of the switching criteria that ensures stability and performance when switching from RC to AC. Given that the success of AC relies on certainties of the parameters, the appropriate moment to switch to AC is when the certainty of the parameter vector is established. To accomplish this, parameter estimation is carried out in parallel with RC. The data generated by RC are good and reliable to estimate the parameters of the system in preparation for switching over to AC. Note, however, that during this time no parametric adaptation that would affect RC is carried out. This approach is in contrast to the parameter adaptation that was embedded into the methodology presented in [12]. The parameter estimates by the estimator that runs in parallel with the robust controller are used as initial parameter vector  $\hat{a}(0)$  for the adaptive controller after the system has switched from RC to AC. As  $\hat{a}(0)$  is close to the actual a(0), the vulnerability of the adaptive controller to poor parameter estimates and parameter mismatching is minimized. Further, the parameter adaptation of the adaptive controller converges much faster. While AC is in use, if the external disturbances cause parameters to drift unacceptably, the system will switch to RC and the parameter estimation will recommence. The switching process between AC and RC will continue to take place guaranteeing the robustness and the accuracy for the entire duration of operation.

#### B. Design of SRAC

The design of our novel SRAC involves the design of the robust controller, the adaptive controller, the parameter estimator and the switching criteria.

1) Dynamic model of robot manipulators: It is well known that a dynamic model of a rigid link manipulator can be written in the following form [13]:

$$H(q)\ddot{q} + C(q,\dot{q})\dot{q} + G(q) + F_f \dot{q} + d(q,\dot{q},t) = u, \quad (2)$$

where:

- $q \in \mathbb{R}^n$  is the joint displacement vector.
- $u \in \mathbb{R}^n$  is the applied joint torque.
- $H(q) \in \mathbb{R}^{n \times n}$  is the inertia matrix.
- $C(q,\dot{q}) \in \mathbb{R}^n$  is the Coriolis and centripetal force matrix.
- $G(q) \in \mathbb{R}^n$  is the gravitational forces.
- $F_f$  is the viscous friction force vector.
- $d(q, \dot{q}, \ddot{q}, t) \in \mathbb{R}^n$  is the external disturbance vector.

The robot manipulators have several fundamental properties useful in designing control systems as described in [13]. They are:

**Property 1:** The inertia matrix H(q) is symmetric, positive definite.

**Property 2:** Each degree of freedom is controlled by an independent control input.

**Property 3:** In the absence of disturbances (i.e.  $d(q, \dot{q}, \ddot{q}, t)$ ), the dynamic model of robot manipulators can be written in the following form:

$$H(q)\ddot{q} + C(q,\dot{q})\dot{q} + G(q) + F_f \dot{q} = Y(q,\dot{q},\ddot{q})a = u, \quad (3)$$

where:

where

- The regressor Y(q, q, q) ∈ R<sup>n×l</sup> is a matrix of known functions.
- a ∈ R<sup>l×1</sup> is a vector of parameters such as link masses, moments of inertia, etc.

The relationship (3) is a linear relationship of parameters to control inputs. This property is useful in designing adaptive control laws.

**Property 4:** The matrix  $D(q, \dot{q}) \stackrel{\Delta}{=} \dot{H}(q) - 2C(q, \dot{q})$  is skew symmetric.

The proofs of these properties of the dynamic model can be found in [13].

2) Robust controller: As described in [3], the robust control law to be used in our control method is:

$$u = \hat{u} - K \cdot sgn(s/\phi), \tag{4}$$

$$\hat{u} = \hat{H}(q)\ddot{q}_r + \hat{C}(q,\dot{q})\dot{q}_r + \hat{G}(q) + \hat{F}_f\dot{q} - d,$$
 (5)

$$s = \dot{\tilde{q}} + \lambda \tilde{q},\tag{6}$$

is the sliding surface;  $q_d(t) \in \mathbb{R}^n$  is the desired joint motion trajectory;  $\tilde{q}(t) \stackrel{\Delta}{=} q(t) - q_d(t)$  is the motion tracking error vector;  $\dot{q}_r \stackrel{\Delta}{=} \dot{q}_d - \lambda \tilde{q}$  is the reference velocity vector;  $\phi$  is the fixed boundary layer;  $\lambda$  is a symmetric, positive definite matrix; K is a matrix chosen to ensure the stability of the system.

The Lyapunov function candidate V(t) is defined as in [3]:

$$V(t) = \frac{1}{2}s'Hs.$$
 (7)

Differentiating V(t) and implementing mathematical transformations, we obtain for our system:

$$\dot{V} = s' \cdot \left[ \tilde{H}(q)\ddot{q}_r + \tilde{C}(q,\dot{q})\dot{q}_r + \tilde{G}(q) + \tilde{F}_f\dot{q} - K \cdot sgn(s/\phi) - d \right],$$
(8)

where ' denotes transpose of matrices or vectors.

For the system to be stable in Lyapunov's sense ( $V \leq 0$ ), the components  $k_i$  of the vector K are chosen such that:

$$k_{i} \geq \left| \left[ \tilde{H}(q)\ddot{q}_{r} + \tilde{C}(q,\dot{q})\dot{q}_{r} + \tilde{G}(q) + \tilde{F}_{f}\dot{q} \right]_{i} \right| + \\ + \left| |d_{i}| \right|_{\infty} + \eta_{i},$$

$$(9)$$

where  $\eta_i > 0$  is a small constant.

It is observed that the robust control law comprises the following terms:

- When decomposing the first term  $\hat{u}$  of (4), we can see in (5) that it introduces a feed forward action (with the presence of  $q_d$ ,  $\dot{q}_d$ ,  $\ddot{q}_d$ ) to improve the tracking errors and the disturbance rejection, and a feedback action (with the feedback of q,  $\dot{q}$ ) and a component containing  $\lambda$  to compensate for nonlinear effects and joint decoupling.
- The second term  $-K \cdot sgn(s/\phi)$  of (4) introduces the robust contribution that accommodates the external disturbances and the uncertainties. Therefore, the

resulting system has improved control performance in the presence of uncertainties. Data generated during RC should be reliable for the parameter estimation.

*3)* Adaptive controller: The adaptive control method that the system will switch to is designed using the adaptive passivity-based control method described in [14]. Accordingly, the proposed control law is

$$u = \hat{Y}(q, \dot{q}, \dot{q}_r, \ddot{q}_r)\hat{a} - K_D s, \qquad (10)$$

$$s = \tilde{q} + \lambda \tilde{q}. \tag{11}$$

The regressor  $\hat{Y}(q, \dot{q}, \dot{q}_r, \ddot{q}_r)$  is determined from:

$$\hat{H}(q)\ddot{q}_{r} + \hat{C}(q,\dot{q})\dot{q}_{r} + \hat{G}(q) + \hat{F}_{f}\dot{q} = \hat{Y}(q,\dot{q},\dot{q}_{r},\ddot{q}_{r})a.$$
(12)

And the online parameter adaptation law is:

$$\hat{a} = -\Gamma \hat{Y}' s, \tag{13}$$

with  $\Gamma$  symmetric positive definite and the initial values of the adapted parameter vector  $\hat{a}(t_0)$  are taken from the values of the parameters estimated at the end of the robust control period at the instant  $t_0$ . Although there is no guaranty that  $\hat{a}(0)$  will converge to the set of true parameters (when the persistent exciting condition in [15] is not satisfied) the convergence time of adaptation by the adaptive controller can be significantly shortened. Thus the performance of the adaptive controller is improved.

Define the system Lyapunov function candidate V(t) as in [14]:

$$V(t) = \frac{1}{2} \left[ s' H(q) s + \tilde{a}' \Gamma^{-1} \tilde{a} \right].$$
 (14)

For Lyapunov stability, it requires:

$$\dot{V}(t) = -s' K_D s \leqslant 0. \tag{15}$$

The control law (10) is composed of the following three contributions:

- The term Y(q, q, q, q, q) â is introduced to compensate for nonlinear dynamics and joint decoupling.
- The term *K*<sub>D</sub>*s* is to introduce a stabilizing linear control feedback action of PD type on the tracking errors.
- The vector of the estimated parameters â is updated using the integral adaptive law (13) to ensure asymptotic compensation of the terms in the manipulator model. Γ determines the convergence rate of parameter adaptation law.

In SRAC, the parameter estimation and update rate should be properly chosen to ensure the robustness of the system. During transient a smaller sampling interval should be used to make the system less sensitive to external disturbances and measurement noises, which are likely to cause the parameter drift. It can be increased in the steady state stage when external disturbances and measurement noises are insignificant. It is preferable to use faster convergence rate for the estimator that is running in parallel with the robust controller as the estimator dynamics do not affect the robust control law. However, after switching to AC, the rate of convergence of parameter adaptation must be kept low as high convergence rate may lead to oscillatory parameter estimation that can adversely affect the controller performance. The low convergence rate, however, is now affordable as  $\hat{a}(t)$  is already a good estimate of a(t).

4) Parameter estimator: In the robotic control, the identification and the parameter adaptation are usually based on the relationship (3).

For simplicity, the regressor matrix Y can be replaced with  $Y_d$  based on the *a priori* information of the system, specifically the reference trajectory of the system instead of the actual trajectory, as follows:

$$\hat{H}(q_d)\ddot{q}_d + \hat{C}(q_d, \dot{q}_d)\dot{q}_d + \hat{G}(q_d) + \hat{F}_f \dot{q}_d 
= Y_d(q_d, \dot{q}_d, \ddot{q}_d)a.$$
(16)

This will reduce the sensitivity of the regressor matrix Y, and therefore the sensitivity of the estimated parameters, to the external disturbances and measurement noise.

Since the uncertainties of the system during the transient stage are strong, parameter estimators with the properties of fast parameter convergence and robustness to uncertainties are needed. The gradient estimators and standard least squares estimators are not suitable because they have slow convergence. Thus they are not capable of estimating time varying parameters and are not robust to the disturbances. As a good candidate, we propose the modified least squares estimator with a bounded forgetting factor and a dead zone [16]. Accordingly, we can derive the estimated parameter vector as follows :

$$\hat{a}[k] = \begin{cases} (I + TP[k]Y'_d[k]Y_d[k])^{-1} \cdot (\hat{a}[k-1] + TP[k]Y'_d[k]u[k]) & |\varepsilon| \ge \Delta \\ & \hat{a}[k-1] & |\varepsilon| < \Delta \end{cases}, (17)$$

where  $\varepsilon = Y_d \hat{a} - u$  (*u*: control torques); *T* is the sampling time; *k* is a current computing instant; *P*[*k*] is the update gain; and  $\Delta$  is the dead zone.

The dead zone  $\Delta$  is introduced to avoid the parameter drift caused by the errors usually mixed with the external disturbances and measurement noises or by the impersistent excitation.

And the gain update law for the estimator is:

$$P[k] = (I + \alpha[k]T) \cdot (P[k-1] + TY'[k]Y[k])^{-1}, \quad (18)$$

with the forgetting factor:

$$\alpha[k] = \alpha_0 \cdot \left(1 - \frac{||P[k]||}{P_0}\right) \ ||P[k]|| \leqslant P_0, \tag{19}$$

where  $\alpha_0, P_0$  are the maximum forgetting rate and the bound for the gain P[k].

Thus, by introducing the dead zone and the forgetting factor the parameter estimator obtains the following properties: large gain bound, smooth parameter estimation, fast parameter convergence and robustness against external disturbances. These properties are beneficial for the parameter estimator.

TABLE I TRACKING ERRORS OBTAINED WITHOUT LOAD

Contr.	Track.	Without load (in degree)	
type	errors	Transient	Steady
PID	$\tilde{q}_1$	-1.50 to $1.20$	-1.50 to $1.20$
	$\tilde{q}_2$	-0.50 to $0.50$	-0.50 to $0.50$
CTC	$\tilde{q}_1$	-0.04 to $0.02$	-0.04 to $0.02$
	$ ilde q_2$	-0.20 to $0.18$	-0.20 to $0.18$
RC	$\tilde{q}_1$	-0.10 to $0.03$	-0.10 to $0.03$
	$\tilde{q}_2$	-0.10 to $0.03$	-0.10 to $0.03$
AC	$\tilde{q}_1$	-0.25 to $-0.08$	-0.08 to $0.01$
	$ ilde q_2$	-0.22 to $0.06$	-0.06 to $0.03$
SRAC	$ ilde q_1$	-0.10 to $0.03$	-0.03 to 0.03
1	$\tilde{q}_2$	-0.10 to $0.03$	-0.025 to $0.025$

TABLE II TRACKING ERRORS OBTAINED WITH LOAD

Contr.	Track.	With load (2kg in degree)	
type	errors	Transient	Steady
PID	$\tilde{q}_1$	-2.20 to $1.30$	-2.20 to $1.30$
	$ ilde q_2$	-0.80 to $0.60$	-0.80 to $0.60$
CTC	$\tilde{q}_1$	-0.06 to $0.04$	-0.06 to $0.04$
	$\tilde{q}_2$	-0.48 to $0.22$	-0.48 to $0.22$
RC	$\tilde{q}_1$	-0.20 to $0.06$	-0.20 to $0.06$
	$\tilde{q}_2$	-0.20 to $0.07$	-0.20 to $0.07$
AC	$\tilde{q}_1$	-0.42 to $0.00$	-0.10 to $0.01$
	$\tilde{q}_2$	-0.42 to $0.06$	-0.06 to $0.03$
SRAC	$\tilde{q}_1$	-0.20 to $0.06$	-0.03 to $0.03$
	$\tilde{q}_2$	-0.20 to $0.07$	-0.03 to $0.03$

5) Switching criteria: As mentioned before, control law can be switched from RC to AC and vice versa in order to harness benefits of both controllers. Greater attention needs to be paid when switching from RC to AC. Given that the stability and performance of adaptive controllers depend on the quality of estimated parameters, the convergence of estimated parameters is used as the condition for switching from RC to AC.

In the case where the parameters of the system a(t) are constant, the standard deviation of the parameters estimated over a certain period of time is used as the switching criterion:

$$\sigma(\hat{a}) = \sqrt{var(\hat{a})} < \sigma_s, \tag{20}$$

where  $var(\hat{a})$  is the variance of the parameter vector  $\hat{a}$ ,  $\sigma_s$  is *a priori* chosen constant threshold vector that triggers switching.

#### **III. EXPERIMENTAL RESULTS**

The control methods are now examined and verified in a typical real system - a 2 DOF articulated manipulator formed by joints 2 and 3 of a Puma 560 Robot - in the presence of gravitational forces, frictional forces and the payload. For simplicity, in our experiments the parameters of the system including the payload are assumed to be unknown but fixed although the method can be applied to the system with slow time varying parameters. Sampling time is 2 ms, control hardware is a 1.8 GHZ Pentium III computer equipped with a PCI interface card. The model of the robot is built using Maple.







Fig. 4. Tracking errors obtained with SRAC.

In the experiment, two trajectory profiles were followed and the method worked well under the both profiles. For brevity, however, only results obtained with the first trajectory profile which is extracted from [3]:

$$q_{d1}(t) = \frac{\pi}{6} (1 - \cos(2\pi t)),$$
$$q_{d2}(t) = \frac{\pi}{4} (1 - \cos(2\pi t)),$$

with initial states of the system being:

$$q_1(0) = 0$$

 $q_2(0) = 0,$ 

are presented in this paper.

The software to control the robot has been written in Visual C++ and is interrupt driven. The program allows the implementation of a variety of different controllers.

The detailed experimental results obtained with different controllers under different conditions are presented in Tables I and II. Table I shows the comparison without the payload. Table II shows the same with the payload.

As can be seen in the two tables, the tracking errors obtained with SRAC in the transient state (labeled transient tracking errors) are the same as those obtained with RC, but they are significantly improved during the steady state (labeled steady tracking errors) in comparison to RC or AC.

Fig. 2, Fig. 3, Fig. 4 show the experimental results corresponding to the results in Table II (i.e. the payload was attached) when only RC, only AC, and the SRAC were used respectively. Up to around 7 seconds, the system was driven by RC while the parameter estimation takes place in the background. When the standard deviations of the estimated parameters fall below  $\sigma_s$  (*a priori* chosen), the controller switches to AC. As can be seen, the transfer was bumpless and the tracking errors are much less. This can be especially appreciated by comparing the time histories of Fig. 2 to Fig. 4 for the periods 7 to 15 seconds.

#### **IV. CONCLUSION**

A method has been proposed that harnesses the benefits of robust control and adaptive control. The control algorithm switches to robust control when the parameter adaptation of the adaptive controller deteriorates beyond a set standard deviation vector. During robust control, an independent parallel parameter estimation is carried out and when the standard deviations of the estimated parameters fall within a pre-set standard deviation vector, control switches back to adaptive control. It is shown by experiments that the SRAC method improves performance by preserving the advantages of AC and RC while removing or limiting their drawbacks. The approach is feasible and applicable in practice and was verified in an application to a 2 DOF manipulator. However, the method proposal does not address control of time varying systems.

It is inferred that the SRAC is very useful in situations where working conditions of systems often change as in the robot trajectory control, the force control in compliant motion tasks and payload uncertainties.

#### REFERENCES

- R.H. Middleton and G.C. Goodwin, "Adaptive computed torque control for rigid link manipulators", *Systems and Control Letters*, vol.10, issue 1, 1988, pp.9-16.
- [2] N. Sadegh and R. Horowitz, "Stability and robustness analysis of a class of adaptive controllers for robot", *Int. Journal of Robotics Research*, vol.9, issue 3, 1990, pp. 74-92.
- [3] J.E. Slotine, and W. Li, *Applied Nonlinear Control*, Prentice Hall Inc., USA; 1991.
- [4] K.S. Kumpati, O. Romeo, and D. Peter, Advanced Adaptive control, IEEE Press, USA, 1991.
- [5] J.S. Reed and P.A. Ioannou, "Instability analysis and robust adaptive control of robotic manipulators", *IEEE Transactions on Robotics and Automation*, vol.5, issue 3, 1989, pp.381-386.
- [6] K. S. Narendra, A. M. Annaswamy, *Stable Adaptive Systems*, Prentice-Hall International, Inc, 1989.
- [7] J. Somlo, B. Lantos, and P.T. Cat, Advanced Robot Control, Akademiai Kiado, Budapest, 1997.
- [8] K.D.Young, Variable Structure Control for Robotics and Aerospace Applications, Elsevier, Netherlands, 1993.
- [9] R. Marino and P. Tomei, Nonlinear Control Design: Geometric, Adaptive and Robust, Prentice-Hall, London, New York, 1995.
- [10] Doyle, John, and *et al.*, "State-space solutions to standard H2 and H infinity control problems", *IEEE Transactions on Automatic Control*, vol. 34, issue 8, 1989, pp. 831-847.
- [11] Boyd and *et al.*, "History of linear matrix inequalities in control theory", *Proceedings of the American Control Conference*, vol.1, 1994, pp. 31-34.
- [12] Bin Yao and M. Tomizuka, "Comparative experiments of robust and adaptive control with new robust adaptive controllers for robot manipulators", *In Proc. of IEEE Conf. on Decision and Control*, Florida, vol. 2, 1994, pp. 1290-1295.
- [13] B. Siciliano and L. Sciavicco, Modelling and Control of Robot Manipulators, 2nd ed., Springer Verlag, UK, 2000, pp. 139-148.
- [14] Carlos A. Canudas de Wit, B. Siciliano, and et al., Theory of Robot Control (Communications and Control Engineering Series), Springer Verlag, 1996, pp. 101-103.
- [15] S. Sastry, Adaptive Control: Stability, Convergence and Robustness, Prentice Hall, Inc., Englewood Clis, NJ 07632, USA, 1989.
- [16] P.A. Ioannou, Jing Sun, *Robust Adaptive Control*, Prentice Hall, 1995.