We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

4,900
Open access books available

124,000
International authors and editors

140M
Downloads

Our authors are among the

154
Countries delivered to

TOP 1%
most cited scientists

12.2%
Contributors from top 500 universities

WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.
For more information visit www.intechopen.com

9

A Robust Reinforcement Learning System Using Concept of Sliding Mode Control for Unknown Nonlinear Dynamical System

Masanao Obayashi, Norihiro Nakahara, Katsumi Yamada,
Takashi Kuremoto, Kunikazu Kobayashi and Liangbing Feng
Yamaguchi University
Japan

1. Introduction

In this chapter, a novel control method using a reinforcement learning (RL) (Sutton and Barto (1998)) with concept of sliding mode control (SMC) (Slotine and Li (1991)) for unknown dynamical system is considered.

In designing the control system for unknown dynamical system, there are three approaches. The first one is the conventional model-based controller design, such as optimal control and robust control, each of which is mathematically elegant, however both controller design procedures present a major disadvantage posed by the requirement of the knowledge of the system dynamics to identify and model it. In such cases, it is usually difficult to model the unknown system, especially, the nonlinear dynamical complex system, to make matters worse, almost all real systems are such cases.

The second one is the way to use only the soft-computing, such as neural networks, fuzzy systems, evolutionary systems with learning and so on. However, in these cases it is well known that modeling and identification procedures for the dynamics of the given uncertain nonlinear system and controller design procedures often become time consuming iterative approaches during parameter identification and model validation at each step of the iteration, and in addition, the control system designed through such troubles does not guarantee the stability of the system.

The last one is the way to use the method combining the above the soft-computing method with the model-based control theory, such as optimal control, sliding mode control (SMC), H_∞ control and so on. The control systems designed through such above control theories have some advantages, that is, the good nature which its adopted theory has originally, robustness, less required iterative learning number which is useful for fragile system controller design not allowed a lot of iterative procedure. This chapter concerns with the last one, that is, RL system, a kind of soft-computing method, supported with robust control theory, especially SMC for uncertain nonlinear systems.

RL has been extensively developed in the computational intelligence and machine learning societies, generally to find optimal control policies for Markovian systems with discrete state and action space. RL-based solutions to the continuous-time optimal control problem have been given in Doya (Doya (2000). The main advantage of using RL for solving optimal

control problems comes from the fact that a number of RL algorithms, e.g. Q-learning (Watkins et al. (1992)) and actor-critic learning (Wang et al. (2002)) and Obayashi et al. (2008)), do not require knowledge or identification/learning of the system dynamics. On the other hand, remarkable characteristics of SMC method are simplicity of its design method, good robustness and stability for deviation of control conditions.

Recently, a few researches as to robust reinforcement learning have been found, e.g., Morimoto et al. (2005) and Wang et al. (2002) which are designed to be robust for external disturbances by introducing the idea of H_∞ control theory (Zhau et al. (1996)), and our previous work (Obayashi et al. (2009)) is for deviations of the system parameters by introducing the idea of sliding mode control commonly used in model-based control. However, applying reinforcement learning to a real system has a serious problem, that is, many trials are required for learning to design the control system.

Firstly we introduce an actor-critic method, a kind of RL, to unite with SMC. Through the computer simulation for an inverted pendulum control without use of the inverted pendulum dynamics, it is clarified the combined method mentioned above enables to learn in less trial of learning than the only actor-critic method and has good robustness (Obayashi et al. (2009a)).

In applying the controller design, another problem exists, that is, incomplete observation problem of the state of the system. To solve this problem, some methods have been suggested, that is, the way to use observer theory (Luenberger (1984)), state variable filter theory (Hang (1976), Obayashi et al. 2009b) and both of the theories (Kung and Chen (2005)).

Secondly we introduce a robust reinforcement learning system using the concept of SMC, which uses neural network-type structure in an actor/critic configuration, refer to Fig. 1, to the case of the system state partly available by considering the variable state filter (Hang (1976)).

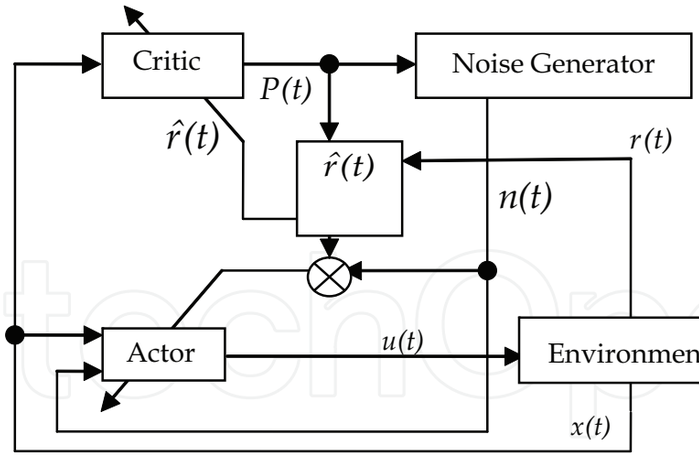


Fig. 1. The construction of the actor-critic system. (symbols in this figure are reffered to section 2)

The rest of this chapter is organized as follows. In Section 2, the conventional actor-critic reinforcement learing system is described. In Section 3, the controlled system, variable filter and sliding mode control are shortly explained. The proposed actor-critic reinforcement learning system with state variable filter using sliding mode control is described in Section 4. Comparison between the proposed system and the conventional system through simulation experiments is executed in Section 5. Finally, the conclusion is given in Section 6.

2. Actor-critic reinforcement learning system

Reinforcement learning (RL, Sutton and Barto (1998)), as experienced learning through trial and error, which is a learning algorithm based on calculation of reward and penalty given through mutual action between the agent and environment, and which is commonly executed in living things. The actor-critic method is one of representative reinforcement learning methods. We adopted it because of its flexibility to deal with both continuous and discrete state-action space environment. The structure of the actor-critic reinforcement learning system is shown in Fig. 1. The actor plays a role of a controller and the critic plays role of an evaluator in control field. Noise plays a part of roles to search the optimal action.

2.1 Structure and learning of critic

2.1.1 Structure of critic

The function of the critic is calculation of $P(t)$: the prediction value of sum of the discounted rewards *r(t)* that will be gotten over the future. Of course, if the value of $P(t)$ becomes bigger, the performance of the system becomes better. These are shortly explained as follows.

The sum of the discounted rewards that will be gotten over the future is defined as $V(t)$.

$$V(t) \equiv \sum_{l=0}^{\infty} \gamma^n \cdot r(t+l), \tag{1}$$

where γ ($0 \le \gamma < 1$) is a constant parameter called discount rate.

Equation (1) is rewritten as

$$V(t) = r(t) + \gamma V(t+1). \tag{2}$$

Here the prediction value of $V(t)$ is defined as $P(t)$. The prediction error $\hat{r}(t)$ is expressed as follows,

$$\hat{r}(t) = \hat{r}_t = r(t) + \gamma P(t+1) - P(t). \tag{3}$$

The parameters of the critic are adjusted to reduce this prediction error $\hat{r}(t)$. In our case the prediction value $P(t)$ is calculated as an output of a radial basis function neural network (RBFN) such as,

$$P(t) = \sum_{j=1}^{J} \omega_j^c y_j^c(t), \tag{4}$$

$$y_j^c(t) = \exp\left[-\sum_{i=1}^{n} (x_i(t) - c_{ij}^c)^2 / (\sigma_{ij}^c)^2 \right]. \tag{5}$$

Here, $y_j^c(t)$: jth node's output of the middle layer of the critic at time t, ω_j^c : the weight of jth output of the middle layer of the critic, x_i : i th state of the environment at time t, c_{ij}^c and σ_{ij}^c : center and dispersion in the i th input of j th basis function, respectively, J : the number of nodes in the middle layer of the critic, n : number of the states of the system (see Fig. 2).

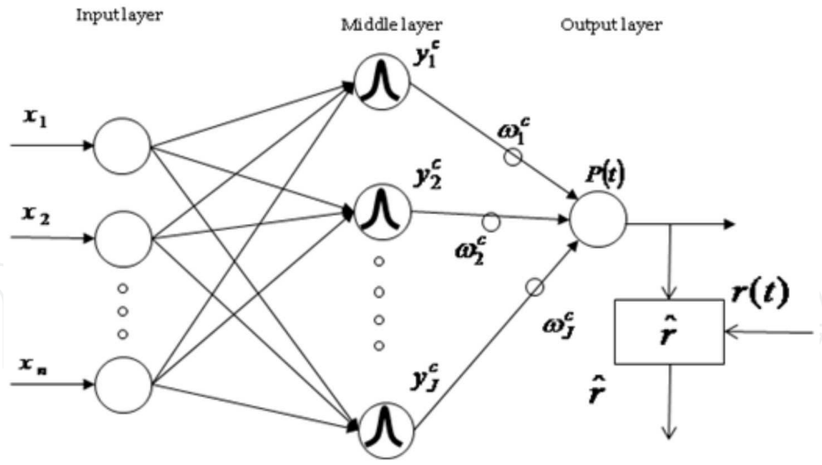


Fig. 2. Structure of the critic.

2.1.2 Learning of parameters of critic

Learning of parameters of the critic is done by back propagation method which makes prediction error $\hat{r}(t)$ go to zero. Updating rule of parameters are as follows,

$$\Delta\omega_i^c = -\eta_c \cdot \frac{\partial \hat{r}_t^2}{\partial \omega_i^c}, \quad (i = 1, \cdots, J). \tag{6}$$

Here η_c is a small positive value of learning coefficient.

2.2 Structure and learning of actor

2.2.1 Structure of actor

Figure 3 shows the structure of the actor. The actor plays the role of controller and outputs the control signal, action $a(t)$, to the environment. The actor basically also consists of radial basis function network. The jth basis function of the middle layer node of the actor is as follows,

$$y_j^a(t) = \exp\left[-\sum_{i=1}^{n}(x_i(t) - c_{ij}^a)^2 / (\sigma_{ij}^a)^2\right], \tag{7}$$

$$u'(t) = \sum_{j=1}^{J} \omega_j^a \cdot y_j^a(t), \tag{8}$$

$$u_1(t) = u_{\max} \cdot \frac{1 + \exp(-u'(t))}{1 - \exp(-u'(t))}, \tag{9}$$

$$u(t) = u_1(t) + n(t). \tag{10}$$

Here y_j^a : jth node's output of the middle layer of the actor, c_{ij}^a and σ_{ij}^a : center and dispersion in ith input of jth node basis function of the actor, respectively, ω_j^a : connection weight from jth node of the middle layer to the output, $u(t)$: control input, $n(t)$: additive noise.

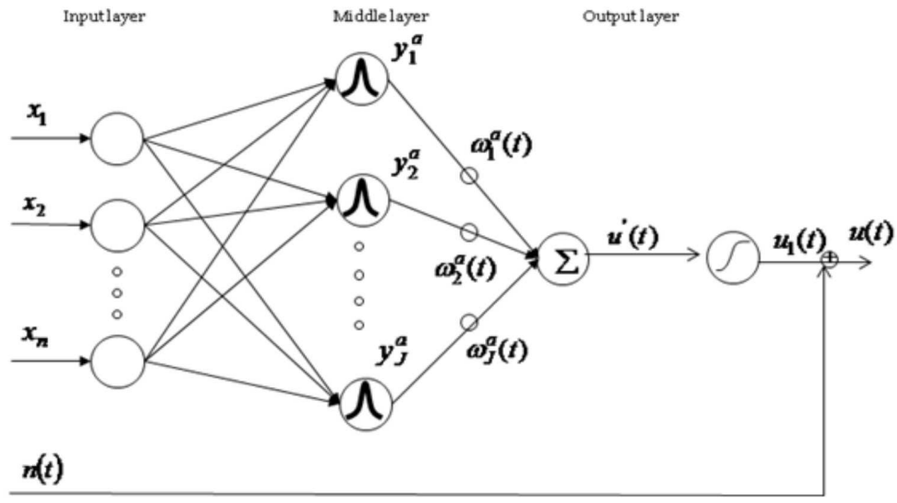


Fig. 3. Structure of the actor.

2.2.2 Noise generator

Noise generator let the output of the actor have the diversity by making use of the noise. It comes to realize the learning of the trial and error according to the results of performance of the system by executing the decided action. Generation of the noise $n(t)$ is as follows,

$$n(t) = n_t = noise_t \cdot \min\left(1, \exp(-P(t))\right), \tag{11}$$

where $noise_t$ is uniformly random number of $[-1, 1]$, min $(\cdot)$: minimum of $\cdot$. As the $P(t)$ will be bigger (this means that the action goes close to the optimal action), the noise will be smaller. This leads to the stable learning of the actor.

2.2.3 Learning of parameters of actor

Parameters of the actor, ω_j^a $(j = 1, \cdots, J)$, are adjusted by using the results of executing the output of the actor, i.e. the prediction error $\hat{r}_t$ and noise.

$$\Delta\omega_j^a = \eta_a \cdot n_t \cdot \hat{r}_t \cdot \frac{\partial u_1(t)}{\partial \omega_j^a}. \tag{12}$$

$\eta_a (> 0)$ is the learning coefficient. Equation (12) means that $(-n_t \cdot \hat{r}_t)$ is considered as an error, ω_j^a is adjusted as opposite to sign of $(-n_t \cdot \hat{r}_t)$. In other words, as a result of executing $u(t)$, e.g. if the sign of the additive noise is positive and the sign of the prediction error is positive, it means that positive additive noise is sucess, so the value of ω_j^a should be increased (see Eqs. (8)-(10)), and vice versa.

3. Controlled system, variable filter and sliding mode control

3.1 Controlled system

This paper deals with next *n*th order nonlinear differential equation.

$$x^{(n)} = f(\mathbf{x}) + b(\mathbf{x})u, \tag{13}$$

$$y = x\,, \tag{14}$$

where $\mathbf{x} = [x, \dot{x}, \cdots, x^{(n-1)}]^T$ is state vector of the system. In this paper, it is assumed that a part of states, $y(=x)$, is observable, u is control input, $f(\mathbf{x}), b(\mathbf{x})$ are unknown continuous functions.

Object of the control system: To decide control input u which leads the states of the system to their targets **x**. We define the error vector e as follows,

$$\begin{aligned} \mathbf{e} &= [e, \dot{e}, \cdots, e^{(n-1)}]^T\,, \\ &= [x - x_d, \dot{x} - \dot{x}_d, \cdots, x^{(n-1)} - x_d{}^{(n-1)}]^T. \end{aligned} \tag{15}$$

The estimate vector of e, $\hat{e}$, is available through the state variable filter (see Fig. 4).

3.2 State variable filter

Usually it is that not all the state of the system are available for measurement in the real system. In this work we only get the state x, that is, e, so we estimate the values of error vector **e**, i.e. $\hat{e}$, through the state variable filter, Eq. (16) (Hang (1976) (see Fig. 4).

$$\hat{e}_i = \frac{\omega_n \cdot p^i}{p^n + \omega_{n-1} p^{n-1} + \cdots + \omega_0}\, e\,, \quad (i = 0, \cdots, n-1) \tag{16}$$

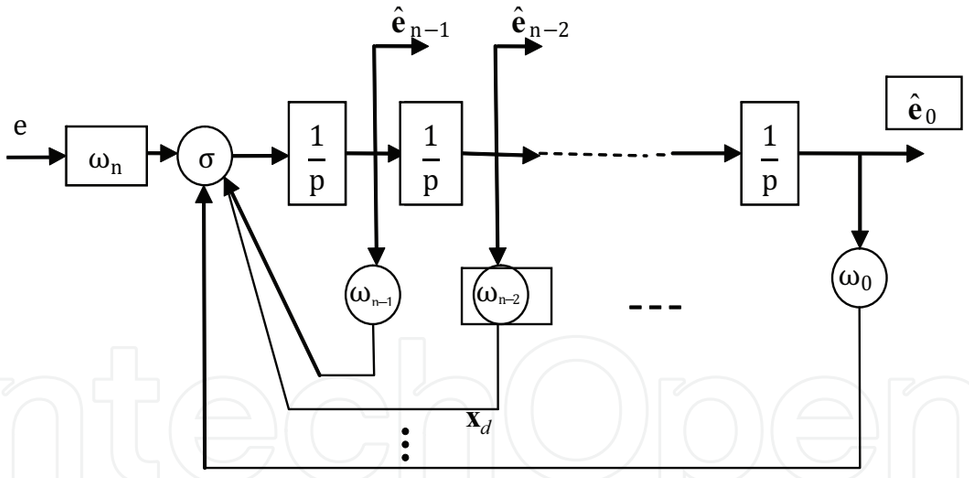


Fig. 4. Internal structure of the state variable filter.

3.3 Sliding mode control

Sliding mode control is described as follows. First it restricts states of the system to a sliding surface set up in the state space. Then it generates a sliding mode s (see in Eq. (18)) on the sliding surface, and then stabilizes the state of the system to a specified point in the state space. The feature of sliding mode control is good robustness.

Sliding time-varying surface H and sliding scalar variable s are defined as follows,

$$H : \{\, \mathbf{e} \mid s(\mathbf{e}) = 0 \,\}\,, \tag{17}$$

$$s(\mathbf{e}) = \boldsymbol{\alpha}^T \mathbf{e}\,, \tag{18}$$

where $\alpha_{n-1} = 1$ $\boldsymbol{\alpha} = [\alpha_0, \alpha_1, \cdots, \alpha_{n-1}]^T$, and $\alpha_{n-1}p^{n-1} + \alpha_{n-2}p^{n-2} + \cdots + \alpha_0$ is strictly stable in Hurwitz, p is Laplace transformation variable.

4. Actor-critic reinforcement learning system using sliding mode control with state variable filter

In this section, reinforcement learning system using sliding mode control with the state variable filter is explained. Target of this method is enhancing robustness which can not be obtained by conventional reinforcement. The method is almost same as the conventional actor-critic system except using the sliding variable s as the input to it inspite of the system states. In this section, we mainly explain the definition of the reward and the noise generation method.

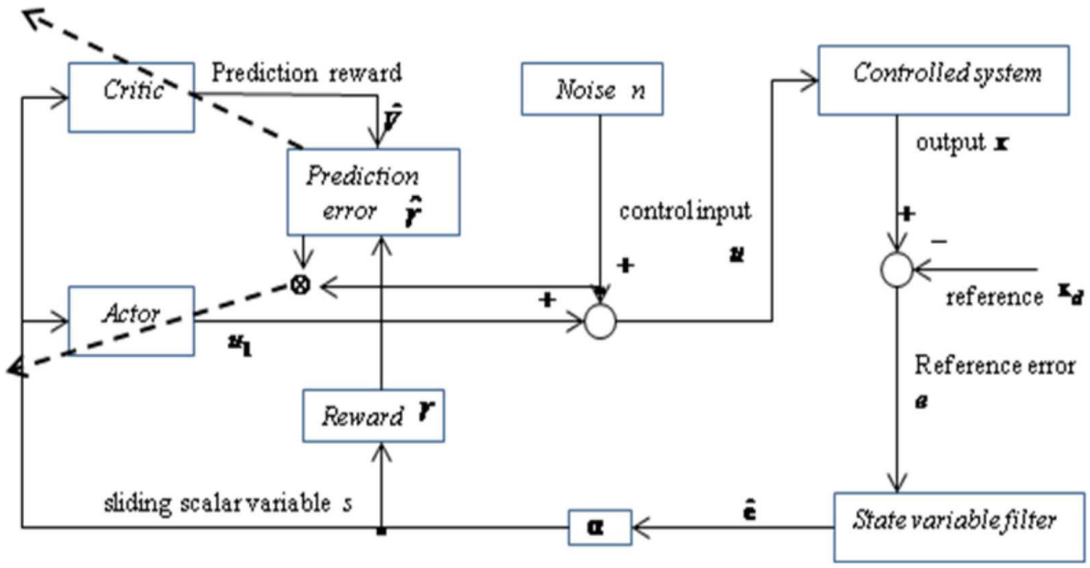


Fig. 5. Proposed reinforcement learning control system using sliding mode control with state variable filter.

4.1 Reward

We define the reward $r(t)$ to realize the sliding mode control as follows,

$$r(t) = \exp\{-s(t)^2\}\,, \tag{19}$$

here, from Eq. (18) if the actor-critic system learns so that the sliding variable s becomes smaller, i.e., error vector **e** would be close to zero, the reward $r(t)$ would be bigger.

4.2 Noise

Noise $n(t)$ is used to maintain diversity of search of the optimal input and to find the optimal input. The absolute value of sliding variable s is bigger, $n(t)$ is bigger, and that of s is smaller, it is smaller.

$$n(t) = z \cdot \bar{n} \cdot \exp\left(-\beta \cdot \frac{1}{s^2} \right), \tag{20}$$

where, z is uniform random number of range [-1, 1]. $\bar{n}$ is upper limit of the perturbation signal for searching the optimal input u. β is predefined positive constant for adjusting.

5. Computer simulation

5.1 Controlled object

To verify effectiveness of the proposed method, we carried out the control simulation using an inverted pendulum with dynamics described by Eq. (21) (see Fig. 6).

$$mg\ddot{\theta} = mgl\sin\theta - \mu_v\dot{\theta} + T_q\,. \tag{21}$$

Parameters in Eq. (21) are described in Table 1.

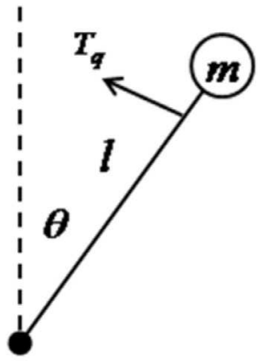


Fig. 6. An inverted pendulum used in the computer simulation.

θ	joint angle	-
m	mass	1.0 [kg]
l	length of the pendulum	1.0 [m]
g	gravity	9.8 [m/sec^2]
μ_V	coefficient of friction	0.02
T_q	input torque	-
$\mathbf{X} = [\theta, \dot{\theta}]$	observation vector	-

Table 1. Parameters of the system used in the computer simulation.

5.2 Simulation procedure

Simulation algorithm is as follows,

Step 1. Initial control input T_{q0} is given to the system through Eq. (21).

Step 2. Observe the state of the system. If the end condition is satisfied, then one trial ends, otherwise, go to Step 3.

Step 3. Calculate the error vector $\mathbf{e}$, Eq. (15). If only $y(=x)$, i.e., e is available, calculate $\hat{\mathbf{e}}$, the estimate value of through the state variable filters, Eq. (16).

Step 4. Calculate the sliding variable s, Eq. (18).
Step 5. Calculate the reward r by Eq. (19).
Step 6. Calculate the prediction reward $P(t)$ and the control input $u(t)$, i.e., torque T_q by Eqs. (4) and (10), respectively.
Step 7. Renew the parameters ω_i^c, ω_j^a of the actor and the critic by Eqs. (6) and (12).
Step 8. Set T_q in Eq. (21) of the system. Go to Step 2.

5.3 Simulation conditions

One trial means that control starts at $(\theta_0, \dot{\theta}_0) = (\pi/18[rad], 0\,[rad/\sec])$ and continues the system control for 20[sec], and sampling time is 0.02[sec]. The trial ends if $|\theta| \geq \pi/4$ or controlling time is over 20[sec]. We set upper limit for output u_1 of the actor. Trial success means that θ is in range $[-\pi/360, \pi/360]$ for last 10[sec]. The number of nodes of the hidden layer of the critic and the actor are set to 15 by trial and error (see Figs. (2)–(3)). The parameters used in this simulation are shown in Table 2.

α_0 : sliding variable parameter in Eq. (18)	5.0
η_c : learning coefficient of the actor in Eqs. (6)-(A6)	0.1
η_a : learning coefficient of the critic in Eqs. (12)-A(7)	0.1
$U_{\max}$: Maximun value of the Torque in Eqs. (9)-(A3)	20
γ : forgetting rate in Eq. (3)	0.9

Table 2. Parameters used in the simulation for the proposed system.

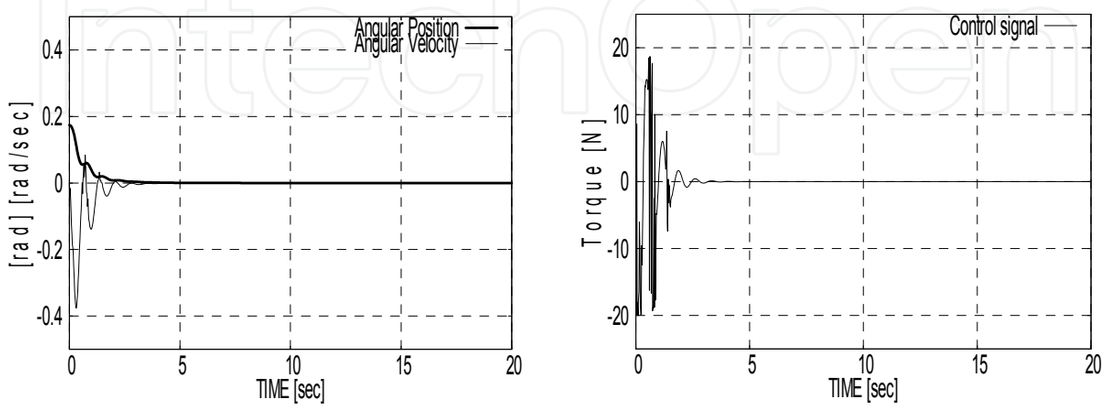
5.4 Simulation results

Using subsection 5.2, simulation procedure, subsection 5.3, simulation conditions, and the proposed method mentioned before, the control simulation of the inverted pendulum Eq. (21) are carried out.

5.4.1 Results of the proposed method

a. The case of complete observation

The results of the proposed method in the case of complete observation, that is, $\theta, \dot{\theta}$ are available, are shown in Fig. 7.



(a) $\theta, \dot{\theta}$ (b) Torque T_q

Fig. 7. Result of the proposed method in the case of complete observation ($\theta, \dot{\theta}$).

b. The case of incomplete observation using the state variable filters .

In the case that only θ is available, we have to estimate $\dot{\theta}$ as $\hat{\dot{\theta}}$. Here, we realize it by use of the state variable filter (see Eqs. (22)-(23), Fig. 8). By trial and error, the parameters, $\omega_0, \omega_1, \omega_2$, of it are set to $\omega_0 = 100$, $\omega_1 = 10$, $\omega_2 = 50$. The results of the proposed method with state variable filter in the case of incomplete observation are shown in Fig. 9.

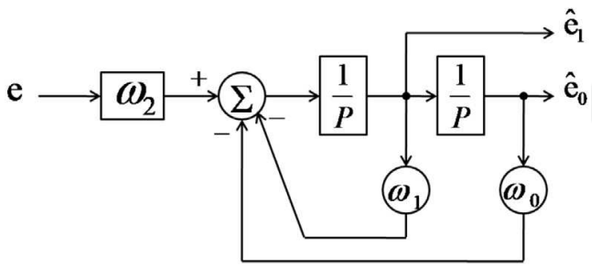
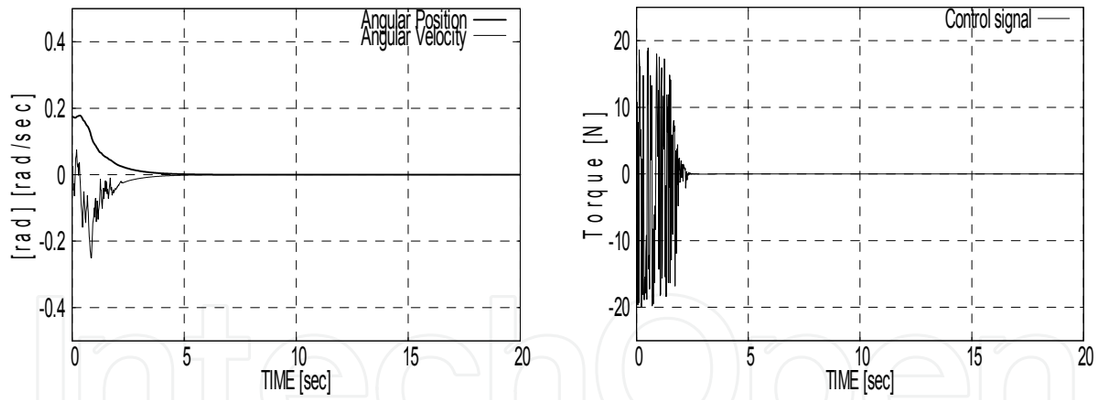


Fig. 8. State variable filter in the case of incomplete observation (θ).

$$\hat{e}_0 = \frac{\omega_2}{p^2 + \omega_1 p + \omega_0} e \tag{22}$$

$$\hat{e}_1 = \frac{\omega_2 p}{p^2 + \omega_1 p + \omega_0} e \tag{23}$$



(a) $\theta, \dot{\theta}$ (b) Torque T_q

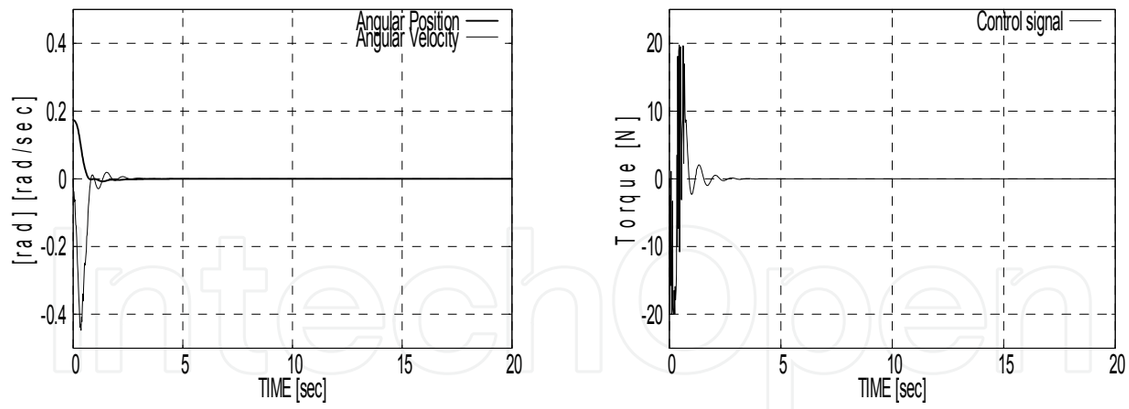
Fig. 9. Results of the proposed method with the state variable filter in the case of incomplete observation (only θ is available).

c. The case of incomplete observation using the difference method

Instead of the state variable filter in 5.4.1 *B*, to estimate the velocity angle, we adopt the commonly used difference method, like that,

$$\hat{\dot{\theta}}_t = \theta_t - \theta_{t-1} . \tag{24}$$

We construct the sliding variable s in Eq. (18) by using $\theta, \hat{\dot{\theta}}$. The results of the simulation of the proposed method are shown in Fig. 10.



(a) $\theta, \dot{\theta}$ (b) Torque T_q

Fig. 10. Result of the proposed method using the difference method in the case of incomplete observation (only θ is available).

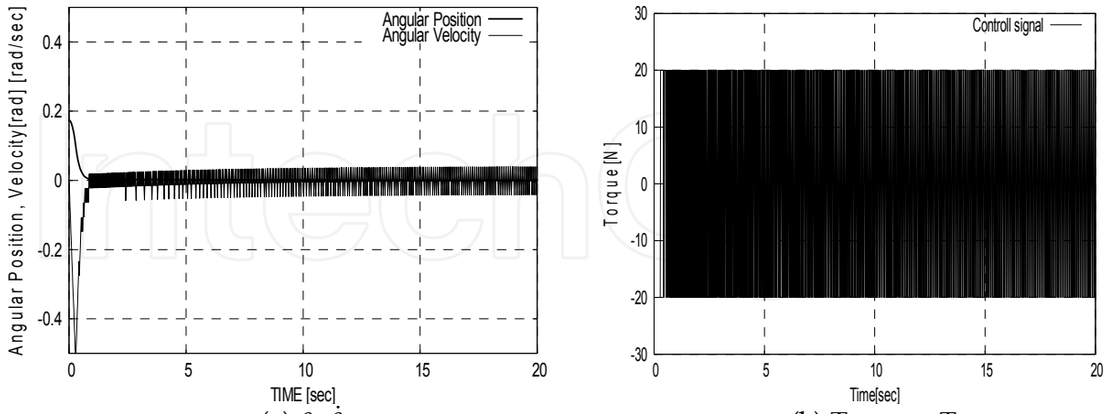
5.4.2 Results of the conventional method.

d. Sliding mode control method

The control input is given as follows,

$$u(t) = \begin{cases} U_{\max}, & if \ \theta \cdot \sigma > 0 \\ -U_{\max}, & if \ \theta \cdot \sigma \le 0 \end{cases}$$
$$\sigma = c\theta + \dot{\theta} \tag{25}$$
$$U_{\max} = 20.0 \ [\mathrm{N}]$$

Result of the control is shown in Fig. 11. In this case, angular, velocity angular, and Torque are all oscillatory because of the bang-bang control.



(a) $\theta, \dot{\theta}$ (b) Torque T_q

Fig. 11. Result of the conventional (SMC) method in the case of complete observation ($\theta, \dot{\theta}$).

e. Conventional actor-critic method

The structure of the actor of the conventional actor-critic control method is shown in Fig. 12. The detail of the conventional actor-critic method is explained in Appendix. Results of the simulation are shown in Fig. 13.

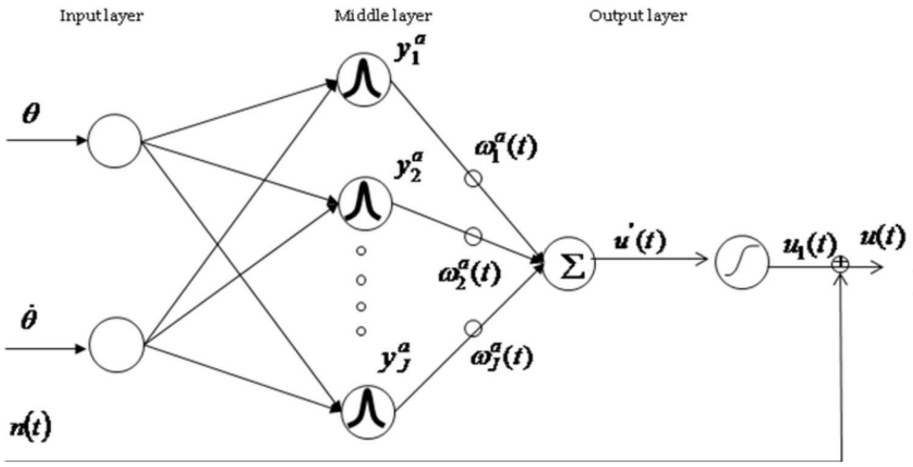
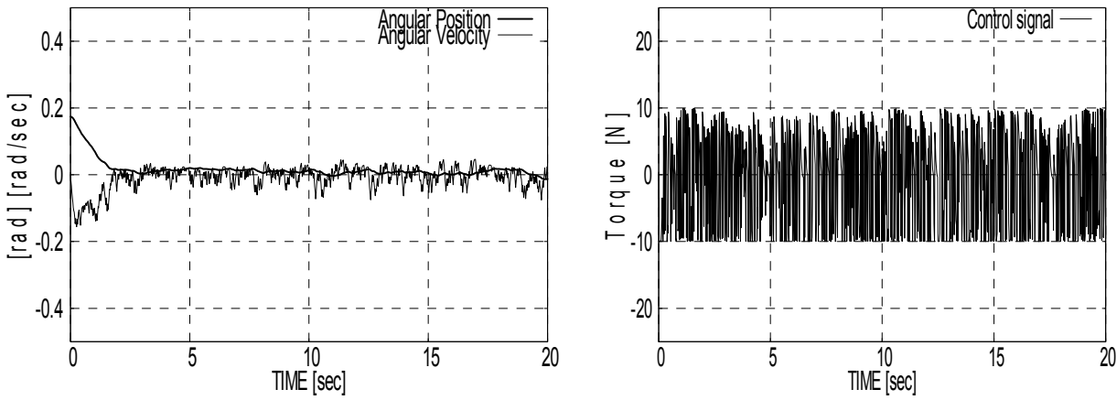
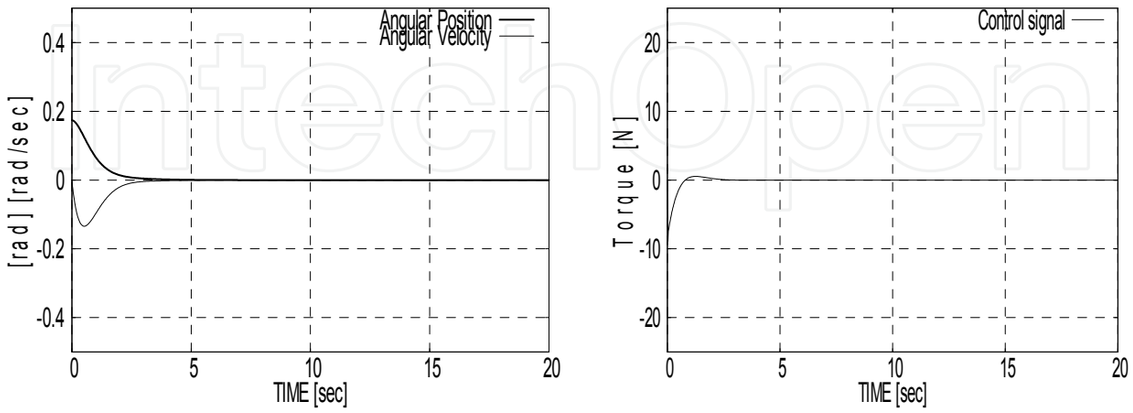


Fig. 12. Structure of the actor of the conventional actor-critic control method.



(a) $\theta, \dot{\theta}$ (b) Torque T_q

Fig. 13. Result of the conventional (actor-critic) method in the case of complete observation $(\theta, \dot{\theta})$.



(a) $\theta, \dot{\theta}$ (b) Torque T_q

Fig. 14. Result of the conventional PID control method in the case of complete observation $(\theta, \dot{\theta})$.

f. Conventional PID control method

The control signal $u(t)$ in the PID control is

$$u(t) = -K_p e(t) - K_I \int_0^t e(t) \cdot dt - K_d \cdot \dot{e}(t) , \tag{26}$$

here, $K_p = 45, K_I = 1, K_d \cdot = 10$. Fig. 14 shows the results of the PID control.

5.4.3 Discussion

Table 3 shows the control performance, i.e. average error of $\theta, \dot{\theta}$, through the controlling time when final learning for all the methods the simulations have been done. Comparing the proposed method with the conventional actor-critic method, the proposed method is better than the conventional one. This means that the performance of the conventional actor-critic method hass been improved by making use of the concept of sliding mode control.

Kinds of Average error	Proposed method			Conventional method		
	Actor-Critic + SMC			SMC	PID	Actor-Critic
	Complete observation	Incomplete Observation (θ : available)		Complete observation		
		S.v.f.	Difference			
$\int \theta dt / t$	0.3002	0.6021	0.1893	0.2074	0.4350	0.8474
$\int \dot{\theta} dt / t$	0.4774	0.4734	0.4835	1.4768	0.4350	1.2396

Table 3. Control performance when final learning (S.v.f. : state variable filter, Difference: Difference method).

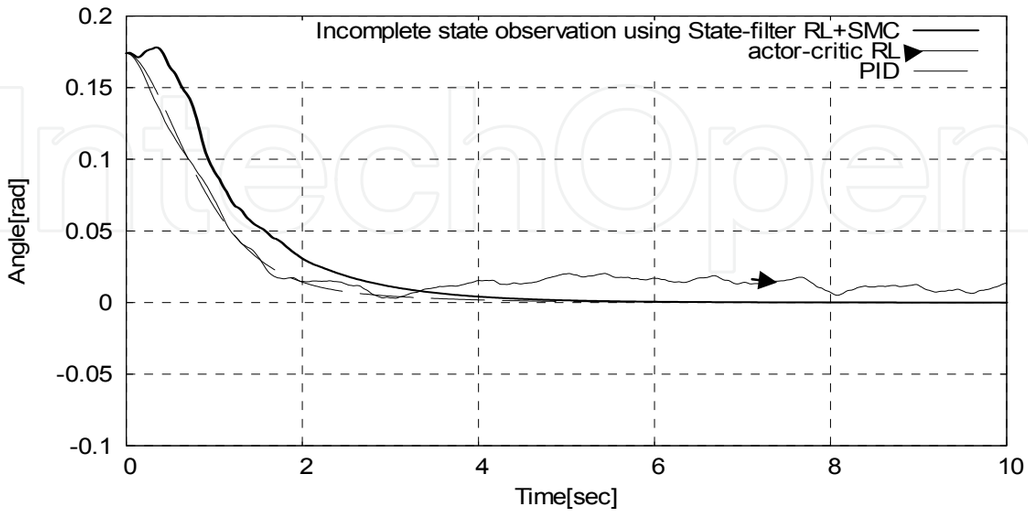


Fig. 15. Comparison of the porposed method with incomplete observation, the conventional actor-critic method and PID method for the angle, θ .

Figure 15 shows the comparison of the porposed method with incomplete observation, the conventional actor-critic method and PID method for the angle, θ. In this figure, the proposed method and PID method converge to zero smoothly, however the conventional actor-critic method does not converge. The comparison of the proposed method with PID control, the latter method converges quickly. These results are corresponding to Fig.16, i.e. the torque of the PID method converges first, the next one is the proposed method, and the conventional one does not converge.

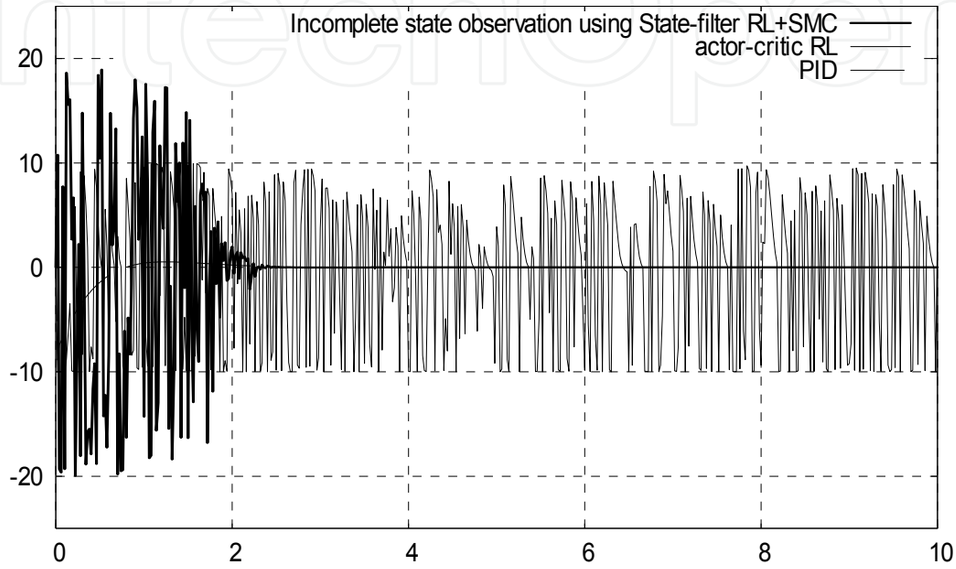


Fig. 16. Comparison of the porposed method with incomplete observation, the conventional actor-critic method and PID method for the Torque, T_q .

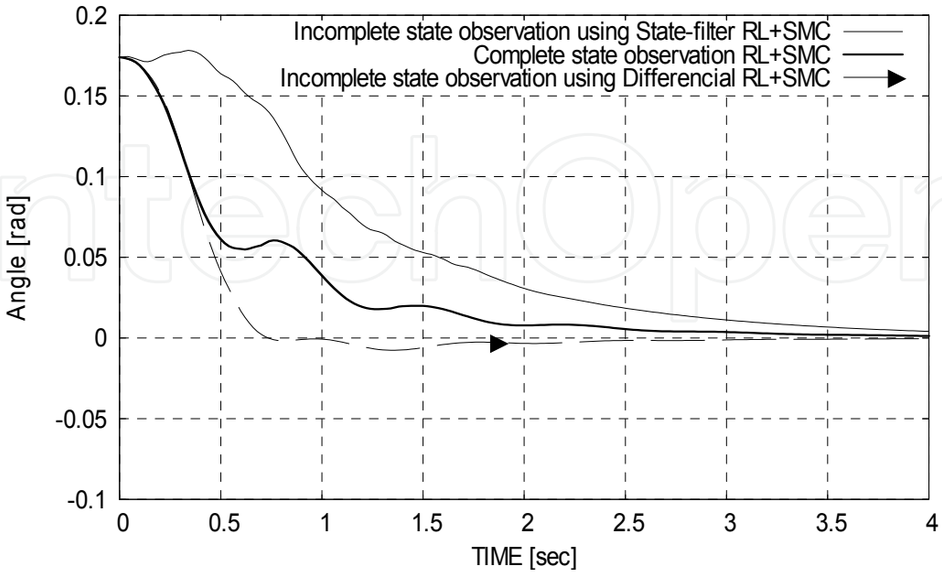


Fig. 17. The comparison of the porposed method among the case of the complete observation, the case with the state variable filter, and with the difference method for the angle, θ .

Fig. 17 shows the comparison of the porposed method among the case of the complete observation, the case with the state variable filter, and with the difference method for the angle, θ. Among them, the incomplete state observation with the difference method is best of three, especially, better than the complete observation. This reason can be explained by Fig. 18. That is, the value of s of the case of the difference method is bigger than that of the observation of the velocity angle, this causes that the input gain becomes bigger and the convergence speed has been accelerated.

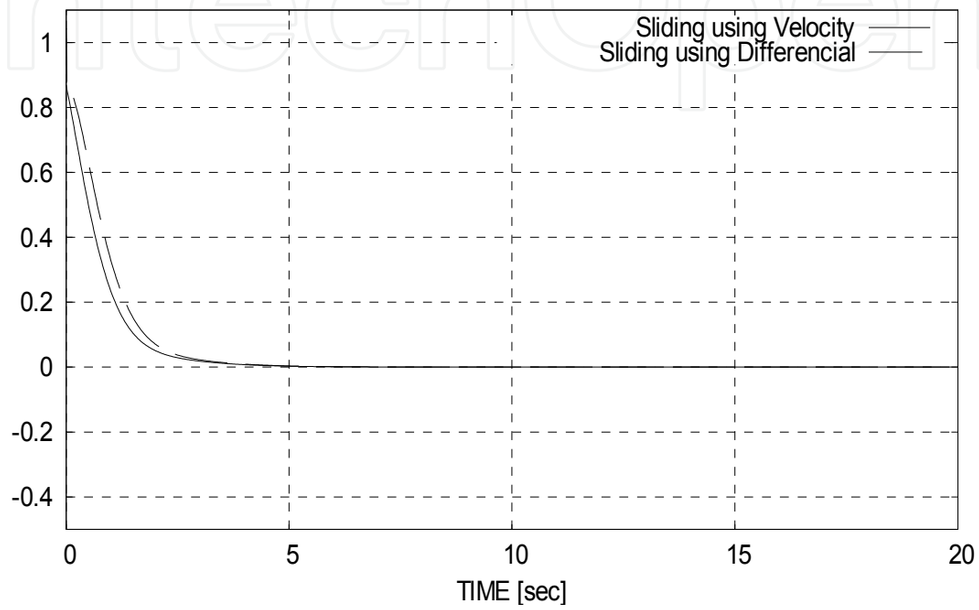


Fig. 18. The values of the sliding variable s for using the velocity and the difference between the angle and 1 sampling past angle.

5.4.4 Verification of the robust performance of each method

At first, as above mentioned, each controller was designed at $m = 1.0\,[\mathrm{kg}]$ in Eq. (21). Next we examined the range of m in which the inverted pendulum control is success. Success is defined as the case that if $|\theta| \le \pi / 45$ through the last 1[sec]. Results of the robust performance for change of m are shown in Table 4. As to upper/lower limit of m for success, the proposed method is better than the conventional actor-critic method not only for gradually changing *m* smaller from 1.0 to 0.001, but also for changing *m* bigger from 1.0 to 2.377. However, the best one is the conventional SMC method, next one is the PID control method.

6. Conclusion

A robust reinforcement learning method using the concept of the sliding mode control was mainly explained. Through the inverted pendulum control simulation, it was verified that the robust reinforcement learning method using the concept of the sliding mode control has good performance and robustness comparing with the conventional actor-critic method, because of the making use of the ability of the SMC method.

The way to improve the control performance and to clarify the stability of the proposed method theoretically has been remained.

	Proposed method		Conventional method		
	Actor-Critic + SMC		SMC	PID	Actor-Critic
	Complete observation	Incomplete observ. + s.v.f.*	Complete observation	Complete observation	Complete observation
m-max [kg]	2.081	2.377	11.788	4.806	1.668
m-min [kg]	0.001	0.001	0.002	0.003	0.021

*(s.v.f.: state variable filter)

Table 4. Robust control performance for change of *m* in Eq. (21).

7. Acknowledgement

This work has been supported by Japan JSPS-KAKENHI (No.20500207 and No.20500277).

8. Appendix

The structure of the critic of the conventional actor-critic control method is shown in Fig. 2. The number of nodes of the hidden layer of it is 15 as same as that of the proposed method. The prediction reward, $P(t)$, is as follow,

$$P(t) = \sum_{i=1}^{J} \omega_i^c \cdot \exp\left\{ \frac{-(\theta - c_{\theta i}^c)^2}{(\sigma_{\theta i}^c)^2} + \frac{-(\theta - c_{\dot{\theta} i}^c)^2}{(\sigma_{\dot{\theta} i}^c)^2} \right\}, \tag{A1}$$

The structure of actor is also similar with critic shown in Fig. 11. The output of the actor, $u'(t)$, and the control input, $u(t)$, are as follows, respectively,

$$u'(t) = \sum_{i=1}^{J} \omega_i^a \cdot \exp\left\{ \frac{-(\theta - c_{\theta i}^a)^2}{(\sigma_{\theta i}^a)^2} + \frac{-(\theta - c_{\dot{\theta} i}^a)^2}{(\sigma_{\dot{\theta} i}^a)^2} \right\}, \tag{A2}$$

$$u_1(t) = u_{\max} \cdot \frac{1 + \exp(-u'(t))}{1 - \exp(-u'(t))}, \tag{A3}$$

$$u(t) = u_1(t) + n(t). \tag{A4}$$

The center, $c_{\theta i}^c, c_{\dot{\theta} i}^c, c_{\theta i}^a, c_{\dot{\theta} i}^a$ of the critic and actor of the RBF network are set to equivalent distance in the range of $-3 < c < 3$. The variance, $\sigma_{\theta i}^c, \sigma_{\dot{\theta} i}^c, \sigma_{\theta i}^a, \sigma_{\dot{\theta} i}^a$ of the critic and actor of

the RBF networks are set to be at equivalent distance in the range of $[0 < \sigma < 1]$. The values mentioned above, particularly, near the original are set to close. The reward $r(t)$ is set as Eq. (A5) in order it to maximize at $(\theta, \dot{\theta}) = (0,0)$,

$$r(t) = \exp\left(-\frac{(\theta_t)^2}{\sigma_1} - \frac{(\dot{\theta}_t)^2}{\sigma_2} \right). \tag{A5}$$

The learning of parameters of critic and actor are carried out through the back-propagation algorithm as Eqs. (A6)-(A7) . $(\eta_c, \eta_a > 0)$

$$\Delta\omega_i^c = -\eta_c \cdot \frac{\partial \hat{r}_t^2}{\partial \omega_i^c}, \quad (i = 1, \cdots, J), \tag{A6}$$

$$\Delta\omega_j^a = \eta_a \cdot n_t \cdot \hat{r}_t \cdot \frac{\partial u_1(t)}{\partial \omega_j^a}, \quad (j = 1, \cdots, J). \tag{A7}$$

9. References

K. Doya. (2000). ”Reinforcement learning in continuous time and space”, *Neural Computatio*n, 12(1), pp.219-245

C.C. Hang. (1976).”On state variable filters for adaptive system design”, *IEEEE Trans. Automatic Control*, Vo.21, No.6,874-876

C.C. Kung & T.H. Chen. (2005).“Observer-based indirect adaptive fuzzy sliding mode control with state variable filters for unknown nonlinear dynamical systems”, *Fuzzy Sets and Systems,* Vol.155, pp.292-308

D.G. Luenberger. (1984).”Linear and Nonlinear Programming”, *Addison-Wesley Publishing Company*, MA

J. Morimoto & K. Doya. (2005) ”Robust Reinforcement Learning”, *Neural Computation* 17,335-359

M. Obayashi & T. Kuremoto & K. Kobayashi. (2008).”A Self-Organized Fuzzy-Neuro Reinforcement Learning System for Continuous State Space for Autonomous Robots”, *Proc. of International Conference on Computational Intelligence for Modeling, Control and Automation (CIMCA 2008)*, 552–559

M. Obayashi & N. Nakahara & T. Kuremoto & K. Kobayashi. (2009a). “A Robust Reinforcement Learning Using Concept of Slide Mode Control”, The Journal of the Artificial Life and Robotics, Vol. 13, No. 2, pp.526-530

M. Obayashi & K. Yamada, T. & Kuremoto & K. Kobayashi. (2009b). “A Robust Reinforcement Learning Using Sliding Mode Control with State Variable Filters”, *Proceedings of International Automatic Control Conference (CACS 2009),CDROM*

J.J.E. Slotine & W. Li. (1991). “Applied Nonlinear Control ”, *Prentice-Hall,* Englewood Cliffs, NJ

R.S. Sutton & A.G. Barto. (1998). ” Reinforcement Learning: An Introduction”, *The MIT Press*.

W.Y. Wang & M.L. Chan & C.C. James & T.T. Lee. (2002). "H_∞ Tracking-Based Sliding Mode Control for Uncertain Nonlinear Systems via an Adaptive Fuzzy-Neural Approach", *IEEE Trans. on Systems, Man, and Cybernetics,* Vol.32, No.4, August, pp.483-492

X. S. Wang & Y. H. Cheng & J. Q. Yi. (2007). "A fuzzy Actor–Critic reinforcement learning network", *Information Sciences*, 177, pp.3764-3781

C. Watkins & P. Dayan. (1992)."Q-learning," *Machine learning*, Vol.8, pp.279-292

K. Zhau & J.C.Doyle & K.Glover. (1996). "Robust optimal control", Englewood Cliffs NJ, Prentice Hall

IntechOpen

Robust Control, Theory and Applications

Edited by Prof. Andrzej Bartoszewicz

ISBN 978-953-307-229-6
Hard cover, 678 pages
Publisher InTech
Published online 11, April, 2011
Published in print edition April, 2011

The main objective of this monograph is to present a broad range of well worked out, recent theoretical and application studies in the field of robust control system analysis and design. The contributions presented here include but are not limited to robust PID, H-infinity, sliding mode, fault tolerant, fuzzy and QFT based control systems. They advance the current progress in the field, and motivate and encourage new ideas and solutions in the robust control area.

How to reference

In order to correctly reference this scholarly work, feel free to copy and paste the following:

Masanao Obayashi, Norihiro Nakahara, Katsumi Yamada, Takashi Kuremoto, Kunikazu Kobayashi and Liangbing Feng (2011). A Robust Reinforcement Learning System Using Concept of Sliding Mode Control for Unknown Nonlinear Dynamical System, Robust Control, Theory and Applications, Prof. Andrzej Bartoszewicz (Ed.), ISBN: 978-953-307-229-6, InTech, Available from: http://www.intechopen.com/books/robust-control-theory-and-applications/a-robust-reinforcement-learning-system-using-concept-of-sliding-mode-control-for-unknown-nonlinear-d

INTECH
open science | open minds

InTech Europe
University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China
Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821

© 2011 The Author(s). Licensee IntechOpen. This chapter is distributed under the terms of the Creative Commons Attribution-NonCommercial-ShareAlike-3.0 License, which permits use, distribution and reproduction for non-commercial purposes, provided the original is properly cited and derivative works building on this content are distributed under the same license.

IntechOpen

IntechOpen