

Article

Self-Learning Variable Structure Control for a Class of Sensor-Actuator Systems

Sanfeng Chen ^{1,†}, Shuai Li ^{2,†}, Bo Liu ^{3,*}, Yuesheng Lou ⁴ and Yongsheng Liang ¹

¹ Key Lab of Visual Media Processing and Transmission, Shenzhen Institute of Information Technology, Shenzhen 518029, Guangdong, China; E-Mails: chensanf@szit.com.cn (S.C.); liangys@szit.com.cn (Y.L.)

² Department of Electrical and Computer Engineering, Stevens Institute of Technology, Hoboken, NJ 07030, USA; E-Mail: sam.shuai.li@gmail.com

³ Department of Computer Science, University of Massachusetts, Amherst, MA 01003, USA

⁴ School of Mechatronics and Information, Yiwu Industrial and Commercial College, Yiwu 322000, Zhejiang, China; E-Mail: lusion@mail.ustc.edu.cn

[†] These authors contributed equally to this work.

* Author to whom correspondence should be addressed; E-Mail: boliu@cs.umass.edu; Tel.: +1-551-333-9638; Fax: +1-413-362-5733.

Received: 4 April 2012; in revised form: 16 April 2012 / Accepted: 29 April 2012 /

Published: 10 May 2012

Abstract: Variable structure strategy is widely used for the control of sensor-actuator systems modeled by Euler-Lagrange equations. However, accurate knowledge on the model structure and model parameters are often required for the control design. In this paper, we consider model-free variable structure control of a class of sensor-actuator systems, where only the online input and output of the system are available while the mathematic model of the system is unknown. The problem is formulated from an optimal control perspective and the implicit form of the control law are analytically obtained by using the principle of optimality. The control law and the optimal cost function are explicitly solved iteratively. Simulations demonstrate the effectiveness and the efficiency of the proposed method.

Keywords: sensor-actuator system; principle of optimality; Bellman equation; variable structure control; self-learning

1. Introduction

With the development of mechatronics, automatic systems consisting of sensors for perception and actuators for action are more and more widely used in applications [1–4]. Besides the proper choices of sensors and actuators and an elaborate fabrication of mechanical structures, the control law design also plays a crucial role in the implementation of automatic systems especially for those with complicated dynamics. For most mechanical sensor-actuator systems, it is possible to model them in Euler-lagrange equations [4,5]. In this paper, we are concerned with the sensor-actuator systems modeled by Euler-lagrange equations.

Due to the importance of Euler-lagrange equations in modeling many real sensor-actuator systems, much attention has been paid to the control of such kind systems. According to the type of constraints, the Euler-lagrange system can be categorized into Euler-lagrange system without nonholonomic constraints (e.g., fully-actuated manipulator [6,7], omni-directional mobile robot [8]), and the system subject to nonholonomic constraint [9] (e.g., the cart-pole system [10], the under-actuated multiple body system [11]). For Euler-lagrange system without nonholonomic constraints, the dimension of inputs are often equal to the dimension of output and the system are often able to be transformed into a double integrator system by employing feedback linearization [12]. Other methods, such as control Lyapunov function method [13], passivity based method [14], optimal control method [15], *etc.*, are also successfully applied to the control of Euler-lagrange system without nonholonomic constraints. In contrast, as the dimension of inputs is lower than that of outputs, it is often impossible to directly transform the Euler-lagrange system subject to nonholonomic constraints to a linear system and thus feedback linearization fails to stabilize the system. To tackle the difficulty, variable structure control based method [16], backstepping based control [17], optimal control based method [18], discontinuous control method [19], *etc.*, are widely investigated and some useful design procedures are proposed. However, due to the inherent nonlinearity and nonholonomic constraints, most existing methods [16–19] are strongly model dependent and the performance are very sensitive to model errors. Inspired by the success of human operators for the control of Euler-lagrange systems, various intelligent control strategies, such as fuzzy logic [20], neural networks [21], evolutionary algorithms [22], to name a few of them, are proposed to solve the control problem of Euler-lagrange systems subject to nonholonomic constraints. As demonstrated by extensive simulations, these type of strategies are indeed effective to the control of Euler-lagrange systems subject to nonholonomic constraints. However, rigorous proof on the stability are difficult for this type of methods and there may exist some initializations of the state, from which the system cannot be stabilized.

In this paper, we propose a self-learning control method applicable to Euler-lagrange systems. In contrast to existing work on intelligent control of Euler-lagrange systems, the stability of the close loop system with the proposed method is proven in theory. On the other hand, different from model based design strategies, such as backstepping based design [17], variable structure based design [16], *etc.*, the proposed method does not require information of the model parameters and therefore is a model independent method. We formulate the problem from an optimal control perspective. In this framework, the goal is to find the input sequence to minimize the cost function defined on infinite horizon under the constraint of the system dynamics. The solution can be found by solving a Bellman equation according

to the principle of optimality [23]. Then an adaptive dynamic programming strategy [24–26] is utilized to numerically solve the input sequence in real time.

The remainder of this paper is organized as follows: in Section 2, preliminaries on Euler-lagrange systems and variable structure control are given briefly. In Section 3, the problem is formulated as a constrained optimization problem and the critic model and the action model are employed to approximate the optimal mappings. The control law is then derived in Section 4. In Section 5, simulations are given to show the effectiveness of the proposed method. The paper is concluded in Section 6.

2. Preliminaries on Variable Structure Control of the Sensor-Actuator System

In this paper, we are concerned with the following sensor-actuator system in the Euler-Lagrange form,

$$D(q)\ddot{q} + C(q, \dot{q})\dot{q} + \phi(q) = u \quad (1)$$

where $q \in \mathbb{R}^n$, $D(q) \in \mathbb{R}^{n \times n}$ is the inertial matrix, $C(q, \dot{q}) \in \mathbb{R}^{n \times n}$, $\phi(q) \in \mathbb{R}^n$ and $u \in \mathbb{R}^n$. Note that the inertial matrix $D(q)$ is symmetric and positive definite. There are three terms on the left side of the above equation. The first term involve the inertial force in the generalized coordinates, the second one models the Coriolis force and friction, the values of which depend on $\dot{q}$ and the third one is the conservative force, which is in correspondence to the potential energy. The control force u applied on the system drives the variation of the coordinate q . It is also noteworthy that we assume the dimension of u is equal to that of q here. This definition also admits the case for u with lower dimension than that of q by imposing constraints to u , e.g., the constraint $u = [u_1, u_2, \dots, u_n]$ with $u_1 = 0$ restricts u in a $n - 1$ dimensional space. Defining state variables $x_1 = q$ and $x_2 = \dot{q}$, the Euler-Lagrange Equation (1) can be put into the following state-space form:

$$\begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= -D^{-1}(x_1)(u + C(x_1, x_2)x_2 + \phi(x_1)) \end{aligned} \quad (2)$$

Note that the matrix $D(x_1)$ is invertible as it is positive definite. The control objective is to asymptotically stabilize the Euler-Lagrange system (2), i.e., design a mapping $(x_1, x_2) \rightarrow u$ such that $x_1 \rightarrow 0$ and $x_2 \rightarrow 0$ when time elapses.

As an effective design strategy, variable structure control finds applications in many different type of control systems including the Euler-Lagrange system. The method stabilizes the dynamics of a nonlinear system by steering the state to a elaborately designed sliding surface, on which the state inherently evolves towards the zero state. Particularly for the system (2), we define $s = s(x_1, x_2)$ as follows:

$$s = c_0 x_1 + x_2 \quad (3)$$

where $c_0 > 0$ is a constant. Note that $s = c_0 x_1 + x_2 = 0$ together with the dynamics of x_1 in Equation (2) gives the dynamics of x_1 as $\dot{x}_1 = -c_0 x_1$ for $c_0 > 0$. Clearly, x_1 asymptotically converges to zero. Also we know $x_2 = 0$ when $x_1 = 0$ according to $s = c_0 x_1 + x_2 = 0$. Therefore, we conclude the states x_1, x_2 on the sliding surface $s = 0$ for s defined in Equation (3) converge to zero with time. With this property of the sliding surface, a control law driving the states to $s = 0$ definitely grants the ultimate convergence to the zero states. Accordingly, the stabilization of the system can be realized

by controlling s to zero. To reach this goal, a positive definite control Lyapunov function $V(s)$, e.g., $V(s) = s^2$, is often used to design the control law. For stability consideration, the time derivative of $V(s)$ is required to be negative definite. In order to guarantee the negative definiteness of the time derivative of $V(s)$, exact information about the system dynamics (2) is often necessary, which results in the model based design strategies.

About the Euler-Lagrange Equation (1) for modeling sensor-actuator systems, we have the following remark:

Remark 1 *In this paper, we are concerned with the class of sensor-actuator systems modeled by the Euler-Lagrange Equation (1). Actually, the dynamics of mechanical systems can be described by the Euler-Lagrange equation according to the rigid body mechanics [4,5], which is essentially equivalent to Newton's laws of motion. Therefore, mechanical sensor-actuator system can be modeled by Equation (1). In this regard, the Euler-Lagrange equation employed in the paper models a general class of sensor-actuator systems.*

3. Problem Formulation

Without losing generality, we stabilize the system (1) by steering it to the sliding surface $s = 0$ with s defined in Equation (3). Different from existing model based design procedures, we design a self-learning controller, which does not require accurate knowledge about $D(q)$, $C(q, \dot{q})$ and $\phi(q)$ in Equation (1). In this section, we formulate such a control problem from the optimal control perspective.

In this paper, we set the origin as the desired operating point, i.e., we consider the problem of controlling the state of the system (1) to the origin. For the case with other desired operating points, the problem can be equivalently transformed to the one with the origin as the operating point by shifting the coordinates. At each sampling period, the norm of $s = c_0 x_1 + x_2$, which measures the distance from the desired sliding surface $s = 0$, can be used to evaluate the one step performance. Therefore, we define the following utility function associated with the one-step cost at the i th sampling period,

$$U_i = U(s) \quad (4)$$

with

$$U(s) = \begin{cases} 0 & |s_1| < \delta_1, |s_2| < \delta_2, \dots, |s_n| < \delta_n \\ 1 & \text{otherwise} \end{cases} \quad (5)$$

where s is defined in Equation (3) and $s = [s_1, s_2, \dots, s_n]^T$, $|s_i|$ denotes the absolute value of the i th component of the vector s , the parameter $\delta_i > 0$ for $i = 1, 2, \dots, n$. At each step, there is a value U_i and the total cost starting from the k th step along the infinite time horizon can be expressed as follows,

$$J_k = J(x(k), \bar{u}(k)) = \sum_{i=k}^{\infty} \gamma^{i-k} U_i \quad (6)$$

where $x(k)$ is the state vector of system (1) sampled at the k th step with $x(k) = [x_1^T(k), x_2^T(k)]^T$, γ is the discount factor with $0 < \gamma < 1$, $\bar{u}(k) = (u_k, u_{k+1}, \dots, u_{\infty})$ is the control sequence starting from the k th step. Note that for the deterministic system (1), the preceding states after the k th step are determined by $x(k)$ and the control sequence $\bar{u}_k$. Accordingly, J_k is a function of $x(k)$ and $\bar{u}(k)$ with

$J_k = J(x(k), \bar{u}(k))$. Also note that both the cost function J_k and the utility function U_k are defined based on the discrete samplings of the continuous system (1). Now, we can define the problem of controlling the sensor-actuator system (1) in this framework as follows,

$$\min_{u(0), u(1), \dots, u(\infty) \in \Omega} J_0 = \sum_{i=0}^{\infty} \gamma^i U_i \quad (7a)$$

subject to:

$$\begin{cases} \dot{x}_1(t) = x_2(t) \\ \dot{x}_2(t) = -D^{-1}(t)(x_1(t))(u(t) + C(x_1(t), x_2(t))x_2(t) + \phi(x_1(t))) \end{cases} \quad (7b)$$

$$u(t) = u(i) \quad \text{for } i\tau \leq t < (i+1)\tau \quad (7c)$$

where U_i is defined by Equations (4) and (5), $\tau > 0$ is the sampling period, the set Ω defines the feasible control actions, J_0 is the cost function for $k = 0$ in Equation (6). It is worth noting that J_0 is a function of $\bar{u}(0) = (u_0, u_1, \dots, u_\infty)$ and $x(0)$ according to Equation (6). The optimization in Equation (7) is relative to $\bar{u}(0)$ with a given initial state $x(0)$. Also note that in the optimization problem in Equation (7), the decision variable $u(0), u(1), \dots, u(\infty)$ are defined in every sampling period. The control action keeps the value in the duration of two consecutive sampling steps. This formulation is consistent with the real implementations of digital controllers.

Remark 2 *There are infinitely many decision variables, which are $u(0), u(1), \dots, u(\infty)$, in the optimization problem in Equation (7). Therefore, this is an infinite dimensional problem. It cannot be solved directly using numerical methods. Conventionally, such kind of problem is often solved by using a finite dimensional approximation [27]. In addition, note that the dynamic model of the system appears in the optimization problem in Equation (7) and it will also show up in the finite dimensional relaxation of the problem, which means the resulting solution requires model information and thus is also model-dependent. In contrast, in this paper we investigate the model-independent variable structure control of sensor-actuator systems on the infinite time horizon.*

4. Model-Free Control of the Euler-Lagrange System

In this section, we present the strategy to solve the constrained optimization problem efficiently without knowing the model information of the chaotic system. We first investigate the optimality condition of Equation (7) and present an iterative procedure to approach the analytical solution. Then, we analyze the convergence of the iterative procedure and the stability with the derived control strategy.

4.1. Optimality Condition

Denoting J^* the optimal value to the optimization problem in Equation (7), i.e.,

$$\begin{aligned} J^* = \min_{u(0), u(1), \dots, u(\infty) \in \Omega} J_0 \\ \text{subject to: (7b), (7c)} \end{aligned} \quad (8)$$

According to the principle of optimality [23], the solution of Equation (7) satisfy the following Bellman equation:

$$J^*(y) = \min_{u_k \in \Omega} (U_k + \gamma J^*(z)) \quad \forall x, \forall k = 0, 1, 2, \dots \quad (9)$$

where z is the solution of Equation (7b) at $t = k + 1$ with $x(k) = y$ and the control action $u(t) = u_k$ for $k\tau \leq t < (k + 1)\tau$. Without introducing confusion, we simply write Equation (9) as follows

$$J^* = \min(U_k + \gamma J^*) \quad (10)$$

Define the Bellman operator $\mathcal{B}$ relative to function $h(z)$ as follows

$$\mathcal{B}h(z) = \min(U_k + \gamma h(z)) \quad (11)$$

Then, the optimality condition in Equation (10) can be simplified into the following with the Bellman operator,

$$J^* = \mathcal{B}J^* \quad (12)$$

Note that the function U_k is implicitly included in the Bellman operator. The Equation (12) constitutes the optimality condition for problem in Equation (7). It is difficult to solve the explicit form of J^* analytically from Equation (9). However, it is possible to get the solution by iterations. We use the following iterations to solve J^* ,

$$\hat{J}(n+1) = \mathcal{B}\hat{J}(n) \quad (13)$$

subject to: (7b), (7c)

The control action keeps constant in the duration between the k th and the $k + 1$ th step, *i.e.*, $u^*(t) = u_k^*$ for $k\tau \leq t < (k + 1)\tau$. u_k^* can be obtained from Equation (9) based on Equation (13),

$$u_k^* = \operatorname{argmin}_{u_k \in \Omega} (U_k + \gamma J^*) \quad (14)$$

4.2. Approximating the Action Mapping and the Critic Mapping

In the previous sections, the iteration (13) is derived to calculate J^* and the optimization (14) is obtained to calculate the control law. The iteration to approach J^* and the optimization to derive u^* have to be run in every time step in order to obtain the most up-to-date values. Inspired by the learning strategies widely studied in artificial intelligence [26,28], a learning based strategy is used in this section to facilitate the processing. After a enough long time, the system is able to memorize the mapping of J^* and the mapping of u^* . After this learning period, there will be no need to repeat any iterations or optimal searching, which will make the strategy more practical.

Note that the optimal cost J^* is a function of the initial state. Counting the cost from the current time step, J^* can also be regarded as a function of both the current state and the optimal action at current time step according to Equation (10). Therefore, $\hat{J}(n)$, the approximation of J^* , can also be regarded as a function relative to the current state and the current optimal input. As to the optimal control action u^* , it is a function of the current state. Our goal in this section is to obtain the mapping from the current state and the current input to $\hat{J}(n)$ and the mapping from the current state to the optimal control action u^*

using parameterized models, denoted as the critic model and the action model, respectively. Therefore, we can write the critic model and the action model as $J_n(u_n^*, x_n, W_c)$ and $u_n^*(x_n, W_a)$ respectively, where W_c is the parameters of the critic model and W_a is the parameters of the action model.

In order to train the critic model with the desired input-output correspondence, we define the following error at time step $n + 1$ to evaluate the learning performance,

$$\begin{aligned} e_c(n+1) &= \mathcal{B}\hat{J}(n) - \hat{J}(n+1) \\ E_c(n+1) &= \frac{1}{2}e_c^2(n+1) \end{aligned} \quad (15)$$

Note that $\mathcal{B}\hat{J}(n)$ is the desired value of $\hat{J}(n+1)$ according to Equation (13). Using the back-propagation rule, we get the following rule for updating the weight W_c of the critic model,

$$\begin{aligned} W_c(n+1) &= W_c(n) + \delta W_c(n) \\ &= W_c(n) - l_c(n) \frac{\partial E_c(n)}{\partial W_c(n)} \\ &= W_c(n) - l_c(n) \frac{\partial E_c(n)}{\partial \hat{J}(n)} \frac{\partial \hat{J}(n)}{\partial W_c(n)} \end{aligned} \quad (16)$$

where $l_c(n)$ is the step size for the critic model at the time step n .

As to the action model, the optimal control u^* in Equation (14) is the one that minimizes the cost function. Note that the possible minimum cost is zero, which corresponds to the scenario with the state staying inside the desired bounded area. In this regard, we define the action error as follows,

$$\begin{aligned} e_a(n) &= \hat{J}_n \\ E_a(n) &= \frac{1}{2}e_a^2(n) \end{aligned} \quad (17)$$

Then, similar to the update rule of W_c for the critic model, we get the following update rule of W_a for the action model,

$$W_a(n+1) = W_a(n) - l_a(n) \frac{\partial E_a(n)}{\partial \hat{J}(n)} \frac{\partial \hat{J}(n)}{\partial u(n)} \frac{\partial u(n)}{\partial W_a(n)} \quad (18)$$

where $l_a(n)$ is the step size for the action model at the time step n .

Equations (16) and (18) update the critic model and the action model progressively. After W_c and W_a have learnt the model information by learning for a long enough time, their values can be fixed at the one obtained at the final step and no further learning is required any longer, which is in contrast to Equation (14) requiring to solve an optimization problem even after a long enough time.

5. Simulation Experiment

In this section, we consider the simulation implementation of the proposed control strategy. The dynamics given in Equation (1) model a wide class of sensor-actuator systems. Particularly, to demonstrate the effectiveness of the proposed self-learning variable structure method, we apply it to the stabilizations of a typical benchmark system: the cart-pole system.

The cart-pole system, as sketched in Figure 1, is a widely used testbed for the effectiveness of control strategies. The system is composed of a pendulum and a cart. The pendulum has its mass above its pivot point, which is mounted on a cart moving horizontally. In this part, we apply the proposed control method to the cart-pole system to test the effectiveness of our method.

5.1. The Model

The cart-pole model used in this work is the same as that in [29], which can be described as follows.

$$\ddot{\theta} = \frac{g \sin \theta + \cos \theta [-F - ml\dot{\theta}^2 \sin \theta + \mu_c \operatorname{sgn}(\dot{y})] - \frac{\mu_p \dot{\theta}}{ml}}{l(\frac{4}{3} - \frac{m \cos^2 \theta}{m_c + m})} \quad (19)$$

$$\ddot{y} = \frac{F + ml[\dot{\theta}^2 \sin \theta - \ddot{\theta} \cos \theta] - \mu_c \operatorname{sgn}(\dot{y})}{m_c + m} \quad (20)$$

where

$$\operatorname{sgn}(x) = \begin{cases} 1, & \text{if } x > 0 \\ 0, & \text{if } x = 0 \\ -1, & \text{if } x < 0 \end{cases} \quad (21)$$

with the following values of the parameters:

g : 9.8 m/s², acceleration due to gravity;

m_c : 1.0 kg, mass of cart;

m : 0.1 kg, mass of pole;

l : 0.5 meter, half-pole length;

μ_c : 0.0005, coefficient of friction of cart on track;

μ_p : 0.000002, coefficient of friction of pole on cart;

F : ± 10 Newtons, force applied to cart center of mass.

This system has four state variables: y is the position of the cart on track, θ is the angle of the pole with respect to the vertical position, and $\dot{y}$ and $\dot{\theta}$ are the cart velocity and angular velocity, respectively.

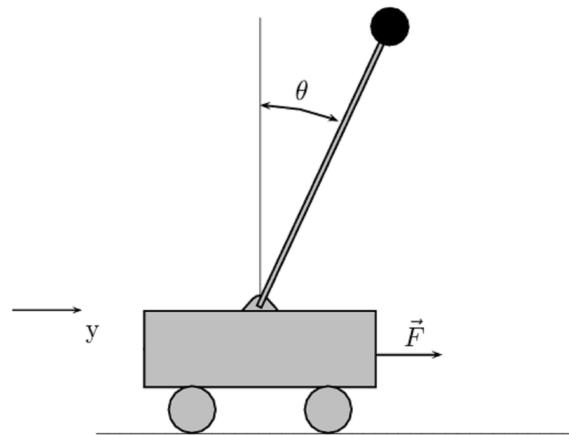
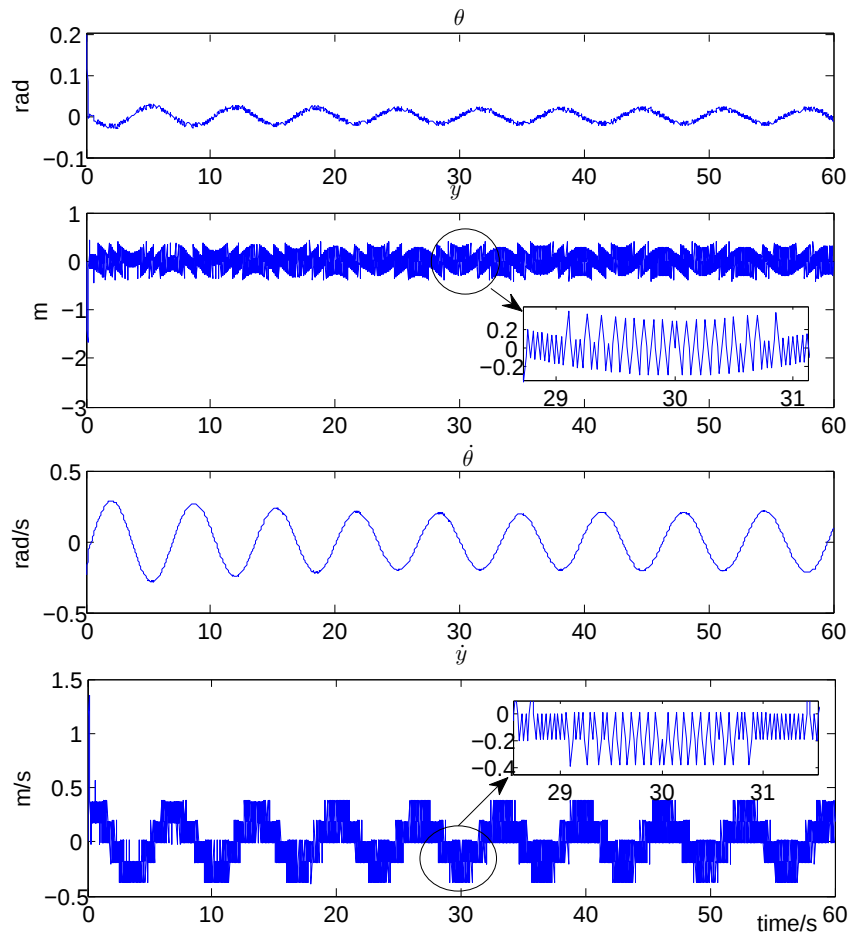
Define $A_1(\theta) = -\frac{l}{\cos \theta}(\frac{4}{3} - \frac{m \cos^2 \theta}{m_c + m})$, $A_2(\theta) = -\frac{g \sin \theta}{\cos \theta}$, $A_3(\theta, \dot{\theta}) = ml\dot{\theta} \sin \theta + \frac{\mu_p}{ml \cos \theta}$, $A_4(\dot{y}) = -\frac{\mu_c \operatorname{sgn}(\dot{y})}{\dot{y}}$, $A_5 = m_c + m$, $A_6(\theta, \dot{\theta}) = ml\dot{\theta} \sin \theta$, $A_7(\theta) = -ml \cos \theta$. With these notations, Equation (19) can be re-written as:

$$\begin{aligned} A_1 \ddot{\theta} &= F + A_2 + A_3 \dot{\theta} + A_4 \dot{y} \\ \frac{A_1 A_5}{A_1 + A_7} \ddot{y} &= F + \frac{A_2 A_7}{A_1 + A_7} + \frac{A_1 A_6 + A_3 A_7}{A_1 + A_7} \dot{\theta} + \frac{A_1 A_4 + A_4 A_7}{A_1 + A_7} \dot{y} \end{aligned} \quad (22)$$

By choosing

$$\begin{aligned} D &= \begin{bmatrix} A_1 & 0 \\ 0 & \frac{A_1 A_5}{A_1 + A_7} \end{bmatrix}, C = - \begin{bmatrix} A_3 & A_4 \\ \frac{A_1 A_6 + A_3 A_7}{A_1 + A_7} & \frac{A_1 A_4 + A_4 A_7}{A_1 + A_7} \end{bmatrix} \\ \phi &= - \begin{bmatrix} A_2 \\ \frac{A_2 A_7}{A_1 + A_7} \end{bmatrix}, q = \begin{bmatrix} \theta \\ y \end{bmatrix}, u = \begin{bmatrix} F \\ F \end{bmatrix} \end{aligned}$$

the system of Equation (19) coincides with the model of Equation (1). Note that the input u in this situation is constrained in the set $\Omega = \{u = [u_1, u_2]^T, u_1 = u_2 \in \mathbb{R}\}$.

Figure 1. The cart-pole system.**Figure 2.** State profiles of the cart-pole system with the proposed control strategy.

5.2. Experiment Setup and Results

In the simulation experiment, we set the discount factor $\gamma = 0.95$, the sliding surface parameter $k = 10$, $\delta_1 = 2$, $\delta_2 = 24$. The feasible control action set Ω in Equation (7) is defined as $\Omega = \{u = [u_1, u_2]^T, u_1 \in \mathbb{R}, u_2 \in \mathbb{R}, u_1 = u_2 = \pm 10 \text{ Newtons}\}$. This definition corresponds to the widely used bang-bang control in industry. To make the output of the action model within the feasible set, the output

of the action network is clamped to 10 if it is greater than or equal to zero and clamped to -10 if less than zero. The sampling period τ is set to 0.02 seconds. Both the critic model and the action model are linearly parameterized. The step size of the critic model, which is $l_c(n)$ and that of the action model, which is $l_a(n)$ are both set to 0.03. Both the update of the critic model weight W_c in Equation (16) and the update of the action model weight W_a in Equation (18) last for 30 seconds. For the uncontrolled cart-pole system with $F = 0$ in Equation (19), the pendulum will fall down. The control objective is to stabilize the pendulum to the inverted direction ($\theta = 0$). Time history of the state variables are plotted in Figure 2 for the system with the proposed self-learning variable structure control strategy. From this figure, it can be observed that θ is stabilized in a small vicinity around zero (with a small error of ± 0.1 rad), which corresponds to the inverted direction.

6. Conclusions and Future Work

In this paper, the self-learning variable structure control is considered to solve a class of sensor-actuator systems. The control problem is formulated from the optimal control perspective and solved via iterative methods. In contrast to existing models, this method does not need pre-knowledge on the accurate mathematic model. The critic model and the the action model are introduced to make the method more practical. Simulations show that the control law obtained by the proposed method indeed achieves the control objective. Future work on this topic includes the theoretical proof of the convergence and exploration on the performance limit of the proposed strategy. Also, the control of other mechanical systems modeled by Euler-Lagrange system, such as manipulators *etc.*, will be explored in our future work.

Acknowledgements

Shuai Li would like to share with the readers the poem by Rabindranath Tagore “The traveler has to knock at every alien door to come to his own and one has to wander through all the outer worlds to reach the innermost shrine at the end”. The authors would like to acknowledge the support by the National Natural Science Foundation of China under Grant No. 61172165 and Guangdong Science Foundation of China under Grant No. S2011010006116 and No. 10151802904000013.

References and Notes

1. Isermann, R. Modeling and design methodology for mechatronic systems. *IEEE/ASME Trans. Mechatr.* **1996**, *1*, 16–28.
2. van de Panne, M.; Fiume, E. Sensor-actuator networks. In *Proceedings of the 20th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '93)*, Anaheim, CA, USA, 1–6 August 1993; pp. 335–342.
3. Liu, B.; Chen, S.; Li, S.; Liang, Y. Intelligent control of a sensor-actuator system via kernelized least-squares policy iteration. *Sensors* **2012**, *12*, 2632–2653.
4. de Silva, C. *Sensors and Actuators: Control System Instrumentation*; Taylor & Francis, CRC Press: Boca Raton, FL, USA, 2007.

5. Beer, F.P. *Vector Mechanics for Engineers: Statics and Dynamics*; McGraw-Hill: New York, NY, USA, 2003.
6. Lewis, F.L.; Dawson, D.M.; Abdallah, C.T. *Manipulator Control Theory and Practice*; Marcel Dekker: New York, NY, USA, 2004; Volume 15.
7. Li, S.; Chen, S.; Liu, B.; Li, Y.; Liang, Y. Decentralized kinematic control of a class of collaborative redundant manipulators via recurrent neural networks. *Neurocomputing* **2012**, *8*, 108–121.
8. Li, S.; Meng M.Q.H.; Chen, W. SP-NN: A novel neural network approach for path planning. In *Proceedings of IEEE International Conference on Robotics and Biomimetics*, Sanya, Hainan, China, 15–18 December 2007; pp. 1355–1360.
9. Bloch, A.M. *Nonholonomic Mechanics and Control*; Springer-Verlag: New York, NY, USA, 2003.
10. Yu, H.; Liu, Y.; Yang, T. Tracking control of a pendulum-driven cart-pole underactuated system. In *Proceedings of IEEE International Conference on Systems, Man and Cybernetics*, Montreal, QC, Canada, 7–10 October 2007; pp. 2425–2430.
11. Seifried, R. Two approaches for feedforward control and optimal design of underactuated multibody systems. *Multibody Syst. Dynam.* **2012**, *27*, 75–93.
12. Isidori, A. *Nonlinear Control Systems II*; Springer-Verlag: New York, NY, USA, 1999.
13. Primbs, J.A.; Nevistic, V.; Doyle, J.C. Nonlinear optimal control: A control lyapunov function and receding horizon perspective. *Asian J. Control* **2009**, *1*, 14–24.
14. Ortega, R.; Loria, A.; Nicklasson, P.J.; Sira-Ramirez, H. *Passivity-Based Control of Euler-Lagrange Systems*; Springer-Verlag: New York, NY, USA, 1998.
15. Azhmyakov, V. Optimal control of mechanical systems. *Diff. Equat. Nonlin. Mech.* **2007**, *12*, 3–16.
16. Huo, W. Predictive variable structure control of nonholonomic chained systems. *Int. J. Comput. Math.* **2008**, *85*, 949–960.
17. Dumitrascu, B.; Filipescu, A.; Minzu, V.; Filipescu, A. Backstepping control of wheeled mobile robots. In *Proceedings of 15th International Conference on System Theory, Control, and Computing (ICSTCC 2011)*, Sinaia, Romania, 14–16 October 2011; pp. 1–6.
18. Hussein, I.I.; Bloch, A.M. Optimal control of underactuated nonholonomic mechanical systems. *IEEE Trans. Autom. Control* **2005**, *53*, doi:10.1109/TAC.2008.919853.
19. Pazderski, D.; Kozowski, K.; Krysiak, B. Nonsmooth stabilizer for three link nonholonomic manipulator using polar-like coordinate representation. In *Robot Motion and Control*; Kozłowski, K., Ed.; Springer: Berlin/Heidelberg, Germany, 2009.
20. Cuesta, F.; Ollero, A.; Arrue, B.C.; Braunstingl, R. Intelligent control of nonholonomic mobile robots with fuzzy perception. *Fuzzy Sets Syst.* **2003**, *134*, 47–64.
21. Wai, R.J.; Liu, C.M. Design of dynamic petri recurrent fuzzy neural network and its application to path-tracking control of nonholonomic mobile robot. *IEEE Trans. Indust. Electr.* **2009**, *56*, 2667–2683.
22. Kinjo, H.; Uezato, E.; Duong, S.C.; Yamamoto, T. Neurocontroller with a genetic algorithm for nonholonomic systems: Flying robot and four-wheel vehicle examples. *Artif. Life Robot.* **2009**, *13*, 464–469.

23. Bertsekas, D.P. *Dynamic Programming and Optimal Control*, 3rd ed.; Athena Scientific: Nashua, NH, USA, 2005.
24. Murray, J.J.; Cox, C.J.; Lendaris, G.G.; Saeks, R. Adaptive dynamic programming. *IEEE Trans. Syst. Man Cyber.* **2002**, *32*, 140–153.
25. Lewis, F.L.; Vrabie, D. Reinforcement learning and adaptive dynamic programming for feedback control. *IEEE Circuits Syst. Mag.* **2009**, *9*, 32–50.
26. Si, J.; Barto, A.; Powell, W.; Wunsch, D. *Handbook of Learning and Approximate Dynamic Programming*; John Wiley and Sons: Hoboken, NJ, USA, 2004.
27. Mayne, D.Q.; Michalska, H. Receding horizon control of nonlinear systems. *IEEE Trans. Autom. Control* **1990**, *35*, 814–824.
28. Bishop, C.M. *Pattern Recognition and Machine Learning*; Springer: New York, NY, USA, 2006.
29. Si, J.; Wang, Y.T. Online learning control by association and reinforcement. *IEEE Trans. Neural Netw.* **2001**, *12*, 264–276.

© 2012 by the authors; licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution license (<http://creativecommons.org/licenses/by/3.0/>.)