5–10 July 2004, Antalya, Turkey — Dynamical Systems and Applications, Proceedings, pp. 712–724

Stabilizing Acrobot by Using Nonlinear Programming Based on Sliding Mode Controller

Ahmet Yazıcı
Electrical Engineering Department, Osmangazi University,
Eskişehir, TURKEY

Abdurrahman Karamancıoğlu
Electrical Engineering Department, Osmangazi University,
Eskişehir, TURKEY

Rafail N. Gasimov
Industrial Engineering Department, Osmangazi University,
Eskişehir, TURKEY

Abstract

We design sliding mode controllers for nonlinear dynamic systems by using a nonlinear programming approach. We show that by appropriate selection of the objective function and the constraints, it is possible to obtain sliding mode controller parameters by solving a sequence of nonlinear programming problems. These parameters determine the forcing function which satisfies possibly nonlinear, even nonconvex constraints and optimize a given nonlinear objective function. We use the Modified Subgradient Algorithm for the nonconvex optimization problems encountered in computing such forcing functions. We illustrate the validity of our approach by stabilizing an under-actuated two link robot manipulator, called Acrobot, at vertically upright position.

Keywords: Sliding mode control, Nonlinear programming, Modified subgradient method, Acrobot stability.

1 Introduction

We introduce a new approach to compute forcing functions for a class of dynamic systems. More specifically, we introduce a nonlinear programming approach for sliding mode control (SMC) of nonlinear dynamic systems and apply it to Acrobot, a two link robot manipulator.

Sliding mode control aims at generating a desired trajectory for a given system by using an input which may be a discontinuous function of the system states.

The SMC control input at first steers the state of a nonlinear dynamic system towards a prespecified neighborhood of a stable surface in the state space. Following this, the control input steers the state towards the origin while keeping it in this neighborhood. SMC techniques have received significant attention of the researchers since the survey paper of Utkin [15]. In the beginning, the researchers focused on the analysis of second order systems using graphical notions. In the following decades SMC techniques have been generalized for application to more general classes of systems. These results take place in ([7, 13, 17]) and the references therein. It has been emphasized in the literature that the superiority of SMC is apparent in its performance in the presence of system modelling errors and disturbances. For the bounded modelling errors and the disturbances, the SMC is shown to achieve its control goals ([7, 13, 15, 17]).

We model the SMC input generation problem as a nonlinear optimization problem which is potentially nonconvex. For this optimization problem we use sharp augmented Lagrangians approach and construct a dual problem. In order to solve the dual problem we use the Modified Subgradient Algorithm (MSA) introduced in [9]. The algorithm presented in [9] does not require any convexity and differentiability assumptions, therefore it is applicable to a large class of problems. The gradient and subgradient methods and their different versions are investigated in ([3, 4]). The duality gap which is a major problem in the nonlinear programming has been investigated and the theoretical tools for zero duality gap property have been improved extensively in [1, 9, 10, 11] and [12] (see also the references therein).

The control problem in this paper partitions the control process in two phases: The reaching and the sliding phases. Each phase is associated with an appropriate objective function and constraints. In each phase the sliding mode controller input is updated at certain time instants by using the solution of a nonlinear programming problem. We show that by appropriate selection of the objective function and the constraints in each phase, it is possible to obtain a fast reaching performance and improve the chattering. In [16], the nonlinear programming based sliding mode controller is applied to an inverted pendulum system, which is known to have a rich and nonlinear dynamic structure.

A very significant research in the literature that considers optimal control problem in the nonlinear programming framework belongs to Betts [5]. In [5], the optimal control problem is viewed as an infinite-dimensional extension of the nonlinear programming problem. Considering that practical methods for solving these problems require Newton-based iterations with a finite set of variables and constraints, the infinite-dimensional problem is converted to a finite-dimensional approximation. It is shown in [5] that the so-formed problem is "large and sparse", and iterative approaches that exploit these properties are proposed to solve the problem.

In the next section we present a brief background on the sliding mode control problem. Using this we model the SMC problem as a nonlinear programming problem. Following this, we present dynamics of Acrobot, a popular two link robot manipulator, and apply our method to it. In the concluding section we make comments on the performance of the controller.

2 Problem statement

In this section we briefly introduce the SMC problem and thereafter present our approach and its major tool, the modified subgradient algorithm (MSA).

Consider the single-input nonlinear differential equation

$$\dot{X} = a(X, u), \tag{1}$$

where $X \in \mathbb{R}^n$ is the state vector and u is the scalar control input. The entries of the n-dimensional vector function a are continuous with continuous bounded derivatives with respect to the components of X. It is assumed that system (1) is controllable.

We use the SMC techniques to generate an input function u which gives rise to a solution of (1) such that its trajectory in the state space at first reaches a prespecified neighborhood of a sliding surface, which then goes to the origin of the state space within this neighborhood asymptotically. The SMC design consists of the following two steps:

1) Designing a stable surface for the system in $\mathbb{R}^n$.

2) Designing a control input that restricts the trajectory of (1) to a prespecified neighborhood of a stable surface. This makes the trajectory move towards the origin without leaving the neighborhood of the surface. In this paper we make a contribution to the second step of the SMC design. We make reference to the literature for the first step.

In practice, for the sake of simplicity in design, the surface mentioned above is generally a subspace. We also use a subspace in the design. Choosing a stable subspace guarantees that every trajectory restricted to the neighborhood of the subspace reaches the origin of the state space asymptotically ([7, 13, 15, 17]). The SMC theory relies on the existence of stable sliding subspaces. Systematic and *ad hoc* methods for determining stable subspaces is available in the literature ([7, 13, 15, 17]).

Next we outline a principle of computing a sliding mode control input. Consider the $(n-1)$-dimensional subspace of $\mathbb{R}^n$

$$\{X \in \mathbb{R}^n : \; GX = 0\},$$

where G is a row matrix. Also define a function of the state $s := GX$, and consider the positive definite function

$$V = \frac{1}{2}s^2. \tag{2}$$

This function decreases as X gets closer to the subspace, and reaches zero value on the subspace. Let us design an input u for the system (1) so that the time derivative of V becomes negative. Negativity of $\frac{dV}{dt}$ in some interval of t means that V decreases in that interval. Hence the trajectory approaches the subspace. The common form of u used in the literature that yields negative $\frac{dV}{dt}$ is

$$u = u_{eq}(X) + u_1(X) + \cdots + u_r(X) + \Gamma \operatorname{sgn}(s), \tag{3}$$

where u_{eq} is the equivalent control input, a fixed function of X. Each of the r terms that follow the equivalent input switches between some fixed functions of X to cancel possibly positive terms in $\frac{dV}{dt}$, and the last term is used to ensure reaching the subspace in finite time. There are numerous methods for the computation of the input u ([7, 13, 15, 17]). Here, in this paper, we present a new way of computing the sliding mode control input.

If the initial state is not in the prespecified neighborhood of the origin, then the input function u has to drive it there. The evolution of the state from such an initial condition to the boundary of the neighborhood of the subspace is called the reaching phase. This phase is characterized by a positive number δ such that when $s > \delta$, the state is said to be in the reaching phase. $s = \delta$ is the boundary for the neighborhood of the sliding subspace. On each side of the boundary the required trajectory behavior is different. Therefore, the decision for the value of δ is made by the designer regarding performance specifications of the system [7]. In the reaching phase we select u which minimizes the objective function $\frac{dV}{dt}$. Obviously, minimizing $\frac{dV}{dt}$ means doing the best to lessen the distance between the state and the subspace. To make sure that V strictly decreases, we require it to be less than a negative number $-\eta s^2$, where η is chosen as a positive number complying with physical specifications of the dynamic system. Because of this constraint, the state strictly approaches the sliding subspace. For this constraint, in addition to the upper bound for $\frac{dV}{dt}$, we also impose a lower bound $-\gamma s^2$, where γ is a positive real number in order to avoid too fast approaching rates. The factor s^2 in this constraint enlarges the feasible $\frac{dV}{dt}$ values for states farther from the subspace, and makes the feasible interval of $\frac{dV}{dt}$ smaller for the states closer the subspace. As the second constraint, we impose upper limit on the size of the input, call it α. As the third constraint, the set Ω contains the admissible feedback coefficients. This set is required to be compact which may contain discrete or continuous elements. The nonlinear programming problem associated with the reaching phase is as follows:

$$\min_{K} \frac{dV}{dt} \tag{4}$$

$$\text{subject to} \begin{cases} -\gamma s^2 \leq \frac{dV}{dt} \leq -\eta s^2, \\ u \in \{KX : |KX| \leq \alpha\}, \\ K \in \Omega \subseteq \mathbb{R}^n. \end{cases}$$

The sliding phase is the system dynamics in the δ-neighborhood of the sliding subspace. The sliding and the reaching phases are complementary phases in the state space (*i.e.*, the state is always in only one of them). When the state is in the sliding phase, we require solution fields which point towards an appropriate combination of the origin and the sliding subspace. More specifically; defining w as the projection of X on the sliding subspace, we select an optimal feasible input function such that the projection w in the next step of simulation is the smallest possible. The nonlinear programming problem associated with the sliding phase is as follows:

$$\min_{K} \left\| w + \frac{w^T \cdot \frac{\dot{X}}{\|\dot{X}\|}}{\|w\|} w \right\| \tag{5}$$

$$\text{subject to} \begin{cases} -\gamma s^2 \leq \frac{dV}{dt} \leq -\eta s^2, \\ u \in \{KX : |KX| \leq \alpha\}, \\ K \in \Omega \subseteq \mathbb{R}^n \end{cases}$$

Notice that the constraints are the same as in the reaching phase (they could be different though). Minimizing the objective function under these constraints yields the aforementioned solution field direction. The solution fields that avoid crossing the sliding subspace obviously improve the chattering. Chattering, frequent crossings of the sliding subspace by the solution trajectory, is one of the major topics in SMC theory, and various solutions to this problem are proposed in the literature ([2, 7, 15, 17]).

A possible special case in the sliding algorithm occurs when $w = 0$. When $w = 0$, the objective function in (5) equals zero. We handle this case by switching to the reaching phase sub-algorithm, which is well defined for the $w = 0$ case, and the feedback coefficient vector K obtained by this sub-algorithm is suitable for our control objective.

The nonlinear programming based SMC algorithm, which we call the Main Algorithm, utilizes the nonlinear programming problems given above in the reaching and sliding phases. It is as follows:

Step 1. *(Initialization Step)* Assign initial values to the time t and the state X, *i.e.*, $t \leftarrow\ t_0,\ X \leftarrow X_0$. Step 2. *($\delta$-checking Step)* Check whether $\|s\| \leq \delta$ or not. If $\|s\| > \delta$, then solve the reaching phase problem, else if $\|s\| \leq \delta$ then solve the sliding phase problem. Use the solution to form $u = KX$. Step 3. Use u found in step 2 and run system (1) from t to $t + \Delta t$. Step 4. Update the time and the state, and go to Step 2.

The control input function updating the interval Δt is determined by regarding the smallest time constant of the physical system. For instance, for the differential equations arising from the systems that are governed by the Newtonian mechanics, such as robotics, these time constants are well-known. For a large class of differential equations of unknown origin, these time constants can be found by using various techniques. For instance, in linear systems, eigenvalues of the coefficient matrix can be used for this purpose.

3 The modified subgradient algorithm

The major tool that we use for solving the nonlinear programming problem is the modified subgradient algorithm [9]. It has a notable performance in having zero duality gaps for a large class of nonconvex problems. Nonlinear programming problems of reaching and sliding phases can be brought into the standard form as in the following problem (P):

$$\min_K f(K) \tag{6}$$

$$\text{subject to} \begin{cases} h(K) = 0, \\ K \in \Omega, \end{cases}$$

where $h(K)$ is the constraint vector. In the sequel we call (6) the primal problem.

In the next section we present the modified subgradient algorithm that solves the dual problem corresponding to the primal problem (6).

The sharp Lagrangian function for problem (6) is defined as follows:

$$L(K, v, c) = f(K) + c\|h(K)\| - v^T h(K), \tag{7}$$

where $v \in \mathbb{R}^2$ and $c \in \mathbb{R}_+$. Defining the dual function as

$$H(v, c) = \min_{K \in \Omega} L(K, v, c), \tag{8}$$

the dual problem (P^*) is

$$\max_{(v,c) \in \mathbb{R}^m \times \mathbb{R}_+} H(v, c). \tag{9}$$

Using the definitions above, the MSA is as follows:

Initialization. Choose a pair (v_1, c_1) with $v_1 \in \mathbb{R}^2$, $c_1 \geq 0$, and let $j = 1$, and go to Step 1.
Step 1. Given (v_j, c_j), solve the following subproblem:

$$\min_{K} f(K) + c_j\|h(K)\| - v_j h(K) =: H(v_j, c_j) \tag{10}$$

$$\text{subject to } K \in \Omega.$$

Let K_j be a solution of (10). If $h(K_j) = 0$, then stop; (v_j, c_j) is an optimal solution to the dual problem and K_j is a solution to (6), so $f(K_j)$ is the optimal value of problem (6). Otherwise, go to Step 2.
Step 2. Update (v_j, c_j) by

$$\begin{aligned} v_{j+1} &= v_j - z_j h(K_j), \\ c_{j+1} &= c_j + (z_j + \epsilon_j)\|h(K_j)\|, \end{aligned} \tag{11}$$

where z_j and ϵ_j are positive scalar step sizes defined in the sequel. Replace j by $j + 1$ and go to Step 1.
Step size calculation:
Let us consider the pair (v_j, c_j) and calculate

$$H(v_j, c_j) = \min_{K \in \Omega}\{f(K) + c_j\|h(K)\| - v_j h(K)\}$$

and let $h(K_j) \neq 0$ for the corresponding K_j, which means that K_j is not optimal. Then the step size parameter z_j can be calculated as

$$\begin{aligned} 0 < z_j \leq \tfrac{2(\overline{H} - H(v_j, c_j))}{5\|h(K_j)\|^2}, \\ 0 < \epsilon_j < z_j, \end{aligned} \tag{12}$$

where $\overline{H}$ is an upper bound for the dual function. For a rigorous treatment of the MSA one may refer to [9].

4 Stabilization of Acrobot

Acrobot is a two link planar robot manipulator with a single actuator at the elbow (Figure 1). The control objective in Acrobot stabilization problem is to drive the links of Acrobot to vertically upright position from every neighboring initial conditions, and keep them in that position thereafter. In other words, it is desired to find a control input function that moves the joint angles (θ_1, θ_2) to the unstable

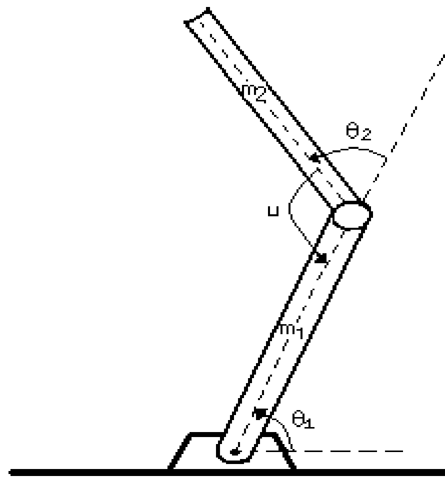

Figure 1: The Acrobot system

equilibrium point $(\frac{\pi}{2}, 0)$ from every initial conditions in the neighborhood of this point, and keep it there in the subsequent times. We use the nonlinear programming based sliding mode control for this stabilization problem. Recently, Acrobot control has gained popularity and received a significant attention in the literature ([6, 14] and references therein). The popularity of Acrobot is due to its under-actuated and rich dynamics, in which the control of two quantities (*i.e.*, θ_1 and θ_2) by using only one control input function is a challenging problem.

The nonlinear differential equation representing Acrobot's equation of motion is

$$\begin{aligned} \dot{x}_1 &= x_2, \\ \dot{x}_2 &= \tfrac{-d_{12}(u-c_2-\phi_2)-d_{22}(c_1+\phi_1)}{d_{11}d_{22}-d_{12}^2}, \\ \dot{x}_3 &= x_4, \\ \dot{x}_4 &= \tfrac{d_{11}(u-c_2-\phi_2)-d_{12}(c_1+\phi_1)}{d_{11}d_{22}-d_{12}^2}, \end{aligned} \tag{13}$$

where the components of the system state $x_1, \ldots, x_4$ are defined by $x_1 := \theta_1, x_2 := \dot{\theta}_1, x_3 := \theta_2, x_4 := \dot{\theta}_2$. In (13), u denotes the control input, and $d_{11}, d_{22}, d_{12}, c_1, c_2, \phi_1$, and ϕ_2 denote the quantities which are functions of masses and dimensions of the links, and the system states (one may refer to [6] or [14] for the details).

Using typical values for the link masses, link lengths, link lengths from joint to mass center, and link inertias as in [6], we obtain Acrobot's equation of motion as

$$\begin{aligned}
\dot{x}_1 &= x_2, \\
\dot{x}_2 &= \frac{-(7.99\ 10^{-3}+8.9\ 10^{-3}\cos(x_3))(u-8.9\ 10^{-3}x_2^2\sin(x_3)-0.4367\cos(x_1+x_3))}{0.84\ 10^{-3}+0.14\ 10^{-3}\cos(x_3)-(7.99\ 10^{-3}+8.9\ 10^{-3}\cos(x_3))^2} \\
&\quad + \frac{(0.071\ 10^{-3}x_4+0.1423\ 10^{-3}x_2)x_4\sin(x_3)-38.84\ 10^{-3}\cos(x_1)-3.49\ 10^{-3}\cos(x_1+x_3)}{0.84\ 10^{-3}+0.14\ 10^{-3}\cos(x_3)-(7.99\ 10^{-3}+8.9\ 10^{-3}\cos(x_3))^2}, \\
\dot{x}_3 &= x_4,
\end{aligned}$$

$$\begin{aligned}
\dot{x}_4 &= \frac{(0.1062+0.178\cos(x_3))(u-8.9\ 10^{-3}x_2^2\sin(x_3)-0.4367\cos(x_1+x_3))}{0.84\ 10^{-3}+0.14\ 10^{-3}\cos(x_3)-(7.99\ 10^{-3}+8.9\ 10^{-3}\cos(x_3))^2} \\
&+ \frac{(7.99\ 10^{-3}+8.9\ 10^{-3}\cos(x_3))((-8.9\ 10^{-3}x_4-17.89\ 10^{-3}x_2)x_4\sin(x_3)+4.86\cos(x_1)+0.436\cos(x_1+x_3))}{0.84\ 10^{-3}+0.14\ 10^{-3}\cos(x_3)-(7.99\ 10^{-3}+8.9\ 10^{-3}\cos(x_3))^2}.
\end{aligned} \tag{14}$$

Using the transformation $\tilde{X} := X - X_e$, we can restate the control objective equivalently as to generate a SMC input function that drives the new state $\tilde{X}$ to the origin of the state space and keep it there in subsequent times.

Analyzing the behavior of the Acrobot system in the neighborhood of the vertically upright position, we found a stable subspace $\{X : \ G\tilde{X} = 0\}$ with $G = \begin{bmatrix} 7.4 & 1.6 & 0.8 & 0.2 \end{bmatrix}$. One may refer to ([7, 13, 15, 17]) for systematic approaches of stable sliding surface design.

We next compare our stabilization approach with that in [7]. Consider the sliding mode control law (Eqn. (7.22) in [7])

$$u = \begin{cases} u_{eq} - \frac{b(X)\nabla_X V}{\|b(X)\nabla_X V\|}\hat{\rho} & \text{if } |s| \geq \delta, \\ u_{eq} - \frac{b(X)\nabla_X V}{\|b(X)\nabla_X V\|}\hat{\rho}\frac{|s|}{\delta} & \text{if } |s| < \delta, \end{cases} \tag{15}$$

with $\delta = 0.01$, $\hat{\rho} = 0.1$, and the initial condition $X(0) = \begin{bmatrix} \frac{\pi}{2} & 0 & \frac{\pi}{36} & 0 \end{bmatrix}^T$. Applying this to the Acrobot yields the simulation results given in Figures 2–3.

We simulate the Acrobot dynamics (14) using the nonlinear programming based SMC approach given by Main Algorithm. Noticing that the absolute value of the input u is bounded by 0.5 in Figure 3, we impose the same bound $\alpha = 0.5$ on u in the simulation using our method. Besides, we imposed additional magnitude constraints on the feedback coefficients vector K in our method by selecting $\Omega = \{(k_1, k_2, k_3, k_4) : \ -6 \leq k_i \leq 6, \ i = 1, \ldots, 4\}$. This obviously makes the feasible solution set smaller, and increases the difficulty level of the problem. For the purpose of comparison, we use the same sliding band parameter $\delta = 0.01$ as in [7]. We specify the desired interval for $\frac{dV}{dt}$ by setting $\gamma = 15$, and $\eta = 10$. Regarding similar robotics applications, we select the control input updating interval as $\Delta t = 0.01$. Figures 4-5 show the computer simulation results of the Acrobot stabilization using the nonlinear programming based SMC. (We coded our algorithm in MATLAB and

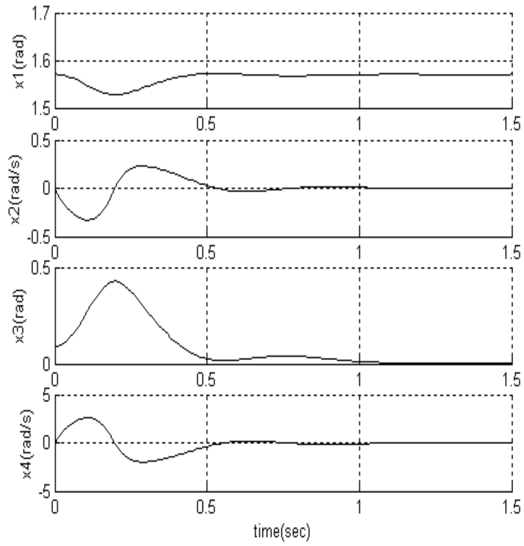

Figure 2: Simulation results: States of Acrobot under the control law in [7]

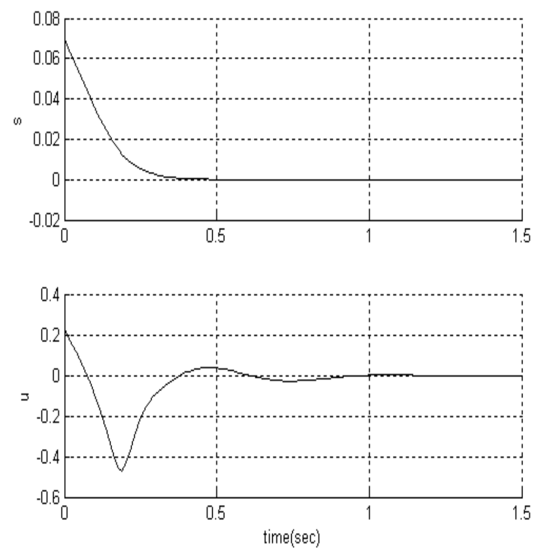

Figure 3: Simulation results: The input u and variable s under the control law in [7]

GAMS [8]. We utilized differential equation solving and graphics capabilities of the former, and optimization capability of the latter. In MATLAB we used the differential equation solver *ode23.m*. In GAMS, for every problem we compared performances of unconstrained problem solvers *conopt* and *minos*, and mostly used the *minos*.)

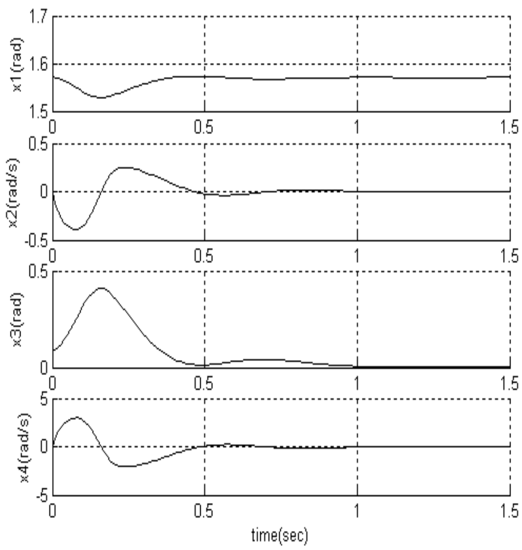

Figure 4: Simulation results: The Acrobot states under the Reaching and Sliding Algorithms

Comparing Figure 3 to Figure 5, in terms of reaching time to the sliding subspace neighborhood and smoothness of the trajectory in this neighborhood, it can be noticed that the nonlinear programming based SMC approach is an alternative solution method for the problem. Owing to using an objective function that rewards faster approaching rates in the reaching phase, and by using another objective function that rewards solutions fields towards the origin in the sliding phase, the performance of our method is comparable with the existing methods.

At this point a few words may be in order about the robustness of the algorithm. Our algorithm carries out the control objective robustly even if the system (1) contains a disturbance term. In case a disturbance term causes a deviation from the nominal trajectory, the Main Algorithm generates the best feasible feedback coefficients K, and the system pursues it goal from the disturbed state. Here we assume that the disturbance term is bounded, and bounds of the control input are sufficiently large to achieve the control objective in the presence of the disturbance.

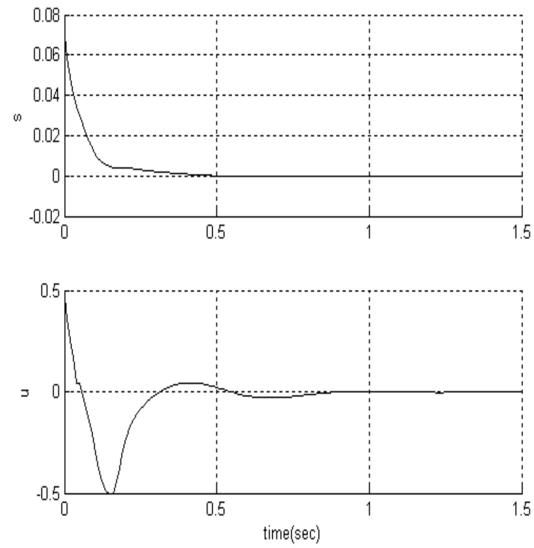

Figure 5: Simulation results: The input u and variable s under the Reaching and Sliding Algorithms

5 Conclusion

It has been shown that the feedback coefficients in the sliding mode control of an Acrobot, as well as any other dynamic system, can be expressed as a nonlinear programming problem with appropriately selected objective function and constraints. We used the modified subgradient algorithm to solve these nonlinear programming problems which are possibly nonconvex. The validity of our approach has been verified by the simulation results.

References

[1] Azimov A. Y. and Gasimov R. N., *On weak conjugacy, weak subdifferentials and duality with zero gap in nonconvex optimization*, International Journal of Applied Mathematics, **1** (1999), 171–192.

[2] Bartolini G., Ferrara A., Usai E. and Utkin V. I., *On multi-input chattering free second order sliding mode control*, IEEE Transactions on Automatic Control, **45** (2000), No. 9, 1711–1717.

[3] Bazaraa M. S., Sherali H. D. and Shetty C. M., *Nonlinear Programming: Theory and Algoritms*, John Wiley and Sons, 1993.

[4] Bertsekas D. P., *Nonlinear Programming*, Athena Scientific, Belmont, MA, 1995.

[5] Betts J. T., *Practical Methods for Optimal Control Using Nonlinear Programming*, Advances in Design and Control, SIAM, 2001.

[6] Brown S. C. and Passino K. M., *Intelligent control for an Acrobot*, Journal of Intelligent and Robotic Systems, **18** (1997), 209–248.

[7] DeCarlo R. A., Zak S. H. and Matthews G. B., *Variable structure control of nonlinear multivariable systems: A tutorial*, Proc. IEEE, **76** (1998), No. 3, 212–232.

[8] Ferris M. C., *MATLAB and GAMS: Interfacing, optimization, and visualization software*. ftp://ftp.cs.wisc.edu/math-prog/tech-reports/98-19.ps April 2004.

[9] Gasimov R. N., *Augmented Lagrangian duality and nondifferentiable optimization methods in nonconvex programming*, Journal of Global Optimization, **24** (2002), 187–203.

[10] Gasimov R. N. and Rubinov A.M., *On augmented Lagrangians for optimization problems with a single constraint*, Journal of Global Optimization, **28** (2004), 153–173.

[11] Rockafellar R. T. and Wets R. J. B., *Variational Analysis*, Springer–Berlin, 1998.

[12] Rubinov A. M. and Gasimov R. N., *Strictly increasing positively homogeneous functions with applications to exact penalization*, Optimization, **52** (2003), No. 1, 1–28.

[13] Slotine J. J. E. and Sastry S. S., *Tracking control of non-linear systems using sliding surfaces, with application to robot manipulators*, International Journal of Control, **38** (1983), No. 2, 465–492.

[14] Spong M. W., *The swing-up control problem for the Acrobot*, IEEE Control Systems Magazine, **15** (1995), 49–55.

[15] Utkin V. I., *Variable structure systems with sliding modes*, IEEE Transactions on Automatic Control, **22** (1977), No. 2, 212–222.

[16] Yazici A., Karamancioğlu A. and Gasimov R. N., *Nonlinear programming based sliding mode control of an inverted pendulum system*, in: Proceedings of ELECO 2003 — International Conference on Electrical and Elecronics Engineering, Bursa, Turkey, 2003, pp. 293–297.

[17] Young K. D., Utkin V. I. and Özgüner Ü., *A control engineer's guide to sliding mode control*, IEEE Transactions on Control Systems Technology, **7** (1999), No. 3, 328–342.