NONLINEAR PROGRAMMING BASED SLIDING MODE CONTROL OF AN INVERTED PENDULUM

A. Yazıcı[1], A. Karamancıoğlu[1], R. N. Gasimov[2]

ayazici@ogu.edu.tr *akaraman@ogu.edu.tr* *gasimovr@ogu.edu.tr*

[1]*Electrical Engineering Department, Osmangazi University, Meselik, Eskisehir TURKEY*

[2]*Industrial Engineering Department, Osmangazi University, Bademlik, Eskisehir TURKEY*

ABSTRACT

We design sliding mode controllers 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. We illustrate validity of our approach by stabilizing an inverted pendulum.

I. INTRODUCTION

In this paper we introduce a nonlinear programming (NLP) approach for sliding mode control of nonlinear dynamic systems. By using NLP techniques we obtain a control input which at first steers the state of a nonlinear dynamic system towards a stable subspace in the state space, and once it enters a prespecified neighborhood of the subspace the control input steers it towards the origin while keeping it in this neighborhood. The input is designed to satisfy possibly nonlinear, even nonconvex constraints and optimize a given nonlinear objective function. The mathematical tool that we utilize for this purpose allows us to consider such problems.

We firstly present a brief background on the sliding mode control (SMC) problem. Following this we model the SMC problem as a NLP problem. It will be shown that by appropriate selection of the objective function and the constraints it is possible to obtain a fast reaching performance and improve the chattering. In the section of Experimental Results, we apply the proposed approach to an inverted pendulum system. In the concluding section we make comments on the performance of the controller.

Sliding mode control aims to generate a desired trajectory for a given system by using an input which may be discontinuous function of the system states. SMC techniques have received increasing attention of the researchers since the survey paper of Utkin [1]. In the beginning the researchers focused on analysis of the second order systems using graphical notions. In the following decades SMC techniques have been generalized to more general classes of systems. These results take place in ([2]-[4]) 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. In this paper by using NLP approaches we select optimal SMC inputs from a set of admissible inputs at a sequence of updating instants. For this optimization problem we use *sharp augmented Lagrangians* that work for nonconvex problems as well as the convex ones. We construct a dual problem with respect to the augmented Lagrangian. In order to solve the dual problem we use the Modified Subgradient Algorithm (MSA) introduced in [5]. The algorithm used here 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 ([7],[8]). The duality gap which is a major problem in the NLP has been investigated and the theoretical tools for zero duality gap condition have been improved extensively in [5],[6], [9],[10] and [11].

In [12] the sliding mode controller design by using NLP tools is investigated. The paper partition the simulation time 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 structure is updated at certain time instants by using a solution of a NLP problem. In [12], the unification the objective functions which results in a single phase is also presented. In the unified approach a pareto optimal solution of a NLP problem is used to update the sliding mode controller structure.

One of the significant researches in the literature that considers optimal control problem in the NLP framework belongs to Betts [13]. In [13], the optimal control problem is viewed as an infinite-dimensional extension of the NLP 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 [13] that the so-formed problem is "large and sparse", and iterative approaches that exploit these properties are proposed to solve the problem.

II. PROBLEM STATEMENT

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

Consider the single-input single-output n-th order nonlinear differential equation

$$\frac{d^n x}{dt^n} = a(X) + b(X)u \tag{1}$$

where $X = \left[x \quad \dot{x} \ldots x^{(n-1)}\right]^T$ is the state vector and u is the scalar control input. The scalar-valued functions a and b are continuous with continuous bounded derivatives with respect to the components of X. It is assumed that $b(X) \neq 0$ for all $X \in R^n$.

The strategy in SMC consists of two steps: 1) Choose a stable subspace of R^n 2) Design a control input that steers the trajectory of (1), at first to a prespecified neighborhood of a stable subspace, then once it reaches to this neighborhood, it steers towards the origin while keeping the states in this neighborhood of the subspace. Choosing a stable subspace guarantees that every trajectory restricted to the neighborhood of the subspace reaches to the origin asymptotically ([1]-[4]). In this paper we present an algorithm that drives the trajectories to the origin using the two steps mentioned above.

Consider *(n-1)*-dimensional subspace of R^n

$$S{:}=\{X \in R^n{:}\,GX = 0\} \tag{2}$$

where G is a row matrix. Also define $S{:}= GX$ for each $X \in R^n$ and consider the positive definite function

$$V = \frac{1}{2}s^2. \tag{3}$$

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

$$u = u_{eq}(X) + u_1(X) + \ldots + u_r(X) + \Gamma \operatorname{sgn}(s) \tag{4}$$

where u_{eq} is called equivalent control input which is a fixed function of X, each of the following r terms 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 to the subspace in finite time. There are numerous methods for the computation of the input u ([1]-[4]).

In this paper we propose a NLP technique to compute the input u which yields fast approaching perfomance in the phase of reaching the neighborhood of the subspace, and improves chattering characteristics in the neighborhood. Even though we try to make this article self-contained, one may refer to [12] for a more detailed treatment of the mathematical tools. For the sake of simplicity in presentation we consider the input having the form $u = KX$ where $K = [k_1 \quad \ldots \quad k_n] \in \Omega$ is a compact, possibly nonconvex, subset of R^n. The feedback gain parameter K is found as follows:

$$K = \begin{cases} so\ln \ \textit{of the prblm in the Reaching Phase if } |s| > \delta \\ so\ln \ \textit{of the prblm in the Sliding Phase if } |s| \leq \delta \end{cases} \tag{5}$$

where the NLP problems in the Reaching and the Sliding phases are defined shortly. In (5), the positive constant δ characterizes the neighborhood of the subspace, and depending on the system specifications its value is selected by the designer. The algorithm ends when a prespecified stop criterion is satisfied, for instance, it ends when the trajectory reaches into a prespecified neighborhood of the origin

The NLP problems for the Reaching and Sliding phases are defined as follows:.

The NLP Problem for the Reaching Phase

Using the MSA solve the following problem for K

$$\min_K \frac{dV}{dt}$$
$$\text{subject to} \begin{cases} \frac{dV}{dt} \leq -|s|\eta \\ u \in \{KX : |KX| \leq \alpha\} \\ \mathrm{K} \in \Omega \subseteq \mathrm{R}^n \end{cases} \tag{6}$$

The Reaching Algorithm is executed as long as the current state of the trajectory is outside $\delta -$neighborhood of the sliding subspace. In the reaching phase the minimization of the objective function $\frac{dV}{dt}$ serves for a speedy arrival in the neigborhood of the sliding subspace. When η is a positive number the first constraint makes sure that V strictly decreases, therefore, approaches the subspace. The factor $|s|$ on the right hand side requires larger $\frac{dV}{dt}$ values as the distance of the trajectory from the sliding subspace increases, and vice versa. For a positive α, the second constraint imposes upper limit on the size of the input. In 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 NLP Problem for the Sliding Phase

Using the MSA solve the following problem for K

$$\min_{K} \quad \left\| w + proj_w \frac{\dot{X}}{\|\dot{X}\|} \right\|$$

$$\text{subject to} \begin{cases} \dfrac{dV}{dt} \leq -|s|\eta \\ u \in \{KX : |KX| \leq \alpha\} \\ \mathrm{K} \in \Omega \subseteq \mathrm{R}^{n} \end{cases} \tag{7}$$

where w is a vector whose initial point is the origin and end point is the projection of X on the sliding subspace. A possible special case in the sliding algorithm occurs when $w=0$. When $w=0$, the objective function in (7) equals zero. We handle this case by switching to the Reaching Algorithm, which is well defined there and suitable for our control objective.

The Overall Algorithm

Let the initial time and state be t_0 and X_0 respectively. Depending on the location of the states in the state space, our main algorithm is as follow:

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. (δ *-checking Step*) Check whether $|s| \leq \delta$ or not. If $|s| > \delta$ then solve the NLP problem associated with the Reaching Phase (6), else if $|s| \leq \delta$ then solve the NLP problem associated with the Sliding Phase (7) to find K and calculate $u = KX$.

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 updating interval Δt is determined regarding the smallest time constant of the differential equation that models the system.

The Modified Subgradient Algorithm

The major tool that we use for solving the NLP problem is the the modified subgradient algorithm. In the literature it has the best performance in eliminating the duality gaps [5]. In this part, the MSA algorithm is explained for the problems that has –for simplicity in presentation- two constraints. It is straightforward to generalize it to more than two constraint case.

NLP problems can be brought into the standart form as in the following problem (P):

$$\min f(K)$$
$$\text{subject to} \begin{cases} h(K) = 0 \\ K \in \Omega \end{cases} \tag{8}$$

where $h(K) = [h_1(K) \quad h_2(K)]^T$ is the constraint vector. In the sequel we call (8) the primal problem. For problem (8), we use the sharp Lagrangian function

$$L(K,v,c) = f(K) + c\|h(K)\| + v^T h(K) \tag{9}$$

where $v \in R^2$ and $c \in R_+$. Defining the dual function as

$$H(v,c) = \min_{K \in \Omega} L(K,v,c) \tag{10}$$

the dual problem (P^*) is

$$\max_{(v,c) \in R^m \times R_+} H(v,c) \tag{11}$$

Using the definitions above the MSA is as follows:

Initialization.Choose a pair $(\mathrm{v}_1, \mathrm{c}_1)$ with $\mathrm{v}_1 \in \mathrm{R}^2$, $\mathrm{c}_1 \geq 0$, and let j=1, and go to Step 1.

Step 1. Given (v_j, c_j), solve the following subproblem:

$$\min_{\mathrm{K}} \mathrm{f(K)} + \mathrm{c_j}\|\mathrm{h(K)}\| - \mathrm{v_j}\ \mathrm{h(K)} =: \mathrm{H}(\mathrm{v_j}, \mathrm{c_j})$$
$$\text{subject to } K \in \Omega \tag{12}$$

Let K_j be the solution of (12). 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 (8), so $f(K_j)$ is the optimal value of problem (8). Otherwise, go to Step 2.

Step 2. Update (v_j, c_j) by

$$v_{j+1} = v_j - z_j\ h(K_j)$$
$$\mathrm{c}_{j+1} = \mathrm{c_j} + (\mathrm{z_j} + \varepsilon_j)\|\mathrm{h}(\mathrm{K_j})\| \tag{13}$$

where $\mathrm{z_j}$ 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 $\mathrm{z_j}$

$$0 < z_j \leq \frac{2(\overline{H}j - H(v_j, c_j))}{5\|h(K_j)\|^2\ \}}, \text{and } 0 < \varepsilon_j < z_j$$

where $\overline{H}_j$ is an upper bound for the dual function. For a rigorous treatment of the MSA one may refer to [5].

3. EXPERIMENTAL RESULTS

In this section we apply the theoretical results presented in the preceding section to an inverted pendulum stabilization problem. The stabilization problem is to design a controller to keep the pendulum in its unstable equilibrium point in the presence of disturbances. Because of its dynamically rich structure, inverted pendulum system is widely used in the literature to check validity of control strategies ([15], [16]).

Next we describe an inverted pendulum and present its nonlinear dynamic model. An inverted pendulum mounted on a motor-driven cart is shown in Figure 1. The inverted pendulum is intrinsically unstable, that is it may fall over any time in any direction unless a suitable control force is applied. Here we consider only a-two dimensional problem that the pendulum moves only in x-y plane. For this inverted pendulum system; *m* is mass of the rod which is assumed to be concentrated at the and of the rod, *l* is length of the rod, *M* is mass of the cart, and *b* is friction constant. The angle of the rod from the vertical line is θ and the distance in horizantal plane from the reference point is *x*.

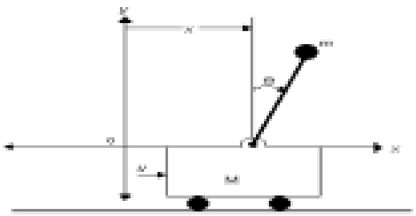

Figure 1 Inverted Pendulum system

Nonlinear model of inverted pendulum system is as follows ([17], [18]):

$$\dot{x}_1 = x_2$$
$$\dot{x}_2 = \frac{-bx_2 + ml\sin(x_3)x_4^2 - mg\sin(x_3)\cos(x_3) + u}{M + m - m\cos(x_3)^2}$$
$$\dot{x}_3 = x_4$$
$$\dot{x}_4 = \frac{(bx_2 - u - ml\sin(x_3)x_4^2)\cos(x_3) + (M+m)g\sin(x_3)}{l(M + m - m\cos(x_3)^2)} \quad (14)$$

where $x_1 := x$, $x_2 := \dot{x}$, $x_3 := \theta$, $x_4 := \dot{\theta}$. Table 1 contains typical parameter values for an inverted pendulum.

Parameters	Symbol	Value	Unit
Mass of the Cart	M	3	kg
Mass of the i. p.	m	0.5	kg
Length of the i. p.	*l*	0.5	m
Friction Constant	b	2	kg/s
Gravitional Force	g	9.8	m/s^2

Table 1 Typical parameter values for an inverted pendulum.

Using the parameter values given in Table 1, the dynamic model becomes:

$$\dot{x}_1 = x_2$$
$$\dot{x}_2 = \frac{-2x_2 + 0.25\sin(x_3)x_4^2 - 4.9\sin(x_3)\cos(x_3) + u}{3.5 - 0.5\cos(x_3)^2}$$
$$\dot{x}_3 = x_4$$
$$\dot{x}_4 = \frac{(2x_2 - u - 0.25\sin(x_3)x_4^2)\cos(x_3) + 34.33\sin(x_3)}{1.75 - 0.25\cos(x_3)^2} \quad (15)$$

A stable sliding subspace for this model can be found as $G = [2 \quad 1.5 \quad 5 \quad 1.5]$. Regarding this subspace we design a sliding controller for this system using the NLP approach of the previous section. Then we simulate the inverted pendulum system with the the sliding mode controller obtained by the NLP tools when the initial conditions are perturbed from the equilibrium state $X = [0 \quad 0 \quad 0 \quad 0]^T$.

Let us choose the sliding band parameter as $\delta = 0.02$ and write the NLP problems associated with the reaching part (6) and the sliding part (7) as

$$\min_K \; GXG\dot{X} \quad \text{subject to} \begin{cases} GXG\dot{X} \le -|GX|\eta \\ u \in \{KX : |KX| \le \alpha\} \\ \mathrm{K} \in \Omega \subseteq \mathrm{R}^n \end{cases} \quad (16)$$

$$\min_K \left\| w + \frac{w^T \frac{\dot{X}}{\|\dot{X}\|}}{\|w\|} w \right\| \quad (17)$$
$$\text{subject to} \begin{cases} GXG\dot{X} \le -|GX|\eta \\ u \in \{KX : |KX| \le \alpha\} \\ \mathrm{K} \in \Omega \subseteq \mathrm{R}^n \end{cases}$$

where $\Omega = \{k_1, \ldots, k_4 : -11 \le k_i \le 11, i = 1,2,3,4\}, \alpha = 8$ $\eta = 0.005$ and $\dot{X}$ is given by expression (15). The second-simulation results for the perturbed initial condition $x^T = [0.1 \quad 0 \quad 0 \quad 0]$ are given in Figures 2 and 3.

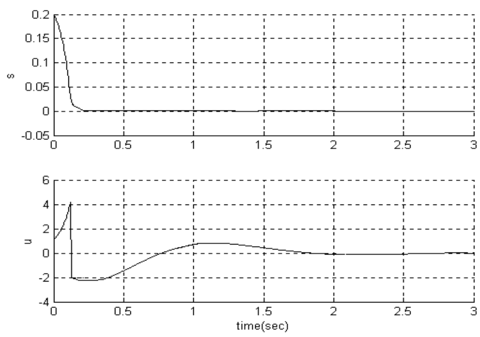

Figure 2 The trajectories relative position with respect to the sliding sub space and SMC input

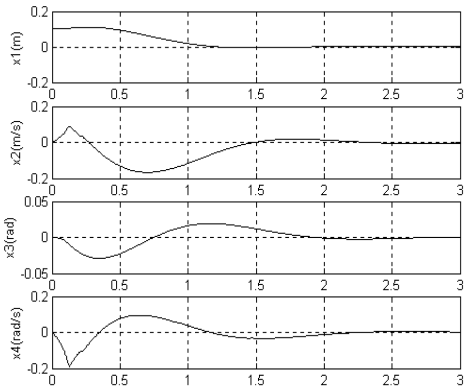

Figure 3 The states of the inverted pendulum versus time

Figures 2 and 3 show that starting from the pertubed initial conditions the states reach back to their equilibrium points. In other words, the trajectory reaches the sliding subspace and remains there in the subsequent times. The speed of the stabilization is comparable to that existing in the literature [17]. In the process of the stabilization the control input obeys the constraints given by (16) and (17).

IV. CONCLUSION

A novel algorithm to design a sliding mode controller using nonlinear programming techniques is introduced. It has been illustrated that in the NLP domain modelling the system specifications (i.e, constraints) is easier, and the solution procedure involves only the algebraic manipulations (not the differential equations) and more efficient. It is demonstrated that the algorithm works for stabilization of an inverted pendulum.

REFERENCES

1. Utkin VI. Variable structure systems with sliding modes. IEEE Transactions on Automatic Control 1977; 22(2): 212-222.

2. Slotine JJE, Sastry SS. Tracking control of nonlinear systems using sliding surfaces, with application to robot manipulators. International Journal of Control 1983; 38 (2): 465-492.

3. DeCarlo RA, Zak SH, Matthews GB. Variable structure control of nonlinear multivariable systems: A tutorial; Proc. IEEE 1998; 76 (3): 212-232.

4. Young KD, Utkin VI, Özgüner Ü. A control engineer's guide to sliding mode control. IEEE Transactions on Control Systems Technology 1999; 7 (3): 328-342.

5. Gasimov RN. Augmented Lagrangian duality and nondifferentiable optimization methods in nonconvex programming. Journal of Global Optimization 2002; 24: 187-203.

6. Rockafellar RT, Wets RJB. Variational Analysis. Springer-Berlin, 1998.

7. Bazaraa MS, Sherali HD, Shetty CM. Nonlinear Programming: Theory and Algoritms. John Wiley and Sons, 1993.

8. Bertsekas DP. Nonlinear Programming. Athena Scientific, Belmont, MA, 1995.

9. Azimov AY, Gasimov RN. On weak conjugacy, weak subdifferentials and duality with zero gap in nonconvex optimization. International Journal of Applied Mathematics, Academic Publications 1999; 1: 171-192

10. Rubinov AM, Gasimov RN. Strictly increasing positively homogeneous functions with applications to exact penalization. Optimization 2003; 52 (1): 1-28.

11. Gasimov RN, Rubinov AM. On augmented Lagrangians for optimization problems with a single constraint. Journal of Global Optimization to appear.

12. R. Gasimov, A. Karamancıoğlu, A. Yazıcı. On Nonlinear Programming Approaches for the Sliding Mode Control Systems. submitted to: International Journal of Robust and Nonlinear Control,2003.

13. Betts JT. Practical Methods for Optimal Control Using Nonlinear programming. Advances in Design and Control, SIAM, 2001.

14. Bartolini G, Ferrara A, Usai E, Utkin VI. On multi-input chattering free second order sliding mode control.

15. Iordanou H., Surgenor B. W. Experimental evaluation of the robustness of discrete sliding mode control versus linear quadratic control. IEEE Trans. Contr. Sys. Tech. 1997; c. 5, s. 254-260.

16. Awtar S., King N., Allen T., Bang I., Hagan M., Skidmore D., Craig K. Inverted pendulum systems: rotary and arm-driven –a mechatronic system design case study. Mechatronics 2002; s. 357-370.

17. Yazıcı A. Kayma Kipli Kontrol Yaklaşımlarının Ters Sarkaç Sistemine Uygulanması. Osmangazi Üniversitesi, Yüksek Lisans Tezi, Aralık 2000.

18. Ogata K. Modern Control Engineering. Prentice-Hall Inc. 1990.