CHAPTER 2

CONVENTIONAL SIMPLEX SLIDING-MODE CONTROL

As preliminaries, the conventional simplex sliding-mode control (SSMC) is briefly introduced in this chapter. In the beginning, the history and features of the conventional SSMC are described in Section 2.1. Then, Section 2.2 presents the definition and properties of the simplex set and Section 2.3 shows the design procedure of the SSMC.

2.1 Introduction

The design procedure of the sliding-mode control (SMC) is mainly divided into two steps [1-4]. In the first step, an appropriate sliding function $\boldsymbol{\sigma}$ should be suitably selected to make sure the system trajectory is stabilized in the sliding mode $\boldsymbol{\sigma} = \mathbf{0}$. In the second step, it is required to design the control algorithm to guarantee the reaching condition, which means the system trajectory would reach the sliding mode in a finite time and stay thereafter. For example, consider a system in a form as

$$\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}, t, \boldsymbol{u}) \tag{2.1}$$

where $\boldsymbol{x} \in \Re^n$ is the state, $\boldsymbol{u} \in \Re^m$ is the control input, and $\boldsymbol{f}$ represents a function which has integral in the domain of continuity of $\boldsymbol{u}$. For the SMC algorithm, it uses a discontinuous feedback control $\boldsymbol{u} = [u_1 \quad u_2 \quad \cdots \quad u_{\boldsymbol{m}}]$ where

$$u_i = \begin{cases} u_i^+(\boldsymbol{x}, t) & \text{for} \quad \sigma_i(\boldsymbol{x}, t) > 0 \\ u_i^-(\boldsymbol{x}, t) & \text{for} \quad \sigma_i(\boldsymbol{x}, t) < 0 \end{cases}, \qquad i = 1, 2, \cdots, m \tag{2.2}$$

and both $u_i^+(\boldsymbol{x},t)$ and $u_i^-(\boldsymbol{x},t)$ are continuous functions. However, these discontinuity surfaces $\sigma_i = 0, i = 1,\cdots,m$ divide the system space into 2^m sub-regions and 2^m distinct control vectors are generated correspondingly.

Based on a simplex set, the simplex sliding mode control (SSMC) was proposed to partition the system space into m+1 sub-regions and generate m+1 control vectors. Obviously, when comparing to the conventional SMC, the number of the sub-regions is reduced from 2^m to m+1, so is the number of the control vectors. The SSMC was first developed by Baida and Izosimov for multi-input continuous systems [18]. Then, Diong further extended the SSMC to linear multivariable systems [19] and Bartolini et al. applied the SSMC to nonlinear control systems with uncertainties [24-27]. Next, a general design procedure of the SSMC will be introduced.

2.2 Simplex Set

The main feature of the SSMC is to adopt a minimum number of distinct control vectors, called the simplex set. For a system with the control input $\boldsymbol{u} = [u_1 \quad u_2 \quad \cdots \quad u_m]^T \in \Re^{\boldsymbol{m}}$, Baida and Izosimov [18] proposed the simplex set as $S_U = \{\boldsymbol{u}^1, \boldsymbol{u}^2, \cdots, \boldsymbol{u}^{\boldsymbol{m}+1}\}$, where $\boldsymbol{u}^i = [u_1^i \quad u_2^i \quad \cdots \quad u_m^i]^T \in \Re^{\boldsymbol{m}}$ for $i = 1,2,\cdots,m+1$. The simplex set S_U must satisfy the two conditions as below:

(C1) $det\,\boldsymbol{U}_i \neq 0$, $i = 1,2,\cdots,m+1$, where $\boldsymbol{U}_i$ is a square matrix containing m column vectors without $\boldsymbol{u}^i$, expressed as

$$U_i = \begin{cases} \begin{bmatrix} u^2 & u^3 & \cdots & u^{m+1} \end{bmatrix} & i=1 \\ \begin{bmatrix} u^1 & \cdots & u^{i-1} & u^{i+1} & \cdots & u^{m+1} \end{bmatrix} & i=2,3,\cdots,m \\ \begin{bmatrix} u^1 & u^2 & \cdots & u^m \end{bmatrix} & i=m+1 \end{cases} \tag{2.3}$$

(C2) $\sum_{i=1}^{m+1} \psi_i u^i = 0$ where $\psi_i > 0$ and $\sum_{i=1}^{m+1} \psi_i = 1$

Note that (C1) implies that any m vectors in S_U are linearly independent and (C2) leads to the fact that any vector $v \in \Re^m$ can be expressed as $v = \sum_{j=1}^{m+1} \phi_j u^j$, which is not a unique express since any m vectors in S_U form a basis in $\Re^m$. For the SSMC, a unique express of v is required, which is purposely set as $v = \sum_{j=1,\, j\neq i}^{m+1} \phi_j u^j$ with $\phi_j > 0$ and $1 \le i \le m+1$.

Therefore, based on the simplex set $S_U = \{u^1, u^2, \cdots, u^{m+1}\}$ satisfying (C1) and (C2), the space $\Re^m$ can be divided into m+1 disjointed open sub-regions as below

$$\Sigma_i = \left\{ v \,\middle|\, v = \sum_{j=1,\, j\neq i}^{m+1} \phi_j u^j, \phi_j > 0 \right\}, \quad i = 1, \cdots, m+1 \tag{2.4}$$

where the vector $v \in \Re^m$ in Σ_i is uniquely determined as a linear combination of m vectors in S_U without u^i. For example, the vectors u^1, u^2, and u^3 illustrated in Figure 2.1 shows a simplex set for $m = 2$ with disjointed open sub-regions Σ_1, Σ_2, and Σ_3.

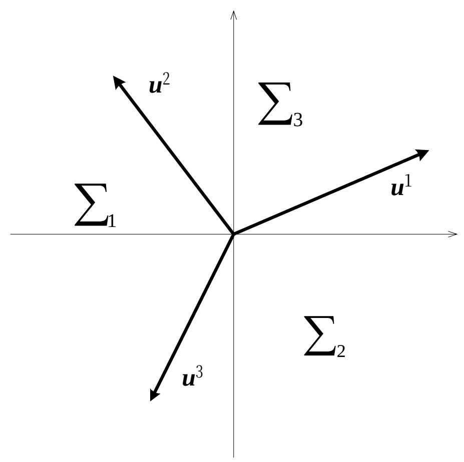

Figure 2.1 Simplex set for $m = 2$

2.3 Conventional Simplex Sliding-Mode Control Design

This section introduces the design procedure of the conventional SSMC using the simplex set described in Section 2.2 [19]. Consider a linear time-invariant system expressed as

$$\dot{\boldsymbol{x}} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{B}\boldsymbol{u} + \boldsymbol{B}\boldsymbol{d}(\boldsymbol{x},t) \tag{2.5}$$

where $\boldsymbol{x} \in \Re^n$ is the state, $\boldsymbol{u} \in \Re^m$ represents the control input, and $\boldsymbol{d}(\boldsymbol{x},t) \in \Re^m$ represents the matched disturbance. Without loss of generality, the pair $(\boldsymbol{A},\boldsymbol{B})$ is assumed to be controllable and $\boldsymbol{B}$ is of full rank.

Similar to the conventional SMC, the design procedure of the SSMC is divided into two steps [19]. The first step is to select an appropriate sliding vector $\boldsymbol{\sigma}$ such that the system is stabilized in the sliding mode $\boldsymbol{\sigma} = 0$. In the second step, the SSMC control algorithm is designed to guarantee that the system trajectory would reach the sliding mode in a finite time and then stay thereafter. Next, these two steps will be respectively discussed.

For the first step, the sliding vector $\boldsymbol{\sigma} \in \Re^{m}$ is generally selected as

$$\boldsymbol{\sigma} = \boldsymbol{C}\boldsymbol{x} \tag{2.6}$$

where $\boldsymbol{\sigma} = [\sigma_1 \quad \sigma_2 \quad \cdots \quad \sigma_{\boldsymbol{m}}]^{\boldsymbol{T}} \in \Re^{m}$ and $\boldsymbol{C} \in \Re^{m \times n}$ is a coefficient matrix. For the coefficient matrix ***C***, it must satisfy the additional condition that the $m \times m$ square matrix ***CB*** is invertible, i.e., $(\boldsymbol{CB})^{-1}$ exists. Note that the sliding vector (2.6) has been widely adopted in most of the sliding-mode control. Further, many approaches have been proposed to design the coefficient matrix ***C***, such as the transformation matrix method [41], the eigenstructure assignment method [42], and the Lyapunov-based method [43]. Fortunately, these developed approaches can be directly applied to the design of ***C*** for the SSMC since the SMC and SSMC have the same objective to guarantee the system stability in the sliding surface $\boldsymbol{\sigma} = \mathbf{0}$. Therefore, this section will not discuss how to design ***C***, but assume ***C*** could be suitably selected. Instead, this section will focus on the second step to design the SSMC algorithm such that the system trajectory could reach the sliding mode $\boldsymbol{\sigma} = \mathbf{0}$ in a finite time.

Now, for the system (2.5), taking the first derivative of the selected sliding vector (2.6) yields

$$\dot{\boldsymbol{\sigma}} = \boldsymbol{C}\dot{\boldsymbol{x}} = \boldsymbol{CAx} + \boldsymbol{CBu} + \boldsymbol{CBd}(\boldsymbol{x}, t) \tag{2.7}$$

To simplify the design process, the condition $\boldsymbol{CB} = \boldsymbol{I}$ is usually considered into the design of $\boldsymbol{C}$ [44,45]. For this reason, (2.7) becomes

$$\dot{\boldsymbol{\sigma}} = \boldsymbol{CAx} + \boldsymbol{u} + \boldsymbol{d}(\boldsymbol{x}, t) \tag{2.8}$$

Apparently, the truth of (2.8) implies the controller $\boldsymbol{u}$ can directly influence the variation of the sliding vector $\boldsymbol{\sigma}$. As a result, with the use of the simplex set described in Section 2.2, the SSMC control algorithm is designed as

$$\boldsymbol{u} = \boldsymbol{u}^{i}, \qquad \text{for } \boldsymbol{\sigma} \in \boldsymbol{\Sigma}_i \tag{2.9}$$

where $\boldsymbol{u}^i$ for $i = 1, 2, \cdots, m+1$ represents the component of the simplex set described in Section 2.2 and $\boldsymbol{\Sigma}_i$ belongs to the disjointed open sub-region given in (2.4). For illustration, Figure 2.2 shows the SSMC control algorithm (2.9) by using the simplex set with regard to the sub-regions $\boldsymbol{\Sigma}_1$, $\boldsymbol{\Sigma}_2$, and $\boldsymbol{\Sigma}_3$ for $m = 2$.

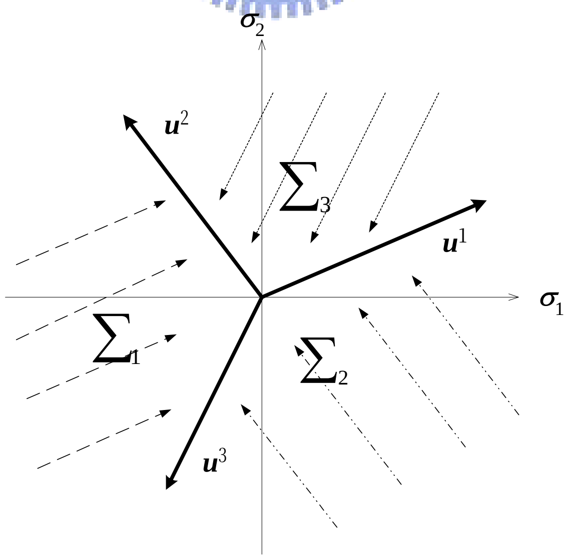

Figure 2.2 The SSMC control algorithm for $m = 2$

Next, let's show that the system (2.5) will be driven to the sliding mode $\boldsymbol{\sigma} = \boldsymbol{0}$ in a finite time by utilizing the control algorithm (2.9). To reach this goal, two important assumptions must be satisfied as follows:

Assumption 2.1 [19]: The vector $\boldsymbol{u}^i$ for $i = 1,2,\cdots,m+1$ in the simplex set S_U are selected so that none of the switching surfaces belongs to an unstable invariant subspace of the open-loop autonomous system, i.e. $\dot{\boldsymbol{x}} = \boldsymbol{A}\boldsymbol{x}$.

Assumption 2.2 [19]: There exist for all time $t \geq 0$, real numbers $\kappa_i(\boldsymbol{x},\ \boldsymbol{u}^i, t)$ and ξ such that the evolution of $\boldsymbol{x}$, as determined by (2.5) and (2.9), satisfies

$$\begin{aligned} &-\boldsymbol{CAx} - \boldsymbol{d}(\boldsymbol{x},t) = \sum_{i=1}^{m+1} \kappa_i \boldsymbol{u}^i,\ \kappa_i \geq 0 \\ &\sum_{i=1}^{m+1} \kappa_i = \xi,\ 0 \leq \xi \leq 1 \end{aligned} \tag{2.10}$$

where $\boldsymbol{u}^i$ for $i = 1,2,\cdots,m+1$ represents the simplex set $S_U = \{\boldsymbol{u}^1, \boldsymbol{u}^2, \cdots, \boldsymbol{u}^{m+1}\}$. Then, it is necessary to verify the closed-loop system described by (2.5) and (2.9) will achieve sliding mode $\boldsymbol{\sigma} = 0$ in a finite time when the above two assumptions are satisfied.

Without loss of generality, let's consider the following case that the sliding vector $\boldsymbol{\sigma}$ is currently in Σ_p, $1 \leq p \leq m+1$. With the use of the simplex set, the sliding vector $\boldsymbol{\sigma}$ can be expressed as

$$\boldsymbol{\sigma} = \sum_{j=1,\ j \neq p}^{m+1} \phi_j \boldsymbol{u}^j, \quad \phi_j > 0 \tag{2.11}$$

Then, the candidate of Lyapunov function can be selected as

$$\boldsymbol{V}(\boldsymbol{\sigma}) = \sum_{j=1,\ j \neq p}^{m+1} \phi_j \tag{2.12}$$

In fact, it is easy to find that $\boldsymbol{V}(\boldsymbol{\sigma}) = 0$ as $\boldsymbol{\sigma} = 0$ and $\boldsymbol{V}(\boldsymbol{\sigma}) = \sum_{j=1,\ j \neq p}^{m+1} \phi_j > 0$ for any $\boldsymbol{\sigma} \neq 0$. Besides, it has been shown that $\boldsymbol{V}$ is continuous, i.e., $\boldsymbol{V}$ doesn't change discontinuously when $\boldsymbol{\sigma}$ switches from Σ_p to the other sub-regions Σ_j, $j \neq p$ [18]. For this reason, $\boldsymbol{V}(\boldsymbol{\sigma})$ can be treated as a Lyapunov function. On the other hand, the truth of $\sum_{i=1}^{m+1} \psi_i \boldsymbol{u}^i = 0$, $\psi_i > 0$ in (C2) implies the control input $\boldsymbol{u}^p$ can be represented as

$$\begin{aligned} \boldsymbol{u}^p &= -\sum_{j=1,\, j\neq p}^{m+1} \frac{\psi_j}{\psi_p} \boldsymbol{u}^j \\ &= -\sum_{j=1,\, j\neq p}^{m+1} \gamma_j \boldsymbol{u}^j, \quad \text{where } \gamma_j = \frac{\psi_j}{\psi_p} > 0 \end{aligned} \tag{2.13}$$

By virtue of (2.3), (2.11), (2.12), and (2.13) are rewritten as

$$\boldsymbol{\sigma} = \sum_{j=1,\, j\neq p}^{m+1} \phi_j \boldsymbol{u}^j = \boldsymbol{U}_p \boldsymbol{\phi}_p \tag{2.14}$$

$$\boldsymbol{V} = \sum_{j=1,\, j\neq p}^{m+1} \phi_j = \boldsymbol{g}^T \boldsymbol{\phi}_p \tag{2.15}$$

$$\boldsymbol{u}^p = -\sum_{j=1,\, j\neq p}^{m+1} \gamma_j \boldsymbol{u}^j = -\boldsymbol{U}_p \boldsymbol{\gamma}_p \tag{2.16}$$

where

$$\begin{aligned} \boldsymbol{\phi}_p &= \begin{bmatrix} \phi_1 & \cdots & \phi_{p-1} & \phi_{p+1} & \cdots & \phi_{m+1} \end{bmatrix}^T \\ \boldsymbol{g} &= \begin{bmatrix} 1 & \cdots & 1 & 1 & \cdots & 1 \end{bmatrix}_{1\times m} \\ \boldsymbol{\gamma}_p &= \begin{bmatrix} \gamma_1 & \cdots & \gamma_{p-1} & \gamma_{p+1} & \cdots & \gamma_{m+1} \end{bmatrix}^T \end{aligned}$$

With use of Assmuption 2.2, (2.10) can be derived as

$$\begin{aligned} \boldsymbol{CAx} + \boldsymbol{d}(\boldsymbol{x}, t) &= -\sum_{i=1}^{m+1} \kappa_i \boldsymbol{u}^i = -\boldsymbol{U}_p \boldsymbol{\kappa}_p - \kappa_p \boldsymbol{u}^p \\ &= -\boldsymbol{U}_p \boldsymbol{\kappa}_p + \kappa_p \boldsymbol{U}_p \boldsymbol{\gamma}_p \end{aligned} \tag{2.17}$$

where $\boldsymbol{\kappa}_p = \begin{bmatrix} \kappa_1 & \cdots & \kappa_{p-1} & \kappa_{p+1} & \cdots & \kappa_{m+1} \end{bmatrix}^T$. By using (2.14), further taking the first derivative of the sliding vector $\boldsymbol{\sigma}$ yields

$$\dot{\boldsymbol{\sigma}} = \boldsymbol{U}_p \dot{\boldsymbol{\phi}}_p \tag{2.18}$$

Based on (2.8), (2.16), and (2.17), (2.18) is rearranged as

$$\begin{aligned} \dot{\boldsymbol{\phi}}_p &= \boldsymbol{U}_p^{-1}\dot{\boldsymbol{\sigma}} = \boldsymbol{U}_p^{-1}\left(\boldsymbol{CAx} + \boldsymbol{u}^p + \boldsymbol{d}(\boldsymbol{x},t)\right) \\ &= \boldsymbol{U}_p^{-1}\left(-\boldsymbol{U}_p\boldsymbol{\kappa}_p + \kappa_p\boldsymbol{U}_p\boldsymbol{\gamma}_p - \boldsymbol{U}_p\boldsymbol{\gamma}_p\right) \\ &= -\boldsymbol{\kappa}_p + \kappa_p\boldsymbol{\gamma}_p - \boldsymbol{\gamma}_p, \qquad \text{for } \boldsymbol{\sigma} \in \Sigma_{\mathrm{p}} \end{aligned} \tag{2.19}$$

With the use of the matrix form, (2.19) becomes

$$\begin{bmatrix} \dot{\phi}_1 \\ \vdots \\ \dot{\phi}_{p-1} \\ \dot{\phi}_{p+1} \\ \vdots \\ \dot{\phi}_{m+1} \end{bmatrix} = -\begin{bmatrix} \kappa_1 \\ \vdots \\ \kappa_{p-1} \\ \kappa_{p+1} \\ \vdots \\ \kappa_{m+1} \end{bmatrix} + \kappa_p \begin{bmatrix} \gamma_1 \\ \vdots \\ \gamma_{p-1} \\ \gamma_{p+1} \\ \vdots \\ \gamma_{m+1} \end{bmatrix} - \begin{bmatrix} \gamma_1 \\ \vdots \\ \gamma_{p-1} \\ \gamma_{p+1} \\ \vdots \\ \gamma_{m+1} \end{bmatrix} \tag{2.20}$$

Now, further taking the first derivative of (2.12) yields

$$\dot{\boldsymbol{V}} = \dot{\phi}_1 + \cdots + \dot{\phi}_{p-1} + \dot{\phi}_{p+1} + \cdots + \dot{\phi}_{m+1} \tag{2.21}$$

By means of the row-by-row summation of (2.20), (2.21) leads to

$$\begin{aligned} \dot{\boldsymbol{V}} &= -\sum_{j=1,\, j\neq p}^{m+1}\kappa_j + \kappa_p \sum_{j=1,\, j\neq p}^{m+1}\gamma_j - \sum_{j=1,\, j\neq p}^{m+1}\gamma_j \\ &= -\sum_{j=1,\, j\neq p}^{m+1}\kappa_j - \left(1-\kappa_p\right)\sum_{j=1,\, j\neq p}^{m+1}\gamma_j \end{aligned} \tag{2.22}$$

According to Assumption 2.1, it implies $\sum_{j\neq p, j=1}^{m+1}\kappa_j > 0$ and $\left(1-\kappa_p\right) > 0$. Based on the truth of (2.13) and (2.22), it obviously shows

$$\dot{\boldsymbol{V}} = -\sum_{j\neq p, j=1}^{m+1}\kappa_j - \left(1-\kappa_p\right)\sum_{j\neq p, j=1}^{m+1}\gamma_j < 0 \tag{2.23}$$

Therefore, $\boldsymbol{V}$ decreases all the time and will decrease to zero in a finite time. In other words, it is evident that the closed-loop system described by (2.5) and (2.9) will achieve sliding mode $\sigma = 0$ in a finite time when Assumption 2.1 is satisfied.

Although the SSMC algorithm has been theoretically derived, some problems still exist and should be further improved. First, though the definition of the simplex set is explicitly given in (C1) and (C2), the conventional SSMC doesn't provide any method for selecting an appropriate simplex set. Second, to fulfill the SSMC algorithm (2.9), it is necessary to determine the sub-region where the current sliding vector σ belongs, i.e. $\sigma \in \Sigma_i$. For this step, it would become a thorny problem as the number of inputs is highly increased and makes the implementation of an SSMC more difficult and sometimes infeasible. The third problem is how to suitably eliminate the chattering which exists in the SSMC. Actually, it is inevitably faced with the chattering phenomenon for the SSMC due to the use of switching functions. The chattering caused by the SSMC happens not only in the sliding mode, but also during the approach mode. Next, we will attempt to solve the three problems by employing a specific simplex set, uniformly distributed simplex set.