

Research Article

Open Access



Operational control with set-points tuning - application to mobile robots

Xiaomo Yan¹, Hong Wang²

¹The School of Electrical Engineering and Electronics, The University of Manchester, Manchester M6- 1QD, UK.

²Oak Ridge National Laboratory, Oak Ridge, TN 37831, USA.

Correspondence to: Prof. Hong Wang, Oak Ridge National Laboratory, Oak Ridge, TN 37831, USA. E-mail: mikewanguk@yahoo.co.uk; ORCID: 0000-0002-9876-0176

How to cite this article: Yan X, Wang H. Operational control with set-points tuning - application to mobile robots. *Intell Robot* 2023;3(2):113-30. <http://dx.doi.org/10.20517/ir.2023.06>

Received: 30 Jan 2023 **First Decision:** 7 Mar 2023 **Revised:** 27 Mar 2023 **Accepted:** 4 Apr 2023 **Published:** 21 Apr 2023

Academic Editor: Simon X. Yang **Copy Editor:** Ying Han **Production Editor:** Ying Han

Abstract

This paper proposes a novel method for the optimal tuning of set points for multiple-layered control system structure widely seen in robotics and other complex industrial processes composed of a number of subsystems. The terminal sliding mode control (SMC) is used as the low-level control strategy to ensure the stability of subsystems. When uncertainties exist, it can be shown that the deteriorated system performance will be improved by the outer loop with set points tuning. For this purpose, the learning of the new set point is designed to compensate for the effects caused by uncertainties during the system operation. At the same time, the system is proven to stay with the original set point when the compensation is introduced. A practical application to a holonomic mobile robot system is given to illustrate the presented method. Desired results have been obtained.

Keywords: Set point reselection, mobile robot, terminal sliding mode control, hamilton-jacobi-bellman(HJB) equation

1. INTRODUCTION

Conventional optimal control has focused on improving system performance by optimizing controller parameters. As the development of control theory grows and the demand for industrial automation strengthens, controllers are embedded in industrial equipment universally. However, change to controller parameters becomes more difficult and costly as a result of universal use. Instead of changing controller parameters, system per-



© The Author(s) 2023. **Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License (<https://creativecommons.org/licenses/by/4.0/>), which permits unrestricted use, sharing, adaptation, distribution and reproduction in any medium or format, for any purpose, even commercially, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made.



formance can also be improved by reselecting the set point^[1-4]. This is particularly true for the multi-layered operational structure of control systems widely seen in robotics and complex industrial processes. Such systems generally consist of two layers - one is the low-level control systems with a number of subsystems, and the other is the operational control layer that generate set points to the subsystems so as to ensure a desired overall system performance. The new set point is designed based on prior knowledge of the system structure. The set point reselection method is achieved by using a cascade structure, which is widely used in complex industrial control strategy design^[5-7]. The inherent control loop is considered to be the inner loop, which guarantees the system's stability. Then system's performance will be improved by the outer loop.

Sliding mode control (SMC) is known for its outstanding robustness^[8-15]. By choosing the proper sliding manifold, a sliding mode controller is capable of guaranteeing system convergence. The chattering phenomenon has long been a notable issue in the SMC design^[16-18]. The high-frequency components will not only affect the system response but also damage the actuators. To eliminate the chattering phenomenon, smooth functions, such as the saturation function, are usually used in the SMC design^[19]. However, when the sliding manifold moves within the scope of a chosen linear portion, the amplifier from the saturation function slows the response. In 2012, Polyakov pointed out a certain type of function^[20] that guarantees a fast arrival with both large and small variances. In the conventional SMC design, a fast response relies on a large control gain, which is another incentive for the chattering issue. Instead of using a linear function, nonlinear sliding manifolds give a more flexible response^[21-24], but it is necessary to check if singular points exist in the system.

To overcome the above difficulties, in this work we focus on the tuning of the set-points to all the subsystems in the considered multi-layered operational system structure. The idea is to tuning these set points so that the system can still achieve the originally targeted performance when the system is subjected to unexpected uncertainties.

Therefore the novelties and contributions are as follows:

- (1) The system considered is of a multi-layered structure that consists of 1) subsystem layers which are low-level control system, and 2) operational layer that generate set points to these subsystems by optimizing a global system performance;
- (2) The proposed method aims at achieving optimal tuning of the set-points so as to guarantee the original optimized performance when the system is subjected to unexpected uncertainties rather than ask the subsystems to follow the newly updated set-points;
- (3) Application to a mobile robot system has been made - showing encouraging results in terms of performance guarantee.

This paper is organized in the following sections. In Section 2, the discrete time kinematic system is described. The inherent control strategy is terminal SMC. The stability of inner loops is also checked. This description is followed by the augmented representation of the state variables and system uncertainties. An optimal outer loop design is proposed in Section 3. Finally, the method proposed in this paper is tested on a holonomic mobile robot in Section 4 to validate system dynamics and quantitatively compare the system's performance with and without an outer loop. Conclusions are given in section 5.

2. METHODS

Because most of the industrial processes are controlled by computers, discrete system dynamics are considered in the design.

2.1. System model

Consider the system described by the following expression:

$$\begin{cases} \mathbf{x}_{1,k+1} = \mathbf{x}_{1,k} + h\mathbf{x}_{2,k}, \\ \mathbf{x}_{2,k+1} = \mathbf{x}_{2,k} + h(\mathbf{f}_k + \mathbf{g}_k\mathbf{u}_k), \end{cases} \tag{1}$$

where h is the system sampling interval, $\mathbf{x}_{i,k} = \mathbf{x}_i(kh)$, $\mathbf{x}_k = (\mathbf{x}_{1,k} \ \mathbf{x}_{2,k})^T \in \mathbb{R}^{2n}$ is the system states vector, $\mathbf{f}_k \in \mathbb{R}^n$ represents the nonlinear parts of the system, and $\mathbf{g}_k \in \mathbb{R}^n$ is an invertible matrix, which represents the relationships between control input \mathbf{u}_k and system states \mathbf{x}_k .

It can be seen that such system expression would represent a wide-range of dynamics systems in practice, and examples are robotic systems and complicated industrial processes such as paper making, mineral processing, chemical plant and car manufacturing systems.

The errors between system states and a set point vector can be expressed as

$$\mathbf{e}_{1,k} = \mathbf{x}_{1,k} - \mathbf{r}_k^*, \tag{2}$$

where \mathbf{r}_k^* is the set point vector. Then,

$$\mathbf{e}_{1,k+1} = \mathbf{x}_{1,k+1} - \mathbf{r}_{k+1}^*. \tag{3}$$

The difference between errors of $k + 1$ and k can be expressed as

$$\mathbf{e}_{1,k+1} - \mathbf{e}_{1,k} = h\mathbf{x}_{2,k} - (\mathbf{r}_{k+1}^* - \mathbf{r}_k^*). \tag{4}$$

In system expression Equation (1), $\mathbf{e}_{2,k}$ describes the change from $\mathbf{e}_{1,k}$ to $\mathbf{e}_{1,k+1}$. Let

$$\mathbf{e}_{2,k} = \mathbf{x}_{2,k} - \frac{1}{h}(\mathbf{r}_{k+1}^* - \mathbf{r}_k^*). \tag{5}$$

Then, the error dynamics can be written as

$$\begin{cases} \mathbf{e}_{1,k+1} = \mathbf{e}_{1,k} + h\mathbf{e}_{2,k}, \\ \mathbf{e}_{2,k+1} = \mathbf{e}_{2,k} + h(\mathbf{f}_k + \mathbf{g}_k\mathbf{u}_k) + \frac{1}{h}(-\mathbf{r}_{k+2}^* + 2\mathbf{r}_{k+1}^* - \mathbf{r}_k^*), \end{cases} \tag{6}$$

which is going to be used in the following control strategy design. In this context, the system in (1) represents the subsystems in a multi-layered system operational structure. The set points grouped in the set point vector is generated by the operational layer as shown in the following figure.

2.2. Terminal sSliding mode controller design

In this subsection, we will focus on the control system design for subsystems represented by equation (1). This requires the selection of control input \mathbf{u}_k to ensure the tracking of the system state to the set point. For this purpose, the well-known sliding mode control (SMC) strategy is used to guarantee the stability of the system. Based on the contribution made by Man et al. [25], to achieve the terminal convergence, the nonsingular sliding manifold is defined as

$$\mathbf{s}_k = \begin{pmatrix} s_{1,k} \\ s_{2,k} \\ \vdots \\ s_{n,k} \end{pmatrix} = \begin{pmatrix} e_{2,1,k} + \beta_1 e_{1,1,k}^{\frac{q_1}{p_1}} \\ e_{2,2,k} + \beta_2 e_{1,2,k}^{\frac{q_2}{p_2}} \\ \vdots \\ e_{2,n,k} + \beta_n e_{1,n,k}^{\frac{q_n}{p_n}} \end{pmatrix}, \tag{7}$$

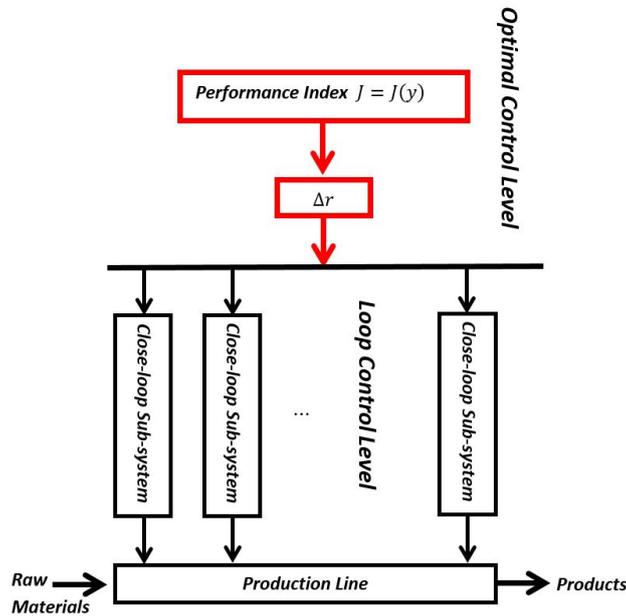


Figure 1. Block diagram of two-layer control structure.

where s_i is the i th sliding surface for the subsystem, $\beta_i \in \mathbb{R}$ is a positive definite, p_i and q_i are positive definite odd integers satisfying $\frac{q_i}{p_i} \in (1, \infty)$, and $e_{i,j,k}$ denotes the k th sample of the j th component in the i th state. The sliding manifold is a constraint to the system states, which drives the system's states along a certain trajectory to an origin point in finite time. System states move along the sliding manifold reproduced

$$s_{i,k+1} = s_{i,k} = 0. \tag{8}$$

Then, by combining Equation (7) and Equation (8) together, the equivalent control law to keep system states on the sliding manifold (7) is given by

$$u_{i,k} = -g_{i,k}^{-1} f_{i,k} + g_{i,k}^{-1} \frac{1}{h} [\beta_i e_{1,i,k}^{\frac{q_i}{p_i}} - \beta_i (e_{1,i,k} + h e_{2,i,k})^{\frac{q_i}{p_i}}] + g_{i,k}^{-1} \frac{1}{h^2} (r_{i,k+2}^* - 2r_{i,k+1}^* + r_{i,k}^*), \tag{9}$$

where $u_{i,k}$ represents the i th controller component based on the states value at the k th sampling time. However, if a system's initial values (x_0) are not placed on the manifold ($s_k = 0$), the controller shown in Equation (9) cannot transport system states to the sliding manifold. Based on [26], the new controller is written as

$$u_{i,k} = -g_{i,k}^{-1} f_{i,k} + g_{i,k}^{-1} \frac{1}{h} [\beta_i e_{1,i,k}^{\frac{q_i}{p_i}} - \beta_i (e_{1,i,k} + h e_{2,i,k})^{\frac{q_i}{p_i}}] + g_{i,k}^{-1} \frac{1}{h^2} (r_{i,k+2}^* - 2r_{i,k+1}^* + r_{i,k}^*) - g_{i,k}^{-1} \frac{1}{h} (\alpha_{1,i} s_{i,k}^{\frac{n_{1,i}}{m_{1,i}}} + \alpha_{2,i} s_{i,k}^{\frac{n_{2,i}}{m_{2,i}}}), \tag{10}$$

where $m_{1,i}, m_{2,i}, n_{1,i}, n_{2,i}$ are positively defined odd integers that satisfy $\frac{n_{1,i}}{m_{1,i}} \in (0, 1)$ and $\frac{n_{2,i}}{m_{2,i}} \in (1, \infty)$. When the controller described in Equation (10) is implemented into the system, error dynamics change to

$$\begin{cases} e_{1,i,k+1} = e_{1,i,k} + he_{2,i,k}, \\ e_{2,i,k+1} = e_{2,i,k} + \beta_i e_{1,i,k}^{\frac{q_i}{p_i}} - \beta_i (e_{1,i,k} + he_{2,i,k})^{\frac{q_i}{p_i}} \\ \quad - (\alpha_{1,i} s_{i,k}^{\frac{n_{1,i}}{m_{1,i}}} + \alpha_{2,i} s_{i,k}^{\frac{n_{2,i}}{m_{2,i}}}). \end{cases} \tag{11}$$

To prove the stability of the inner closed-loop system (i.e., the subsystems), the convergence of system states must be analyzed. This is summarized in the following lemma.

Lemma 1 For the error dynamics shown in Equation (6), if the controller is designed as

$$\begin{aligned} u_{i,k} = & -g_{i,k}^{-1} f_{i,k} + g_{i,k}^{-1} \frac{1}{h} [\beta_i e_{1,i,k}^{\frac{q_i}{p_i}} - \beta_i (e_{1,i,k} \\ & + he_{2,i,k})^{\frac{q_i}{p_i}}] + g_{i,k}^{-1} \frac{1}{h^2} (r_{i,k+2}^* - 2r_{i,k+1}^* \\ & + r_{i,k}^*) - g_{i,k}^{-1} \frac{1}{h} (\alpha_{1,i} s_{i,k}^{\frac{n_{1,i}}{m_{1,i}}} + \alpha_{2,i} s_{i,k}^{\frac{n_{2,i}}{m_{2,i}}}), \end{aligned} \tag{12}$$

where $m_{1,i}, m_{2,i}, n_{1,i}, n_{2,i}$ are positive definite odd integers that satisfy $\frac{n_{1,i}}{m_{1,i}} \in (0, 1)$ and $\frac{n_{2,i}}{m_{2,i}} \in (1, \infty)$, then the system state will converge to zero within a finite time.

Proof 1 By using the sliding manifold described in Equation (7), system states go along the sliding manifold ($s_i = 0$), and when they arrive at it,

$$s_{i,k+1} = s_{i,k} = 0. \tag{13}$$

However, the system states are not always on the manifold. To make the sliding manifold an attractor, the system states must be ensured to move toward the manifold when the system states are outside the sliding manifold ($s_i \neq 0$), which means that

$$s_{i,k}(s_{i,k+1} - s_{i,k}) < 0. \tag{14}$$

If it can be proved that $(s_{i,k+1} - s_{i,k})$ has an opposite sign to the sliding manifold ($s_{i,k}$), then system stability can be proved. Combining the sliding manifold with Equation (14),

$$\begin{aligned} & s_{i,k}(s_{i,k+1} - s_{i,k}) \\ & = (e_{2,i,k+1} + \beta_i e_{1,i,k+1}^{\frac{q_i}{p_i}} - e_{2,i,k} - \beta_i e_{1,i,k}^{\frac{q_i}{p_i}}) s_{i,k} \\ & = [e_{2,i,k} + \beta_i e_{1,i,k}^{\frac{q_i}{p_i}} - \beta_i (e_{1,i,k} + he_{2,i,k})^{\frac{q_i}{p_i}} \\ & \quad - (\alpha_{1,i} s_{i,k}^{\frac{n_{1,i}}{m_{1,i}}} + \alpha_{2,i} s_{i,k}^{\frac{n_{2,i}}{m_{2,i}}}) + \beta_i (e_{1,i,k} \\ & \quad + he_{2,i,k})^{\frac{q_i}{p_i}} - e_{2,i,k} - \beta_i e_{1,i,k}^{\frac{q_i}{p_i}}] s_{i,k} \\ & = - (\alpha_{1,i} s_{i,k}^{\frac{n_{1,i}}{m_{1,i}}+1} + \alpha_{2,i} s_{i,k}^{\frac{n_{2,i}}{m_{2,i}}+1}). \end{aligned} \tag{15}$$

As described previously, $m_1, m_2, n_1,$ and n_2 are positive definite odd integers, which guarantees the numerators of $\frac{n_{j,i}}{m_{j,i}} + 1$ are even. An even power, $(n_{j,i} + m_{j,i})$, will erase the sign. $\alpha_{1,i}$ and $\alpha_{2,i}$ are also positive definite real numbers. The derivative of the Lyapunov candidate presented here is less than zero as long as $s_{i,k} \neq 0$; so,

$$s_{i,k}(s_{i,k+1} - s_{i,k}) \leq 0. \tag{16}$$

Zero is the locally stable equilibrium point of the system. Basing on these results, $|s_{i,k+1}| < |s_{i,k}|$ when $|s_{i,k}| \neq 0$, and $s \rightarrow 0$ when $t \rightarrow 0$. Then,

$$e_{2,i,k} = -\beta_i e_{1,i,k}^{\frac{q_1}{p_1}}. \tag{17}$$

The system is globally asymptotically stable.

2.3. Set point reselection method

2.3.1. System model with distribution

The system dynamics with uncertainties are

$$\begin{cases} \mathbf{x}_{1,k+1} = \mathbf{x}_{1,k} + h\mathbf{x}_{2,k} + \mathbf{d}_1, \\ \mathbf{x}_{2,k+1} = \mathbf{x}_{2,k} + h(\mathbf{f}_k + \mathbf{g}_k \mathbf{u}_k) + \mathbf{d}_2, \end{cases} \quad (18)$$

where \mathbf{d}_1 and \mathbf{d}_2 are uncertainties in the system. To improve the performance of the system, an additional reselected set point vector $\Delta \mathbf{r}_\zeta$ is introduced. The new reference signal is

$$\mathbf{r}_k = \Delta \mathbf{r}_\zeta + \mathbf{r}_k^* \quad (19)$$

where \mathbf{r}_k^* is the origin reference vector that system states want to track, $\Delta \mathbf{r}$ is the new compensatory set point to be designed, and $(\bullet)_\zeta$ shows a different sampling rate with the system states. Because of the introduction of a new set point vector, some new vectors need to be defined to emphasize the difference between new states error and the one used in Section 2.

$$\begin{cases} \boldsymbol{\varepsilon}_{1,k} = \mathbf{x}_{1,k} - \mathbf{r}_k \\ \boldsymbol{\varepsilon}_{2,k} = \mathbf{x}_{2,k} - \frac{1}{h}(\mathbf{r}_{k+1} - \mathbf{r}_k). \end{cases} \quad (20)$$

Based on the new vectors defined above, system dynamics change to

$$\begin{cases} \boldsymbol{\varepsilon}_{1,k+1} = \boldsymbol{\varepsilon}_{1,k} + h\boldsymbol{\varepsilon}_{2,k} + \mathbf{d}_1, \\ \boldsymbol{\varepsilon}_{2,k+1} = \boldsymbol{\varepsilon}_{2,k} + h(\mathbf{f}_k + \mathbf{g}_k \hat{\mathbf{u}}_k) + \frac{1}{h}(-\mathbf{r}_{k+2} \\ + 2\mathbf{r}_{k+1} - \mathbf{r}_k) + \mathbf{d}_2, \end{cases} \quad (21)$$

where $\boldsymbol{\varepsilon} = (\boldsymbol{\varepsilon}_1 \quad \boldsymbol{\varepsilon}_2)^T$ is the new state of the error dynamics. $\hat{\mathbf{u}}$ is used to represent the controller based on the new error dynamics described in Equation (26). h is the length of the system states' sampling interval. Based on the error dynamics described in Equation (26), controller ($\hat{\mathbf{u}}_{i,k}$) is expressed as

$$\begin{aligned} \hat{\mathbf{u}}_{i,k} = & -g_{i,k}^{-1} f_{i,k} + g_{i,k}^{-1} \frac{1}{h} [\beta_i \boldsymbol{\varepsilon}_{1,i,k}^{\frac{q_i}{p_i}} - \beta_i (\boldsymbol{\varepsilon}_{1,i,k} \\ & + h\boldsymbol{\varepsilon}_{2,i,k})^{\frac{q_i}{p_i}}] + g_{i,k}^{-1} \frac{1}{h^2} (\mathbf{r}_{i,k+2}^* - 2\mathbf{r}_{i,k+1}^* \\ & + \mathbf{r}_{i,k}^*) - g_{i,k}^{-1} \frac{1}{h} (\alpha_{1,i} \hat{\boldsymbol{\varepsilon}}_{1,i,k}^{\frac{n_{1,i}}{m_{1,i}}} + \alpha_{2,i} \hat{\boldsymbol{\varepsilon}}_{2,i,k}^{\frac{n_{2,i}}{m_{2,i}}}), \end{aligned} \quad (22)$$

where $\hat{\boldsymbol{\varepsilon}}_{i,k} = \boldsymbol{\varepsilon}_{2,i,k} + \beta_i \boldsymbol{\varepsilon}_{1,i,k}^{\frac{q_i}{p_i}}$ is the sliding manifold related to the new system state vector $\boldsymbol{\varepsilon}$. $f_{i,k} = f_i(\boldsymbol{\varepsilon}_k)$ and $\boldsymbol{\varepsilon}_{j,i,k} = \boldsymbol{\varepsilon}_{j,i}(kh)$. Similar to the previous steps, by combining system dynamics described in Equation (27) and controller $\hat{\mathbf{u}}_{i,k}$, the system dynamics become

$$\begin{cases} \boldsymbol{\varepsilon}_{1,k+1} = \boldsymbol{\varepsilon}_{1,k} + h\boldsymbol{\varepsilon}_{2,k} + \mathbf{d}_1 \\ \boldsymbol{\varepsilon}_{2,k+1} = \boldsymbol{\varepsilon}_{2,k} + \beta \boldsymbol{\varepsilon}_{1,k}^{\frac{q}{p}} - \beta (\boldsymbol{\varepsilon}_{1,k} + h\boldsymbol{\varepsilon}_{2,k})^{\frac{q}{p}} + \mathbf{d}_2 \\ + \frac{1}{h} (-\Delta \mathbf{r}_{k+2} + \Delta 2\mathbf{r}_{k+1} - \Delta \mathbf{r}_k). \end{cases} \quad (23)$$

2.4. System performance optimization

This section details a new method of selecting an alternative set point to improve the system performance. In this context, the system model with disturbances will be presented first. This is then followed by the proposed method on the set point tuning that guarantees the achievement of the originally targeted system performance.

2.4.1. System model with distribution

The system dynamics with uncertainties or disturbances can be expressed by adding extra terms in equation (1) as follows.

$$\begin{cases} \mathbf{x}_{1,k+1} = \mathbf{x}_{1,k} + h\mathbf{x}_{2,k} + \mathbf{d}_1, \\ \mathbf{x}_{2,k+1} = \mathbf{x}_{2,k} + h(\mathbf{f}_k + \mathbf{g}_k\mathbf{u}_k) + \mathbf{d}_2, \end{cases} \quad (24)$$

where \mathbf{d}_1 and \mathbf{d}_2 are uncertainties or disturbances in the system.

To maintain and improve the performance of the system when the system is subjected to uncertainties and disturbances, we propose to tune the set points rather than changing the control parameters. For this purpose, an additionally reselected set point vector $\Delta\mathbf{r}_\zeta$ is introduced. The means that new set point vector should be of the following form

$$\mathbf{r}_k = \Delta\mathbf{r}_\zeta + \mathbf{r}_k^* \quad (25)$$

where \mathbf{r}_k^* is the original reference vector that system states want to track, $\Delta\mathbf{r}$ is the new compensatory set point to be designed as an additional term onto the original set point vector, and $(\bullet)_\zeta$ shows a different sampling rate with the system states. Because of the introduction of a new set point vector, some new vectors need to be defined to emphasize the difference between error from new states and the one used in Section 2.

$$\begin{cases} \boldsymbol{\varepsilon}_{1,k} = \mathbf{x}_{1,k} - \mathbf{r}_k \\ \boldsymbol{\varepsilon}_{2,k} = \mathbf{x}_{2,k} - \frac{1}{h}(\mathbf{r}_{k+1} - \mathbf{r}_k). \end{cases} \quad (26)$$

Based on the new vectors defined previously, system dynamics change to

$$\begin{cases} \boldsymbol{\varepsilon}_{1,k+1} = \boldsymbol{\varepsilon}_{1,k} + h\boldsymbol{\varepsilon}_{2,k} + \mathbf{d}_1, \\ \boldsymbol{\varepsilon}_{2,k+1} = \boldsymbol{\varepsilon}_{2,k} + h(\mathbf{f}_k + \mathbf{g}_k\hat{\mathbf{u}}_k) + \frac{1}{h}(-\mathbf{r}_{k+2} + 2\mathbf{r}_{k+1} - \mathbf{r}_k) + \mathbf{d}_2, \end{cases} \quad (27)$$

where $\boldsymbol{\varepsilon} = (\boldsymbol{\varepsilon}_1 \ \boldsymbol{\varepsilon}_2)^T$ is the new state of the error dynamics, $\hat{\mathbf{u}}$ is used to represent the controller based on the new error dynamics described in Equation (26), and h is the length of the system states sampling interval. Based on the error dynamics described in Equation (26), controller ($\hat{\mathbf{u}}_{i,k}$) is expressed as

$$\begin{aligned} \hat{\mathbf{u}}_{i,k} = & -g_{i,k}^{-1}f_{i,k} + g_{i,k}^{-1}\frac{1}{h}[\beta_i\varepsilon_{1,i,k}^{\frac{q_i}{p_i}} - \beta_i(\varepsilon_{1,i,k} \\ & + h\varepsilon_{2,i,k})^{\frac{q_i}{p_i}}] + g_{i,k}^{-1}\frac{1}{h^2}(r_{i,k+2}^* - 2r_{i,k+1}^* \\ & + r_{i,k}^*) - g_{i,k}^{-1}\frac{1}{h}(\alpha_{1,i}\hat{\delta}_{i,k}^{\frac{n_{1,i}}{m_{1,i}}} + \alpha_{2,i}\hat{\delta}_{i,k}^{\frac{n_{2,i}}{m_{2,i}}}), \end{aligned} \quad (28)$$

where $\hat{\delta}_{i,k} = \varepsilon_{2,i,k} + \beta_i\varepsilon_{1,i,k}^{\frac{q_i}{p_i}}$ is the sliding manifold related to the new system state vector $\boldsymbol{\varepsilon}$, $f_{i,k} = f_i(\boldsymbol{\varepsilon}_k)$, and $\varepsilon_{j,i,k} = \varepsilon_{j,i}(kh)$. Similar to the previous steps, by combining system dynamics described in Equation (27) and controller $\hat{\mathbf{u}}_{i,k}$, the system dynamics become

$$\begin{cases} \boldsymbol{\varepsilon}_{1,k+1} = \boldsymbol{\varepsilon}_{1,k} + h\boldsymbol{\varepsilon}_{2,k} + \mathbf{d}_1 \\ \boldsymbol{\varepsilon}_{2,k+1} = \boldsymbol{\varepsilon}_{2,k} + \beta_i\varepsilon_{1,k}^{\frac{q_i}{p_i}} - \beta_i(\boldsymbol{\varepsilon}_{1,k} + h\boldsymbol{\varepsilon}_{2,k})^{\frac{q_i}{p_i}} + \mathbf{d}_2 \\ \quad + \frac{1}{h}(-\Delta\mathbf{r}_{k+2} + \Delta 2\mathbf{r}_{k+1} - \Delta\mathbf{r}_k). \end{cases} \quad (29)$$

2.4.2. Performance optimization

To eliminate the effect of uncertainties, a new compensatory set point is designed in this section. Since the objective is still to ensure that all the subsystems can track their original set points when the system is subjected

to uncertainties or disturbances, the following performance function is used.

$$J = e_{1,N}^T e_{1,N} + \sum_{\zeta=0}^{N-1} e_{1,\zeta}^T e_{1,\zeta}, \tag{30}$$

where N is the bound of sampling times. The performance index previously chosen tries to guarantee the minimum position error between the robot and inherent set point r^* . The outer loop is designed based on this performance index, where the idea is to select Δr so that J is minimized. Based on the chosen performance index, the Hamiltonian becomes

$$\begin{aligned} H^\zeta &= e_{1,\zeta}^T e_{1,\zeta} + \lambda_{\zeta+1}^T (\varepsilon_{1,\zeta} + h\varepsilon_{2,\zeta} + d_1 \\ &\quad \varepsilon_{2,\zeta} + h(f_\zeta + g_\zeta \hat{u}_\zeta) + d_2) \\ &= \sum_{i=1}^n \{ \varepsilon_{1,i,\zeta}^2 + 2\varepsilon_{1,i,\zeta} \Delta r_{i,\zeta} + \Delta^2 r_{i,\zeta} \\ &\quad + \lambda_{1,i,\zeta+1} (\varepsilon_{1,i,\zeta} + h\varepsilon_{2,i,\zeta} + d_1) \\ &\quad + \lambda_{2,i,\zeta+1} [\varepsilon_{2,i,\zeta} + \beta_i \varepsilon_{1,i,\zeta}^{\frac{q_i}{p_i}} - \beta_i (\varepsilon_{1,i,\zeta} \\ &\quad + h\varepsilon_{2,i,\zeta})^{\frac{q_i}{p_i}} - (\alpha_{1,i} \hat{s}_{i,\zeta}^{\frac{n_{1,i}}{m_{1,i}}} + \alpha_{2,i} \hat{s}_{i,\zeta}^{\frac{n_{2,i}}{m_{2,i}}}) \\ &\quad - \frac{1}{h} (\Delta r_{i,\zeta+2} - 2\Delta r_{i,\zeta+1} + \Delta r_{i,\zeta}) + d_2] \}, \end{aligned} \tag{31}$$

where $\lambda \in \mathbb{R}^{2 \times n}$ is the costate.

Remark 1 Because of the multi-rate method used in the design, more than one sampling period of system states $((\bullet)_k)$ may arrive during the sampling period of a reselected set point $((\bullet)_\zeta)$. If the reselected set point remains the same during the period considered in the Hamilton equation $(\Delta r_{i,k+2} = \Delta r_{i,k+1} = \Delta r_{i,k})$, the reselected set point is unrelated to the system states during this period. In the Hamilton equation, $\Delta r_{i,\zeta}$ takes the place of $\Delta r_{i,k}$ to make sure the new strategy is sensitive to the change of system state.

The basic idea of the Hamilton equation is to find the minimum value by using the partial derivative related to the performance index. To get the dynamics of the cost function, the partial derivative of each state is needed.

$$\begin{aligned} \lambda_{1,i,\zeta} &= \frac{\partial H^\zeta}{\partial \varepsilon_{1,i,\zeta}} \\ &= \lambda_{2,i,\zeta+1} \left[\beta_i \frac{q_i}{p_i} \varepsilon_{1,i,\zeta}^{\frac{q_i}{p_i}-1} - \beta_i \frac{q_i}{p_i} (\varepsilon_{1,i,\zeta} \right. \\ &\quad \left. + h\varepsilon_{2,i,\zeta})^{\frac{q_i}{p_i}-1} - (\alpha_{1,i} \frac{n_{1,i}}{m_{1,i}} \hat{s}_{i,\zeta}^{\frac{n_{1,i}}{m_{1,i}}-1} \right. \\ &\quad \left. + \alpha_{2,i} \frac{n_{2,i}}{m_{2,i}} \hat{s}_{i,\zeta}^{\frac{n_{2,i}}{m_{2,i}}-1}) \beta_i \frac{q_i}{p_i} \varepsilon_{1,i,\zeta}^{\frac{q_i}{p_i}-1} \right] \\ &\quad + 2\varepsilon_{1,i,\zeta} + 2\Delta r_{i,\zeta} + \lambda_{1,i,\zeta+1}, \\ \lambda_{2,i,\zeta} &= \frac{\partial H^\zeta}{\partial \varepsilon_{2,i,\zeta}} \\ &= \lambda_{1,i,\zeta+1} h + \lambda_{2,i,\zeta+1} \left[1 - \beta_i h \frac{q_i}{p_i} (\varepsilon_{1,i,\zeta} \right. \\ &\quad \left. + h\varepsilon_{2,i,\zeta})^{\frac{q_i}{p_i}-1} - (\alpha_{1,i} \frac{n_{1,i}}{m_{1,i}} \hat{s}_{i,\zeta}^{\frac{n_{1,i}}{m_{1,i}}-1} \right. \\ &\quad \left. + \alpha_{2,i} \frac{n_{2,i}}{m_{2,i}} \hat{s}_{i,\zeta}^{\frac{n_{2,i}}{m_{2,i}}-1}) \right]. \end{aligned} \tag{32}$$

To make sure that the Hamilton equation has an extremum, Δr_i needs to satisfy the stationary condition, which is

$$0 = \frac{\partial H^\zeta}{\partial \Delta r_{i,\zeta}} = 2\varepsilon_{1,i,\zeta} + 2\Delta r_{i,\zeta} - \frac{1}{h}\lambda_{2,i,\zeta+1}. \tag{33}$$

Owing to the nonlinear controller expression used in this equation, it is necessary to check if singular points exist in the Hamilton equation is necessary. This is described in the following subsection.

2.4.3. Singular problem analysis

Because the component that relates to $\hat{s}_{i,\zeta}^{\frac{n_{2,i}}{m_{2,i}}-1}$ appears in costate dynamics and the parameter selections $\frac{n_1}{m_1} \in (0, 1)$, the singular problem has to be considered when $\varepsilon_1 = 0$ and when the system states arrive at the surface. To solve the singular problem, a new coefficient is introduced. [27]

$$\mu_{i,\zeta} = \begin{cases} \sin\left(\frac{\pi}{2} \frac{|\hat{s}_{i,\zeta}^{\frac{n_{1,i}}{m_{1,i}}}|}{\tau}\right) & \text{if } |\hat{s}_{i,\zeta}^{\frac{n_{1,i}}{m_{1,i}}}| \leq \tau, \\ 1 & \text{otherwise} \end{cases} \tag{34}$$

where τ is the enclosing scope of the sinusoidal function, and $\tau \ll 1$. When $\hat{s}_{i,\zeta} \rightarrow 0$, $\mu_{i,\zeta} \hat{s}_{i,\zeta}^{\frac{n_{1,i}}{m_{1,i}}-1} \rightarrow 1$. After introducing the new coefficient, the costates dynamics change into

$$\begin{aligned} \lambda_{1,i,\zeta} &= \frac{\partial H^\zeta}{\partial \varepsilon_{1,i,\zeta}} \\ &= \lambda_{2,i,\zeta+1} \left[\beta_i \frac{q_i}{p_i} \varepsilon_{1,i,\zeta}^{\frac{q_i}{p_i}-1} - \beta_i \frac{q_i}{p_i} (\varepsilon_{1,i,\zeta} + h\varepsilon_{2,i,\zeta})^{\frac{q_i}{p_i}-1} - (\alpha_{1,i} \frac{n_{1,i}}{m_{1,i}} \mu_{i,\zeta} \hat{s}_{i,\zeta}^{\frac{n_{1,i}}{m_{1,i}}-1} + \alpha_{2,i} \frac{n_{2,i}}{m_{2,i}} \hat{s}_{i,\zeta}^{\frac{n_{2,i}}{m_{2,i}}-1}) \beta_i \frac{q_i}{p_i} \varepsilon_{1,i,\zeta}^{\frac{q_i}{p_i}-1} \right] \\ &\quad + 2\varepsilon_{1,i,\zeta} + 2\Delta r_{i,\zeta} + \lambda_{1,i,\zeta+1}, \tag{35} \\ \lambda_{2,i,\zeta} &= \frac{\partial H^\zeta}{\partial \varepsilon_{2,i,\zeta}} \\ &= \lambda_{2,i,\zeta+1} \left[1 - \beta_i h \frac{q_i}{p_i} (\varepsilon_{1,i,\zeta} + h\varepsilon_{2,i,\zeta})^{\frac{q_i}{p_i}-1} - (\alpha_{1,i} \frac{n_{1,i}}{m_{1,i}} \mu_{i,\zeta} \hat{s}_{i,\zeta}^{\frac{n_{1,i}}{m_{1,i}}-1} + \alpha_{2,i} \frac{n_{2,i}}{m_{2,i}} \hat{s}_{i,\zeta}^{\frac{n_{2,i}}{m_{2,i}}-1}) \right] \\ &\quad + \lambda_{1,i,\zeta+1} h. \end{aligned}$$

The reselected set point is

$$\Delta r_{i,\zeta} = -\varepsilon_{1,i,\zeta} + \frac{1}{2h}\lambda_{2,i,\zeta+1}. \tag{36}$$

The above equation gives the tuning of the set point vector when the uncertainties or disturbances exist. It is also a feedback control for the outer loop of the system shown in Fig. 1. This differs from the most existing methods where the optimization was performed for the tuning of control parameters in the low-level control systems.

Remark 2 It is possible to avoid the singular problem by choosing some other method to use in the design of the arriving part of the sliding mode controller. The singular problem is considered in this paper to provide a more adaptable solution.

Theorem 1 Suppose that the system (6) with the controller shown in Equation (28) experiences uncertainties denoted by \mathbf{d}_1 and \mathbf{d}_2 . Then, under the reselected set point $\Delta \mathbf{r}_\zeta$, the performance index described in Equation (30) will be optimized, and system states will also track the original set point \mathbf{r}^* .

Proof 2 Because the set point update law was introduced, $\mathbf{r}_\zeta = \mathbf{r}_\zeta^* + \Delta \mathbf{r}_\zeta$, where $\Delta \mathbf{r}_\zeta$ is shown in Equation (36). With the uncertainties, control inputs change from \mathbf{e} to $\boldsymbol{\varepsilon}$. Because the changes are in the set point, the SMC controller can guarantee the stability of the system. If the convergence of $\boldsymbol{\varepsilon}$ to \mathbf{e} can be proved, then the system states will track \mathbf{r}^* .

By combining Equation (35) and Equation (36), the relationships between costates are

$$\begin{cases} \lambda_{1,i,\zeta+1} = \lambda_{1,i,\zeta} - A\lambda_{2,i,\zeta+1}, \\ \lambda_{2,i,\zeta+1} = \lambda_{2,i,\zeta} - \lambda_{1,i,\zeta+1}h + B\lambda_{2,i,\zeta+1}, \end{cases} \quad (37)$$

where

$$\begin{aligned} A = & \frac{1}{h} - \beta_i \frac{q_i}{p_i} \varepsilon_{1,i,\zeta}^{\frac{q_i}{p_i}-1} + \beta_i \frac{q_i}{p_i} (\varepsilon_{1,i,\zeta} + h\varepsilon_{2,i,\zeta})^{\frac{q_i}{p_i}-1} \\ & + (\alpha_{1,i} \frac{n_{1,i}}{m_{1,i}} \mu_{i,\zeta} \hat{\delta}_{i,\zeta}^{\frac{n_{1,i}}{m_{1,i}}-1} \\ & + \alpha_{2,i} \frac{n_{2,i}}{m_{2,i}} \hat{\delta}_{i,\zeta}^{\frac{n_{2,i}}{m_{2,i}}-1}) \beta_i \frac{q_i}{p_i} \varepsilon_{1,i,\zeta}^{\frac{q_i}{p_i}-1}, \end{aligned} \quad (38)$$

and

$$\begin{aligned} B = & \beta_i h \frac{q_i}{p_i} (\varepsilon_{1,i,\zeta} + h\varepsilon_{2,i,\zeta})^{\frac{q_i}{p_i}-1} \\ & + (\alpha_{1,i} \frac{n_{1,i}}{m_{1,i}} \mu_{i,\zeta} \hat{\delta}_{i,\zeta}^{\frac{n_{1,i}}{m_{1,i}}-1} + \alpha_{2,i} \frac{n_{2,i}}{m_{2,i}} \hat{\delta}_{i,\zeta}^{\frac{n_{2,i}}{m_{2,i}}-1}). \end{aligned} \quad (39)$$

If the system has equilibrium points, then system states will remain at one of the equilibrium points at the end. Assume the system described in Equation (37) has equilibrium points. Based on the properties of equilibrium points,

$$\lambda_{i,\zeta} = \lambda_{i,\zeta+1}. \quad (40)$$

Combining with Equation (37) produces

$$\begin{cases} 0 = A\lambda_{2,i,\zeta+1} \\ 0 = -\lambda_{1,i,\zeta+1}h + B\lambda_{2,i,\zeta+1}. \end{cases} \quad (41)$$

If $A\lambda_{2,i,\zeta+1} = 0$ is satisfied, either A or $\lambda_{2,i,\zeta+1}$ should be zero. When $A = 0$ and $\varepsilon_{i,\zeta} \neq 0$, the costate equation has an equilibrium point. However, when the sliding mode controller is applied to the system, $\varepsilon_{i,\zeta} \neq 0$ is not an equilibrium point for the system. According to Equation (38), a constant $\frac{1}{h}$ exists in A , which means if $A = 0$ is desired, the system states have to compensate for the constant.

Based on Equation (37), if $\lambda_{2,\zeta+1} = 0$, the system will stay at $\lambda_{i,\zeta+1} = 0$.

When $\lambda = 0$,

$$\begin{cases} 0 = \mathbf{x}_{1,i,\zeta} - \mathbf{r}_{i,\zeta}^* \\ 0 = \mathbf{x}_{2,i,\zeta} - \frac{1}{h}(\mathbf{r}_{\zeta+1}^* - \mathbf{r}_\zeta^*). \end{cases} \quad (42)$$

This equation shows that $\mathbf{x} \rightarrow \mathbf{r}^*$.

Remark 3 The structure in Figure 1 is generic in the sense that many industrial systems have exhibited such a multiple-layered structure^[28,29] in their operations, where the top-layer generates the optimized set-points and the lower-layer consists of a number of closed loop tracking systems. This indicates that the method described above can be readily applied to a wide-range of complex systems.

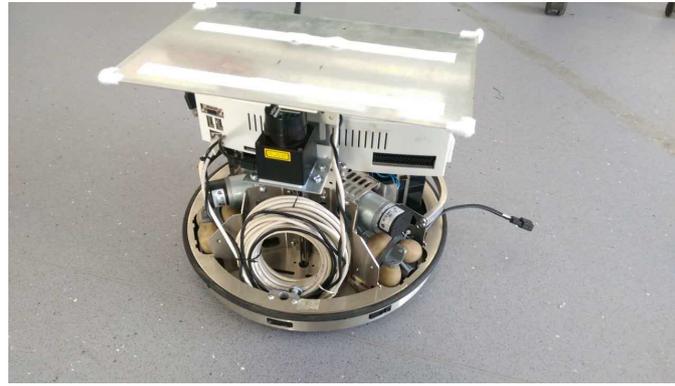


Figure 2. Robotino mobile robot.

The procedures of the algorithm design and implementation in practice is given in the following.

- **Step 1:** Design the paths based upon the destinations for the actuators. The path used in this paper is a direct line. As the hardware structure of Robotino is stated, the motion is factorized in the X and Y direction, whilst with the rotation in the Z direction;
- **Step 2:** Design the controller with the MCU in the actuators as the SMC described in this paper. This is for the actuators to guarantee the stability of the whole motion system;
- **Step 3:** Add/simulate faults in the system. Indeed, faults can be added in any loop such as the position, velocity or current loops. Faults used in this paper are sinusoidal functions added in position and velocity loops.
- **Step 4:** Record the reading from the encoders as the input for optimization;
- **Step 5:** Run the optimal algorithm with the encoder reading to obtain the re-selection of the set-points array, then add the array to the designed path. In this phase the limited computing resource of Robotino has to be considered. The optimal algorithm are therefore executed in a PC instead of the MCU in the Robot;
- **Step 6:** Repeat Step 3 to Step 5 until the system operational performance reaches its expectation.

3. RESULTS

With increasing interest in industrial automation, mobile robots have been playing an important role in transportation, rescue, and other fields. Mobile robots always work in complex situations in which uncertainties are inevitable. Most of those applications rely on accurate information about the robot's location; however, the original controller may not be able to guarantee that accuracy when uncertainties occur.

The Robotino mobile robot in Figure 2 was used in this experiment. The programming environment was Ubuntu Linux. The Robotino is a holonomic mobile robot in 2D. The wheels were driven by direct current (DC) motors. Figure 3 shows the block diagram of the inner closed-loop design of the Robotino. The trajectory planning agency generated the set points based on the requirements. The control signals were passed to the Robotino through its inherent WLAN-Link. The embedded system in the Robotino transferred the velocity setting values into torques. To guarantee the primary stability of the mobile robot, the Robotino had a PI controller inside.

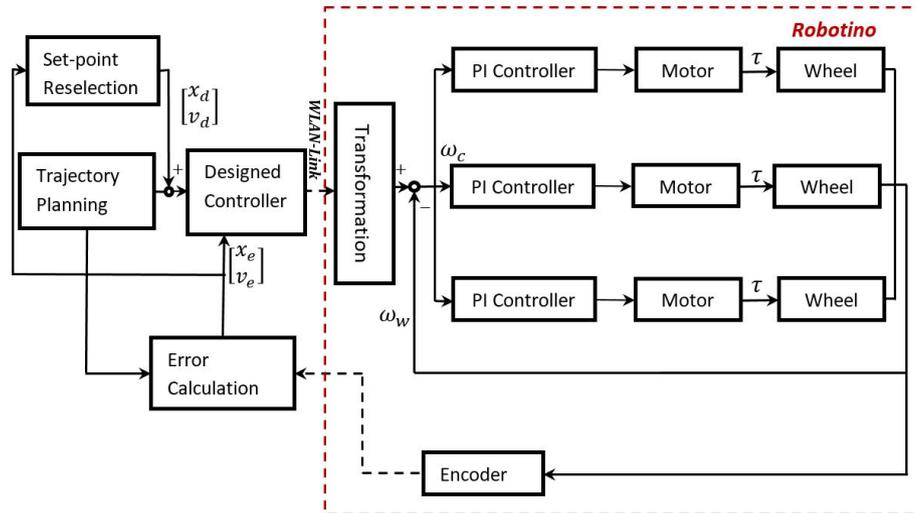


Figure 3. Block diagram of Robotino control.

To demonstrate this method, the sliding manifold picked for x and y was

$$s = v + 2p^{\frac{11}{9}}, \tag{43}$$

where v represents the velocity, and p is the position feedback from encoders. To make sure that the Robotino could be initialised at any position,

$$u_a = -(0.02s^{\frac{9}{11}} + 0.05s^{\frac{11}{9}}). \tag{44}$$

The performance index used in this study is

$$J = e_{1,N}^T e_{1,N} + \sum_{k=0}^{N-1} e_{1,k}^T e_{1,k}. \tag{45}$$

Figure 4 and Figure 5 are system state errors. Encoder faults were added onto the controller inputs. The controller's inputs that were affected by faults are represented by solid curves. The figures also show that when the system worked without the optimal loop, the robot's stability was still guaranteed. According to the hardware structure and programming logic, the optimized control signal may not have been able to follow the system if their sampling frequencies remained the same. The multirate sampling method was used in this experiment. Dotted curves represent the resampled signals. The sampling frequency of the resampled signal was one-tenth of the system's clocking frequency.

Velocities of the mobile in each direction are shown in Figures 6 and 7. The resampled velocities are represented with dotted curves.

Figure 8 and Figure 9 are the comparisons of optimized and original position errors relating to r^* and r . From the figures, the reselected set points increased the feedback error at the beginning, which offered a faster response. Δr also decreased the steady errors, and the reselected set point did not affect the original set point tracking. The mobile robot not only had a faster response but also improved accuracy. Figure 10 is the performance index of the mobile robot. The performance index with the reselected set point was better than the original one.

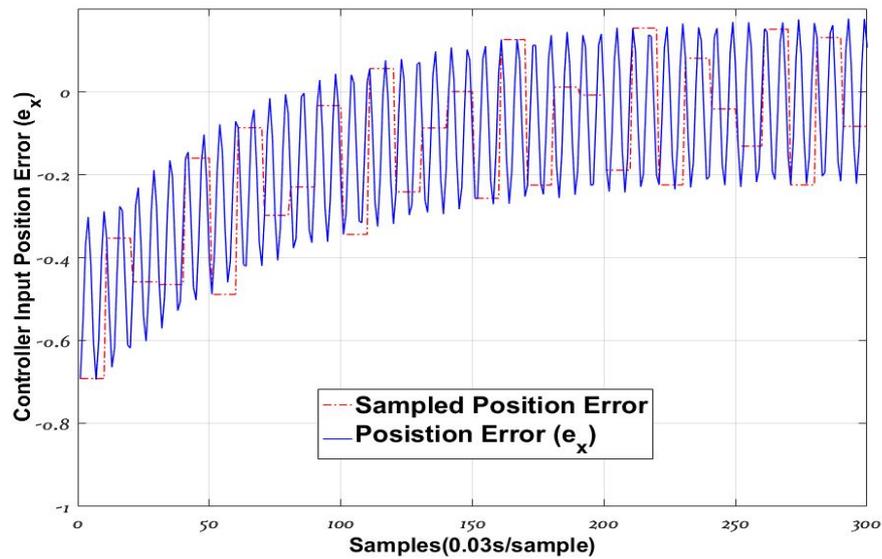


Figure 4. Sampled position error in x direction.

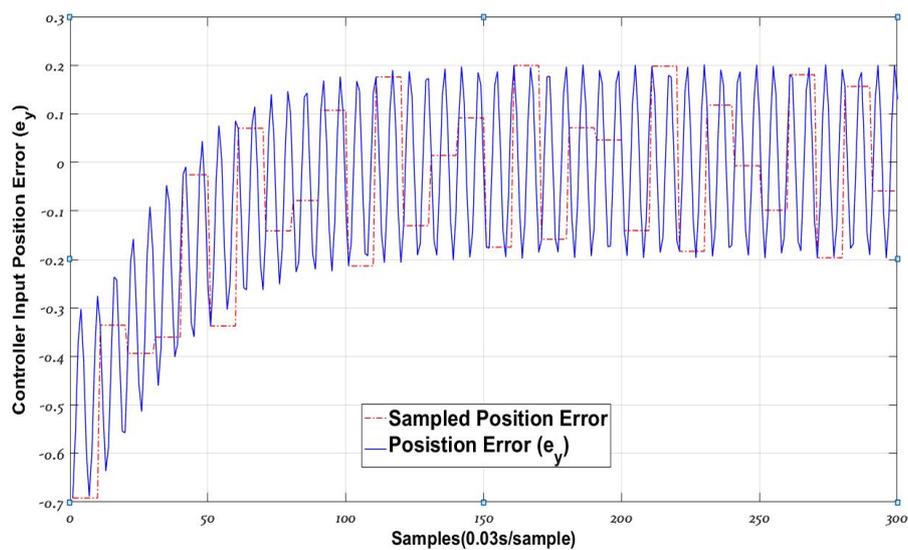


Figure 5. Sampled position error in y direction.

On the other hand, it can be seen that the possible comparison can be made. However, this would mean that we need to compare the results of the set point resection with respect to the case when the set point are not tuned at all regardless of the presence of uncertainties. As the proposed algorithm is an optimization based design. It is believed that the set point tuning would generally produce a better result in terms of minimizing the impact of the uncertainties.

4. CONCLUSIONS

In this paper, a novel set point reselection method was proposed. A sliding mode controller was applied onto the inner loop as the basic controller, which guaranteed the system stability. To eliminate the effects of uncer-

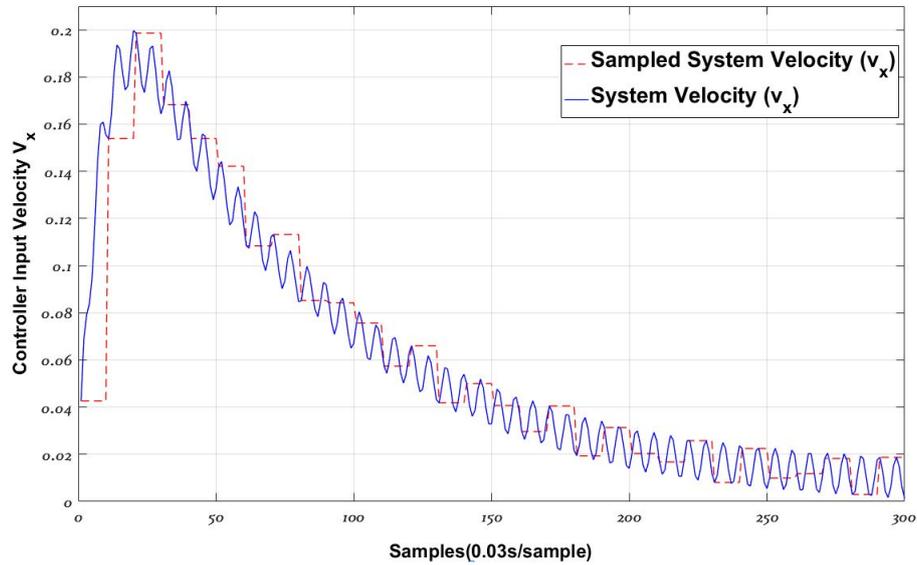


Figure 6. Sampled velocity in x direction.

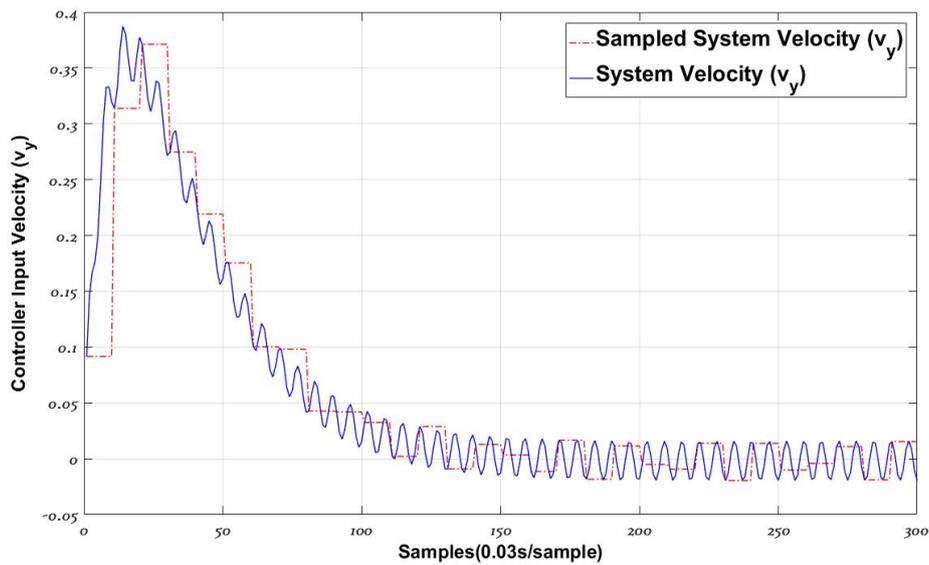


Figure 7. Sampled velocity in y direction.

tainties, the outer loop was designed based on the Hamilton equation used in the optimality principle, which focused on optimizing the chosen system performance index. However, as the nonlinear sliding manifold was introduced, the singular problem existed in the Hamilton equation. To solve the singular problem, an extra sine function was added to the Hamilton equation. The singular problem could also be avoided by carefully choosing the value of the parameters in the sliding manifold. To show a more general result, singular problem was still considered in this paper. Because the new set point may have changed the stability of this system, the convergence to e was proved. The proposed algorithm has been applied to a mobile robot and encouraging results have been obtained.

The system considered in this paper is assumed having known dynamics for f_k and g_k . However, in practice

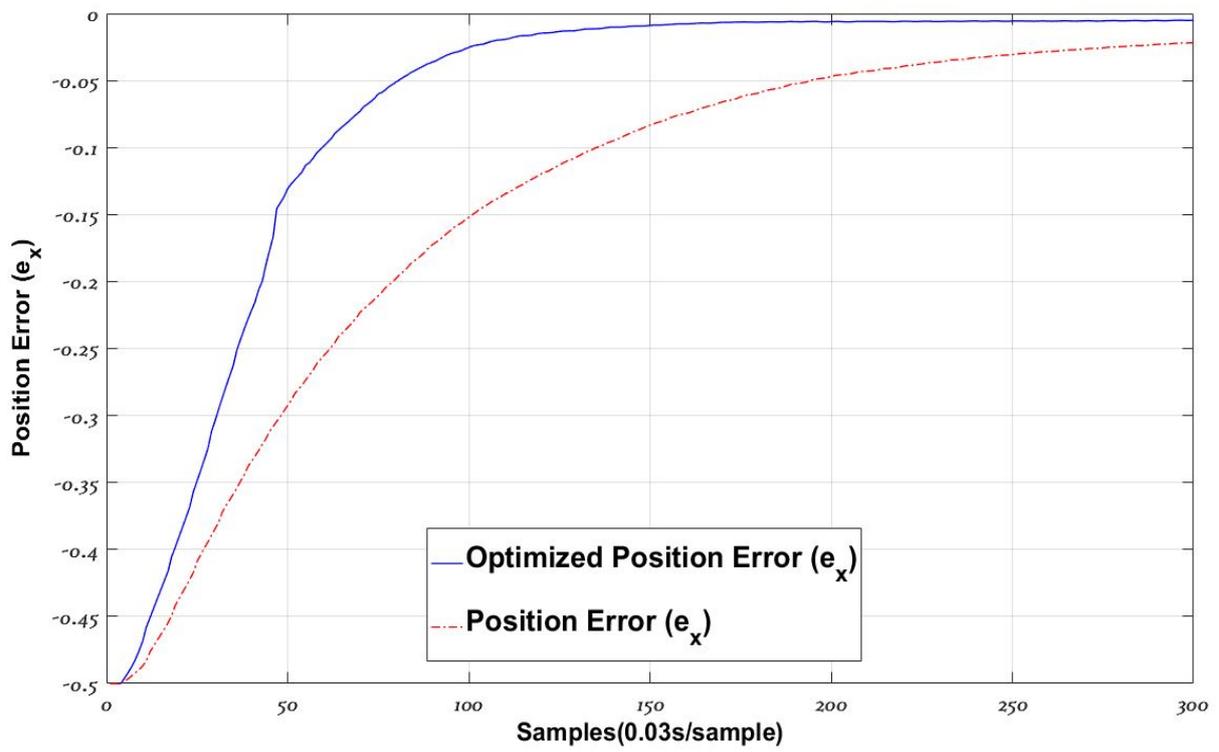


Figure 8. Optimal system output error (e_x).

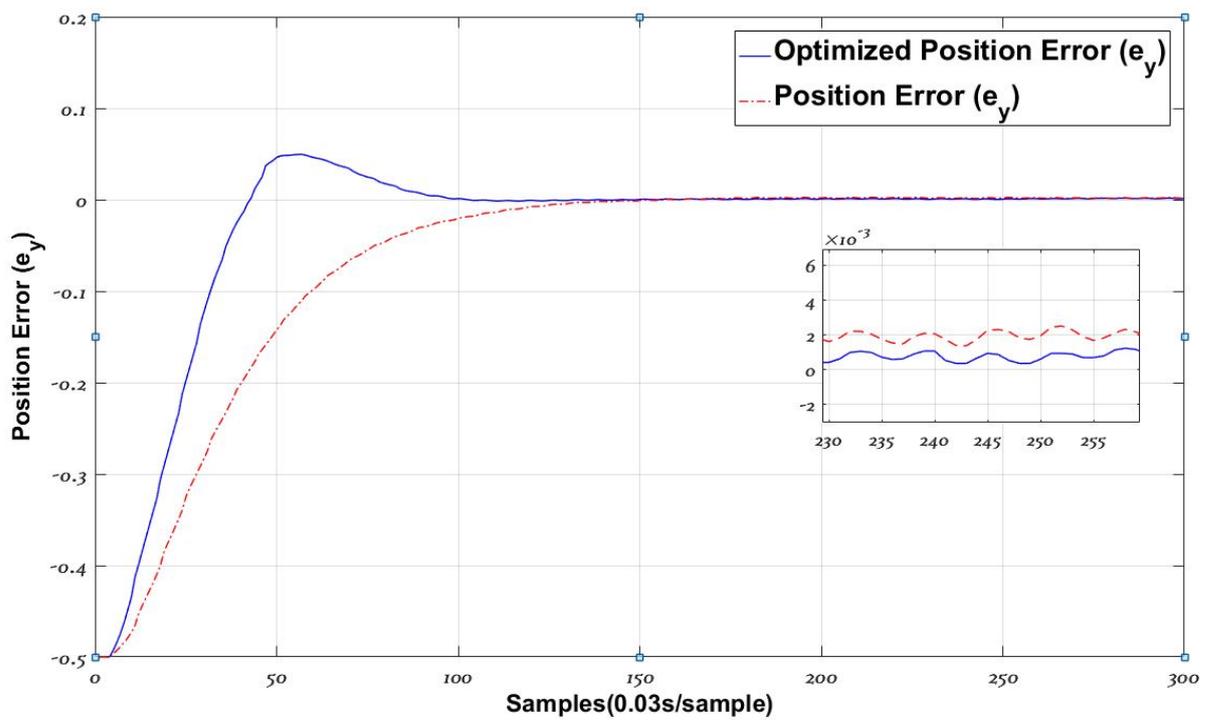


Figure 9. Optimal system output error (e_y).

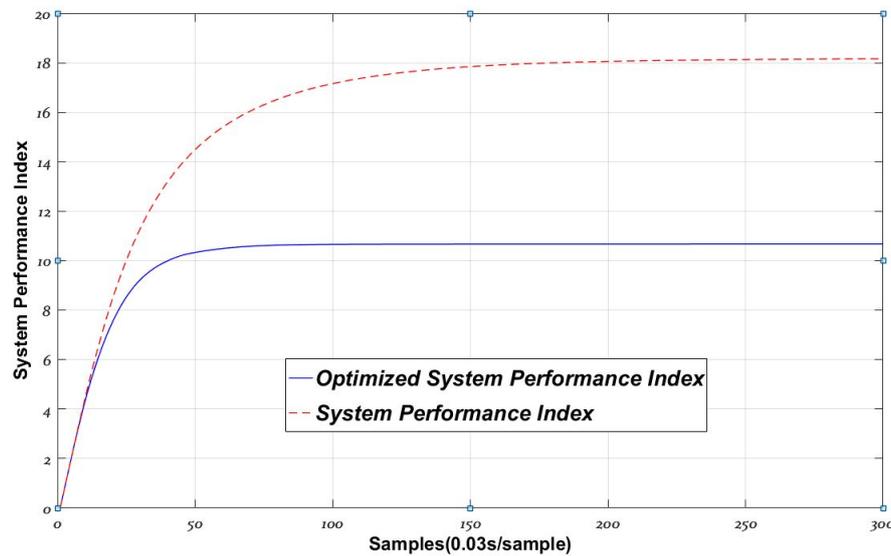


Figure 10. System performance index.

there are many systems of which their dynamics would be subjected to unknown changes in operation environment and other unpredictable factors. As a result, for the future work, we need to consider the adaptive control as well where the model parameters and system dynamics will be learnt using data-driven approaches such as neural networks to estimate dynamics f_k and g_k in equation 1), This would lead to an extra adaptive tuning loop for the closed loop system as shown in Figure 1.

DECLARATIONS

Acknowledgement

The authors would like to thank Dr. Eduard A. Codres and Dr. Mohamed Mustafa from Control Center at the University of Manchester and Dr. Liping Yin from Nanjing University of Information Science and Technology. They provided insight and expertise that greatly assisted this research. The work was performed when Dr. Xiaomo Yan was a PhD student under the supervision of Professor Hong Wang (the second author) between 2013 and 2016. Support from the University of Manchester is gratefully acknowledged.

Authors' contributions

Formulation of detailed algorithm together with testing on mobile robot: Yan X
Main idea on the operational control with set-point tuning: Wang H

Availability of data and materials

The experimental data are not available for open-source usage.

Financial support and sponsorship

The work is partly supported by the University of Manchester UK.

Conflicts of interest

All authors declared that there are no other conflicts of interest.

Ethical approval and consent to participate

Not applicable.

Consent for publication

Not applicable.

Copyright

© The Author(s) 2023.

REFERENCES

1. Chai T, Qin SJ, Wang H. Optimal operational control for complex industrial processes. *Annu Rev Control* 2014;38:81-92. DOI
2. Dai W, Chai T, Yang SX. Data-driven optimization control for safety operation of hematite grinding process. *IEEE Trans Ind Electron* 2015;62:2930-41. DOI
3. Wang A, Zhou P, Wang H. Performance analysis for operational optimal control for complex industrial processes under small loop control errors. In: Proceedings of the 2014 international conference on advanced mechatronic systems. IEEE, 2014:159-64. DOI
4. Wu M, Xu C, She J, Yokoyama R. Intelligent integrated optimization and control system for lead–zinc sintering process. *Control Eng Pract* 2009;17:280-90. DOI
5. Zhou P, Chai T, Sun J. Intelligence-based supervisory control for optimal operation of a DCS-controlled grinding system. *IEEE Trans Contr Syst Technol* 2013;21:162-75. DOI
6. Dai W, Zhou P, Zhao D, Lu S, Chai T. Hardware-in-the-loop simulation platform for supervisory control of mineral grinding process. *Powder Tech* 2016;288:422-34. DOI
7. Chai T, Ding J, Wu F. Hybrid intelligent control for optimal operation of shaft furnace roasting process. *Control Eng Pract* 2011;19:264-75. DOI
8. Haq IU, Khan Q, Ullah S, et al. Neural network-based adaptive global sliding mode MPPT controller design for stand-alone photovoltaic systems. *PLoS One* 2022;17:e0260480. DOI
9. Ilyas M, Iqbal J, Ahmad S, Uppal AA, Imtiaz WA, Riaz RA. Hypnosis regulation in propofol anaesthesia employing super-twisting sliding mode control to compensate variability dynamics. *IET Syst Biol* 2020;14:59-67. DOI
10. Anjum MB, Khan Q, Ullah S, et al. Maximum power extraction from a standalone photo voltaic system via neuro-adaptive arbitrary order sliding mode control strategy with high gain differentiation. *Appl Sci* 2022;12:2773. DOI
11. Iqbal J. Modern control laws for an articulated robotic arm: modeling and simulation. *Eng Technol Appl Sci Res* 2019;9:4057-61. DOI
12. Yan X, Zuo Z, Yin L, Wang A, Wang H. Chattering-free sliding mode control for mimo nonlinear manipulator systems based on adaptive neural networks. In: 2015 54th IEEE Conference on Decision and Control (CDC). Dec 2015, pp. 6300–5. DOI
13. Li S, Zhou M, Yu X. Design and implementation of terminal sliding mode control method for PMSM speed regulation system. *IEEE Trans Ind Inf* 2013;9:1879-91. DOI
14. Zhang J, Lin Y, Feng G. Analysis and synthesis of memory-based fuzzy sliding mode controllers. *IEEE Trans Cybern* 2015;45:2880-9. DOI
15. Naouar MW, Monmasson E, Naassani AA, Slama-belkhdja I. FPGA-based dynamic reconfiguration of sliding mode current controllers for synchronous machines. *IEEE Trans Ind Inf* 2013;9:1262-71. DOI
16. Zhang X, Sun L, Zhao K, Sun L. Nonlinear speed control for PMSM system using sliding-mode control and disturbance compensation techniques. *IEEE Trans Power Electron* 2013;28:1358-65. DOI
17. Lee JD, Khoo S, Wang Z. DSP-based sliding-mode control for electromagnetic-levitation precise-position system. *IEEE Trans Ind Inf* 2013;9:817-27. DOI
18. Basin MV, Rodriguez-ramirez PC. Sliding mode controller design for stochastic polynomial systems with unmeasured states. *IEEE Trans Ind Electron* 2014;61:387-96. DOI
19. Khalil HK. *Nonlinear Systems*. New Jersey: Prentice-Hall, 1996.
20. Polyakov A. Nonlinear feedback design for fixed-time stabilization of linear control systems. *IEEE Trans Automat Contr* 2012;57:2106-10. DOI
21. Feng Y, Yu X, Han F. High-order terminal sliding-mode observer for parameter estimation of a permanent-magnet synchronous motor. *IEEE Trans Ind Electron* 2013;60:4272-80. DOI
22. Janardhanan S, Bandyopadhyay B. On discretization of continuous-time terminal sliding mode. *IEEE Trans Automat Contr* 2006;51:1532-6. DOI
23. Lin C. Nonsingular terminal sliding mode control of robot manipulators using fuzzy wavelet networks. *IEEE Trans Fuzzy Syst* 2006;14:849-59. DOI
24. Zou AM, Kumar KD, Hou ZG, Liu X. Finite-time attitude tracking control for spacecraft using terminal sliding mode and chebyshev Neural Network. *IEEE Trans Syst Man Cybern B Cybern* 2011;41:950-63. DOI
25. Man Zhihong, Paplinski A, Wu H. A robust MIMO terminal sliding mode control scheme for rigid robotic manipulators. *IEEE Trans Automat Contr* 1994;39:2464-9. DOI
26. Wang H, Wang YJ, Kabore P. Time-varying controller for known nonlinear dynamic systems with guaranteed stability. *Int J Syst Sci* 2002;33:931-8. DOI
27. Zuo Z. Nonsingular fixed-time consensus tracking for second-order multi-agent networks. *Automatica* 2015;54:305-9. DOI
28. Wang A, Wang H, Sheng N, Yin X. Performance analysis for operational optimal control for complex industrial processes — the square

- impact principle. In: 2015 21st International Conference on Automation and Computing (ICAC), Glasgow, UK, 2015, pp. 1-6. [DOI](#)
29. Yin L, Wang H, Yan X, Zhang H. Disturbance observer-based dynamic optimal setting control. *IET Contr Theo Appl* 2018;12:2423-32. [DOI](#)