

First of all, the authors greatly appreciate and value all comments and suggestions given by the referee. All comments and inquires are addressed and replied to the best of the authors' knowledge. Before presenting our replies to specific comments, the authors would like to point out some points at first.

1. The whole paper is re-organized so that the description is more logical. Hence, Some sections may be placed different from the previous version.
2. The modified contents are highlighted with red color, so that it is easier for readers to figure out the modified contents.
3. In order to distinguish from the output $\tilde{x}$, the state vector $\tilde{\mathbf{X}}$ is in bold font, whereas it was not bold in the previous version.
4. Some contents are re-written to strengthen the novelty, clarify ideas, or correct errors in the previous edition.
5. The paper has been considerably edited, and the numbering of equations, figures, and tables etc. is changed. As a result, all numbering in the replies refer to the **NEW** numbers in the revised paper, where as the numbering provided by reviewers keeps unchanged (in the previous version).

Reviewer #1

Comment #1

- **Comment:** The paper is fairly well written although containing numerous mistakes or unclarities. The contribution seems limited as it evolves around nonlinear control of an inverted pendulum cart with adding a second sliding surface in the state transformation (s_2) also including integral action of the output error. The topic of integral sliding mode control is far from new and the innovation in this paper seems cumbersome. All the terms in s_1 and s_2 are common except from the integral part, such that you end up, sort of, increasing the gain on each derivative of $\tilde{x}$ as a sum of λ and α parts plus $\alpha^2 \int e d\tau$. Wouldn't it be easier to realize it as $s = s_1 + \alpha \int e d\tau$ which is normally done when designing integral sliding mode controllers? That makes tuning seemingly much easier. I have the following comments/suggestions (numbers at start of each point is referring to line in original manuscript):
- **Reply:** Thank you for the suggestion about the design of the sliding surface. $s = s_1 + \alpha \int e d\tau$ could be a sliding surface worthy of trying. However, our design is fundamentally different from the commonly known integral sliding mode control (ISMC), whose sliding surface has the suggested form. As shown in the example, actually, $s = 2\dot{e} + (\lambda + 2\alpha)e + \alpha^2 \int e d\tau$ is a special form only for a dynamic system described by a 2nd ordered ODE. Our algorithm can extend the control regime to a system described by an n -th ordered ODE. The tuning of parameters will be illustrated and discussed in the reply to Comment #2.

Regarding the novelty or contribution, the following words are added to the paper to illustrate our novelty and contributions: The work presented in this paper extends the work presented in [19] to a wider range. The following novelties or contributions are made in this paper. *First*, different from the commonly known ISMC, PISMC, or PID sliding surface, which usually has the sliding surface of $s = k_p e + k_I \int e d\tau + k_d \dot{e} + k_1 e(0) + k_2 \dot{e}(0)$ and five parameters or more to tune, our algorithm in [19] extends the sliding surface to the n th order derivative of the output error with only two parameters to tune. In [19] however, a system of unweighted control was considered. Based on the work in [19], this paper proposes the sufficient condition to design a control for a system of weighted control. *Secondly*, our algorithm was realized and verified in a two wheel vehicle system. In the literature, researchers either applied their work to nonlinear systems using numerically simulated results, or realized their algorithms in a linear system. This paper not only verifies our capability of controlling a nonlinear system by numerical simulations, but also realizes the algorithms in a TWVS using its original nonlinear model. *Thirdly*, tuning of control parameters is always critical in designing a control. In our proposed algorithms, there are only four parameters to tune for all systems. From analytical analysis and past experiences we roughly understand the effects of these parameters, though they were not rigorously proved. A systematic tuning procedures are also presented in this paper.

Comment #2

- **Comment:** Normally introducing extra parameters makes tuning more complex and is thus not necessarily a pro but rather a con, if it isn't strictly necessary.
- **Reply:** There are only four parameters to tune in total for an arbitrarily high order system: λ and α to construct s -surface, η to construct v_{sw} , and Φ to determine the saturation function. Usually, a higher order controller has more parameters to tune. This characteristics may sacrifice some degrees of freedom in tuning parameters and the performance they may bring, but it simplifies the tuning procedures.

On the other hand, the proposed algorithm has more degrees of freedom to adjust the system performance, compared to the conventional SMC algorithm. Although this characteristics may result in complexity in selecting parameters, with the methods and analysis detailed in the paper, one is able to tune parameters systematically and efficiently.

Another concern is the cost of the control. Better performance may be resulted from larger effort, such as larger control signal or larger energy. From our experiences, though not rigorously proved, in some cases the proposed algorithm actually cost less than the conventional SMC, which only tunes λ to build the sliding surface and Φ to defined the boundary layer.

An example is presented in Fig. 1. The example is simulated with using the aforementioned nonlinear system. In order to evaluate the control laws, a cost function J is defined as follows:

$$J = \frac{1}{2} \int_0^t \tilde{u}^2 d\tau \quad (1)$$

Two control laws are employed to control the system: the SMC and our proposed PISMC. Parameters are tuned so that the two systems have similar settling time. A smaller overshoot occurs in the system using PISMC. The simulations in Fig. 1 show that the proposed algorithm costs less while keeping similar transient performance. Notable, under the designated parameters chattering occurs once using the control from SMC. As a result, it can be concluded that our algorithm performs better than the conventional SMC in this specific case.

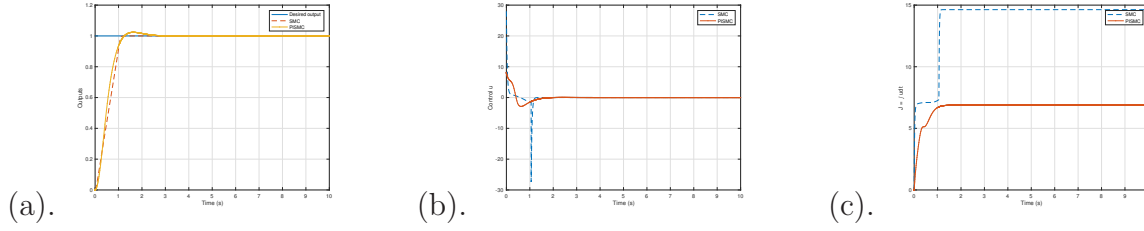


Figure 1: Comparison of parameter effectiveness. (a). Outputs $\tilde{x}$; (b). control input $\tilde{u}$; (c). cost of controls J .

Comment #3

- **Comment:** The dimensions of all defined states are missing. Is $\tilde{x} \in R$ or R^n ? What about $\tilde{x}_d$? if $\tilde{x} \in R$ and as $(d/dt + \lambda)^{(n-1)} \in R$ then according to Eq. (3) $s(\tilde{x}, t) \in R$ but in 93 it is stated that $s(\tilde{x}, t) \in R^n$.
- **Reply:** Thank you for the remind. The following phrases are added and colored to highlight in the text to clarify the idea: “ $\dots$, where both of $\tilde{x}_d(t)$ and $\tilde{x}(t)$ are in $\mathbb{R}$.”, and “where $e(t) \in \mathbb{R}$.” below Eq. (3)

Comment #4

- **Comment:** It is stated that $\tilde{x}$ is output, which normally means the measured state. On the other hand in Eq. (12) the knowledge of all output derivatives up to n is needed (middle term on RHS) which is really full state plus derivative of last state. This is also evident from the example given below 157, and is thus far from output only?
- **Reply:** In order to generate the control, knowledge of full states is required, as mentioned in the paragraph below Eq. (8). The states are assumed attainable via a state observer, which sometimes requires measurement of outputs or errors. However, our statement may lead to misunderstanding. Hence, it is re-phrased as: To design the control system, the system states are assumed attainable via a state observer, which sometimes requires measurement of an output or output error as defined in (3).

Comment #5

- **Comment:** The notation in (12) seems cumbersome. Why split the α related terms? Couldn't it be expressed as $\sum_{m=0}^{m=n-1} C_m^{n-1} \alpha^{(n-m-1)} e^m (\alpha + e)$ or similar (if my derivation was correct)?
- **Reply:** We are really sorry about the confusion with the notation. The notation is directly introduced from our previous work in [19]. The superscript in e denotes the order of derivative, whereas the superscript in λ or α denotes the power. In order to avoid the ambiguity, we change the notation in the text, and add a paragraph to detail the definition. As a result, the two terms you mentioned cannot be combined together since they belong to different derivative orders.

Comment #6

- **Comment:** (11) it seems here that k is constant but in 135.5 and example 157.5 it is state dependent (varying with the whole state vector $\tilde{X}$ plus extra derivative).
- **Reply:** k is a parameter associated with the bound of dynamics variation. It could be, but is not necessarily to be, a constant. Actually, as defined in Theorem 1, $k = \tilde{F}(\tilde{X} + \eta)$. In the example, $k = 0.5\dot{\tilde{x}}^2 |\cos 3\tilde{x}| + \eta$ is a function of states.

Comment #7

- **Comment:** 5 sgn function is not defined at $s = 0$.
- **Reply:** Thank you for the correction. The definition of the sgn function has been changed as

$$\text{sgn}(s) = \begin{cases} 1 & \text{if } s > 0 \\ 0 & \text{if } s = 0 \\ -1 & \text{if } s < 0 \end{cases}.$$

Comment #8

- **Comment:** 170 There is a significant difference in control action also explaining why PISM is faster. I assume that is due to the increase in gains due to definition of $s = s_1 + s_2$ as discussed earlier.
- **Reply:** Yes. The example is a 2nd ordered dynamic system. As a result, the coefficient α^2 exists in s_2 and this may cause a considerably large control if $\alpha_2 > 1$ as a trade-off for fast response.

Comment #9

- **Comment:** (19)-(20) Some definitions are missing ($T, F, H, f_d, L/R$). Also, from Figure 6 there seems to be something odd with signs. If $F_{l,r}$ is negative it would mean they cause deceleration in $\ddot{x}_{L,R}$ but in Fig. 6 $x_{L,R}$ and $F_{L,R}$ are pointing in same direction.

- **Reply:** Thank you for the correction. The whole section has been re-write, all notations has been defined, and errors has been corrected. Please refer to the new version of the paper for the corrections.

Comment #10

- **Comment:** It is not explicitly mentioned, but is it assumed that the ground is flat, i.e. no roll motion?
- **Reply:** This algorithm can be applied in any environment and not affected by terrain. When doing experiments, we didn't intentionally select the terrain. However, a significantly varied terrain is not suggested since the cart is not laterally controlled. It might roll over during the experiments.

Comment #11

- **Comment:** F_p is missing from Fig. 6. What is difference between F_p and f_p ? Also, it seems that positive Θ is rotating in opposite direction as is normal for right handed system. This will impact sign of eq. (21) and possibly others. Also, the force F_Θ which seems to point along negative z direction seems odd, is there any transversal force present?
- **Reply:** Thank you for the correction. The whole section has been re-write, all notations has been defined, and errors has been corrected. Please refer to the new version of the paper for the corrections.

Comment #12

- **Comment:** 221 It is stated that Eq. (33) is derived from Eqs. (19), (20) and (22) although none of them contains any trigonometric functions. Thus the derivation seems not straight forward.
- **Reply:** Thank you for the correction. The whole section has been re-write, all notations has been defined, and errors has been corrected. Please refer to the new version of the paper for the corrections.

Comment #13

- **Comment:** 8. Normally a fair comparison is to tune the controller such that they have similar performance and then look at the energy spent or tune them such that they are spending the same energy and then compare transient and steady state performance. As now as both are different, it is difficult to evaluate which is actually performing better.
- **Reply:** Please refer to the reply to Comment #2. The comparison is presented in the reply.

Comment #14

- **Comment:** 12 an oscillation starts at about 12 seconds on the SMC test while the PISMC plot is cut just at that point. Is this a coincidence or is there something going on also in the PISMC? Can you comment on the cause of this oscillation?
- **Reply:** Thank you for the comment. PISMC worked fine after 12 seconds, but we made a mistake when presenting the data. An updated figure has been provided in the revised paper. The whole system were shot down at 18 seconds, because the SMC started to oscillate at 12 second and the control could not stabilize it. When we plotted the experiment results, the stupid mistake was made due to the idea of “12 second”. On the other hand, a SMC system might need longer settling time under the tuned parameter, as shown in the simulations. If the perturbation is faster than the control, the system diverges.

Reviewer #2

Comment #1

- **Comment:** The novelty is not well stressed within the current research context.
- **Reply:** Thank you for the comment. The following words are added to the paper to illustrate our novelty and contributions: The work presented in this paper extends the work presented in [19] to a wider range. The following novelties or contributions are made in this paper. *First*, different from the commonly known ISMC, PISMC, or PID sliding surface, which usually has the sliding surface of $s = k_p e + k_I \int e d\tau + k_d \dot{e} + k_1 e(0) + k_2 \dot{e}(0)$ and five parameters or more to tune, our algorithm in [19] extends the sliding surface to the n th order derivative of the output error with only two parameters to tune. In [19], however, a system of unweighted control was considered. Based on the work in [19], this paper proposes the sufficient condition to design a control for a system of weighted control. *Secondly*, our algorithm was realized and verified in a two wheel vehicle system. In the literature, researchers either applied their work to nonlinear systems using numerically simulated results, or realized their algorithms in a linear system. This paper not only verifies our capability of controlling a nonlinear system by numerical simulations, but also realizes the algorithms in a TWVS using its original nonlinear model. *Thirdly*, tuning of control parameters is always critical in designing a control. In our proposed algorithms, there are only four parameters to tune for all systems. From analytical analysis and past experiences we roughly understand the effects of these parameters, though they were not rigorously proved. A systematic tuning procedures are also presented in this paper.

Comment #2

- **Comment:** In the conventional SMC control systems, insensitivity to (matched) perturbations is guaranteed once the system enters the sliding phase. ISMC has been proposed to extend this property over the reaching phase, at the expense of enlarged space (order reduction does not occur). Consequences of this fact for the investigated two-wheel vehicle system are not discussed.
- **Reply:** Unlike the design concept of conventional SMC, the insensitivity is included in the dynamics model and also guaranteed in our algorithm. In the conventional SMC design, a precise dynamics model is employed and the perturbation is viewed as an extra input. An example is given in Eq. (22) at page 106 in *Applications of Sliding Mode Control in Science and Engineering* [Vaidyanathan and Lien (Eds), 2017]. The change rate of s is formulated by

$$\dot{s} = \sigma + M\Delta x + O(\Delta x)^2, \quad (2)$$

and the robustness is discussed based on this formulation. In our approach, however, the perturbations are already included in the parameter k for v_{sw} . Recall that $k = \tilde{F} + \eta$, where $\tilde{F}$ is the bound of the variation model, satisfying $\tilde{f} = \hat{f} + \Delta f$ and

$$|\Delta f| \leq \tilde{F} \quad (3)$$

Let $\tilde{\mathbf{X}} = \tilde{\mathbf{X}}_n + \Delta\mathbf{X}$, where $\tilde{\mathbf{X}}_n$ is the nominal states. Then,

$$\begin{aligned}\tilde{f}(\tilde{\mathbf{X}}) &= \tilde{f}(\tilde{\mathbf{X}}_n + \Delta\mathbf{X}) \\ &= \tilde{f}(\tilde{\mathbf{X}}_n) + \left. \frac{\partial \tilde{f}}{\partial \tilde{\mathbf{X}}} \right|_{\tilde{\mathbf{X}}_n} \Delta\mathbf{X} + \text{H.O.T} \\ &= \hat{f}(\tilde{\mathbf{X}}) + \Delta f\end{aligned}\tag{4}$$

where $\hat{f}(\tilde{\mathbf{X}}) = \tilde{f}(\tilde{\mathbf{X}}_n)$ and Δf includes all the remainder terms. By properly investigating and choosing the bound of Δf . The influence of perturbations is included and considered in the control. The advantage of our approach is that the robustness of the control is guaranteed with a proper choice of $\tilde{F}$, and we don't have to worry about specific perturbations. The drawback, of course, is that the control might be inefficient if $\tilde{F}$ is defined too large for the purpose of robustness. Since this is the characteristics of the algorithm itself, it was not discussed in the TWVS experiments specifically.

Thank you very much for the suggestion and remind. The highlighted sentences are included in the paper to clarify our materials.

Comment #3

- **Comment:** Does the matching condition need not be imposed with respect to disturbance (6) so that the properties stated in the theorems hold?
- **Reply:** Since the control is built based on the bound of the dynamics variations, instead of specific perturbations or disturbances, the theorem is hold without imposing disturbances. Simulations in the section of experiment also verify this statement.

Comment #4

- **Comment:** More flexibility in the controller tuning in the proposed method means more effort to adjust the system performance. No guidelines have been provided. Be specific.
- **Reply:** Thank you very much for the advice. We have re-written the paper to include the tuning process. From the analysis provided in the revised paper and our experience, the tuning procedures applying our algorithm are concluded as follows:
 1. Select Φ . $0.1 \leq \Phi \leq 1$ is suggested to speed up the convergence.
 2. Select η . Large η is helpful to speed up the convergence. However, the value of η is proportional to $\tilde{u}$. Too large value of η may cause too large control signal.
 3. Select λ . From our experience, $1 \leq \lambda \leq 10$ is a better choice.
 4. Select ρ and $\alpha = \rho\lambda$. If oscillatory response is allowed, ρ can be selected larger than 1. If not, smaller ρ is suggested.
 5. Estimate the transient properties:

- (a) The overshoot can be estimated from damping ratio $\zeta = (1 + 2\rho)/\sqrt{2\rho}$.
 - (b) Estimate the total time constant. Since the convergence procedure is $s \rightarrow 0$ and then $e \rightarrow 0$, the total time constant of convergence, τ_e can be approximated by $\tau_e = \tau_\eta + \tau_\alpha$.
 - (c) The settling time t_s , according to control theory, approximated by $t_s = 4\tau_e$.
6. Build the nonlinear mathematical model and run numerical simulations with initially selected parameters. Tune the parameters according to the transient performance and the effect of parameters.
 7. Remember to keep α as small as possible, because large α usually indicates a very large control input.

Comment #5

- **Comment:** k in (13) requires measurement of perturbation, which is difficult to obtain in the physical systems. Feasibility issues are not discussed.
- **Reply:** Actually, measurement of perturbation is not required to compute k . $k = \tilde{F}(\tilde{X} + \eta)$, where the function $\tilde{F}$ is a boundary of the potential variation of system dynamics. As a result, by studying the boundary of the system variation, we are able to define $\tilde{F}$ and its associating k .

Comment #6

- **Comment:** In order to give a proper perspective on the benefits of the proposed method a comparison with [15] should be provided in the numerical/experimental section. In the analytical section, the conceptual difference between PISM and PID sliding surface design could be illustrated by a figure?
- **Reply:** Thank you for the suggestion. The comparison of control laws has been added into the paper. The comparison between our work and that in Ref. [15] is not provided because the employed algorithm, controller design, and dynamics model to control are all different. Our work was indeed inspired by Ref. [15]. However, there are fundamental difference. In the following list the differences:
 1. Design of sliding surface for the controlled system: Our paper presents $s = 2\dot{e} + (\lambda + 2\alpha)e + \alpha^2 \int e d\tau$. In Ref. [15], the sliding surface is designed as $s = k_p e + k_i \int e d\tau$.
 2. The system to control in this paper is nonlinear, whereas the system in Ref. [15] is a linearized system.
 3. Because the system to control in Ref. [15] is linear, their proposed controller is merely a function of the system, given by

$$u = Kx - (CB)^{-1}Cf - (CB)^{-1}\eta^2 \text{sgn}(s), \quad (5)$$

where f is the original form represent a nonlinear function describing the deviation from linearity in term of disturbances and un-modeled dynamics. Our controller, however, has two fundamental differences from theirs: *First*, we don't employ the remainder nonlinearity terms directly. Instead, we employ the bound of the remainder nonlinearity terms. *Secondly*, the saturation function in our controller is more than $\text{sgn}(s)$ function, much less the selection of parameters are different.

4. The tuning procedures of parameters are discussed in this paper, but not in [?].

As a result, in our opinions comparison of two completely different systems is unfair.

Reviewer #3

Comment #1

- **Comment:** The paper “PI Sliding Mode Control with Application to Balance Control of a Two-Wheel Vehicle System” presents a control method for an inverted pendulum type system, which the authors call a Two-Wheel Vehicle System (TWVS). Inverted pendulum systems are often used as testbeds for various control algorithms. In this case, the authors propose a PI Sliding Mode Control (PISMC) algorithm and a self built physical setup to test the algorithm. Both of these components are discussed in the paper. In the Introduction the authors present the state of the art with respect to sliding mode control (SMC) and PISMC. Except [19] (2011), all the cited references are more than 10 years old. This indicates the low interest these algorithms have received from the scientific community over the past years. With respect to the novelty of the approach, this paper builds upon a paper from 2006 ([15]). The authors claim there are major differences between this paper and the one previously mentioned. The new contributions compared to [15] should be clearly stated and better emphasized throughout the paper. They are only briefly mentioned in the Introduction, then, as the model of the controller and the plant develop, no other references are made to paper [15].

- **Reply:** Thank you for the comments. Our work was indeed inspired by Ref. [15]. However, there are fundamental difference. In the following list the differences:

1. Design of sliding surface for the controlled system: Our paper presents $s = 2\dot{e} + (\lambda + 2\alpha)e + \alpha^2 \int e d\tau$. In Ref. [15], the sliding surface is designed as $s = k_p e + k_i \int e d\tau$.
2. The system to control in this paper is nonlinear, whereas the system in Ref. [15] is a linearized system.
3. Because the system to control in Ref. [15] is linear, their proposed controller is merely a function of the system, given by

$$u = Kx - (CB)^{-1}Cf - (CB)^{-1}\eta^2 \text{sgn}(s), \quad (6)$$

where f is the original form represent a nonlinear function describing the deviation from linearity in term of disturbances and un-modeled dynamics. Our controller, however, has two fundamental differences from theirs: *First*, we don't employ the remainder nonlinearity terms directly. Instead, we employ the bound of the remainder nonlinearity terms. *Secondly*, the saturation function in our controller is more than $\text{sgn}(s)$ function, much less the selection of parameters are different.

4. The tuning procedures of parameters are discussed in this paper, but not in [?].

The above words are added in the introduction to clarify the difference. More recent references are reviewed and provided to strengthen our contributions.

Comment #2

- **Comment:** In Section 2, the authors introduce the theoretical aspects related to the proposed control methods. They start with conventional SMC, then move on to PISMC. This section

is heavily theoretical and ends with a simple example intended to show the performance of the PISMC. The paper continues with the presentation of the physical setup. Some details are given regarding the hardware. However, these are not well documented and the authors simply rely on some photos. Instead of images of the boards, a hardware architecture of the system would have been more useful.

- **Reply:** Thank you for the suggestion. The whole paper is re-organized so that the description is more logical. Two new block diagrams are presented to illustrate the hardware architecture of the system.

Comment #3

- **Comment:** By reading the content of Section 3.1 it is unclear how many PIC18F258 microcontrollers (2 or 3) the board uses and what functions do they implement in the system (apart from reading the sensors). This section, relating to the hardware, is confusing and incomplete. It remains unclear where the PISMC controller is implemented and how it was implemented. Some software implementation details would complete this section. As a side argument, according to Microchip, PIC18F258 controller "is not recommended for new designs".
- **Reply:** The motor driving board contains two PIC18F258 microcontrollers. One microcontroller computed the control signal based on the PISMC algorithm, whereas the other handled the readings of sensors. The computation capability of a PIC18F258 microcontroller is actually powerful enough to implement the proposed algorithm. No new designs is required. Thank you for the suggestion. These words has been added into the paper to make things clear.

Comment #4

- **Comment:** An extensive modeling of the plant is shown which is later used for simulation purposes. The simulation details need improvement. While the modeling part is quite thorough, there is a gap between the resulted models, the available simulation details and the experimental setup.
- **Reply:** Thank you for the suggestion. The whole paper is re-organized, including the presentation of experiments and experiment setup, so that the description is more logical.

Comment #5

- **Comment:** For instance, how are the dynamics equations of the plant and the equations of the controller comprised in the block schematic of Fig. 7. Also, different notations are used across the paper which are not consistent with the simulations model. It was not specified which simulation environment was used and how the developed models were implemented.
- **Reply:** We have re-formulate the dynamics of the TWVS and re-organized the presentation of the paper. The definition of notations should be consistent in the revised version. The

only notation that looks similar is x . In our paper $\tilde{x}$ denotes the output state in general sense, whereas x denotes the position of the TWVS. Hence, $\tilde{x}$ can be x or θ regarding the experiment. However, we only investigated the control implementation of θ in the experiment in this paper.

TBA

Comment #6

- **Comment:** The tuning process of the controller's parameters for the given experimental plant/simulation model is missing. The tuning parameters of the PISMIC are advertised as the paper's main contribution. They need more attention. At the end of the paper simulation and experimental results are shown and discussed. In the case of the experimental tests in the presence of disturbances it is not specified how were the disturbances introduced to the system, or what were these disturbances like.
- **Reply:**
 1. Thank you very much for the advice. We have re-written the paper to include the tuning process, and treat it as one of the main contributions. From the analysis and our experience, the tuning procedures applying our algorithm are concluded as follows:
 - (a) Select Φ . $0.1 \leq \Phi \leq 1$ is suggested to speed up the convergence.
 - (b) Select η . Large η is helpful to speed up the convergence. However, the value of η is proportional to $\tilde{u}$. Too large value of η may cause too large control signal.
 - (c) Select λ . From our experience, $1 \leq \lambda \leq 10$ is a better choice.
 - (d) Select ρ and $\alpha = \rho\lambda$. If oscillatory response is allowed, ρ can be selected larger than 1. If not, smaller ρ is suggested.
 - (e) Estimate the transient properties:
 - i. The overshoot can be estimated from damping ratio $\zeta = (1 + 2\rho)/\sqrt{2}\rho$.
 - ii. Estimate the total time constant. Since the convergence procedure is $s \rightarrow 0$ and then $e \rightarrow 0$, the total time constant of convergence, τ_e can be approximated by $\tau_e = \tau_\eta + \tau_\alpha$.
 - iii. The settling time t_s , according to control theory, approximated by $t_s = 4\tau_e$.
 - (f) Build the nonlinear mathematical model and run numerical simulations with initially selected parameters. Tune the parameters according to the transient performance and the effect of parameters.
 - (g) Remember to keep α as small as possible, because large α usually indicates a very large control input.
 2. We have re-formulate the dynamics of the TWVS and take away disturbances. At beginning, the derivation includes disturbances so that the model is more complete. However, we realized that this is unnecessary because the control law does **NOT** measure disturbances. The control actually uses the bound of system variations, which already consider

potential disturbances. As a result, in the revised version, we take away disturbances when re-formulating the dynamics.