Second order sliding mode control with adaptation

Please check the document version of this publication:

• A submitted manuscript is the version of the article upon submission and before peer-review. There can be important differences between the submitted version and the official published version of record. People interested in the research are advised to contact the author for the final version of the publication, or visit the DOI to the publisher's website.
• The final author version and the galley proof are versions of the publication after peer review.
• The final published version features the final layout of the paper including the volume, issue and page numbers.

Link to publication

Download date: 04. Oct. 2023

Second order sliding mode control with adaptation

Bram de Jager
Faculty of Mechanical Engineering, WH 2.137
Eindhoven University of Technology, P.O. Box 513, 5600 MB Eindhoven, The Netherlands
Email: jag@wfw.wtb.tue.nl, Fax: 00 31 40 447355

Abstract

The paper proposes a new control scheme for nonlinear robotic systems. The base of the scheme is an existing second order sliding mode controller augmented with an adaptive component. A computed-torque-like adaptive controller with a second order system type of measure of tracking accuracy is the source of the adaptation law.

Extensive simulations of a nonlinear mechanical system, a robot with one rotational and one translational degree-of-freedom with unmodeled actuator dynamics and parameter mismatch, confirm the suitability of the proposed scheme.

In the presence of parameter errors adaptation reduces the tracking error substantially without giving up robustness, by still guaranteeing attractiveness of the sliding surface. On the other hand, adaptation allows smaller feedback gains to be used, by that improving robustness without the cost of reduced performance.

Adaptation may improve the performance when unmodeled actuator dynamics is present, although to a lesser degree and not uniformly. The stability robustness in this case is slightly worse.

1. Introduction

For the robust control of nonlinear systems at least two possibilities are known, *adaptive* and *robust* control. Adaptive control is aimed at reducing the performance degradation due to unknown or imperfectly known parameters. Robust control aims at conquering the effects of more general model errors. Sliding mode control is especially suitable as a robust control methodology if the structure of the model errors is unknown but matched, and only an upper bound for their effects can be derived. It guarantees robust performance in the presence of model errors and persistent disturbances, but at the cost of increased control authority. When an increase of the control authority is not desired, robustness can often be attained only by reducing nominal performance.

To reduce the control action one could increase the model accuracy. A way to achieve this is to find more accurate model parameters, like they can be generated by an adaptive controller. By merging adaptive and sliding mode control a controller robust against parameter errors, unstructured model errors, and persistent disturbances seems possible, without too much, or even any, increase in control authority.

Several variants of sliding mode control exist. A sliding mode control with second order sliding condition has been proposed [1] to increase the tuning capabilities compared with a controller based on a first order sliding condition. Also, an adaptive controller has been proposed [2] that uses a second order type measure of tracking accuracy for the adaptation law. It seems profitable to merge the two approaches and by that to get better tuning capability and robustness than is possible with a standard first order sliding mode controller [3].

The contributions of this paper are a derivation and evaluation of a new control scheme, an evolution of two other control schemes. These schemes are presented, and a straightforward combination thereof is derived. The usefulness of the newly proposed scheme is assessed by extensive simulations on a nonlinear mechanical system, a rotation translation robot.

The next section discusses the standard model used to describe the mechanical system to be controlled. Is also presents the control schemes. The simulation results, discussed in Section 3, allow us to draw some conclusions in the last section. Finally, the Appendix contains additional material related to the simulation model and controller settings, to allow independent verification of the results.

2. Model and controller

The system to be controlled is assumed to be a mechanical system with n degrees-of-freedom q whose dynamics can be represented by the model

$$H(q, \theta)\ddot{q} + C(q, \dot{q}, \theta)\dot{q} + g(q, \dot{q}, \theta) = f \quad (1)$$

with θ the model parameters and f the generalized force generated by the actuators. The inertia matrix H is positive definite, the column $C\dot{q}$ represents Coriolis and centrifugal forces while the column g

contains gravity and friction terms. The matrix C, if chosen appropriately, has the property that $\dot{H} - 2C$ is skew symmetric. Remark that by choosing this model structure we do not allow for joint or link flexibility, or for actuator or sensor dynamics. These effects contribute to the unmodeled dynamics.

A SOSMC (second order sliding mode controller) for this model is given by [3]

$$f_s = \hat{H}(\ddot{q}_d + K_d\dot{\tilde{q}} + K_p\tilde{q}) + \hat{C}\dot{q} + \hat{g}$$
$$+ \hat{H}(-\Lambda\dot{s} + \Omega s + K_s\,\mathrm{sgn}\,\dot{s})$$

with K_s large enough to guarantee attractiveness of the sliding surface $\dot{s} = 0$, where the measure of tracking accuracy s is defined by

$$\dot{s} + \Lambda s = \dot{\tilde{q}} + K_d\tilde{q} + K_p\int_0^t \tilde{q}\,d\tau. \qquad (2)$$

Here $\hat{H} = H(q, \hat{\theta})$, etc., indicate estimates, while $\tilde{q} = q_d - q$ is the tracking error with q_d the desired trajectory.

The adaptive computed-torque-like controller proposed by Kelly [2] is

$$f_k = \hat{H}(\ddot{q}_d + K_d\dot{\tilde{q}} + K_p\tilde{q}) + \hat{C}\dot{q} + \hat{g} + \hat{C}v$$

with v a filtered version of the tracking error $\tilde{q}$

$$\dot{v} + \Lambda v = \ddot{\tilde{q}} + K_d\dot{\tilde{q}} + K_p\tilde{q}. \qquad (3)$$

The adaptation law is

$$\dot{\hat{\theta}} = \Gamma Y_k^T v$$

with Y_k derived from a linear parameterization of the control law

$$f_k = Y_k(q, \dot{q}, \ddot{q}_d + K_d\dot{\tilde{q}} + K_p\tilde{q}, v)\hat{\theta}.$$

It is assumed that this parameterization is possible. For inertia parameters this is no strict assumption, but some friction models need approximations before they can be linearly parameterized [4,5].

Comparison of both schemes leads to the proposal of the following adaptation law for the second order sliding mode control scheme (because v can be identified with $\dot{s}$)

$$\dot{\hat{\theta}} = \Gamma Y_s^T \dot{s}$$

where now

$$f_s = Y_s(q, \dot{q}, \ddot{q}_d + K_d\dot{\tilde{q}} + K_p\tilde{q} - \Lambda\dot{s} + \Omega s + K_s\,\mathrm{sgn}\,\dot{s})\hat{\theta}.$$

Some remarks on these control schemes follow.

In practice the sgn function can cause chattering and is therefore replaced by a smoothed variant, for which the sat function, defined component wise by

$$\mathrm{sat}(x, \Phi) = \begin{cases} \mathrm{sgn}(x) & \text{for } |x| > \Phi \\ \Phi^{-1}x & \text{otherwise} \end{cases}$$

is selected, although several other common variants exist. The width Φ of the band in which the sgn function is smoothed is a controller parameter to be selected by the designer. It should be chosen as small as possible to increase the "guaranteed" performance, but still large enough to avoid chattering. A way to suppress chattering without giving up performance is treated in [6].

The controller matrices K_d, K_p, Λ, Ω, and Γ are required to be positive definite to get the desired stability properties. In practice the actual values used are limited, e.g., by bandwidth considerations, to avoid exciting unmodeled dynamics and by that causing instability. The unmodeled dynamics can be neglected joint or link flexibility, actuator or sensor dynamics, sampled data controller implementation, computational time delay in the controller, etc. See [3] for some guidelines on the practical tuning of the controller parameters.

It is possible, without endangering stability, to drop the term $-\Lambda\dot{s}$ in f_s, by that simplifying the control law. This follows from an analysis of the Lyapunov function V used in the stability proof of the SOSMC. Removing the term mentioned improves its time derivative $\dot{V}$.

It proved profitable in practice to clip the time derivative in the parameter update laws. This is even necessary whenever the initial measure of tracking accuracy is large, as happens when the initial state of the system is not close to the desired one. The resulting large time derivative of $\hat{\theta}$, having no relation at all to parameter errors, leads to extreme variations in the estimated parameters, often in the wrong direction. To avoid too large excursions or even instability, the adaptation gain Γ is restricted. A small Γ leads consequently to a slow adaptation and this reduces performance considerably. Clipping $\dot{\hat{\theta}}$ gets past these problems and allows a large Γ.

3. Simulations

The system used to evaluate the controller is an RT-robot with two degrees-of-freedom, a rotational and a translational joint, both indirectly driven by a motor and transmission unit. For a sketch of the system and some notation see Fig. 1.

The model based part of the control law uses a nominal model of this system. An extended model, including motor dynamics, is used as the system to be controlled in the simulation. More details of these models and the parameter data are in the Appendix.

To verify the hoped for qualities of the proposed scheme extensive simulations were done. Some of the ensuing results are presented.

These results are for tracking of a circular trajectory, with center at $x_c = .5$ [m], $y_c = 0$ [m], and radius $r_c = .25$ [m]. One complete revolution takes 2 [s].
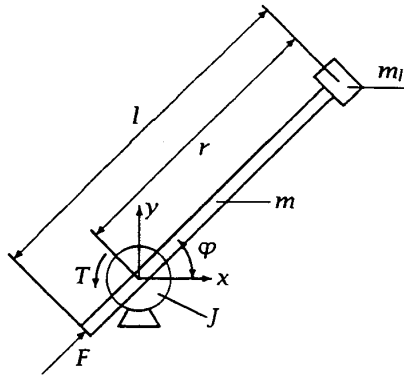
1176

Figure 1: Sketch of RT-robot

Figure 2: RMS of tracking errors for both degrees-of-freedom against initial parameter estimates for several values of the adaptation parameter y

The presented data uses as measure of tracking accuracy the RMS (root mean square) of the components of the tracking error. Before computing the RMS the error $\tilde{\varphi}$ in angular direction is scaled with the desired state r_d to avoid comparing apples and oranges.

Only data for the second revolution is used to compute the RMS. This allows for settling of the parameter adaptation and avoids littering the results with initial transient effects caused by the discontinuous change in the derivative of the tracking error $\tilde{q}$ at $t = 0$ (the desired angular velocity $\dot{\varphi}_d(0) \neq 0$ while the initial velocity $\dot{\varphi}(0) = 0$).

The following factors influencing the tracking error are studied

1. adaptation gain
2. errors in initial parameter estimates
3. sensitivity to unmodeled dynamics
4. simplification of the control law.

At least two ways are open to introduce parameter errors. First, change the model parameters θ. Secondly, vary the initial parameter estimates $\hat{\theta}(0)$. In the first way the baseline performance changes with changes in the model parameters. The second way has the advantage of equal expected performance after convergence of the parameter estimates, because the controlled system does not change. So in a graph of performance against (initial) parameter error a straight line is expected. Therefore parameter errors are introduced by varying $\hat{\theta}(0)$.

Figure 2 illustrates the influence of the first two factors by giving a comparison between adaptive second order sliding mode controllers with different adaptation gains for several values of the initial parameter estimates.

The curves may be clipped because either the tracking error became large, or the system went unstable. Without adaptation the last occurred for small initial parameter estimates. The curves, parameterized
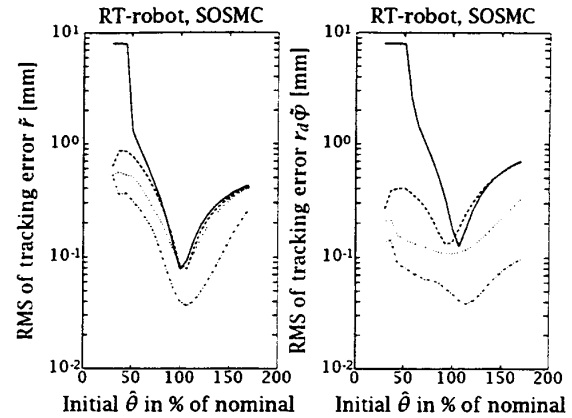
by the scalar adaptation parameter y, can be distinguished by using the next legend

y	linetype
0	solid
1	dashed
10	dotted
100	dash-dotted

The following observations are made.

- The adaptation has not converged yet, this can be concluded because the curves for $y > 0$ are not flat. From other simulations over an extended period it is seen that the performance flattens out at the level obtained for correct initial parameters. For $y < \pm 30$ this takes considerable time.
- With adaptation it is possible to compensate part of the effects of unmodeled dynamics. This follows from the lower error around the nominal parameters, but it occurs only for larger adaptation gains, and especially in angular direction.
- Without adaptation the system is only stable if the initial parameter estimates are at least 50% of their nominal values. With adaptation the system is stable for all initial parameters investigated. Adaptation therefore improves stability robustness because the onset of instability for small initial parameters is shifted markedly, even for small y.
- Robust performance is increased because the larger the parameter errors the more relative improvement the adaptive controllers bring. This is especially true for the angular direction, but could also reflect the situation for the radial direction after additional time for parameter convergence.

While during the previous simulations the model parameters were fixed and the controller parame-

ters, *i.e.*, the initial parameter estimates, were varied, to assess the sensitivity to unmodeled dynamics, which have no related controller parameters, the simulation model parameters are changed.

Here, the parameters of the motor dynamics were varied, while the initial parameter estimates were fixed at their nominal values. Analyzing the effects of unmodeled dynamics in this way is somewhat artificial, but suffices to assess the robustness for parameter errors and unmodeled dynamics. A more rigorous approach is to study the effects of production variations in motor data or the use of different types of motors. This was investigated also, but the results were not quite as illustrative as the presented ones.

Results for variation of the motor's electrical time constant τ_e and mechanical time constant τ_m are in Figs. 3 and 4 respectively. The nominal values of the time constants are indicated by vertical lines.
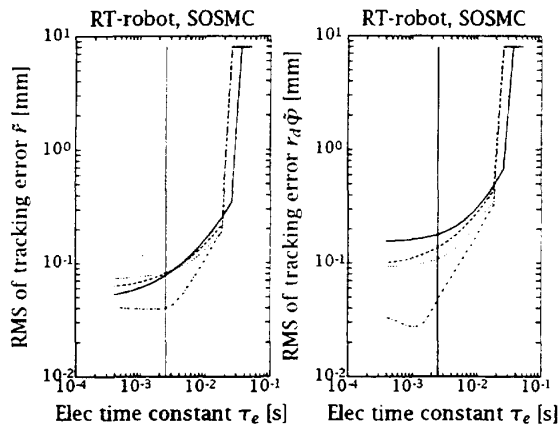
Figure 3: RMS of tracking errors for both degrees-of-freedom against electrical time constant for several values of the adaptation parameter y

For τ_e variations, a large τ_e makes the system unstable. Adaptation, especially for large y, improves the performance, although not uniformly, but it does not improve robust stability because the onset of instability is already at smaller values of τ_e if $y > 0$.

The results for varying mechanical time constant τ_m are inconclusive with respect to the profitability of adaptation. In the region around the nominal time constant adaptation is exceptionally profitable, which is pure coincidence. For smaller τ_m adaptation has detrimental effects, especially in radial direction. For larger τ_m adaptation is consistently profitable.

The increase of the tracking errors with decreasing τ_m may be counter intuitive. It is however easy to explain by considering the relation between the influence of the motor constant k_m, used to vary τ_m,
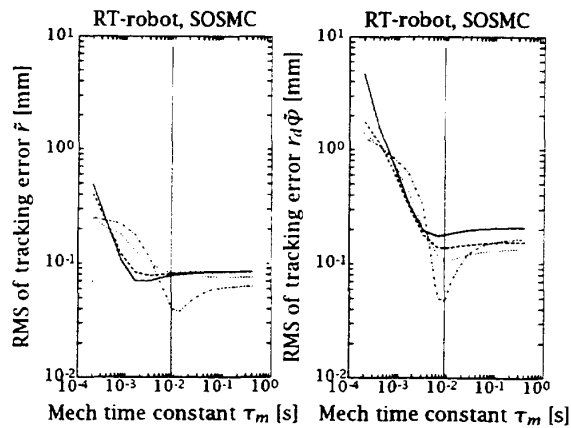
Figure 4: RMS of tracking errors for both degrees-of-freedom against mechanical time constant for several values of the adaptation parameter y

and the back-emf effect. Increasing k_m, so decreasing τ_m, does increase the effects of the back-emf, making the motor dynamics more pronounced. See the Appendix for the relevant equations.

Unlike the results for errors in the initial parameters, which could be interpreted straightforwardly, the results for unmodeled dynamics are less clear. It is hard to explain why there is such a difference in results for positive or negative variations in τ_e and τ_m and between radial and angular direction. The changes in tracking errors are caused by a complex interplay between the nonlinear adaptation mechanism and the effects of the motor dynamics on the nonlinear system. An additional complication is the sampled data implementation of the controller with its associated sampling rate. A fundamental explanation seems hard to get by.

From these results the conclusion should be that adaptation has it shortcomings when unmodeled dynamics is present. Fortunately, the sliding mode character of the control provides already for some robustness for these types of model errors.

Finally, we evaluate the modification of the basic adaptive SOSMC by dropping the term $-\Lambda \dot{s}$. Figure 5 shows the results for initial parameter errors for both control schemes. Comparing the curves reveals that removing the indicated term improves the performance slightly and has no detrimental effects. The main advantage is that, without adaptation, the stability range of the controlled system is increased. This follows from the later onset of instability for small values of $\hat{\theta}(0)$.

Simulations for the modified control scheme with varying unmodeled dynamics only show insignificant differences. It does not deserve a detailed presentation.
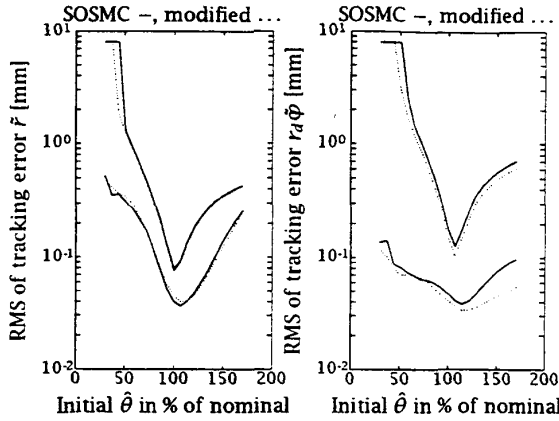
Figure 5: RMS of tracking errors for both degrees-of-freedom against initial parameter errors for $y = 0$ and $y = 100$, original and modified control law

4. Conclusion and recommendation

The proposed control scheme does improve substantially over its nonadaptive counter part. It is especially suitable for systems with substantial parameter errors. Adaptation is less profitable in the presence of significant unmodeled dynamics, or even not profitable at all.

To further assess the potential of the proposed control scheme, experiments are a conditio sine qua non. Extensive evaluation and validation, on at least one mechanical system, is expected to be performed shortly.

Acknowledgment

References

[1] L.-W. Chang, "A MIMO sliding control with a second-order sliding condition," in Proc. of the ASME Winter Annual Meeting, (Dallas, Texas), pp. 1-7, ASME, 1990. Paper 90-WA/DSC-5.

[2] R. Kelly, "Adaptive computed torque plus compensation control for robot manipulators," Mech. Mach. Theory, vol. 25, pp. 161-165, Feb. 1990.

[3] A. Blom and A. G. de Jager, "Robustness of a second order sliding mode controller," in Proc. of the 1993 ASME Winter Annual Meeting (E. A. Misawa, ed.), vol. DSC-Vol. 53, (New Orleans, Louisiana), pp. 71-77, New York: ASME, Dec. 1993.

[4] C. Canudas de Wit, P. Noël, A. Aubin, and B. Brogliato, "Adaptive friction compensation in robot manipulators: Low velocities," Internat. J. Robotics Res., vol. 10, pp. 189-199, June 1991.

[5] B. de Jager, "Adaptive compensation of torque disturbances and beyond," in Proc. of the Internat. Conf. CONTROL'94, vol. 2, (Coventry, UK), pp. 1040-1044, IEE, Mar. 1994.

[6] M. M. J. van de Wal, A. G. de Jager, and F. E. Veldpaus, "Design and application of a dynamical sliding mode controller for uncertain nonlinear MIMO systems," in Proc. of the Internat. Conf. CONTROL'94, vol. 1, (Coventry, UK), pp. 866-871, IEE, Mar. 1994.

[7] J. Banens, "Documentation on TCE modules," Report WFW 94.050, Eindhoven University of Technology, Eindhoven, The Netherlands, Apr. 1994.

Appendix

This section contains the models and data of the RT-robot, see Fig. 1, used for design purposes and in the simulations. The equations of motion for the two degrees-of-freedom system are

$$\theta_1 \ddot{r} - (\theta_1 r - \theta_2)\dot{\varphi}^2 = F \tag{4}$$

$$(\theta_1 r^2 - 2\theta_2 r + \theta_3)\ddot{\varphi} + 2(\theta_1 r - \theta_2)\dot{r}\dot{\varphi} = T$$

where

$$\theta_1 = m + m_l$$

$$\theta_2 = \frac{1}{2}ml$$

$$\theta_3 = J + \frac{1}{3}ml^2.$$

The equations do easily fit in the general model (1) by taking

$$q = \begin{bmatrix} r \\ \varphi \end{bmatrix}.$$

It is assumed that the manipulator moves in the horizontal plane, so gravity terms are not included.

The parameterization used is minimal and linear. Not all parameterizations are linear, e.g., it is not possible to use the length l as parameter because it enters nonlinearly in the differential equations. When using these equations of motion and parameterization, the following expressions result for the matrices Y_s and Y_k used in the control laws

$$Y_s = \begin{bmatrix} h_r - r\dot{\varphi}^2 & \dot{\varphi}^2 & 0 \\ r(rh_\varphi + 2\dot{r}\dot{\varphi}) & -2rh_\varphi - 2\dot{r}\dot{\varphi} & h_\varphi \end{bmatrix}$$

with

$$\begin{bmatrix} h_r \\ h_\varphi \end{bmatrix} = \ddot{q}_d + K_d\dot{\tilde{q}} + K_p\tilde{q} - \Lambda\dot{s} + \Omega s + K_s \operatorname{sat}(\dot{s}, \Phi)$$

and

$$Y_k =$$

$$\begin{bmatrix} h_r - r\dot{\varphi}\bar{v}_\varphi & \dot{\varphi}\bar{v}_\varphi & 0 \\ r(rh_\varphi + \dot{\varphi}\bar{v}_r + \dot{r}\bar{v}_\varphi) & -2rh_\varphi - \dot{\varphi}\bar{v}_r - \dot{r}\bar{v}_\varphi & h_\varphi \end{bmatrix}$$

with

$$\bar{v} = \dot{q} + v,$$

$$\begin{bmatrix} h_r \\ h_\varphi \end{bmatrix} = \ddot{q}_d + K_d\dot{\tilde{q}} + K_p\tilde{q}.$$

These expressions are used to compute both the control laws according to $f = Y\hat{\theta}$ and the parameter update laws $\dot{\hat{\theta}} = \Gamma Y_s^T \dot{s}$ or $\dot{\hat{\theta}} = \Gamma Y_k^T v$, where $\dot{s}$ and v follow from (2) and (3). The initial values for s and v were taken equal to zero. Changing these values can have a marked influence on the initial transient.

To accommodate unmodeled dynamics the equations of motion (4) are extended with actuator dynamics for use in a simulation model. Both motors are assumed to be of the same type and are accounted for by the differential equation

$$\dot{f} = \frac{R}{L}(u - f) - \frac{1}{L}\left(\frac{k_m}{i}\right)^2 \dot{q}$$

and by modifications of inertia parameters to include the moments of inertia of motors and transmissions. Here u is the (new) controller command and R, L, and k_m are constant motor parameters, whose meaning is clear from the usual notation. The parameter i is the transmission ratio.

In the stationary state, so $\dot{f} = 0$, and with zero velocity, the torque f applied on the system with or without motor dynamics is the same. When the motor is turning, so $\dot{q} \neq 0$, but still stationary, so $\dot{f} = 0$, the back-emf reduces the available torque f. For instationary events the dynamics of the motor, partly determined by the electrical time constant $\tau_e = \frac{L}{R}$, plays a more intricate role.

The data used in the simulations is given in Table 1 for the basic controller data and in Table 2 for the basic data in the simulation model.

Table 1: Controller data

Parameter	Value	Unit
$\hat{\theta}_1(0) = m + m_l$	15	kg
$\hat{\theta}_2(0) = \frac{ml}{2}$	5	kg m
$\hat{\theta}_3(0) = J + \frac{ml^2}{3}$	$\frac{25}{3}$	kg m^2
K_p	15^2	s^{-2}
K_d	$2 \cdot 15$	s^{-1}
Ω	15^2	s^{-2}
Λ	$\begin{bmatrix} 1 & 0 \\ 0 & 5 \end{bmatrix}$	s^{-1}
Γ	$\gamma \begin{bmatrix} 10 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$	SI
K_s	10	SI
Φ	0.47	SI

The controller parameter matrices, unless indicated otherwise, are taken diagonal with identical values. When appropriate, only those values are in the table. The adaptation gain Γ is a diagonal matrix of a fixed structure with varying scalar gain γ.

Table 2: Simulation data

Parameter	Value	Unit
$\theta_1 = m + m_l + \frac{J_m}{i_r^2} + m_r$	15.51	kg
$\theta_2 = \frac{ml}{2}$	5.00	kg m
$\theta_3 = J + \frac{ml^2}{3} + \frac{J_m}{i_\varphi^2} + m_\varphi$	8.85	kg m^2
k_m	$70 \cdot 10^{-3}$	Nm A^{-1}
R	1.6	Ω
L	$4 \cdot 10^{-3}$	H
J_m	$27 \cdot 10^{-6}$	kg m^2
m_r	0.5	kg
m_φ	0.5	kg m^2
$\tau_e = \frac{L}{R}$	$2.5 \cdot 10^{-3}$	s
$\tau_m = \frac{J_m R}{k_m^2}$	$8.8 \cdot 10^{-3}$	s
i_r	0.05	m
i_φ	0.04	-

The data of the mechanical system does not represent a real system. The motor data stems from the specifications of an Electro Craft E-540 SA motor. The values of m_r and m_φ take account of the mass and moment of inertia of the transmission between motor and link. These values are not based on real data. The transmission ratios i_r and i_φ are used to fit the force and torque requirements of the mechanical system and its desired trajectories to the torque characteristics of the motor.

For the simulations with parameter variation the controller data for the initial estimated parameters $\hat{\theta}(0)$ was varied, based on their nominal values in Table 1. For the unmodeled dynamic simulations the simulation data for τ_e and τ_m were varied by changing the nominal values of L or k_m. Changing these values did influence only one time constant, not both, see Table 2.

Clipping of $\dot{\hat{\theta}}$, as mentioned in Section 2, was performed at the current value of $\hat{\theta}$. This implies that $\hat{\theta}$ cannot vary faster than a first order system with a time constant of 1 [s]. A good value for this time constant depends of course on the system to be controlled. The value of 1 [s] seemed quite appropriate here, but was still determined ad-hoc.

The controller implementation is a sampled data one with a sampling frequency of 100 [Hz]. The simulation model's differential equations were solved by a third order Runge-Kutta scheme with a precision controlled step size, synchronized with the sampling instances of the controller.

The simulation program is coded in C++, using an object oriented package called TCE (Tools for Control Experiments) [7]. This package includes facilities for matrix and vector computations, time series handling, etc., and it provides a data exchange interface with MATLAB.