Nonlinear Model Reference Adaptive Control (NMRAC) for First Order Systems

Dalim Wahby
Université Côte d'Azur
I3S/CNRS
Sophia Antipolis, France
wahby@i3s.unice.fr

Alvaro Detailleur
IDSC
ETH Zürich
Zürich, Switzerland
adetailleur@student.ethz.ch

Guillaume Ducard
Université Côte d'Azur
I3S/CNRS
Sophia Antipolis, France
ducard@i3s.unice.fr

***Abstract*—**

***Keywords*—Neural Networks, Gradient Descent, Control, Lyapunov Method**

I. Introduction

II. Literature Review and Contributions

III. Methodology

IV. Results

In this section, we will present our simulation for the proposed MRAC strategy for nonlinear NNCs. We carefully construct an example to show that with the proposed schematic we are able to learn the desired parameter. To this end, we choose to use a simple NNC that consists of only one neuron with an activation function, which can be seen as a smooth saturation function, as defined in (**??**). The parameters of the neural network consist of one value that is updated.

For simplicity, we choose the dynamics of the reference model and the system to be equal, and we choose the activation function to be the same. The dynamics of the system and the reference model have a form as defined in equation (**??**) and equation (**??**). The only parameters that are different are θ and θ_m. Hence, to ensure that the behavior of the reference model and the system are the same, we require θ to converge towards θ_m. We choose $a_m = a = 10$, $b_m = b = 10$, $\theta_m = 4$, and the activation function saturates at a value of 1. The precision threshold ϵ is chosen to be computer precision in all our simulations. We compute the weight update as the analytical solution for each time instance. The sampling time for our simulations is $T_s = 0.01$, and the learning rate differs in each simulation to showcase faster and slower convergence.

We will analyze the behavior with respect to different reference signals. Note, that we require the reference signal to be differentiable, because $\dot{r}$ is used in the update law. Hence, we will firstly look at the response of a constant signal, secondly a sine signal, and thirdly a smooth pulse signal.

A. *Constant Reference Signal*

We commence by using a constant reference signal. We choose the system's initial conditions to be at 0. The value of the constant reference signal is 1. This will result in a steady state error for the reference model since we only take a feedback controller into account. However, we will ignore this steady state error, since for us the ability of the controller to learn the behavior of the reference model is of interest. Additionally, we set $\theta_m = 4$ and the initial value of the weight $\theta = 0$. Finally, we simulate the system response for 5 seconds and choose the constant parameter $c = 0.05$.

In Figure **??**, we observe the steady state error, which is at around 0.2. As aforementioned, we will disregard this error, since we are interested in how well our NNC can learn the behavior of our reference model. To this end, we can observe that the system converges to the reference model within 5 seconds. Furthermore, in Figure **??**, we can see the weights of the NNC and the error e are equal to their desired values, $\theta = 4$ and $e = 0$ respectively. In Figure **??**, we can see that the Lyapunov function converges to zero over time. Additionally, we can see that the derivate of the Lyapunov function is negative and only reaches zero when the Lyapunov function itself converges to zero. This indicates that the proposed learning schematic is stable for the desired reference signal, and we can learn the desired parameter.

B. *Sine Signal*

The second reference is a sinusoid signal, with an amplitude of 0.25, to stay in a region that is not affected too much by the saturation function. As before, we choose the initial $\theta = 0$, $\theta_m = 4$, and the initial conditions of both the system and the reference model to be at -1. We choose this specific initial condition only to be able to see the convergence in the plots. Finally, we simulate the system response for 30 seconds and choose the constant parameter $c = 0.15$, which corresponds to a less aggressive learning strategy compared to the constant input signal.

Figure **??** shows the response of the system. We observe that the system state, and reference model state both converge towards the desired reference model within around 25 seconds. Simultaneously, the parameter θ converges to towards the desired value, which can be seen in Figure **??**. Even though the parameter converges, we observe oscillations. This behavior is expected, due to the update law. Since the update law is dependent on $\dot{e}_x$, it is dependent on $\dot{r}$ and $\dot{x}$. We observe that whenever $\dot{e}_x$ switches signs and simultaneously e is small, $\dot{\theta}$ changes sign, and hence the parameter starts oscillating. This behavior can be changed thourgh adapting the learning rate. A more aggressive learning strategy will be showcased in the next example.

In Figure **??** we can see that the Lyapunov function converges to zero over time. Additionally, we can see that the derivate of the Lyapunov function is negative and only reaches zero when the Lyapunov function itself converges to zero. The results of this simulation indicate again that the proposed learning schematic is stable for the desired reference signal, and we can learn the desired parameter.

C. *Smooth Pulse Signal*

The third and last input signal we will look at is a pulse signal. More specifically, we will use a smooth pulse signal, because we require the reference signal to be continuous since we require its derivative in the update law. We construct the smooth pulse signal, by using a sine function and applying a smooth saturation function, as defined in equation (**??**). We choose the signal to have an amplitude of 0.25 and shift the signal such that its minimum value touches upon 0. As before, we choose the initial conditions of the system and plant to be -1. We simulate the system response for 5 seconds and choose the constant parameter $c = 0.005$.

Figure **??** shows the response of the system. We can observe that the system converges towards the desired reference model within just under 1 second. The parameter θ converges to towards the desired value within 2 seconds, which can be seen in Figure **??**. Even though the parameter converges, we observe that it oscillates. This behavior is expected, due to the update law. Since the update law is dependent on $\dot{e}_x$, it is dependent on $\dot{r}$ and $\dot{x}$. We observe that whenever $\dot{e}_x$ switches signs and simultaneously e is small, $\dot{\theta}$ changes sign, and hence the parameter starts oscillating. Similarly, as in the case of the sine input, this behavior can be influenced by decreasing the value of c.

The Lyapunov function and its derivative, as shown in Figure **??** indicate the convergence of the system and the parameters, since V converges to 0 and its derivative is always negative and approach zero when V converges. The results of this simulation indicate that the proposed learning schematic is stable for the desired reference signal, and we are able to learn the desired parameter. The three simulation results strongly indicate that with the proposed update law we are able to learn the desired parameters in the presence of a nonlinear activation function.

V. Conclusion

Acknowledgment

We acknowledge preliminary work of Micha Bosshart towards the approach in of this paper.

[b]0.49

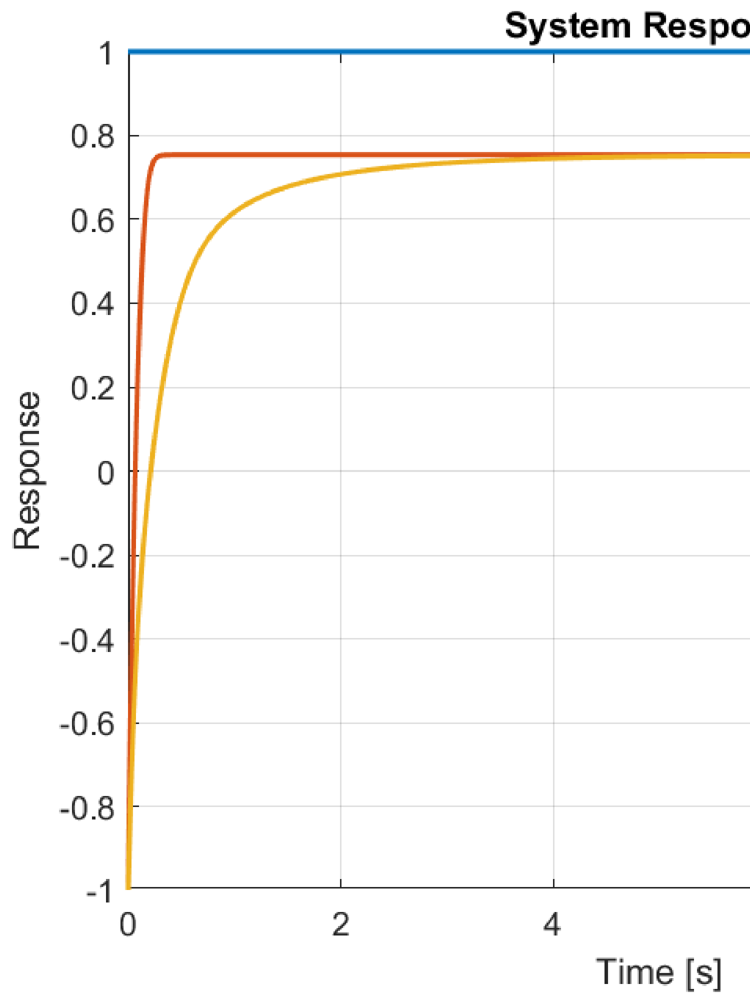
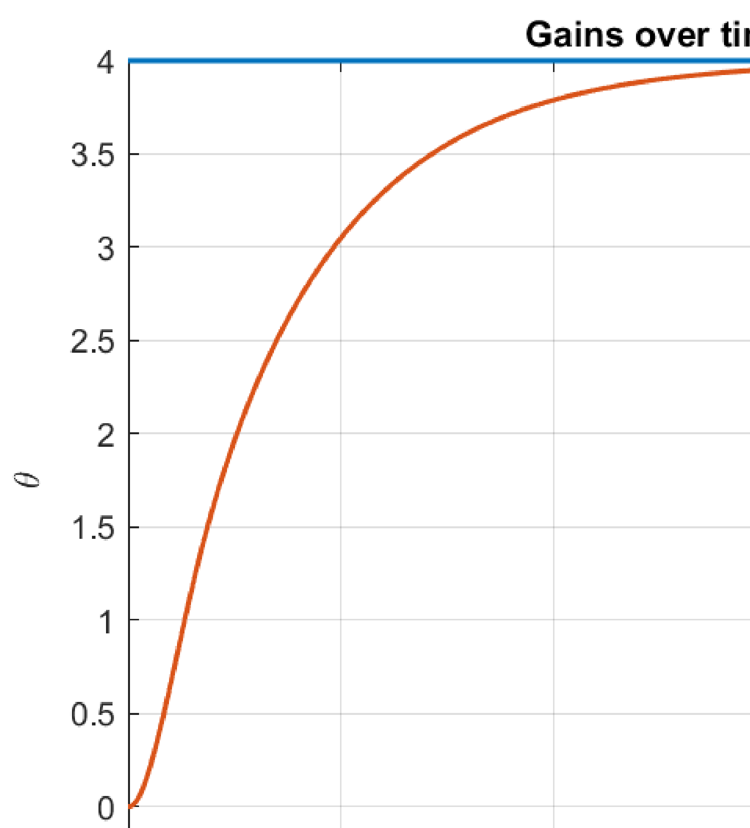


Fig. 1. System response

[b]0.49



[b]0.49

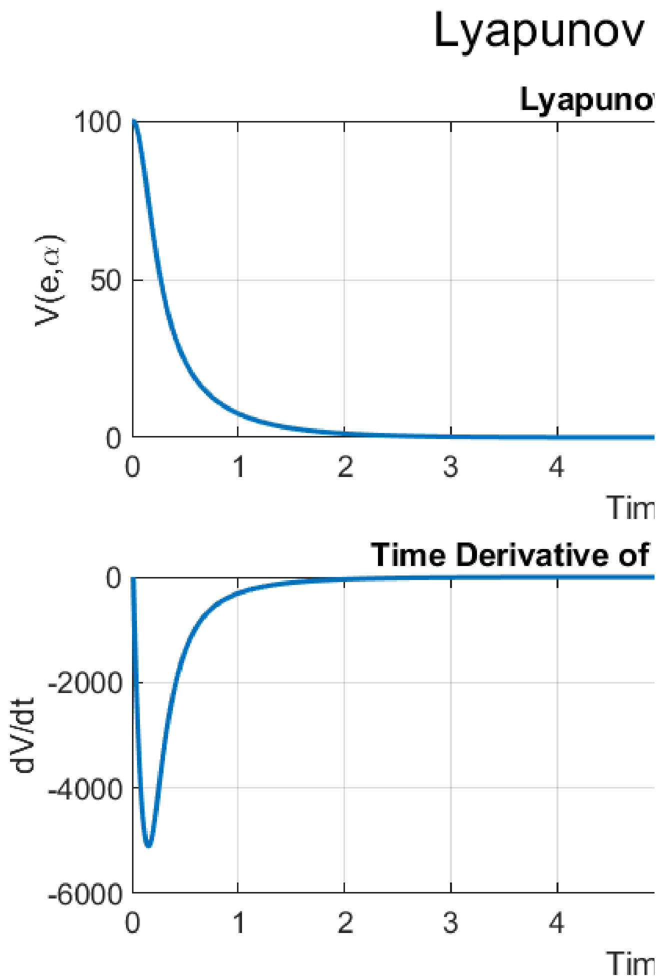


Fig. 4. Lyapunov function and its time derivate

[b]0.49

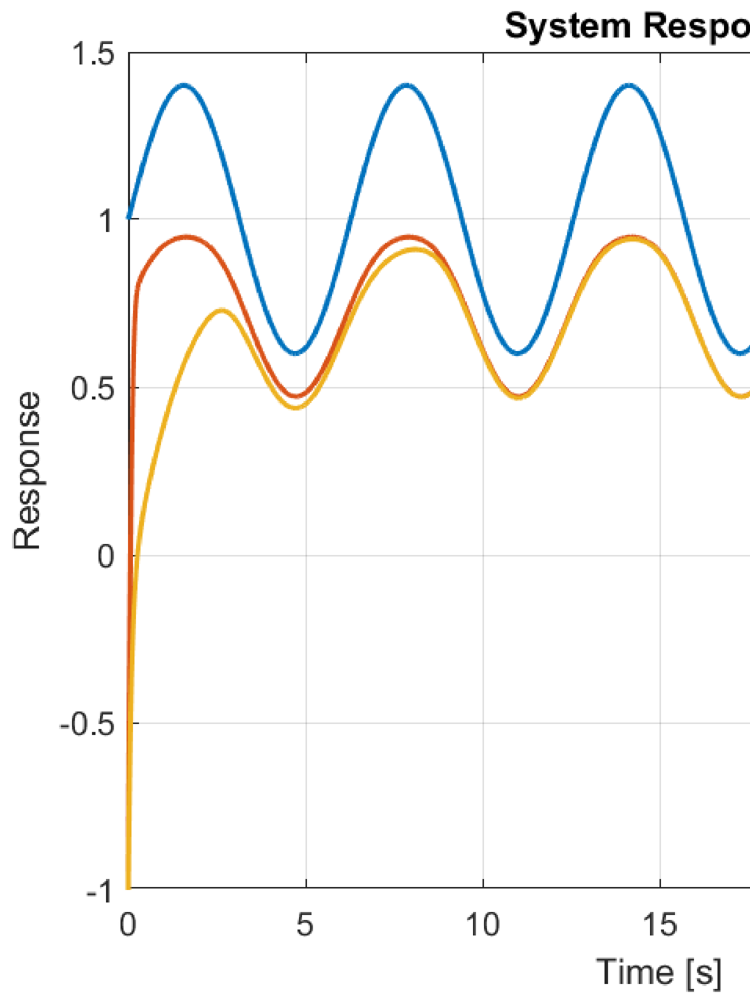
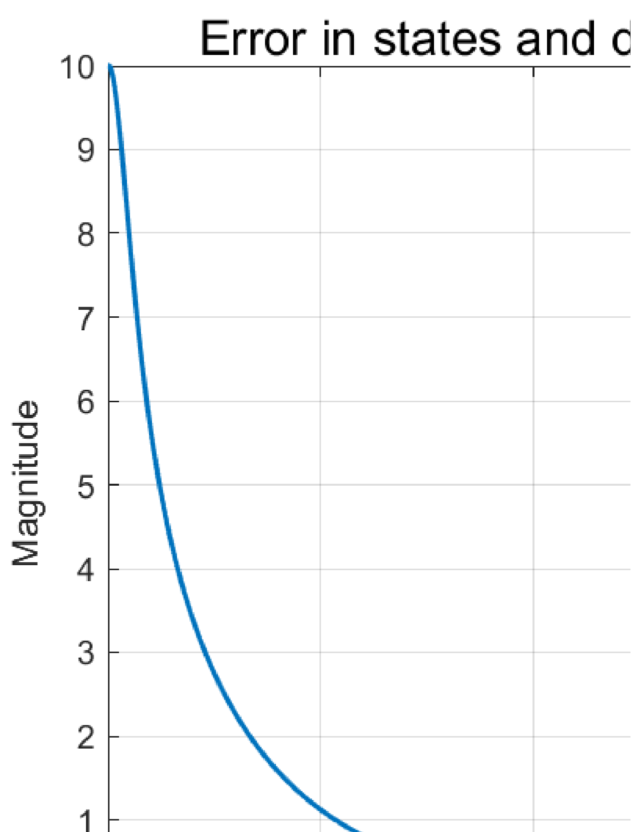
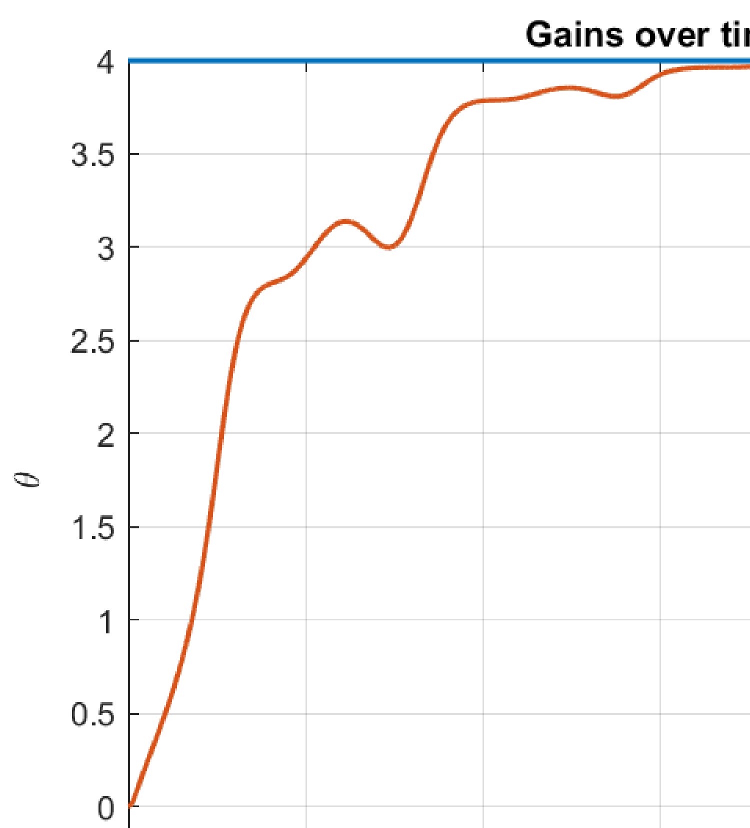


Fig. 7. System response

[b]0.49



[b]0.49



[b]0.49

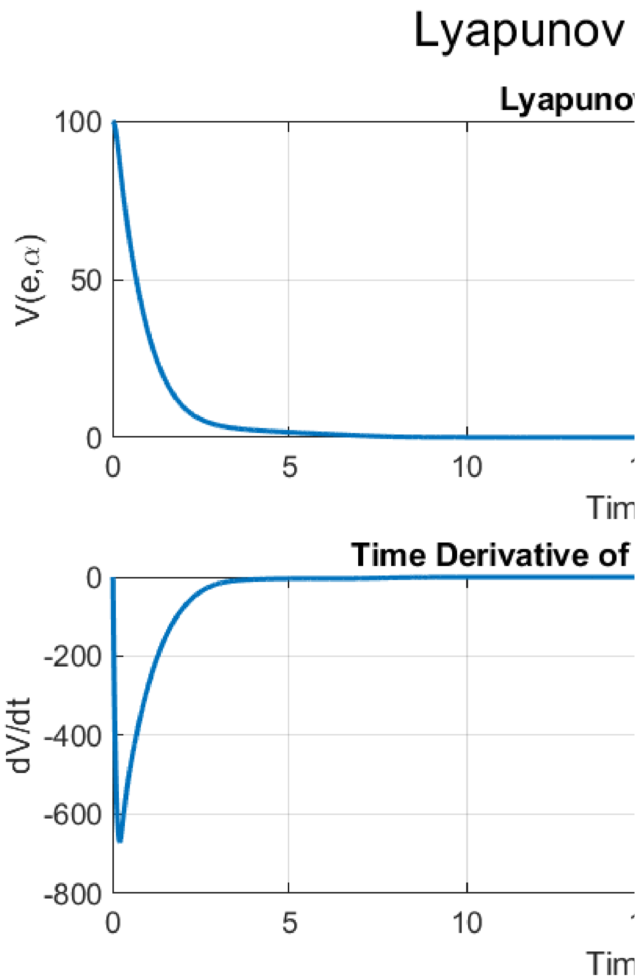


Fig. 10. Lyapunov function and its time derivate

[b]0.49

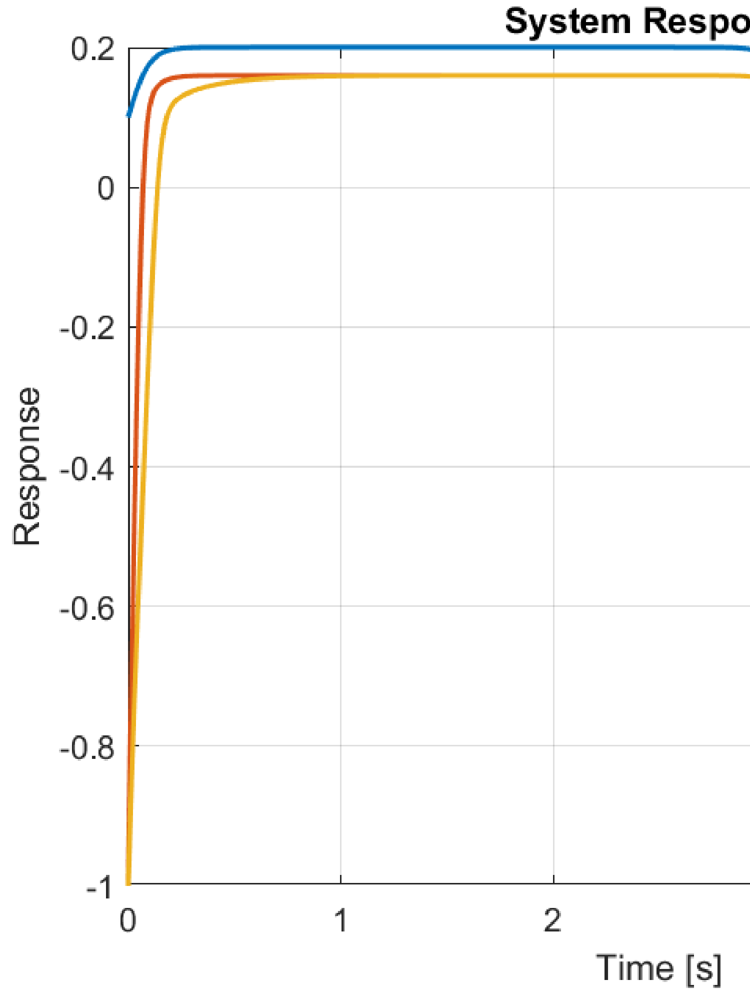
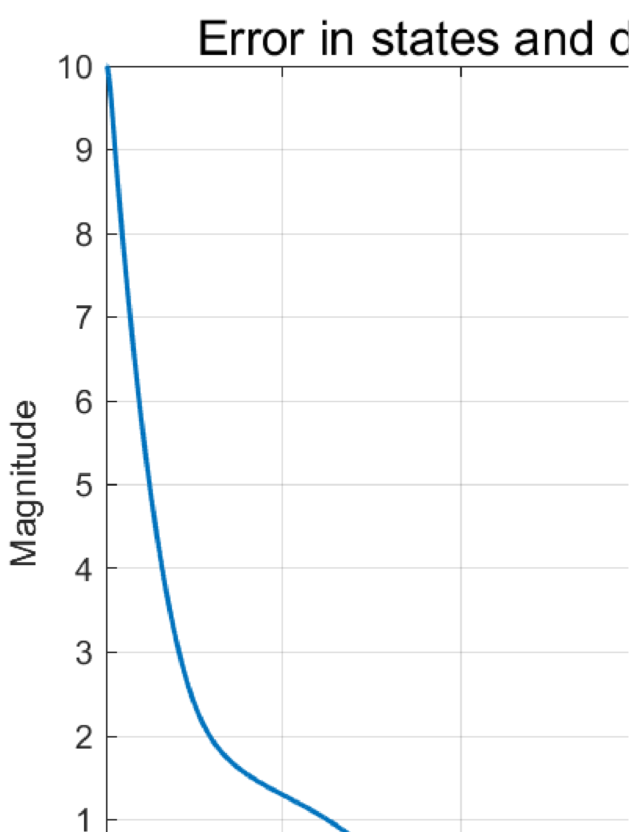
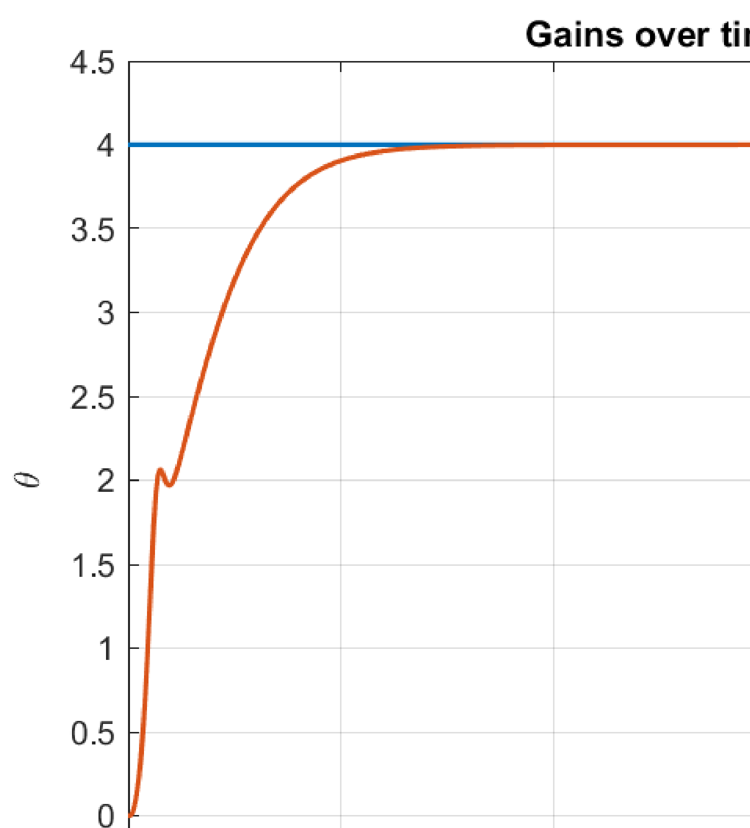


Fig. 13. System response

[b]0.49



[b]0.49



[b]0.49

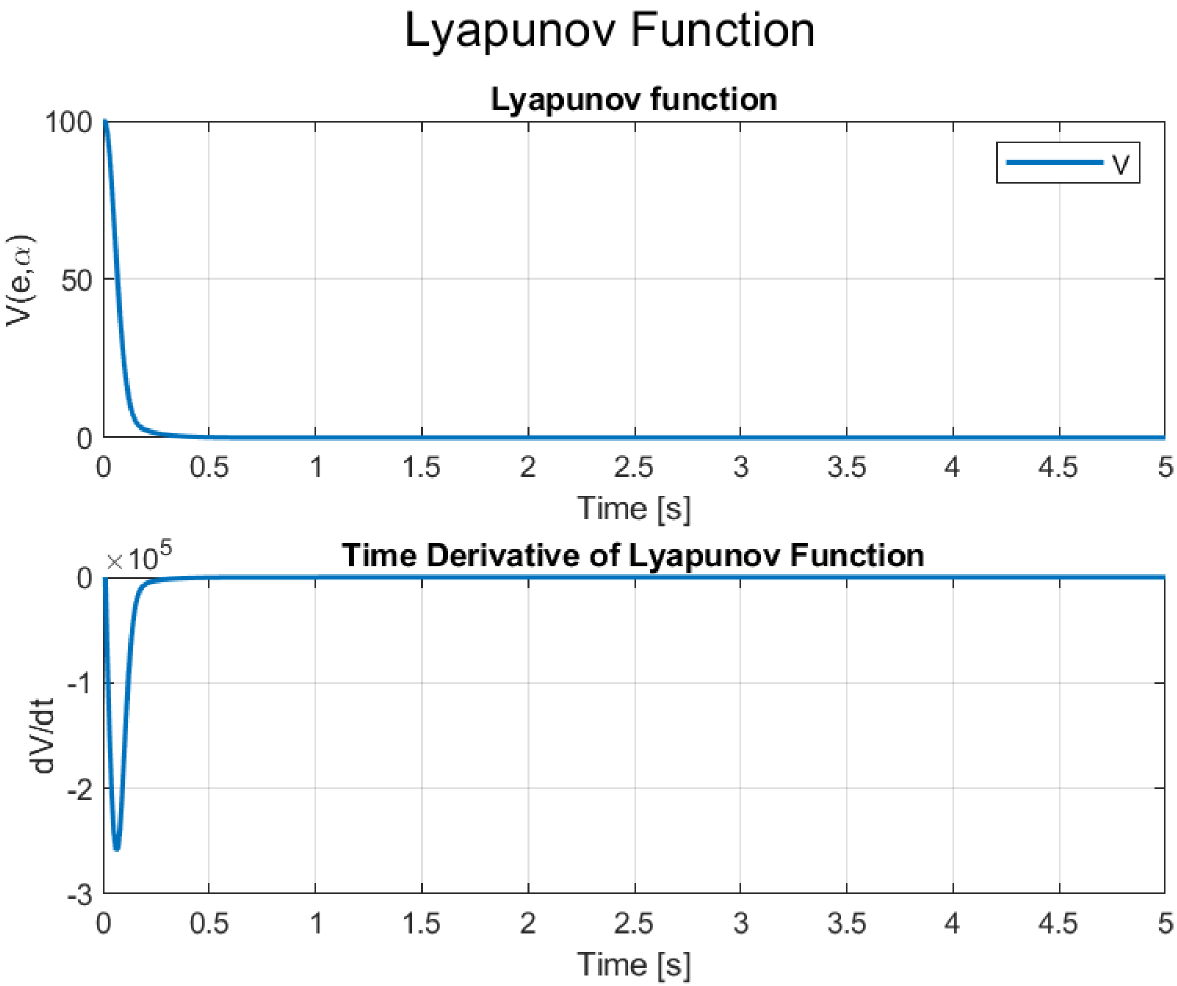


Fig. 16. Lyapunov function and its time derivate

[b]0.49