Nonlinear Control of the Balancing Robot

Course: Nonlinear Systems and Control

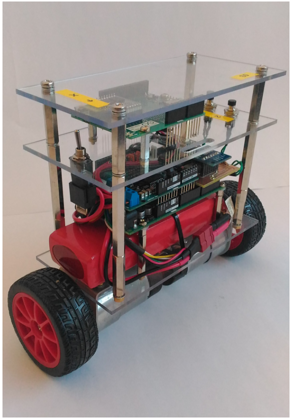
Bilato Alessandro, Peron Davide

June 9, 2023

Università degli Studi di Padova

Activity

Goal: Test the nonlinear tools studied to control the Balancing robot



Outline

Model Description

First step: Identification of the matrix formulation of the dynamical model

$$\tau' = M(q)\ddot{q} + C(q,\dot{q})\dot{q} + F_v'\dot{q} + g(q)$$

$q = [x_1, x_2]^T = [\gamma, \theta]^T$
$\tau' = k[1 - 1]^T u_a$
u_a: is the control input [V]
Description of the matrices:

- M(q): inertia matrix (**NONLINEAR**);
- C(q,$\dot{q}$): matrix of centrifugal and Coriolis-related coefficients (**NONLINEAR**);
- F_v': matrix of viscous friction coefficients;
- g(q): the torque contribution due to the gravity (**NONLINEAR**);

Model description

The state vector is defined as:

$$x = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} \gamma \\ \theta \\ \dot{\gamma} \\ \dot{\theta} \end{bmatrix}, \qquad \mathcal{X} = \mathbb{R} \times \mathbb{S}^1 \times \mathbb{R} \times \mathbb{R}$$

The input space is defined as : $\mathcal{U} = [-255, 255]\,\mathrm{DC} = [-11.1, 11.1]\,V$

The resulting control-affine dynamics are:

$$\dot{x} = \begin{bmatrix} x_3 \\ x_4 \\ -M(x)^{-1}\left(C(x)\begin{bmatrix} x_3 \\ x_4 \end{bmatrix} + F_v{}' \begin{bmatrix} x_3 \\ x_4 \end{bmatrix} + g(x) - \begin{bmatrix} 1 \\ -1 \end{bmatrix} u\right) \end{bmatrix} = \underbrace{\begin{bmatrix} x_3 \\ x_4 \\ f_1(x) \\ f_2(x) \end{bmatrix}}_{f(x)} + \underbrace{\begin{bmatrix} 0 \\ 0 \\ g_1(x) \\ g_2(x) \end{bmatrix}}_{g'(x)} u$$

Simulink Model

- Controller implemented through Matlab functions;
- Switch to chose which input is fed to the controller;

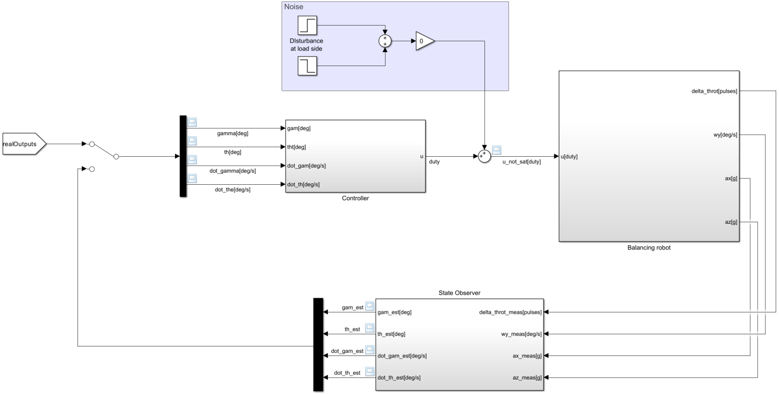


Figure: Simulink Model

Accessibility and Controllability Analysis I

- Very complex analysis due to high complexity of expressions;
- Studied only in the origin;
- We first tried $\left[f(x), g(x), ad_f g(x), ad_f^2 g(x)\right]$ since no notable properties could be assessed to chose a smarter combination;
- This attempt failed since $det\left(\left[f, g, ad_f g, ad_f^2 g\right](\mathbf{0})\right) = \mathbf{0}$
- This was expected since $f(0) = 0$ hence we need to substitute that vector field;
- We need to iterate further...

- We studied $\mathcal{G}(x) = \left[g(x), ad_f g(x), ad_f^2 g(x), ad_f^3 g(x)\right]$ and found out that $det\left(\mathcal{G}(0)\right) \neq \mathbf{0}$;
- We cannot say anything about the neighborhood around which we have accessibility since: $det\left(\left[g, ad_f g, ad_f^2 g, ad_f^3 g\right](x)\right) = \mathbf{0}$ for some x than cannot be computed analytically but only numerically in Matlab.
- We can conclude that the system is STLC at $x = 0$, moreover it is also strongly accessible at $x = 0$;
- To prove accessibility and controllability more in general not only in the origin we should iterate further the adjoint action but this is not feasible from a computational point of view.

Control Lyapunov function: first attempt

First attempt The first Lyapunov function considered is: $V(x) = \frac{1}{2}x^T x$
To be a control Lyapunov function:

$$\dot{V}(x) = \frac{\partial V}{\partial x}(x)f(x) + \frac{\partial V}{\partial x}(x)g(x)u < 0$$

In particular, must exist u such that:

$$\frac{\partial V}{\partial x}g'(x)u < -\frac{\partial V}{\partial x}(x)f(x)$$

Notice:

$$\frac{\partial V}{\partial x}(x)g'(x) = x^T g'(x) = \frac{1}{detM}[(M_{22} + M_{12})x_3 - (M_{12} + M_{11})x_4] \neq 0$$

Sontag's Universal Formula

The first control law considered is Sontag's Universal Formula:

$$u(x) = -\frac{\frac{\partial V}{\partial x}(x)f(x)) + \sqrt{(\frac{\partial V}{\partial x}(x)f(x))^2 + (\frac{\partial V}{\partial x}(x)g'(x))^4}}{\frac{\partial V}{\partial x}(x)g'(x)}$$

as said before

$$\frac{\partial V}{\partial x}(x)g'(x) \neq 0$$

Simulation's results I, γ

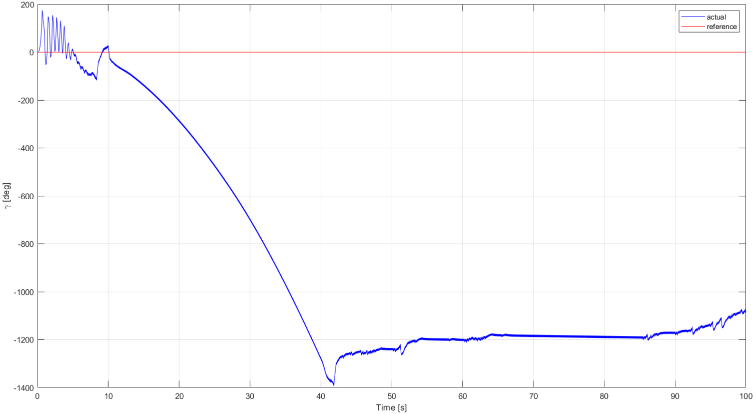


Figure: System controlled using Sontag's Universal Formula

Simulation's results I, θ

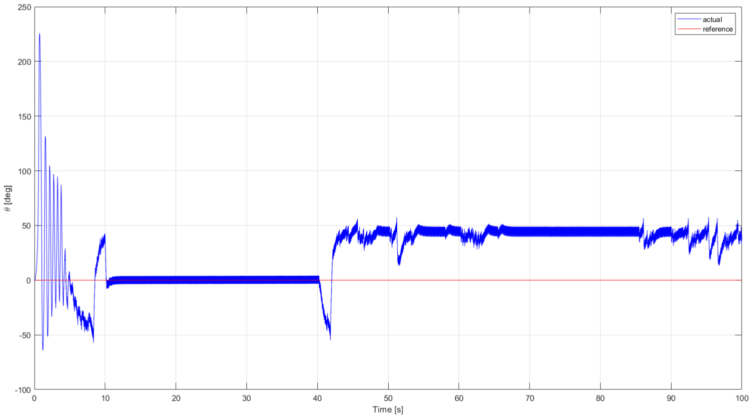


Figure: System controlled using Sontag's Universal Formula

Second control law

Since

$$\frac{\partial V}{\partial x}(x)g'(x) \neq 0$$

is always possible to find:

$$u(x) = -\frac{\frac{\partial V}{\partial x}(x)f(x) + \xi(x)}{\frac{\partial V}{\partial x}(x)g'(x)}$$

where $\xi(x)$ is positive definite. We considered:

$$\xi(x) = K_{gain} \left\|x\right\|^2$$

In this way, the imposed behavior of $\dot{V}$ is:

$$\dot{V} = -\xi(x) = -K_{gain} \left\|x\right\|^2$$

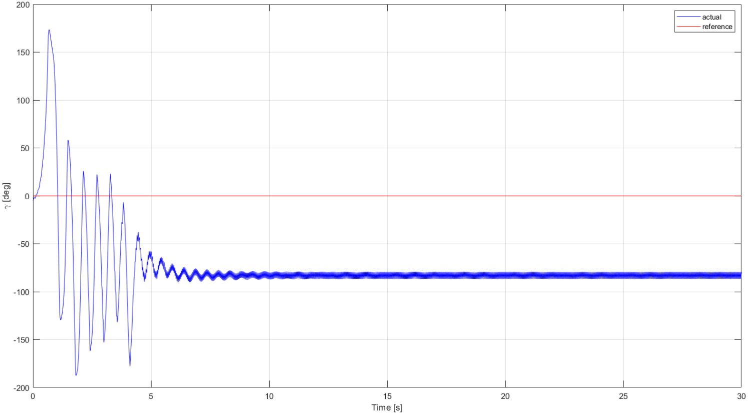


Figure: System controlled using the second control law, with $K_{gain} = 100$

Simulation's results II, θ

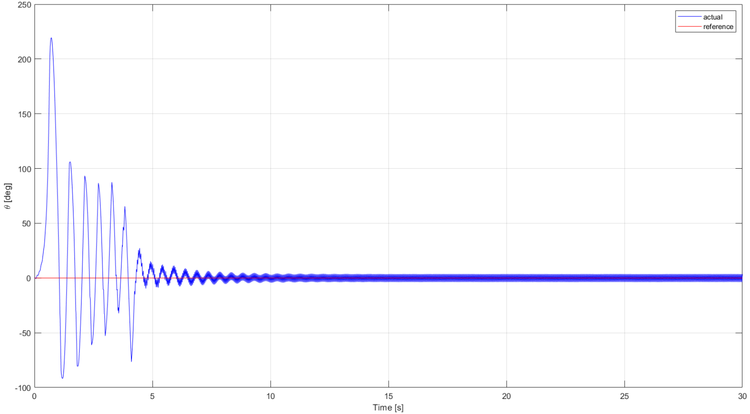


Figure: System controlled using the second control law, with $K_{gain} = 100$

Third control law: Minimum error and gradient-based control law

- Imposing a minimum acceptable convergence rate -W(x).
- $W(x) = K_{gain} \left\| x \right\|^2$

$$u_{me}(x) = -\frac{\frac{\partial V}{\partial x}(x)g'(x)}{\left|\frac{\partial V}{\partial x}(x)g'(x)\right|^2}\left(\frac{\partial V}{\partial x}(x)f(x) + W(x)\right)$$

Simulation's results III, γ

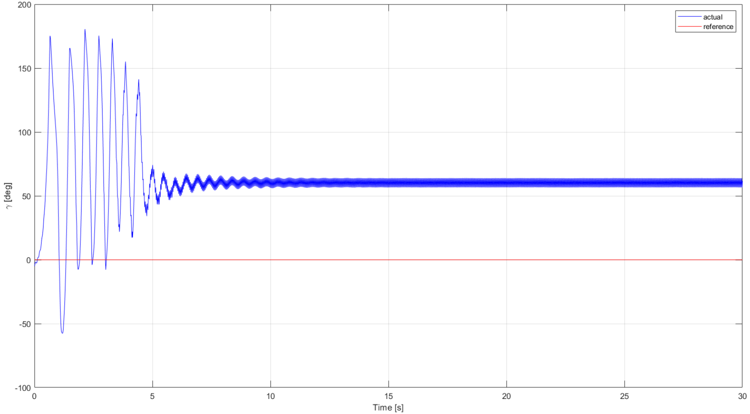


Figure: System controlled using the ME and Gradient-based control law

Simulation's results III, θ

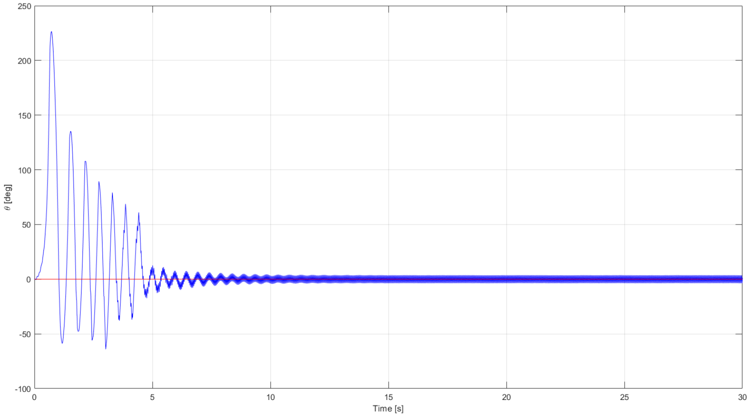


Figure: System controlled using the ME and Gradient-based control law

Another control Lyapunov function

New control Lyapunov function:

$$V(x) = \frac{1}{2} x^T P x$$

where the matrix P is :

$$P = \begin{bmatrix} a & 0 & 0 & 0 \\ 0 & b & 0 & 0 \\ 0 & 0 & c & 0 \\ 0 & 0 & 0 & d \end{bmatrix} \text{ with } a, b, c, d > 0$$

now it is possible to weight individually the states.

Sontang's results with the weighted norm, γ

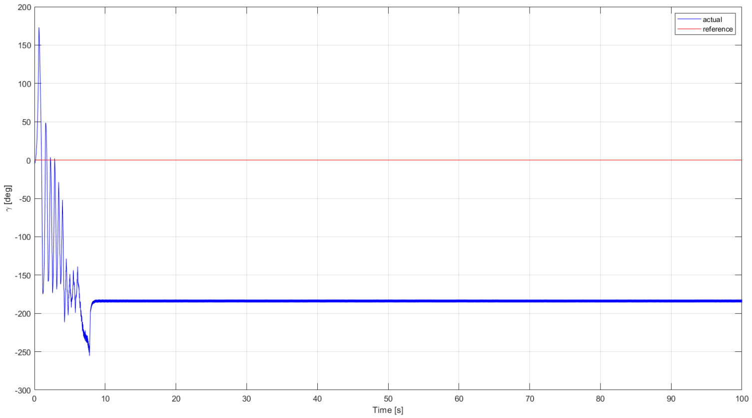


Figure: System controlled using Sontag's Universal Formula, with $a = 10$, $b = c = d = 1$

Sontang's results with the weighted norm, θ

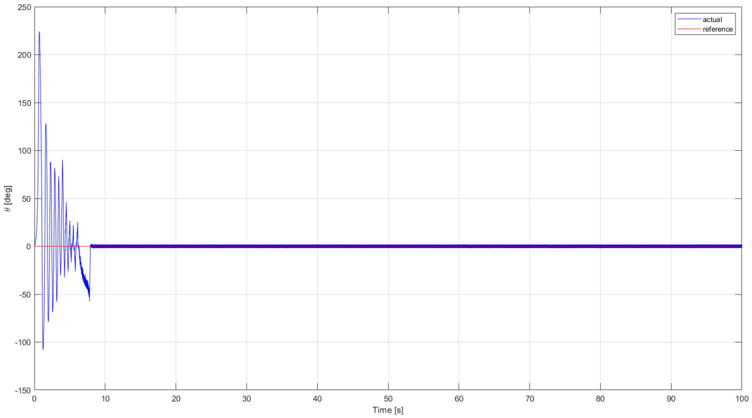


Figure: System controlled using Sontag's Universal Formula, with $a = 10$, $b = c = d = 1$

Second control law with the weighted norm, γ

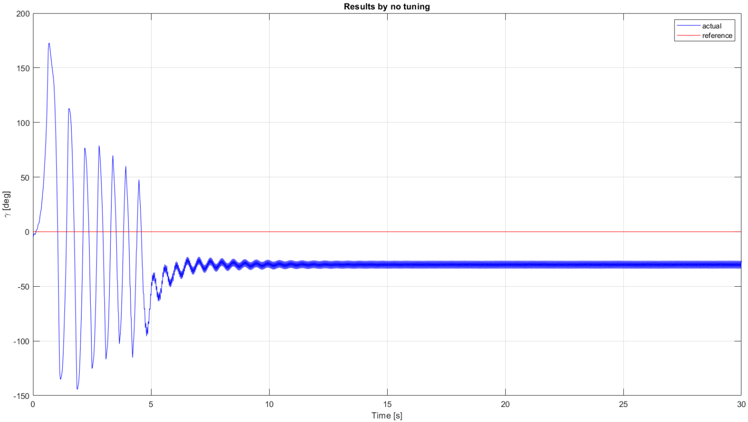


Figure: System controlled using the second control law, with $K_{gain} = 100$, $a = 10$, $b = c = d = 1$

Second control law with the weighted norm, θ

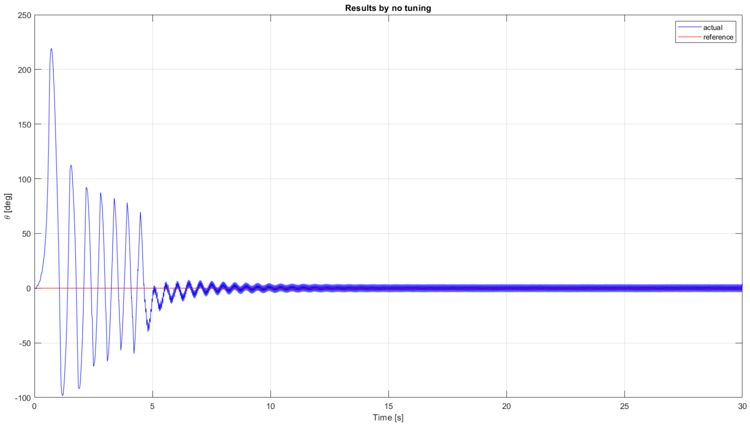


Figure: System controlled using the second control law, with $K_{gain} = 100$, $a = 10$, $b = c = d = 1$

Fast Switching I

- Main idea: use saturated input to stabilize only θ
- Sliding surface: $s = \dot{\theta} + K\theta$
- $s \to 0$ implies $\dot{\theta} = -K\theta$ guarantees convergence of $\theta \to 0$

Control action

$$u = 255\ sign(s)$$

Fast Switching II

PRO

- Simple to implement;
- Correctly stabilizes θ;

CONS

- Does not take into account γ dynamics;
- May not be robust to disturbances;
- Suffers from chattering;
- May not be suited for real-world use;

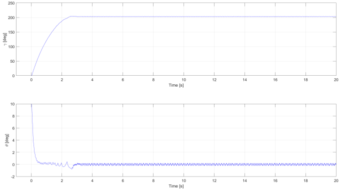


Figure: Results considering $K = 10$ on real state

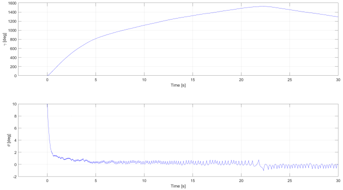


Figure: Results considering $K = 5$ on real state

Adding Saturation Function I

We can improve the previous design by using saturation function instead of sign function used previously.

$$sat(s) = \begin{cases} sign(s) & \text{if } |s| > a \\ \frac{s}{a} & \text{otherwise} \end{cases}$$

Where a can be used as a tuning parameter and to guarantee the neighborhood of convergence;

Control action

$$u = 255\ sat(s)$$

Università degli Studi di Padova

PRO

- Simple to implement;
- Correctly stabilizes θ;
- Reduces chattering;

CONS

- Does not take into account γ dynamics;
- May not be robust to disturbances;
- Does not work with observers;

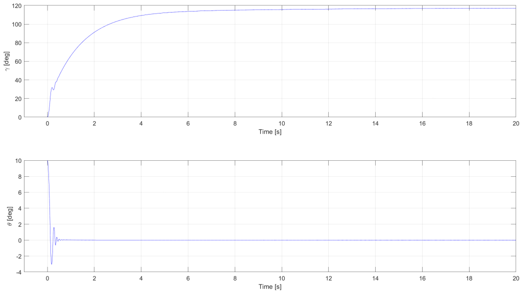
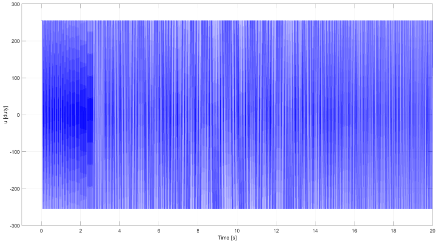
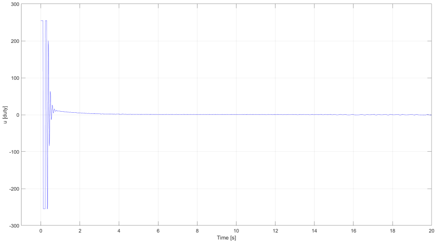


Figure: Results considering $K = 200$ and $a = 1$ on real state

Confronting the control inputs



(a) Control Action considering sign function



(b) Control Action considering saturation function

IS Feedback Linearization

Università degli Studi di Padova

- $\mathcal{G}(x) = \left[g(x), ad_f g(x), ad_f^2 g(x), ad_f^3 g(x)\right]$ is not full-rank everywhere on $\mathcal{X}$, moreover we cannot define a suitable domain $\mathcal{D}$ into which a diffeomorphism exist;
- We tried an approximate I-S Feedback linearization as described in [1], where we try to find an output function that gives a robust relative degree by solving the following equation:

$$\left[\frac{\partial h}{\partial x_1}, \frac{\partial h}{\partial x_2}, 0, 0\right] \left[g(x), ad_f g(x), ad_f^2 g(x)\right] = 0_{1\times 3}$$

- The only solution was $\dfrac{\partial h}{\partial x_1} = 0$ and $\dfrac{\partial h}{\partial x_2} = 0$

[1] Asuk Amba J., Feedback linearization, sliding mode and swing up control for the inverted pendulum on a cart, University of Manchester, 2015

IO Feedback Linearization on θ I

We want to consider $h(x) = x_2 = \theta$ as output function to linearize the system;
We recall that we can write the system as:

$$\dot{x} = \begin{bmatrix} x_3 \\ x_4 \\ f_1(x) \\ f_2(x) \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ g_1(x) \\ g_2(x) \end{bmatrix} u \quad \text{with } g'(x) = \begin{bmatrix} 0 \\ 0 \\ -\dfrac{5.099 + 2.093\cos(x_2)}{0.026\cos^2(x_2) - 0.500\cos(x_2) - 1.008} \\ \dfrac{2.713 + 2.093\cos(x_2)}{0.026\cos^2(x_2) - 0.500\cos(x_2) - 1.008} \end{bmatrix}$$

IO Feedback Linearization on θ II

We derive the input function until the output appears and then design our control action.

$$\dot{h}(x) = \dot{x}_2 = x_4$$
$$\ddot{h}(x) = \dot{x}_4 = f_2(x) + g_2(x)u$$

Control action

$$u = \frac{v_1 - f_2(x)}{g_2(x)} \quad \text{with} \quad v_1 = -k_1 x_2 - k_2 x_4$$

IO Feedback Linearization on θ III

PRO

- Simple to implement;
- Correctly stabilizes θ;

CONS

- Does not take into account γ dynamics;
- May not be robust to disturbances or too imprecise knowledge of the system;

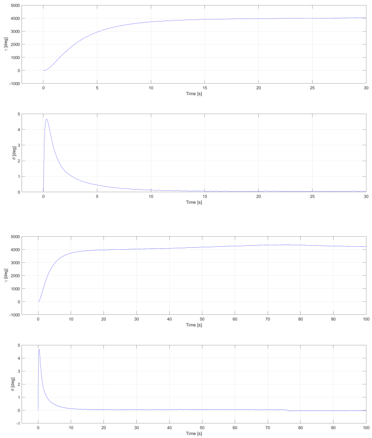


Figure: Results considering $K_1 = 100.25$ and $K_2 = 20$ on estimated state

Adding Robustness I

We want to find a control input that acts on top of the one previously developed to stabilize also the dynamics of γ.
Start by considering the effect of the following control action: $u' = \frac{v_1 - f_2(x)}{g_2(x)} + v_2$ so the dynamics becomes:

$$
\begin{aligned}
\dot{x}_1 &= x_3 \\
\dot{x}_2 &= x_4 \\
\dot{x}_3 &= f_1(x) + g_1(x)\left(\frac{v_1 - f_2(x)}{g_2(x)} + v_2\right) \\
\dot{x}_4 &= v_1 + g_2(x)v_2
\end{aligned}
$$

Find v_2 via Lyapunov based control

Adding Robustness II

Add a tracking requirement for γ and so a desired reference to track $x_{1,d}$, so define the tracking error $e_1 = x_1 - x_{1,d}$
Define the Lyapunov function: $V(x) = \frac{1}{2}c_1 e_1^2 + \frac{1}{2}c_2 x_3^2$
Impose a desired convergence rate $\beta(x) = c_3 x_3^2$ and impose through v_3 that $\dot{V}(x) = -\beta(x)$;

$$\dot{V}(x) = c_1 e_1 \dot{e}_1 + c_2 x_3 \dot{x}_3 = c_1 e_1 x_3 + c_2 x_3 \dot{x}_3$$

Substituting the dynamics and doing some calculations we obtain:

$$v_2 = \frac{\left(-c_1 e_1 - c_3 x_3 - c_2 f_1(x)\right) g_2(x) - c_2 g_1(x) \left(v_1 - f_2(x)\right)}{c_2 g_1(x) g_2(x)}$$

Adding Robustness III

Such v_2 will cancel the effect of v_1;
As shown in [1] also taking $-v_2$ stabilizes the whole system while keeping v_1

Control action

$$u = \frac{v_1 - f_2(x)}{g_2(x)} - \frac{(-c_1 e_1 - c_3 x_3 - c_2 f_1(x))\, g_2(x) - c_2 g_1(x)\,(v_1 - f_2(x))}{c_2 g_1(x) g_2(x)}$$

Adding Robustness IV

PRO

- Correctly stabilizes θ;
- Correctly tracks γ;
- Works well also on the real robot;

CONS

- May not be robust to disturbances or to imprecise knowledge of the system;

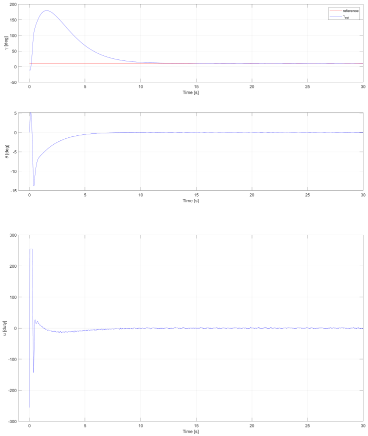


Figure: Results considering $K_1 = 72$, $K_2 = 17$, $c_1 = 10$, $c_2 = 1$ and $c_3 = 15$ on estimated state

Adding Robustness V

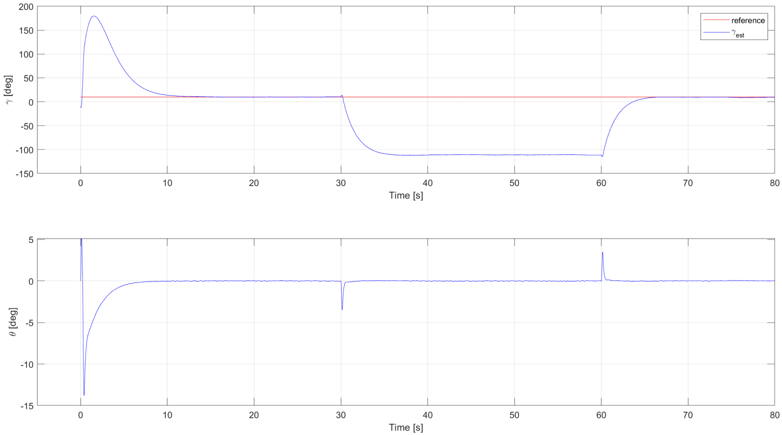


Figure: Results with Disturbances

Results on Real Robot I

Università degli Studi di Padova

We can see the behaviour of the system with external disturbances

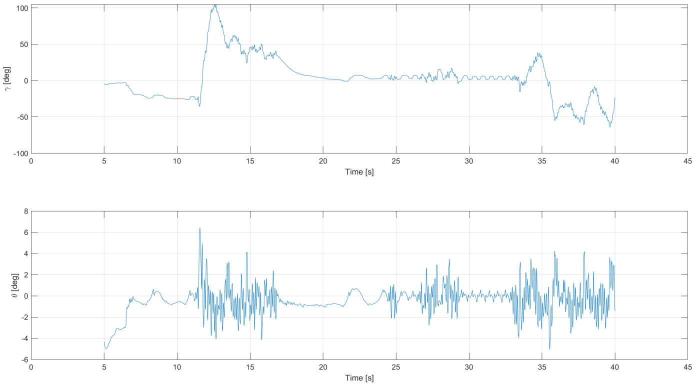


Figure: Results considering $K_1 = 100.25,\ K_2 = 20,\ c_1 = 10,\ c_2 = 5$ and $c_3 = 15$

Results on Real Robot II

We can see the stabilization of θ and the behaviour of γ in steady state

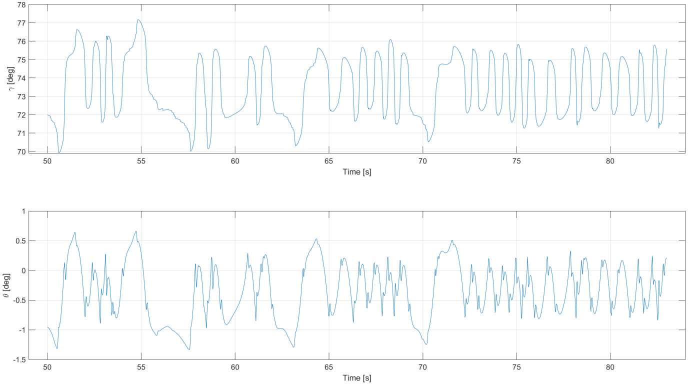


Figure: Results considering $K_1 = 100.25$, $K_2 = 20$, $c_1 = 10$, $c_2 = 5$ and $c_3 = 15$