Adaptive and filtered visual servoing of planar robots

Nolcos 1998, July 2

Erjen Lefeber

Joint work with R. Ortega, R. Kelly, H. Nijmeijer

Department of Systems Signals and Control,

Faculty of Mathematical Sciences,

University of Twente, Enschede

{A.A.J.Lefeber,H.Nijmeijer}@math.utwente.nl

outline

- Introduction
- Case of known θ (orientation of camera).
- Adaptive controller for unknown θ
- A simulation
- A second attempt
- Simulation
- Concluding remarks

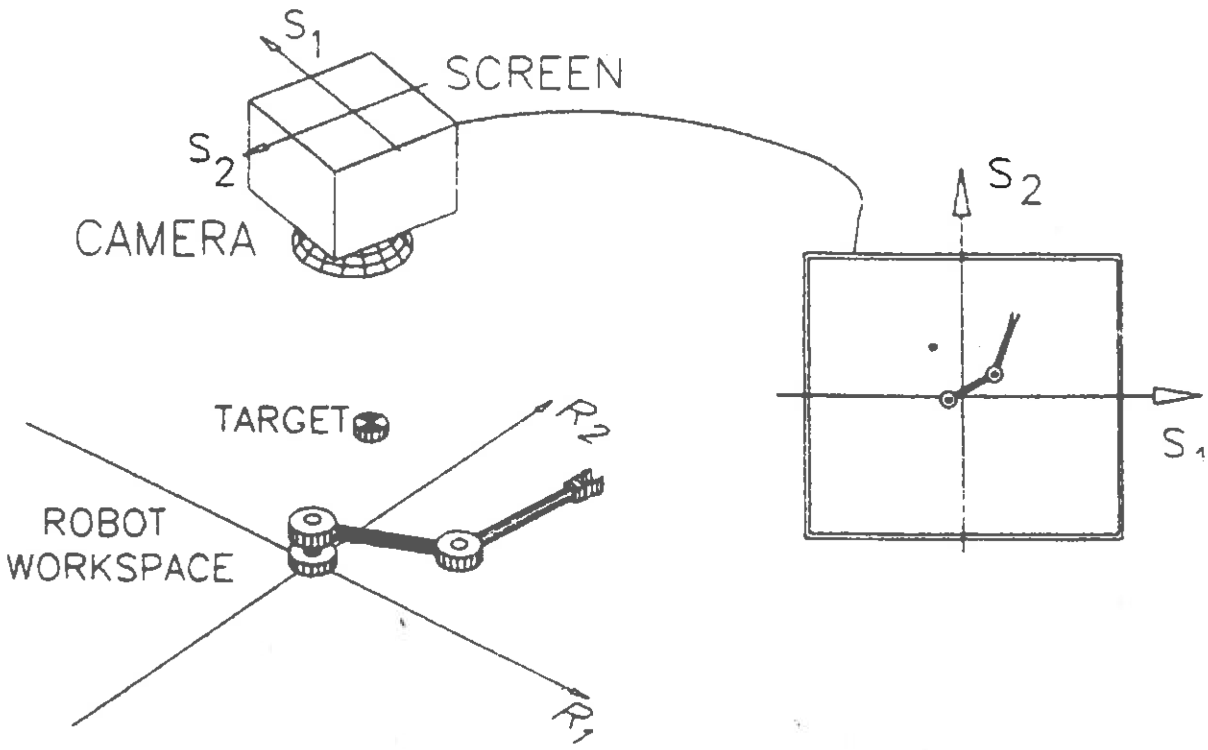


Fig. 1. Fixed-camera robotic system.

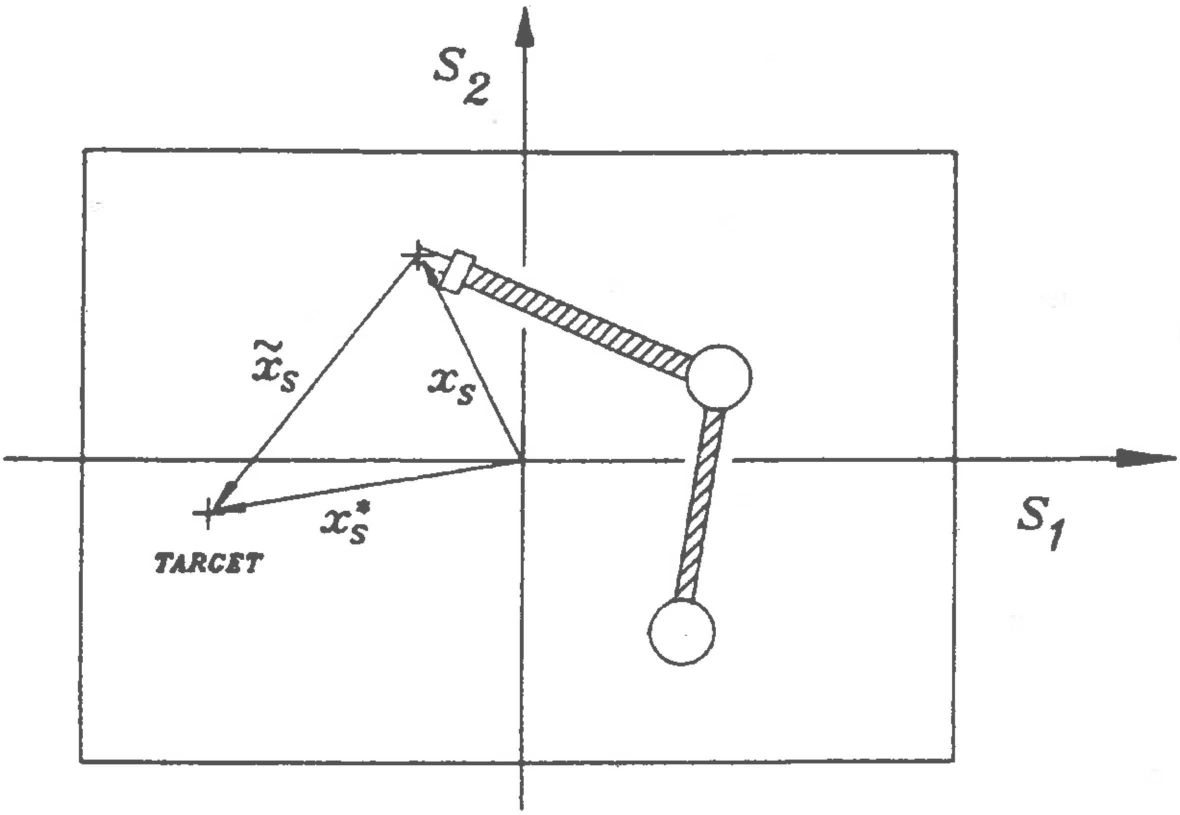


Fig. 3. Definition of image position error in screen coordinate.

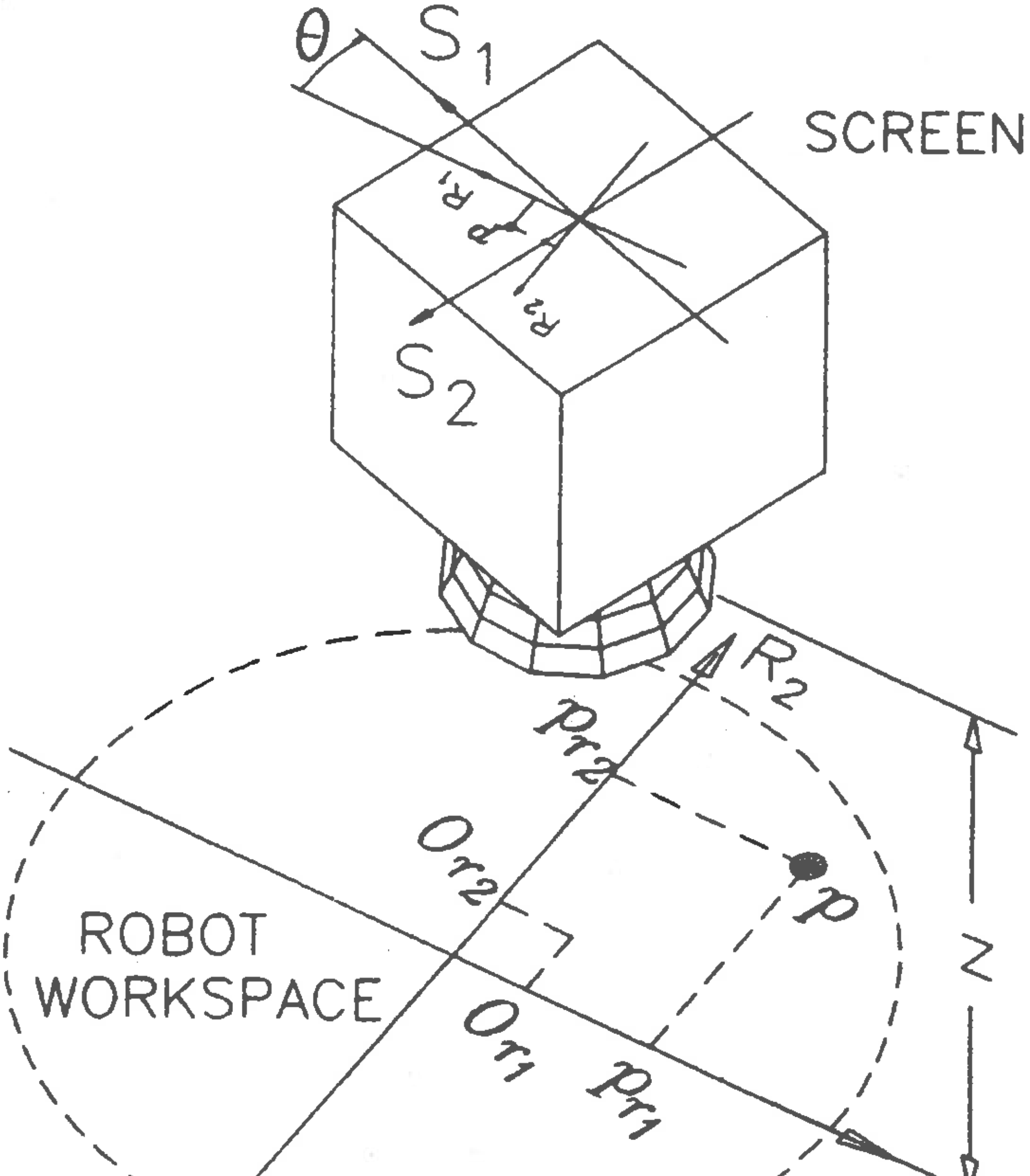


Fig. 2. Coordinate frames

Rigid robot manipulator:

$$M(q)\ddot{q} + C(q,\dot{q})\dot{q} + G(q) = \tau$$

Output (in pixel coordinates)

$$y = ae^{-J\theta}[k(q) - \vartheta_1] + \vartheta_2$$

where $k(q)$ are the direct kinematics and

$$J = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} \quad \text{and therefore} \quad e^{-J\theta} = \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix}$$

We also use the Jacobian $\mathcal{J}(q) = \frac{\partial k(q)}{\partial q}$.

Control Problem

System

$$M(q)\ddot{q} + C(q,\dot{q})\dot{q} + G(q) = \tau$$
$$y = ae^{-J\theta}[k(q) - \vartheta_1] + \vartheta_2$$

Assume q, $\dot{q}$ and y available for measurement.

a, ϑ_1, ϑ_2, θ unknown.

Find controller that yields $y \to y_d$ (fixed point).

Assumptions

Problem solvability Exists (unknown) q_d such that

$$y_d = ae^{-J\theta}[k(q_d) - \vartheta_1] + \vartheta_2$$

Nonsingularity at desired configuration For unknown q_d:

$$\det \mathcal{J}(q_d) \neq 0$$

Solution for known θ

Proposed controller

$$\tau = G(q) - K_d\dot{q} - \mathcal{J}(q)^T e^{J\theta} K_p \tilde{y}$$

where $\tilde{y} = y - y_d$, $K_p = K_p^T > 0$, $K_d = K_d^T > 0$.

Remark: $K_d\dot{q} \Leftrightarrow f_1(\dot{q})$ and $K_p\tilde{y} \Leftrightarrow f_2(\tilde{y})$

where $f(x) = Kx$, $f(x) = K\,\mathrm{Tanh}(Lx)$, etc.

Closed-loop system

$$
\begin{gathered}
M(q)\ddot{q} + C(q,\dot{q})\dot{q} + K_d\dot{q} + \mathcal{J}(q)^T e^{J\theta} K_p \tilde{y} = 0 \\
\tilde{y} = a e^{-J\theta}[k(q) - k(q_d)]
\end{gathered}
$$

Lyapunov function

$$
V(q,\dot{q}) = \frac{1}{2}\dot{q}^T M(q)\dot{q} + \frac{1}{a}\tilde{y}^T K_p \tilde{y}
$$

Differentiating along closed-loop:

$$
\begin{aligned}
\dot{V} &= -\dot{q}^T K_d \dot{q} - \dot{q}^T \mathcal{J}(q)^T e^{J\theta} K_p \tilde{y} + \dot{q}^T \mathcal{J}(q)^T e^{J\theta} K_p \tilde{y} \\
&= -\dot{q}^T K_d \dot{q}
\end{aligned}
$$

La Salle: $\tilde{y} \to 0$ (local, $\mathcal{J}(q)$ nonsingular).

Unknown θ: a first attempt

Adaptive controller might look like

$$\tau = G(q) - K_d \dot{q} - \mathcal{J}(q)^T e^{J\hat{\theta}} K_p \tilde{y}$$

New Lyapunov function

$$V(q, \dot{q}, \tilde{\theta}) = \frac{1}{2} \dot{q}^T M(q) \dot{q} + \frac{1}{a} \tilde{y}^T K_p \tilde{y} + \frac{1}{\gamma}(1 - \cos \tilde{\theta})$$

where $\tilde{\theta} = \hat{\theta} - \theta$.

Question: Why $(1 - \cos \tilde{\theta})$ instead of $\tilde{\theta}^2$?

$$\dot{V} = -\dot{q}^T K_d \dot{q} - \dot{q}^T \mathcal{J}(q)^T e^{J\hat{\theta}} K_p \tilde{y} + \dot{q}^T \mathcal{J}(q)^T e^{J\theta} K_p \tilde{y} + \frac{1}{\gamma} \sin\tilde{\theta} \cdot \dot{\hat{\theta}}$$

$$\begin{aligned} \dot{q}^T \mathcal{J}(q)^T e^{J\theta} K_p \tilde{y} &= \dot{q}^T \mathcal{J}(q)^T e^{J(\hat{\theta}-\tilde{\theta})} K_p \tilde{y} \\ &= \dot{q}^T \mathcal{J}(q)^T [\cos\tilde{\theta} I - \sin\tilde{\theta} J] e^{J\hat{\theta}} K_p \tilde{y} \end{aligned}$$

Therefore using

$$\dot{\hat{\theta}} = \gamma \dot{q}^T \mathcal{J}(q)^T J e^{J\hat{\theta}} K_p \tilde{y}$$

Results into

$$\dot{V} = -\dot{q}^T K_d \dot{q} - \dot{q}^T \mathcal{J}(q)^T e^{J\hat{\theta}} K_p \tilde{y} + \cos\tilde{\theta} \dot{q}^T \mathcal{J}(q)^T e^{J\hat{\theta}} K_p \tilde{y}$$

To 'copy' proof we are more glad when we have

$$\dot{V} = -\dot{q}^T K_d \dot{q} - \left| \dot{q}^T \mathcal{J}(q)^T e^{J\hat{\theta}} K_p \tilde{y} \right| + \cos\tilde{\theta} \dot{q}^T \mathcal{J}(q)^T e^{J\hat{\theta}} K_p \tilde{y}$$

Resulting adaptive controller

$$\tau = \begin{cases} G(q) - K_d\dot{q} - \mathcal{J}(q)^T e^{J\hat{\theta}} K_p \tilde{y} & \text{if} \quad \dot{q}^T \mathcal{J}(q)^T e^{J\hat{\theta}} K_p \tilde{y} \leq 0 \\ G(q) - K_d\dot{q} + \mathcal{J}(q)^T e^{J\hat{\theta}} K_p \tilde{y} & \text{if} \quad \dot{q}^T \mathcal{J}(q)^T e^{J\hat{\theta}} K_p \tilde{y} > 0 \end{cases}$$

$$\dot{\hat{\theta}} = \gamma \dot{q}^T \mathcal{J}(q)^T J e^{J\hat{\theta}} K_p \tilde{y}$$

Simulations

System under consideration:

$$M(q)\ddot{q} + C(q,\dot{q})\dot{q} + G(q) = \tau$$

Where

$$\begin{aligned}
M(q) &= \begin{bmatrix} 2.3516 + 0.1676\cos q_2 & 0.1019 + 0.0838\cos q_2 \\ 0.1019 + 0.0838\cos q_2 & 0.1019 \end{bmatrix} \\
C(q,\dot{q}) &= 0.0838\sin q_2 \begin{bmatrix} -\dot{q}_2 & -(\dot{q}_1 + \dot{q}_2) \\ \dot{q}_1 & 0 \end{bmatrix} \\
G(q) &= \begin{bmatrix} 38.4640\sin q_1 + 1.8270\sin(q_1 + q_2) \\ 1.8270\sin(q_1 + q_2) \end{bmatrix}
\end{aligned}$$

Output

$$y = ae^{-J\theta}(k(q) - \vartheta_1) + \vartheta_2$$

where

$$a = 1359.0, \quad \vartheta_1 = \begin{bmatrix} 0.729 \\ -0.481 \end{bmatrix}, \quad \vartheta_2 = \begin{bmatrix} 320 \\ 240 \end{bmatrix}, \quad \theta = \frac{\pi}{9}$$

$$k(q) = \begin{bmatrix} l_1 \sin q_1 + l_2 \sin(q_1 + q_2) \\ -l_1 \cos q_1 - l_2 \cos(q_1 + q_2) \end{bmatrix}, \quad l_1 = 0.45, \quad l_2 = 0.55.$$

We use the controller

$$\tau = G(q) - K_d \,\mathrm{Tanh}(L_d \dot{q}) \pm \mathcal{J}(q)^T e^{J\hat{\theta}} K_p \,\mathrm{Tanh}(L_p \tilde{y})$$

Where

$$\mathcal{J}(q) = \begin{bmatrix} l_1 \cos q_1 + l_2 \cos(q_1 + q_2) & l_2 \cos(q_1 + q_2) \\ l_1 \sin q_1 + l_2 \sin(q_1 + q_2) & l_2 \sin(q_1 + q_2) \end{bmatrix}$$

$$\mathrm{Tanh}(\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}) = \begin{bmatrix} \tanh x_1 \\ \tanh x_2 \end{bmatrix}, \quad K_p = \begin{bmatrix} 7 & 0 \\ 0 & 7 \end{bmatrix},$$

$$K_d = \begin{bmatrix} 99 & 0 \\ 0 & 7.7 \end{bmatrix}, \quad L_p = \begin{bmatrix} 0.3 & 0 \\ 0 & 0.3 \end{bmatrix}, \quad L_d = \begin{bmatrix} 0.3 & 0 \\ 0 & 0.2 \end{bmatrix}.$$

Our paramater update-law is

$$\dot{\hat{\theta}} = \gamma \dot{q}^T \mathcal{J}(q)^T J e^{J\hat{\theta}} K_p \operatorname{Tanh}(L_p \tilde{y})$$

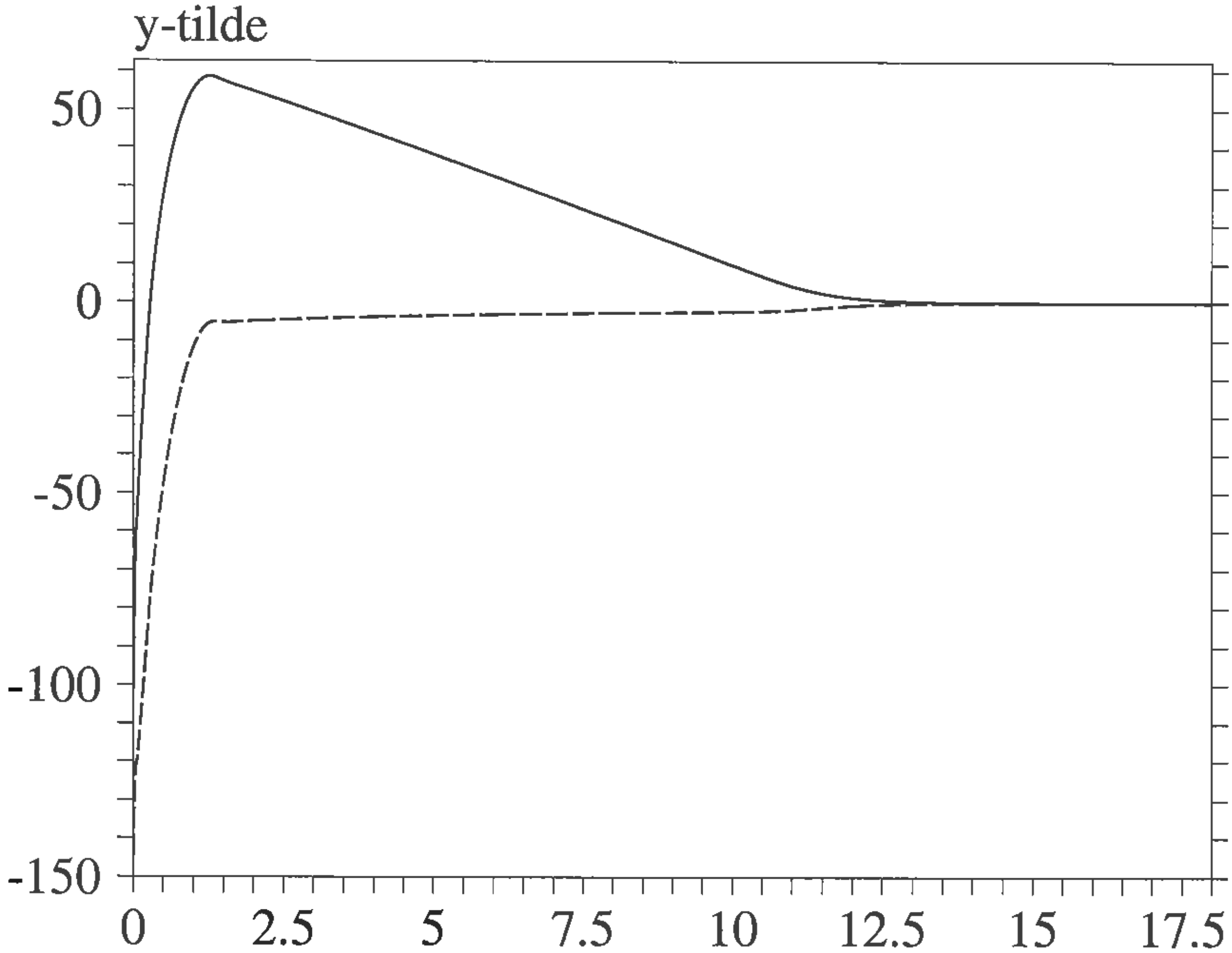
with $\gamma = 1$.

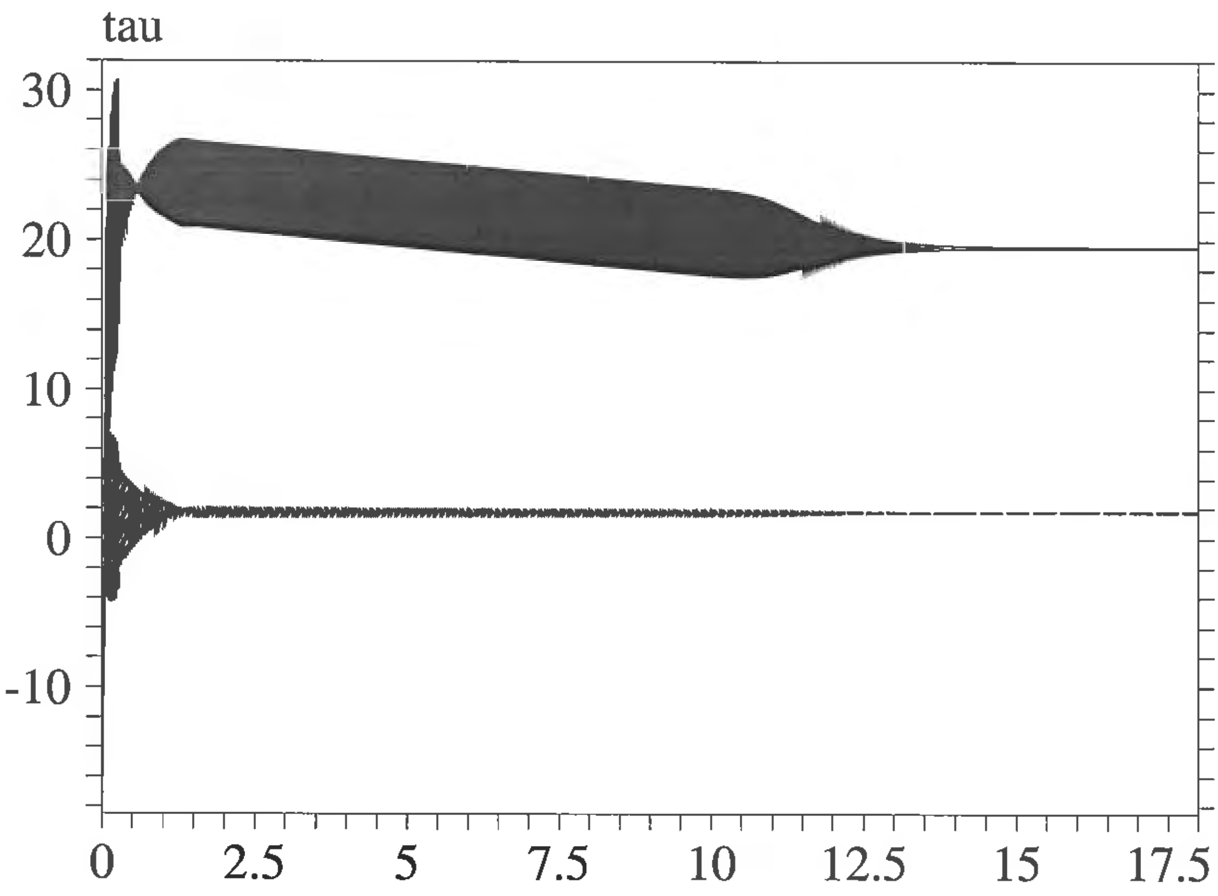
We start from the initial conditions

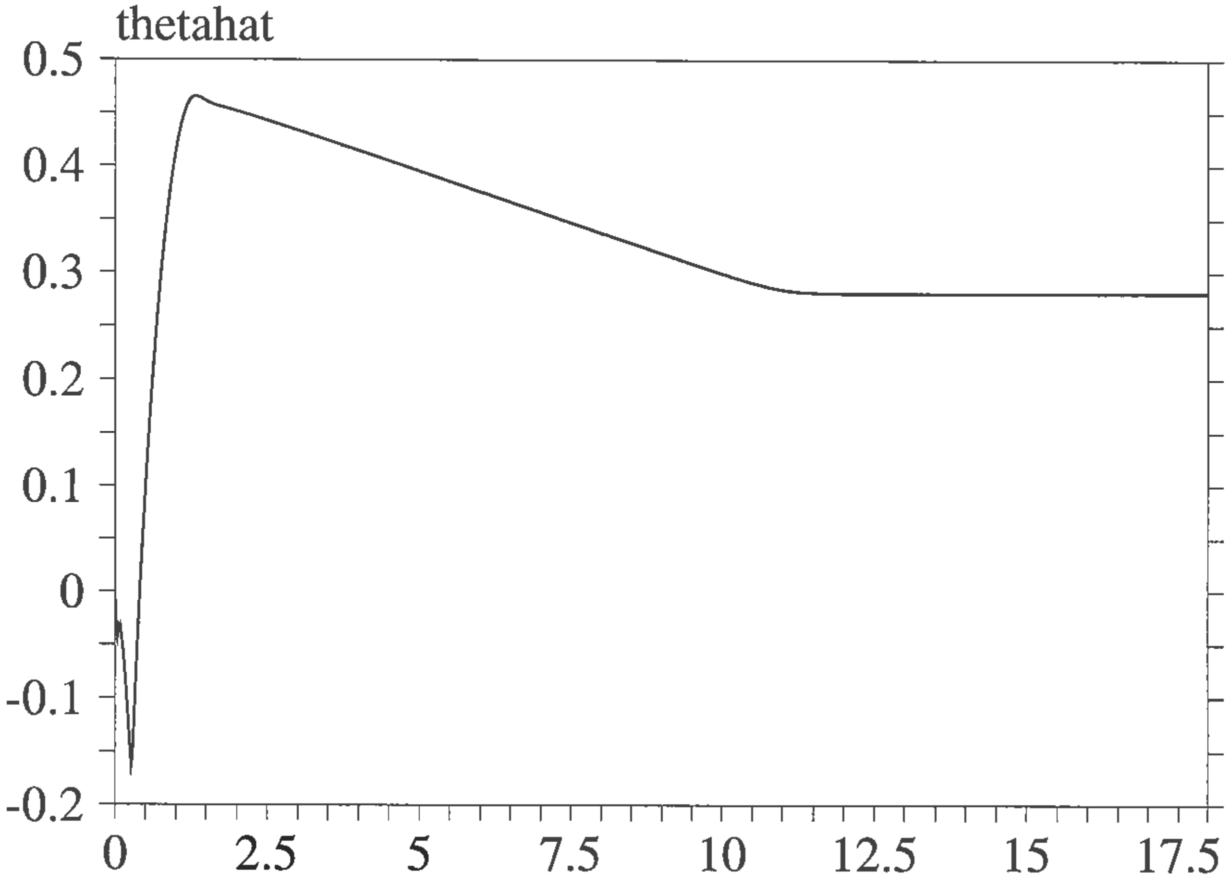
$$q(0) = \begin{bmatrix} 30 \\ 32 \end{bmatrix} \frac{\pi}{180}, \quad \dot{q}(0) = \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \quad \hat{\theta}(0) = 0$$

We want the output y to be steered to

$$y_d = \begin{bmatrix} 320 \\ 180 \end{bmatrix}$$


y-tilde
50
0
-50
-100
-150
0
2.5
5
7.5
10
12.5
15
17.5


tau
30
20
10
0
-10
0
2.5
5
7.5
10
12.5
15
17.5


thetahat
0.5
0.4
0.3
0.2
0.1
0
-0.1
-0.2
0
2.5
5
7.5
10
12.5
15
17.5

An other try

Problem: θ does not appear linear.

However, $\sin\theta$ and $\cos\theta$ do

So, try to estimate $p_1 = \sin\theta$ and $p_2 = \cos\theta$ using $\hat{p}_1$ and $\hat{p}_2$ and 'standard-approach'.

Problem in proof: How to avoid singularity of $\begin{bmatrix} \widehat{\cos\theta} & \widehat{\sin\theta} \\ -\widehat{\sin\theta} & \widehat{\cos\theta} \end{bmatrix}$?

Apply change of coordinates:

$$\hat{\rho} = \ln\left(\widehat{\cos\theta}^2 + \widehat{\sin\theta}^2\right) \quad \hat{\theta} = \arctan\left(\frac{\widehat{\sin\theta}}{\widehat{\cos\theta}}\right)$$

Then we have

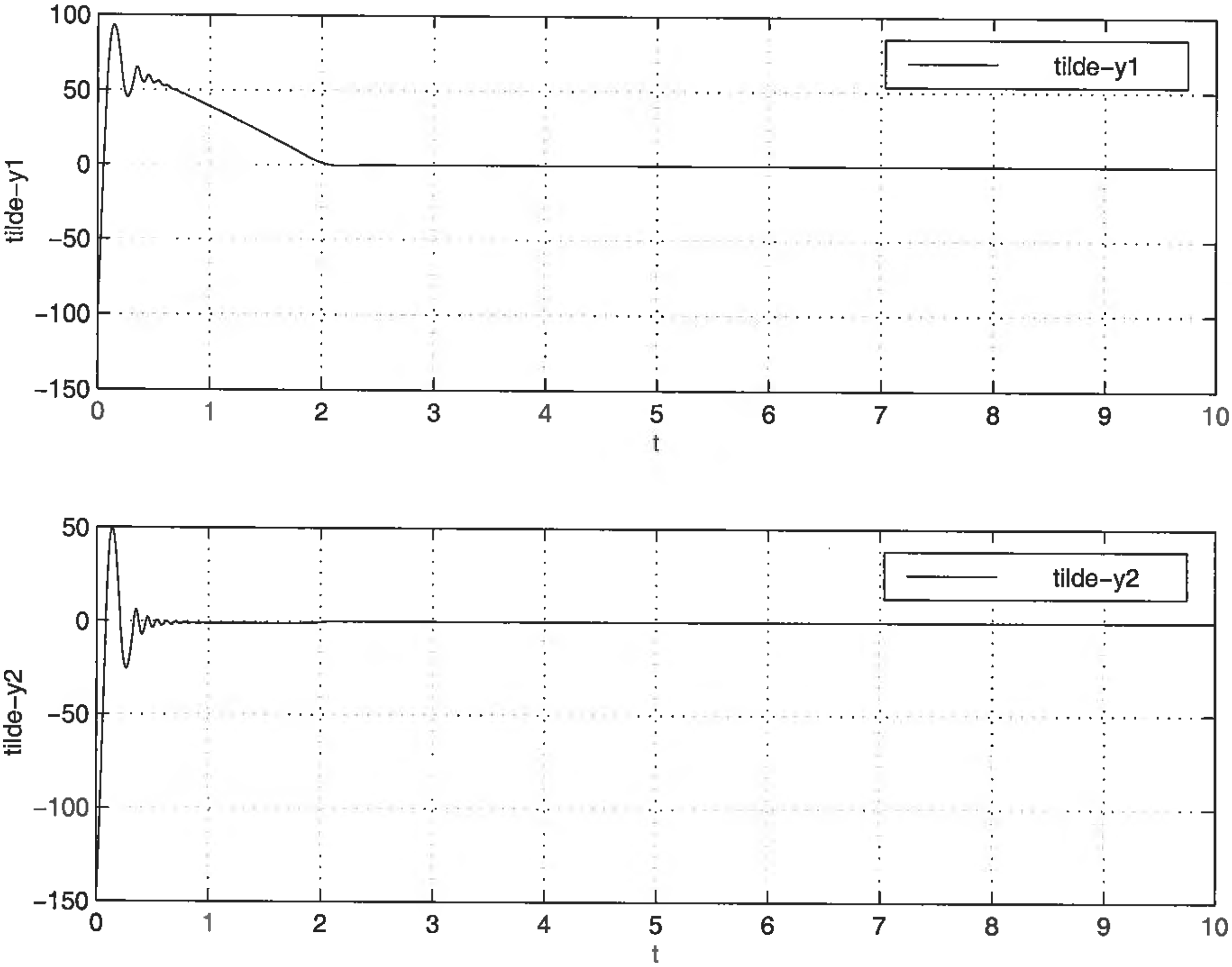
$$\begin{aligned}
\tau &= G(q) - K_d\dot{q} - \mathcal{J}(q)^T e^{\frac{\gamma_1}{\gamma_2}\hat{\rho}} e^{J\hat{\theta}} K_p\tilde{y} \\
\dot{\hat{\theta}} &= \gamma_1 e^{\frac{\gamma_1}{\gamma_2}\hat{\rho}} J e^{J\hat{\theta}} K_p\tilde{y} \\
\dot{\hat{\theta}} &= \gamma_2 e^{\frac{\gamma_1}{\gamma_2}\hat{\rho}} e^{J\hat{\theta}} K_p\tilde{y}
\end{aligned}$$

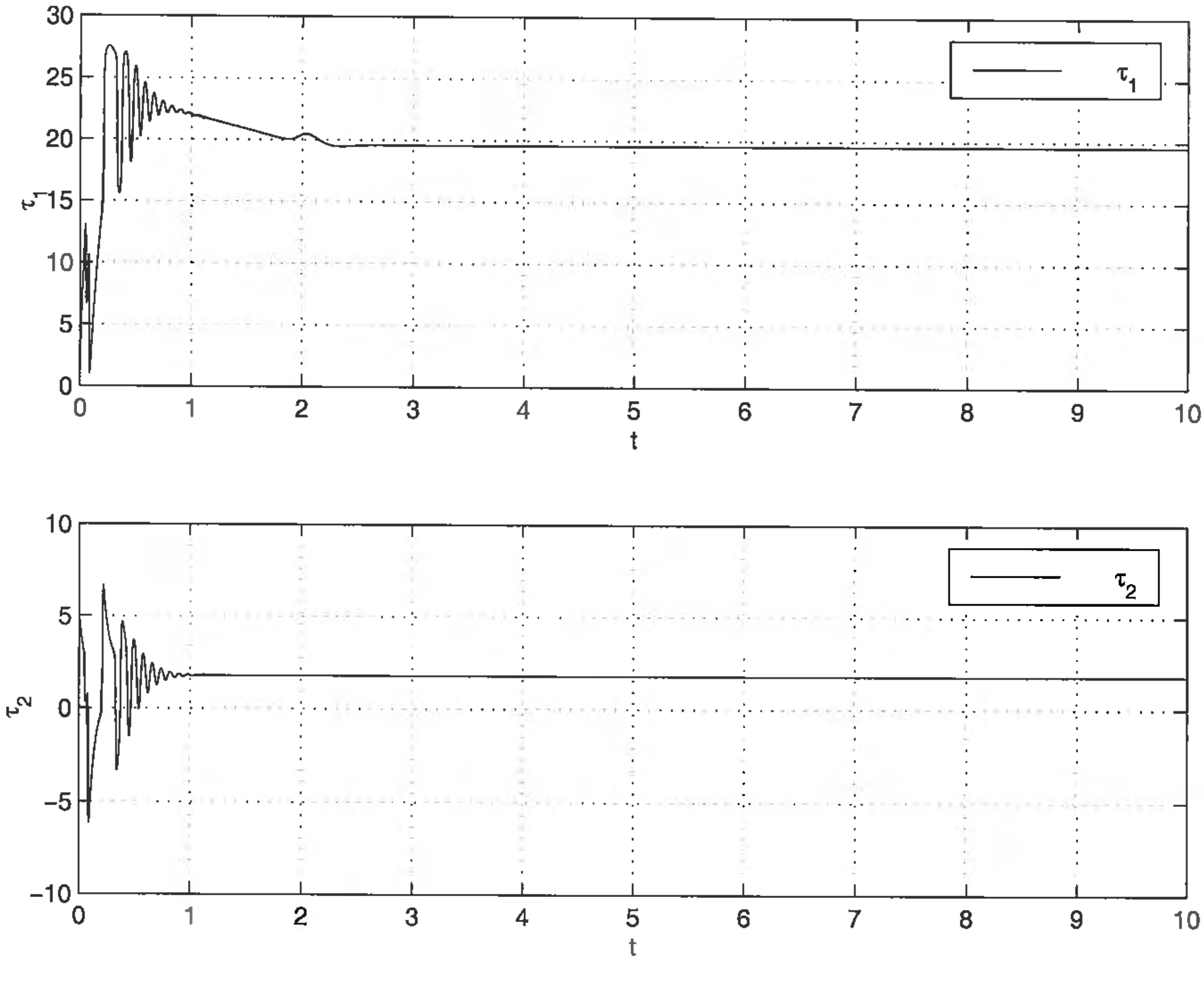
Using

$$V = \frac{1}{2}\dot{q}^T M(q)\dot{q} + \frac{1}{a}\tilde{y}^T K_p\tilde{y} + \frac{1}{2\gamma_1}\left(e^{\frac{\gamma_1}{\gamma_2}\hat{\rho}} - 1\right)^2 + \frac{1}{\gamma_1} e^{\frac{\gamma_1}{\gamma_2}\hat{\rho}}(1 - \cos\tilde{\theta})$$

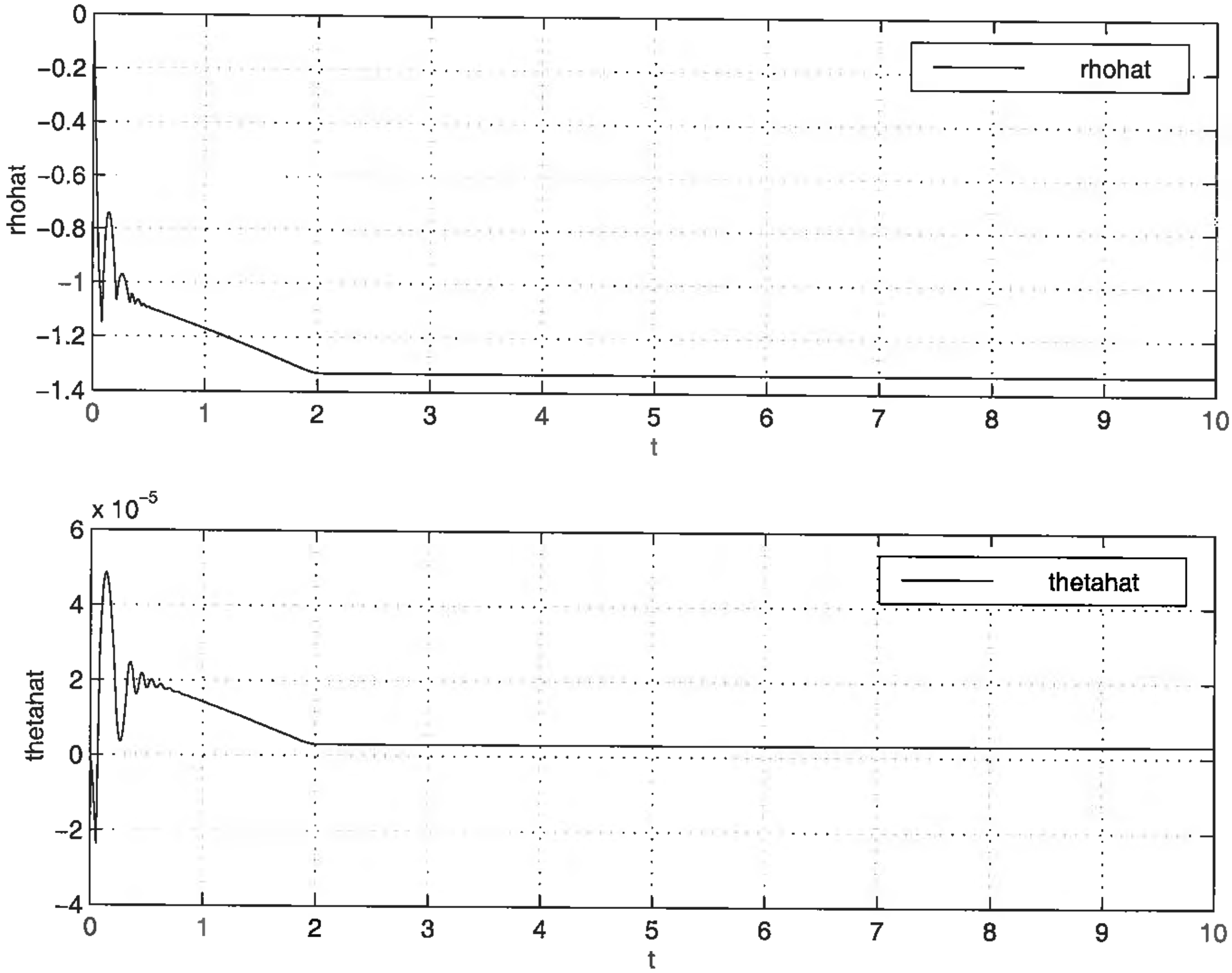
Results into

$$\dot{V} = -\dot{q}^T K_d\dot{q}$$


tilde-y1
tilde-y1
100
50
0
-50
-100
-150
0
1
2
3
4
5
6
7
8
9
10
t
tilde-y2
tilde-y2
50
0
-50
-100
-150
0
1
2
3
4
5
6
7
8
9
10
t



τ_1
τ_1
t
τ_2
τ_2
t



rhohat
0
-0.2
-0.4
-0.6
-0.8
-1
-1.2
-1.4
0 1 2 3 4 5 6 7 8 9 10
t
x 10^-5
thetahat
6
4
2
0
-2
-4
0 1 2 3 4 5 6 7 8 9 10
t

Concluding remarks

- Two different adaptive controllers have been presented for case of unknown θ.
- Is a solution possible without switching/overparameterisation?
- Can ideas be used in problem of n-link manipulator in 3D?
- How to tackle problem of unknown parameters that *not* appear linearly?