Hybrid open-loop closed-loop path-following control

D. von Wissel, Ramine Nikoukhah

INRIA

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

Hybrid Open-Loop Closed-Loop Path-Following Control

Dirk VON WISSEL, Ramine NIKOUKHAH

N° 2173
Janvier 1994

PROGRAMME 5

Traitement du signal,
automatique et
productique

Rapport
de recherche

1994

Hybrid Open-Loop Closed-Loop Path-following Control with Preliminary feedback

Dirk von Wissel and Ramine Nikoukhah

INRIA
ROCQUENCOURT BP 105
78153 LE CHESNAY CEDEX, FRANCE

Abstract

A path following control strategy for nonlinear systems is introduced. This control strategy, which is an extension of the work of Jankowski and Van Brussel [1], presents an alternative to input-output linearization. The advantage of this strategy is that it can be applied to very complex nonlinear systems as opposed to input-output linearization for which the required symbolic computation can be prohibitive. A complete analysis of this controller in the linear case is presented and its application is illustrated through a number of examples (linear an nonlinear).

Commande hybride boucle-ouverte, boucle-fermée avec feedback préliminaire pour le suivi de trajectoires

Résumé

Comme une alternative à la linéarization entrée-sortie qui est souvent prohibitive pour des systèmes complexes Jankowski et Van Brussel [1] ont développè une stratégie de commande pour le suivi des trajectoires pour des systèmes non linéaires. On présente une extension de ce travail avec une analyse complete dans le cas linéaire. Son application est illustré à travers diverse exemples linéaires et non linéaires.

1 Introduction

In this report, we consider the problem of path-following (tracking) for complex nonlinear systems. In tracking control problems, the goal is to design a controllers such that the output of the system tracks a given time dependent reference trajectory. Tracking problems arise in the design of controllers for robot manipulators, mobile robots, air-crafts, etc...

The solution to tracking problems can be found in nonlinear control theory as exact input-output linearization [2]. Using this method, as long as the original system is minimum-phase, it is possible to construct a stabilizing controller which achieves asymptotic tracking for any sufficiently differentiable reference trajectory. The main drawback with this approach is its computational complexity. Even for small nonlinear systems, the symbolic computation required for exact input-output linearization can be prohibitive. In order to overcome this difficulty Jankowski and Van Brussel [1] have proposed a new design methodology, closly related to predictive control. Their approach is in part open-loop and in part closed-loop; they measure the state, compute an open-loop control which is applied on a short time period and start over. They have successfully applied their method to a two-link manipulator with flexible joints. We shall explain in detail this approach in Section 2.2. In the sequel, we shall refer to this controller as hybrid open-loop closed-loop (HOC) control.

It turns out however that HOC control does not work systematically on arbitrary systems and it is one of the objectives of this report to show why. But more importantly, the main objective of this report is to show how this approach can be modified to work on a broader class of systems. In particular, we show that one such modification is the addition of a preliminary feedback to the system before the application of the HOC controller. Let us illustrate the idea through a very simple example. Consider the following system

$$\dot{x} = -\alpha x + u \tag{1.1a}$$
$$x(0) = x^0 , \tag{1.1b}$$

and suppose that we like to find a controller such that $x(t)$ converges to (tracks) $\sin(t)$, i.e., $\sin(t) - x(t) \to 0$ as $t \to \infty$. A simplified HOC controller measures the state x at time t, finds the open-loop control for t to $t + \Delta t$, applies this control and starts again at $t + \Delta t$. In the real implementation of the HOC controller, the solution of (1.1) is computed using some numerical integration method. But here because of simplicity of the system we can obtain a closed-form solution which is

$$u(\tau) = \cos(\tau) - \alpha \sin(\tau) , \ \tau \in]0 , \Delta t] . \tag{1.2}$$

Note that $u(\tau)$ does not depend on x^0 and the HOC controller for this particularly simple example is an open-loop controller. Applying this control to system (1.1) yields

$$x(t) = x^0 e^{-\alpha t} + \sin(t)$$

which does not converge to the desired trajectory as long as $\alpha \leq 0$. In particular if $\alpha = 0$ we have

$$x(t) = x^0 + \sin(t) .$$

Let us now apply a preliminary feedback

$$u = -kx + v$$

before applying the HOC controller. Then clearly as long as $k + \alpha > 0$, the HOC controller will work. In this case the preliminary feedback does the job as long as it stabilizes the system, but in general the construction of an adequate preliminary feedback is not as straightforward. We shall present a detailed analysis in Section 4. The above example, even though it illustrates the idea of preliminary feedback, it does not really show how HOC controllers work, because in this case the HOC control reduces to an open-loop controller. As we shall see later, the reason for this is that the solution of the DAE

$$\begin{aligned} \dot{X} &= -\alpha X + U \\ \sin(\tau) &= X \\ X(0) &= x^0 \end{aligned}$$

is independent of the initial conditions x^0, $X(\tau)$ is discontinuous at 0. In general, the solution of DAE's that come up for the construction of the open-loop control $u(t)$ depends on a part of x. It is exactly this part of x^0 which the HOC controller uses for feedback.

The outline of the report is as follows. In Section 2.1, we review the classical tracking method which is based on exact input-output linearization and the use of the structure algorithm. In Section 2.2, we explain the HOC controller proposed in [1]. In Section 3, we apply both of these methods to an example. We show that the classical approach works fine but that the resulting controller is very complex, on the other hand the HOC controller has less complexity but does not work! In Section 4, we analyze conditions under which HOC controller works by considering the class of linear systems and show how a preliminary feedback can be designed to make it work if the conditions are not fulfilled. Section 5 is devoted to issues concerning actual implementation of the controller and in particular the effects of discretization. Finally, the application of the HOC controller with preliminary feedback is presented in Section 6 on the example of Section 3. We show in particular that, with preliminary feedback, HOC does work.

2 Problem statement and existing solutions

Given a system of differential equations

$$\dot{x} = f(x) + \sum_{i=1}^{m} g_i(x) u_i \tag{2.1a}$$

and the output vector function

$$y = h(x) \tag{2.1b}$$

the objective is to find a stabilizing feedback control $u(t, x)$, such that the output y follows a given function of time $\xi(t)$, i.e. that

$$e(t) = y(t) - \xi(t) \tag{2.2}$$

converges to zero as t goes to infinity.

2.1 Exact input-output linearization

In this section, we review two solutions to this problem. The first solution uses exact input-output linearization and is based on the structure algorithm. The second solution is the HOC controller proposed in [1]. To see if there exists a linearizing controller and to design such a controller we can use the structure algorithm as defined in [2], which is a nonlinear extension of an algorithm for linear system inversion introduced by Silverman [3]. For clarity of presentation we first review Silverman's algorithm for linear system inversion and then give its nonlinear extension.

Silverman Structure-Algorithm for linear system inversion Let

$$\dot{x} = A\,x + B\,u \tag{2.3a}$$
$$y = C\,x \tag{2.3b}$$

and suppose that m, the number of outputs y, equals the number of inputs u. The idea of the structure algorithm is to construct u in terms of x, y and its derivatives. This is done by differentiating and applying linear transformations to (2.3b) until u appears with an invertible coefficient.

- **Step 1:** Let $y_0 = y$ and differentiate (2.3b):

  $$\dot{y}_0 = C\,A\,x + C\,B\,u.$$

  Let $r_1 = \text{rank}(CB)$. Then there exists a nonsingular $m \times m$ matrix V_1 row-compressing CB

  $$V_1\,C\,B = \begin{bmatrix} D_1 \\ 0 \end{bmatrix},$$

  where D_1 is $r_1 \times m$ and $\text{rank}(D_1) = r_1$. If $r_1 = m$ the algorithm ends; if $r_1 < m$, define the new outputs

  $$y_{1,1} = C_{1,1}x + D_1 u \tag{2.4}$$
  $$y_{1,2} = C_{1,2}x\,, \tag{2.5}$$

  where $\begin{bmatrix} C_{1,1} \\ C_{1,2} \end{bmatrix} = V_1 C\,A$ and $\begin{bmatrix} y_{1,1} \\ y_{1,2} \end{bmatrix} = V_1 \dot{y}_0$.

- **Step i:** Differentiate (2.5) but not (2.4):

  $$\begin{aligned} y_{i-1,1} &= C_{i-1,1}x + D_{i-1}u \\ \dot{y}_{i-1,2} &= C_{i-1,2}A\,x + C_{i-1,2}B\,u\,. \end{aligned}$$

  Let $r_i = \text{rank}(\begin{bmatrix} D_{i-1} \\ C_{i-1,2}B \end{bmatrix})$. Then there exists a nonsingular $m \times m$ matrix V_i row-compressing $\begin{bmatrix} D_{i-1} \\ C_{i-1,2}B \end{bmatrix}$. We obtain

  $$V_i \begin{bmatrix} D_{i-1} \\ C_{i-1,2}B \end{bmatrix} = \begin{bmatrix} D_i \\ 0 \end{bmatrix},$$

  where D_i is $r_i \times m$ and $\text{rank}(D_i) = r_i$. If $r_i = m$ or $i = n$, the algorithm ends. If $i = n$ but $r_i \neq m$, there is no solution, i.e., system (2.3) is not invertible. If $r_i < m$ and $i \neq n$, define the new outputs.

  $$\begin{aligned} y_{i,1} &= C_{i,1}x + D_i u \\ y_{i,2} &= C_{i,2}x \end{aligned} \tag{2.6}$$

  where $\begin{bmatrix} C_{i,1} \\ C_{i,2} \end{bmatrix} = V_i \begin{bmatrix} C_{i-1,1} \\ C_{i-1,2}A \end{bmatrix}$, $\begin{bmatrix} y_{i,1} \\ y_{i,2} \end{bmatrix} = V_i \begin{bmatrix} y_{i-1,1} \\ \dot{y}_{i-1,2} \end{bmatrix}$ and continue.

If the algorithm ends at step ν, we can solve for the control u as a function of y and its derivatives, and x

$$\begin{aligned} u &= -(D_\nu)^{-1}C_{\nu,1}x + (D_\nu)^{-1}y_{\nu,1} \\ &\stackrel{\text{def}}{=} Kx + \sum_{i=0}^{\nu-1} L_i y^{(i)}\,. \end{aligned} \tag{2.7}$$

To perform tracking now, we can consider replacing y in (2.7) by the reference trajectory $\xi(t)$ and apply the control

$$u = Kx + \sum_{i=1}^{\nu-1} L_i \xi(t)^{(i)} , \tag{2.8}$$

in which case the modes of the resulting closed-loop system (eigenvalues of $A+BK$) include the zero dynamics of the original system (2.3) and a bunch of zeros due to differentiations in the structure algorithm. Nothing can be done about the modes coming from the zero dynamics; if the system is not minimum phase, the closed-loop system will be unstable. The zero modes introduced by the structure algorithm however can be placed anywhere by a slight modification of the algorithm as follows:

- **Step 1*.:** Differentiate (2.3b) and subtract from it (2.3b) pre-multiplied by Υ_1, where Υ_1 is a stable matrix.
$$\dot{y} - \Upsilon_1 y = (CA - \Upsilon_1 C)\, x + CB\, u \,.$$
Continue as in the Step 1.
- **Step i*.:** Differentiate (2.5) and subtract from it (2.5) pre-multiplied by Υ_i, where Υ_i is a stable matrix:
$$\dot{y}_{i-1,2} - \Upsilon_i y_{i-1,2} = (C_{i-1,2}A - \Upsilon_i C)\, x + C_{i-1,2} B\, u \,.$$
Continue as in the Step i.

With this version of the algorithm the applied control is still of the form (2.8), but the closed-loop poles of the system will be at the zero dynamics of (2.3) and the eigenvalues of the Υ_i's. We will refer to this modified version of the structure algorithm as stabilized structure algorithm.

Note that if we let $H(s) = [C(sI - A)^{-1}B]$ be the transfer-function associated with (2.3) the application of the algorithm can be interpreted as a left multiplication of $H(s)$ by a polynomial matrix $\Upsilon(s)$

$$\Upsilon(s)y = \Upsilon(s)[C(sI - A)^{-1}B]\, u = (\overline{C}(sI - A)^{-1}B + \overline{D})\, u \tag{2.9}$$

so as to obtain an invertible $\overline{D}$. The condition for $\overline{D}^{-1}$ to exist is that $H(s)$ be invertible. Also note that the algorithm generalizes trivially to the case where

$$y = Cx + Du \,,$$

and to the case where $\dot{x} = Ax + Bu + Jw(t)$ where $w(t)$ is a known perturbation to be rejected.

Nonlinear Extension [2] Consider the nonlinear system

$$\begin{aligned} \dot{x} &= f(x) + G(x)u && (2.10a)\\ y &= h(x) && (2.10b)\\ x(0) &= x^0 && (2.10c) \end{aligned}$$

where $G(x) = [g_1(x), \ldots, g_m(x)]$, and $f(x)$, $g_i(x)$ and $h(x)$ are analytic vector fields. As in the linear case we assume that the number m of inputs equals the number of outputs. For clarity of presentation the differentiation of the output is written in the Lie derivative notation. Let

$$L_f h(x) = \frac{\partial h(x)}{\partial x} f(x) \,,$$

where $\frac{\partial h(x)}{\partial x}$ is the Jacobian-matrix of $h(x)$.

- **Step 1:** We proceed like in the linear case. Let $y_0 = y$ and differentiate the output (2.10b) !

$$\begin{aligned}\dot{y} &= L_f h(x) + \sum_{j=1}^{m} L_{g_j} h(x) u_j \\ &= L_f h(x) + T_1(x) u ,\end{aligned}$$

where

$$T_1(x) = [\ L_{g_1} h(x) \ \mid \ L_{g_2} h(x) \ \mid \ \dots \ \mid \ L_{g_m} h(x)\ ] .$$

Clearly in the linear case $T_1(x)$ would be CB. If rank of $T_1(x)$ is fixed in a neighborhood of x_0 and equals the dimension of the vector field spanned by columns of $T_i(x)$ over the field of real numbers denoted $\mathrm{rank}_{\mathbf{r}}(T_i(x))$, we can proceed. Otherwise the algorithm fails. If $r_1 = \mathrm{rank}_{\mathbf{r}}(T_1(x)) = \mathrm{rank}(T_1(x^0))$ then there exists a nonsingular matrix V_1 row-compressing $T_1(x)$.

$$V_1 T_1(x) = \begin{bmatrix} S_1(x) \\ 0 \end{bmatrix} .$$

If $r_1 < m$ define the new output

$$\begin{aligned} y_{1,1} &= h_{1,1}(x) + S_1(x) \\ y_{1,2} &= h_{1,2}(x) \end{aligned} \tag{2.11}$$

where $\begin{bmatrix} h_{1,1}(x) \\ h_{1,2}(x) \end{bmatrix} = V_1 L_f h(x)$ and $\begin{bmatrix} y_{1,1} \\ y_{1,2} \end{bmatrix} = V_1 \dot{y}$.

- **Step i:** Differentiate the $y_{i-1,2}$ and consider the new output

$$\begin{aligned} y_{i-1,1} &= h_{i-1,1} + S_{i-1}(x) \\ \dot{y}_{i-1,2} &= L_f h_{i-1,2} + \sum_{j=1}^{m} L_{g_j} h_{i-1,2} u_j \\ &= L_f h_{i-1,2}(x) + T_i(x) u ,\end{aligned}$$

where

$$T_i(x) = [\ L_{g_1} h_{i-1,2}(x) \ \mid \ L_{g_2} h_{i-1,2}(x) \ \mid \ \dots \ \mid \ L_{g_m} h_{i-1,2}(x)\ ] .$$

If rank of $\begin{bmatrix} S_{i-1}(x) \\ T_i(x) \end{bmatrix}$ is fixed in a neighborhood of x_0 and equals the dimension of the vector field spanned by columns of $\begin{bmatrix} S_{i-1}(x) \\ T_i(x) \end{bmatrix}$ over the field of real numbers, we can proceed. Otherwise the algorithm fails. If $\mathrm{rank}_{\mathbf{r}}(\begin{bmatrix} S_{i-1}(x) \\ T_i(x) \end{bmatrix}) = \mathrm{rank}(\begin{bmatrix} S_{i-1}(x^0) \\ T_i(x^0) \end{bmatrix}) = r_i$ then there exists a nonsingular matrix V_i row-compressing $\begin{bmatrix} S_{i-1}(x) \\ T_i(x) \end{bmatrix}$ such that

$$V_i \begin{bmatrix} S_{i-1}(x) \\ T_i(x) \end{bmatrix} = \begin{bmatrix} S_i(x) \\ 0 \end{bmatrix} .$$

If $r_i = m$ or $i = n$ the algorithm ends. For $r_i \neq m$ and $i = n$ there is no solution, i.e., there exists no exact input-output linearizing feedback. If $r_i < m$ and $i \neq n$ define the new output

$$\begin{aligned} y_{i,1} &= h_{i,1}(x) + S_i(x) u \\ y_{i,2} &= h_{i,2}(x) \end{aligned} \tag{2.12}$$

where $\begin{bmatrix} h_{i,1}(x) \\ h_{i,2}(x) \end{bmatrix} = V_i \begin{bmatrix} h_{i-1,1} \\ L_f h_{i-1,2} \end{bmatrix}$, $\begin{bmatrix} y_{1,1} \\ y_{1,2} \end{bmatrix} = V_i \begin{bmatrix} y_{i-1,1} \\ \dot{y}_{i-1,2} \end{bmatrix}$ and continue.

If the algorithm gives a solution at step ν we can solve for the control u as a function of $y_{\nu,1}$, and x.

$$\begin{aligned} u &= (S_\nu(x))^{-1}h_{\nu,1}(x) + (S_\nu(x))^{-1}y_{\nu,1} \\ &= K(x) + \sum_{i=0}^{\nu-1} L_i(x)y^{(i)} \end{aligned}$$

Just as in the linear case, we can modify the algorithm to avoid introducing zero modes in the system and finally tracking control is

$$u = K(x) + \sum_{i=0}^{\nu-1} L_i(x)\xi^{(i)}(t) ,$$

where $\xi(t)$ is the reference trajectory. This approach works if (2.10) is minimum phase.

2.2 HOC Controller

For many nonlinear problems the successive differentiation of the output-function results in huge expressions, or worse, becomes impossible, as the size of the expressions exceeds any symbolic calculation power available. In such cases it is desirable to interrupt the structure algorithm before end and if possible replace the remaining steps by something else, such as a numerical method. The HOC controller introduced in [1] does exactly that: first it applies the structure algorithm as much as possible to reduce the "index" [5] of the system and then computes open-loop controls over short intervals based on the value of the measured state vectors.

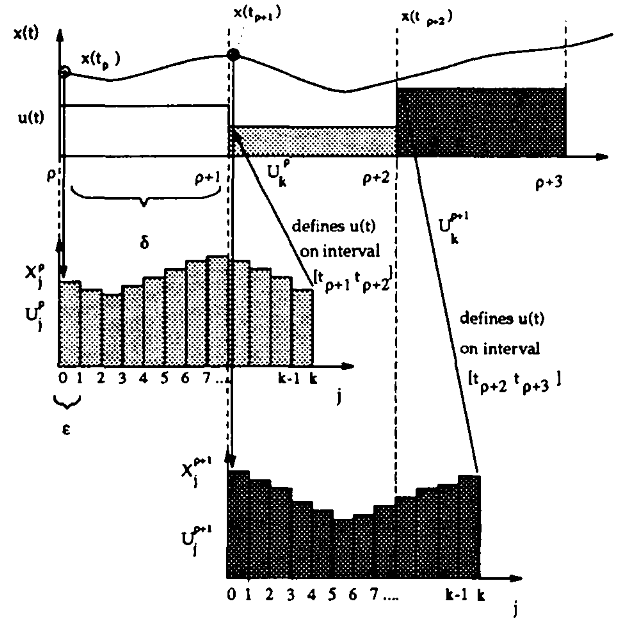

Figure 2.1 Representation of the HOC control strategy proposed in [1]. *$x(t)$ represents the state evolution of the controlled plant over time t. At each instant t_ρ, the plant is sampled and the vector $x(t_\rho)$ is used as initial value X_0^ρ for system (2.14). The plot X_j^ρ, U_j^ρ represents the result of the numerical integration of system (2.14) for $j = 0$ to $j = k$. The control value of u over the interval $t_{\rho+1}$ to $t_{\rho+2}$ is chosen to be U_k^ρ. Clearly, to be implementable, the numerical integration of the DAE has to be faster than real-time.*

The steps in the design of the HOC controller in [1] are

- Application of a number of steps in the stabilized structure algorithm such that the new system

$$\dot{x} = f(x) + G(x)u$$
$$\overline{y} = \widehat{h}(x) + \widehat{J}(x)u \tag{2.13}$$

has low index (less than or equal 3[1])

- Discretization of the DAE

$$\dot{X} = f(X) + G(X)U \tag{2.14a}$$
$$\xi(t) = \widehat{h}(X) + \widehat{J}(X)U \tag{2.14b}$$
$$X(t_\rho) = x(t_\rho) \tag{2.14c}$$

using an implicit discretization scheme (BDF)

$$\dot{X}(t+\epsilon) = \frac{X(t+\epsilon) - X(t)}{\epsilon} ,$$

where ϵ is the discretization time interval. We note $X_0^\rho = x(t_\rho)$ the sampled state vector x at instants $t = t_\rho$, $\rho = 1, 2, \cdots$, where $t_{\rho+1} = t_\rho + \delta$ and $\delta > 0$ is the sampling interval and let X_j^ρ the computed estimation for $x(t_\rho + \epsilon j)$. The resulting discrete approximation of (2.13) is

$$X_{j+1}^\rho - X_j^\rho = \epsilon\{f(X_{j+1}^\rho + G(X_{j+1}^\rho)U_{j+1}^\rho\}$$
$$\widehat{\xi}(t_\rho + j\epsilon) - \widehat{h}(X_{j+1}^\rho) - \widehat{J}(X_{j+1}^\rho)U_{j+1}^\rho = 0 \tag{2.15}$$
$$X_0^\rho = x(t_\rho) .$$

We use uppercase letters for the to be numerically integrated DAE (2.14) as opposed to the original system (2.10).

- Computation of the numerical solution of (2.15) over the time interval $[t_\rho, t_\rho+\tau]$, where$\tau_\rho = k\epsilon$, i.e., from $j = 0$ to $j = k$, where k is an integer to be chosen larger than the number of left out steps in the structure algorithm.

- Application of a piecewise constant control $u(t)$ to (2.1)

$$u(t) = \begin{cases} \vdots & & & & \\ U_k^\rho & \text{for} & t_\rho \le t < t_\rho + \delta & \text{computed with initial value} & X_0^{\rho-1} = x(t_{\rho-1}) \\ U_k^{\rho+1} & \text{for} & t_{\rho+1} \le t < t_{\rho+1} + \delta & \text{computed with initial value} & X_0^\rho = x(t_\rho) \\ \vdots & & & & \end{cases}$$

where $\delta \ge \epsilon$ and ϵ is to be chosen such that $2\delta \ge k\epsilon \ge \delta$.

This procedure is illustrated in Figure 2.1.

Unlike the exact input-output linearization method, the application of of the HOC-controller to a linear or non linear minimum phase system may result in an unstable closed-loop system. This can be seen in the example presented in the next section

[1] This is necessary as we use numerical integration schemes to solve the DAE. These integration schemes may not converge for indexes > 3.

3 Example: Wheel rolling on a plane

Modeling the dynamics of a wheel rolling on a plane requires at least a set of five variables:

- x, y coordinates of center of mass of the wheel
- ϕ, θ, ψ Euler angles associated with the orientation of the wheel
  - ϕ direction
  - θ lean angle
  - ψ roll angle

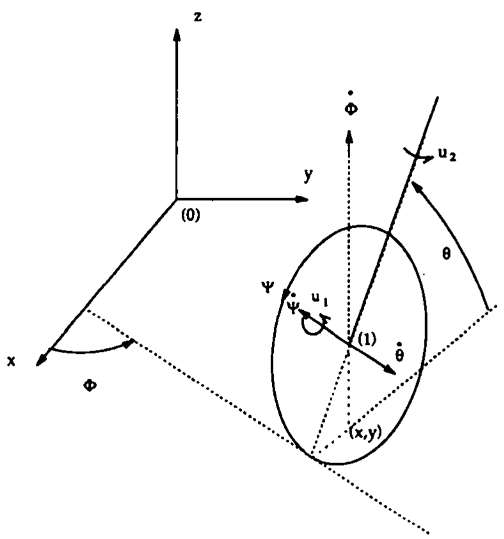

Figure 3.1 *Parametrization of the wheel rolling on a plane.*

As control we introduce two external forces:

- u_1 steering torque. Torque u_1 is such that it is always normal to $\dot{\theta}$ and $\dot{\psi}$.
- u_2 pedalling torque. Torque in the direction of $\dot{\psi}$, normal to the wheel.

Finally we need to define the following constant parameters of the wheel:

- m mass of the wheel $(= 1kg)$
- r radius of the wheel $(= 1m)$
- I_r radial moment of inertia $(= 0.5kgm^2)$
- I_n normal moment of inertia $(= 0.25kgm^2)$

The equations of motion are composed, first, of a set of equations independent of the position $q_2 = [x, y]$ and the absolute orientation ϕ of the wheel on the plane

$$M(\theta)\begin{bmatrix} \ddot{\phi} \\ \ddot{\theta} \\ \ddot{\psi} \end{bmatrix} = F(\theta, \dot{\phi}, \dot{\theta}, \dot{\psi}) + B(\theta)\begin{bmatrix} u_1 \\ u_2 \end{bmatrix} \tag{3.1}$$

or specifically

$$\begin{bmatrix} I_r \sin(\theta)^2 + I_n \cos(\theta)^2 + m\,r^2 \cos(\theta)^2 & 0 & I_n \cos(\theta) + m\,r^2 \cos(\theta) \\ 0 & I_r + m\,r^2 & 0 \\ I_n \cos(\theta) + m\,r^2 \cos(\theta) & 0 & I_n + m\,r^2 \end{bmatrix}\begin{bmatrix} \ddot{\phi} \\ \ddot{\theta} \\ \ddot{\psi} \end{bmatrix} =$$

$$\begin{bmatrix} I_n \dot{\theta}\,\dot{\psi}\,\sin(\theta) + 2\,(m\,r^2 + I_n - I_r)\,\dot{\phi}\,\dot{\theta}\,\cos(\theta)\sin(\theta) \\ -(m\,r^2 + I_n)\,\dot{\phi}\,\dot{\psi}\,\sin(\theta) - (m\,r^2 + I_n - I_r)\,\dot{\phi}^2 \cos(\theta)\sin(\theta) - m\,g\,r\cos(\theta) \\ (2\,m\,r^2 + I_n)\,\dot{\phi}\,\dot{\theta}\,\sin(\theta) \end{bmatrix} + \begin{bmatrix} u_1 \sin(\theta) \\ 0 \\ u_2 \end{bmatrix},$$

(note that $M(\theta)$ is nonsingular for all values of θ except $\theta = i\pi$, $i = .. -1, 0, 1, 2..$) and, second, of two separate equations that can be used to compute the position of the center of mass $\mathbf{q}_2 = [x, y]$ on the plane.

$$\begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix} = \mathbf{Q}(\phi, \theta)\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix}.$$

For a detailed development of the equations of motion see Appendix A.

Tracking objectives As output we take the lean-angle θ and the roll angle of the wheel ψ. The objective is to make θ and ψ follow reference trajectories $\theta_{ref}(t)$ and $\psi_{ref}(t)$.

Explicit solution An application of the structure algorithm shows that we have to differentiate θ three times and ψ twice to obtain explicit expressions for u_1 and u_2 as functions of $\mathbf{q}_1$, $\dot{\mathbf{q}}_1$, $\theta_{ref}(t)$ and $\psi_{ref}(t)$ and their derivatives. The resulting expressions are valid for $0 < \theta < \pi$ and $\dot{\psi} \neq 0$. To begin with, we compute the explicit system representation

$$\dot{x} = f(x) + g(x)u\ ,$$

where $x = \begin{bmatrix} \phi & \theta & \psi & \dot{\phi} & \dot{\theta} & \dot{\psi} \end{bmatrix}$ and

$$f(x) = \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \\ (M(\theta))^{-1} F(\theta, \dot{\phi}, \dot{\theta}, \dot{\psi}) \end{bmatrix},\ g(x) = \begin{bmatrix} 0 \\ 0 \\ 0 \\ (M(\theta))^{-1} B(\theta) \end{bmatrix}.$$

After two differentiations of the output the control variables u_1 and u_2 appear, i.e., for the first step of the structure algorithm, we have $V_1 = I$ and

$$y_{2,1} = h_{2,1}(x) = \begin{bmatrix} \dot{\theta} \\ \dot{\psi} \end{bmatrix}$$

and $y_{1,1}$ is empty.

$$\dot{y}_{2,1} = \begin{bmatrix} \ddot{\theta} \\ \ddot{\psi} \end{bmatrix} = \begin{bmatrix} \begin{array}{l} \frac{\dot{\phi}^2 \sin(\theta)\cos(\theta)}{r^2 m + I_r}(-I_n - r^2 m + I_r) + \\ \quad \frac{1}{r^2 m + I_r}[(-r^2 m - I_n)\sin(\theta)\dot{\psi}\dot{\phi} - m\,g\,r\cos(\theta)] \\ \\ \frac{1}{A_1}\{[\sin(\theta)\cos(\theta)^2(I_n{}^2 + r^2 m\,I_n - I_r\,I_n) - (2\,I_r\,r^2 m + I_r\,I_n)\sin(\theta)]\dot{\phi} - \\ \quad (I_n^2 + I_n\,r^2 m)\cos(\theta)\sin(\theta)\dot{\psi}\dot{\theta} + \\ \quad \sin(\theta)\cos(\theta)(I_n + r^2 m)\,u_1 - [(r^2 m + I_n - I_r)\cos(\theta)^2 + I_r]\,u_2\} \end{array} \end{bmatrix}$$

$$A_1 = -I_r\,\sin(\theta)^2(r^2 m + I_n)$$

However in the matrix $L_g h_{2,1}(x)$ has rank one for all x (verifying $r_2 = \mathrm{rank_r}(L_g h_{2,1}(x)) = \mathrm{rank}(L_g h_{2,1}(x^0)) = 1$). With

$$V_2 = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$$

we compute

$$\begin{aligned}
S_2(x) &= \begin{bmatrix} \sin(\theta)\cos(\theta)(I_n + r^2 m) & (r^2 m + I_n - I_r)\cos(\theta)^2 + I_r \end{bmatrix} \\
h_{2,1}(x) &= \frac{1}{A_1}\{[\sin(\theta)\cos(\theta)^2(I_n{}^2 + r^2 m\,I_n - I_r\,I_n) - (2\,I_r\,r^2 m + I_r\,I_n)\sin(\theta)]\dot{\phi} - \\
&\quad (I_n^2 + I_n\,r^2 m)\cos(\theta)\sin(\theta)\dot{\psi}\dot{\theta} \\
h_{2,2}(x) &= \frac{\dot{\phi}^2\sin(\theta)\cos(\theta)}{r^2 m + I_r}(-I_n - r^2 m + I_r) + \frac{1}{r^2 m + I_r}[(-r^2 m - I_n)\sin(\theta)\dot{\psi}\dot{\phi} - m\,g\,r\cos(\theta)]
\end{aligned}$$

In the third step we differentiate $h_{2,2}(x)$ and obtain

$$\begin{bmatrix} y_{2,1} \\ \dot{y}_{2,2} \end{bmatrix} = \begin{bmatrix} \ddot{\psi} \\ \theta^{(3)} \end{bmatrix} = \begin{bmatrix} h_{2,1}(x) \\ L_f h_{2,2}(x) \end{bmatrix} + \begin{bmatrix} S_2(x) \\ L_g h_{2,2}(x) \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}$$

Now we find that $r_3 = \mathrm{rank_r}(\begin{bmatrix} S_2(x) \\ L_g h_{2,2}(x) \end{bmatrix}) = \mathrm{rank}(\begin{bmatrix} S_2(x^0) \\ L_g h_{2,2}(x^0) \end{bmatrix}) = 2$ hence

$$h_{3,1}(x) =$$

$$\begin{bmatrix} \begin{array}{l} \frac{1}{A_1}[\sin(\theta)\cos(\theta)^2(I_n^2 + r^2 m\,I_n + I_n\,I_r) - (2\,I_r\,r^2 m + I_r\,I_n)\sin(\theta)]\dot{\phi} - \\ \quad (I_n{}^2 + I_n\,r^2\,m)\cos(\theta)\sin(\theta)\,\dot{\psi}\dot{\theta} \\ \\ (-\frac{1}{A_1}(-I_r{}^2 + (3\,r^2\,m\,I_r - I_n\,r^2 m - I_n{}^2 + 3\,I_n\,I_r - I_r{}^2)\cos(\theta)^2 + \\ \quad (2\,I_r{}^2 + I_n^2 - 3\,I_n\,I_r - 2\,I_r\,r^2 m + I_n\,r^2 m)\cos(\theta)^4 - r^2 m\,I_r)\dot{\phi}^2 - \\ \quad \frac{1}{A_1}(2\,(I_n\,r^2 m - 3\,I_n\,I_r - I_r\,r^2 m + 2\,I_n{}^2)\cos(\theta)^3 - \\ \quad (2\,I_n\,r^2 m + 3\,I_n - I_r - 2\,I_n{}^2 + r^2\,m\,I_r)\cos(\theta))\dot{\psi}\dot{\phi} - \\ \quad \frac{1}{A_1}(I_n{}^2 + I_n\,r^2 m - I_n\,r^2 m)(\cos(\theta)^2 - 1)\dot{\psi}^2 - \\ \quad \frac{1}{A_1}(-m\,r\,\sin(\theta)I_r\,\cos(\theta)^2 + m\,r\,\sin(\theta)I_r)\,g\,\dot{\theta} \\ \\ \end{array} \end{bmatrix}$$

further that

$$h_{3,2} = \text{empty}$$

and

$S_3(x) =$

$$\begin{bmatrix} \frac{1}{A_1}\cos(\theta)\sin(\theta)(I_n + r^2 m) & \frac{1}{A_1}\cos(\theta)^2(I_r - r^2 m - I_n) - \frac{I_r}{A_1} \\ -\frac{1}{A_1}(-2 I_r \cos(\theta)^3 + 2 I_r \cos(\theta) + \cos(\theta)^3 I_n - & -\frac{1}{A_1}(\sin(\theta)\cos(\theta)^2(r^2 m + I_n - I_r) - \\ I_n\cos(\theta) - r^2\cos(\theta)m + \cos(\theta)^3 r^2 m)\dot{\phi} - \dot{\psi} & \sin(\theta) I_r)\dot{\phi} - \frac{1}{A_1}(\sin(\theta)\cos(\theta)(r^2 m - I_n)\dot{\psi} \end{bmatrix}$$

As $S_3(x)$ is invertible the controller law for exact system-inversion is

$$u = S^{-1}(x)\,(y_{3,1} - h_{3,1}(x)) \; . \tag{3.2}$$

Then the structure of the controller for asymptotic tracking is

$$\begin{aligned} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} &= S_3(x)^{-1}\left\{ \begin{bmatrix} c_{01}(\theta - \theta_{ref}(t)) \\ c_{02}(\psi - \psi_{ref}(t)) \end{bmatrix} + \begin{bmatrix} c_{11}(\dot{\theta} - \dot{\theta}_{ref}(t)) \\ c_{12}(\dot{\psi} - \dot{\psi}_{ref}(t)) \end{bmatrix} \right. \\ &\left. + \begin{bmatrix} c_{21}(L_f^2\theta - \ddot{\theta}_{ref}(t)) \\ (L_f^2\psi - \ddot{\psi}_{ref}(t)) \end{bmatrix} + \begin{bmatrix} (L_f^3\theta - \theta_{ref}^{(3)}(t)) \\ 0 \end{bmatrix} \right\} \end{aligned}$$

where $\begin{bmatrix} c_{i1} & 0 \\ 0 & c_{i2} \end{bmatrix} = \Upsilon_i$. Note that here we have used standard structure algorithm and added the stability terms at the end; the result is of cause identical. In the following simulation the poles are allways set to -5. To this end the $c_{i,j}$ are chosen as follows:

$$\begin{array}{lllllllll} c_{01} & = & 125 & , & c_{11} & = & 75 & , & c_{21} = 15 \; , \\ c_{02} & = & 25 & , & c_{12} & = & 10 & . & \end{array}$$

HOC controller The discrete approximation of the equations of motion (3.1) and the path constraints $\theta - \theta_{ref}(t) = 0$, $\psi - \psi_{ref}(t) = 0$ results in the following DAE

$$\begin{aligned} \phi_{j+1} - \phi_j &= \epsilon\dot{\phi}_{j+1} \\ \theta_{j+1} - \theta_j &= \epsilon\dot{\theta}_{j+1} \\ \psi_{j+1} - \psi_j &= \epsilon\dot{\psi}_{j+1} \\ M(\theta_{j+1})\begin{bmatrix} \dot{\phi}_{j+1} - \dot{\phi}_j \\ \dot{\theta}_{j+1} - \dot{\theta}_j \\ \dot{\psi}_{j+1} - \dot{\psi}_j \end{bmatrix} &= \epsilon F(\theta_{j+1}, \dot{\phi}_{j+1}, \dot{\theta}_{j+1}, \dot{\psi}_{j+1}) \\ &+ \epsilon B(\theta_{j+1})u_{j+1} \\ 0 &= \theta_{j+i} - \theta_{ref}(t_p + j\epsilon) + \alpha_1(\dot{\theta}_{j+i} - \dot{\theta}_{ref}(t_p + j\epsilon)) \\ 0 &= \psi_{j+i} - \psi_{ref}(t_p + j\epsilon) + \alpha_2(\dot{\psi}_{j+i} - \dot{\psi}_{ref}(t_p + j\epsilon)) \end{aligned} \tag{3.3}$$

with initial condition

$$X_0 = \begin{bmatrix} \phi(t_\rho) \\ \theta(t_\rho) \\ \psi(t_\rho) \\ \dot\phi(t_\rho) \\ \dot\theta(t_\rho) \\ \dot\psi(t_\rho) \end{bmatrix} = \begin{bmatrix} \phi_0 \\ \theta_0 \\ \psi_0 \\ \dot\phi_0 \\ \dot\theta_0 \\ \dot\psi_0 \end{bmatrix}$$

The system (3.3) can be solved numerically for $j = 1, \cdots, k$ to compute $u(k) = [u_1(k), u_2(k)]^T$. More specifically, we find

$$\begin{aligned}
\theta_{j+i} &= \tfrac{\epsilon}{\epsilon+\alpha_1}\left[\theta_{ref}(t_\rho + j\epsilon) + \alpha_1(\dot\theta_{ref}(t_\rho + j\epsilon) + \tfrac{\theta_j}{\epsilon})\right] \\
\psi_{j+i} &= \tfrac{\epsilon}{\epsilon+\alpha_2}\left[\psi_{ref}(t_\rho + j\epsilon) + \alpha_2(\dot\psi_{ref}(t_\rho + j\epsilon) + \tfrac{\psi_j}{\epsilon})\right] \\
\psi_{j+i} &= \psi_{ref}(t_\rho + j\epsilon) \\
\dot\theta_{j+1} &= \tfrac{1}{\epsilon}(\theta_{j+1} - \theta_j) \\
\dot\psi_{j+1} &= \tfrac{1}{\epsilon}(\psi_{j+1} - \psi_j)\,.
\end{aligned}$$

With $\alpha_i > 0$ we apply one step of the stabilized structure algorithm. If $\theta_{j+1} \neq i\pi$ for $i = 0, 1, \ldots$, to compute $\dot\phi_{j+1}$, we have to solve the polynomial

$$-a_1\,\dot\phi_{j+1}\dot\psi_{j+1}\sin(\theta_{j+1}) - a_2\,\dot\phi_{j+1}^2\sin(\theta_{j+1})\cos(\theta_{j+1}) - a_4\,\cos(\theta_{j+1}) = a_5\frac{(\dot\phi_{j+1} - \dot\phi_j)}{\epsilon}$$

for $\dot\phi_{j+1}$, where

$$\begin{aligned}
a_1 &= m\,r^2 + I_n \\
a_2 &= m\,r^2 + I_n - I_r \\
a_4 &= m\,g\,r \\
a_5 &= m\,r^2 + I_r\,.
\end{aligned}$$

With

$$\begin{aligned}
b_2 &= (m\,r^2 + I_n - I_r)\sin(\theta_{j+1})\cos(\theta_{j+1}) \\
b_1 &= (m\,r^2 + I_r) - \epsilon\dot\psi_{j+1}(m\,r^2 + I_n)\sin(\theta_{j+1}) \\
b_0 &= (m\,r^2 + I_r) - \epsilon m\,g\,r\,\cos(\theta_{j+1})
\end{aligned}$$

the polynomial can be expressed as

$$\dot\phi_{j+1}^2 + \frac{b_1}{\epsilon b_2}\dot\phi_{j+1} - \frac{b_0}{\epsilon b_2} = 0$$

and admits two solutions as long $\theta_{j+1} \neq \frac{\pi}{2}$:

$$\dot\phi_{j+1} = \frac{b_1}{2\epsilon b_2}\left\{-1 \pm \sqrt{1 + 4\frac{b_2\,b_0}{b_1^2}\dot\phi_j\epsilon}\right\}.$$

For small ϵ we can approximate $\sqrt{1-x} \approx 1 - \frac{1}{2}x$ and obtain

$$\dot\phi_{j+1} = \frac{b_1}{2\epsilon b_2}\left\{-1 \pm \left(1 + 2\frac{b_2\,b_0}{b_1^2}\dot\phi_j\epsilon\right)\right\}.$$

Taking the solution (+) we get

$$\dot{\phi}_{j+1} = \frac{b_0}{b_1}\dot{\phi}_j \ ,$$

but this is exactly the solution we find for the excluded case $\theta_{j+1} = \frac{\pi}{2}$, since there $b_2 = 0$. As a consequence, solution (−) is not valid. To compute $u_{1_{j+1}}$ and $u_{2_{j+1}}$ we use simply the forth and six rows of (3.3). Again, as for the explicit solution of the control $[u_1, u_2]$ discussed in the previous paragraph, the solution of u_1 is not defined for $\theta_{j+1} \in \{0, \pi\}$ since in the expression for u_1 there is a division by $\sin(\theta_{j+1})$ Note that instead of solving the DAE explicitly, one could wish to use iterative methods to compute X^{ρ}_{j+1}. However, if the solution of the DAE is not unique, as in the present case, the iterative method might converge to the wrong solution, and special care must be taken.

Simulations The following plots show first a simulation result using a controller, obtained by exact input-output linearization and discretization (Figure 3.2). Then, the simulation result of the same system with HOC controller is shown in Figure 3.3 with zero steps of the stabilized structure algorithm ($\alpha_1 = \alpha_2 = 0$) and in Figure 3.4 with one step of the stabilized structure algorithm on θ ($\alpha_1 = \frac{1}{5}, \alpha_2 = 0$).

In the simulation we have chosen the following reference functions for the output $y = [\theta, \psi]^T$.

$$\begin{aligned}
\psi_{ref}(t) &= -9\,t + 2\,(t-5)\,\mathrm{atan}(t-5) - \ln(1 + (t-5)^2) \\
\dot{\psi}_{ref}(t) &= -9 + 2\,\mathrm{atan}(t-5) \\
\theta_{ref}(t) &= \tfrac{\pi}{2} + 0.3\sin(t) \\
\dot{\theta}_{ref}(t) &= 0.3\cos(t)
\end{aligned}$$

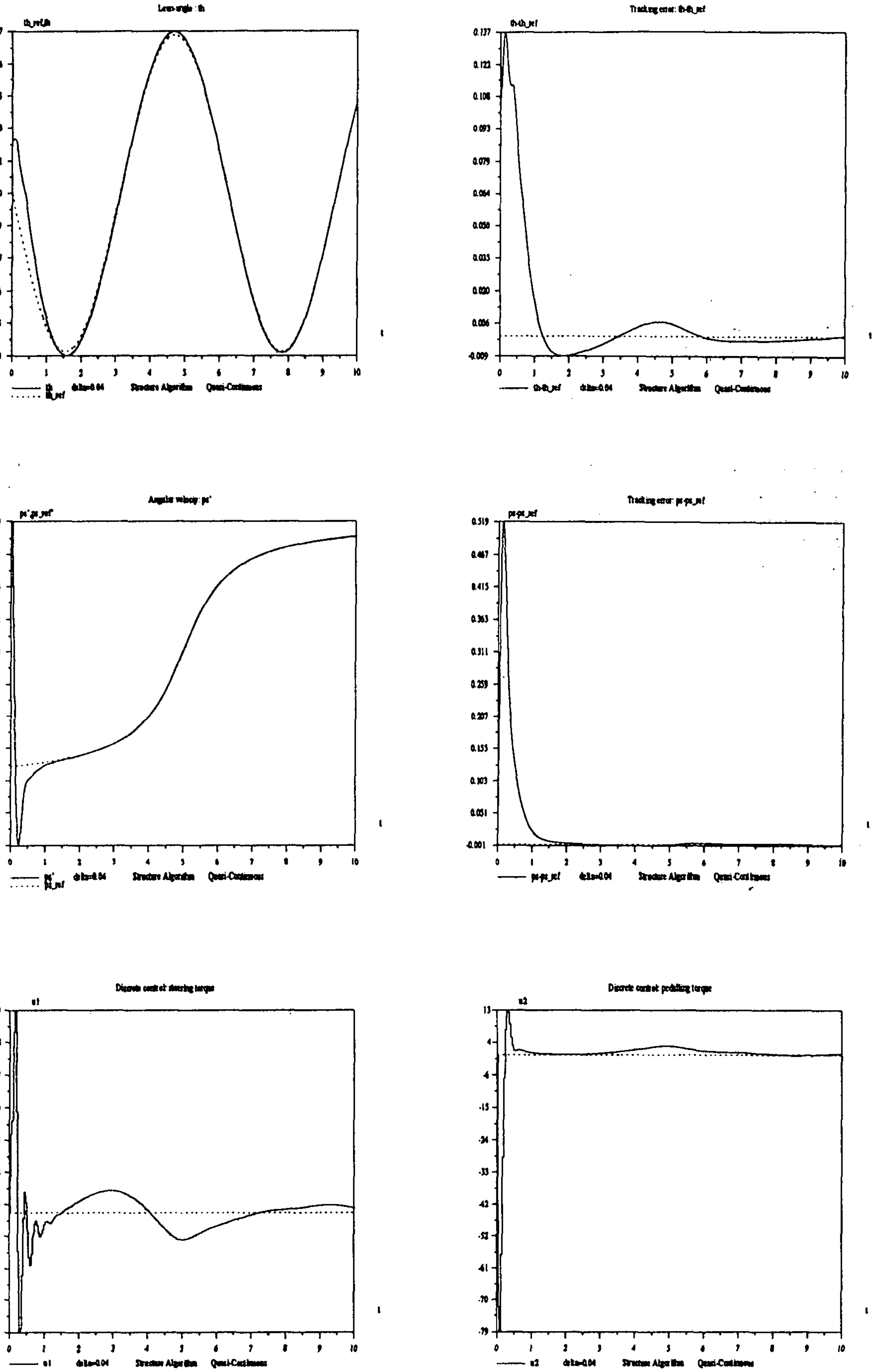

Figure 3.2 Simulation results with controller obtained by exact input-output linearization *Tracking of the lean-angle and the roll angle. The discrete realization of the controller works as long as $\delta \leq 0.04$, for $\delta > 0.04$ the controlled system becomes unstable.*

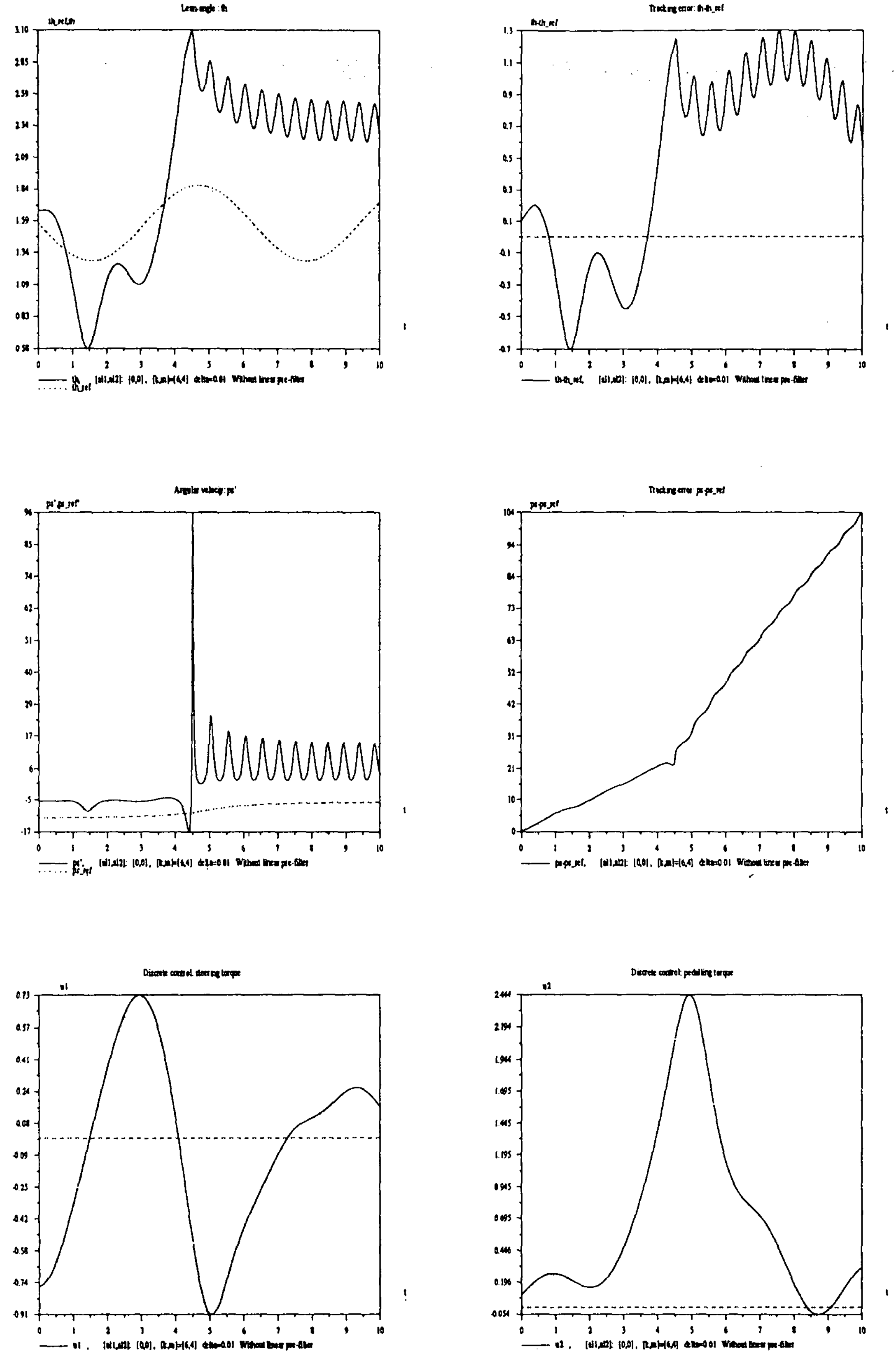

Figure 3.3 Simulation results with HOC controller with $\alpha_1 = \alpha_2 = 0$. *Tracking of the lean angle and the roll angle of the wheel turns. The closed-loop system is unstable even for very small δ*

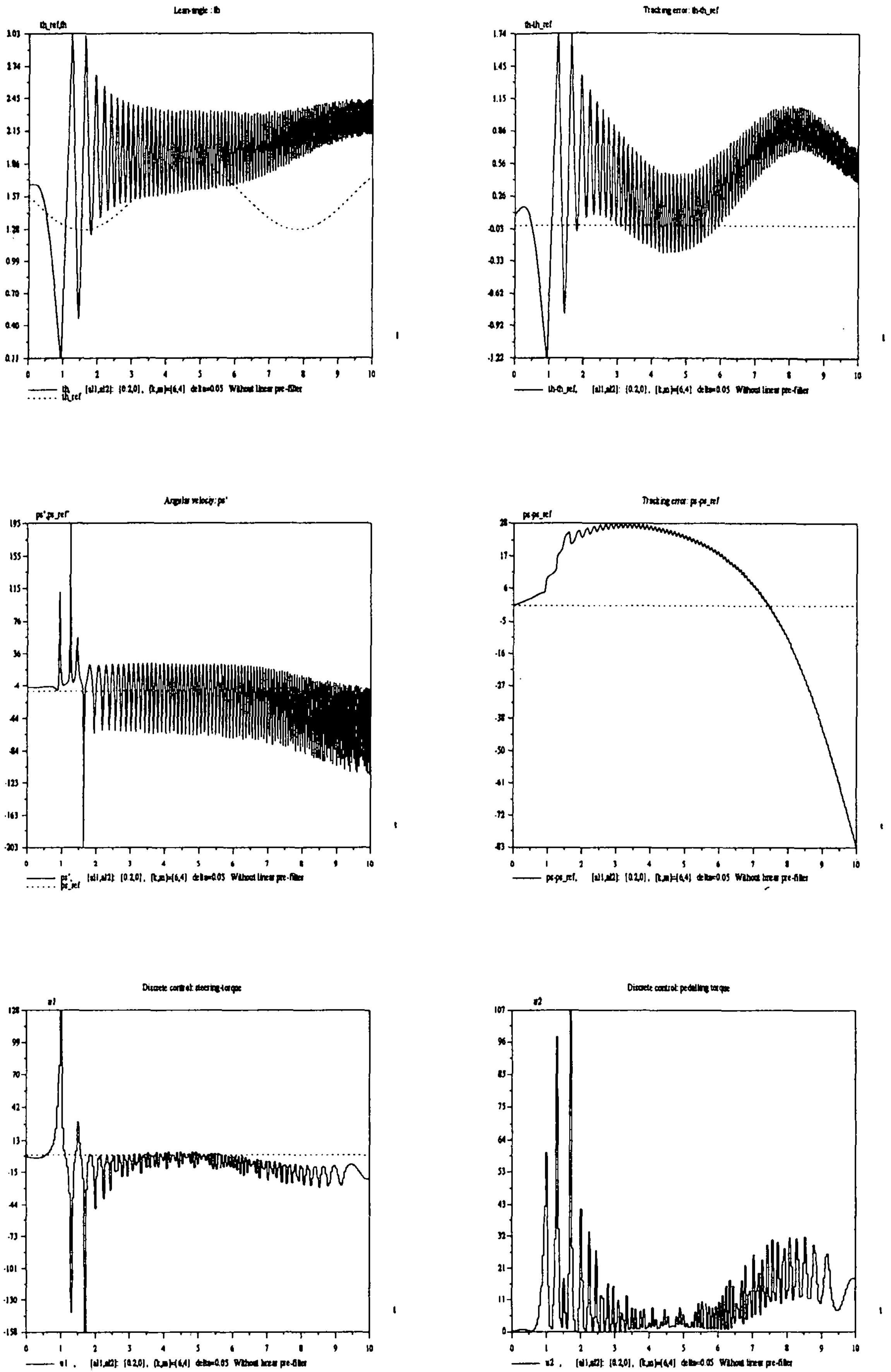

Figure 3.4 Simulation results with HOC controller with $\alpha_1 = \frac{1}{5}, \alpha_2 = 0$. *Tracking of the lean angle and the roll angle of the wheel. The closed-loop system is unstable even though we applied one step of the stabilized structure algorithm*

In order to get more insight into how HOC control works and where are its limits we present here an analysis in the linear case.

4 Tracking controller for linear systems

In this section, we consider the HOC controller for linear systems. Consider the linear system

$$\dot{x} = Ax + Bu \quad (4.1a)$$
$$y = Cx + Du . \quad (4.1b)$$

The dimension of the state x is n, the dimensions of y and u are are both m. The goal is to find a stabilizing control.

$$u = F(x, \xi(t), \dot{\xi}(t), \ldots, \xi^{\nu}(t)) \quad (4.2)$$

such that $y(t)$ converges exponentially to $\xi(t)$, the reference trajectory. We shall assume that system (4.1) is invertible and minimum phase.

The standard solution of this problem is given by the structure algorithm presented in Section 2.1. From equation (2.9) follows the controller

$$u = \overline{D}^{-1}[\overline{C}x - \Upsilon(\frac{d}{dt})\xi(t)] . \quad (4.3)$$

If we denote the transfer-function of (4.1) by $H(s)$ (i.e.,$y(s) = H(s)u(s)$), the polynomial-matrix $R(s) = \overline{D}^{-1}\Upsilon(s)$ and the feedback matrix K_s satisfy

$$K_s = \overline{D}^{-1}\overline{C} , \quad (4.4)$$

$$H_b(s)R(s) = [(C + DK_s)(sI - (A + BK_s))^{-1}B + D]\overline{D}^{-1}\Upsilon(s) = I , \quad (4.5)$$

where $H_b(s)$ is the transfer-function of the closed-loop system. The closed loop system is stable (remember that (4.1) is minimum phase).

4.1 HOC controller for linear systems

To analyze the properties of the HOC controller applied to system (4.1), we are going to make a few simplifying assumptions. The first assumption is that the solution of the DAE (2.14) is constructed without any error, the second assumption is that we consider the case where ϵ and δ are very small so that we can consider the limiting behavior as ϵ and δ go to zero. For the sake of simplicity we shall assume that (4.1) represents the system after the application of the steps of the stabilized structure algorithm, if any, that may be required in the implementation of the HOC controller.

A key result which is needed in our analysis is how to construct the solution of a linear DAE.

Solution of a linear DAE Consider the linear DAE.

$$\dot{X} = AX + BU \quad (4.6a)$$
$$\xi(t) = CX + DU \quad (4.6b)$$
$$X(t_\rho) = x(t_\rho) , \quad (4.6c)$$

where $x(t_\rho)$ is known and $\xi(t)$ is a given time function. Clearly (4.6) is the linear version of (2.14) for. The solution $U(t)$ of (4.6) can be constructed by noting that system (4.6) can be expressed, in transfer domain, as

$$\begin{bmatrix} -sI+A & B \\ C & D \end{bmatrix} \begin{bmatrix} X \\ U \end{bmatrix} = \begin{bmatrix} 0 \\ \xi(t) \end{bmatrix} . \tag{4.7}$$

The row-compression of $\begin{bmatrix} B \\ D \end{bmatrix}$ by the matrix

$$V = \begin{bmatrix} V_1 & V_2 \\ V_3 & V_4 \end{bmatrix}$$

puts the system pencil in (4.7) to

$$\begin{bmatrix} -sV_1 + V_1A + V_2C & V_1B + V_2D \\ -sV_3 + V_3A + V_4C & V_3B + V_4D \end{bmatrix} = \begin{bmatrix} -Es+F & 0 \\ -Hs+J & I \end{bmatrix} \tag{4.8}$$

so that we get

$$\begin{bmatrix} -Es+F & 0 \\ -Hs+J & I \end{bmatrix} \begin{bmatrix} X \\ U \end{bmatrix} = \begin{bmatrix} V_2\xi(t) \\ V_4\xi(t) \end{bmatrix} . \tag{4.9}$$

Since the pencil $\{E, F\}$ is regular (thanks to invertibility assumption on (4.1)) there exist two matrices M and Q such that

$$M(-Es+F)Q = \begin{bmatrix} -Is+A_1 & 0 \\ 0 & -sN+I \end{bmatrix} \tag{4.10}$$

is in Kronecker normal form (N is a nilpotent matrix). The right multiplication by Q implies a change of variable $Z = Q^{-1}X$. Obviously, the transformation into Kronecker normal form separates the new state Z in a continuous part Z_1 and a discontinuous or impulsive part Z_2. In the sequel we note $Z_1 = P_1^T Z$ and $Z_2 = P_2^T Z$, where $P_1^T = [I\ 0]$ and $P_2^T = [0\ I]$. P_1^T determines the projection of X which is continuous at 0, and more importantly the projection of $x(t_\rho)$ which contributes to the solution of (4.6)[2].

Now (4.9) can be rewritten as

$$\begin{bmatrix} M & 0 \\ 0 & I \end{bmatrix} \begin{bmatrix} -Es+F & 0 \\ -Hs+J & I \end{bmatrix} \begin{bmatrix} Q & 0 \\ 0 & I \end{bmatrix} \begin{bmatrix} Q^{-1} & 0 \\ 0 & I \end{bmatrix} \begin{bmatrix} X \\ U \end{bmatrix} = \begin{bmatrix} MV_2\xi(t) \\ V_4\xi(t) \end{bmatrix} \tag{4.11}$$

or, equivalently

$$\Leftrightarrow \begin{bmatrix} -sI+A_1 & 0 & 0 \\ 0 & -sN+I & 0 \\ (-HQs+JQ)P_1 & (-HQs+JQ)P_2 & I \end{bmatrix} \begin{bmatrix} Z_1 \\ Z_2 \\ U \end{bmatrix} = \begin{bmatrix} \gamma_1 \\ \gamma_2 \\ V_4\xi(t) \end{bmatrix} , \tag{4.12}$$

where $\gamma_1 = P_1^T MV_2\xi(t)$ and $\gamma_2 = P_2^T MV_2\xi(t)$. From (4.12) we obtain an expression for the control at t_ρ^+

$$\begin{aligned} U(t_\rho^+) &= (HQ\frac{d}{dt} - JQ)P_1Z_1(t_\rho^+) + (HQ\frac{d}{dt} - JQ)P_2Z_2(t_\rho^+) + V_4\xi(t_\rho) \\ &= H\,Q\,P_1\dot{Z}_1(t_\rho^+) - J\,Q\,P_1Z_1(t_\rho^+) + H\,Q\,P_2\dot{Z}_2(t_\rho^+) - J\,Q\,P_2Z_2(t_\rho^+) + V_4\xi(t_\rho) \,. \end{aligned} \tag{4.13}$$

[2] It is exactly this projection that the HOC controller uses to generate the control. So that in some sense the HOC controller can be thought of as an output feedback controller, the output being $P_1^T x$. This projection may be completely empty (as it was for the example given in the introduction), but each step of the structure algorithm applied to the original system prior to the application of the numerical method, increases the size of this projection. If the structure algorithm is carried through completely then, $P_1^T = I$ and (4.6) can be solved exactly by noting that in this case D is invertible.

We are only interested in the solution of U at t_ρ^+ even though what is really used in the HOC control is $U(t_\rho + k\epsilon)$. But here we are studying the limiting case where ϵ and thus $k\epsilon \to 0$. Expression (4.13) does not specify $U(t_\rho^+)$ yet because even though $Z_1(t_\rho^+)$ is known to be

$$Z_1(t_\rho^+) = Z_1(t_\rho^-) = P_1^T x_{t_\rho} \ ,$$

the quantities $\frac{d}{dt} Z_1(t_\rho^+)$ (i.e., $\dot{Z}_1(t_\rho^+)$), $Z_2(t_\rho^+)$ and $\frac{d}{dt} Z_2(t_\rho^+)$ are not known. The first two block-rows of (4.12) can be used to compute them. The first row gives $\dot{Z}_1(t_\rho^+)$

$$sZ_1 = A_1 Z_1 - \gamma_1$$

and

$$(-sN + I)\, Z_2 = \gamma_2 \ .$$

Now N is nilpotent of index ν so that $(-s\,N + I)Z_2 = \gamma_2$ implies that

$$Z_2 = \sum_{i=0}^{\nu} (sN)^i \gamma_2$$

and

$$sZ_2 = \sum_{i=0}^{\nu} s(sN)^i \gamma_2 \ .$$

Plugging back these expressions in (4.13) yields an equation depending exclusively on $Z_1(t_\rho^+) = P_1 Q^{-1} x(t_\rho^+)$, $\xi(t)$ and its derivatives.

$$\begin{aligned} U(t_\rho^+) &= (HQP_1A_1 - JQP_1)Z_1(t_\rho) + R_\xi\left(\frac{d}{dt}\right)\xi(t)|_{t=t_\rho} \\ &= K_J X(t_\rho) + R_\xi\left(\frac{d}{dt}\right)\xi(t)|_{t=t_\rho} \end{aligned} \tag{4.14}$$

where

$$K_J = (HQP_1A_1P_1^T - JQP_1P_1^T)Q^{-1} \tag{4.15}$$

and

$$\begin{aligned} R_\xi(s) &= V_4 - \left(HQP_1P_1^T - (sHQ - JQ)P_2(\sum_{i=0}^{\nu}(sN)^i)P_2^T\right)MV_2 \\ &= (V_4 - HQP_1P_1^T - JQP_2P_2^T)MV_2 + \\ &\quad (HQP_2P_2^T - JQP_2NP_2^T)MV_2 s + \\ &\quad (HQP_2NP_2^T - JQP_2N^2P_2^T)MV_2 s^2 + \cdots \\ &= \sum_{i=0}^{\nu+1} R_i s^i \ . \end{aligned} \tag{4.16}$$

Following the HOC control procedure where $\delta \to 0$ the control $u(t)$ is chosen as

$$u(t) \ = \ K_J x(t) + R_\xi\left(\frac{d}{dt}\right)\xi(t) \tag{4.17}$$

4.2 Stability analysis

The HOC controller (4.17) is stable if and only if the closed-loop system

$$\dot{x} = (A + B\,K_J)x \tag{4.18}$$

is stable, which means that the eigenvalues of $A + B\,K_J$ have negative real parts. The stability property clearly is independent of the choice of coordinate system. The control (4.17) is computed from the system (4.1) after the application of some steps of the stabilized structure algorithm, but is to be applied to the original system. However, the A, B matrices are the same for both systems. Then in order to determine the stability properties of $A + B\,K_J$ it suffices to see what happens if (4.17) is fed back to system (4.1). For reasons that will become clear shortly, we consider analyzing the stability of the system in the coordinate system

$$z = Q^{-1}x \ ,$$

where Q is defined as in the previous section. Let us suppose that in the z coordinate system (4.1) can be expressed as

$$\begin{aligned} \begin{bmatrix} \dot{z}_1 \\ \dot{z}_2 \end{bmatrix} &= \begin{bmatrix} \overline{A}_1 & \overline{A}_2 \\ \overline{A}_3 & \overline{A}_4 \end{bmatrix} \begin{bmatrix} z_1 \\ z_2 \end{bmatrix} + \begin{bmatrix} \overline{B}_1 \\ \overline{B}_2 \end{bmatrix} u && (4.19a) \\ y &= \begin{bmatrix} \overline{C}_1 & \overline{C}_2 \end{bmatrix} \begin{bmatrix} z_1 \\ z_2 \end{bmatrix} + \overline{D}u \ . && (4.19b) \end{aligned}$$

Theorem 4.1 *Let K_J and Q as defined in the previous section and let $\overline{K}_J = K_J Q$, then*

a) The application of $u = \overline{K}_J z$ sets $\overline{C}_1$ and $\overline{A}_3$ to zero, i.e.,

$$\begin{aligned} \begin{bmatrix} \overline{A}_1 & \overline{A}_2 \\ \overline{A}_3 & \overline{A}_4 \end{bmatrix} + \begin{bmatrix} \overline{B}_1 \\ \overline{B}_2 \end{bmatrix} \overline{K}_J &= \begin{bmatrix} \overline{\overline{A}}_1 & \overline{A}_2 \\ 0 & \overline{A}_4 \end{bmatrix} && (4.20a) \\ \begin{bmatrix} \overline{C}_1 & \overline{C}_2 \end{bmatrix} + \overline{D}\,\overline{K}_J &= \begin{bmatrix} 0 & \overline{C}_2 \end{bmatrix} \ . && (4.20b) \end{aligned}$$

b) The eigenvalues of $\overline{\overline{A}}_1$ are the transmission zeros of (4.1).

c) The decomposition (z_1, z_2) isolates the largest output-nulling (A, B)-invariant subspace $\mathcal{V}^*$ of (4.1).

Proof Note that the feedback matrix $\overline{K}_J = K_J Q$ is nothing but the HOC feedback in the z-coordinates, so that it has the following structure

$$\overline{K}_J = [K_1 \ 0] \ . \tag{4.21}$$

a) Consider system (4.1) in the form

$$\begin{bmatrix} -s\,I + A & B \\ C & D \end{bmatrix} \begin{bmatrix} x \\ u \end{bmatrix} = \begin{bmatrix} 0 \\ y \end{bmatrix} .$$

The change of coordinates with $z = Q^{-1}x$ implies that

$$\begin{bmatrix} -s\,I + A & B \\ C & D \end{bmatrix} \begin{bmatrix} Q & 0 \\ 0 & I \end{bmatrix} \begin{bmatrix} z \\ u \end{bmatrix} = \begin{bmatrix} 0 \\ y \end{bmatrix} . \tag{4.22}$$

The left multiplication by $\begin{bmatrix} M & 0 \\ 0 & I \end{bmatrix}$ and V yields

$$\begin{bmatrix} -sI + A_1 & 0 & 0 \\ 0 & -sN + I & 0 \\ (-HQs + JQ)P_1 & (-HQs + JQ)P_2 & I \end{bmatrix} \begin{bmatrix} z_1 \\ z_2 \\ u \end{bmatrix} = \begin{bmatrix} P_1^T M V_2 y \\ P_2^T M V_2 y \\ V_4 y \end{bmatrix} \tag{4.23}$$

The substitution of u in (4.23) by U as defined in (4.14),i.e.,

$$u = (HQP_2A_1 - JQP_1)z_1 + R_\xi(s)\xi \ . \tag{4.24}$$

yields

$$\begin{bmatrix} -sI + A_1 & 0 \\ 0 & -sN + I \\ HQP_1(-sI + A_1) & (-HQs + JQ)P_2 \end{bmatrix} \begin{bmatrix} z_1 \\ z_2 \end{bmatrix} = \begin{bmatrix} P_1^T M V_2 y \\ P_2^T M V_2 y \\ V_4 y - R_\xi(s)\xi(t) \end{bmatrix} . \tag{4.25}$$

By left multiplication of (4.25) with $\begin{bmatrix} I & 0 & 0 \\ 0 & I & 0 \\ -HQP_1 & 0 & I \end{bmatrix}$ we get

$$\begin{bmatrix} -sI + A_1 & 0 \\ \begin{bmatrix} 0 \\ 0 \end{bmatrix} & -s\begin{bmatrix} N \\ HQP_2 \end{bmatrix} + \begin{bmatrix} I \\ JQP_2 \end{bmatrix} \end{bmatrix} \begin{bmatrix} z_1 \\ z_2 \end{bmatrix} = \begin{bmatrix} P_1^T M V_2 y \\ \begin{bmatrix} P_2^T M V_2 y \\ (-HQP_1P_1^T M V_2 + V_4)y - R_x i(s))\xi \end{bmatrix} \end{bmatrix} . \tag{4.26}$$

Let $L = \begin{bmatrix} L_{11} & L_{12} \\ L_{21} & L_{22} \end{bmatrix}$ be an invertible matrix that row-compresses $\begin{bmatrix} N \\ HQP_2 \end{bmatrix}$. Then a pre-multiplication of (4.26) with $\begin{bmatrix} I & 0 \\ 0 & L \end{bmatrix}$ puts equation (4.25) into the form

$$\underbrace{\begin{bmatrix} -sI + A_1 & 0 \\ 0 & -sI + A_4 \\ 0 & C_2 \end{bmatrix}}_{\Sigma} \begin{bmatrix} z_1 \\ z_2 \end{bmatrix} = \underbrace{\begin{bmatrix} P_1^T M V_2 y \\ (L_{11}P_2^T M V_2 - L_{12}(HQP_1P_1^T M V_2 - V_4))y - L_{12}R_\xi(s)\xi(t) \\ (L_{21}P_2^T M V_2 - L_{22}(HQP_1P_1^T M V_2 - V_4))y - L_{22}R_\xi(s)\xi(t) \end{bmatrix}}_{\chi} . \tag{4.27}$$

The proceeding procedure might be done in one step: Let (A, B, C, D) the original system, K_J the computed feedback-matrix, and the pair (M, Q) defined in (4.10). Then

$$\underbrace{\begin{bmatrix} I & 0 & 0 \\ 0 & L_{11} & L_{12} \\ 0 & L_{21} & L_{22} \end{bmatrix} \begin{bmatrix} I & 0 & 0 \\ 0 & I & 0 \\ -HQP_1 & 0 & I \end{bmatrix} \begin{bmatrix} M_{11} & M_{12} & 0 \\ M_{21} & M_{22} & 0 \\ 0 & 0 & I \end{bmatrix} V \begin{bmatrix} Q_{11} & Q_{12} & 0 \\ Q_{21} & Q_{22} & 0 \\ 0 & 0 & I \end{bmatrix}}_{T}$$

$$\underbrace{\begin{bmatrix} \begin{bmatrix} Q_{11} & Q_{12} \\ Q_{21} & Q_{22} \end{bmatrix}^{-1} & \begin{matrix} 0 \\ 0 \end{matrix} \\ \begin{matrix} 0 & 0 \end{matrix} & I \end{bmatrix}}_{\Xi} \begin{bmatrix} -sI + A + BK_J \\ C + DK_J \end{bmatrix} Q\,z = \chi$$

where

$$\Xi = \begin{bmatrix} -s\,I + \overline{A} + \overline{B}\,\overline{K}_J \\ \overline{C} + D\overline{K}_J \end{bmatrix}$$

or equivalently

$$\Sigma\, z = T\,\Xi\, z = \chi\ .$$

To get more information about the particular structure of the matrix T consider just the coefficients of the terms in s.

$$\begin{bmatrix} T_1 & T_2 & T_3 \\ T_4 & T_5 & T_6 \\ T_7 & T_8 & T_9 \end{bmatrix} \begin{bmatrix} I & 0 \\ 0 & I \\ 0 & 0 \end{bmatrix} = \begin{bmatrix} I & 0 \\ 0 & I \\ 0 & 0 \end{bmatrix} . \tag{4.28}$$

From (4.28) it follows immediately that

$$T = \begin{bmatrix} I & 0 & T_3 \\ 0 & I & T_6 \\ 0 & 0 & T_9 \end{bmatrix} ,$$

where T_9 is an invertible matrix. Now we find that

$$\begin{aligned} T \begin{bmatrix} -s\,I + \overline{A} + \overline{B}\,\overline{K}_J \\ \overline{C} + D\overline{K}_J \end{bmatrix} &= \begin{bmatrix} -sI + \overline{A}_1 + \overline{B}_1\overline{K}_J + T_3(\overline{C}_1 + D\overline{K}_J) & \overline{A}_2 + T_3\overline{C}_2 \\ \overline{A}_3 + \overline{B}_2\overline{K}_J + T_6(\overline{C}_1 + D\overline{K}_J) & -s\,I + \overline{A}_4 + T_6\overline{C}_2 \\ T_9(\overline{C}_1 + D\overline{K}_J) & T_9\overline{C}_2 \end{bmatrix} \\ &= \begin{bmatrix} -sI + A_1 & 0 & 0 \\ 0 & -s\,I + A_4 & 0 \\ 0 & C_2 & 0 \end{bmatrix} \end{aligned} \tag{4.29}$$

and since T_9 is invertible we verify that

$$\begin{aligned} T_9^{-1}\overline{\overline{C}}_1 &= \overline{C}_1 + D\overline{K}_J \;=\; 0 \\ \Leftrightarrow \overline{\overline{C}}_1 &= 0 \end{aligned}$$

and consequently that

$$\begin{aligned} \overline{\overline{A}}_1 &= \overline{A}_1 + \overline{B}_1\overline{K}_J \\ \overline{A}_2 &= -T_3\overline{C}_2 \\ \overline{\overline{A}}_3 &= 0 \\ \overline{A}_4 &= A_4 - T_6\overline{C}_2 \end{aligned}$$

b) The pencil $\{E, F\}$ in (4.9) contains all zeros (finite and infinite) of (4.7) (Note that $\det(-s\,E + F) = c \det\left(\begin{bmatrix} -s\,I + A & B \\ C & D \end{bmatrix}\right)$). The zero-structure of a linear system is invariant with respect to the three algebraic operations on (4.7): change of variables, state-feedback and output-injection (we can multiply matrix $\begin{bmatrix} -s\,I + A & B \\ C & D \end{bmatrix}$ from the left and the right by invertible constant matrices without changing the zero-structure). (M, Q) puts (4.9) in Kronecker normal form, i.e., separates finite and infinite zeros. By definition A_1 contains all finite zeros and $(-s\,N + I)$ all zeros at infinity, which proves b) since $A_1 = \overline{\overline{A}}_1$.

c) The computations in paragraph a) showed that the feedback $u = \overline{K}_J z$ renders z_1 unobservable, i.e., the first columns of Q span an output-nulling sub-space $\mathcal{V}$ and $z_1 \in \mathcal{V}$. In b) we have seen that A_1 containes all finite zeros of (4.7). By assumption (4.7) is invertible and for this class of linear system it is known that the dimension of the largest output-nulling subspace $\mathcal{V}^*$, which is uniquely defined, is equal to the number of finite zeros, counting multiplicity [6]. This shows that $\mathcal{V} = \mathcal{V}^*$. □

Note that stability of the closed-loop system depends on the stability of $\overline{A}_4$ ($\overline{\overline{A}}_1$ is stable under the minimum phase assumption), which is unaffected by the HOC feedback. If $\overline{A}_4$ is not stable, the HOC controller will produce an unstable system. But by construction, we know that $(\overline{A}_4, \overline{B}_2)$ is a controllable pair, and thus there exists a matrix K_2 such that $\overline{A}_4 + \overline{B}_2 K_2$ is stable. If we apply a preliminary feedback

$$\begin{aligned} u &= K_2 z_2 + v \\ &\stackrel{\text{def}}{=} \overline{K}_n z + v \end{aligned}$$

before we apply the HOC controller, then system (4.19) becomes

$$\begin{bmatrix} \dot{z}_1 \\ \dot{z}_2 \end{bmatrix} = \begin{bmatrix} \overline{A}_1 & \overline{\overline{A}}_2 \\ \overline{A}_3 & \overline{\overline{A}}_4 \end{bmatrix} \begin{bmatrix} z_1 \\ z_2 \end{bmatrix} + \begin{bmatrix} \overline{B}_1 \\ \overline{B}_2 \end{bmatrix} v \tag{4.30a}$$

$$y = \begin{bmatrix} \overline{C}_1 & \overline{\overline{C}}_2 \end{bmatrix} \begin{bmatrix} z_1 \\ z_2 \end{bmatrix} + \overline{D} v \ , \tag{4.30b}$$

where $\overline{\overline{A}}_4 = \overline{A}_4 + \overline{B}_2 K_2$ is stable. If we now apply HOC control to (4.30), we obtain a stable closed-loop system!

To summerize the preliminary feedback procedure, we first have to find the coordinate transformation matrix Q as explained in Section 4.1 and the new representation (4.19). Test the stability of $\overline{A}_4$, and construct K_2 if necessary such that $\overline{A}_4 + \overline{B}_2 K2$ is stable. Then let

$$K_n = \begin{bmatrix} 0 & K_2 \end{bmatrix} Q^{-1}$$

and apply to the original system (4.1) the preliminary feedback

$$u = K_n x + v \ .$$

The HOC controller can now be applied to the new system

$$\begin{aligned} \dot{x} &= (A + B\,K_n)x + B\,v \\ y &= (C + D\,K_n)x + D\,v \end{aligned}$$

with guaranteed success.

Of course the preliminary feedback and more generally the HOC control is not at all needed in the case of linear systems since established alternatives exist. The above analysis is only presented to illustrate the ideas behind these concepts. To apply the preliminary feedback idea in the nonlinear case, a simple approach would be to construct the preliminary feedback based on the model of the system linearized around some nominal operation point x^0. Of course there is no guarantee that such a preliminary feedback does the job in particular if the actual trajectory $x(t)$ of the system does not remain close to the nominal operating point x^0. But if HOC does not work or has poor performance such a preliminary feedback may improve the situation as we shall see later in an example.

There may be the other ways of constructing a preliminary feedback by taking into account the nonlinearities of the system. This problem is currently under investigation.

4.3 Tracking properties

Just as the stability analysis, we can study tracking properties in any coordinate system. Let us consider the representation (4.20). It is clear that, after the application of the preliminary feedback if needed, HOC controller yields the following output

$$y = (\overline{\overline{C}}_2(s\,I - \overline{\overline{A}}_4)^{-1}\overline{B}_2 + \overline{D})R_\xi(s)\xi + \overline{\overline{C}}_2(s\,I - \overline{\overline{A}}_4)^{-1}z_2(0)\ .$$

Theorem 4.2

$$(\overline{\overline{C}}_2(s\,I - \overline{\overline{A}}_4)^{-1}\overline{B}_2 + \overline{D})R_\xi(s) = I \tag{4.31}$$

and since $\overline{\overline{A}}_4$ is stable (by construction), $e(t) = y(t) - \xi(t)$ converges exponentially to zero.

Proof The system representation (4.12) in Kronecker normal form is clearly nothing but another representation of (4.1), to compute U we just replaced $y(t)$ by $\xi(t)$. The unique transfer-matrix of the closed loop system-representations (4.12) and (4.1) is $H_b(s)$, as defined before. Since equation (4.14) is computed using (4.12) the polynomial matrix $R_\xi(s)$ is by construction the polynomial part $P(s)$ of the inverse of the transfer-matrix $H_b(s)^{-1} = P(s) + R(s)$, where $R(s)$ is the rational part of $H(s)$. In fact, the polynomial part $P(s)$ corresponds to the discontinuous part z_2 the rational part $R(s)$ to the continuous part z_1. Since after application of the feedback-matrix K_J to (4.1) z_1 is unobservable the inverse of $H_b(s)$ is only polynomial and we have $H_b(s)^{-1} = R_\xi(s)$. □

Note that $e(t)$ represents the tracking error for the system which is not the original system that we had considered, but the system obtained from possible applications of few steps of the structure algorithm to the original system. But as we have already seen, as long as we use the stabilized version of the structure algorithm, if the tracking error of the resulting system converges to zero so does that of the original system.

4.4 Simple Example

Let us consider the following example

$$\dot{x}_1 = x_2 \tag{4.32a}$$

$$\dot{x}_2 = c_1x_1 + c_2x_2 + u \tag{4.32b}$$

$$y = x_1 + \lambda x_2 \tag{4.32c}$$

The transfer-matrix of (4.32) is clearly

$$H(s) = \frac{1+\lambda s}{-c_1 - c_2 s + s^2}\ ,$$

which is invertible and minimum phase as long as $\lambda \geq 0$. Apply the procedure given in Section (4.1). With

$$V = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

we obtain

$$V\left[\left|\begin{array}{cc} -s & 1 \\ c_1 & -s+c_2 \\ \hline 1 & \lambda \end{array}\right| \quad \left|\begin{array}{c} 0 \\ 1 \\ \hline 0 \end{array}\right|\right] = \left[\left|\begin{array}{c} -E\,s+I \\ \hline -H\,s+J \end{array}\right| \quad \left|\begin{array}{c} 0 \\ \hline I \end{array}\right|\right] = \left[\left|\begin{array}{cc} -s & 1 \\ 1 & \lambda \\ \hline c_1 & c_2-s \end{array}\right| \quad \left|\begin{array}{c} 0 \\ 0 \\ \hline 1 \end{array}\right|\right] \tag{4.33}$$

Case 1: $\lambda \neq 0$.

Then the pair

$$(M,Q) = \left(\begin{bmatrix} \lambda & -1 \\ 1 & 0 \end{bmatrix}, \begin{bmatrix} \lambda^{-1} & 0 \\ -\lambda^{-2} & \lambda^{-1} \end{bmatrix} \right)$$

puts $-E\,s + F = \begin{bmatrix} -s & 1 \\ 1 & \lambda \end{bmatrix}$ in Kronecker normal-form.

$$\begin{bmatrix} -\lambda^{-1} - s & 0 \\ 0 & 1 \end{bmatrix},$$

i.e., the nilpotent matrix $N = 0$ and $A_1 = -\lambda^{-1}$ and $P_1 = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$, $P_1 = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$.

Now we compute the feedback-vector K_J using (4.15)

$$K_J = \begin{bmatrix} (\lambda^{-3} - c_2\lambda^{-2} - c_1\lambda^{-1}) & 0 \end{bmatrix}$$

and $R_\xi(s)$ using (4.16).

$$R_\xi(s) = \frac{s\lambda - \lambda c_2 - 1}{\lambda^2}.$$

System (4.32) in z-coordinates takes the form

$$\begin{aligned} \dot{z} &= \begin{bmatrix} -\lambda^{-1} & 1 \\ (-\lambda^{-2} - c_2\lambda^{-1} + c_1) & (c_2 + \lambda^{-1}) \end{bmatrix} + \begin{bmatrix} 0 \\ \lambda^{-1} \end{bmatrix} u \\ y &= \begin{bmatrix} 0 & 1 \end{bmatrix} z. \end{aligned} \tag{4.34}$$

A trivial computation shows that $\overline{BK_J} = \lambda^{-2} + c_2\lambda^{-1} - c_1$ and that the transmission zero of (4.32) are is $-\lambda^{-1}$. For the closed loop system

$$\dot{z} = (\overline{A} + \overline{BK_J})z + \overline{B}v$$

we compute the transfer-function

$$H_b(s) = \frac{(s+\lambda^{-1})\,\lambda^2}{(s+\lambda^{-1})(s\lambda - \lambda c_2 - 1)}$$

and we verify that $R_\xi(s)\,H_b(s) = I$.

The conditions for stability are first that $\lambda^{-1} > 0$, (i.e., that the original system is minimum phase) and second $c_2 < -\lambda^{-1}$. If $c_2 < -\lambda^{-1}$ is not satisfied, we have to perform a preliminary feedback

$$u = \alpha\, c_2 + v.$$

Case 2: $\lambda = 0$.

In this case

$$(M,Q) = \left(\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \right)$$

puts the upper left sub-matrix$-E\,s + F = \begin{bmatrix} -s & 1 \\ c_1 & c_2 \end{bmatrix}$ in Kronecker normal-form.

$$\begin{bmatrix} 1 & -s \\ 0 & 1 \end{bmatrix}$$

Here z_1 is empty, K_J is empty and $P_2 = I$. The polynomial matrix $R_\xi(s)$ is

$$R_\xi(s) = c_1 + c_2 s - s^2$$

A preliminary feedback K_n is to be chosen such that the system

$$(\overline{A}, \overline{B}) = \left(\begin{bmatrix} c_2 & c_1 \\ 1 & 0 \end{bmatrix} , \begin{bmatrix} 1 \\ 0 \end{bmatrix} \right) .$$

has stable poles.

4.5 Example

In this example we apply the method to the linearized model of the wheel. As the equations of motion (3.1) are independent of the value of ϕ and the position of the center of mass (x, y), in the linear model we keep just the variables $x = [\theta, \psi, \phi, \dot{\theta}, \dot{\psi}]^T$. To compute the linear model we linearize the non-linear system around the trajectory

$$x_0(t) = \begin{pmatrix} \frac{\pi}{2} \\ -6t \\ 0 \\ 0 \\ -6 \end{pmatrix}$$

for $t > 0$. Then the result is

$$\dot{x} = \begin{pmatrix} 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & -3 & 0 \\ 6.54 & 0 & 5 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix} x + \begin{pmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{pmatrix} u \qquad (4.35)$$

$$y = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \end{pmatrix} x$$

The transfer matrix of (4.35) is

$$H(s) = \begin{pmatrix} \frac{5}{8.46s + s^3} & 0 \\ 0 & \frac{1}{s^2} \end{pmatrix}$$

We observe that (4.35) has no transmission zeros, and that the linearized model is completely decoupled (the nonlinear model is not !). We shall consider two controllers for this system. One in which no step of the structure algorithm is applied and one where one step is performed. The complete solution based on the structure algorithm requires 3 steps.

Case 1: O Step structure algorithm The application of the algorithm shows that A_1 is empty, i.e., the whole state is discontinuous and we might pose $z_2 = x$. To place all poles at -5 we apply the preliminary-feedback $u = K_n x$, where

$$K_n = \begin{pmatrix} -44.62 & 0 & -15 & -13.308 & 0 \\ 0 & -25 & 0 & 0 & -10 \end{pmatrix} .$$

The system-matrix after preliminary feedback $A^* = A + BK_n$ is

$$A^* = \begin{pmatrix} 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ -44.62 & 0 & -15 & -16.308 & 0 \\ 6.54 & 0 & 5 & 0 & 0 \\ 0 & -25 & 0 & 0 & -10 \end{pmatrix}$$

and the transfer-matrix is

$$H(s)^* = \begin{pmatrix} \dfrac{1}{25 + 15s + 3s^2 + 0.2s^3} & 0 \\ 0 & \dfrac{1}{25 + 10s + s^2} \end{pmatrix}.$$

Then restart the algorithm. We compute

$$V = \begin{pmatrix} -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \end{pmatrix}$$

and

$$Q = \begin{pmatrix} 0.7738 & 0 & 0 & 0 & 0.2777 \\ 0 & 0 & 0 & 1 & 0 \\ -1.0272 & 0 & 0 & 0 & 0.2092 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 \end{pmatrix}.$$

Further we have

$$A_1 = \text{empty}$$

$$N = \begin{pmatrix} 0 & -0.1242 & 0 & 0 & 0 \\ 0.7738 & 0 & 0 & 0 & 0.2777 \\ 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0.3460 & 0 & 0 & 0 \end{pmatrix}$$

$$J = \begin{pmatrix} -44.62 & 0 & -15 & -16.308 & 0 \\ 0 & -25 & 0 & 0 & -10 \end{pmatrix}$$

$$H = \begin{pmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

$$R_\xi(s) = \begin{pmatrix} 25 + 15s + 3s^2 + 0.2s^3 & 0 \\ 0 & 25 + 10s + s^2 \end{pmatrix}$$

Since As A_1 is empty $\overline{K}_J$ is empty also and it is easy to see that $R_\xi(s) = H(s)^*$.

Case 2: 1 Step structure algorithm In applying the first step of the stabilized structure algorithm with $\Upsilon_i = \begin{pmatrix} -5 & 0 \\ 0 & -5 \end{pmatrix}$ we obtain the output

$$\hat{y} = \begin{pmatrix} 5 & 0 & 0 & 1 & 0 \\ 0 & 5 & 0 & 0 & 1 \end{pmatrix} x$$

Clearly the transfer-function of the new system is $\widehat{H}(s) = W(s)H(s)$ where $W(s) = \begin{pmatrix} 5+s & 0 \\ 0 & 5+s \end{pmatrix}$ and the system has two transmission zeros at -5. The trajectories to be followed by this new system are

$$\hat{\xi}(t) = W\left(\frac{d}{dt}\right) y(t) \ .$$

Proceeding as before we find the same V as in Case 1, for

$$Q = \begin{pmatrix} -0.4739 & 0 & 0 & 0 & 0 \\ 0 & -0.5266 & 0 & 0 & 0 \\ -1.7498 & 0 & 1.0903 & 0 & -0.4108 \\ 2.3698 & 0 & -0.6619 & 0 & -0.6767 \\ 0 & 2.6332 & 0 & -1 & 0 \end{pmatrix} .$$

and for the system in z-coordinates

$$\dot{z} = \begin{pmatrix} -5 & 0 & 1.396 & 0 & 1.427 \\ 0 & -5 & 0 & 1.898 & 0 \\ -10.627819 & 0 & 2.097 & 0 & 5.247 \\ 0 & -13.166 & 0 & 5 & 0 \\ 10.395 & 0 & -5.216 & 0 & 2.902 \end{pmatrix} z + \begin{pmatrix} 0 & 0 \\ 0 & 0 \\ 0.6701 & 0 \\ 0 & -1 \\ -0.6554 & 0 \end{pmatrix} u \tag{4.36}$$

$$\hat{y} = \begin{pmatrix} 0 & 0 & -0.6619 & 0 & -0.6767 \\ 0 & 0 & 0 & -1 & 0 \end{pmatrix} z \ .$$

Sub-matrix

$$\overline{A}_4 = \begin{pmatrix} 2.097 & 0 & 5.247 \\ 0 & 5 & 0 \\ -5.216 & 0 & 2.902 \end{pmatrix}$$

is unstable since its eigenvalues are $\{5, 2.5 \pm 5.216\,i\}$. To stabilize it we apply the preleminary-feedback $v = \overline{K}_n z$, where

$$\overline{K}_n = \begin{pmatrix} 0 & 0 & -7.545 & 0 & 15.169 \\ 0 & 0 & 0 & -10 & 0 \end{pmatrix}$$

or in x-coordinates

$$K_n = \begin{pmatrix} -11.16 & 0 & -15 & -13.308 & 0 \\ 0 & -50 & 0 & 0 & -10 \end{pmatrix} ,$$

and start again. We compute

$$
\begin{aligned}
A_1 &= \begin{pmatrix} -5 & 0 \\ 0 & -5 \end{pmatrix} \\
N &= \begin{pmatrix} -0.0887 & 0 & -0.0907 \\ 0 & 0 & 0 \\ 0.0867 & 0 & 0.0887 \end{pmatrix} \\
J &= \begin{pmatrix} -11.16 & 0 & -15 & -16.308 & 0 \\ 0 & -50 & 0 & 0 & -10 \end{pmatrix} \\
H &= \begin{pmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix} \\
R_\xi(s) &= \begin{pmatrix} 5+2s+0.2s^2 & 0 \\ 0 & 5+s \end{pmatrix} \\
\overline{K}_J &= \begin{pmatrix} 15.858 & 0 & 0 & 0 & 0 \\ 0 & -13.166 & 0 & 0 & 0 \end{pmatrix} \\
K_J &= \begin{pmatrix} -33.46 & 0 & 0 & 0 & 0 \\ 0 & 25 & 0 & 0 & 0 \end{pmatrix} ,
\end{aligned}
\tag{4.37}
$$

the transformation-matrix Q is as above. Now the system in z–coordinates is

$$
\dot{z} = \begin{pmatrix} -5 & 0 & 1.396 & 0 & 1.427 \\ 0 & -5 & 0 & -1.898 & 0 \\ -10.627 & 0 & -2.959 & 0 & 15.413 \\ 0 & 13.166 & 0 & -5 & 0 \\ 10.395 & 0 & -0.2701 & 0 & -7.040 \end{pmatrix} z + \begin{pmatrix} 0 & 0 \\ 0 & 0 \\ 0.6701 & 0 \\ 0 & -1 \\ -0.6554 & 0 \end{pmatrix} u
\tag{4.38}
$$

$$
\hat{y} = \begin{pmatrix} 0 & 0 & -0.6619 & 0 & -0.6767 \\ 0 & 0 & 0 & -1 & 0 \end{pmatrix} z .
$$

and its transfer-matrix is

$$
H^*(s) = \begin{pmatrix} \dfrac{25+5s}{-42.3+75s+15s^2+s^3} & 0 \\ 0 & \dfrac{5+s}{50+10s+s^2} \end{pmatrix}
$$

and

$$
H^*(s)^{-1} = \begin{pmatrix} 5+2s+0.2s^2 & 0 \\ 0 & 5+s \end{pmatrix} + \frac{1}{25+10\,s+s^2} \begin{pmatrix} -33.46\,s-167.3 & 0 \\ 0 & 25\,s+125 \end{pmatrix}
$$

The application of $\overline{K}_J$ on (4.38) sets the sub-matrix

$$
\overline{A}_3 = \begin{pmatrix} -10.627 & 0 \\ 0 & -13.166 \\ 10.395 & 0 \end{pmatrix}
$$

to zero and since sub-matrix $\overline{C}_1 = 0$ the continuous part z_1 is rendered unobservable. The transfer-function $H_b(s)$ of the closed-loop system with preliminary feedback is

$$H_b^*(s) = \begin{pmatrix} \dfrac{1}{5+2s+0.2s^2} & 0 \\ 0 & \dfrac{1}{5+s} \end{pmatrix}$$

and it is easy to verify that $H_b^*(s)^{-1}$ is just the polynomial part of $H^*(s)^{-1}$ and that $R_\xi(s) = H_b^*(s)^{-1}$.

Here the application of either $u = \overline{K}_J z$ or $u = \overline{K}_n z$ to system (4.36) results in an unstable system. Only the application of $u = (\overline{K}_J + \overline{K}_n)z$ on (4.36) results in a system with all poles stable, i.e. the stabilization of the subsystem $(\overline{A}_4, \overline{B}_2)$ in (4.19b) is an essential part in the synthesis of feedback-control for tracking.

The resulting stable closed-loop system is

$$\dot{x} = \begin{pmatrix} 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ -44.62 & 0 & -15 & -16.308 & 0 \\ 6.54 & 0 & 5 & 0 & 0 \\ 0 & -25 & 0 & 0 & -10 \end{pmatrix} x + \begin{pmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{pmatrix} R(\tfrac{d}{dt})\xi(t)$$

$$y = \begin{pmatrix} 5 & 0 & 0 & 1 & 0 \\ 0 & 5 & 0 & 0 & 1 \end{pmatrix} x$$

5 HOC controller for linear system: discrete implementation

In the previous analysis, we considered the case ϵ and $\delta \to 0$. In this section we consider the case where ϵ and δ are not small.

Since the applied control by the HOC controller is piece-wise constant, we can study the exact discretization of (4.1).

$$\begin{aligned} x(t_{\rho+1}) &= e^{A\delta}x(t_\rho) + A^{-1}(e^{A\delta} - I)\,Bu_\rho && (5.1\text{a}) \\ y(t_\rho) &= C\,x(t_\rho) + D\,u_\rho && (5.1\text{b}) \\ x(t_0) &= x_0 && (5.1\text{c}) \end{aligned}$$

where $\rho = 0, 1, \ldots$, $t_{\rho+1} = \delta + t_\rho$ and u_ρ is the constant value of the control between t_ρ and $t_{\rho+1}$, computed using the following system of equations:

$$\begin{aligned} X_{j+1}^\rho - X_j^\rho &= \epsilon(AX_{j+1}^\rho + BU_{j+1}^\rho) \\ y(t_\rho + j\epsilon) &= CX_{j+1}^\rho + DU_{j+1}^\rho \\ X_j^\rho &= x(t_\rho)\ , \end{aligned} \qquad (5.2)$$

for each ρ and $u_\rho = U_k^\rho$ where k is such that $2\delta < k\epsilon < \delta$. For simplicity we drop the suffix ρ in the following computations. System (5.2) can be explicitly solved for U_k. We have

$$\begin{aligned}
X_{j+1} &= (I-\epsilon A)^{-1}\{X_j + \epsilon B U_{j+1}\} \\
\xi(t_p + j\epsilon) &= C X_{j+1} + D U_{j+1} \\
&= C(I-\epsilon A)^{-1} X_j + (\epsilon C(I-\epsilon A)^{-1} B + D) U_{j+1} \\
\Leftrightarrow U_{j+1} &= (\epsilon C(I-\epsilon A)^{-1} B + D)^{-1}\{\xi(t_p + j\epsilon) - C(I-\epsilon A)^{-1} X_j\} \\
\Leftrightarrow X_{j+1} &= (I-\epsilon A)^{-1}\{I - B[C(I-\epsilon A)^{-1} B + \tfrac{D}{\epsilon}]^{-1} C(I-\epsilon A)^{-1}\} X_j \\
&\quad + (I-\epsilon A)^{-1}\{B[C(I-\epsilon A)^{-1} B + \tfrac{D}{\epsilon}]^{-1}\}\xi(t_p + j\epsilon) .
\end{aligned}$$

By Reordering the last equations and by the introduction of new constants $(\Delta, \Omega, \Lambda)$ the expression for U_j and X_j can be expressed in a more condensed form:

$$\begin{aligned}
U_{j+1} &= \Delta\{\xi(t_p + j\epsilon) - C(I-\epsilon A)^{-1} X_j\} \\
X_{j+1} &= \Omega X_j + \Lambda\, \xi(t_p + j\epsilon)
\end{aligned}$$

which yields

$$\begin{aligned}
U_k &= \Delta\left\{ y(t_p + k\epsilon) - C(I-\epsilon A)^{-1}\left(\sum_{i=0}^{k-2} \Omega^{(k-i-2)} \Lambda\, y(t_p + (i+1)\epsilon) \right) \right\} \\
&\quad - \Delta\, C(I-\epsilon A)^{-1}\, \Omega^{(k-1)} X_0
\end{aligned} \tag{5.3}$$

where

$$\begin{aligned}
\Delta &= (\epsilon C(I-\epsilon A)^{-1} B + D)^{-1} \\
\Omega &= (I-\epsilon A)^{-1}\{I - B[C(I-\epsilon A)^{-1} B + \tfrac{D}{\epsilon}]^{-1} C(I-\epsilon A)^{-1}\} \\
\Lambda &= (I-\epsilon A)^{-1}\{B[C(I-\epsilon A)^{-1} B + \tfrac{D}{\epsilon}]^{-1}\} .
\end{aligned}$$

But $X_0 = x(t_p)$ and thus the feedback matrix is

$$\tilde{K}_J = \Delta\, C\,(I-\epsilon A)^{-1}\, \Omega^{(k-1)} . \tag{5.4}$$

If we plug back the control into (5.1) we see that the stability of this control scheme is equivalent to the stability of the following system:

$$x(t_p) = e^{A\delta} x(t_p) + A^{-1}(e^{A\delta} - I) B \tilde{K}_J\, x(t_{p-1}) \tag{5.5}$$

or equivalently the roots of

$$\det(z^2 I - e^{A\delta} z - A^{-1}(e^{A\delta} - I) B \tilde{K}_J) = 0 .$$

This formula can be used to get an idea on how large δ can be chosen without jeopardizing system stability. Analysis presented in this section shows that the HOC controller is really of predictive type. Notice that in (5.3) to compute u, k future values of the reference trajectory are used.

6 Example: Wheel rolling on a plane with linear preliminary feedback

The linear analysis has shown that preliminary feedback may make the HOC strategy applicable to the wheel example of Section 3 where we saw that HOC control without preliminary feedback does not work. Here we use as preliminary feedback what we found in Section 4.4. Figures 6.1 and 6.2 show the simulation results with zero steps of the structure algorithm (Figure 6.1 for small δ, Figure 6.2 for large δ; for $\delta > 0.2$ the error does not converge any longer). Figure 6.3 shows simulation results with one step of the structure algorithm on θ.

In each of the figures the first plot shows the lean angle θ as a solid line and the reference function $\theta_{ref}(t)$ as dashed line. Next to it we have the error $e_\theta(t) = \theta(t) - \theta_{ref}(t)$. As we see $e_\theta(t)$ converges to zero, i.e., the preliminary feedback stabilizes the HOC controller. The plots in the middle show the rolling angular velocity $\dot{\psi}$ (solid line), $\dot{\psi}_{ref}(t)$ (dashed line) and $e_{\dot{\psi}}(t) = \dot{\psi}(t) - \dot{\psi}_{ref}(t)$. On the bottom, we have the two control inputs: on the left side the "pedalling" torque, on the right, side the steering torque.

In all three simulations preliminary feedback stabilizes the HOC controller. Clearly, for small δ (see Figure 6.1) the result is the best since the estimation for u is more frequently updated as in Figure 6.2 where δ is set to its upper limit. The chattering in the steering torque u_1 of Figure 6.2 has the frequence of the sampling frequence and is due to the linear continuous preliminary feedback. Its amplitude grows with δ.

By doing one step of the structure algorithm on θ in Figure 6.3 we did not improve performance considerably. We observe again chattering (Note that $\delta = 0.05$).

7 Conclusion

In this report, we have presented a hybrid open-loop closed-loop strategy based on a control strategy introduced in [1]. In particular, we have shown that the controller in [1] can be applied to a much broader class of systems if it is modified by a preliminary feedback. We have done a complete analysis in the linear case and shown how such a preliminary feedback can be designed and how it can be applied to nonlinear systems.

We have only considered preliminary feedback that are of static state feedback type. If the state is only partially observable, it should be possible to design adequate dynamic preliminary feedbacks. It may be also interesting to study nonlinear preliminary feedbacks.

The result of the paper can trivially be generalized to the case where system dynamics is perturbated by a known disturbance functions.

There are many similarities between HOC control and predictive control which are under investigation.

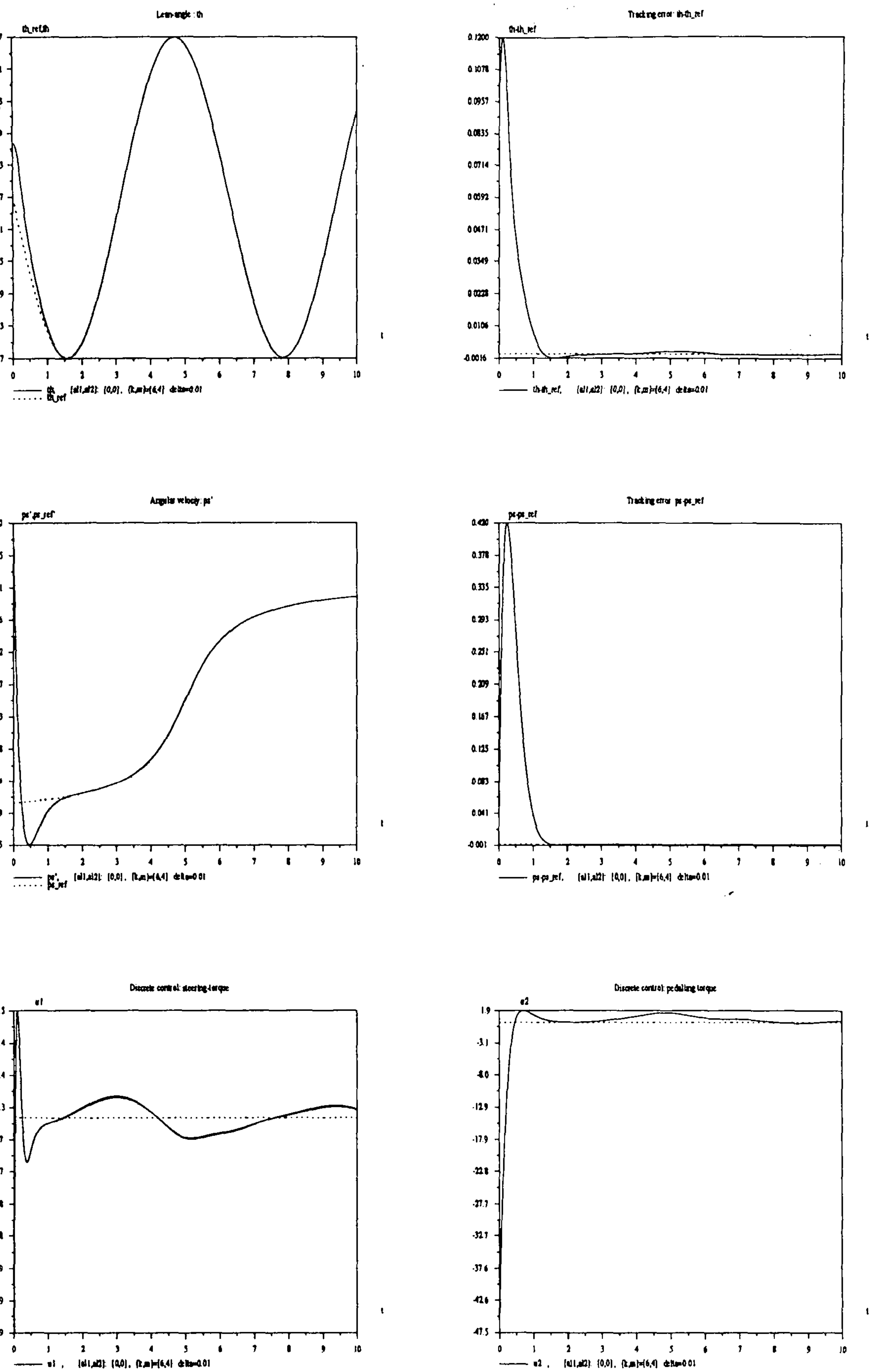

Figure 6.1 *Simulation of the controlled system with discrete controller using HOC controller with linear preliminary feedback. For* $\delta \to 0$ *we have perfect tracking performance.*

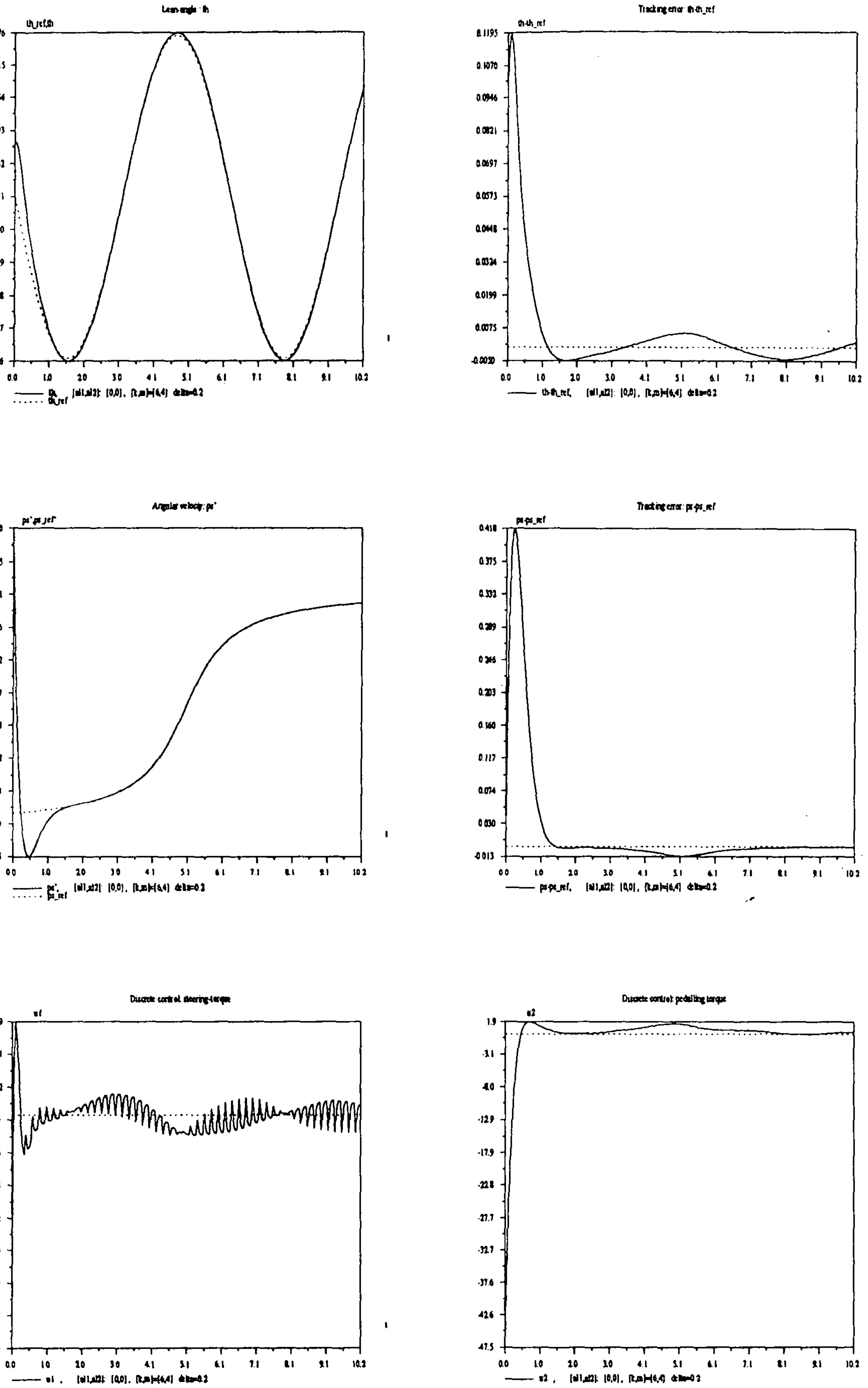

Figure 6.2 *Same simulation as in Figure 6.1, but large sampling-period. Obviously, tracking is worse than for small δ.*

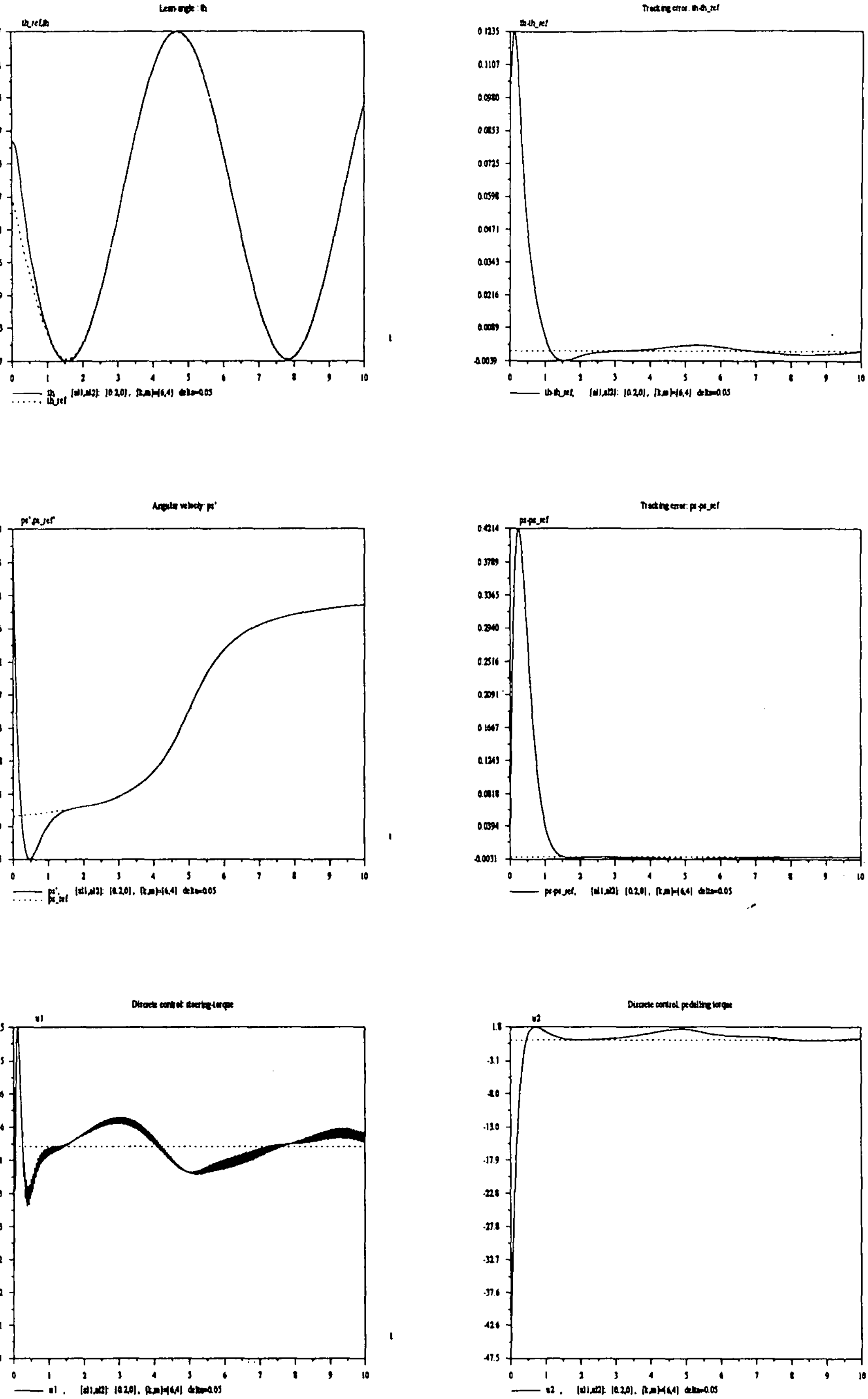

Figure 6.3 *Simulation of the controlled system with discrete controller using HOC controller with linear preliminary feedback and one step of the structure algorithm on θ (see Figure 3.4).*

A Model of the wheel rolling on a plane

The equations of motion can be obtained using the Lagrange method for nonholonomic systems [7]. Nonholonomic constraints result from the condition of rolling-without-slipping on the plane. If we note by $\Delta\mathbf{v}$ the difference between the velocities of the point of contact of the wheel with the ground and the point of contact with the wheel on the ground, then the condition rolling-without-sliding is written as

$$\Delta\mathbf{v} = \mathbf{v}_{(0)} + \omega_{(0)} \times \mathbf{r}_{(0)} = 0 \tag{A.1}$$

where $\omega_{(0)}$ is the angular velocity and $\mathbf{r}_{(0)}$ the radius-vector of the wheel in the inertial coordinate system (0) (see Figure 3.1). Denoting by the matrix A^z_ϕ a rotation of the angle ϕ around the z-axis and by $A^{x'}_\phi$ a rotation around the x'-axis after a rotation around the z-axis (that is why we write x' and not x) from the inertial coordinate system (0) to the rotated and moving coordinate system (x', y', z'), we find the following local representations for $\omega_{(0)}$ and $\mathbf{r}_{(0)}$:

$$\omega_{(0)} = \begin{pmatrix} 0 \\ 0 \\ \dot\phi \end{pmatrix} + (A^z_\phi)^T \left[\begin{pmatrix} \dot\theta \\ 0 \\ 0 \end{pmatrix} + (A^{x'}_\theta)^T \begin{pmatrix} 0 \\ 0 \\ \dot\psi \end{pmatrix} \right] = \begin{pmatrix} \dot\theta\cos(\phi) + \dot\psi\sin\theta\sin(\phi) \\ \dot\theta\sin(\phi) - \dot\psi\sin(\theta)\cos(\phi) \\ \dot\phi + \dot\psi\cos(\theta) \end{pmatrix},$$

$$\mathbf{r}_{(0)} = (A^z_\phi)^T (A^{x'}_\theta)^T \begin{pmatrix} 0 \\ -r \\ 0 \end{pmatrix} = \begin{pmatrix} r\cos(\theta)\sin(\phi) \\ -r\cos(\theta)\cos(\phi) \\ -r\sin(\theta) \end{pmatrix},$$

where $\mathbf{A}^z_\phi$ and $\mathbf{A}^{x'}_\theta$ are

$$\mathbf{A}^z_\phi = \begin{pmatrix} \cos(\phi) & \sin(\phi) & 0 \\ -\sin(\phi) & \cos(\phi) & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad \text{and} \quad \mathbf{A}^{x'}_\theta = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos(\theta) & \sin(\theta) \\ 0 & -\sin(\theta) & \cos(\theta) \end{pmatrix}.$$

Then the nonholonomic constraints become

$$\Delta\mathbf{v} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} = \begin{bmatrix} \dot x + \dot\phi\, r\cos(\theta)\cos(\phi) - \dot\theta\, r\sin(\phi)\sin(\theta) + \dot\psi\, r\cos(\phi) \\ \dot y + \dot\phi\, r\cos(\theta)\sin(\phi) + \dot\theta\, r\cos(\phi)\sin(\theta) + \dot\psi\, r\sin(\phi) \end{bmatrix}. \tag{A.2}$$

To write the Lagrangian we have to compute the difference between the kinetic energy T and the potential energy U. Clearly, as the moving coordinate system is attached to the center of mass of the wheel we have for the kinetic energy

$$T = \frac{1}{2}\omega^T_{(1)} \begin{bmatrix} I_n & 0 & 0 \\ 0 & I_n & 0 \\ 0 & 0 & I_r \end{bmatrix} \omega_{(1)}$$

where $\omega_{(1)}$ represents the angular velocity of the wheel in the moving coordinates (1).

$$\omega_{(1)} = \begin{bmatrix} 0 \\ 0 \\ \dot\psi \end{bmatrix} + A^z_\psi \left[\begin{pmatrix} \dot\theta \\ 0 \\ 0 \end{pmatrix} + A^x_\theta \begin{pmatrix} 0 \\ 0 \\ \dot\phi \end{pmatrix} \right]$$

or equivalently

$$\omega_{(1)} = \begin{pmatrix} \dot\theta\cos(\psi) + \dot\phi\sin(\theta)\sin(\psi) \\ -\dot\theta\sin(\psi) + \dot\phi\sin(\theta)\cos(\psi) \\ \dot\phi\cos(\theta) + \dot\psi \end{pmatrix}.$$

The potential energy U is proportional to the hight of the center of mass of the wheel: $U = zg$. Then the Lagrangian is simply $L = T - U$.

$$L = \frac{1}{2}\left(I_r\left(\dot{\theta}^2 + \dot{\phi}^2\sin(\theta)^2\right) + I_n\left(\dot{\phi}\cos(\theta) + \dot{\psi}\right)^2 + m\left(\dot{x}^2 + \dot{y}^2 + (\dot{r\theta cos(\theta)})^2\right)\right) - m\,g\,r\,\sin(\theta)\ .$$

where we have substituted z by $r\sin(\theta)$. In $\mathbf{q} = [x, y, \phi, \theta, \psi]$ the equations of motion are given by

$$\frac{\partial L(\mathbf{q},\dot{\mathbf{q}})}{\partial \mathbf{q}} - \frac{d}{dt}\frac{\partial L(\mathbf{q},\dot{\mathbf{q}})}{\partial \dot{\mathbf{q}}} + \mathbf{G}^T(\mathbf{q})\lambda = 0\ , \tag{A.3}$$

$$\mathbf{G}(\mathbf{q})\dot{\mathbf{q}} = 0\ , \tag{A.4}$$

where $\mathbf{G}(\mathbf{q}) = \frac{\partial \Delta \mathbf{V}}{\partial \mathbf{q}}$ and $\lambda = [\lambda_1, \lambda_2]$ is the vector of Lagrange multipliers. The result is

$$\begin{bmatrix} m & 0 & 0 & 0 & 0 \\ 0 & m & 0 & 0 & 0 \\ 0 & 0 & I_r\,\sin(\theta)^2 + I_n\,\cos(\theta)^2 & 0 & I_n\cos(\theta) \\ 0 & 0 & 0 & m\,r^2\,\cos(\theta)^2 + I_r & 0 \\ 0 & 0 & I_n\,\cos(\theta) & 0 & I_n \end{bmatrix} \begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{\phi} \\ \ddot{\theta} \\ \ddot{\psi} \end{bmatrix} =$$

$$\begin{bmatrix} 0 \\ 0 \\ 2\,(I_n - I_r)\,\dot{\phi}\,\dot{\theta}\,\sin(\theta)\cos(\theta) + I_n\dot{\theta}\dot{\psi}\,\sin(\theta) \\ \{(I_r - I_n)\,\dot{\phi}^2 + m\,r^2\,\dot{\theta}^2\}\sin(\theta)\cos(\theta) - I_n\,\dot{\phi}\,\dot{\psi}\,\sin(\theta) - m\,g\,r\,\cos(\theta) \\ I_n\,\dot{\phi}\,\dot{\theta}\,\sin(\theta) \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ u_1\sin(\theta) \\ 0 \\ u_2 \end{bmatrix} +$$

$$\begin{bmatrix} 1 & 0 \\ 0 & 1 \\ r\,\cos(\theta)\cos(\phi) & r\,\cos(\theta)\sin(\phi) \\ -r\,\sin(\phi)\sin(\theta) & r\,\cos(\phi)\sin(\theta) \\ r\,\cos(\phi) & r\,\sin(\phi) \end{bmatrix} \begin{bmatrix} \lambda_1 \\ \lambda_2 \end{bmatrix}\ .$$

Since the Lagrangian is independent of the position (x, y) of the center of mass of the wheel, we can find a more condensed form of the equations of motion, in particular the nonholonomic-constraints can be eliminated using a modified version of the Euler-Lagrange-equations (A.3) exploiting (A.2).

Obviously, the constraints (A.2) can be rewritten in the form

$$\begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix} = \mathbf{Q}(\phi, \theta) \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix}$$

or equivalently

$$\begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix} = \begin{bmatrix} -r\left(\dot{\phi}\,\cos(\theta)\cos(\phi) - \dot{\theta}\,\sin(\phi)\sin(\theta) + \dot{\psi}\,\cos(\phi)\right) \\ -r\left(\dot{\phi}\,\cos(\theta)\sin(\phi) + \dot{\theta}\,\cos(\phi)\sin(\theta) + \dot{\psi}\,\sin(\phi)\right) \end{bmatrix}\ .$$

As the Lagrangian is independent of the position of the center of mass $\mathbf{q}_2 = [x, y]$ and depends only on its derivatives $\dot{\mathbf{q}}_2 = [\dot{x}, \dot{y}]$ and the angles $\mathbf{q}_1 = [\phi, \theta, \psi]$ and angular velocities $\dot{\mathbf{q}}_1 = [\dot{\phi}, \dot{\theta}, \dot{\psi}]$, the solution of the DAE ((A.3),(A.4)) is equivalent to that of to

$$\frac{\partial L(\mathbf{q}_1, \dot{\mathbf{q}}_1, \dot{\mathbf{q}}_2)}{\partial \mathbf{q}_1} - \frac{d}{dt}\frac{\partial L(\mathbf{q}_1, \dot{\mathbf{q}}_1, \dot{\mathbf{q}}_2)}{\partial \dot{\mathbf{q}}_1} - \frac{d}{dt}\frac{\partial L(\mathbf{q}_1, \dot{\mathbf{q}}_1, \dot{\mathbf{q}}_2)}{\partial \dot{\mathbf{q}}_2}\mathbf{Q}(\mathbf{q}_1) = 0 \ , \tag{A.5}$$

where

$$\begin{aligned} \mathbf{q}_2 &= \mathbf{Q}(\mathbf{q}_1)\dot{\mathbf{q}}_1 \ , \\ \dot{\mathbf{q}}_2 &= \frac{\partial \mathbf{Q}(\mathbf{q}_1)\dot{\mathbf{q}}_1}{\partial \mathbf{q}_1} + \dot{\mathbf{q}}_1 \mathbf{Q}(\mathbf{q}_1)\ddot{\mathbf{q}}_1 \ . \end{aligned}$$

For a detailed explanation see [8] or the appendix of [9]. Now the equations of motion are composed, first, of a set of equations independent of the position $\mathbf{q}_2 = [x, y]$ and the absolute orientation ϕ of the wheel on the plane

$$M(\theta)\begin{bmatrix} \ddot{\phi} \\ \ddot{\theta} \\ \ddot{\psi} \end{bmatrix} = F(\theta, \dot{\phi}, \dot{\theta}, \dot{\psi}) + B(\theta)\begin{bmatrix} u_1 \\ u_2 \end{bmatrix} \tag{A.6}$$

or specifically

$$\begin{bmatrix} I_r \sin(\theta)^2 + I_n \cos(\theta)^2 + m\, r^2 \cos(\theta)^2 & 0 & I_n \cos(\theta) + m\, r^2 \cos(\theta) \\ 0 & I_r + m\, r^2 & 0 \\ I_n \cos(\theta) + m\, r^2 \cos(\theta) & 0 & I_n + m\, r^2 \end{bmatrix}\begin{bmatrix} \ddot{\phi} \\ \ddot{\theta} \\ \ddot{\psi} \end{bmatrix} =$$

$$\begin{bmatrix} I_n \dot{\theta}\, \dot{\psi} \sin(\theta) + 2\, (m\, r^2 + I_n - I_r)\, \dot{\phi}\, \dot{\theta} \cos(\theta)\sin(\theta) \\ -(m\, r^2 + I_n)\, \dot{\phi}\, \dot{\psi} \sin(\theta) - (m\, r^2 + I_n - I_r)\, \dot{\phi}^2 \cos(\theta)\sin(\theta) - m\, g\, r \cos(\theta) \\ (2\, m\, r^2 + I_n)\, \dot{\phi}\, \dot{\theta} \sin(\theta) \end{bmatrix} + \begin{bmatrix} u_1 \sin(\theta) \\ 0 \\ u_2 \end{bmatrix}$$

(Note that the matrix at the left side of the equation is nonsingular for all values of θ except $\theta = k\pi$, $k = ..-1, 0, 1, 2..$) and, second, of two separate equations which can be used to compute the position of the center of mass $\mathbf{q}_2 = [x, y]$ on the plan.

$$\begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix} = \mathbf{Q}(\phi, \theta)\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} .$$

References

[1] Jankowski, K.,P., and Van Brussel, H., An Approach to Descete Inverse Dynamics Control of Flexible-Joint Robots, *IEEE Transaction on Robotics and Automation*, Vol.8, No.5, October 1992, pp. 651-658.

[2] Isidori, A., Nonlinear Control Systems: An Introduction, *Springer Berlin* 1989.

[3] Silverman, L., M., Inversion of Multivariable Linear Systems, *IEEE Transactions on Automatic Control*, Vol. Ac-14, No. 3, June 1969.

[4] Campbell,S., L.,Nikoukhah, R. and von Wissel, D., *Numerically Generated Paht Stabilizing Controllers I: Theoritical Concerns* Proc. of Automatic Control Conf. June 30,1994 (to appear).

[5] Brenan, K.,E., Campbell, S.,L. and Petzold, L.,R., Numerical Solution of Inital-Value Problems in Differential-Algebraic Equations, *North-Holland*, 1989

[6] Wonham, W., *Linear multivariable Control: a geometric approach*, Springer Verlag, 1972.

[7] Lanczos, C., The variational principles of mechanics *Mathematical Expositions, No.4, University of Toronto press*, 1962.

[8] Neimark, Ju., I. and Fufaev, F.,A., Dynamics of nonholonomic systems, *A.M.S. Translations of Mathematical Monographs*, Providence, RI, AMS, 1972.

[9] von Wissel, D., Nikoukhah, R. and Delebecque, F., Maneuver-Based Obstacle-Avoiding Trajectory Optimization: Example of a Riderless Bicycle, submitted to *Special issue of Mechanics of Structures and Machines: Modelling and Analysis of Mechanical Systems and Algorithms for Real-Time Vehicle Simulation* .

[10] Delebecque, F. and Nikoukhah, R., A mixed symbolic-numeric software environment and its application to control system engineering, *Recent advances in computer aided control*, eds. Hergert, H. and Jamshidi, M., pp. 221-245, 1992.

Unité de Recherche INRIA Rocquencourt
Domaine de Voluceau - Rocquencourt - B.P. 105 - 78153 LE CHESNAY Cedex (France)

Unité de Recherche INRIA Lorraine Technopôle de Nancy-Brabois - Campus Scientifique
615, rue du Jardin Botanique - B.P. 101 - 54602 VILLERS LES NANCY Cedex (France)
Unité de Recherche INRIA Rennes IRISA, Campus Universitaire de Beaulieu 35042 RENNES Cedex (France)
Unité de Recherche INRIA Rhône-Alpes 46, avenue Félix Viallet - 38031 GRENOBLE Cedex (France)
Unité de Recherche INRIA Sophia Antipolis 2004, route des Lucioles - B.P. 93 - 06902 SOPHIA ANTIPOLIS Cedex (France)

EDITEUR
INRIA - Domaine de Voluceau - Rocquencourt - B.P. 105 - 78153 LE CHESNAY Cedex (France)

ISSN 0249 - 6399

RR_2173