SINGULARITY ROBUST, DYNAMIC LINEARIZATION CONTROL ALGORITHM FOR MK MOBILE ROBOT

Marek Kabała, Robert Muszyński, Marek Wnuk

Institute of Engineering Cybernetics,
Wrocław University of Technology,
Janiszewskiego 11/17,
50–372 Wrocław, Poland
email: `mkabala|mucha|mw@ict.pwr.wroc.pl`

Abstract: MK robot is a two wheel nonholonomic cart with driving moments referenced to a pendulum (the robot body). A modification of the described in (Tchoń *et al.*, 2002) dynamic linearization control algorithm for MK robot is proposed, which provides singularity robust version of this algorithm. The simulation results show good performance of the modified algorithm even for trajectories which do not fit dynamic linearization algorithm restrictions.

Keywords: mobile robots, dynamics, feedback linearization, robustness

1. INTRODUCTION

Development of control algorithms for nonholonomic mobile robots, in which it is structurally impossible to separate the kinematics from the dynamics, is one of the significant problems in contemporary robotics. A simple example of such an object is a two wheel cart with no supporting points. The cart body forms a pendulum, suspended on the main axis, connecting two parallel wheels. The moments driving the wheels are related to the pendulum (the cart body). Such a cart, called MK robot, described in (Kabała and Wnuk, 2002*b*), has been built as a preliminary project for RoBall, a spherical mobile robot, being currently under development in our Robotics Laboratory, (Kabała and Wnuk, 2002*a*).

Both the dynamics model and dynamic linearization control algorithm for the MK robot are described in (Tchoń *et al.*, 2002). Dynamic linearization technique is employed for synthesis of trajectory tracking algorithm. This model-based method consists in extending the dynamics of the object in order to enable transforming the input-output mapping into a linear, decoupled form. The transformation is performed by means of dynamic feedback. Unfortunately, the decoupling matrix becomes singular when the robot stalls, what restricts the applicability of the algorithm to trajectories with no stop points. Obviously, starting the robot motion also requires some other control algorithm.

In the paper a modification of the dynamic linearization is proposed, which provides singularity robust version of this algorithm. It consists in using the singularity robust pseudoinverse, (Nakamura, 1991), in order to avoid singularities of the decoupling transformation.

The simulation results show good behaviour of the modified algorithm even for trajectories which do not fit dynamic linearization algorithm restrictions.

[1] This research has been done within a statutory research project. Computer simulations have been done in MATLAB® environment provided by Wrocław Centre for Networking and Supercomputing.

2. DYNAMICS MODEL OF MK ROBOT

MK robot is a two wheel cart with driving moments referenced to a pendulum (the robot body) swinging on the robot axle (see figure 1). The robot wheels are

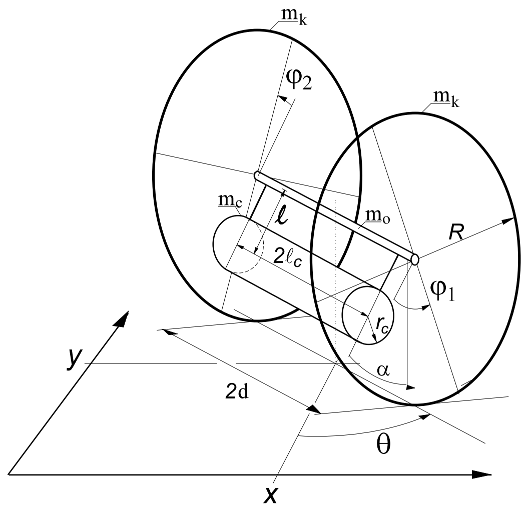

Fig. 1. MK robot coordinates and parameters.

assumed to be uniform rings of radius R and mass m_k. The axle is considered to be an infinitely thin wire of mass m_o and length $2d$. The robot body is modelled by a uniform cylinder of mass m_c, radius r_c and length $2l_c$, with its axis parallelly displaced by l according to the robot axle.

In state coordinates $(x, y, \varphi_1, \varphi_2, \alpha, \eta_1, \eta_2, \eta_3)$, with auxiliary longitudinal velocity of the axle center $\eta_{123} = \eta_1 + \eta_2 + \eta_3$, orientation $\theta = \frac{R}{2d}(\varphi_2 - \varphi_1)$, taking the torques driving the cart wheels as the control inputs u_1, u_2, and assuming absence of slipping, one obtains kinematics and dynamics model of the MK robot:

$$\begin{aligned} &\dot{x} = \eta_{123}\cos\theta,\ \dot{y} = \eta_{123}\sin\theta, \\ &\dot{\varphi}_1 = \frac{2}{R}\eta_1, \quad \dot{\varphi}_2 = \frac{2}{R}\eta_2, \quad \dot{\alpha} = \frac{1}{R}\eta_3. \end{aligned} \tag{1}$$

$$M(\alpha)\dot{\eta} + N(\alpha, \eta) = Bu, \tag{2}$$

with the matrix of inertia:

$$M(\alpha) = M = \begin{bmatrix} \left(m_c + \frac{4}{3}m_o + 8m_k\right)d^2 + m_k R^2 + m_c\left(\ell^2\sin^2\alpha + \frac{1}{3}\ell_c^2 + \frac{1}{2}r_c^2\right) & \left(m_c + \frac{2}{3}m_o\right)d^2 - m_k R^2 - m_c\left(\ell^2\sin^2\alpha + \frac{1}{3}\ell_c^2 + \frac{1}{2}r_c^2\right) & \left(m_c + m_o + 4m_k + \frac{m_c\ell}{R}\cos\alpha\right)d^2 \\ \left(m_c + \frac{2}{3}m_o\right)d^2 - m_k R^2 - m_c\left(\ell^2\sin^2\alpha + \frac{1}{3}\ell_c^2 + \frac{1}{2}r_c^2\right) & \left(m_c + \frac{4}{3}m_o + 8m_k\right)d^2 + m_k R^2 + m_c\left(\ell^2\sin^2\alpha + \frac{1}{3}\ell_c^2 + \frac{1}{2}r_c^2\right) & \left(m_c + m_o + 4m_k + \frac{m_c\ell}{R}\cos\alpha\right)d^2 \\ \left(m_c + m_o + 4m_k + \frac{m_c\ell}{R}\cos\alpha\right)d^2 & \left(m_c + m_o + 4m_k + \frac{m_c\ell}{R}\cos\alpha\right)d^2 & \left(m_c + m_o + 4m_k + \frac{2m_c\ell}{R}\cos\alpha + \frac{m_c}{R^2}\left(\ell^2 + \frac{1}{2}r_c^2\right)\right)d^2 \end{bmatrix},$$

the vector of Coriolis, centrifugal, and viscous friction moments

$$N(\alpha, \eta) = N = \begin{pmatrix} -m_c\ell\sin\alpha(2\eta_2 + \eta_3)(\eta_2 - \eta_1) - \frac{2m_c\ell^2}{R}\sin\alpha\cos\alpha\eta_3(\eta_2 - \eta_1) - \frac{m_c d^2\ell\sin\alpha}{R^2}\eta_3^2 + \frac{4d^2k_1}{R^2}\eta_1 \\ m_c\ell\sin\alpha(2\eta_1 + \eta_3)(\eta_2 - \eta_1) + \frac{2m_c\ell^2}{R}\sin\alpha\cos\alpha\eta_3(\eta_2 - \eta_1) - \frac{m_c d^2\ell\sin\alpha}{R^2}\eta_3^2 + \frac{4d^2k_2}{R^2}\eta_2 \\ -\frac{m_c\ell\sin\alpha}{R}(\eta_2 - \eta_1)^2(R + \ell\cos\alpha) - \frac{m_c\ell d^2}{R^2}\sin\alpha\eta_3^2 + \frac{m_c g\ell d^2}{R}\sin\alpha \end{pmatrix},$$

and the matrix of controls

$$B = \frac{2d^2}{R}\begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix}.$$

3. CONTROL ALGORITHM

Considering the problem of tracking a given trajectory $(x_d(t), y_d(t))$, we choose linearization outputs

$$\begin{cases} y_1 = x, \\ y_2 = y \end{cases} \tag{3}$$

and search a decoupling feedback for the equations (1)-(2) by successive differentiation of them.

By introducing a new state coordinate

$$q = PM^{-1}Bu,$$

and defining new inputs

$$\dot{q} = v_1, \qquad QM^{-1}Bu = v_2, \tag{4}$$

such that

$$\begin{bmatrix} PM^{-1}B \\ QM^{-1}B \end{bmatrix} u = \begin{pmatrix} q \\ v_2 \end{pmatrix} \quad \Rightarrow \quad u = R\begin{pmatrix} q \\ v_2 \end{pmatrix}$$

$$R = R(\alpha) = \begin{bmatrix} PM^{-1}B \\ QM^{-1}B \end{bmatrix}^{-1},$$

where

$$P = \begin{bmatrix} 1 & 1 & 1 \end{bmatrix}, \qquad Q = \frac{1}{d}\begin{bmatrix} -1 & 1 & 0 \end{bmatrix},$$

we transform the equations (3) to

$$\begin{aligned} &\begin{pmatrix} y_1^{(3)} \\ y_2^{(3)} \end{pmatrix} = g(\alpha, \eta, \theta, q) + \\ &\begin{bmatrix} \cos\theta & -\dot{y}_2 - h(\alpha, \eta)\cos\theta \\ \sin\theta & \dot{y}_1 - h(\alpha, \eta)\sin\theta \end{bmatrix}\begin{pmatrix} v_1 \\ v_2 \end{pmatrix} = g + Dv. \end{aligned} \tag{5}$$

The obtained decoupling matrix $D(\alpha, \eta, \theta)$ is non-singular as long as the robot is moving, i.e.

$$\dot{y}_1\cos\theta + \dot{y}_2\sin\theta \neq 0.$$

Under the above assumption, feedback of the form

$$w = g + Dv \quad \Rightarrow \quad v = -D^{-1}g + D^{-1}w \tag{6}$$

reduces (1)-(2) extended by (4) to a decoupled input-output system

$$\begin{pmatrix} y_1^{(3)} \\ y_2^{(3)} \end{pmatrix} = \begin{pmatrix} w_1 \\ w_2 \end{pmatrix}. \tag{7}$$

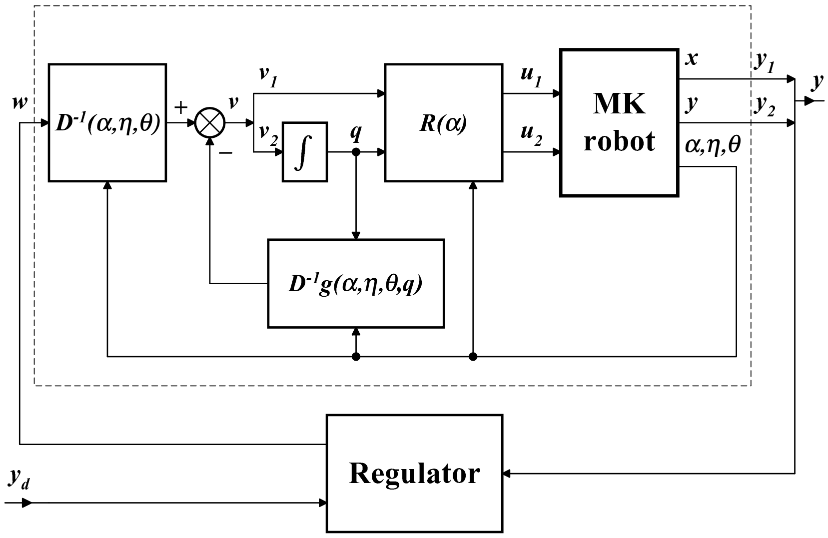

Fig. 2. Block diagram of the resulting control algorithm

For tracking $(y_{1d}(t), y_{2d}(t)) = (x_d(t), y_d(t))$ it suffices to apply a linear regulator

$$\begin{cases} w_1 = y_{1d}^{(3)} - k_{12}(\ddot{y}_1 - \ddot{y}_{1d}) - k_{11}(\dot{y}_1 - \dot{y}_{1d}) - k_{10}(y_1 - y_{1d}) \\ w_2 = y_{2d}^{(3)} - k_{22}(\ddot{y}_2 - \ddot{y}_{2d}) - k_{21}(\dot{y}_2 - \dot{y}_{2d}) - k_{20}(y_2 - y_{2d}) \end{cases} \quad (8)$$

with appropriate gains, providing asymptotic stability of the regulation error ($e_1 = y_1 - y_{1d},\ e_2 = y_2 - y_{2d}$) equations

$$\begin{cases} e_1^{(3)} + k_{12}\ddot{e}_1 + k_{11}\dot{e}_1 + k_{10}e_1 = 0 \\ e_2^{(3)} + k_{22}\ddot{e}_2 + k_{21}\dot{e}_2 + k_{20}e_2 = 0, \end{cases}.$$

The block diagram of the resulting control algorithm is presented in figure 2.

4. ALGORITHM MODIFICATION

One of the major drawbacks of the described above algorithm is that the decoupling matrix D becomes singular when the robot does not move. It restricts its use to tracking specific trajectories with specific initial conditions. In practical implementations, the restrictions are even stronger, due to the limited precision of calculations. To ensure proper tracking, the velocity along the trajectory cannot fall below a certain limit.

The proposed modification of the algorithm is based upon the singularity robust pseudoinverse, (Nakamura, 1991). Replacing in equation (6) the matrix D^{-1} by

$$D^\star = D^T\left(DD^T + diag(\varepsilon_1, \varepsilon_2)\right)^{-1},$$

where $\varepsilon_1, \varepsilon_2$ are small positive constants, gives the approximation of decoupling feedback

$$v = -D^\star g + D^\star w. \quad (9)$$

The approximation (9) converges to (6) when $\varepsilon_1, \varepsilon_2$ decrease to zero.

5. SIMULATION RESULTS

To illustrate the performance of the controller, based on the modified version of dynamic linearization, we present some simulation results for both the original algorithm and its new, singularity robust version. The simulations have been performed in the MATLAB® environment.

In all the simulations we have assumed robot model parameters corresponding to the real cart parameters (Tchoń *et al.*, 2002): $R = 0.254[m]$, $d = 0.225[m]$, $l = 0.12[m]$, $l_c = 0.165[m]$, $r_c = 0.035[m]$, $m_c = 7[kg]$, $m_o = 0.5[kg]$, $m_k = 2[kg]$, $k_1 = k_2 = 0.05\left[\frac{kgm^2}{s}\right]$. The regulator gains have been set as follows: $k_{10} = k_{20} = 7$, $k_{11} = k_{21} = 20$, $k_{12} = k_{22} = 20$.

The selected trajectories contain:

wide ellipse:

$$\begin{cases} x_d(t) = \frac{1}{20}\cos\left(\frac{\pi}{12}t\right) \\ y_d(t) = \sin\left(\frac{\pi}{12}t\right), \end{cases}$$

narrow ellipse:

$$\begin{cases} x_d(t) = \frac{1}{60}\cos\left(\frac{\pi}{12}t\right) \\ y_d(t) = \sin\left(\frac{\pi}{12}t\right), \end{cases}$$

straight line segment:

$$\begin{cases} x_d(t) = 0 \\ y_d(t) = \sin\left(\frac{\pi}{12}t\right). \end{cases}$$

We have assumed a common initial state ($x = 0$, $y = 0$, $\varphi_1 = -1$, $\varphi_2 = 1$, $\alpha = 0$, $\eta_1 = 0.05$, $\eta_2 = 0.05$, $\eta_3 = 0$, $q = 0$) and simulation time $25[s]$. The results for the singularity robust algorithm have been obtained with $\varepsilon_1 = \varepsilon_2 = 0.01$.

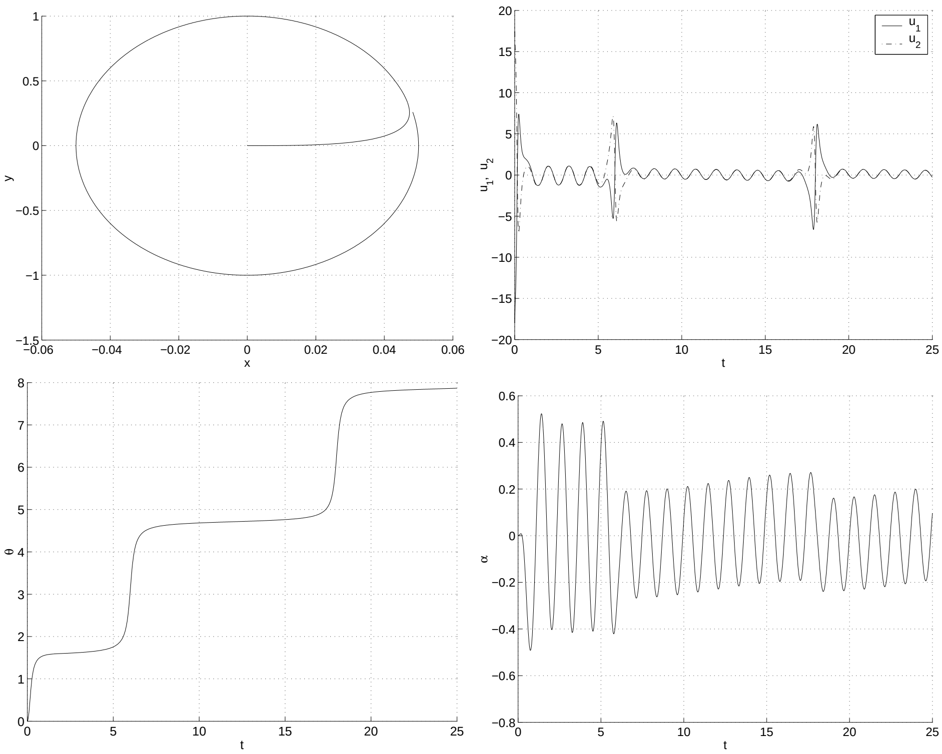

Fig. 3. Simulation results for the elliptical trajectory with the original algorithm.

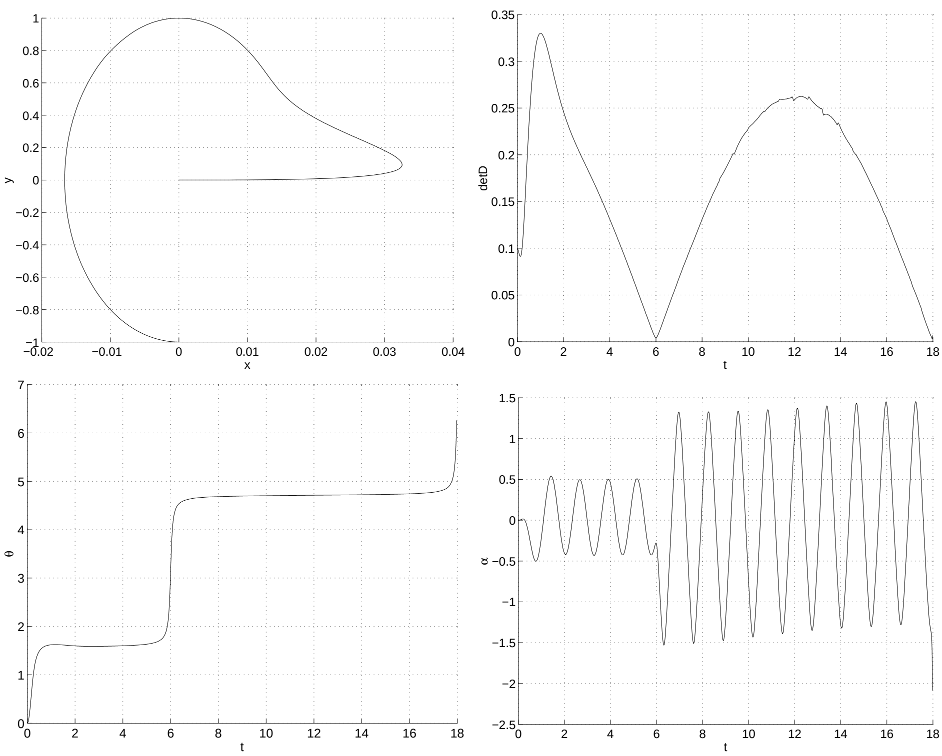

Fig. 4. Simulation results for the narrow elliptical trajectory with the original algorithm.

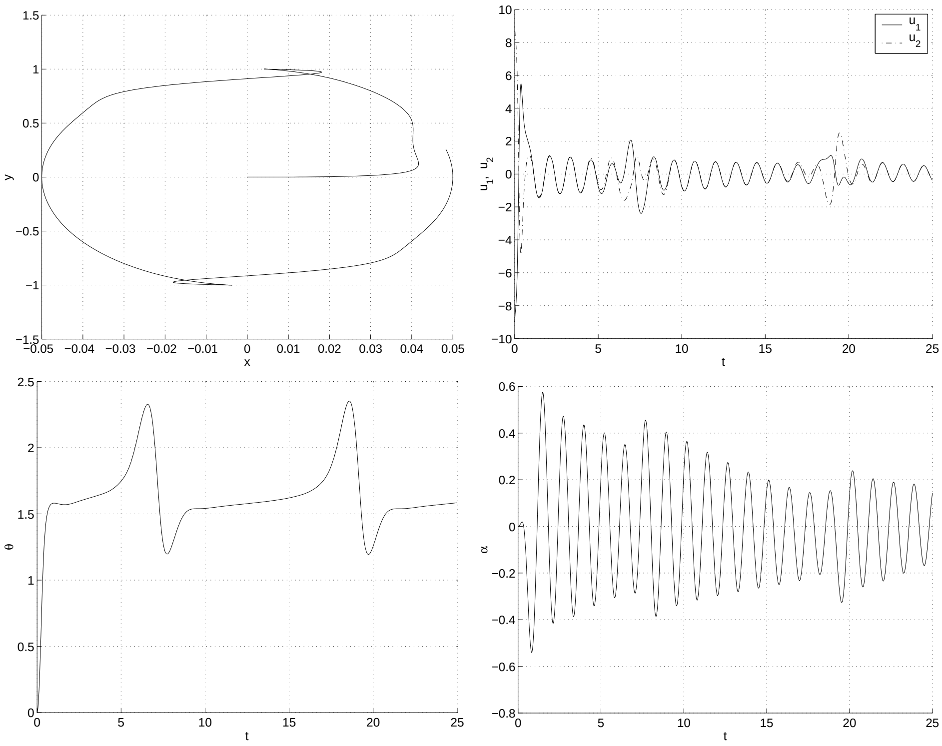

Fig. 5. Simulation results for the elliptical trajectory with the singularity robust algorithm.

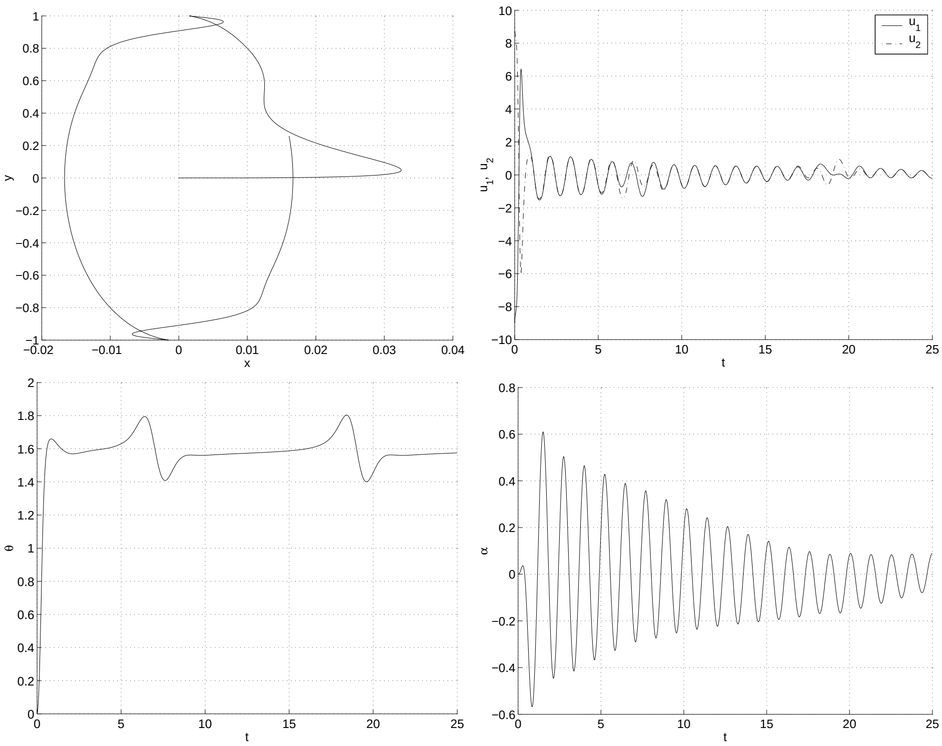

Fig. 6. Simulation results for the narrow elliptical trajectory with the singularity robust algorithm.

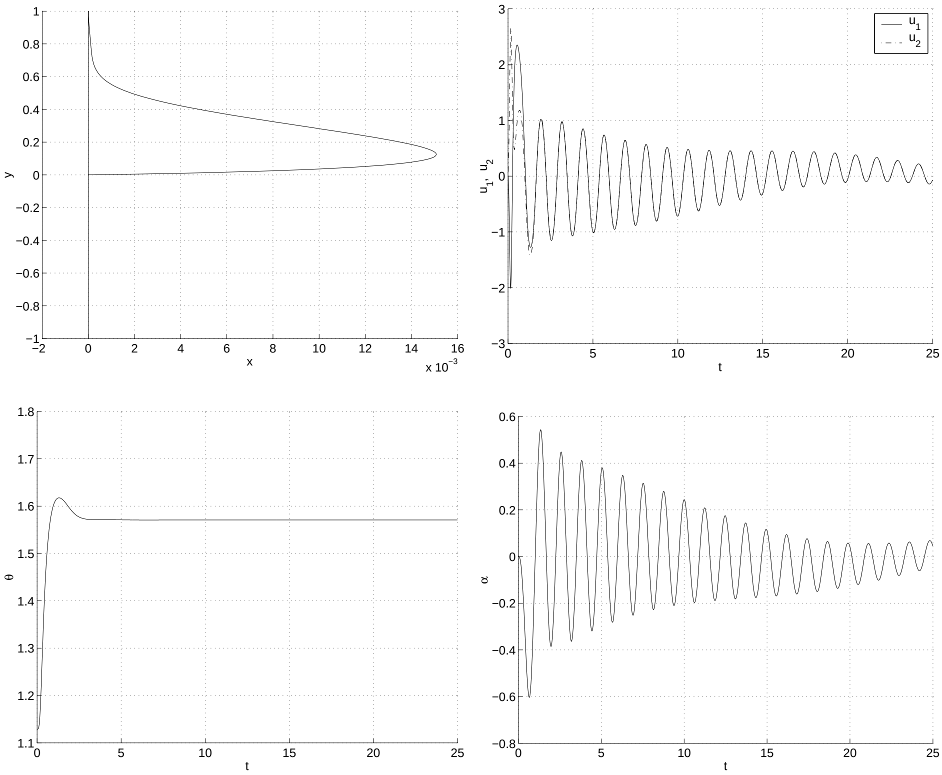

Fig. 7. Simulation results for the straight line trajectory with the singularity robust algorithm.

The results of wide ellipse tracking with the original algorithm are presented in figure 3. Narrowing the ellipse leads to the failure of the original algorithm (see figure 4), caused by a singularity of the decoupling matrix D. Actually, in the case of elliptical trajectory, the condition $\det D = 0$ is caused by numeric simulation errors. Obviously, any discrete implementation of the algorithm would behave similarly.

The results for both the ellipses obtained with the presented singularity robust algorithm are shown in figures 5 and 6. The controller provides good trajectory tracking not only in the cases allowed for the original version, but also in such cases in which the original algorithm fails. The calculated control signals (u_1, u_2) are in general smaller for the singularity robust dynamic linearization than for the original algorithm. Another positive side effect is that the pendulum swing (α) oscillations are damped. One can notice a significant difference in behaviour of both algorithms on the trajectory turnarounds. The singularity robust algorithm tends to reverse the direction of wheel movement (φ_1, φ_2), while the original one causes monotonic change of the cart orientation θ.

It is worth stressing, that the presented singularity robust algorithm allows robot starting and stopping along the trajectory. A good example are the results of straight line segment tracking, with zero speed initial conditions ($\eta_1 = \eta_2 = \eta_3 = 0$), presented in figure 7. On both the ends of the line segment, the robot stops and reverts the movement direction without the change of orientation θ.

6. CONCLUSIONS

Applying the singularity robust pseudoinverse, described in (Nakamura, 1991), for avoiding the singularities of the decoupling transformation in dynamic linearization algorithm, described in (Tchoń *et al.*, 2002), resulted in effective trajectory tracking not only in the cases allowed for the non-modified version, but also when the original algorithm fails.

Implementation of both versions of dynamic linearization algorithm in the MK robot controller requires fast floating point unit, (Kabała and Wnuk, 2002*b*). Respective upgrade is currently under development.

REFERENCES

Kabała, Marek and Marek Wnuk (2002*a*). Spherical mobile robot RoBall (preliminary project). Technical report. Institute of Engineering Cybernetics, Wrocław University of Technology. (in Polish).

Kabała, Marek and Marek Wnuk (2002*b*). Structure and software of two-wheel mobile robot. Technical report. Institute of Engineering Cybernetics, Wrocław University of Technology. (in Polish).

Nakamura, Y. (1991). *Advanced Robotics: Redundancy and Optimization*. Addison-Wesley. Reading.

Tchoń, Krzysztof, Marek Kabała and Marek Wnuk (2002). Trajectory tracking algorithm for MK mobile robot. In: *XIV KKA Proceedings*. Vol. II. pp. 663–668. (in Polish).