eProceedings of RoManSy 2012, 19th CISM-IFTOMM Symposium on Robot Design Dynamics and Control, June 12-15, 2012, Paris. Paper id.: 17.

Redundancy Resolution of the Underactuated Manipulator ACROBOTER

Ambrus Zelei [*], László Bencsik [‡], László L. Kovács [*] and Gábor Stépán [‡]

[*] HAS-BUTE Research Group on Dynamics of Machines and Vehicles, Budapest, Hungary, zelei@mm.bme.hu, kovacs@mm.bme.hu

[‡] Department of Applied Mechanics, Budapest University of Technology and Economics, Budapest, Hungary, bencsik@mm.bme.hu, stepan@mm.bme.hu

Abstract The domestic robot platform ACROBOTER exploits a novel concept of ceiling based locomotion. A climber unit moves on the almost obstacle free ceiling, while carries a swinging unit with a system of suspending and orienting cables. The objective of the robot is the fine positioning of the swinging unit that accomplishes path following or pick and place tasks. Its motion is controlled by ducted fan actuators additionally to the variable length suspending cables. The complexity of the mechanical structure induces the use of natural coordinates for the kinematical description. An algorithm is proposed to control this underactuated and also redundant manipulator, which can be characterized as a control-constraint based computed torque control strategy.

1 Introduction

Recently, more and more robotic systems try to utilize the advantages of the underactuation, like the ones in cases of agile motion and energy efficient operation. The indoor domestic robot called ACROBOTER published by [Stépán and et al. (2009)] is suspended from the ceiling on a cable similarly to cranes, so it is able to utilize the pendulum-like motion efficiently, while the suspending cable also provides power and signal transmission for the swinging unit. The robot utilizes the ceiling that is almost obstacle-free compared to the ground, since it avoids all problems of floor based locomotion caused by randomly placed small objects obstructing the free motion.

The mechanical structure of ACROBOTER can be divided into two main parts; the climber unit (CU) carries the swinging unit (SU), which hangs on a main cable (MC) and three orienting secondary cables (SC) as shown in Fig.1. The CU is a fully actuated, planar RRT robot and is able to move the upper mounting point of the MC in arbitrary location in a plain parallel

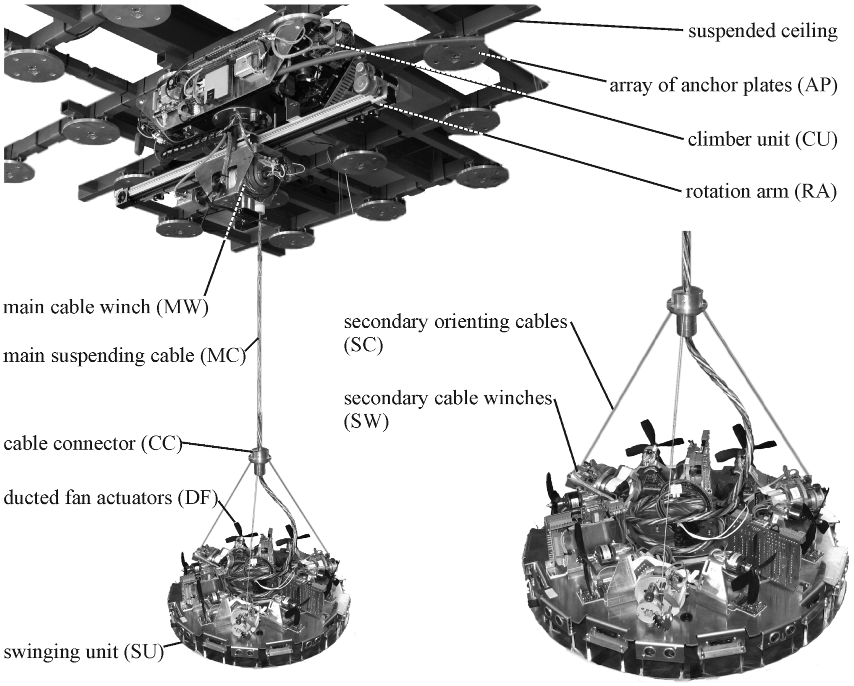

Figure 1. ACROBOTER prototype

to the ceiling. The length of the MC and SCs are varied by servo motors, and the positioning of the SU is assisted by ducted fan actuators. Despite the large number of actuators, the system is still underactuated, because the number of actuators is lower than the DoFs. Additionally, the system is kinematically redundant, because the position and orientation of the SU are prescribed only and there are no requirements for the motion of the CU and the horizontal motion of the cable connector (CC).

We propose a control algorithm for the ACROBOTER system and give a general formalism, which is applicable for underactuated systems with kinematic redundancy, and suits to the principle of natural coordinates.

2 Mechanical model and control task

The planar model of the ACROBOTER manipulator is shown in Fig.2. The CU is substituted by a single horizontal linear drive, the CC is modeled by a particle with 2 DoFs and the SU is modeled by a rigid body with 3 DoFs. We handle the manipulator as a multibody system described by the Cartesian coordinates of the base points P_L, P_{CC}, P_1 and P_2 of the included bodies. Such set of dependent coordinates is called natural coordinates by [de Jalón and Bayo (1994)]. The vector of the $n = 7$ dependent descriptor coordinates and the $m = 1$ dimensional single geometric constraint representing the constant distance between P_1 and P_2 are introduced as:

$$\mathbf{q} = \begin{bmatrix} x_L & x_{CC} & z_{CC} & x_1 & z_1 & x_2 & z_2 \end{bmatrix}^{\mathrm{T}}, \tag{1}$$

$$\boldsymbol{\varphi}(\mathbf{q}) = \left[(x_2 - x_1)^2 + (z_2 - z_1)^2 - (\overline{\mathrm{P}_1\mathrm{P}_2})^2\right]. \tag{2}$$

The system is controlled by $g = 5$ actuators, which is less than the $n - m = 6$ DoFs, thus the system is underactuated. The actuator forces are shown in Fig.2 and are arrayed in the control input vector $\mathbf{u}$:

$$\mathbf{u} = \begin{bmatrix} F_L & F_M & F_1 & F_2 & F_T \end{bmatrix}^{\mathrm{T}}. \tag{3}$$

The task of the manipulator is to move a specified point of the SU on a prescribed trajectory given by x^d and z^d as functions of time. Besides, the cable connector has to be kept in a given vertical distance h^d_{CC} above the centre of the swinging unit, and the SU has to be kept horizontal. These tasks are expressed by the $l = 4$ dimensional control-constraint vector:

$$\boldsymbol{\gamma}(\mathbf{q}, t) = \begin{bmatrix} \frac{z_1+z_2}{2} + h^d_{CC} - z_{CC} \\ \frac{x_1+x_2}{2} - x^d \\ \frac{z_1+z_2}{2} - z^d \\ z_1 - z_2 \end{bmatrix}. \tag{4}$$

In general, fully actuated manipulators equipped with more internal DoFs than required to perform a specified task are called kinematically redundant [Spong and Vidyasagar (1989)]. For such systems the inverse kinematic calculation is not unique. The above definition can be specialized for underactuated manipulators, for which the inverse kinematical and dynamical calculations may lead to a unique solution, if the number of independent control inputs and the dimension of the task is equal: $g = l$ [Blajer and Kolodziejczyk (2008); Kovács et al. (2010); Zelei et al. (2011)]. Consequently it is possible to introduce the definition: *underactuated manipulators equipped with more independent control inputs than required to perform a specified task are called dynamically redundant underactuated systems.* This definition is equivalent to kinematic redundancy for fully actuated robots. In contrast, the inverse kinematics of underactuated systems cannot be solved uniquely, so these are always kinematically redundant.

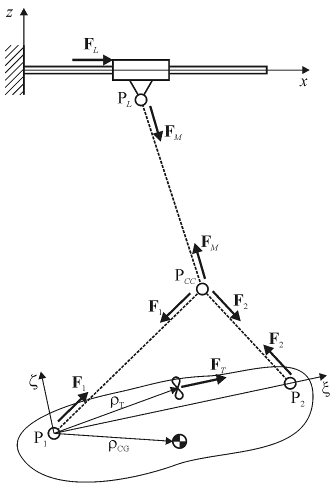

Figure 2. Planar model

	underactuated	kinematically redundant	dynamically redundant
$n-m=g=l$	no	no	no
$n-m>g=l$	yes	yes	no
$n-m>g>l$	yes	yes	yes

Table 1. Redundancy

However, in $g = l$ cases, considering the inverse dynamics, the determination of the control input and kinematics is unique, so dynamic redundancy does not stand. Table 1. summarizes the possible cases for $n - m$ DoFs, l number of specified tasks and g independent actuators.

In the investigated problem $l < g$, thus the system is kinematically and dynamically redundant. One can observe, that in (4) there are no prescriptions for the x_L position of the linear drive. For the redundancy resolution, we use the idea of virtual springs adopted from [McLean and Cameron (1996)]. The angle of the MC is minimized, thus we apply a virtual spring between the x_L and x_{CC} horizontal positions with stiffness k_v. Additionally, the speed of the linear drive should be small, so we introduce a virtual damping element with damping ratio d_v. This optimization rule can be formulated by the $g - l = 1$ dimensional non-holonomic constraint equation $\boldsymbol{\psi}(\mathbf{q}, \dot{\mathbf{q}}, t) = \mathbf{0}$ with

$$\boldsymbol{\psi}(\mathbf{q}, \dot{\mathbf{q}}, t) \quad = \quad \left[\; d_v \dot{x}_L + k_v(x_L - x_{CC}) \; \right]. \qquad (5)$$

The formalism published by [de Jalón and Bayo (1994)] provides the equations of motion for the physical system showed in Fig.2. The dynamical model can be written in the form of a differential-algebraic equation:

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) + \boldsymbol{\varphi}_{\mathbf{q}}^{\mathrm{T}}(\mathbf{q})\boldsymbol{\lambda} \quad = \quad \mathbf{Q}(\mathbf{q}) + \mathbf{H}(\mathbf{q})\mathbf{u}, \qquad (6)$$

$$\boldsymbol{\varphi}(\mathbf{q}) \quad = \quad \mathbf{0}, \qquad (7)$$

where the positive definite mass matrix $\mathbf{M}(\mathbf{q}) \in \mathbb{R}^{n \times n}$ is constant in case of the use of natural coordinates. However, in the next section we do not focus on this special case, thus, in general the mass matrix may depend on the descriptor coordinates. $\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) \in \mathbb{R}^n$ is the vector of forces arising partly from the dynamics of the system (Coriolis, centrifugal, etc.) and from active forces (springs, dampers, etc.). $\mathbf{Q}(\mathbf{q}) \in \mathbb{R}^n$ is the vector of gravitational forces. $\mathbf{H}(\mathbf{q}) \in \mathbb{R}^{n \times g}$ is the control input matrix and $\mathbf{u} \in \mathbb{R}^g$ is the control input vector. Matrix $\boldsymbol{\varphi}_{\mathbf{q}}(\mathbf{q}) = \partial \boldsymbol{\varphi}(\mathbf{q})/\partial \mathbf{q} \in \mathbb{R}^{m \times n}$ is the constraint Jacobian associated with the geometric constraints $\boldsymbol{\varphi}(\mathbf{q}) \in \mathbb{R}^m$. $\boldsymbol{\lambda} \in \mathbb{R}^m$ is the vector of the Lagrange multipliers. The dimension g of the control input vector is lower than the DoFs $n - m$, thus the system is underactuated.

3 The control method

The eq. of motion (6) and (7) are complemented by the control-constraint equation (also named servo-constraints) and the optimization rule:

$$\boldsymbol{\gamma}(\mathbf{q}, t) = \mathbf{0}, \tag{8}$$

$$\boldsymbol{\psi}(\mathbf{q}, \dot{\mathbf{q}}, t) = \mathbf{0}. \tag{9}$$

The control-constraint vector can be handled similarly to the geometric constraints (7), however the control-constraints usually depend explicitly on time. We assume that the geometric constraints (7), the control-constraints (8) and the optimization rule (9) are linearly independent and consistent, furthermore (8) and (9) can be satisfied with bounded control forces.

The method of Lagrange multipliers is well known from [de Jalón and Bayo (1994)] regarding the numerical integration of the governing differential-algebraic equation of multibody systems. That method is based on the double time differentiation of the geometric constraints. Similarly, in our work, the geometric constraint equation (7) and the control-constraint equation (8) are formulated at the level of acceleration by differentiating them twice with respect to time, in order to make the acceleration $\ddot{\mathbf{q}}$ appear explicitly:

$$\boldsymbol{\varphi}_{\mathbf{q}}(\mathbf{q})\ddot{\mathbf{q}} + \dot{\boldsymbol{\varphi}}_{\mathbf{q}}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} = \mathbf{0}, \tag{10}$$

$$\boldsymbol{\gamma}_{\mathbf{q}}(\mathbf{q}, t)\ddot{\mathbf{q}} + \dot{\boldsymbol{\gamma}}_{\mathbf{q}}(\mathbf{q}, \dot{\mathbf{q}}, t)\dot{\mathbf{q}} + \dot{\boldsymbol{\gamma}}_t(\mathbf{q}, \dot{\mathbf{q}}, t) = \mathbf{0}, \tag{11}$$

where $\boldsymbol{\gamma}_{\mathbf{q}}(\mathbf{q}, t) \in \mathbb{R}^{l \times n}$ is the Jacobian of the control-constraint and vector $\boldsymbol{\gamma}_t(\mathbf{q}, t) \in \mathbb{R}^l$ is the partial time derivative of the explicitly time dependent part of the control-constraint. In the application of the method of Lagrange multipliers the geometric constraint equations can be stabilized by the Baumgarte stabilization technique [Baumgarte (1972)]. Similarly, we extend the acceleration level control-constraint equation (11) as follows:

$$\boldsymbol{\gamma}_{\mathbf{q}}(\mathbf{q},t)\ddot{\mathbf{q}} + \dot{\boldsymbol{\gamma}}_{\mathbf{q}}(\mathbf{q},\dot{\mathbf{q}},t)\dot{\mathbf{q}} + \dot{\boldsymbol{\gamma}}_t(\mathbf{q},\dot{\mathbf{q}},t) + \\ \mathbf{K}_D[\boldsymbol{\gamma}_{\mathbf{q}}(\mathbf{q},t)\dot{\mathbf{q}} + \boldsymbol{\gamma}_t(\mathbf{q},t)] + \mathbf{K}_P\boldsymbol{\gamma}(\mathbf{q},t) = \mathbf{0}. \tag{12}$$

On the contrary, (10) is not stabilized, because the geometric constraints are naturally satisfied. In (12) $\mathbf{K}_P \in \mathbb{R}^{l\times l}$ and $\mathbf{K}_D \in \mathbb{R}^{l\times l}$ are positive definite gain matrices. Equation (12) is asymptotically stable for fixed desired positions [Bencsik and Kovács (2011)].

The optimization rule (9) is formulated by a non-holonomic constraint, hence the acceleration $\ddot{\mathbf{q}}$ appears in its first time derivative:

$$\boldsymbol{\psi}_{\mathbf{q}}(\mathbf{q},\dot{\mathbf{q}},t)\dot{\mathbf{q}} + \boldsymbol{\psi}_{\dot{\mathbf{q}}}(\mathbf{q},\dot{\mathbf{q}},t)\ddot{\mathbf{q}} + \boldsymbol{\psi}_t(\mathbf{q},\dot{\mathbf{q}},t) = \mathbf{0}, \tag{13}$$

where $\boldsymbol{\psi}_{\mathbf{q}}(\mathbf{q},\dot{\mathbf{q}},t) \in \mathbb{R}^{(g-l)\times n}$ and $\boldsymbol{\psi}_{\dot{\mathbf{q}}}(\mathbf{q},\dot{\mathbf{q}},t) \in \mathbb{R}^{(g-l)\times n}$ are the Jacobian of the optimization rule regarding $\mathbf{q}$ and $\dot{\mathbf{q}}$ respectively. Vector $\boldsymbol{\psi}_t(\mathbf{q},\dot{\mathbf{q}},t)$ is the partial time derivative of the explicitly time dependent part of (9). Since the optimization rule is given in the form of an artificial constraint, (13) has to be stabilized similarly to the control-constraint. We extend (13) with the positive definite gain matrix $\mathbf{K}_\psi$:

$$\boldsymbol{\psi}_{\mathbf{q}}(\mathbf{q},\dot{\mathbf{q}},t)\dot{\mathbf{q}} + \boldsymbol{\psi}_{\dot{\mathbf{q}}}(\mathbf{q},\dot{\mathbf{q}},t)\ddot{\mathbf{q}} + \boldsymbol{\psi}_t(\mathbf{q},\dot{\mathbf{q}},t) + \mathbf{K}_\psi\boldsymbol{\psi}(\mathbf{q},\dot{\mathbf{q}},t) = \mathbf{0}. \tag{14}$$

The unconstrained dynamic equation (6), the acceleration level geometric constraint equation (10), the stabilized acceleration level control-constraint equation (12) and the optimization rule (14) is incorporated in hyper-matrix form as follows:

$$\begin{bmatrix} \mathbf{M}(\mathbf{q}) & \boldsymbol{\varphi}_{\mathbf{q}}^{\mathrm{T}}(\mathbf{q}) & -\mathbf{H}(\mathbf{q}) \\ \boldsymbol{\varphi}_{\mathbf{q}}(\mathbf{q}) & \mathbf{0} & \mathbf{0} \\ \boldsymbol{\gamma}_{\mathbf{q}}(\mathbf{q},t) & \mathbf{0} & \mathbf{0} \\ \boldsymbol{\psi}_{\dot{\mathbf{q}}}(\mathbf{q},\dot{\mathbf{q}},t) & \mathbf{0} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \ddot{\mathbf{q}} \\ \boldsymbol{\lambda} \\ \mathbf{u} \end{bmatrix} = \\ \begin{bmatrix} \mathbf{Q}(\mathbf{q}) - \mathbf{C}(\mathbf{q},\dot{\mathbf{q}}) \\ -\dot{\boldsymbol{\varphi}}_{\mathbf{q}}(\mathbf{q},\dot{\mathbf{q}})\dot{\mathbf{q}} \\ -\dot{\boldsymbol{\gamma}}_{\mathbf{q}}(\mathbf{q},\dot{\mathbf{q}},t)\dot{\mathbf{q}} - \dot{\boldsymbol{\gamma}}_t(\mathbf{q},\dot{\mathbf{q}},t) - \mathbf{K}_D[\boldsymbol{\gamma}_{\mathbf{q}}(\mathbf{q},t)\dot{\mathbf{q}} + \boldsymbol{\gamma}_t(\mathbf{q},t)] - \mathbf{K}_P\boldsymbol{\gamma}(\mathbf{q},t) \\ -\boldsymbol{\psi}_{\mathbf{q}}(\mathbf{q},\dot{\mathbf{q}},t)\dot{\mathbf{q}} + -\boldsymbol{\psi}_t(\mathbf{q},\dot{\mathbf{q}},t) + -\mathbf{K}_\psi\boldsymbol{\psi}(\mathbf{q},\dot{\mathbf{q}},t) \end{bmatrix} \tag{15}$$

from which the control input $\mathbf{u}$, the acceleration $\ddot{\mathbf{q}}$ and the vector of Lagrange multipliers $\boldsymbol{\lambda}$ can be calculated as the function of the measured state $\mathbf{q}$ and $\dot{\mathbf{q}}$ of the system, where $\mathbf{q}$ and $\dot{\mathbf{q}}$ come from the measured values.

4 Simulation results

A round-cornered rectangular shaped trajectory was prescribed, on which the SU was moved along periodically. Fig.3. shows one period of the motion.

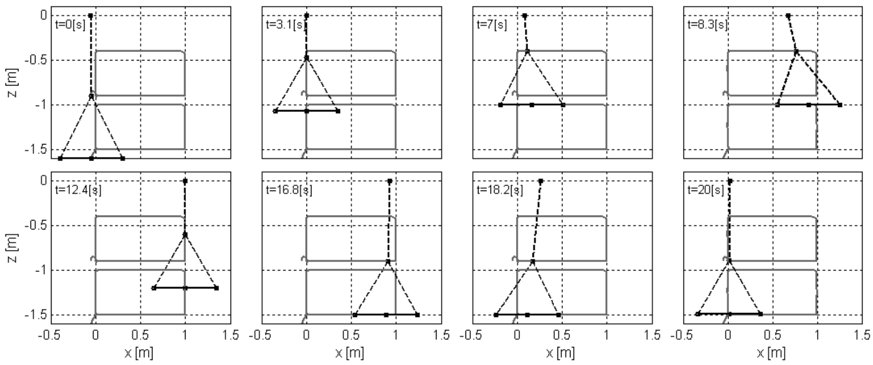

Figure 3. Simulated motion

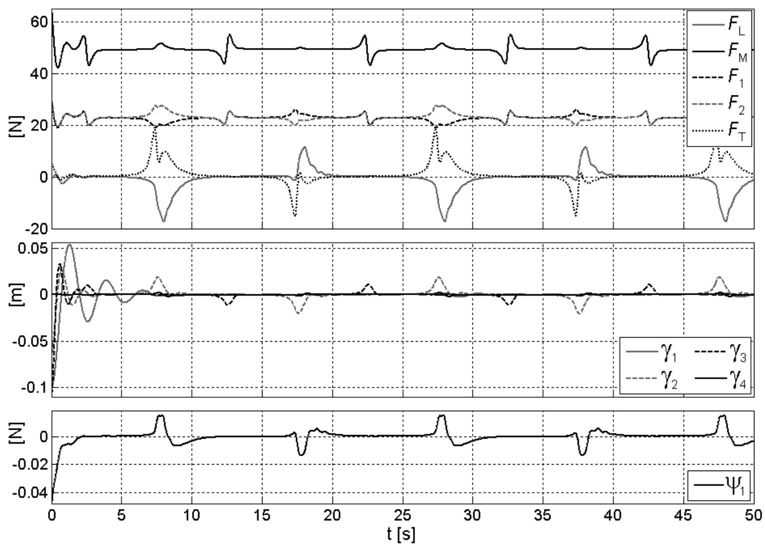

Figure 4. Simulation results: time histories

In Fig.4. the time histories of the control forces, the control-constraint violation and the time history of the violation of the optimization rule can be seen in approximately 2.5 periods. It can be clearly observed that the perturbation applied in the initial time instant is eliminated quickly by the controller, however $\gamma_1 ... \gamma_4$ and ψ_1 values grow up, when larger accelerations are required in the corners of the prescribed trajectory. As a summary of the simulation work, we can say that the violation of the control-constraints and the optimization rule could be driven to zero in a stable way, and the motion of the manipulator was also stable and smooth.

5 Conclusions

An efficient motion control algorithm was presented and analyzed in case of the cable suspended domestic robot ACROBOTER, which is an underactuated system and also dynamically redundant. A definition of dynamic redundancy was given for underactuated robots. An existing computed torque control algorithm was extended by means of the introduction of an optimization rule in the form of an artificial non-holonomic constraint. The corresponding general formalism was presented, and the numerical simulations of case studies showed the efficiency of the method.

Bibliography

J. Baumgarte. Stabilization of constraints and integrals of motion in dyanamical systems. *Computer methods in applied mechanics and engineering*, 1(1):1–16, 1972.

L. Bencsik and L. L. Kovács. Stability case study of an underactuated service robot. In *Dynamical Systems - Nonlinear Dynamics and Control*, pages 89–94, 2011.

W. Blajer and K. Kolodziejczyk. Modeling of underactuated mechanical systems in partly specified motion. *Journal of Theoretical and Applied Mechanics*, 46(2):383–394, 2008.

J.G. de Jalón and E. Bayo. *Kinematic and dynamic simulation of multibody systems: the real-time challenge*. Springer-Verlag, 1994.

L. L. Kovács, A. Zelei, L. Bencsik, J. Turi, and G. Stépán. Motion control of an under-actuated service robot using natural coordinates. In *Proceedings of ROMANSY 18 Robot Design, Dynamics and Control*, pages 331–338, 2010.

A. McLean and S. Cameron. The virtual springs method: Path planning and collision avoidance for redundant manipulators. *Intl. Journal of Robotics Research*, 15(4):300–319, 1996.

M. W. Spong and M. Vidyasagar. *Robot Dynamics and Control*. John Wiley & Sons, 1989.

G. Stépán and et al. A ceiling based crawling, hoisting and swinging service robot platform. In *Proceedings of Beyond Gray Droids: Domestic Robot Design for the 21st Century Workshop at HCI 2009*, 2009.

A. Zelei, L. L. Kovács, and G. Stépán. Computed torque control of an under-actuated service robot platform modeled by natural coordinates. *Commun Nonlinear Sci Numer Simulat*, 16(5):2205–2217, 2011.