

Case studies for computed torque control of constrained underactuated systems

Ambrus Zelei / Gábor Stépán

Received 2012-04-30

Abstract

Computed torque control (CTC) method is an efficient technique for trajectory tracking control of robot manipulators. As a model-based control, CTC needs the inverse dynamics calculation of the dynamical system. A special group of these systems is formed by the underactuated ones, in which the number of independent control inputs is lower than the degrees of freedom of the system. In these systems, the inverse dynamics calculation is a challenging task, because the inverse calculation leads to the solution of a differential-algebraic equation (DAE). Complex robotic structures like the ones with closed kinematic loops are generally modeled by redundant descriptor coordinates instead of the Lagrangian approach when a minimum set of generalized coordinates are chosen. The use of non-minimum set of descriptor coordinates requires the introduction of a set of geometric constraints, which are expressed in the form of algebraic equations. Thus, the mathematical model of the robotic structure itself is also a DAE. A few different CTC method based algorithms are studied. The control algorithms are compared in the case of the simplest possible linear dynamical system with special attention to the choice of the descriptor coordinate set.

Keywords

Underactuated robotics · computed torque control · computed desired computed torque control

Ambrus Zelei

Department of Applied Mechanics, BME, H-1111, Budapest, Műegyetem rkp. 5, Hungary
e-mail: zelei@mm.bme.hu

Gábor Stépán

Department of Applied Mechanics, BME, H-1111, Budapest, Műegyetem rkp. 5, Hungary
e-mail: stepan@mm.bme.hu

1 Introduction

One of the most common ways of controlling robot motion is based on a linear control system obtained by feeding back the dynamics of the original non-linear system. After this feedback linearization, an arbitrary motion can be prescribed, which is realizable until the actuators are able to provide the required torques. This method cannot be directly applied in case of the so-called underactuated systems because the number of the independent control inputs is lower than the degrees of freedom (DoF) of the system. For a general overview of this problem let us consider the following equation of motion:

$$\ddot{\mathbf{q}} = \mathbf{f}_1(\mathbf{q}, \dot{\mathbf{q}}, t) + \mathbf{f}_2(\mathbf{q}, \dot{\mathbf{q}}, t)\mathbf{u}, \quad (1)$$

where \mathbf{q} is the vector of the generalized coordinates of minimum number and \mathbf{u} is the control input vector. The only assumption in the general equation of motion (1) is that the control input \mathbf{u} appears linearly in the system. The system is fully actuated, if the rank of the input matrix $\mathbf{f}_2(\mathbf{q}, \dot{\mathbf{q}}, t)$ equals to the DoFs of the system:

$$\text{rank}(\mathbf{f}_2(\mathbf{q}, \dot{\mathbf{q}}, t)) = \text{dim}(\mathbf{q}). \quad (2)$$

In case of such a fully actuated system the control input can be formulated by inverting the control input matrix $\mathbf{f}_2(\mathbf{q}, \dot{\mathbf{q}}, t)$ as:

$$\mathbf{u} = \mathbf{f}_2(\mathbf{q}, \dot{\mathbf{q}}, t)^{-1} [-\mathbf{f}_1(\mathbf{q}, \dot{\mathbf{q}}, t) + \ddot{\mathbf{u}}]. \quad (3)$$

If we substitute (3) into the equation of motion (1), we obtain the following linear system:

$$\ddot{\mathbf{q}} = \ddot{\mathbf{u}}, \quad (4)$$

where the synthetic control input $\ddot{\mathbf{u}}$ can be chosen, for example, as

$$\ddot{\mathbf{u}} = \ddot{\mathbf{q}}^d + \mathbf{K}_D(\dot{\mathbf{q}}^d - \dot{\mathbf{q}}) + \mathbf{K}_P(\mathbf{q}^d - \mathbf{q}), \quad (5)$$

with superscript d referring to the desired values. Here, \mathbf{K}_D and \mathbf{K}_P are positive definite gain matrices. If we calculate the control input \mathbf{u} according to (3) and we can measure \mathbf{q} and $\dot{\mathbf{q}}$ exactly than the acceleration of the system can be prescribed arbitrarily via the synthetic control input $\ddot{\mathbf{u}}$. For the above explained method vector $\mathbf{f}_1(\mathbf{q}, \dot{\mathbf{q}}, t)$ and matrix $\mathbf{f}_2(\mathbf{q}, \dot{\mathbf{q}}, t)$ has to be known exactly.

The above explained computed torque method cannot be accomplished in case of underactuated systems, because $\mathbf{f}_2(\mathbf{q}, \dot{\mathbf{q}}, t)$ is not invertible in that case. We speak about underactuated systems if the number of the independent control inputs is lower than the DoFs of the system [1, 2], thus

$$\text{rank}(\mathbf{f}_2(\mathbf{q}, \dot{\mathbf{q}}, t)) < \text{dim}(\mathbf{q}). \quad (6)$$

In spite of the difficulties of controlling underactuated systems, the acquired knowledge lets us design and build more energy efficient and more agile robots. Cranes are typically underactuated systems because there is no direct influence of the actuators on the swinging payload. The motion of the top mounting point of the cable can be controlled directly, but the motion of the payload is defined by the dynamics of the system. A lot of different approaches were developed in the past for the anti-sway control of the cranes [3, 4]. Some newly designed robotic systems utilize the advantages of the underactuation, like agile motion and energy efficient operation. The domestic robot called ACROBOTER [5] hangs down from the ceiling on a suspending cable similarly to cranes, and it is able to utilize the pendulum-like motion efficiently. In practice, also the flexibility of the machine components can cause underactuation [6, 7]. Typical examples are rotors driven via a flexible shafts, or lightweight robots with flexible arms.

An important step of the control design of robotic manipulators is the choice of the generalized or descriptor coordinates. The dynamical modeling of mechanically complex robotic structures via the use of minimum number generalized coordinates can be inefficient. A more effective way is to choose a non-minimum set of descriptor coordinates (redundant or dependent ones) of number larger than the DoFs and to introduce appropriate geometric constraints in the form of algebraic equations. Several techniques are available in the literature, e.g., the use of reference point coordinates, natural coordinates or mixed coordinates [8]. The numerical computation is more effective with the use of non-minimum set descriptor coordinates, however the dynamical investigation is more difficult because of the presence of the algebraic equations.

In this study, we investigate the computed torque control (CTC) of underactuated systems described by non-minimum set of coordinates via a case study which is as simple as possible but catches the main character of such systems.

2 Dynamics of underactuated systems

In general a robotic manipulator system as well as any controlled mechanical system can be described by the following equation of motion using minimum set generalized coordinates:

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) = \mathbf{Q}(\mathbf{q}) + \mathbf{H}(\mathbf{q})\mathbf{u}. \quad (7)$$

where $\mathbf{M}(\mathbf{q}) \in \mathbb{R}^{n \times n}$ is the positive definite mass matrix, $\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) \in \mathbb{R}^n$ 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. $\mathbf{H}(\mathbf{q}) \in \mathbb{R}^{n \times l}$ is the control input matrix and $\mathbf{u} \in \mathbb{R}^l$ is the control input vector.

Many studies like [6, 9–13] assume that it is possible to decompose the generalized coordinates \mathbf{q} into active (actuated) \mathbf{q}_a and passive (unactuated) \mathbf{q}_p coordinates leading to:

$$\begin{bmatrix} \mathbf{M}_{aa}(\mathbf{q}) & \mathbf{M}_{ap}(\mathbf{q}) \\ \mathbf{M}_{pa}(\mathbf{q}) & \mathbf{M}_{pp}(\mathbf{q}) \end{bmatrix} \begin{bmatrix} \ddot{\mathbf{q}}_a \\ \ddot{\mathbf{q}}_p \end{bmatrix} + \begin{bmatrix} \mathbf{C}_a(\mathbf{q}, \dot{\mathbf{q}}) \\ \mathbf{C}_p(\mathbf{q}, \dot{\mathbf{q}}) \end{bmatrix} = \begin{bmatrix} \mathbf{Q}_a(\mathbf{q}) \\ \mathbf{Q}_p(\mathbf{q}) \end{bmatrix} + \begin{bmatrix} \mathbf{u} \\ \mathbf{0} \end{bmatrix}. \quad (8)$$

This decomposition is directly possible if the assumption

$$\mathbf{H}(\mathbf{q}) = [\mathbf{I}, \mathbf{0}] \quad (9)$$

stands [11]. However, the full state feedback linearization cannot be carried out for (8), because of the lack of control in the second part of the equations. This second part forms a second order non-holonomic constraint in the system. From the viewpoint of the inverse dynamics calculation we can conclude that (8) is a differential-algebraic equation (DAE), because the control input \mathbf{u} appears as an algebraic variable additionally to \mathbf{q} .

In general cases when the assumption (9) is not satisfied, we can transform the system (7) into a form similar to (8) via the projection of the system into the null-space of the input matrix $\mathbf{H}(\mathbf{q})$. Let us consider the null-space projection matrix $\mathbf{V}(\mathbf{q}) \in \mathbb{R}^{(n-l) \times n}$ of $\mathbf{H}(\mathbf{q})$ as:

$$\mathbf{V}(\mathbf{q}) = \text{null}(\mathbf{H}^T(\mathbf{q}))^T. \quad (10)$$

With (10), the $n - l$ dimensional passive part of the equation of motion (7), also named as internal-dynamics of the system, can be reformulated as:

$$\mathbf{V}(\mathbf{q})\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{V}(\mathbf{q})\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) - \mathbf{V}(\mathbf{q})\mathbf{Q}(\mathbf{q}) = \mathbf{0}. \quad (11)$$

If we apply the idea of the Moore-Penrose pseudo-inverse then the l dimensional active part of the equation of motion (7) can also be derived in the form:

$$\mathbf{H}^\dagger(\mathbf{q}) [\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) - \mathbf{Q}(\mathbf{q})] = \mathbf{u}, \quad (12)$$

where it can be shown in algebraic way that the $\mathbf{H}^\dagger(\mathbf{q}) \in \mathbb{R}^{l \times n}$ Moore-Penrose pseudo-inverse of the input matrix can be calculated as:

$$\mathbf{H}^\dagger(\mathbf{q}) = \text{inv}(\mathbf{H}^T(\mathbf{q})\mathbf{H}(\mathbf{q}))\mathbf{H}(\mathbf{q})^T. \quad (13)$$

The split into active and passive parts of the equation of motion is even more difficult in the presence of geometric constraint equations, when mechanically complex robotic structures are modeled by non-minimum set of descriptor coordinates. A possible solution is to project the equation of motion into the subspace of kinematically possible motions [14]. After this projection, we end up with an equation of motion of the form of (7) and any control technique developed for underactuated systems

(e.g. partial feedback linearization [10, 11, 15], computed desired computed torque control [6, 9]) can be applied. However the sequence of projections may require high computational effort. Consequently the other possibility is to apply the control algorithm directly for the DAE form of the system, which will be detailed in the subsequent sections.

3 Problem formulation

In the present work, we study the CTC method applied to underactuated systems described geometrically by non-minimum set of descriptor coordinates (dependent coordinates). In case of non-minimum number of descriptor coordinates, geometric constraint equations provide the connection between the redundant descriptor coordinates. Thus, the equation of motion of such systems is given in DAE form. Instead of transforming the equation of motion to ODE, CTC algorithms can directly be applied for the DAE system, so the CTC method for underactuated systems can be generalized for systems modeled by non-minimum set of coordinates.

Using non-minimum set of descriptor coordinates the dynamical model can be written in the form of a differential-algebraic equation, which has the following general form [1, 8]:

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) + \boldsymbol{\varphi}_q^T(\mathbf{q})\boldsymbol{\lambda} = \mathbf{Q}(\mathbf{q}) + \mathbf{H}(\mathbf{q})\mathbf{u}, \quad (14)$$

$$\boldsymbol{\varphi}(\mathbf{q}) = \mathbf{0}, \quad (15)$$

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 [8]. However, we do not focus on this special case, thus, in general the mass matrix may depend on the descriptor coordinates. Vector $\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) \in \mathbb{R}^n$ is the vector of the forces arising from the dynamics of the system. The matrix $\boldsymbol{\varphi}_q(\mathbf{q}) = \partial\boldsymbol{\varphi}(\mathbf{q})/\partial\mathbf{q} \in \mathbb{R}^{m \times n}$ is the constraint Jacobian associated with the geometric constraints $\boldsymbol{\varphi}(\mathbf{q}) \in \mathbb{R}^m$. For the sake of simplicity, the geometric constraints are considered to be scleronomic, but any deduction in this work can be generalized for explicitly time dependent geometric constraints, too. $\mathbf{Q}(\mathbf{q}) \in \mathbb{R}^n$ is the vector of gravitational forces. $\mathbf{H}(\mathbf{q}) \in \mathbb{R}^{n \times l}$ is the control input matrix and $\mathbf{u} \in \mathbb{R}^l$ is the control input vector. We assume that the system is underactuated, so the dimension of the control input l is lower than the DoFs $n - m$.

The inverse kinematical and dynamical calculations may lead to a unique solution if the number of control inputs and the dimension of the task is equal [7, 16, 17]. In order to reach this, we assume that the task is defined by l number of algebraic equations. This set of additional constraint equations are the so-called servo-constraints (also named control-constraints):

$$\boldsymbol{\sigma}(\mathbf{q}, t) = \mathbf{0}, \quad (16)$$

where $\boldsymbol{\sigma}(\mathbf{q}, t) \in \mathbb{R}^l$. The servo-constraint vector can be handled similarly to the geometric constraints (15), however the servo-constraints usually depend on time explicitly, thus they are rheonomic. We assume that the servo constraints and the geometric constraints are linearly independent and consistent, and the servo constraints can be satisfied with bounded control forces.

In some special cases, we can also assume that these servo-constraints can be written in the following form:

$$\boldsymbol{\sigma}(\mathbf{q}, t) = \mathbf{g}(\mathbf{q}) - \mathbf{p}(t), \quad (17)$$

where $\mathbf{g}(\mathbf{q})$ represents, for example, the end-effector position of the robot as the function of the descriptor coordinates and $\mathbf{p}(t)$ is an arbitrarily prescribed function of time expressing the performance goal to be realized [7].

With the loss of generality, reference [9] introduces the notion of the so-called controlled and uncontrolled coordinates. In the simplest case we can say that those coordinates whose trajectories are prescribed in time via the task definition are the controlled coordinates. The uncontrolled ones must be calculated with respect to the dynamics of the system, and there is no direct restriction for them from the side of the task description of the manipulator. In general, we cannot say that there is a set of coordinates which are prescribed by the task. Still, in some cases, the servo-constraints and a well chosen subset of geometric constraints can be solved for the controlled coordinates \mathbf{q}_c in closed form [17]. However, the choice of the controlled coordinates is not always unique and in case of complex systems, the choice is not obvious. Additionally we mention that the choice of the descriptor coordinates influences the above method. Finally, if we can use the notion of the controlled coordinates, then the task can be defined as

$$\mathbf{q}_c = \mathbf{q}_c^d(t), \quad (18)$$

where the superscript d refers to the desired trajectory. In this formulation, the controlled coordinates are prescribed functions of time. In such case, the descriptor coordinates can be split into controlled and uncontrolled parts as:

$$\mathbf{q}_c = \mathbf{S}_c^T \mathbf{q}, \quad (19)$$

$$\mathbf{q}_u = \mathbf{S}_u^T \mathbf{q}, \quad (20)$$

respectively, where \mathbf{S}_c and \mathbf{S}_u are task dependent selector matrices. This simplification can be utilized for saving computational time.

Reference [9] introduces the notion of collocated and non-collocated cases of the control of underactuated systems. These terms are in connection with the separation into controlled coordinates \mathbf{q}_c and uncontrolled coordinates \mathbf{q}_u . Besides, let us consider the separation between active (actuated) and passive (unactuated) coordinates denoted by \mathbf{q}_a and \mathbf{q}_p , respectively. Reference [9] defines the collocated case by

$$\mathbf{q}_c = \mathbf{q}_a, \quad (21)$$

which means that the controlled coordinates are the actuated ones. In the so-called non-collocated case

$$\mathbf{q}_c = \mathbf{q}_p, \quad (22)$$

which means that the controlled coordinates are the passive ones. An important message of [9] is that the partial feedback

linearization and the CTC cannot be done in the non-collocated case when there is no inertial coupling between the controlled and the active coordinates. In this study, we point out the importance of the choice of the descriptor coordinates, because this choice can effect the inertial coupling between the coordinates.

Note that the collocated and non-collocated cases are very special ones, because in general, the actuation of the coordinates depends on the mechanical design of the manipulator while the separation between controlled and uncontrolled coordinates depends on the control task; in general, they are not in connection to each other. Besides, the separation between controlled and uncontrolled coordinates is not straightforward.

4 The investigated mechanical model

The aim of this work is to investigate the CTC method in case of the simplest non-trivial dynamical system having the properties described in the previous section. Consequently, we consider an underactuated $n - m = 2$ DoF linear system with $l = 1$ control input only. We also introduce $m = 1$ geometric constraint equation, so the number of descriptor coordinates has to be $n = 3$. The model shown in Fig. 1 is actuated by $l = 1$ control force F , and the single geometric constraint is represented by a rigid rod of length l_{23} .

The equation of motion formulated in (14, 15) can be simplified with the stiffness (\mathbf{K}) and the damping (\mathbf{D}) matrices:

$$\mathbf{M}\ddot{\mathbf{q}}(\mathbf{q}) + \mathbf{K}\dot{\mathbf{q}} + \mathbf{D}\dot{\mathbf{q}} + \boldsymbol{\varphi}_q^T(\mathbf{q})\boldsymbol{\lambda} = \mathbf{Q}(\mathbf{q}) + \mathbf{H}(\mathbf{q})\mathbf{u}, \quad (23)$$

$$\boldsymbol{\varphi}(\mathbf{q}) = \mathbf{0}. \quad (24)$$

One has several options to choose descriptor coordinates. The most trivial way is to choose the absolute Cartesian coordinates of the blocks of masses m_1 , m_2 and m_3 as it can be seen in Fig. 1:

$$\mathbf{q} = [z_1, z_2, z_3]^T. \quad (25)$$

The mass, the stiffness and the damping matrices are

$$\mathbf{M} = \text{diag}(m_1, m_2, m_3), \quad (26)$$

$$\mathbf{K} = \begin{bmatrix} k & -k & 0 \\ -k & k & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad (27)$$

$$\mathbf{D} = \begin{bmatrix} d & -d & 0 \\ -d & d & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad (28)$$

respectively. The geometric constraint and the constraint Jacobian are:

$$\boldsymbol{\varphi}(\mathbf{q}) = [z_2 - z_3 - l_{23}], \quad (29)$$

$$\boldsymbol{\varphi}_q(\mathbf{q}) = \begin{bmatrix} 0 & 1 & -1 \end{bmatrix}. \quad (30)$$

The generalized force vector due to the gravity is

$$\mathbf{Q}(\mathbf{q}) = \begin{bmatrix} -m_1g & -m_2g & -m_3g \end{bmatrix}^T. \quad (31)$$

The coefficient of the control input and the control input itself are:

$$\mathbf{H}(\mathbf{q}) = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix}, \quad (32)$$

$$\mathbf{u} = [F], \quad (33)$$

respectively.

As already emphasized in the previous section, the applicability of the control approaches highly depend on the chosen descriptor coordinate set. To represent this, we also choose relative coordinates, which are also typical in robotics, since these are measured by the encoders. Introduce the descriptor coordinate vector $\hat{\mathbf{q}} = [z_1, \hat{z}_2, \hat{z}_3]^T$, where the relative coordinates can be calculated from the previously introduced absolute ones as $\hat{z}_2 = z_2 - z_1$ and $\hat{z}_3 = z_3 - z_2$. The new form of the system matrices are the following:

$$\hat{\mathbf{M}} = \begin{bmatrix} m_1 + m_2 + m_3 & m_2 + m_3 & m_3 \\ m_2 + m_3 & m_2 + m_3 & m_3 \\ m_3 & m_3 & m_3 \end{bmatrix}, \quad (34)$$

$$\hat{\mathbf{K}} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & k & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad (35)$$

$$\hat{\mathbf{D}} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & d & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad (36)$$

$$\hat{\boldsymbol{\varphi}}(\hat{\mathbf{q}}) = [\hat{z}_3 + l_{23}], \quad (37)$$

$$\hat{\boldsymbol{\varphi}}_q(\hat{\mathbf{q}}) = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}, \quad (38)$$

$$\hat{\mathbf{Q}}(\hat{\mathbf{q}}) = \begin{bmatrix} -(m_1 + m_2 + m_3)g & -(m_2 + m_3)g & -m_3g \end{bmatrix}^T. \quad (39)$$

$$\hat{\mathbf{H}}(\hat{\mathbf{q}}) = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix}, \quad (40)$$

$$\hat{\mathbf{u}} = [F]. \quad (41)$$

The most important difference between the two descriptions is that the absolute coordinates do not present inertial coupling between the coordinates due to the diagonal mass matrix, while inertial coupling is present in case of relative coordinates. These trivial observations become especially important because reference [9] shows that the CTC in non-collocated cases is possible only if there is inertial coupling between the controlled and the active coordinates.

5 Control approaches

In this section, the method based on the backward Euler (BE) direct discretization of the control law [16] and the application of the method of Lagrange multipliers (MLM) [8] are overviewed. For the application of most of the control algorithms, the constraining forces, in mathematical terms the Lagrange multipliers, have to be eliminated from the equation of motion (14, 15). For the approaches presented below, this is not needed.

For the presented methods, the formulation of the geometric constraint equation (15) and the servo-constraint equation (16)

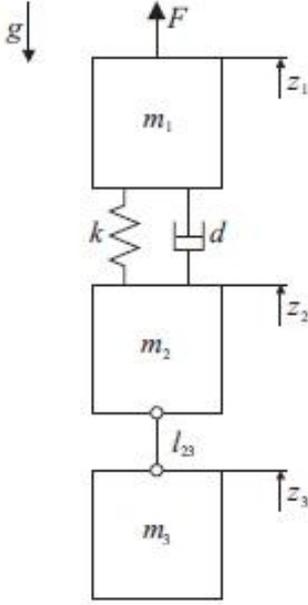


Fig. 1. Underactuated and constrained mechanical model for CTC case study

are needed at the level of acceleration by differentiating them twice with respect to time, in order to make the acceleration $\ddot{\mathbf{q}}$ appear explicitly:

$$\boldsymbol{\varphi}_{\mathbf{q}}(\mathbf{q})\ddot{\mathbf{q}} + \boldsymbol{\varphi}_{\dot{\mathbf{q}}}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} = \mathbf{0}, \quad (42)$$

$$\boldsymbol{\sigma}_{\mathbf{q}}(\mathbf{q}, t)\ddot{\mathbf{q}} + \dot{\boldsymbol{\sigma}}_{\mathbf{q}}(\mathbf{q}, \dot{\mathbf{q}}, t)\dot{\mathbf{q}} + \dot{\mathbf{c}}(\mathbf{q}, \dot{\mathbf{q}}, t) = \mathbf{0}, \quad (43)$$

where $\boldsymbol{\sigma}_{\mathbf{q}}(\mathbf{q}) \in \mathbb{R}^{l \times n}$ is the Jacobian of the servo-constraint and $\mathbf{c}(t) \in \mathbb{R}^l$ is the time derivative of the explicitly time dependent part of the servo-constraint. In the application of the method of Lagrange multipliers the geometric constraint equations can be stabilized by the Baumgarte method [8, 18]. Similarly, we extend the acceleration level servo-constraint equation (43) as follows:

$$\boldsymbol{\sigma}_{\mathbf{q}}(\mathbf{q}, t)\ddot{\mathbf{q}} + \dot{\boldsymbol{\sigma}}_{\mathbf{q}}(\mathbf{q}, \dot{\mathbf{q}}, t)\dot{\mathbf{q}} + \dot{\mathbf{c}}(\mathbf{q}, \dot{\mathbf{q}}, t) + \mathbf{K}_D[\boldsymbol{\sigma}_{\mathbf{q}}(\mathbf{q}, t)\dot{\mathbf{q}} + \mathbf{c}(\mathbf{q}, t)] + \mathbf{K}_P\boldsymbol{\sigma}(\mathbf{q}, t) = \mathbf{0}, \quad (44)$$

where $\mathbf{K}_P \in \mathbb{R}^{l \times l}$ and $\mathbf{K}_D \in \mathbb{R}^{l \times l}$ are positive definite gain matrices. Note that the method can be simplified if these gain matrices are substituted by scalar parameters K_P and K_D as it appears in the original work of Baumgarte [18]. Equation (44) is asymptotically stable for fixed desired positions, which is utilized in the control methods presented in this section.

The methods introduced in the following subsections calculate both the control input and the desired coordinates as it is done by the CDCTC method summarized in reference [6, 9].

5.1 Control via direct backward discretization

As opposed to the CDCTC method, we directly apply the backward Euler discretization of the DAE system (14, 15, 16) and the resulting set of nonlinear algebraic equations are solved by the Newton-Raphson method for the desired actuator forces, uncontrolled coordinates and Lagrange multipliers in the upcoming time step [16, 17].

First, we transform the unconstrained dynamic equation (14) into first order system via introducing the new variable $\mathbf{y} = \dot{\mathbf{q}}$. Then we consider the geometric constraint equation (15) and the stabilized second time derivative (44) of the servo-constraint equation (16). After that the control law can be obtained for \mathbf{u} from:

$$\dot{\mathbf{q}} = \mathbf{y}, \quad (45)$$

$$\dot{\mathbf{y}} = -\mathbf{M}(\mathbf{q})^{-1} [\mathbf{C}(\mathbf{q}, \mathbf{y}) + \boldsymbol{\varphi}_{\mathbf{q}}^T(\mathbf{q})\boldsymbol{\lambda} - \mathbf{Q}(\mathbf{q}) - \mathbf{H}(\mathbf{q})\mathbf{u}], \quad (46)$$

$$\boldsymbol{\varphi}(\mathbf{q}) = \mathbf{0}, \quad (47)$$

$$\boldsymbol{\sigma}_{\mathbf{q}}(\mathbf{q}, t)\dot{\mathbf{y}} + \dot{\boldsymbol{\sigma}}_{\mathbf{q}}(\mathbf{q}, \mathbf{y}, t)\mathbf{y} + \dot{\mathbf{c}}(\mathbf{q}, \mathbf{y}, t) + \mathbf{K}_D[\boldsymbol{\sigma}_{\mathbf{q}}(\mathbf{q}, t)\mathbf{y} + \mathbf{c}(\mathbf{q}, t)] + \mathbf{K}_P\boldsymbol{\sigma}(\mathbf{q}, t) = \mathbf{0}. \quad (48)$$

Applying the backward Euler discretization to (45-48) a $2n-l+m$ dimensional system of nonlinear algebraic equations arises in the form:

$$\mathbf{q}_i - \mathbf{q}_{i-1} - h\mathbf{y}_i = \mathbf{0}, \quad (49)$$

$$\mathbf{y}_i - \mathbf{y}_{i-1} - h\mathbf{M}(\mathbf{q}_i)^{-1}$$

$$[-\mathbf{C}(\mathbf{q}_i, \mathbf{y}_i) - \boldsymbol{\varphi}_{\mathbf{q}}^T(\mathbf{q}_i)\boldsymbol{\lambda}_i + \mathbf{Q}(\mathbf{q}_i) + \mathbf{H}(\mathbf{q}_i)\mathbf{u}_i] = \mathbf{0}, \quad (50)$$

$$\boldsymbol{\varphi}(\mathbf{q}_i) = \mathbf{0}, \quad (51)$$

$$\boldsymbol{\sigma}_{\mathbf{q}}(\mathbf{q}_i, t_i)\dot{\mathbf{y}}_i + \dot{\boldsymbol{\sigma}}_{\mathbf{q}}(\mathbf{q}_i, \mathbf{y}_i, t_i)\mathbf{y}_i + \dot{\mathbf{c}}(\mathbf{q}_i, \mathbf{y}_i, t_i) +$$

$$\mathbf{K}_D[\boldsymbol{\sigma}_{\mathbf{q}}(\mathbf{q}_i, t_i)\mathbf{y}_i + \mathbf{c}(\mathbf{q}_i, t_i)] + \mathbf{K}_P\boldsymbol{\sigma}(\mathbf{q}_i, t_i) = \mathbf{0} \quad (52)$$

for the i -th value of the desired coordinates \mathbf{q}_i , their time derivatives \mathbf{y}_i , the control inputs \mathbf{u}_i and the Lagrange multipliers $\boldsymbol{\lambda}_i$. It can be formulated as a function $\mathbf{F}(\mathbf{z}_i)$ of the vector of unknowns \mathbf{z}_i :

$$\mathbf{z}_i = [\mathbf{q}_i, \mathbf{y}_i, \mathbf{u}_i, \boldsymbol{\lambda}_i]^T. \quad (53)$$

The system (49-52) can be solved by the Newton-Raphson iteration. The j -th approximation of this iteration for the unknowns in the i -th time step can be formulated as:

$$\mathbf{z}_i^j = \mathbf{z}_i^{j-1} - \mathbf{J}^{-1}(\mathbf{z}_i^{j-1})\mathbf{F}(\mathbf{z}_i^{j-1}), \quad (54)$$

where $\mathbf{J}(\mathbf{z}_i^{j-1})$ is the Jacobian of $\mathbf{F}(\mathbf{z}_i^{j-1})$. This Newton-Raphson iteration gives accurate result in a few steps, because the initial estimation \mathbf{z}_i^0 comes partly from the solution \mathbf{u}_{i-1} and $\boldsymbol{\lambda}_{i-1}$ calculated in the previous time step and the previously sampled measurement data \mathbf{q}_{i-1} and \mathbf{y}_{i-1} .

The calculation of the Jacobian can be accomplished analytically and also numerically. In order to save computational time it is enough to calculate the Jacobian once in each time step, and after that the Newton-Raphson iteration uses the same Jacobian.

In some cases the Jacobian matrix may be ill-conditioned, but the problem can be handled by singular value decomposition.

5.2 Control by means of the extended method of Lagrange multipliers

The method of Lagrange multipliers (MLM) is well known regarding the numerical integration of the governing differential-algebraic equation of multibody systems [8]. In this section, the

MLM is generalized for the servo-constraint based control of underactuated dynamical systems described by non-minimum set of descriptor coordinates and additional geometric constraint equations [19].

The basic idea of the method is to join the unconstrained dynamical equation (14) to the geometric constraint and to the servo-constraint in acceleration level given in (42) and (44). We describe two versions of this method, first the two-step MLM, and secondly the one-step MLM. These two methods differ only in numerical computation issues.

In the first step of the two-step MLM, the null-space projection matrix $\mathbf{V}(\mathbf{q})$ of the input matrix $\mathbf{H}(\mathbf{q})$ is calculated as in (10) with which the control input \mathbf{u} can be eliminated from the unconstrained dynamic equation (14). Thus, we obtain the internal dynamics (also named uncontrolled dynamics or zero dynamics) of the system. The internal dynamics extended with the servo-constraints is expressed in the following hyper matrix form:

$$\begin{bmatrix} \mathbf{V}(\mathbf{q})\mathbf{M}(\mathbf{q}) & \mathbf{V}(\mathbf{q})\boldsymbol{\varphi}_q^T(\mathbf{q}) \\ \boldsymbol{\varphi}_q(\mathbf{q}) & \mathbf{0} \\ \boldsymbol{\sigma}_q(\mathbf{q}, t) & \mathbf{0} \end{bmatrix} \begin{bmatrix} \ddot{\mathbf{q}} \\ \boldsymbol{\lambda} \end{bmatrix} = \begin{bmatrix} \mathbf{V}(\mathbf{q})\mathbf{Q}(\mathbf{q}) - \mathbf{V}(\mathbf{q})\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) \\ -\dot{\boldsymbol{\varphi}}_q(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} \\ -\dot{\boldsymbol{\sigma}}_q(\mathbf{q}, \dot{\mathbf{q}}, t)\dot{\mathbf{q}} - \dot{\mathbf{c}}(\mathbf{q}, \dot{\mathbf{q}}, t) - \mathbf{f}_k(\mathbf{q}, \mathbf{t}) \end{bmatrix}, \quad (55)$$

where $\mathbf{f}_k(\mathbf{q}, \mathbf{t}) = \mathbf{K}_D[\boldsymbol{\sigma}_q(\mathbf{q}, t)\dot{\mathbf{q}} + \mathbf{c}(\mathbf{q}, t)] - \mathbf{K}_P\boldsymbol{\sigma}(\mathbf{q}, t)$. These equations can be solved for the second derivatives of the descriptor coordinates and for the Lagrange multipliers, where \mathbf{q} and $\dot{\mathbf{q}}$ come from the measured values. As a second step of the method, the pseudo-inverse of $\mathbf{H}(\mathbf{q})$ is calculated with which the control input can be given as

$$\mathbf{u} = \mathbf{H}^\dagger(\mathbf{q}) [\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) + \boldsymbol{\varphi}_q^T(\mathbf{q})\boldsymbol{\lambda} - \mathbf{Q}(\mathbf{q})], \quad (56)$$

where $\mathbf{H}^\dagger(\mathbf{q})$ is the Moore-Penrose pseudo-inverse of the control input matrix calculated as equation (13) shows.

In the one-step MLM, the unconstrained dynamic equation (14), the acceleration level geometric constraint equation (42) and the acceleration level servo-constraint equation (44) can be incorporated in hyper-matrix form as follows:

$$\begin{bmatrix} \mathbf{M}(\mathbf{q}) & \boldsymbol{\varphi}_q^T(\mathbf{q}) & -\mathbf{H}(\mathbf{q}) \\ \boldsymbol{\varphi}_q(\mathbf{q}) & \mathbf{0} & \mathbf{0} \\ \boldsymbol{\sigma}_q(\mathbf{q}, t) & \mathbf{0} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \ddot{\mathbf{q}} \\ \boldsymbol{\lambda} \\ \mathbf{u} \end{bmatrix} = \begin{bmatrix} \mathbf{Q}(\mathbf{q}) - \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) \\ -\dot{\boldsymbol{\varphi}}_q(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} \\ -\dot{\boldsymbol{\sigma}}_q(\mathbf{q}, \dot{\mathbf{q}}, t)\dot{\mathbf{q}} - \dot{\mathbf{c}}(\mathbf{q}, \dot{\mathbf{q}}, t) - \mathbf{f}_k(\mathbf{q}, \mathbf{t}) \end{bmatrix}, \quad (57)$$

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. It has to be noticed that in the case of the so-called non-collocated systems one must be aware of singularity problems related to the coefficient hyper-matrix of the unknowns $\ddot{\mathbf{q}}$, $\boldsymbol{\lambda}$ and \mathbf{u} .

6 Comparison of the introduced control strategies

In this section, we give a numerical comparison of the introduced control algorithm for the benchmark problem presented in Fig. 1. The different test cases and their evaluations are summarized in table 1 and 2. The tested algorithms were the direct backward Euler discretization method (see section 5.1) and the 2-step and 1-step methods of Lagrange multipliers (see section 5.2). There were two different mathematical descriptions of the benchmark system: the case when the absolute Cartesian coordinates are used (Tab. 1), and the case when the relative coordinates are used (Tab. 2), as it is explained in details in section 4.

Additionally, all numerical simulations are carried out for four tasks. In the first task, the position of mass m_1 is prescribed by $z^d(t)$. The mathematical formulation of the servo-constraint is the same in the case of absolute and relative coordinates:

$$\boldsymbol{\sigma}_c(\mathbf{q}, t) = \hat{\boldsymbol{\sigma}}_c(\hat{\mathbf{q}}, t) = [z_1 - z^d(t)], \quad (58)$$

Since the servo constraint can be solved for the coordinate of m_1 and the actuator force is also applied to m_1 , this is a collocated case.

Similarly, the non-collocated case can also be formulated when the prescribed trajectory is related to the ‘‘passive’’ DoFs represented by m_2 and m_3 . When the position of m_2 is prescribed, the servo-constraint has the forms

$$\boldsymbol{\sigma}_{nc2}(\mathbf{q}, t) = [z_2 - z^d(t)], \quad (59)$$

$$\hat{\boldsymbol{\sigma}}_{nc2}(\hat{\mathbf{q}}, t) = [z_1 + \hat{z}_2 - z^d(t)], \quad (60)$$

in absolute and relative coordinate sets respectively. Call this case ‘‘non-collocated type 1’’. Similarly, when the position of mass m_3 is prescribed (‘‘non-collocated type 2’’) we can write:

$$\boldsymbol{\sigma}_{nc3}(\mathbf{q}, t) = [z_3 - z^d(t)], \quad (61)$$

$$\hat{\boldsymbol{\sigma}}_{nc3}(\hat{\mathbf{q}}, t) = [z_1 + \hat{z}_2 + \hat{z}_3 - z^d(t)]. \quad (62)$$

Finally, a general case is considered, when the servo-constraint is in relation with both the ‘‘active’’ and ‘‘passive’’ DoFs. The task is to keep the average position of mass m_1 and m_2 on the prescribed trajectory $z^d(t)$.

$$\boldsymbol{\sigma}_g(\mathbf{q}, t) = [z_1 + z_2 - 2z^d(t)], \quad (63)$$

$$\hat{\boldsymbol{\sigma}}_g(\hat{\mathbf{q}}, t) = [2z_1 + \hat{z}_2 - 2z^d(t)]. \quad (64)$$

Now, there is no unique solution for the servo-constraint, which means that there is no unique choice for controlled and uncontrolled coordinates.

Tab. 1 and 2 show the summary of the results of numerical simulations. Three qualitatively different behaviors of the controlled systems are distinguished. In the cases denoted by ‘‘A’’, the controller drives the system into the desired state without any problem, thus the servo-constraint is driven to zero even in the presence of an initial perturbation.

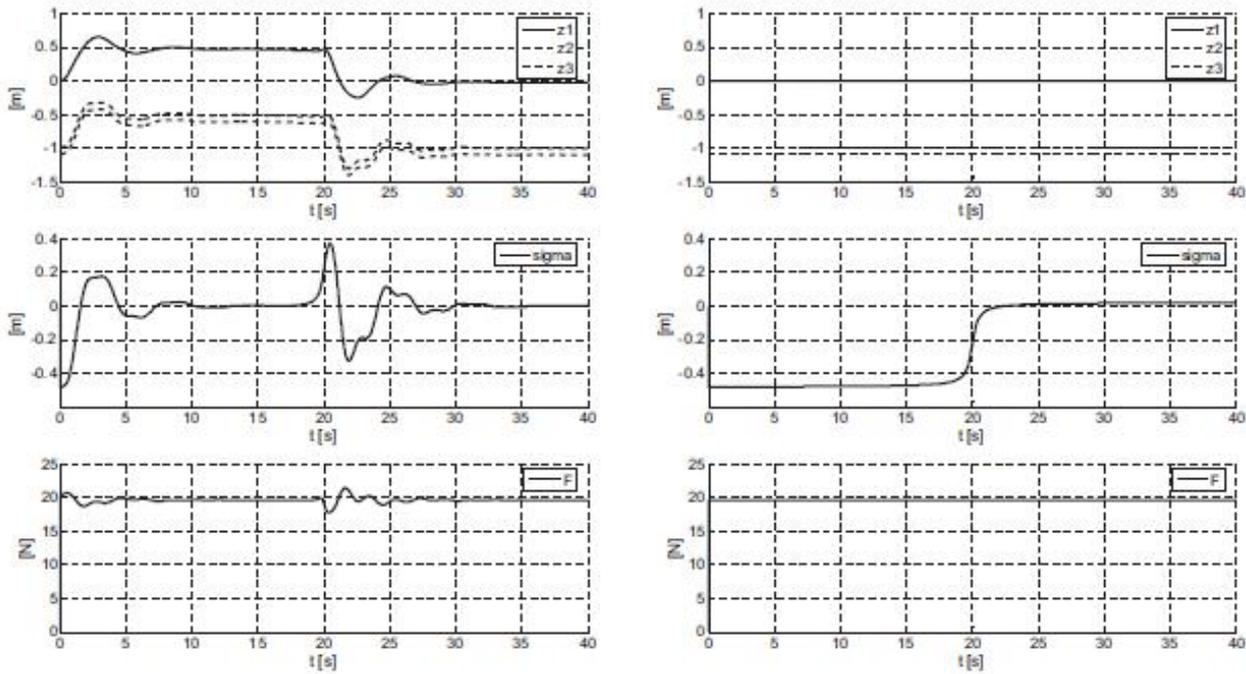


Fig. 2. Simulation results: relative coordinates (left column), absolute coordinates (right column)

Tab. 1. Benchmark of the control algorithms for absolute coordinates

	2step		1step
	BE	MLM	MLM
Collocated	A	A	A
Non-collocated, type 1	A	B	C
Non-collocated, type 2	A	B	C
General	A	A	A

Tab. 2. Benchmark of the control algorithms for relative coordinates

	2step		1step
	BE	MLM	MLM
Collocated	A	A	A
Non-collocated, type 1	A	A	C
Non-collocated, type 2	A	A	C
General	A	A	A

In the cases denoted by “C”, the controller encounters singularity problems, and it does not work. This problem appears only in the 1-step MLM method, when the coefficient matrix of the unknowns becomes singular (see equation (57)).

In cases denoted by “B”, the control algorithm seems to work without numerical problems, but the servo-constraint is not driven to zero, consequently, the control is unsuccessful. This occurs in case of the 2-step MLM method. This method is more robust against singularity problems than the 1-step MLM, because it utilizes null-space projection and pseudo-inverse calculation instead of inverse calculation, as shown in (55) and (56). In the cases “B”, equation (55) gives zero acceleration for mass m_1 , therefore the computed control force is constant. Because

of the null space projection, the controller loses the information about the desired motion of the “passive” DoF.

We can conclude that the direct discretization method is successful in all test cases, even in the so-called non-collocated cases. The 2-step MLM is applicable in some cases when the 1-step MLM fails. We can also observe that the change of the descriptor coordinates from absolute to relative ones influences the operation of the control algorithm in two cases: for the non-collocated case the 2-step MLM works successfully with relative coordinates. Note that the operation of the control algorithms is the same in type 1 and type 2 non-collocated cases.

Typical trajectories, and the time history of the servo-constraints and control inputs are presented in Fig. 2, where the left column shows a simulation “A” for a non-collocated 2-step MLM relative coordinate case, and the right column shows a simulation “B” for a non-collocated 2-step MLM absolute coordinate case.

7 Conclusion

The computed torque control of underactuated systems is a challenging task especially if the controlled mechanical system includes closed kinematic loops described by non-minimum set of coordinates. In the present study, we numerically investigated three different computed torque control algorithms for these systems with special attention to the choice of the descriptor coordinates of the controlled mechanical system.

All the introduced control algorithms used the servo-constraint on acceleration level with Baumgarte stabilization,

which is widely used in numerical simulation of multibody systems. In the cases of successful control strategies, the simulations showed that the servo-constraints were driven to zero. The detailed study of the several combinations of the choices of system coordinates, tasks and control algorithms showed that the proper choice of the descriptor coordinates may result successful control while the same control algorithm may fail with other choices of coordinates. The numerical simulations also showed the robustness of the backward Euler direct discretization of the control law, which works even for non-collocated systems with any choice of coordinates.

Acknowledgments

This work was supported by the Hungarian National Science Foundation (Grant No. OTKA K068910), by the HAS-BME Research Group on Dynamics of Machines and Vehicles and by the New Széchenyi Plan (Project ID: TÁMOP-4.2.1/B-09/1/KMR-2010-0002). Thanks to László Bencsik (Dept. of Applied Mechanics, Budapest Univ. of Technology and Economics) for his contribution.

References

- 1 **Goldstein H**, *Classical mechanics*, 2nd ed. Addison-Wesley, 1980.
- 2 **Oriolo G, Nakamura Y**, *Control of mechanical systems with second-order nonholonomic constraints: Underactuated manipulators*, 30th conference on decision and control, Brighton, England, 1991, pp. 2398–2403.
- 3 **Abdel-Rahman E., Nayfeh A., Masoud Z.**, *Dynamics and control of cranes: A review*, Journal of Vibration and Control **9** (2003), no. 7, 863–908.
- 4 **Omar H., Nayfeh A.**, *Control of gantry and tower cranes*, Ph.D. Thesis, 2003.
- 5 **Stépán G et al.**, *A ceiling based crawling, hoisting and swinging service robot platform*, Proceedings of beyond gray droids: Domestic robot design for the 21st century workshop at hci 2009, 2009.
- 6 **Lammerts I.**, *Adaptive computed reference computed torque control*, Ph.D. Thesis, 1993.
- 7 **Blajer W, Kolodziejczyk K**, *Modeling of underactuated mechanical systems in partly specified motion*, Journal of Theoretical and Applied Mechanics **46** (2008), no. 2, 383–394.
- 8 **de Jalón J., Bayo E**, *Kinematic and dynamic simulation of multibody systems: the real-time challenge*, Springer-Verlag, 1994.
- 9 **Tackenberg P.**, *Control methods for vibration reduction in an underactuated nonlinear beam system*, Master's Thesis, 1997.
- 10 **Luca D. A, R.Mattone, G.Oriolo**, *Control of redundant robots under end-effector commands: A case study in underactuated systems*, Applied Mathematics and Computer Science **7** (1997), no. 2, 225–251.
- 11 **Olfati-SR**, *Nonlinear control of underactuated mechanical systems with application to robotics and aerospace vehicles*, Ph.D. Thesis, 2001.
- 12 **Aneke N.**, *Control of underactuated mechanical systems*, Ph.D. Thesis, 2002.
- 13 **Bommer A**, *Control of underactuated mechanical systems*, Technische Universiteit Eindhoven, 2004.
- 14 **Kövecses J, Piedoboeuf J., Lange C**, *Dynamic modeling and simulation of constrained robotic systems*, IEEE/ASME Transactions on mechatronics **8** (2003), no. 2, 165–177, DOI 10.1109/TMECH.2003.812827.
- 15 **Isidori A**, *Nonlinear control systems*, 2nd ed. Springer, 1999.
- 16 **Kovács L., Zelei A, Bencsik L, Turi J, Stépán G**, *Motion control of an under-actuated service robot using natural coordinates*, Proceedings of romansy 18 – robot design, dynamics and control, 2010, pp. 331–338.
- 17 **Zelei A, Kovács L., Stépán G**, *Computed torque control of an under-actuated service robot platform modeled by natural coordinates*, Commun Nonlinear Sci Numer Simulat **16** (2011), no. 5, 2205–2217, DOI 10.1016/j.cnsns.2010.04.060.
- 18 **Baumgarte J**, *Stabilization of constraints and integrals of motion in dynamical systems*, Computer methods in applied mechanics and engineering **1** (1972), no. 1, 1–16.
- 19 **Bencsik L., Kovács L. L.**, *Stability case study of an underactuated service robot*, Dynamical systems - nonlinear dynamics and control, August DSTA Dec. 5, pp. 89–94.