

# Underactuation Induced by Actuator Saturation

A. Zelei\*

Budapest University of Technology and Economics  
Hungary

G. Stepan†

Budapest University of Technology and Economics  
Hungary

**Abstract**—*The saturation of actuators is an essential nonlinearity, and the consideration of the bounded actuator torques during the design of the computed torque control (CTC) is a challenging task. In this study, the saturation of the actuator torques is treated as a temporary reduction of the number of independent control inputs. Consequently, a manipulator, which is fully actuated in the neighbourhood of its desired motion, becomes underactuated when the intricate combinations of the actuator saturations occur. The cascade of two CTC algorithms is applied to resolve the problem of the temporary underactuation: the classical CTC method is self-interrupted and exchanged to a generalized CTC that is extended to underactuated systems. The corresponding control algorithm is applicable even in those cases of multi-body problems where the use of non-minimum set of coordinates provides the computationally efficient mathematical description together with servo-constraint based task definitions leading to differential algebraic equations.*

**Keywords:** underactuated systems, actuator saturation, constrained manipulators, redundancy, servo-constraints

## I. Introduction

Robotic structures may change their topology during their operation for several reasons. The cause of the topology variation could be related to the variation of the contact points between the robot and the environment, or to the intermittent variation in the number of the actually controllable actuators. The present work is about the second case: the change of the topology is due to the loss of some of the accessible actuators because they reached their saturation levels.

Every driver applied in robotic systems has some limitations which are typically speed, power and/or torque limits. These limitations can be included in the mathematical models and so they can be taken into account already during the design of the task. Still, the actuator torque saturation may cause essential problems when a manipulator performs the tracking of a desired trajectory. Clearly, the risk of actuator saturation is higher, for example, when high accelerations are prescribed.

Several CTC based control algorithms can be found in the literature, which take into account the limited actua-

tor torques. In [1], a continuous-time predictive control approach is used to derive the nonlinear constrained control law for trajectory tracking control in the presence of actuator saturation. The proposed method is limited for those systems that are input-output feedback-linearizable after a specific treatment called dynamic expansion. The resulting control minimizes the tracking errors even with saturated actuators. In reference [2], an adaptive full-state feedback controller as well as an exact-model-knowledge output feedback controller are designed, and a comparative numerical analysis is carried out to demonstrate the benefits of the two proposed controllers. On the basis of the classical CTC method, a composite nonlinear feedback design method is presented in [3] for robot manipulators with bounded torques at the actuators. The controller consists of two loops. The inner loop is for the full compensation of the manipulator's nonlinear dynamics, while the outer loop is the composite nonlinear feedback controller for stabilization and performance enhancement.

The above mentioned control approaches handle the actuator saturation as a nonlinearity of the system. Alternatively, actuator saturation can also be modelled as the decrement of the number of accessible control inputs, which is practically equivalent to the variation of the manipulator's topology.

In the subsequent sections, first, the classical computed torque control (CTC) method (see, for example, [4, 5]) is formulated briefly when non-minimum set of descriptor coordinates (in other terminology, redundant set of generalized coordinates) is used and the task is defined by servo-constraints. Then extended CTC methods (see [6–8]) are introduced for underactuated systems. In order to handle actuator saturation in fully- and/or underactuated manipulators, a specific combination of the classical CTC method and one of the extended CTC methods (see [8]) is developed, which is applicable even for complex multi-body systems that are modeled by non-minimum set of descriptor coordinates. After the discussion of the possible ways of the dimension reduction of servo-constraints, the efficiency of the proposed combined CTC algorithm is demonstrated in the case study of an RR manipulator with essential actuator saturations.

## II. Classical and extended CTC algorithms

In case of serial and fully actuated robotic manipulators, an independent control input is associated with each de-

\*zelei@mm.bme.hu  
†stepan@mm.bme.hu

MTA-BME Research Group on  
Dynamics of Machines and Vehicles

gree of freedom (DOF). Consequently, the classical CTC method can easily be applied for such systems, especially when they are modeled in the classical way of using minimum set of generalized coordinates, and the corresponding equation of motion has an ordinary differential equation (ODE) form [4, 5]. This is not the case for underactuated robotic manipulators.

The computed torque control method was generalized for underactuated systems by [6]: it was named computed desired computed torque control (CDCTC) method, where the term “desired” refers to the fact that the desired values of a set of uncontrolled coordinates have to be calculated first, and after the calculation of this desired zero dynamics, the control inputs can be determined. This method requires the separation of the generalized coordinates into controlled and uncontrolled ones. Partial feedback linearization can also be used for the control of underactuated systems [9]. The main idea of the method is to substitute the original nonlinear system with a partially equivalent linear system by means of a non-linear transformation. The CTC method for underactuated systems can be further generalized for systems modeled by non-minimum set of descriptor coordinates when the mathematical model is extended with geometric constraint equations and arranged into a system of differential algebraic equations (DAE).

The subsections below detail a servo-constraint based CTC method for constrained systems described by non-minimum set of descriptor coordinates. Then the method is applied for underactuated systems with unbounded control inputs, while the application for bounded systems is explained in the subsequent section III.

#### A. Problem formulation

When multi-body systems are described by non-minimum set of descriptor coordinates  $\mathbf{q} \in \mathbb{R}^n$ , geometric constraint equations provide the specific connections of the redundant descriptor coordinates. These additional equations are represented by  $\varphi(\mathbf{q}, t) \in \mathbb{R}^m$ , so the system has  $n - m$  DoF. The Lagrangian equation of the first kind has the form

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) + \varphi_{\mathbf{q}}^T(\mathbf{q}, t)\boldsymbol{\lambda} = \mathbf{H}(\mathbf{q})\mathbf{u} \quad (1)$$

$$\varphi(\mathbf{q}, t) = \mathbf{0}, \quad (2)$$

which is a DAE [10, 11]. is a positive definite mass matrix. The descriptor coordinates are chosen intuitively, but if they are chosen properly like in case of the so-called natural coordinates (see [11]), the mass matrix  $\mathbf{M}(\mathbf{q}) \in \mathbb{R}^{n \times n}$  is a constant matrix  $\mathbf{M}(\mathbf{q}) \equiv \mathbf{M}$ . This is a relevant observation when the advantages of modeling by DAE are listed. The vector  $\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) \in \mathbb{R}^n$  describes the inertial, gyroscopic, Coriolis terms and all external forces, including gravity, spring and damping forces if present.

The Jacobian matrix associated with the geometric constraints is presented by and  $\varphi_{\mathbf{q}}(\mathbf{q}, t) = \partial\varphi(\mathbf{q})/\partial\mathbf{q} \in$

$\mathbb{R}^{m \times n}$ . The corresponding Lagrange multipliers are collected in the time dependent vector  $\boldsymbol{\lambda} \in \mathbb{R}^m$ ; these provide the time-history of the constraint forces corresponding to the additional geometric constraints, so their calculation will not be important from control view-point.

The  $l$  dimensional control input vector is  $\mathbf{u} \in \mathbb{R}^l$  and  $\mathbf{H}(\mathbf{q}) \in \mathbb{R}^{n \times l}$  is the generalized control input matrix. If the number  $l$  of the control inputs is less than the  $n - m$  DoF of the system, then it is called *underactuated*, while if  $l = n - m$  than the system is *fully actuated*.

The task of the manipulator is defined in the form of holonomic and rheonomic constraint equations called *servo-constraints* or *control-constraints* [7, 12–15]. This way, any kind of manipulator tasks can be handled similarly to the geometric constraints in (2). By means of the servo-constraint vector  $\boldsymbol{\sigma}(\mathbf{q}, t) \in \mathbb{R}^l$ , the servo-constraint equation can be written as:

$$\boldsymbol{\sigma}(\mathbf{q}, t) = \mathbf{0}. \quad (3)$$

We assume that the investigated underactuated system has desired outputs of the same number  $l$  as inputs. In spite of the fact that the inverse dynamical calculation leads to the solution of a system of DAE, the desired control inputs can be determined uniquely by the method of computed torques [7, 8, 16]. Reference [17] mentions that the classical Lagrangian multiplier technique works only for independent constraints, where the constraint Jacobian is a full row rank matrix. Considering this, we assume that the servo-constraints are *linearly independent*. Besides, we assume that they are also *consistent*, that is, there are no contradictory constraints, and first, we also assume that they can be satisfied with *bounded control inputs*.

After the introduction of servo-constraint equations, the number  $n$  of independent descriptor coordinates are constrained by the same number  $n = m + l$  constraint equations in fully actuated cases. When  $n > m + l$  in underactuated systems, a part of the dynamics is independent from the geometric and the servo-constraints, which is also called *zero dynamics*.

Our goal is to determine the input vector  $\mathbf{u}$ , which requires the determination of the desired values of the descriptor coordinates in  $\mathbf{q}$ , and adjunctively the vector  $\boldsymbol{\lambda}$  of Lagrange multipliers, which all satisfy the DAE system (1), (2) and (3). While in some simple cases this goal can be achieved analytically, efficient numerical methods have to be used in general cases in order to carry out the calculations in each sampling period. In order to explain the difficulties related to the solution of these systems, consider first the case of fully actuated systems.

#### B. Servo-constraint based CTC for fully actuated systems

To provide a general overview of the difference between the inverse dynamic calculation of fully actuated and underactuated systems, let us consider an unconstrained dynam-

ical system with the complementary servo-constraint equation:

$$\mathbf{M}\ddot{\bar{\mathbf{q}}} + \mathbf{C}(\bar{\mathbf{q}}, \dot{\bar{\mathbf{q}}}) = \mathbf{H}(\bar{\mathbf{q}})\mathbf{u}, \quad (4)$$

$$\boldsymbol{\sigma}(\bar{\mathbf{q}}, t) = \mathbf{0}, \quad (5)$$

where  $\bar{\mathbf{q}} \in \mathbb{R}^n$  is now the vector of minimum set generalized coordinates.

The desired values in  $\bar{\mathbf{q}}$  can be obtained from the servo-constraint equation (5) as the function of time. However, the servo-constraint vector  $\boldsymbol{\sigma}(\bar{\mathbf{q}}, t)$  is usually a nonlinear function of  $\bar{\mathbf{q}}$ , so numerical methods should be applied at this point. After this, the control input can easily be calculated, because in case of unconstrained, fully actuated systems, the control input matrix  $\mathbf{H}(\bar{\mathbf{q}}) \in \mathbb{R}^{n \times n}$  is invertible [5]:

$$\mathbf{u} = \mathbf{H}^{-1}(\bar{\mathbf{q}}) [\mathbf{M}\ddot{\bar{\mathbf{q}}} + \mathbf{C}(\bar{\mathbf{q}}, \dot{\bar{\mathbf{q}}})]. \quad (6)$$

The inverse dynamical calculation becomes somewhat more challenging if the fully actuated system is described by non-minimum set of descriptor coordinates  $\mathbf{q} \in \mathbb{R}^n$ , and geometric constraints are introduced as the governing equations (1) and (2) show. In such cases, the geometric constraint equation (2) and the servo constraint equation (3) are both needed to obtain the desired values of the descriptor coordinates  $\mathbf{q} \in \mathbb{R}^n$ . In contrast with the unconstrained systems, here, the control input can not be calculated with the inverse of  $\mathbf{H}(\mathbf{q}) \in \mathbb{R}^{n \times l}$ , because it is not a square matrix. The algorithm introduced in the following subsection will resolve this problem.

### C. Servo-constraint based CTC for underactuated systems

In order to solve the inverse dynamic problem for underactuated systems where even non-minimum set of descriptor coordinates are used, the application of the direct backward Euler discretization of the DAE system (1), (2) and (3) is proposed here. The backward Euler method requires the solution of a system of nonlinear algebraic equations in each sampling time step of the control. This can efficiently be carried out by NewtonRaphson iterations. The method works even in those cases when the separation of the coordinates into controlled and uncontrolled ones (see [6]) is not possible [16].

Transform the unconstrained dynamic equation (1) into a first order system via introducing the new variable  $\mathbf{y} = \dot{\mathbf{q}}$ . Then we consider the geometric constraint equation (2) and the stabilized second time derivative of the servo-constraint equation (3). After that, the control law will be obtained for  $\mathbf{u} \in \mathbb{R}^l$  as a solution of the  $2n + l + m$  dimensional DAE

system:

$$\dot{\mathbf{q}} = \mathbf{y}, \quad (7)$$

$$\dot{\mathbf{y}} = -\mathbf{M}(\mathbf{q})^{-1} [\mathbf{C}(\mathbf{q}, \mathbf{y}) + \boldsymbol{\varphi}_{\mathbf{q}}^T(\mathbf{q})\boldsymbol{\lambda} - \mathbf{H}(\mathbf{q})\mathbf{u}], \quad (8)$$

$$\boldsymbol{\varphi}(\mathbf{q}) = \mathbf{0}, \quad (8)$$

$$\begin{aligned} &\boldsymbol{\sigma}_{\mathbf{q}}(\mathbf{q}, t)\dot{\mathbf{y}} + \dot{\boldsymbol{\sigma}}_{\mathbf{q}}(\mathbf{q}, \mathbf{y}, t)\mathbf{y} + \dot{\boldsymbol{\sigma}}_t(\mathbf{q}, \mathbf{y}, t) + \\ &\quad \mathbf{K}_{\alpha}[\boldsymbol{\sigma}_{\mathbf{q}}(\mathbf{q}, t)\mathbf{y} + \boldsymbol{\sigma}_t(\mathbf{q}, t)] + \\ &\quad \mathbf{K}_{\beta}\boldsymbol{\sigma}(\mathbf{q}, t) = \mathbf{0}, \end{aligned} \quad (9)$$

where the subscripts for  $\mathbf{q}$  and  $t$  refer to corresponding partial derivatives, and  $\mathbf{u} \in \mathbb{R}^l$ ,  $\mathbf{q} \in \mathbb{R}^n$ ,  $\mathbf{y} \in \mathbb{R}^n$  and  $\boldsymbol{\lambda} \in \mathbb{R}^m$  are the unknowns of same number as equations. The gain matrices  $\mathbf{K}_{\alpha}$  and  $\mathbf{K}_{\beta}$  are chosen either by trial and error, or by certain optimization algorithms that are designed for the discretized version of the calculation.

In order to get the discretized version, the problem is reformulated in the following way. The backward Euler discretization of the DAE system (7-9) results in a system of nonlinear algebraic equations:

$$\mathbf{q}_i^d - \mathbf{q}_{i-1} - h\mathbf{y}_i^d = 0, \quad (10)$$

$$\begin{aligned} &\mathbf{y}_i^d - \mathbf{y}_{i-1} + h\mathbf{M}(\mathbf{q}_i^d)^{-1} \times \\ &\quad (\mathbf{C}(\mathbf{q}_i^d, \mathbf{y}_i^d) + \boldsymbol{\varphi}_{\mathbf{q}}^T(\mathbf{q}_i^d)\boldsymbol{\lambda}_i - \mathbf{H}(\mathbf{q}_i^d)\mathbf{u}_i) = 0, \\ &\boldsymbol{\varphi}(\mathbf{q}_i^d) = \mathbf{0}, \end{aligned} \quad (11)$$

$$\begin{aligned} &\boldsymbol{\sigma}_{\mathbf{q}}(\mathbf{q}_i^d, t_i)(\mathbf{y}_i^d - \mathbf{y}_{i-1}) + \\ &\quad \dot{\boldsymbol{\sigma}}_{\mathbf{q}}(\mathbf{q}_i^d, \mathbf{y}_i^d, t_i)\mathbf{y}_i^d + \dot{\boldsymbol{\sigma}}_t(\mathbf{q}_i^d, \mathbf{y}_i^d, t_i) + \\ &\quad \mathbf{K}_{\alpha}[\boldsymbol{\sigma}_{\mathbf{q}}(\mathbf{q}_i^d, t_i)\mathbf{y}_i^d + \boldsymbol{\sigma}_t(\mathbf{q}_i^d, t_i)] + \\ &\quad \mathbf{K}_{\beta}\boldsymbol{\sigma}(\mathbf{q}_i^d, t_i) = \mathbf{0}, \end{aligned} \quad (12)$$

where  $h$  denotes the sampling time used for the time discretization. In this discretized form, the unknowns are the  $i^{\text{th}}$  values of the desired coordinates  $\mathbf{q}_i^d$ , their time derivatives  $\mathbf{y}_i^d$ , the control inputs  $\mathbf{u}_i$  and the Lagrange multipliers  $\boldsymbol{\lambda}_i$ , while  $\mathbf{q}_{i-1}$  and  $\mathbf{y}_{i-1}$  are known as the measured values at the previous sampling instant, that is, at the preceding  $(i-1)^{\text{st}}$  time step. This system can also be formulated in a compact form  $\mathbf{F}(\mathbf{z}_i) = \mathbf{0}$  for the vector of unknowns:

$$\mathbf{z}_i = [\mathbf{q}_i^d, \mathbf{y}_i^d, \mathbf{u}_i, \boldsymbol{\lambda}_i]^T. \quad (13)$$

The only important part of the solution is the control input  $\mathbf{u}_i$  since  $\mathbf{q}_i^d$ ,  $\mathbf{y}_i^d$  and  $\boldsymbol{\lambda}_i$  are dropped in the next timestep: the coordinates and velocities will be substituted by the measured ones and the constraining forces are not needed at all.

The nonlinear algebraic equation (13) is solved by Newton-Raphson iteration. The  $(j+1)^{\text{st}}$  estimation for the unknown vector  $\mathbf{z}_i$  in the  $i^{\text{th}}$  time step is expressed as:

$$\mathbf{z}_i^{j+1} = \mathbf{z}_i^j - \mathbf{J}^{-1}(\mathbf{z}_i^j)\mathbf{F}(\mathbf{z}_i^j); \quad j = 0, 1, \dots, n_{\text{NR}}. \quad (14)$$

If the initial estimations  $\mathbf{z}_i^0 = \mathbf{z}_{i-1}^{n_{\text{NR}}}$  are used from the previous time step, then accurate enough results are obtained in a few steps of iterations, that is  $n_{\text{NR}}$  is usually not larger

than 2 or 3. If the above iteration process is simplified by the use of the initial approximation of the inverse Jacobian in the form:

$$\mathbf{z}_i^{j+1} = \mathbf{z}_i^j - \mathbf{J}^{-1}(\mathbf{z}_i^0) \mathbf{F}(\mathbf{z}_i^j); \quad j = 0, 1, \dots, n_{\text{NR}}, \quad (15)$$

we still obtain an accurate enough result in 2-3 iteration steps, and the whole numerical calculation becomes fast enough that it is manageable even in on-line control algorithms by providing the control input  $\mathbf{u}_i$  within  $\mathbf{z}_i^{n_{\text{NR}}}$  in each sampling time instant.

### III. Combined fully- and underactuated CTC algorithm for handling saturation

In the proposed concept, the control algorithm first calculates the desired control input vector  $\mathbf{u} \in \mathbb{R}^l$  for all actuators, and then it checks whether the value of each control input  $u_i$ ,  $i = 1 \dots l$  exceeds the maximum or minimum limiting values  $u_i^+$  or  $u_i^-$ , respectively (see Fig. 1).

If some actuators saturate, then the number of the non-saturated actuators is reduced to  $\hat{l}$ . The saturated actuators provide a constant torque  $u_i^+$  or  $u_i^-$ , and the algorithm recalculates the desired control inputs  $\hat{\mathbf{u}} \in \mathbb{R}^{\hat{l}}$  with  $\hat{l} < l$  as an underactuated system. The number  $\hat{l}$  of the still accessible non-saturated control inputs is always less than the number of DoF, so the system is actually underactuated. For the  $l - \hat{l}$  number of saturated actuators, the limiting values  $u_i^+$  or  $u_i^-$  are commanded by the controller. This operation is repeated in each sampling time step again and again if new and new actuators saturate, but it stops if there are no further actuators to saturate, or if all the actuators are saturated.

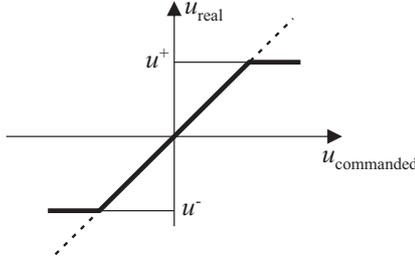


Fig. 1. Actuator saturation: nonlinear connection between commanded and real control input

The partitioning of the control input vector into saturated and non-saturated parts leads to the following structure of the equation of motion:

$$\mathbf{M}\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) + \varphi_{\mathbf{q}}^{\text{T}}(\mathbf{q})\boldsymbol{\lambda} = \mathbf{H}(\mathbf{q})\mathbf{R}\mathbf{u}^{\pm} + \mathbf{H}(\mathbf{q})\mathbf{T}\hat{\mathbf{u}}, \quad (16)$$

where  $\mathbf{u}^{\pm}$  contains the corresponding maximal and minimal values  $u_i^+$  or  $u_i^-$ , while  $\hat{\mathbf{u}} \in \mathbb{R}^{\hat{l}}$  contains the accessible control inputs,  $\mathbf{T} \in \mathbb{R}^{l \times \hat{l}}$  and  $\mathbf{R} \in \mathbb{R}^{l \times l}$  are selector matrices defined by

$$\mathbf{u} = \mathbf{R}\mathbf{u}^{\pm} + \mathbf{T}\hat{\mathbf{u}}. \quad (17)$$

In (16) and (17), the selector matrix  $\mathbf{R}$  collects the saturated control inputs, and  $\mathbf{T}$  identifies the non-saturated (still accessible) control input vector as:

$$\hat{\mathbf{u}} = \mathbf{T}^{\text{T}}\mathbf{u}. \quad (18)$$

In (16), the term  $\mathbf{H}(\mathbf{q})\mathbf{R}\mathbf{u}^{\pm}$  is a known, constant external force vector, while the term  $\mathbf{H}(\mathbf{q})\mathbf{T}\hat{\mathbf{u}}$  is responsible for the actuation of the system. Thus, we can introduce a reduced size control input matrix  $\hat{\mathbf{H}}(\mathbf{q}) \in \mathbb{R}^{n \times \hat{l}}$  for the saturated system as

$$\hat{\mathbf{H}}(\mathbf{q}) = \mathbf{H}(\mathbf{q})\mathbf{T} \quad (19)$$

and the new vector of inertial forces in the form

$$\hat{\mathbf{C}}(\mathbf{q}, \dot{\mathbf{q}}) = \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) - \mathbf{H}(\mathbf{q})\mathbf{R}\mathbf{u}^{\pm}. \quad (20)$$

This way, equation (16) assumes the form

$$\mathbf{M}\ddot{\mathbf{q}} + \hat{\mathbf{C}}(\mathbf{q}, \dot{\mathbf{q}}) + \varphi_{\mathbf{q}}^{\text{T}}(\mathbf{q})\boldsymbol{\lambda} = \hat{\mathbf{H}}(\mathbf{q})\hat{\mathbf{u}}, \quad (21)$$

which is fully compatible with the form (1) used in the problem formulation of underactuated systems described by redundant set of coordinates.

It is still true, however, that the inverse calculation can be unique only if the dimension of the servo-constraint vector (that is, the dimension of the task) is equal to the number  $\hat{l}$  of accessible control inputs  $\hat{\mathbf{u}}$ . Consequently, a reduced size servo-constraint vector has to be defined for the case when some of the actuators saturate, which practically means the redesign of the desired task. In the saturated cases, we use this reduced size servo-constraint vector  $\hat{\boldsymbol{\sigma}}(\mathbf{q}, t) \in \mathbb{R}^{\hat{l}}$  instead of the original  $\boldsymbol{\sigma}(\mathbf{q}, t) \in \mathbb{R}^l$  used in (3).

The dimension reduction of the servo-constraint vector is a critical step because the transformation between  $\hat{\boldsymbol{\sigma}}(\mathbf{q}, t)$  and  $\boldsymbol{\sigma}(\mathbf{q}, t)$  is not unique: several optimization techniques can be used, which are explained in the following section.

### IV. Dimension reduction of servo-constraints

The dimension reduction of the desired task in case of actuator saturations is introduced first for specific, then for general cases.

#### A. Specific case

In the equation of motion (1) of controlled dynamical systems, the term  $\mathbf{H}(\mathbf{q})\mathbf{u}$  represents the control force. This control force can also be viewed as a constraining force (or “control reaction force”) interpreted by means of a vector  $\boldsymbol{\lambda}_u \in \mathbb{R}^l$  of Lagrange multipliers, which are associated with the servo-constraint  $\boldsymbol{\sigma}(\mathbf{q}, t)$  and its Jacobian  $\boldsymbol{\sigma}_{\mathbf{q}}^{\text{T}}(\mathbf{q}, t)$  (see [7] and [13]). Using this concept, the equation of motion can be rewritten in the form:

$$\mathbf{M}\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) + \varphi_{\mathbf{q}}^{\text{T}}(\mathbf{q})\boldsymbol{\lambda} + \boldsymbol{\sigma}_{\mathbf{q}}^{\text{T}}(\mathbf{q}, t)\boldsymbol{\lambda}_u = \mathbf{0} \quad (22)$$

considering that

$$\sigma_{\mathbf{q}}^T(\mathbf{q}, t)\lambda_u = -\mathbf{H}(\mathbf{q})\mathbf{u}. \quad (23)$$

The connection of the control input  $\mathbf{u}$  and the multiplier  $\lambda_u$  is trivial if the servo-constraint Jacobian and the negative control input matrix are just equal:

$$\sigma_{\mathbf{q}}^T(\mathbf{q}, t) = -\mathbf{H}(\mathbf{q}), \quad (24)$$

$$\mathbf{u} = \lambda_u. \quad (25)$$

This is called ‘‘specific case’’ when the directions defined by the control input matrix and the servo-constraint Jacobian are the same.

The structure of the matrix  $\mathbf{H}(\mathbf{q})$  definitely depends on the mechanical design of the system and the chosen set of generalized coordinates, while  $\sigma_{\mathbf{q}}(\mathbf{q}, t)$  depends on the task description. In these specific cases, the effect of the control inputs on the system coordinates is the same as the effect of the servo-constraints on the motion: the correspondence of the control inputs and the servo-constraints is trivial. Clearly, the above conditions can be satisfied in specific cases only (see examples later in section V).

In the saturated system, the control input matrix of the still non-saturated inputs is defined by equation (19). In the specific case, the connection of the reduced servo-constraint Jacobian  $\hat{\sigma}_{\mathbf{q}}(\mathbf{q}, t)$  and the reduced size control input matrix  $\hat{\mathbf{H}}(\mathbf{q})$  should be defined as equation (24) does:

$$\hat{\sigma}_{\mathbf{q}}^T(\mathbf{q}, t) = -\hat{\mathbf{H}}(\mathbf{q}). \quad (26)$$

Considering equation (19) we can write:

$$\hat{\sigma}_{\mathbf{q}}^T(\mathbf{q}, t) = \sigma_{\mathbf{q}}^T(\mathbf{q}, t)\mathbf{T}. \quad (27)$$

From (27), the suggested reduced servo-constraint is simply:

$$\hat{\sigma}(\mathbf{q}, t) = \mathbf{T}^T\sigma(\mathbf{q}, t). \quad (28)$$

Note that in these special cases, the servo-constraint reduction transformation is the same as the transformation between the full control input vector and the vector of the non-saturated control inputs in equation (18).

### B. General case

In general cases, the servo-constraint Jacobian and the negative control input matrix are not equal:

$$\sigma_{\mathbf{q}}^T(\mathbf{q}, t) \neq -\mathbf{H}(\mathbf{q}), \quad (29)$$

which means that the directions defined by the control input matrix and the servo-constraint Jacobian are not necessarily the same. If minimum set of generalized coordinates are used, the transformation between  $\mathbf{u}$  and  $\lambda_u$  can be deduced from (23) as

$$\lambda_u = -(\sigma_{\mathbf{q}}^T(\bar{\mathbf{q}}, t))^{-1}\mathbf{H}(\bar{\mathbf{q}})\mathbf{u}. \quad (30)$$

The reduced servo constraint then can be obtained by means of the same transformation:

$$\hat{\sigma}(\bar{\mathbf{q}}, t) = -\mathbf{T}^T\mathbf{H}(\bar{\mathbf{q}})^{-1}\sigma_{\mathbf{q}}^T(\bar{\mathbf{q}}, t)\sigma(\bar{\mathbf{q}}, t), \quad (31)$$

which simplifies to (28) in the special case of (24).

If non-minimum set of descriptor coordinates are used and/or some actuators saturate, the same procedure should be repeated for an already constrained and/or underactuated system, respectively. Since the matrices are non-symmetric in these cases, the transformation between  $\mathbf{u}$  and  $\lambda_u$  can be deduced from (23) with the use of the Moore-Penrose generalized inverse (pseudo-inverse). Although the transformation is not unique, we can choose, for example, the formula:

$$\lambda_u = -(\sigma_{\mathbf{q}}^T(\mathbf{q}, t))^\dagger\mathbf{H}(\mathbf{q})\mathbf{u}. \quad (32)$$

The pseudo-inverse calculation itself is not unique either, although the standard optimized calculation

$$(\sigma_{\mathbf{q}}^T)^\dagger = (\sigma_{\mathbf{q}}\sigma_{\mathbf{q}}^T)^{-1}\sigma_{\mathbf{q}}, \quad (33)$$

is widely used for convenience.

Similarly to the above cases, equation (32) generates the reduced servo-constraint in the form

$$\hat{\sigma}(\mathbf{q}, t) = \hat{\Gamma}^\dagger(\mathbf{q}, t)\sigma(\mathbf{q}, t), \quad (34)$$

where  $\hat{\Gamma}(\mathbf{q}, t)$  is defined as

$$\hat{\Gamma}(\mathbf{q}, t) = -(\sigma_{\mathbf{q}}^T(\mathbf{q}, t))^\dagger\mathbf{H}(\mathbf{q})\mathbf{T}. \quad (35)$$

In the following case study, this formula will also be used for the dimension reduction of the servo-constraint.

## V. Case study for an RR manipulator

Numerical simulations were accomplished for a two-link (SCARA-type) manipulator shown in Fig. 2, which consists of two homogeneous prismatic bars with parameters  $m_1 = 0.2$  kg,  $L_1 = 0.4$  m,  $m_2 = 0.2$  kg and  $L_2 = 0.4$  m. The manipulator moves in the horizontal plane.

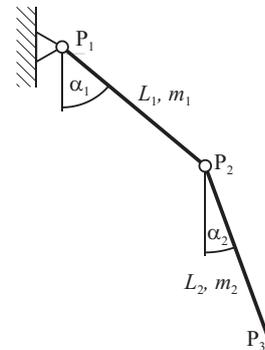


Fig. 2. Mechanical model of the studied RR manipulator

The geometric description and the derivation of the equations of motion for the chosen system would be straightforward with the minimum set of generalized coordinates  $\alpha_1$  and  $\alpha_2$ . Still, in order to show the applicability of the generalized CTC algorithm, a non-minimum set of descriptor coordinates is used. Thus, the Cartesian coordinates  $\mathbf{q} = [x_1, y_1, x_2, y_2, x_3, y_3]^T$  of the endpoints of the bars are chosen as descriptor coordinates of number  $n = 6$ . Consequently, a 4 dimensional geometric constraint vector is introduced in the form:

$$\varphi(\mathbf{q}) = \begin{bmatrix} x_1 \\ y_1 \\ (x_2 - x_1)^2 + (y_2 - y_1)^2 - L_1^2 \\ (x_3 - x_2)^2 + (y_3 - y_2)^2 - L_2^2 \end{bmatrix}. \quad (36)$$

Both joints  $P_1$  and  $P_2$  are actuated, so the  $l = 2$  dimensional control input vector is  $\mathbf{u} = [\tau_1 \ \tau_2]^T$  with torques  $\tau_{1,2}$ . The actuators saturate when the actuator torques reach the limiting values:  $u_1^\pm = \pm 0.04 \text{ Nm}$  and  $u_2^\pm = \pm 0.04 \text{ Nm}$ .

Following the derivation detailed in [11], the control input matrix  $\mathbf{H}(\mathbf{q})$  is obtained after the virtual substitution of the control torques by pairs of forces applied at the base points:

$$\mathbf{H}(\mathbf{q}) = \begin{bmatrix} -\frac{y_1 - y_2}{L_1^2} & 0 \\ \frac{x_1 - x_2}{L_1^2} & 0 \\ \frac{y_1 - y_2}{L_1^2} & -\frac{y_2 - y_3}{L_2^2} \\ -\frac{x_1 - x_2}{L_1^2} & \frac{x_2 - x_3}{L_2^2} \\ 0 & \frac{y_2 - y_3}{L_2^2} \\ 0 & -\frac{x_2 - x_3}{L_2^2} \end{bmatrix}. \quad (37)$$

In the following subsections the manipulator is subjected to two tasks defined by two servo-constraints. In the first specific case, the angles  $\alpha_1$  and  $\alpha_2$  of the bars are prescribed in time by  $\alpha_1^d(t)$  and  $\alpha_2^d(t)$ . The desired initial and end configurations are shown in the left panel of Fig. 3. In the second case study, a linear path of the endpoint  $P_3$  of the manipulator is prescribed (see the right panel of Fig. 3). The transitions from the initial to the end configurations are defined in time by arc tangent functions in both cases.

#### A. Specific case

By intuition, we can define the servo-constraint in the form

$$\sigma(\mathbf{q}, t) = \begin{bmatrix} -\tan^{-1}\left(\frac{x_1 - x_2}{y_1 - y_2}\right) - \alpha_1^d(t) \\ -\tan^{-1}\left(\frac{x_2 - x_3}{y_2 - y_3}\right) - \alpha_2^d(t) \end{bmatrix}. \quad (38)$$

A lengthy calculation shows that  $\sigma_{\mathbf{q}}^T(\mathbf{q}, t) = -\mathbf{H}(\mathbf{q})$ , that is, (24) is satisfied. The above intuition can be explained physically since the servo-constraints are related to the angles of the bars, and the control torques also act directly at the same angles.

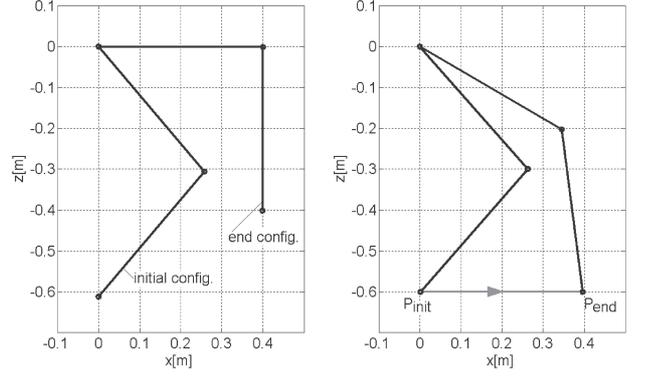


Fig. 3. The prescribed initial and end configurations in specific (left) and in general (right) cases

The simulation results in Fig 4 and 5 are presented for three different cases. In case A, the actuator torques do not saturate, the value of the control torque  $u_1$  reaches even 0.13 Nm as it can be seen in Fig 5. With the chosen set of proportional and differential gains  $\mathbf{K}_\alpha = 60 \text{ I/s}$ ,  $\mathbf{K}_\beta = 48 \text{ I/s}^2$  and sampling time  $h = 40 \text{ ms}$  in equations (10-12), the servo-constraint violation was kept under 0.02 rad as shown in Fig 4. In the graphs,  $\sigma_1$  and  $\sigma_2$  denotes the violation of the first and the second servo-constraint respectively. The fluctuation in the servo-constraint violation at  $t = 8 \text{ s}$  was caused by the high acceleration demand at the inflection point of the arc tangent time histories.

Case B shows the effect of the actuator saturation when the above introduced control algorithm is not implemented, that is, the control inputs are simply truncated. Significant increment occurs in the servo-constraint violation when the control input  $u_1$  reaches the critical 0.04 Nm saturation value as it can be seen in the graphs of panel B of Fig. 4 and 5. Note that the second control input  $u_2$  remains under the saturation level all the time. In case C, the controller switches to the reduced servo-constraint (28) with selector matrix

$$\mathbf{T} = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \quad (39)$$

during the saturation of the first actuator as explained in Section III:

$$\tilde{\sigma}(\mathbf{q}, t) = \begin{bmatrix} -\tan^{-1}\left(\frac{x_2 - x_3}{y_2 - y_3}\right) - \alpha_2^d(t) \end{bmatrix}. \quad (40)$$

When the first actuator saturates, the second control input is recalculated as if the system were underactuated, and this causes the increased value of the input  $u_2$ . As a result, the first servo-constraint violation  $\sigma_1$  is reduced substantially as compared to case B.

The physical meaning of the use of equation (40) during the saturation of the first actuator is that the control effort then focuses on the second servo-constraint which is in connection with the non-saturated and still accessible control

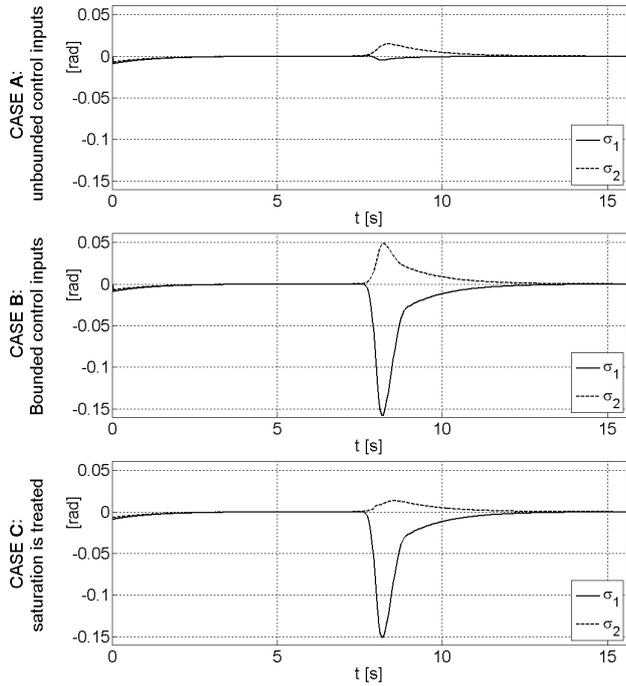


Fig. 4. Numerical results for specific cases A, B and C: servo-constraint violations

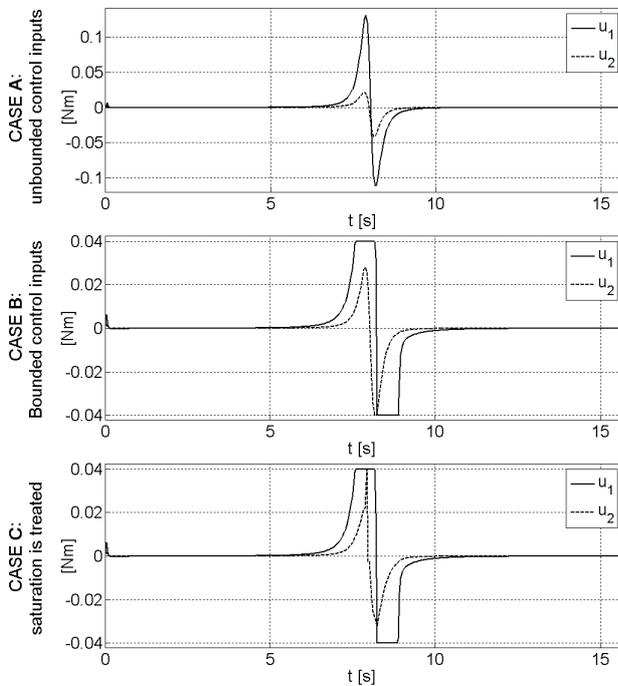


Fig. 5. Numerical results for specific cases A, B and C: control inputs

input  $u_2$ . One can observe that the second actuator also saturates for a very short time in case C. This shows that the available performance of the actuators are utilized more efficiently than they are in case B.

### B. General case

The task of the manipulator is to move the endpoint  $P_3$  of the manipulator from point  $P_{init}$  to  $P_{end}$  on a prescribed straight line trajectory as shown in the right panel of Fig. 3. The task is defined by the servo-constraint vector:

$$\sigma(\mathbf{q}, t) = \begin{bmatrix} x_3 - x^d(t), \\ y_3 - y^d(t) \end{bmatrix}, \quad (41)$$

where  $x^d(t)$  is described by an arc tangent function in time and  $y^d(t) \equiv -0.6$  m is a constant value. Clearly, for the task defined by (41),  $\mathbf{H}(\mathbf{q}) \neq -\sigma_{\mathbf{q}}^T(\mathbf{q}, t)$ , so the connection of the servo constraints and the control inputs is general. The dimension reduction of the servo-constraint vector is carried out by the method explained in Section IV-B.

Three cases are considered again, and the corresponding results are shown in Fig. 6 and 7. In case A, the actuators do not saturate and the value of the control torque  $u_1$  reaches  $-0.08$  Nm. With the same set of control gains and sampling time as above, the initially set servo-constraint errors 4 and 12 mm tend to zero in a short time and after that the servo-constraint violation is kept under 0.2 mm. Case B shows the effect of the actuator saturation when the reduced servo-constraint is not implemented. Significant increment occurs in the servo-constraint violation when the actuator torque  $u_1$  reaches the critical level of 0.04 Nm. In case C, the controller uses the reduced servo-constraint (34) during the actuator saturation. After a large transient, the servo constraint violation  $\sigma_1$  approaches zero much faster than in case B, while  $\sigma_2$  also decreases somewhat.

## VI. Conclusion

The combination of computed torque control algorithms developed for constrained and underactuated systems was successfully implemented for handling actuator saturation. The developed frame algorithm manages the continuous variation of the number of saturated and non-saturated actuators via employing two inferior CTC algorithms: one for fully actuated and one for underactuated systems. Numerical simulations in a case study presented the efficiency of the proposed method. The developed method requires the temporary dimension reduction of the original servo-constraint either in the specific case, when the servo-constraint Jacobian and the control input matrix are equal or in the general case when they are not equal. While the case study shows moderate improvement in following the prescribed task during the saturation of one or more of the actuators, further research can optimize the servo-constraint reduction further by means of the large number of free parameters appearing in the pseudo inverse calculations in (34).

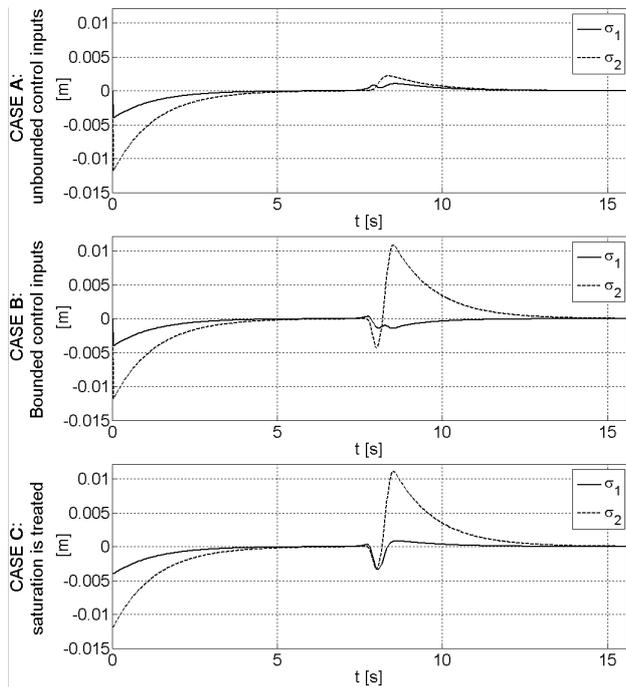


Fig. 6. Numerical results for cases A, B and C: servo-constraint violations

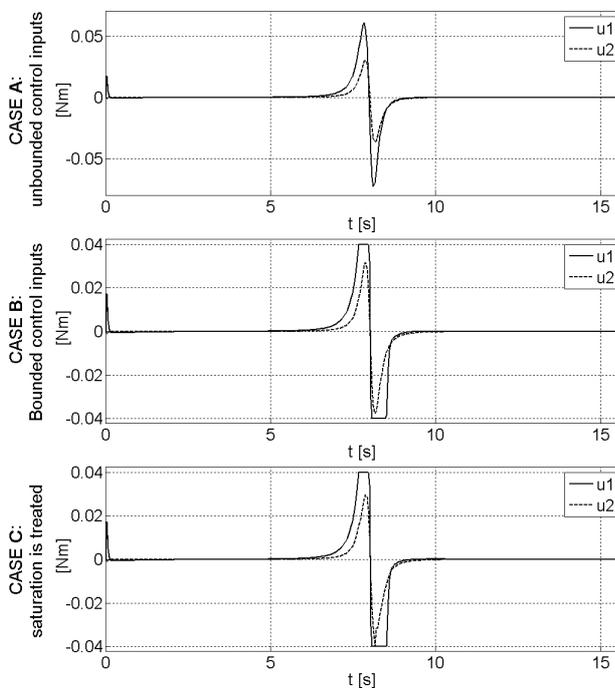


Fig. 7. Numerical results for case A, B and C: control inputs

## References

- [1] P. Lu, "Tracking control of nonlinear systems with bounded controls and control rates," *Automatica*, vol. 33, no. 6, pp. 1199–1202, 1997.
- [2] W. E. Dixon, M. S. de Queiroz, F. Zhang, and D. M. Dawson, "Tracking control of robot manipulators with bounded torque inputs," *Robotica*, vol. 17, no. 2, pp. 121–129, 1999.
- [3] W. Peng, Z. Lin, and J. Su, "Computed torque control-based composite nonlinear feedback controller for robot manipulators with bounded torques," *IET Control Theory and Applications*, vol. 3, no. 6, pp. 701–711, 2009.
- [4] M. W. Spong and M. Vidyasagar, *Robot Dynamics and Control*, John Wiley & Sons, 1989.
- [5] B. Siciliano and O. Khatib (eds.), *Handbook of Robotics*, Springer, 2008, ISBN 978-3-540-23957-4.
- [6] I. M. M. Lammerts, *Adaptive Computed Reference Computed Torque Control*, Ph.D. thesis, Eindhoven University of Technology, 1993.
- [7] W. Blajer and K. Kolodziejczyk, "Modeling of underactuated mechanical systems in partly specified motion," *Journal of Theoretical and Applied Mechanics*, vol. 46, no. 2, pp. 383–394, 2008.
- [8] 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*, vol. 16, no. 5, pp. 2205–2217, 2011.
- [9] A. Isidori, *Nonlinear Control Systems*, 3rd ed. Springer Verlag, 1995.
- [10] H. Goldstein, *Classical Mechanics*, 2nd ed. Addison-Wesley, 1980.
- [11] J.G. de Jalón and E. Bayo, *Kinematic and dynamic simulation of multibody systems: the real-time challenge*, Springer-Verlag, 1994.
- [12] V. I. Kirgetov, "The motion of controlled mechanical systems with prescribed constraints (servo constraints)," *Prikl. Mat. Mekh.*, vol. 31, no. 3, pp. 433–447, 1967.
- [13] W. Blajer and K. Kolodziejczyk, "A geometric approach to solving problems of control constraints: Theory and a dae framework," *Multibody System Dynamics*, vol. 11, no. 4, pp. 343–364, 2004.
- [14] W. Blajer and K. Kolodziejczyk, "Control of underactuated mechanical systems with servo-constraints," *Nonlinear Dynamics*, vol. 50, no. 4, pp. 781–791, 2007, From the issue entitled "Dynamical Systems: Theory and Applications".
- [15] W. Blajer, K. Dziewiecki, K. Kolodziejczyk, and Z. Mazur, "Inverse dynamics of a two-degree freedom underactuated system: Analysis and experiment," in *10th Conference on Dynam Systems - Theory and Applications DSTA-2009, December 7-10 Lodz, Poland*, 2009.
- [16] 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*, 2010, pp. 331–338.
- [17] J. Kövecses, J.-C. Piedboeuf, and C. Lange, "Dynamic modeling and simulation of constrained robotic systems," *IEEE/ASME Transactions on mechatronics*, vol. 8, no. 2, pp. 165–177, 2003.