

Handling actuator saturation as underactuation – case study with Acroboter service robot

Ambrus Zelei

MTA-BME Research Group on
Dynamics of Machines and Vehicles
Budapest, Hungary
Email: zelei@mm.bme.hu

László Bencsik

MTA-BME Research Group on
Dynamics of Machines and Vehicles
Budapest, Hungary
Email: bencsik@mm.bme.hu

Gábor Stépán

Department of Applied Mechanics
Budapest University of Technology and Economics
Budapest, Hungary
Email: stepan@mm.bme.hu

ABSTRACT

Model based control methods, like inverse dynamics control and computed torque control encounter difficulties if actuator saturation occurs. However, saturation is a common phenomenon in robotics leading to significant non-linearity in system behavior. In this study, the saturation of the actuator torques is considered as a temporary reduction of the number of independent control inputs. The reduction of the number of actuators leads to an underactuated control problem which typically involves the handling of differential algebraic equation systems. The saturated system may become especially complex when intricate combinations of the actuator saturations appear. A servo-constraint based inverse dynamics control method for underactuated multibody systems is applied for the treatment of actuator torque saturation. In case of human-friendly robots, the problem of saturation cannot be avoided on the level of trajectory planning because unexpected human perturbations may take place, which result such abrupt changes of the desired trajectory that lead to saturation at some actuators. A case study for the service robot Acroboter shows the applicability of the proposed approach.

1 Introduction

Any actuator used in robotic systems has some limitations which are typically speed, power and/or torque. These limitations can be included in the mathematical models and so they can be taken into account already during the design of the task of the robot. In case of human-friendly robots, for example, the problem of saturation cannot be avoided on the level of trajectory planning, because unexpected events like human interaction may lead to the sudden change of the prescribed trajectory. Thus, the actuator torque saturation may cause significant problems when a manipulator performs the tracking of a desired trajectory. Most commonly, the inverse dynamics based control or the similar computed torque control (CTC) techniques are used especially when the goal is accurate trajectory tracking.

Several inverse dynamics based control algorithms can be found in the literature, which take into account the limited actuator torques. In [1], a “continuous-time predictive control approach” is used to derive the non-linear 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 (see, for example, [3, 4]), a “composite non-linear feedback” design method is presented in [5] 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 non-linear dynamics, while the outer loop is the composite non-linear feedback controller for stabilization and performance enhancement.

The above mentioned control approaches handle the actuator saturation as a non-linearity of the system. Alternatively, actuator saturation can also be modeled as the decrement of the number of accessible control inputs, which is practically equivalent to the variation of the manipulator’s topology. In other words, actuator saturation causes intermittent variation in the number of the actually controllable actuators. The system loses some of the accessible actuators because they reached their saturation levels. In the resulting situation, the number of actuators will be less than the number of the degrees of freedom. Such systems are called underactuated [6]. As a new concept, actuator saturation is handled by underactuated control algorithm adopted from the literature.

In the subsequent sections, first, a classical inverse dynamics method is presented briefly when the system is underactuated, non-minimum set of coordinates (in other terminology, redundant set of generalized coordinates) is used and the task is defined by servo-constraints (see [7–11]). In order to handle actuator saturation in fully- or underactuated manipulators, the underactuated CTC method is generalized for varying number of actuators, similarly as introduced in [12]. After discussing the possible ways of the dimension reduction of servo-constraints, the efficiency of the proposed inverse dynamics control algorithm is demonstrated in a case study for the service robot Acroboter [13].

2 The applied inverse dynamics control algorithm

In case of serial and fully actuated robotic manipulators, an independent control input is associated with each degree of freedom (DoF). Consequently, the classical inverse dynamics 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 a form of an ordinary differential equation (ODE) [3, 4]. This is not the case for underactuated robotic manipulators.

Partial feedback linearization can be used for the control of underactuated systems [14]. The CTC method for underactuated systems can be generalized for systems modeled by non-minimum set of coordinates when the mathematical model is extended with geometric constraints and arranged into a system of differential algebraic equations (DAE) [8–11, 15, 16].

The subsections below detail a servo-constraint based CTC method for underactuated systems described by non-minimum set of coordinates. The original method from literature is applied for systems with unbounded control inputs, while the application for bounded systems is explained in Section 3, which is a novel concept.

2.1 Inverse dynamics problem for underactuated multibody systems

When a g DoF multibody system is described by non-minimum set of coordinates $\mathbf{q} \in \mathbb{R}^n$, $m = n - g$ number of geometric constraints $\boldsymbol{\varphi}(\mathbf{q}, t) \in \mathbb{R}^m$ are required to provide the specific connections of the redundant coordinates. The Lagrangian equation has the form:

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{c}(\mathbf{q}, \dot{\mathbf{q}}) + \boldsymbol{\varphi}_q^T(\mathbf{q}, t)\boldsymbol{\lambda} = \mathbf{H}(\mathbf{q})\boldsymbol{\tau}, \quad (1)$$

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

which is a differentiation index 3 DAE [17]. $\mathbf{M}(\mathbf{q}) \in \mathbb{R}^{n \times n}$ is the generalized mass matrix and vector $\mathbf{c}(\mathbf{q}, \dot{\mathbf{q}}) \in \mathbb{R}^n$ describes the centrifugal, 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 $\partial \boldsymbol{\Phi}(\mathbf{q}, t) / \partial \mathbf{q} = \boldsymbol{\Phi}_q(\mathbf{q}, t) \in \mathbb{R}^{m \times n}$. $\boldsymbol{\lambda} \in \mathbb{R}^m$ contains the corresponding Lagrange multipliers. The control input vector is $\boldsymbol{\tau} \in \mathbb{R}^l$ while $\mathbf{H}(\mathbf{q}) \in \mathbb{R}^{n \times l}$ is the generalized control input matrix. If the number l of the control inputs is less than the DoF of the system ($l < g = n - m$), then it is called *underactuated*, while if $l = g$ than the system is *fully actuated*.

In order to obtain unique solution for the inverse dynamics problem, the number of desired outputs has to be equal to the number of inputs. The manipulator's task is defined in the form of additional constraints $\boldsymbol{\sigma}(\mathbf{q}, t) \in \mathbb{R}^l$ called *servo-constraints* [8, 18] in the form of a servo-constraint equation:

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

The goal of the inverse dynamic calculation is to determine the required control input $\boldsymbol{\tau}$ from DAE (1), (2) and (3). A numerical approach is presented to carry out the calculations in each sampling period.

2.2 A servo-constraint based control method

Here we present a possible approach for the control problem detailed in Section 2.1. Although, other control methods from the literature of underactuated systems would be possible to apply similarly for handling saturation.

The method of Lagrange multipliers provides DAE index reduction by applying the constraints on the level of acceleration. The unconstrained dynamic equation (1) is augmented by the time derivatives of the constraints (2) and (3) and Baumgarte stabilization is applied [17] in the form:

$$\begin{bmatrix} \mathbf{M} & \boldsymbol{\Phi}_q^T & -\mathbf{H} \\ \boldsymbol{\Phi}_q & \mathbf{0} & \mathbf{0} \\ \boldsymbol{\sigma}_q & \mathbf{0} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \ddot{\mathbf{q}}^d \\ \boldsymbol{\lambda} \\ \boldsymbol{\tau} \end{bmatrix} = \begin{bmatrix} -\mathbf{c} \\ -\dot{\boldsymbol{\Phi}}_q \dot{\mathbf{q}} - \dot{\boldsymbol{\Phi}}_t \\ -\dot{\boldsymbol{\sigma}}_q \dot{\mathbf{q}} - \dot{\boldsymbol{\sigma}}_t - \alpha \dot{\boldsymbol{\sigma}} - \beta \boldsymbol{\sigma} \end{bmatrix} \quad (4)$$

where α and β are the Baumgarte-type control gains for the servo-constraint equations. The geometric constraints are not necessary to be stabilized, because they are naturally satisfied by measured values of \mathbf{q} . Eq (4) is not applicable for motion simulation in the present form. It provides the control input $\boldsymbol{\tau}$, the desired accelerations $\ddot{\mathbf{q}}^d$ and, the Lagrange multipliers $\boldsymbol{\lambda}$.

The rows of the servo-constraint Jacobian $\boldsymbol{\sigma}_q$ and the geometric constraint Jacobian $\boldsymbol{\Phi}_q$ have to be independent in the operation range of the robot.

3 Extended algorithm for handling saturation

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

If some actuators saturate, then the number of the non-saturated actuators is reduced to \hat{l} . The dimension \hat{l} of the still accessible non-saturated control input vector $\hat{\boldsymbol{\tau}}$ is always less than the number of DoF, so the system is actually underactuated. The $l - \hat{l}$ number of saturated actuators provide a constant torque τ_i^+ or τ_i^- , and the algorithm recalculates the desired control inputs $\hat{\boldsymbol{\tau}} \in \mathbb{R}^{\hat{l}}$ with $\hat{l} < l$ as an underactuated system. This process 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.

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

$$\mathbf{M}\ddot{\mathbf{q}} + \mathbf{c}(\mathbf{q}, \dot{\mathbf{q}}) + \boldsymbol{\Phi}_q^T(\mathbf{q})\boldsymbol{\lambda} = \mathbf{H}(\mathbf{q})\mathbf{R}\boldsymbol{\tau}^\pm + \mathbf{H}(\mathbf{q})\mathbf{T}\hat{\boldsymbol{\tau}}, \quad (5)$$

where $\boldsymbol{\tau}^\pm \in \mathbb{R}^l$ contains the corresponding maximal and minimal limiting values τ_i^+ or τ_i^- , while $\hat{\boldsymbol{\tau}}$ 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

$$\boldsymbol{\tau} = \mathbf{R}\boldsymbol{\tau}^\pm + \mathbf{T}\hat{\boldsymbol{\tau}}. \quad (6)$$

In (5) and (6), the selector matrix \mathbf{R} identifies the saturated control inputs, and \mathbf{T} identifies the non-saturated (still accessible) control input vector as:

$$\hat{\boldsymbol{\tau}} = \mathbf{T}^\top \boldsymbol{\tau}, \quad (7)$$

where \mathbf{T}^\top is identical with the Moore-Penrose generalized inverse (pseudo-inverse) \mathbf{T}^\dagger since \mathbf{T} is built on an orthogonal basis.

In (5), the term $\mathbf{H}(\mathbf{q})\mathbf{R}\boldsymbol{\tau}^\pm$ is a known, constant external generalized force, while the term $\mathbf{H}(\mathbf{q})\mathbf{T}\hat{\boldsymbol{\tau}}$ 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

$$\hat{\mathbf{H}}(\mathbf{q}) = \mathbf{H}(\mathbf{q})\mathbf{T} \quad (8)$$

and the new vector of inertial forces in the form

$$\hat{\mathbf{c}}(\mathbf{q}, \dot{\mathbf{q}}) = \mathbf{c}(\mathbf{q}, \dot{\mathbf{q}}) - \mathbf{H}(\mathbf{q})\mathbf{R}\boldsymbol{\tau}^\pm. \quad (9)$$

This way, equation (5) assumes the form

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \hat{\mathbf{c}}(\mathbf{q}, \dot{\mathbf{q}}) + \boldsymbol{\Phi}_q^T(\mathbf{q}, t)\boldsymbol{\lambda} = \hat{\mathbf{H}}(\mathbf{q})\hat{\boldsymbol{\tau}}, \quad (10)$$

which is fully compatible with the form (1) appears 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 is equal to the number \hat{l} of accessible control inputs $\hat{\boldsymbol{\tau}}$. 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. Possible techniques are proposed in the following section.

4 Dimension reduction of servo-constraints

The dimension reduction of the desired task in case of actuator saturations is firstly introduced for "specific case", which provides the fundamentals of the dimension reduction in "general case".

4.1 Specific case

In the equation of motion (1), the term $\mathbf{H}(\mathbf{q})\boldsymbol{\tau}$ 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 $\boldsymbol{\mu} \in \mathbb{R}^l$ of Lagrange multipliers, which are associated with the servo-constraint $\boldsymbol{\sigma}(\mathbf{q}, t)$ and its Jacobian $\boldsymbol{\sigma}_q(\mathbf{q}, t)$ (see [8,9] and [12]). Using this concept, (1) is rewritten as:

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{c}(\mathbf{q}, \dot{\mathbf{q}}) + \boldsymbol{\Phi}_q^T(\mathbf{q}, t)\boldsymbol{\lambda} + \boldsymbol{\sigma}_q^T(\mathbf{q}, t)\boldsymbol{\mu} = \mathbf{0}, \quad (11)$$

considering that

$$\boldsymbol{\sigma}_q^T(\mathbf{q}, t)\boldsymbol{\mu} = -\mathbf{H}(\mathbf{q})\boldsymbol{\tau}. \quad (12)$$

The connection between the control input $\boldsymbol{\tau}$ and the multiplier $\boldsymbol{\mu}$ is trivial if the servo-constraint Jacobian and the negative control input matrix are just equal:

$$\boldsymbol{\sigma}_q^T(\mathbf{q}, t) = -\mathbf{H}(\mathbf{q}), \quad (13)$$

$$\boldsymbol{\tau} = \boldsymbol{\mu}. \quad (14)$$

This is called "specific case" when the directions defined by the control input matrix and the servo-constraint Jacobian are identical. The structure of the matrix $\mathbf{H}(\mathbf{q})$ definitely depends on the mechanical design of the system and the chosen set of coordinates \mathbf{q} , while the Jacobian $\boldsymbol{\sigma}_q(\mathbf{q}, t)$ 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 between the control inputs and the servo-constraints is trivial. In [8], this situation is called ideal orthogonal realization, which refers to the situation when the control reactions $\mathbf{H}(\mathbf{q})\boldsymbol{\tau}$ are explicitly represented in all directions constrained by $\boldsymbol{\sigma}(\mathbf{q}, t)$. Clearly, the above conditions can be satisfied in specific cases only.

In the saturated system the control input matrix of the still non-saturated inputs is defined by equation (8). The connection between the reduced servo-constraint Jacobian $\hat{\boldsymbol{\sigma}}_q(\mathbf{q}, t)$ and it is reasonable to define the reduced size control input matrix $\hat{\mathbf{H}}(\mathbf{q})$ as equation (13) does:

$$\hat{\boldsymbol{\sigma}}_q^T(\mathbf{q}, t) = -\hat{\mathbf{H}}(\mathbf{q}). \quad (15)$$

Considering equations (8) and (15) we can write:

$$\hat{\boldsymbol{\sigma}}_q^T(\mathbf{q}, t) = \boldsymbol{\sigma}_q^T(\mathbf{q}, t)\mathbf{T}. \quad (16)$$

From (16), the proposed reduced servo-constraint is simply:

$$\hat{\boldsymbol{\sigma}}(\mathbf{q}, t) = \mathbf{T}^\top \boldsymbol{\sigma}(\mathbf{q}, t). \quad (17)$$

Note that in these special cases, the servo-constraint reduction transformation is the same as the transformation between the full control input vector $\boldsymbol{\tau}$ and the vector $\hat{\boldsymbol{\tau}}$ of the non-saturated control inputs in equation (7).

4.2 General case

The servo-constraint Jacobian and the negative control input matrix are not identical in general:

$$\boldsymbol{\sigma}_q^\top(\mathbf{q}, t) \neq -\mathbf{H}(\mathbf{q}), \quad (18)$$

which means that the directions defined by the control input matrix and the servo-constraint Jacobian are not necessarily the same (called as non-ideal orthogonal realization in [8]). Since the matrices are non-square, the transformation between $\boldsymbol{\tau}$ and $\boldsymbol{\mu}$ can be deduced from (12) using the Moore-Penrose generalized inverse:

$$\boldsymbol{\mu} = -(\boldsymbol{\sigma}_q^\top(\mathbf{q}, t))^\dagger \mathbf{H}(\mathbf{q}) \boldsymbol{\tau}. \quad (19)$$

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

$$(\boldsymbol{\sigma}_q^\top)^\dagger = (\boldsymbol{\sigma}_q \boldsymbol{\sigma}_q^\top)^{-1} \boldsymbol{\sigma}_q, \quad (20)$$

is widely used for convenience. Based on (19) we introduce the linear mapping between $\boldsymbol{\mu}$ and $\boldsymbol{\tau}$ as

$$\boldsymbol{\Gamma}(\mathbf{q}, t) = -(\boldsymbol{\sigma}_q^\top(\mathbf{q}, t))^\dagger \mathbf{H}(\mathbf{q}). \quad (21)$$

Similarly as in (17) we define a linear mapping between $\hat{\boldsymbol{\sigma}}(\mathbf{q}, t)$ and $\boldsymbol{\sigma}(\mathbf{q}, t)$ as:

$$\hat{\boldsymbol{\sigma}}(\mathbf{q}, t) = \hat{\boldsymbol{\Gamma}}^\dagger(\mathbf{q}, t) \boldsymbol{\sigma}(\mathbf{q}, t). \quad (22)$$

As it was derived in the specific case, when the identity mapping appears in (14) and the selector matrix \mathbf{T} gives the linear mapping, in general case $\boldsymbol{\Gamma}(\mathbf{q}, t)$ and \mathbf{T} defines the transformation matrix in the following way:

$$\hat{\boldsymbol{\Gamma}}(\mathbf{q}, t) = -(\boldsymbol{\sigma}_q^\top(\mathbf{q}, t))^\dagger \mathbf{H}(\mathbf{q}) \mathbf{T}, \quad (23)$$

which obviously simplifies to (17) in the special case defined by (13). In the following case study, this formula will be used for the dimension reduction of the servo-constraint.

5 Case study for the Acroboter service robot platform

In order to verify the efficiency of the proposed control algorithm, numerical simulations were carried out in case of a crane-like service robot called Acroboter [13]. The simplified planar model of the investigated robot is shown in Fig. 2. The robot is composed of two main subsystems namely the climber unit (CU) which provides the main horizontal motion and the swinging unit (SU) which carries out the required manipulations. The vertical motion of SU is provided by the main cable (MC) and the orientation is maintained by the secondary orienting cables (SC1 and SC2), which are connected together via the cable connector (CC). The horizontal stabilization and the fine positioning are performed by a ducted-fan actuator.

The geometric description is provided by using the so-called natural coordinates [17]. The CU and CC are modeled as particles, while the SU is represented by a rigid body. Thus $\mathbf{q} = [x_{CU}, x_{CC}, y_{CC}, x_{P1}, y_{P1}, x_{P2}, y_{P2}]^T$ are chosen as descriptor coordinates (see: Fig. 2 left). The active forces appear in the free-body diagram on the right hand side of Fig. 2. The system has $l = 5$ independent input forces $\boldsymbol{\tau} = [F_L, F_M, F_1, F_2, F_T]^T$, therefore the task also has to be 5 dimensional. The coordinates are dependent hence the following constraint condition is formulated for the SU:

$$\boldsymbol{\varphi}(\mathbf{q}) = [(x_{P1} - x_{P2})^2 + (y_{P1} - y_{P2})^2 - L_{12}^2]. \quad (24)$$

The corresponding members of the equation of motion (1) can be derived based on [17], and the parameters are the following: $m_{CU} = 5 \text{ kg}$, $m_{CC} = 1 \text{ kg}$, $m_{SU} = 4 \text{ kg}$, $I_{SU} = 0.8 \text{ kgm}^2$, $L_{12} = 0.7 \text{ m}$. The actuators saturate, when the actuator forces reach the limiting values, which are symmetric in case of the CU linear drive and the thrust force $\tau_1^\pm = \pm 10 \text{ N}$ and $\tau_5^\pm = \pm 5 \text{ N}$, while the cable forces have asymmetric limiting values: $\tau_2^+ = 55 \text{ N}$, $\tau_3^+ = \tau_4^+ = 35 \text{ N}$ and $\tau_2^- = \tau_3^- = \tau_4^- = 0 \text{ N}$ because only tensioning forces are physically possible.

According to the task definition, the SU has to follow a rectangular path defined by the time dependent terms $x_{SU}^d(t)$ and $y_{SU}^d(t)$ appears in the servo-constraint

$$\boldsymbol{\sigma}(\mathbf{q}, t) = \begin{bmatrix} (x_{P1} + x_{P2})/2 - x_{SU}^d(t) \\ (y_{P1} + y_{P2})/2 - y_{SU}^d(t) \\ y_{P1} - y_{P2} \\ x_{CU} - x_{CU}^d(t), \\ y_{CC} - (y_{P1} + y_{P2})/2 - h_{CC}^d(t) \end{bmatrix}. \quad (25)$$

The corresponding trajectory is shown in Fig. 5 by the dashed line. Additionally, the SU has to be kept horizontally. The vertical position of the CU and the height of the CC is defined by $x_{CU}^d(t)$ and $h_{CC}^d(t) = 0.7 \text{ m}$ respectively.

For the task defined by (25), $\mathbf{H}(\mathbf{q}) \neq -\boldsymbol{\sigma}_q^T(\mathbf{q}, t)$, so the dimension reduction of the servo-constraint vector is carried out by the ‘‘general’’ method explained in Section 4.2.

Three cases are considered, and the corresponding results are shown in Figures 3, 4 and 5. The derivative and proportional control gains were $\alpha = 20 \text{ N/s/m}$ and $\beta = 200 \text{ N/m}$ respectively and sampling time was $\Delta t = 0.01 \text{ s}$ in all cases.

I. First, the actuators do not saturate, because the actuator limits are not considered. The corresponding graphs are plotted by dashed curves. The initially set servo-constraint errors tend to zero in a short time and after that the servo-constraint violation is kept under 0.2 m. High errors occur when high horizontal accelerations are required in the corners of the rectangular path. The tracking error could be set smaller with different tuning of the control gains, but we scaled them up intentionally for the sake of better visualization.

II. In the second case, illustrated by solid gray curves, the effect of the actuator saturation is shown, when the reduced servo-constraint is not implemented, and the controller does not have any information about the loss of available actuators. Significant increment of the servo-constraint violation occurs about $t = 7 \text{ s}$ and $t = 17 \text{ s}$ when a high acceleration is required

and many of the actuator forces reach the limiting value (see Fig 4).

III. The third case shows significantly better results, when the controller uses the reduced servo-constraint (22) during the actuators' saturation. The peak values of the servo-constraint violation are much lower, as the solid black curve shows in Fig. 3.

The RMS error was 42mm in case of unbounded actuators, which increased up to 109mm when the saturation of actuators was taken into account. The application of the proposed algorithm reduced the RMS error by 21% to 86mm. The time histories of the actuator forces are depicted in Fig. 4, while Fig. 5 shows the path of the SU and the CC in all three simulation cases.

6 Conclusion

As a novel concept, the actuator torque saturation was treated as a temporary reduction in the number of independent accessible actuators, which leads to temporary underactuation. The generalization of a specific model based control algorithm developed for underactuated multibody systems was successfully implemented for this problem. However, other underactuated control algorithms would be possible to generalize similarly.

A higher level frame algorithm is responsible for the monitoring of actuator saturation and involves an underactuated CTC algorithm. Considering the actual number of available control inputs and the dynamics of the system, the dimension of the original task is reduced via the reformulation of the servo-constraints.

Numerical simulations in a case study for a crane-like service robot called Acroboter presented the efficiency of the proposed method even in the general case when the servo-constraint Jacobian and the control input matrix are not equal. The case study shows significant improvement in the realization of the prescribed task during the saturation of one or more of the actuators. However, the servo-constraint reduction uses non-unique calculation steps, which can be further optimized as future work. Besides, the important question of the effect and tuning of PD gains will be studied in future work.

Acknowledgements

This work has been supported by the MTA-BME Research Group on Dynamics of Machines and Vehicles and by the Hungarian Scientific Research Fund, Hungary (OTKA) under grant numbers K-105433 and K-101714.

References

- [1] Lu, P., 1997. "Tracking control of nonlinear systems with bounded controls and control rates". *Automatica*, **33**(6), pp. 1199–1202.
- [2] Dixon, W. E., de Queiroz, M. S., Zhang, F., and Dawson, D. M., 1999. "Tracking control of robot manipulators with bounded torque inputs". *Robotica*, **17**(2), pp. 121–129.
- [3] Spong, M. W., and Vidyasagar, M., 1989. *Robot Dynamics and Control*. John Wiley & Sons.
- [4] Siciliano, B., and (eds.), O. K., 2008. *Handbook of Robotics*. Springer. ISBN 978-3-540-23957-4.
- [5] Peng, W., Lin, Z., and Su, J., 2009. "Computed torque control-based composite nonlinear feedback controller for robot manipulators with bounded torques". *IET Control Theory and Applications*, **3**(6), pp. 701–711.
- [6] Spong, M., Munich, Germany, 1994. "Partial feedback linearization of underactuated mechanical systems". In IROS'94, pp. 314–321.
- [7] Lammerts, I. M. M., 1993. "Adaptive computed reference computed torque control". PhD thesis, Eindhoven University of Technology.
- [8] Blajer, W., and Kolodziejczyk, K., 2004. "A geometric approach to solving problems of control constraints: Theory and a dae framework". *Multibody System Dynamics*, **11**(4), pp. 343–364.

- [9] Blajer, W., and Kolodziejczyk, K., 2007, From the issue entitled "Dynamical Systems: Theory and Applications". "Control of underactuated mechanical systems with servo-constraints". *Nonlinear Dynamics*, **50**(4), pp. 781–791.
- [10] Zelei, A., Kovács, L. L., and Stépán, G., 2011. "Computed torque control of an under-actuated service robot platform modeled by natural coordinates". *Commun Nonlinear Sci Numer Simulat*, **16**(5), pp. 2205–2217.
- [11] Masarati, P., Morandini, M., and Fumagalli, A., 2014. "Control constraint of underactuated aerospace systems". *ASME Journal of Computational and Nonlinear Dynamics*, **9**(2), pp. 433–447. doi:10.1115/1.4025629.
- [12] Zelei, A., and Stépán, G., 2015. "Underactuation induced by actuator saturation". In In Proc. of the 14th IFToMM World Congress, Taipei, Taiwan, October 25-30. paper ID: OS13-089, DOI Number: 10.6567/IFToMM.14TH.WC.OS13.089.
- [13] Stépán, G., and et al., 2009. "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.
- [14] Isidori, A., 1995. *Nonlinear Control Systems*. 3rd ed. Springer Verlag.
- [15] Zelei, A., Kovács, L. L., and Stépán, G., August 28-August 31, Washington, USA, 2011. "Servo-constraint based computed torque control of underactuated mechanical systems". In Proceedings of the ASME 2011 8th International Conference on Multibody Systems, Nonlinear Dynamics, and Control, pp. Paper id: DETC2011–48533.
- [16] Kovács, L. L., and Bencsik, L., 2012. "Stability case study of the acroboter underactuated service robot". *Theoretical and Applied Mechanics Letters*, **2**(043004), p. 7 pages. DOI: 10.1063/2.1204304.
- [17] de Jalón, J., and Bayo, E., 1994. *Kinematic and dynamic simulation of multibody systems: the real-time challenge*. Springer-Verlag.
- [18] Kirgetov, V. I., 1967. "The motion of controlled mechanical systems with prescribed constraints (servo constraints)". *Prikl. Mat. Mekh.*, **31**(3), pp. 433–447.

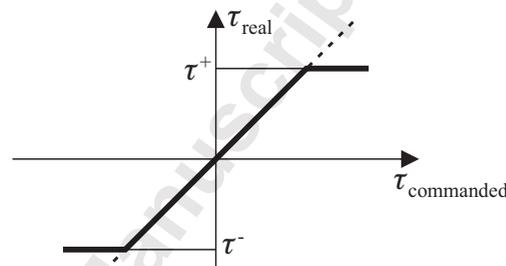


Fig. 1. Actuator saturation: non-linear connection between commanded and real control input

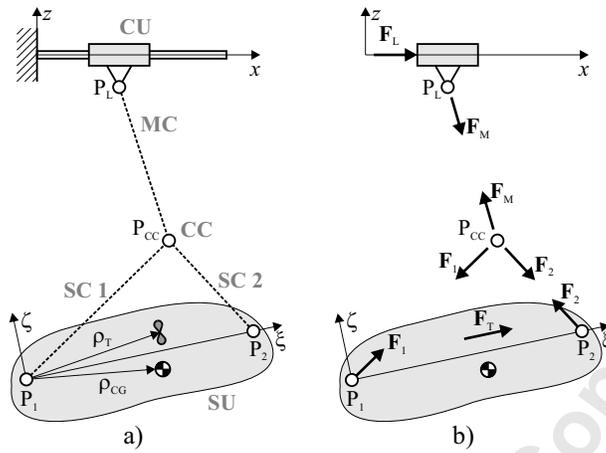


Fig. 2. Planar mechanical model (left) and free-body-diagrams (right) of the Acroboter platform

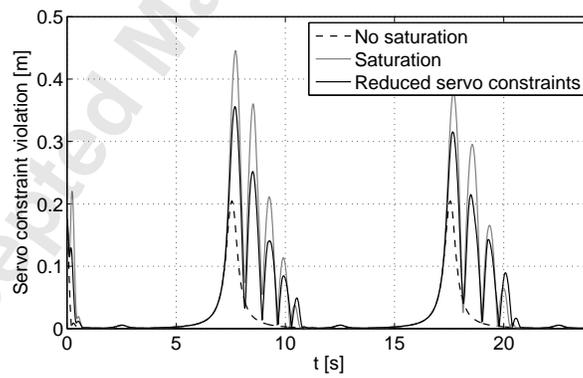


Fig. 3. Numerical results for specific cases **I.**, **II.** and **III.**: servo-constraint violations

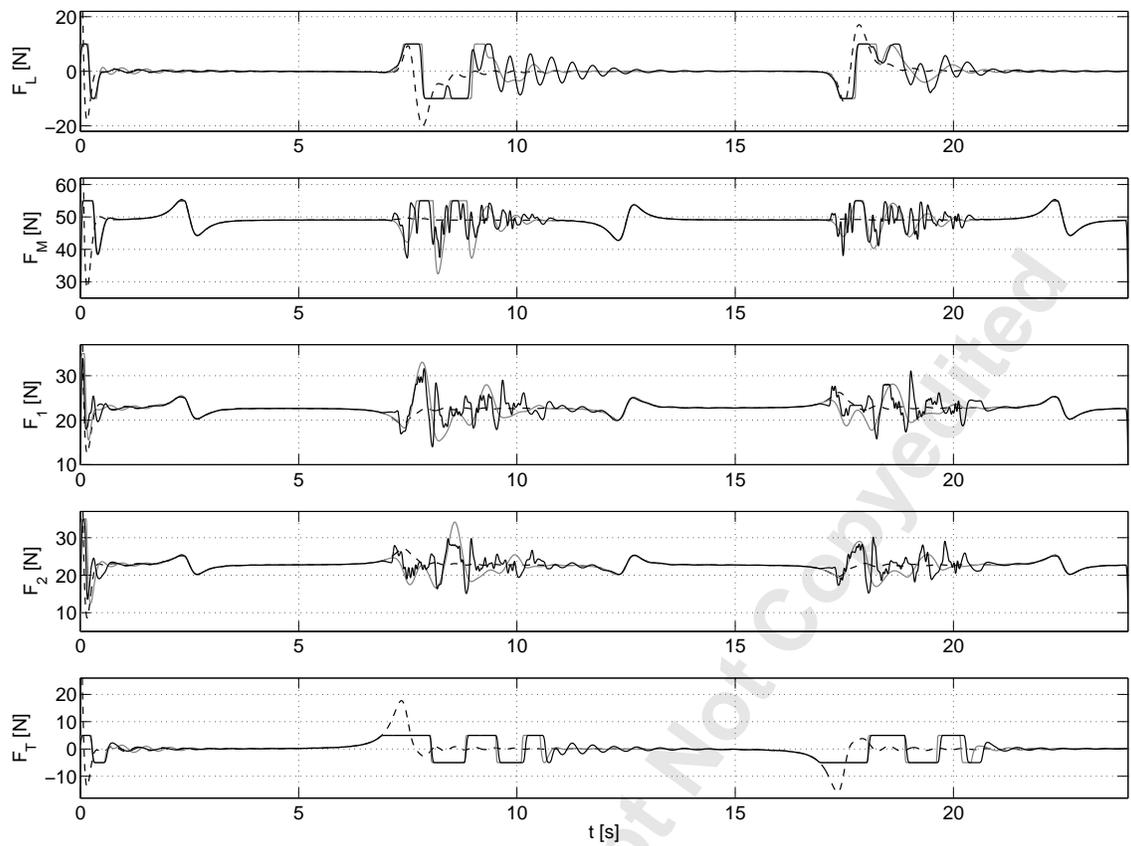


Fig. 4. Numerical results for specific cases **I**, **II**, and **III**.: control inputs

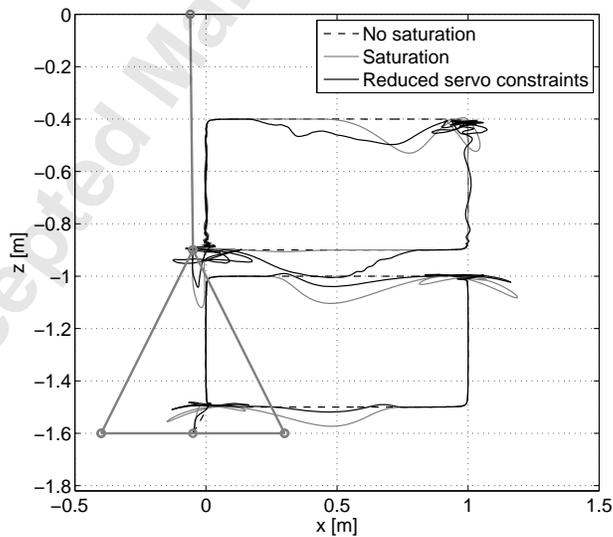


Fig. 5. Numerical results for specific cases **I**, **II**, and **III**.: path