Available online at www.sciencedirect.com

ScienceDirect

IFAC PapersOnline CONFERENCE PAPER ARCHIVE

IFAC PapersOnLine 51-20 (2018) 265–272

Offset-Free MPC for Resource Sharing on a Nonlinear SCARA Robot ⋆

M. Bianchi ** **A. van der Maas** * **E. Maljaars** * **W.P.M.H. Heemels** *

* *Department of Mechanical Engineering, Control System Technology Group, Eindhoven University of Technology, 5600MB, Eindhoven, The Netherlands*
(e-mail: A.v.d.Maas@tue.nl, E.Maljaars@tue.nl, M.Heemels@tue.nl).
** *Department of Information Engineering, Computer Science and Mathematics, Center of Excellence DEWS, University of L'Aquila, 67100, L'Aquila, Italy*
(e-mail: mattia.bianchi@student.univaq.it)

Abstract: High-precision motion industrial systems must satisfy tight performance requirements. Both positioning accuracy and throughput demands are typically achieved through improvements in hardware, thereby raising the bill of materials. A cost saving alternative could be to strive for a reduction in the hardware components needed, in combination with advanced motion control, to still meet the desired specifications. Particularly, in this paper, the possibility is analyzed to allow for resource sharing among several actuators. This results in a switched system, for which we develop a real-time MPC algorithm for optimization of both the input and the switching signals. This implementation applies to a fairly general class of nonlinear systems and uses a novel offset-free formulation in velocity form for LTV prediction models, to realize good tracking performance under the resource sharing constraints. We provide a proof of concept for this MPC solution on a high fidelity model of an industrial SCARA robot, where it is proposed to use a single amplifier to serve two actuators. The MPC solution is compared to heuristically switched LTI controllers, and the potential of the proposed approach is shown in simulations.

Keywords: Control applications, switching systems, hybrid systems, robotics, motion control, offset-free MPC, mixed-integer quadratic programming

1. INTRODUCTION

Robotic handling systems are important in many industrial applications, with increasingly higher demands on the accuracy and throughput. Generally, more expensive hardware components are needed to comply with the increased performance specifications. However, in several situations it might be of interest to use cheaper hardware solutions to actually decrease the overall bill of materials, while advanced motion control algorithms are exploited to meet the desired accuracy and throughput requirements. One example, being the benchmark case in this paper, is the control of a robotic system that has multiple actuators (motors), each of them individually driven by its dedicated amplifier. To reduce the bill of materials it could be of interest to use only one (or a few) amplifiers that are shared by several actuators, thereby reducing the number of amplifiers. It is clear that, to achieve this reduction, non-standard motion control algorithms are required that determine the continuous input (voltage to the amplifier) and the switching signal (which actuator is allowed to use the shared resource) to still meet the performance requirements.

Although the control architecture in this paper is applicable in many control settings, where resources are shared, we will illustrate the main ideas on a particular case study of a SCARA [1] robot, to investigate the feasibility and potential of amplifier sharing. A wide range of applications for SCARA robots is available, such as solar cells production, LED manufacturing, wafer handling in lithographic machines, and many other systems in the semiconductor industry. The SCARA robot, as in Fig. 1, consists of an upper and a lower arm, and has three degrees of freedom associated with the rotation of the arms and translation along the z-axis, . Currently, three amplifiers are used to control the three actuators related to the degrees of freedom. The possibility to remove a single amplifier could already result in a significant cost reduction. As such, in this case, the focus will be on a single amplifier that will be shared between the upper and lower arm of the robot, with the consequence that only one of the rotational joints actuator is served at any given time instant. As a result, next to the power level (continuous input), also the choice for which actuator is being powered (discrete decision) has to be made.

The idea of amplifier sharing was already mentioned in (Driessen et al., 2011) for single-stage planar actuators, with the focus on the required electronics behind amplifier sharing. However, the advanced control required to allow for this sharing was never considered. Therefore, we investigate this controller design problem in the present paper.

In our preliminary feasibility study (van der Maas et al., 2017), a switched linear controller design has been explored, using a periodical switching strategy, where the power is switched at every discrete time sample between the two actuators. The results underline the potential of reducing the bill of materials

⋆ This work is part of the research programme CHAMeleon "Hybrid solutions for cost-aware high-performance motion control" with project number 13896, which is (partly) financed by the Netherlands Organisation for Scientific Research (NWO).

[1] SCARA stands for Selective Compliance Articulated Robot Arm

10.1016/j.ifacol.2018.11.024

Fig. 1. Experimental SCARA robot. The object is normally positioned at the end of the wrist, with its center between the two black rubber rings.

by sharing hardware components between multiple actuators and it serves as a proof of concept, encouraging further research in this interesting area.

A drawback of the periodically switched LTI controllers in (van der Maas et al., 2017) is that they were designed heuristically and do not optimally select the actuator being powered (discrete input). In contrast, model predictive control (MPC) can optimally deal with the "switched" actuator (and other) constraints in the actuator input computation itself and is therefore of interest for this application. In the literature, MPC for switched linear systems has been researched extensively and many results for state-dependent switches in the plant (Borrelli et al., 2005; Bemporad and Morari, 1999; Lazar et al., 2006; Zhang et al., 2012, 2016). Some extensions of these control methods to nonlinear systems have also been developed (Mhaskar et al., 2006; Müller et al., 2012). The main difference with respect to our application is that in this work the switching between multiple actuators is part of the control input calculation and not of the plant (as is the case in the above references). The application of MPC to systems with binary actuator inputs (actuators that can only take discrete values) resulting in Mixed-Integer Programming (MIP) optimization problems is also investigated in the literature (see, e.g., (Giorgetti et al., 2006; Azuma and Sugie, 2008)), but these works assume that all actuators can be active at all times.

In this paper we essentially deal with the control of a non-linear system with switched actuator inputs and we aim for good tracking performance. We propose an MPC formulation for this setting combining

(i) linearisation of the non-linear robot dynamics along the (known) reference trajectory, leading to linear time-varying (LTV) state-space models that are used in a time-varying prediction model,
(ii) offset-free MPC setup to cope with model mismatch and persistent disturbances, based on the velocity form as given in (Pannocchia, 2015), but extended to LTV models to be applicable in our setting,
(iii) integration of the actuator switching constraints as hard constraints in a Mixed-Integer Quadratic Programming (MIQP) optimization problem.

The remainder of this paper is organized as follows. In Sec. 2, the benchmark case is introduced, together with a problem statement. In Sec. 3, the LTV-MPC controller in velocity form is proposed. In Sec. 4, the novel MPC scheme is evaluated in comparison to heuristically switched linear controllers, based on a simulation study on a high-fidelity model of the industrial SCARA robot. In Sec. 5, conclusions are given.

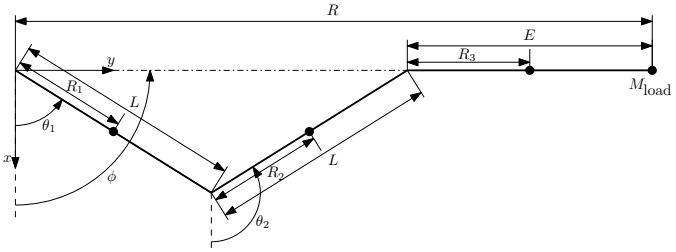


Fig. 2. Schematic top view of a SCARA robot. The two degrees of freedom θ_1 and θ_2 are included.

2. SCARA ROBOT PROBLEM FORMULATION

In this section, the SCARA robot, as shown schematically in Fig. 2, is introduced . In Sec. 2.1, a general description of the system is provided. In Sec. 2.2, a non-linear model for the SCARA robot is presented. In Sec. 2.3, the control objectives are introduced. In Sec. 2.4, the actuator sharing problem is stated.

2.1 System Description

The SCARA robot studied in this paper is generally used as an object handler, with high performance demands, leading to strict requirements on the controller. Since this object is typically not mounted physically to the wrist, the lateral forces have to be limited and collisions have to be avoided at all times.

The position of the center of the mass of the object M_{load}, which is to be handled and positioned, can be described in polar coordinates by the radius R and the orientation ϕ, defined as:

$$R = 2L\cos\left(\frac{\theta_1 - \theta_2}{2}\right) + E, \qquad \phi = \frac{\theta_1 + \theta_2}{2}, \tag{1}$$

where L represents the (identical) lengths of both arms, E the length of the wrist, and θ_1 and θ_2 the absolute angles of the upper and lower arm, respectively, as shown in Fig. 2. This results, in the Cartesian coordinate system, in the position $\left[\, x_{\text{load}} \;\; y_{\text{load}} \,\right]^\top \in \mathbb{R}^2$ given by

$$x_{\text{load}} = R\cos(\phi), \qquad y_{\text{load}} = R\sin(\phi). \tag{2}$$

2.2 Model description

A non-linear model for the SCARA robot has been presented in (Steinbuch, 2015) with state

$$x = \left[\, \theta_1 \;\; \theta_2 \;\; \theta_{m,1} \;\; \theta_{m,2} \;\; \dot\theta_1 \;\; \dot\theta_2 \;\; \dot\theta_{m,1} \;\; \dot\theta_{m,2} \,\right]^\top, \tag{3}$$

where θ_1, θ_2, $\theta_{m,1}$ and $\theta_{m,2}$ represent the orientation of the upper arm, lower arm and of the motors driving the upper and lower arm, respectively, and $\dot\theta$ represent the rotational velocity of the corresponding angles. This results in the system dynamics

$$\begin{cases} \dot x = f(x) + Bu + d, \\ y = Cx \end{cases} \tag{4}$$

with $u = \left[\, u^1 \;\; u^2 \,\right]^\top \in \mathbb{R}^2$ the input of the system, i.e., the voltages applied to the amplifiers, d the disturbances, including

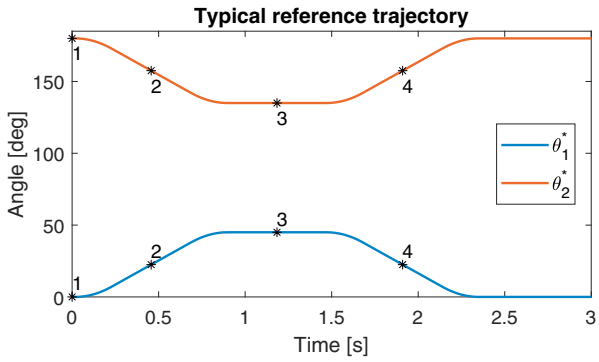


Fig. 3. Typical desired motion for the SCARA, showing the extension ad retraction of the robot. The stars are used later in the paper for the design of the LTI controller.

the friction acting on the robot. For the exact formulation of the nonlinear function $f : \mathbb{R}^8 \to \mathbb{R}^8$ and of the input matrix $B \in \mathbb{R}^{8\times 2}$, see (Steinbuch, 2015). The output matrix $C \in \mathbb{R}^{2\times 8}$ is such that the measured outputs are $y = Cx = [\,\theta_{m,1}\ \theta_{m,2}\,]^\top$.

2.3 Design requirements

The main control objective is to obtain reference tracking, i.e., it would be desirable to have $x_{\text{load}} \approx x^*_{\text{load}}$ and $y_{\text{load}} \approx y^*_{\text{load}}$, where the superscript $*$ indicates the reference signal. A typical desired motion of the system, which complies to a full extension and retraction of the robot arm, is shown in Fig. 3, in terms of the desired reference signals θ_1^* and θ_2^* for the upper and lower arm angles θ_1 and θ_2, respectively. Moreover, in an extending or retracting motion as in Fig. 3, it is desired to maintain the orientation of the load M_{load}, in the sense of keeping ϕ constant (in case of Fig. 3, keeping $\phi \approx 180$ degrees). The reason for this is to avoid collisions into surrounding objects. The performance criteria resulting from these specifications are defined as the line error ϵ_{line} (as a measure for the deviation from ϕ being constant), and the absolute error ϵ_{abs} (as a measure for tracking), respectively. Mathematically, these errors are given by

$$\epsilon_{\text{line}}(t) = \frac{y_{\text{load}}(t)x^*_{\text{load}}(t) - x_{\text{load}}(t)y^*_{\text{load}}(t)}{\sqrt{x^*_{\text{load}}(t)^2 + y^*_{\text{load}}(t)^2}}, \tag{5}$$

$$\epsilon_{\text{abs}}(t) = \sqrt{(x^*_{\text{load}}(t) - x_{\text{load}}(t))^2 + (y^*_{\text{load}}(t) - y_{\text{load}}(t))^2}. \tag{6}$$

The industrial specifications for the design of the controllers are given by

$$|\epsilon_{\text{line}}(t)| \le 1 \text{ mm} \qquad \epsilon_{\text{abs}}(t) \le 10\ \mu\text{m}. \tag{7}$$

The line error requirement needs to hold for all $t \in \mathbb{R}_{\ge 0}$. However, the absolute error specification does not need to hold for all $t \in \mathbb{R}_{\ge 0}$, but only when the SCARA robot is in the fully extended or retracted position. The reason for this is that, in these points, the positioning performance is most critical, since the objects are moved on and from the robot.

It should be noted that the output of the system, as defined in Sec. 2.2, is given by the motor positions $\theta_{m,1}$ and $\theta_{m,2}$. However, the performance requirements are based on the actual orientations θ_1 and θ_2 of the arms. In the SCARA robot , there are no sensors available at the arm positions, but only encoders on actuator axes, providing $\theta_{m,1}$ and $\theta_{m,2}$.

2.4 Resource Sharing Problem Statement

Summarizing the above control objectives combined with the desire of resource sharing as discussed in the introduction lead to the following aim of this paper. Obtain a control strategy that

(i) realizes accurate tracking performance in the sense that $|\epsilon_{\text{line}}| \le 1$mm is realized all the time and $\epsilon_{\text{abs}} \le 10\mu$m in the fully retracted/extended configurations of the robot;
(ii) only requires one shared amplifier (instead of two) for the two actuators to reduce the hardware costs. This implies

$$u^1(t) = 0 \text{ or } u^2(t) = 0 \text{ for all } t \in \mathbb{R}_{\ge 0}. \tag{8}$$

Note that the problem formulated for the SCARA robot is quite generic in nature as the idea of hardware sharing introducing switching constraints as in (8) for nonlinear systems as in (4) can be envisioned to have a broad range of applications.

3. MPC DESIGN

In this section, the design of the MPC strategy is presented. In Sec. 3.1, the general control setup based on linearisation around the desired reference trajectory and discretisation (exploiting the structure of (4)) are discussed leading to an LTV model. In Sec. 3.2, an offset-free formulation for MPC of LTV systems is discussed. Based on this, in Sec. 3.3 we propose an off-set free MPC solution for the resource sharing problem 8.

3.1 Linearisation and discretisation

The aim of the envisioned MPC strategy is to enable tracking of the arms reference trajectories in Fig. 3 given by θ_1^* and θ_2^* including the shared resource constraint. To accomplish this, we computed a desired trajectory for the complete state denoted by x^* and an accompanying input signal u^* that satisfies

$$\dot{x}^*(t) = f(x^*(t)) + Bu^*(t) \tag{9}$$

with

$$z(t) = C_a x^*(t) = \theta^*(t) := [\,\theta_1^*(t)\ \theta_2^*(t)\,]^\top \tag{10}$$

for all $t \in \mathbb{R}_{\ge 0}$ for an appropriate initial condition $x^*(0)$ and $C_a = [I_2\ 0_{2\times 6}]$. Hence, x^* complies with the arms reference trajectories θ_1^* and θ_2^* and the system dynamics (4) (ignoring disturbances).

We will now linearise the SCARA robot model (4) along the reference trajectory (x^*, u^*). Therefore, we define

$$\tilde{x} = x - x^* \text{ and } \tilde{u} = u - u^*. \tag{11}$$

Then we have

$$\begin{aligned}\dot{\tilde{x}}(t) &= f(x(t)) + Bu(t) + d(t) - f(x^*(t)) - Bu^*(t)\\ &\approx \frac{\partial f}{\partial x}(x^*(t))\tilde{x}(t) + B\tilde{u}(t) = A(t)\tilde{x}(t) + B\tilde{u}(t)\end{aligned} \tag{12}$$

with $A(t) = \frac{\partial f}{\partial x}(x^*(t))$ the Jacobian matrix of f at the point $x^*(t)$, $t \in \mathbb{R}_{\ge 0}$. Clearly, this linearisation is only valid when $\tilde{x}(t)$ is small, i.e., when the reference $x^*(t)$ is tracked rather closely by the state $x(t)$. Interestingly, note that $\tilde{u}(t)$ does *not* have to small in order for the linearisation to be a good approximation. The latter is due to the structure of (4), particularly, that the input appears in an affine manner with a constant input matrix B. This is important as due to the resource sharing constraint (8) typically $\tilde{u}(t)$ will *not* be small (every time step we have either $\tilde{u}^1(t) = -u^{1,*}(t)$ or $\tilde{u}^2(t) = -u^{2,*}(t)$, which are typically not small). Hence, the structure in (4) is essential in order for the linearisation to work

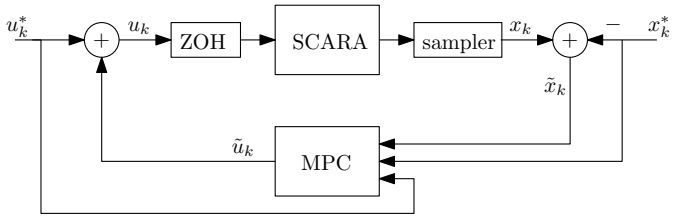


Fig. 4. Control diagram.

in combination with the resource sharing constraint (8). This is rather a crucial observation.

As we strive for a discrete-time MPC setup we discretise the continuous-time linear time-varying (LTV) model (12) with sample time $T_s = 10^{-3}s$ assuming zero-order hold for the input, which leads to the discrete-time LTV model

$$\begin{cases} \tilde{x}_{k+1} &= A_k^d \tilde{x}_k + B_k^d \tilde{u}_k + d_k \\ \tilde{z}_k &= C\tilde{x}_k, \end{cases} \tag{13a}$$

where we added the performance output $\tilde{z}_k$ as the tracking error in the motors positions (i.e., $Cx - \theta_m^*$). In (13) we take

$$A_k^d := e^{A(kT_s)T_s}, \tag{13b}$$

$$B_k^d := \int_0^{T_s} e^{A(kT_s)r} B dr, \tag{13c}$$

$$d_k := \int_{kT_s}^{(k+1)T_s} e^{A(kT_s)((k+1)T_s - r)} d(r) dr \tag{13d}$$

and, obviously, $\tilde{x}_k$ and $\tilde{u}_k$ correspond to $\tilde{x}(kT_s)$ and $\tilde{u}(kT_s)$, respectively, for $k \in \mathbb{N}$. The obtained LTV model (13) (with $d_k = 0$, $k \in \mathbb{N}$) will be used as an LTV prediction model in the envisioned MPC setup. Note that due to the linearisation around the reference trajectory (and not around the actual position, which would lead to an LPV model) the prediction model for each time step can be computed offline, which is a big advantage in an MPC application with a control rate of 1kHz.

In Fig. 4 the envisioned control scheme is shown. Note specifically that the MPC controller requires x^* and u^* to compute the prediction models and constraints.

3.2 Velocity form for LTV systems

Several methods are available to obtain offset-free control in the presence of model-plant mismatch and/or unmodeled disturbances. We will adopt a scheme based on the velocity form, which includes an integral action in the control input optimisation by looking at the variation of state measurements between two time steps, see (Pannocchia, 2015). (Pannocchia, 2015) mostly focusses on linear time-invariant (LTI) models. In this paper we need the velocity form for LTV models as shown in (13), which has a few subtleties compared to the LTI case, as detailed below.

To discuss these subtleties, consider the general LTV model

$$\begin{cases} x_{k+1} &= A_k x_k + B_k u_k \\ y_k &= C_k x_k, \end{cases} \tag{14}$$

and let us define the variables

$$x_k^v = x_k - x_{k-1} \qquad \tilde{y}_k = y_k - \bar{y} \tag{15}$$

$$u_k^v = u_k - u_{k-1} \qquad \xi_k = \begin{bmatrix} x_k^v \\ \tilde{y}_k \end{bmatrix} \tag{16}$$

with $\bar{y}$ a fixed value (in the setup of the previous subsection $\bar{y}$ will be zero, as $\tilde{z}$ is already the tracking error). In the LTI case, i.e., when $A_k = A, B_k = B, C_k = C$ for all $k \in \mathbb{N}$, the system can be rewritten in a standard LTI state-space formulation

$$\begin{cases} \xi_{k+1} &= A^v \xi_k + B^v u_k^v \\ \tilde{y}_k &= C^v \xi_k, \end{cases} \tag{17}$$

with $A^v = \begin{pmatrix} A & 0 \\ C & I \end{pmatrix}$, $B^v = \begin{pmatrix} I \\ 0 \end{pmatrix}$, and $C^v = (0 \ I)$, see (Pannocchia, 2015). If the system is but LTV, as in our case, the velocity form cannot be (straightforwardly) written in a state-space form as in (17). However, it is still possible to write the sequence of predicted future output errors $\begin{bmatrix} \tilde{y}_{k+1}^\top & \tilde{y}_{k+2}^\top & \dots & \tilde{y}_{k+N}^\top \end{bmatrix}^\top$ at time $k \in \mathbb{N}$ as a linear function of

- the previous state x_{k-1},
- the previous input u_{k-1},
- the current state increment x_k^v,
- the current output error $\tilde{y}_k = y_k - \bar{y}$,
- and the sequence of input increments $U_k^v = \begin{bmatrix} (u_k^v)^\top & (u_{k+1}^v)^\top & \dots & (u_{k+N-1}^v)^\top \end{bmatrix}^\top$.

We will show this below, where we denote by I the identity matrix and by 0 the zero matrix of appropriate dimensions. The state increments x_{k+i}^v can be obtained at discrete time $k \in \mathbb{N}$ as follows. Trivially, we have

$$\begin{aligned} x_k^v &= I x_k^v + 0 U_k^v + 0 x_{k-1} + 0 u_{k-1} \\ &:= T_0^k x_k^v + S_0^k U_k^v + M_0^k x_{k-1} + L_0^k u_{k-1}, \end{aligned}$$

where

$$T_0^k = I \qquad S_0^k = 0 \qquad M_0^k = 0 \qquad L_0^k = 0. \tag{18}$$

Moreover, we have for $p \in \mathbb{N}$ that

$$\begin{aligned} x_{k+p+1}^v &= x_{k+p+1} - x_{k+p} = \\ &= A_{k+p} x_{k+p} + B_{k+p} u_{k+p} - A_{k+p-1} x_{k+p-1} \\ &\quad - B_{k+p-1} u_{k+p-1} + (A_{k+p} x_{k+p-1} + \\ &\quad B_{k+p} u_{k+p-1} - A_{k+p} x_{k+p-1} - B_k u_{k+p-1}) = \\ &= A_{k+p} x_{k+p}^v + B_{k+p} u_{k+p}^v + \\ &\quad \underbrace{(A_{k+p} - A_{k+p-1})}_{=:\Delta A_{k+p}} x_{k+p-1} + \\ &\quad \underbrace{(B_{k+p} - B_{k+p-1})}_{=:\Delta B_{k+p}} u_{k+p-1}. \end{aligned} \tag{19}$$

Hence, using (19) multiple times in combination with

$$u_{k+p-1} = u_{k-1} + u_k^v + \dots + u_{k+p-1}^v \tag{20a}$$

$$x_{k+p-1} = x_{k-1} + x_k^v + \dots + x_{k+p-1}^v, \tag{20b}$$

which holds for all $p \in \mathbb{N}_{>0}$, we can write

$$x_{k+p+1}^v = T_{p+1}^k x_k^v + S_{p+1}^k U_k^v + M_{p+1}^k x_{k-1} + L_{p+1}^k u_{k-1}, \tag{21}$$

where the time-varying matrices T_p^k, S_p^k, M_p^k, L_p^k are given through the initialisation in (18) and the recursions

$$\begin{aligned} T_{p+1}^k &= A_{k+p} T_p^k + \Delta A_{k+p} (I + T_1^k + \dots + T_{p-1}^k) \\ S_{p+1}^k &= A_{k+p} S_p^k + \Delta A_{k+p} (S_1^k + \dots + S_{p-1}^k) + \\ &\quad [\underbrace{\Delta B_{k+p} \mid \Delta B_{k+p} \mid \dots \mid \Delta B_{k+p}}_{(p-1) \text{ times}} \mid B_{k+p} \mid 0] \\ M_{p+1}^k &= A_{k+p} M_p^k + \Delta A_{k+p} (I + M_1^k + \dots + M_{p-1}^k) \\ L_{p+1}^k &= A_{k+p} L_p^k + \Delta A_{k+p} (L_1^k + \dots + L_{p-1}^k) + \Delta B_{k+p} \end{aligned}$$

for $k \in \mathbb{N}$ and $p \in \mathbb{N}$. Note that in the LTI case (when $\Delta A_{k+p} = 0$ and $\Delta B_{k+p} = 0$, all $k, p \in \mathbb{N}$), we recover

indeed the expression (21) complying with (17). Similarly, we can write the output deviation $\tilde{y}_k$ as

$$\begin{aligned}\tilde{y}_k &= 0 \cdot x_k^v + 0 \cdot U_k^v + 0 \cdot x_{k-1} + 0 \cdot u_{k-1} + I\tilde{y}_k \\ &=: \bar{T}_0^k x_k^v + \bar{S}_0^k U_k^v + \bar{M}_0^k x_{k-1} + \bar{L}_0^k u_{k-1} + \bar{N}_0^k \tilde{y}_k.\end{aligned}$$

where

$$\bar{T}_0^k = 0 \quad \bar{S}_0^k = 0 \quad \bar{M}_0^k = 0 \quad \bar{L}_0^k = 0 \quad \bar{N}_0^k = I. \tag{22}$$

and

$$\begin{aligned}\tilde{y}_{k+p+1} &= y_{k+p} + (y_{k+p+1} - y_{k+p}) - \bar{y} \\ &= \tilde{y}_{k+p} + C_{k+p+1} x_{k+p+1}^v + \underbrace{(C_{k+p+1} - C_{k+p})}_{=:\Delta C_{k+p+1}} x_{k+p}.\end{aligned}$$

Then, from (20) and (21), we get

$$\begin{aligned}\tilde{y}_{k+p+1} = \bar{T}_{p+1}^k x_k^v + \bar{S}_{p+1}^k U_k^v + \bar{M}_{p+1}^k x_{k-1} + \\ \bar{L}_{p+1}^k u_{k-1} + \bar{N}_{p+1}^k \tilde{y}_k,\end{aligned} \tag{23}$$

where the matrices $\bar{T}_p^k$, $\bar{S}_p^k$, $\bar{M}_p^k$, $\bar{L}_p^k$, and $\bar{N}_p^k$ are obtained via (22) and the recursions

$$\begin{aligned}\bar{T}_{p+1}^k &= \bar{T}_p^k + C_{k+p+1} T_{p+1}^k + \\ &\quad \Delta C_{k+p+1} \left[I + T_1^k + \cdots + T_p^k \right] && \text{(24a)} \\ \bar{S}_{p+1}^k &= \bar{S}_p^k + C_{k+p+1} S_{p+1}^k + \\ &\quad \Delta C_{k+p+1} \left[S_1^k + \cdots + S_p^k \right] && \text{(24b)} \\ \bar{M}_{p+1}^k &= \bar{M}_p^k + C_{k+p+1} M_{p+1}^k + \\ &\quad \Delta C_{k+p+1} \left[I + M_1^k + \cdots + M_p^k \right] && \text{(24c)} \\ \bar{L}_{p+1}^k &= \bar{L}_p^k + C_{k+p+1} L_{p+1}^k + \\ &\quad \Delta C_{k+p+1} \left[L_1^k + \cdots + L_p^k \right] && \text{(24d)} \\ \bar{N}_{p+1}^k &= I. && \text{(24e)}\end{aligned}$$

Using these prediction matrices, we can obtain a prediction model in the velocity form at discrete time $k \in \mathbb{N}$ and prediction horizon N given by

$$\begin{aligned}\begin{bmatrix} \tilde{y}_{k+1} \\ \vdots \\ \tilde{y}_{k+N} \end{bmatrix} = \begin{bmatrix} \bar{T}_1^k \\ \vdots \\ \bar{T}_N^k \end{bmatrix} x_k^v + \begin{bmatrix} \bar{S}_1^k \\ \vdots \\ \bar{S}_N^k \end{bmatrix} U_k^v + \begin{bmatrix} \bar{M}_1^k \\ \vdots \\ \bar{M}_N^k \end{bmatrix} x_{k-1} + \\ \begin{bmatrix} \bar{L}_1^k \\ \vdots \\ \bar{L}_N^k \end{bmatrix} u_{k-1} + \begin{bmatrix} \bar{N}_1^k \\ \vdots \\ \bar{N}_N^k \end{bmatrix} \tilde{y}_k.\end{aligned} \tag{25}$$

It is easy to show that this output prediction is exact even if constant and state disturbances are present in system (14). The prediction model (21)-(25) will be used to develop an offset-free MPC formulation in the next subsection. To do so, observe already that at discrete time $k \in \mathbb{N}$ we need the previous state x_{k-1}, input u_{k-1} and output $\tilde{y}_k$, next to x_k^v. Moreover, as usual, the velocity form (25) has as the optimisation variables the increments of the input signal, i.e., u^v rather than u. However, u can be reconstructed via

$$u_{k+j} = u_{k-1} + \sum_{i=0}^{j} u_{k+i}^v, \tag{26}$$

which allows to translate all constraints that are present on the inputs u in term of the new optimisation vector U^v.

3.3 Offset-free MPC setup

In this subsection we will use the prediction model (25), of course, based on the model (13) with $d_k = 0$, $k \in \mathbb{N}$, i.e., we replace (14) by (13), to setup an offset-free MPC strategy.

	k	k+1	k+2	k+3	k+4	k+5	k+6	k+7	k+8	k+9
Σ_1	1	1	1	0	1	0	1	0	1	0
Σ_2	1	0	1	0	1	0	1	0	1	0
Σ_3	0	1	1	0	1	0	1	0	1	0
Σ_4	0	0	1	0	1	0	1	0	1	0
Σ_5	1	1	0	1	0	1	0	1	0	1
Σ_6	1	0	0	1	0	1	0	1	0	1
Σ_7	0	1	0	1	0	1	0	1	0	1
Σ_8	0	0	0	1	0	1	0	1	0	1

Table 1
Possible switching sequences for $N = 10$ and $N_s = 3$

Hence, u_k, x_k, y_k in (14) become $\tilde{u}_k$, $\tilde{x}_k$ and $C\tilde{x}_k$ (tracking error) in (13). As already mentioned in Subsection 3.2, $\bar{y}$ is 0. The constraints on $\tilde{u}_k$ coming from the resource sharing constraint indicate that only one actuator can be active at each discrete time. The MPC controller should decide upon both the continuous input signal and the discrete switching sequence (which actuator can use the shared resource at each time) over a prediction horizon N. Therefore, we introduce the switching variable $\sigma_k \in \{0, 1\},\ k \in \mathbb{N}$, as

$$\sigma_k = \begin{cases} 1, \text{ if input } u^1 \text{ is active at time } k \ (u_k^2 = 0) \\ 0, \text{ if input } u^2 \text{ is active at time } k \ (u_k^1 = 0). \end{cases} \tag{27}$$

Let $\Sigma_k = [\sigma_k\ \sigma_{k+1} \dots \sigma_{k+N-1}]$. In the following, we use the subscript $k+i|k$ to indicate predictions of the corresponding variables at time $k+i$ made at time $k \in \mathbb{N}$, as is common in the MPC literature. The controller has to solve at each discrete time $k \in \mathbb{N}$ the following optimisation problem:

$$\begin{aligned}\min_{\tilde{U}_{k|k}^v, \Sigma_{k|k}} \quad & J && \text{(28)} \\ s.t. \quad & J = \tilde{z}_{k+N|k}^T P \tilde{z}_{k+N|k} + \\ & \quad + \sum_{i=0}^{N-1} \tilde{z}_{k+i|k}^T Q \tilde{z}_{k+i|k} + \tilde{u}_{k+i|k}^{vT} R \tilde{u}_{k+i|k}^v; \\ & u_m \le u_{k+i|k}^j \le u_M,\ i \in \{1, \dots, N-1\}, j = 1, 2; \\ & \sigma_{k+i|k} = 0 \Rightarrow u_{k+i|k}^1 = 0,\ \ i \in \{1, \dots, N-1\}; \\ & \sigma_{k+i|k} = 1 \Rightarrow u_{k+i|k}^2 = 0,\ \ i \in \{1, \dots, N-1\}; \\ & \sigma_{k+i|k} \in \{0, 1\}, i \in \{1, \dots, N-1\}; \\ & \text{(25), (26)} \\ & \Sigma_{k|k} \in \mathcal{S},\end{aligned}$$

where u_m, u_M are the minimal and maximal value for the control input (related to the maximal voltages), and Q, P, R are positive definite weight matrices, $\mathcal{S}$ is a set of switching sequences along the prediction horizon. One can choose $\mathcal{S}$ to be all the possible 2^N switching sequences, if the complexity of the resulting MIQP problem is not too high. In the application to the SCARA robot, we will reduce the computational burden by adopting a switch horizon N_s in which the controller can freely choose the actuator that is being served for the first N_s steps in the prediction horizon, after which predefined roll-out policies (Bertsekas, 2017), consisting of periodically switching between using the two actuators, are used (σ will switch at every step in the prediction horizon after the switch horizon). This reduces the number of allowable switching sequences to 2^{N_s}. In Tab. 1 the possible switching sequences for $N = 8$ and $N_s = 3$ are shown. This leads to 8 instead of $2^8 = 256$ switching sequences. From (14), (25), (26), knowing x_{k-1}, u_{k-1}, x_k, y_k (and thus x_k^v), it is easy to see that the optimisation problem (28) becomes a mixed-integer quadratic program (MIQP), which will be solved on-line. An optimal solution to (28) is obtained,

denoted by $U^{v,\text{opt}}_{k|k}$, $\Sigma^{\text{opt}}_{k|k}$, at each discrete time $k \in \mathbb{N}$. The optimal solutions are converted into a feedback control strategy using a receding horizon principle, giving

$$\sigma_k = \sigma^{\text{opt}}_{k|k}, \tag{29a}$$

$$u_k = u^{\text{opt}}_{k|k}, \tag{29b}$$

where $u^{\text{opt}}_{k|k}$ can be obtained from

$$\tilde{U}^{v,\text{opt}}_{k|k} = \begin{bmatrix} \tilde{u}^{v,\text{opt}^\top}_{k|k} & \tilde{u}^{v,\text{opt}^\top}_{k+1|k} & \dots & \tilde{u}^{v,\text{opt}^\top}_{k+N-1|k} \end{bmatrix}$$

via (26) and adding $u^*(kT_s)$, see (11).

4. SIMULATION RESULTS

In this section, the MPC controller is implemented on a high-fidelity industrial model of the robot, both with and without switching of the actuators. For comparison, an industrial LTI controller is implemented. This controller design is described in Sec. 4.1, including the extension to the switched case in Sec. 4.1.1. In Sec. 4.2, the simulations are shown.

4.1 LTI controller

As discussed in Sec. 3.1, the system is non-linear, and can be linearised along the desired trajectory, resulting in an LTV system. In (van der Maas et al., 2017), it has been illustrated that a robustly designed linear time-invariant (LTI) controller for a few points along the trajectory (indicated by the stars in Fig. 3), results in a stable controller along the full trajectory.

In particular, the LTI controller is designed using sequential loop closing (Franklin et al., 2006), resulting in the Laplace domain in the controller

$$C(s) = \begin{bmatrix} C_{11}(s) & 0 \\ 0 & C_{22}(s) \end{bmatrix}, \tag{30}$$

where $C_{11}(s)$ and $C_{22}(s)$ are (the transfer functions of) the linear SISO controllers consisting of a series connection of a gain, an integrator, a lead-lag filter, a notch filter, and a low-pass filter. In fact, this results in a slightly retuned version of the controllers used in (van der Maas et al., 2017). Note that $C_{11}(s)$ and $C_{22}(s)$ of the LTI controller are designed for the measured output $y = [\,\theta_{m,1}\ \theta_{m,2}\,]^\top$, while the performance measures are based on the performance output $z = [\,\theta_1\ \theta_2\,]^\top$.

4.1.1. Periodic LTI controller

The LTI controller in Sec. 4.1 is designed for the system with two inputs active at the same time. Since the aim of this research is to eliminate one of the amplifiers, the controller should be redesigned keeping the switching constraint (8) in mind. For comparison, we choose to implement periodic switching, i.e., switch at every sample. On average, this means that each controller is active half of the time. To approach the same performance as in the non-switching situation, the gains of the LTI controllers are increased with a factor 2, such that each actuator gets about the same power on average as for the non-switched LTI controller.

4.2 Simulation Results

All the simulations of this section are performed on the high fidelity model of the industrial SCARA robot (4) for the reference trajectory in Fig. 3.

N \ N_s	0	1	2	3	5	7	10
1	246	147					
4	24.9	6.6	4.9	4.6	6.6		
7	25.2	6.5	4.8	3.8	3.8	5.5	
10	25.2	6.0	4.3	3.2	3.8	4.3	3.8
15	25.2	6.0	4.5	3.8	3.8	4	3.8

Table 2
Average absolute error for different values of N and N_s. The unit of the error is $10^{-7}m$.

First we assume full state measurement to show the potential of the MPC scheme given by (28), with weighing matrices

$$Q = P = \begin{bmatrix} 100 & 0 \\ 0 & 80 \end{bmatrix}, \quad R = \begin{bmatrix} 7 & 0 \\ 0 & 8 \end{bmatrix} \cdot 10^{-5}.$$

Table 2 shows the average absolute error over time obtained for different values of the prediction horizon N and the switching horizon N_s. With $N_s = 0$ we are referring to the case of a periodic switching, i.e. switch at every sample. Next to tracking accuracy, the computational time has to be taken into account. In Fig. 5, in semilogarithmic scale, the average time needed to solve the single optimization problem is shown for two different implementations. The first one makes use of Gurobi, an optimizer for mixed integer programs, see (LLC and GurobiOptimization, 2018). The second is a brute force approach, based on solving 2^{N_s} QPs (one for each admissible switching sequence), with qpas solver, see (Wills, 2007). All the simulations are performed on a laptop with Intel(R)Core i7 processor and 8GB RAM. Considering the computation

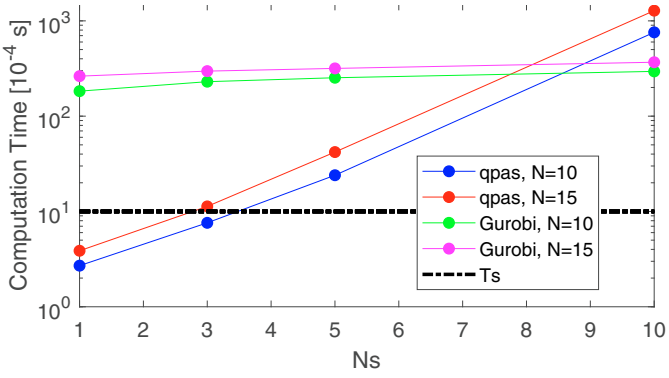


Fig. 5. Average computational times for different controller choices

times in Fig. 5 together with the performance as in Table 2, the horizons have been fixed as $N = 10$ and $N_s = 3$; simulations show that only very rarely the computation time for these parameters exceed the sample time of $T_s = 10^{-3}s$.

Fig. 6 shows the absolute error obtained, including the switching constraint (8), with the velocity form MPC and with a standard, non-offset free, MPC. Since Coulomb friction is acting on the SCARA robot, the output prediction in the standard MPC is incorrect and this leads to an offset in the error. On the contrary, the velocity form MPC is compensating quickly for the friction when the motors are moving (and therefore the disturbance is constant) and it is only showing a small offset for the absolute error ($\leq 10^{-6}m$) in the stick phases, when the Coulomb friction is varying. The benefit of the integral action is immediately evident.

Next, output feedback control simulations, including quantization effects, are performed. In the industrial setting in which the

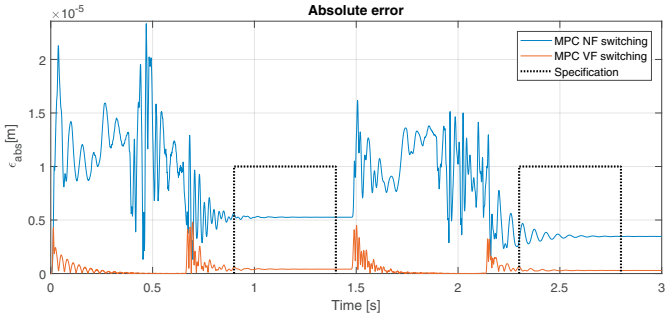
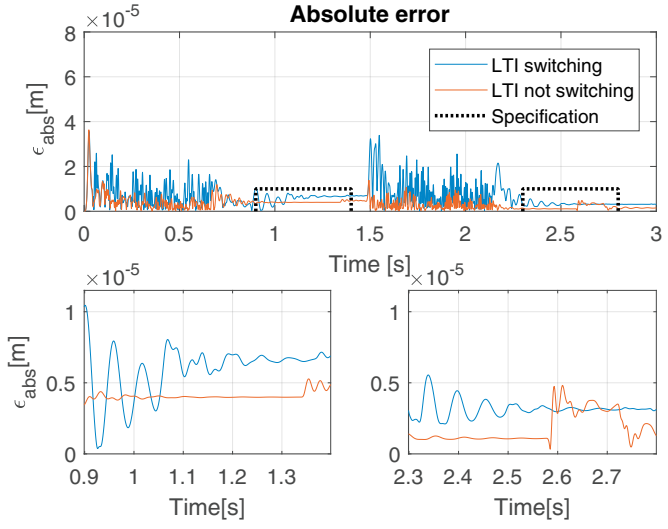


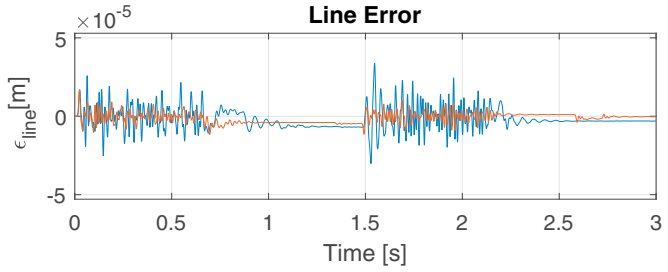
Fig. 6. Absolute error for a standard MPC formulation (NF) and for the velocity-form MPC (VF), assuming state measurement.

SCARA robot under investigation is used, 12-bits encoders are employed and, consequently, the resolution on the motors position measurements is $1.5mrad$. The robot is sampled at $5kHz$. The offset-free MPC is compared to the (periodically switched and non-switched) LTI controller from Sec. 4.1. The design requirements as in Sec. 2.3 are used to evaluate the controllers. The LTI controller is only using motors position measurement, see Sec. 4.1. On the other hand, the MPC controller needs a full state estimation. We adopt an unscented Kalman filter for this aim. To do so, the model (4) has been augmented with a disturbance model, related to constant disturbance torques acting on the motors, and the Coulomb friction has been approximated through a smooth symmetrical sigmoid function. The model is discretised, exploiting the Runge-Kutta 4 method, with sampling rate of $5kHz$ (but note that the control rate is $1kHz$), and used in the Kalman filter. Notice that quantization error is not explicitly incorporated in the observer design. Because of quantization, the observer shows a non zero offset estimation error.

Fig. 7 shows the result for the LTI controller, with and without switching. Fig. 8 shows the result for the velocity-form MPC, with and without switching. The controllers are satisfying the specifications (7) on the absolute and line error, also when considering the resource sharing constraint. When the switching is applied, the performance of the MPC controller is not affected significantly, while the performance loss for the periodically switched LTI controller is evident. In the fully extended (time span $0.9-1.4s$) and retracted (time span $2.3-2.8s$) position the absolute errors of the MPC and LTI controllers are comparable. The MPC controller exhibits peaks in the tracking error, when the motors have to accelerate from standing still and when the reference motors velocity becomes zero. We observed that this kind of peaks in the tracking error are always present when considering Coulomb friction, even when assuming full state knowledge, with both controllers. However, in the MPC they are accentuated because the same points of the trajectory are also critical for the observer. As such, we expect the performance of the MPC controller to improve when a better state estimation is available (see Fig. 6). To verify this in the output feedback case, we consider both the velocity-form MPC and the LTI controller, with switching, when we assume a resolution on the encoders of $0.1mrad$ (i.e., 16 bits encoders), see Fig. 9. Both controllers have improved, but the MPC is outperforming

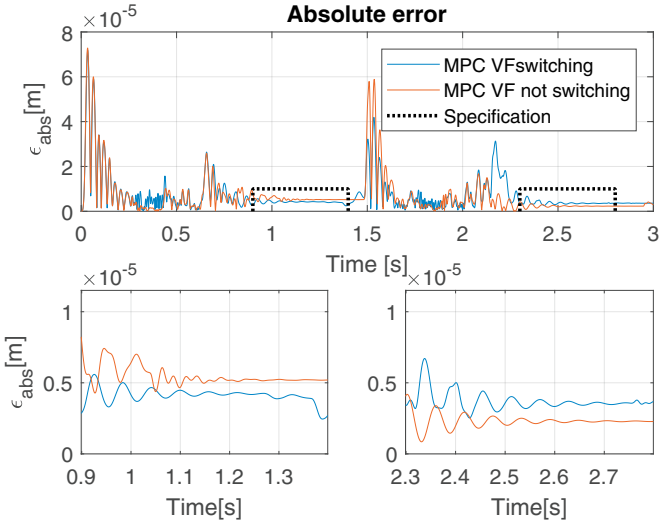


(a) Absolute error, where the lower plots are zooms of the upper plot

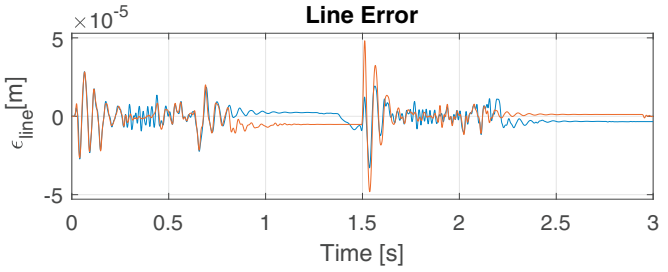


(b) Line error

Fig. 7. Performance of the LTI controller, with and without switching, for an encoders' resolution of $1.5mrad$.



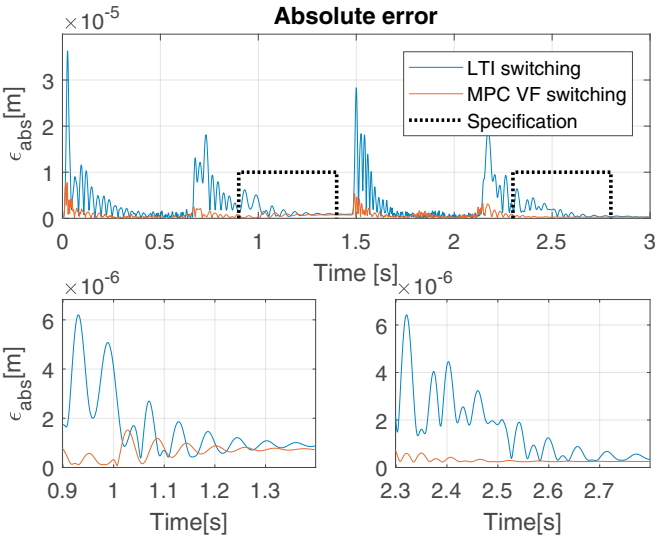
(a) Absolute error, where the lower plots are zooms of the upper plot



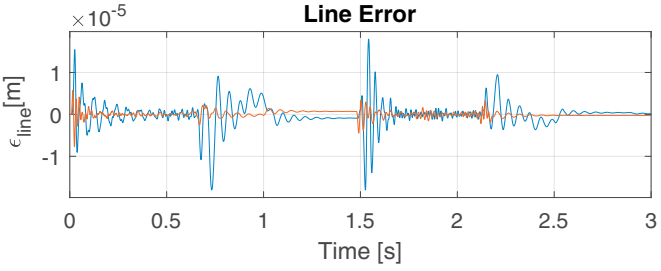
(b) Line error

Fig. 8. Performance of the offset free MPC controller, with and without switching, for an encoders resolution of $1.5mrad$.

the LTI controller. The peaks have reduced of a factor 10 and the absolute errors in the zero velocity positions are smaller than $10^{-6}m$, being ten times smaller than the requirements.



(a) Absolute error, where the lower plots are zooms of the upper plot



(b) Line error

Fig. 9. Comparison between the LTI controller and the offset-free form MPC, for encoders resolution of $0.1mrad$

5. CONCLUSIONS AND FUTURE WORK

In this paper, the feasibility and potential of model predictive control (MPC) for resource sharing in robotic and motion systems was demonstrated. The novel resource-shared MPC framework, applicable to control-affine nonlinear systems, was based on linearisation around a reference trajectory combined with discretisation. The resulting LTV prediction model was reformulated in the velocity form for offset-free MPC, revealing a few subtleties for the LTV case compared to the LTI case. The overall setup of the MPC problem resulted in an MIQP to be solved on-line. The MPC implementation has been compared to a manually tuned switched LTI motion controller with sufficient robustness margins on a high-fidelity model of an industrial SCARA robot. The offset-free MPC strategy performed significantly better than the periodically switched LTI controllers (certainly in case of accurate output measurements), showing the great potential of this advanced control strategy to achieve good performance while reducing the number of required hardware components and thereby the bill of materials of the system. For future work, we envision improvements of the MPC setup, including the Kalman filter. This will be used to perform a full experimental validation of the presented ideas on the SCARA robot (in cooperation with our industrial partners) in order to show the full potential of the proposed methods.

REFERENCES

Azuma, S.i. and Sugie, T. (2008). Synthesis of optimal dynamic quantizers for discrete-valued input control. *IEEE Transactions on Automatic Control*, 53(9), 2064–2075.

Bemporad, A. and Morari, M. (1999). Control of systems integrating logic, dynamics, and constraints. *Automatica*, 35, 407–427.

Bertsekas, D. (2017). *Dynamic Programming and Optimal Control*, volume 1. 4th edition.

Borrelli, F., Baotic, M., Bemporad, A., and Morari, M. (2005). Dynamic programming for constrained optimal control of discrete-time linear hybrid systems. *Automatica*, 41, 1709–1721.

Driessen, A., Wijnands, C., Duarte, J., and Roes, M. (2011). Reduction of the number of power amplifiers for an advanced single-stage planar actuator. In *Proceedings of the 14th European Conference on Power Electronics and Applications and Exhibition*, 1–10. Birmingham, UK, England.

Franklin, G.F., Powell, J.D., and Emami-Naeini, A. (2006). *Feedback Control of Dynamic Systems*. Pearson Education, Inc., Upper Saddle River, New Jersey, USA, 5 edition.

Giorgetti, N., Ripaccioli, G., Bemporad, A., Kolmanovsky, I.V., and Hrovat, D. (2006). Hybrid model predictive control of direct injection stratified charge engines. *IEEE/Asme Transactions on Mechatronics*, 11(5), 499–506.

Lazar, M., Heemels, W., Weiland, S., and Bemporad, A. (2006). Stabilizing Model Predictive Control of Hybrid Systems. *IEEE Transactions on Automatic Control*, 51(11), 1813–1818.

LLC and GurobiOptimization (2018). htpp://www.gurobi.com.

Mhaskar, P., El-Farra, N.H., and Christofides, P.D. (2006). Stabilization of nonlinear systems with state and control constraints using lyapunov-based predictive control. *Systems & Control Letters*, 55(8), 650–659.

Müller, M.A., Martius, P., and Allgöwer, F. (2012). Model predictive control of switched nonlinear systems under average dwell-time. *Journal of Process Control*, 22(9), 1702–1710.

Pannocchia, P. (2015). Offset-free tracking MPC: A tutorial review and comparison of different formulations. In *Proceedings of the 2015 European Control Conference (ECC)*, 527–532. Linz, Austria.

Steinbuch, Y. (2015). Control of a SCARA robot with a reduced number of amplifiers. Technical report, Eindhoven University of Technology.

van der Maas, A., Steinbuch, Y., Boverhof, A., and Heemels, W. (2017). Switched control of a SCARA robot with shared actuation resources. In *IFAC-PapersOnLine*, volume 50, 1931–1936. Elsevier B.V., Toulouse, France.

Wills, A. (2007).

Zhang, L., Zhuang, S., and Braatz, R.D. (2016). Switched model predictive control of switched linear systems: Feasibility, stability and robustness. *Automatica*, 67, 8–21.

Zhang, W., Hu, J., and Abate, A. (2012). Infinite-Horizon Switched LQR Problems in Discrete Time: A Suboptimal Algorithm With Performance Analysis. *IEEE Transactions on Automatic Control*, 57(7), 1815–1821.