MPC and spatial governor for multistage precision manufacturing machines

Di Cairano, S.; Goldsmith, A.M.; Bortoff, S.A.

TR2015-107 September 2015

Abstract

We highlight the potential of model predictive control (MPC) in precision manufacturing by considering the application to multistage processing machines that combine actuators with different operating ranges and different bandwidths to optimize the processing time, product quality, and to increase flexibility. The operation of these machines results in a constrained trajectory generation and control problem with reference-dependent constraints. We propose a design based on a spatial reference governor and a tracking MPC for real-time control of a dual stage processing machine. The method guarantees constraint satisfaction, finite time processing of a given spatial pattern, and real-time execution. Results on a real processing pattern generated using CAD-CAM are shown.

IFAC Conference on Nonlinear Model Predictive Control Conference (NMPC)

MPC and spatial governor for multistage precision manufacturing machines

Stefano Di Cairano * **Abraham Goldsmith** * **Scott A. Bortoff** *

* *Mitsubishi Electric Research Laboratories, Cambridge, MA 02139 USA (e-mail: dicairano,goldsmith,bortoff@merl.com)*

Abstract: We highlight the potential of model predictive control (MPC) in precision manufacturing by considering the application to multistage processing machines that combine actuators with different operating ranges and different bandwidths to optimize the processing time, product quality, and to increase flexibility. The operation of these machines results in a constrained trajectory generation and control problem with reference-dependent constraints. We propose a design based on a spatial reference governor and a tracking MPC for real-time control of a dual stage processing machine. The method guarantees constraint satisfaction, finite time processing of a given spatial pattern, and real-time execution. Results on a real processing pattern generated using CAD-CAM are shown.

Keywords: Model predictive control, manufacturing, tracking control, constraint satisfaction.

1. INTRODUCTION

Following the same path as other industries, such as automotive and aerospace (Hrovat et al. (2012); Di Cairano (2012)), the architectures of precision manufacturing machines are becoming more complex, for instance by combining actuators with large operating range with others that have smaller operating range but higher bandwidth. Such manufacturing machines are multiple input single output systems. Furthermore, the processing machines need to be operated in their entire operational envelop, and hence close to the physical, performance, and safety constraints. As a consequence, modern techniques for multivariable constrained control are being investigated.

MPC is particularly effective for controlling multivariable systems subject to constraints while optimizing a cost function encoding the performance metrics of the controlled system. However, some of the current limitations of MPC in manufacturing relate to the relatively low computing power and memory resources of the microprocessors used in these applications, the high bandwidth of operation of certain controllers, and the performance requirements, which may be quite different from standard control problems. The machine worktool must follow a desired spatial path that represents the machining path of the part being produced. The common performance requirements are the time to process the path, the precision of the actually obtained machining path, which is also affected by the induced vibrations of the motion of the actuators on the machine base, and the energy consumption for the processing. In (Faulwasser and Findeisen (2009); Lam et al. (2013)) MPC algorithms are proposed for contouring control that operate in the spatial domain, and hence require (explicitly or implicitly) the linearization of the spatial nonlinear dynamics.

A class of applications that cannot be straightforwardly handled with existing methods, either classical, or spatial-based MPC, is the control of multistage machines. Multistage machines combine actuators with significantly different bandwidths and operating ranges, up to 500x, so that they can achieve fast processing of parts composed of both detailed and large features. The small-and-fast actuators in the so called *fast stages* provide advantages in rapidly processing the detailed features that require small motions with large accelerations, and the large-and-slow actuators in the so called *slow stages* provide advantages in the processing of large features, that require long motions. The path of the machine is the combination of the motions of the slow and fast stages. Standard trajectory generation methods in factory automation are either specific for single actuator systems, or based on frequency separation (Staroselsky and Stelson (1988)) and hence clearly suboptimal. When constraints are present, the latter enforces them by iterative procedures that must be executed before the machine begins processing. Instead, real-time trajectory generation and control during machine operation is advantageous because it allows increased flexibility of operation without reducing the throughput.

In this paper we design an architecture based on MPC and reference governor to control multistage processing machines, with particular focus on a dual-stage dual-axis machine provided with a small-and-fast actuator and a large-and-slow actuator per processing axis. Since the fast stages operate with update frequencies that are beyond the capabilities of MPC in factory automation microprocessors (e.g., > 100kHz), we exploit the time-scale separation to formulate the control problem of the slow stage as a tracking MPC with constraints that depend on the reference trajectory. Due to such dependency, the feasibility of MPC may require the modification of the reference trajectory which is a recently studied problem (Limon et al. (2008); Ferramosca et al. (2009); Falugi and Mayne (2012)). However, such methods cannot be directly applied to this problem because the modification to the setpoint will cause a modification to the spatial pattern, which results in an incorrectly machined part. In this paper, starting with a trajectory generated using standard methods, e.g., based only on an "ideal" fast-and-large actuator, we exploit a spatial reference governor to obtain the fastest feasible reference trajectory with guaranteed future constraint satisfaction that does not cause machining error while modifying the infeasible parts of the trajectory. Then, we use the reference and the maximum constraint admissible set of the reference governor in the MPC, thus obtaining recursively feasibility, and under mild assumptions, finite time processing of the machined path.

In Section 2 we describe the control problem for the dual-stage machine, in Section 3 we describe the reference governor, and in Section 4 the MPC for controlling the dual stage machine and its real-time implementation. In Section 5 we report simulations of the real machine dynamics on a processing pattern obtained using real CAD and CAM software, where the proposed algorithm is used. Conclusions are summarized in Section 6.

Notation: $\mathbb{R}$, $\mathbb{R}_{0+}$, $\mathbb{R}_+$ and $\mathbb{Z}$, $\mathbb{Z}_{0+}$, $\mathbb{Z}_+$ are the sets of real, nonnegative real, positive real, and integer, nonnegative integer, positive integer numbers, and we use the notation $\mathbb{Z}_{[a,b)} = \{z \in \mathbb{Z} : a \le z < b\}$ to denote intervals. By $[a]_i$ we denote the i-th component of a, for $a \in \mathbb{R}^n$, $b \in \mathbb{R}^m$, $(a, b) = [a' \; b']' \in \mathbb{R}^{n+m}$ is the stacked vector, and I and 0 are the identity and the zero matrices of appropriate size. Relational operators between vectors are intended componentwise, while for matrices denote (semi)definiteness. Given a set $\mathcal{A}$ and $(a, b) \in \mathcal{A}$ we denote by $\mathcal{A}(b)$ the section of $\mathcal{A}$ in the coordinates of b at the values of b. We denote the Minkowski set sum by $\oplus$ and $\eta\mathcal{A}$ where $\eta \in \mathbb{R}_{0+}$ is the scaling of $\mathcal{A}$ by η. $\mathcal{B}(\rho)$, $\rho \in \mathbb{R}_+$ denotes the norm-ball of radius ρ. For a discrete-time signal $x \in \mathbb{R}^n$ with sampling period T_s, x_t is the value at sampling instant t, i.e., at time $T_s t$, and $x_{k|t}$ denotes the predicted value of x at sample $t + k$, i.e., x_{t+k}, based on data at sample t, where $x_{0|t} = x_t$. The operator $*$ denotes the time-domain convolution for systems and signals.

2. MULTI-STAGE PROCESSING MACHINES

While the concepts are generalizable to many applications in precision manufacturing, here we consider a two-stage dual-axis (i.e., 2D) machine that processes raw material into finished parts by a specific worktool. The machine must process at a high rate and with high precision parts that have small and large features, where often multiple copies of the part are made from a single block of raw material. The multistage machine aims at solving two conflicting objectives in precision manufacturing. Due to the small features and the requirement to process at high rate, the worktool must sustain large accelerations. However, due to the large features and multiple copies of the part, the worktool must have a large operating range, and hence a large mass. High accelerations and large mass are obviously in conflict. Thus, multistage machines combine slow and fast stages, as shown in Figure 1.

The fast stages have smaller operating range and smaller mass, so that they can achieve large accelerations, while the slow stages have larger operating ranges and larger mass, and hence can achieve smaller accelerations. With

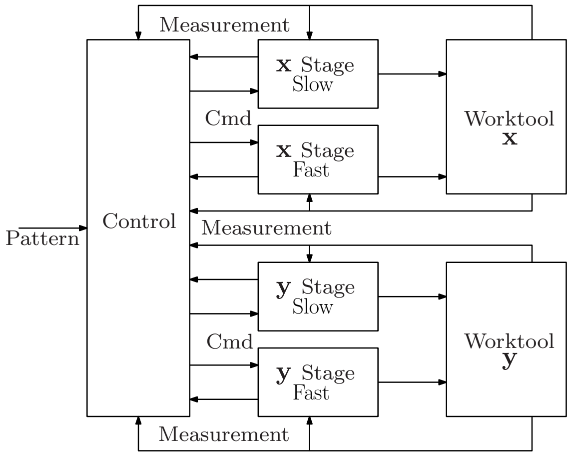

Fig. 1. Dual-stage dual-axis machine architecture.

such an architecture, the small features can be processed quickly by high-acceleration movements of the fast stage, while large features can still be processed by superimposing the large range movements of the slow stage.

The stages can be implemented by different actuators, such as motors, piezoelectric actuators, electromagnetic actuators, all in closed-loop with their servocontrollers. Here, we consider the closed-loop stage model

$$y_j^i(t) = G_j^i(t) * u_j^i(t), \; j \in \{s, f\}, \; i \in \{\mathbf{x}, \mathbf{y}\}, \tag{1}$$

where $*$ is the convolution operator, y is the position, u is the position command, and G is the closed-loop transfer function from position command to position, which has unitary dc-gain, $j \in \{s, f\}$ indexes the stage (slow vs fast) and $i \in \{\mathbf{x}, \mathbf{y}\}$ indexes the axis ($\mathbf{x}$ vs $\mathbf{y}$). The stages are subject to constraints on range

$$-\overline{y}_j^i \le y_j^i \le \overline{y}_j^i, \tag{2}$$

and motion velocity and acceleration

$$-\overline{\dot{y}}_j^i \le \dot{y}_j^i \le \overline{\dot{y}}_j^i, \qquad -\overline{\ddot{y}}_j^i \le \ddot{y}_j^i \le \overline{\ddot{y}}_j^i. \tag{3a}$$

The difference between the slow and fast stages are in the bandwidth of the transfer functions in (1) and in the constraints in (2) and (3), where $\overline{y}_j^f \ll \overline{y}_j^s$, $\overline{\ddot{y}}_j^f \gg \overline{\ddot{y}}_j^s$. The overall position of the worktool is the algebraic sum of positions of the two stages

$$y^i(t) = G_f^i(t) * u_f^i(t) + G_s^i(t) * u_s^i(t), \; i \in \{\mathbf{x}, \mathbf{y}\}. \tag{4}$$

System (1) is controlled in discrete-time, and, due to the different bandwidths, the sampling periods for the fast and slow stage are different, $T_s^f \ll T_s^s$. Specifically, $T_s^s = M \cdot T_s^f$, where $M \in \mathbb{Z}_+$ and $M \gg 1$.

2.1 Tracking control for multistage processing machines

Given a spatial curve representing the pattern to be processed, $p(\sigma) = [p^{\mathbf{x}}(\sigma) \; p^{\mathbf{y}}(\sigma)]'$, $\sigma \in \mathbb{R}_{[0,1]}$, the objective is to control (4) subject to (2), (3) such that

$$\|(y^{\mathbf{x}}(\sigma), y^{\mathbf{y}}(\sigma)) - (p^{\mathbf{x}}(\sigma), p^{\mathbf{y}}(\sigma))\| \le \varepsilon, \; \forall \sigma \in \mathbb{R}_{[0,1]}, \tag{5}$$

i.e., the worktool follows the spatial pattern within a given small tolerance $\varepsilon \in \mathbb{R}_+$.

For solving (5) in single stage machines Faulwasser and Findeisen (2009); Lam et al. (2013) propose a spatial nonlinear MPC, which however does not directly apply to multistage machines, due to timescale separation and non-uniqueness of the trajectory for linearization. Instead, here we first generate by standard method a trajectory $\{q(hT_s^f)\}_h = \{(q^{\mathbf{x}}(hT_s^f), q^{\mathbf{y}}(hT_s^f))\}_h$, $h \in \mathbb{Z}_{0+}$, so that $\tilde{y}(t) = T_s^f(t) * q(t)$ satisfies (3) for $j = f$, (2) for $j = s$, and (5) within the desired $\varepsilon \in \mathbb{R}_{0+}$. Thus, $\{q(hT_s^f)\}_h$ is a trajectory for an ideal fast-and-large single stage machine. Next, we design a controller for (1), (3) such that

$$-\overline{y}_f^i \le y_s^i(t) - q^i(t) \le \overline{y}_f^i,\ i \in \{\mathbf{x}, \mathbf{y}\}. \tag{6}$$

For for $i \in \{\mathbf{x}, \mathbf{y}\}$, this amounts to solving at every sampling period T_s^s, the receding horizon control problem

$$\min_{U_{st}} F(y_{N|t}^i, q_{N|t}^i) + \sum_{k=0}^{N-1} L(y_{sk|t}^i, u_{sk|t}^i, q_{k|t}^i) \tag{7a}$$

$$\text{s.t. } (1),\ (2),\ (3),\ \text{where } j = s \tag{7b}$$

$$-\overline{y}_f^i \le y_{sk|t}^i - q_{k|t}^i \le \overline{y}_f^i\ . \tag{7c}$$

where $N \in \mathbb{Z}_{0+}$ is the prediction horizon, $U_{st}^i = [u_{s0|t}^i \dots u_{sN-1|t}^i]$, F, L are the terminal and stage cost, respectively, and perfect preview of the reference q for at least N steps is available. In (7) the constraints depend on the reference trajectory and thus (recursive) feasibility is not guaranteed.

Problem 1. Given $\{q(hT_s^f)\}_h$, $h \in \mathbb{Z}_{0+}$ that satisfies (3) for $j = f$, (2) for $j = s$, and $\tilde{y}(t) = T_s^f(t) * q(t)$ satisfies (5), compute a modified reference trajectory $\{r(tT_s^s)\}_t = \{(r^{\mathbf{x}}(tT_s^f), r^{\mathbf{y}}(tT_s^s))\}_t$ such that $y(t) = T_s^f(t) * r(t)$ satisfies (5) within $\varepsilon \in \mathbb{R}_{0+}$, and design F, L, and additional constraints for (7) such that when r is substituted for q, (7) is recursively feasible and strictly convex. Also, finite length references $\{q(hT_s^f)\}_{h=0}^{\bar{h}}$ should be processed in finite time.

Problem 1 involves simultaneous reference manipulation and tracking control, and has attracted considerable interest in recent years, see, e.g., Limon et al. (2008); Ferramosca et al. (2009); Falugi and Mayne (2012). In particular, Limon et al. (2008); Ferramosca et al. (2009) developed a virtual setpoint augmented MPC for solving this problem, which maintains feasibility and ensures steady state tracking. Howevr, such an approach is not directly applicable here because it modifies the "shape" of the reference and hence the machining pattern does not enforce (5). Thus, next we develop a"spatial" reference governor in cascade with a tracking MPC.

3. SPATIAL REFERENCE GOVERNOR

In order to obtain a reference trajectory that guarantees the feasibility (7), and ensures satisfaction of (5) we develop a reference governor that operates on the spatial points $\{q(h)\}_h$. First, we recall few useful notions.

Definition 1. Given $x(t+1) = f(x(t))$, $x \in \mathbb{R}^n$, with $z = h(x(t))$, $z \in \mathbb{R}^q$, such that $z \in \mathcal{Z} \subset \mathbb{R}^q$, a constraint admissible set $\mathcal{S}_\infty \subseteq \mathcal{Z}$ is a set such that

$$x(t) \in \mathcal{S}_\infty \Rightarrow h(x(\tau)) \in \mathcal{Z},\ \forall \tau \ge t. \tag{8}$$

Any constraint admissible set $\mathcal{S}_\infty$ is positive invariant (PI) for $x(t+1) = f(x(t))$, that is, if $x \in \mathcal{S}_\infty$, then $f(x) \in \mathcal{S}_\infty$. The maximal constraint admissible set, $\mathcal{O}_\infty$, is the largest constraint admissible set, meaning that there exists no $x \in \mathbb{R}^n$ and constraint admissible set $\mathcal{S}_\infty$, such that $x \in \mathcal{S}_\infty$, and $x \notin \mathcal{O}_\infty$.

For $i \in \{\mathbf{x}, \mathbf{y}\}$ and $j = s$, consider the state space representation of (1), (2), (3), (6), where $u_s^i(t) = q_s^i(t)$. By adding a constant reference dynamics $q^i(t+1) = q^i(t)$, and defining $z^i = C_x^i x^i + C_q q^i = [y_s^i\ \dot{y}_s^i\ \ddot{y}_s^i\ y_s^i - q^i]'$,

$$x^i(t+1) = A^i x^i(t) + B^i q^i(t) \tag{9a}$$

$$q^i(t+1) = q^i(t) \tag{9b}$$

$$z^i(t) = C_x^i x^i(t) + C_q^i q^i(t) \tag{9c}$$

$$H_z^i z^i(t) \le K^i. \tag{9d}$$

Result 1. (Gilbert and Kolmanovsky (2002)). Consider (9), $i \in \{\mathbf{x}, \mathbf{y}\}$, where $C_z = [C_x\ C_q]$, $A_z = \left[\begin{smallmatrix} A^i & 0 \\ 0 & 1 \end{smallmatrix}\right]$, (C_z^i, A_z^i) is observable and $\mathcal{Z}^i = \{z : H_z^i z^i \le K^i\}$ is a polytope, closed and bounded. Let $\mathcal{Q}^i$ be such that all $q^i \in \mathcal{Q}^i$ are steady state admissible, i.e., the corresponding equilibrium $x_e(q^i)$ satisfies $C_x x_e(q^i) + C_q q^i \in \text{int}(\mathcal{Z}^i)$. Then, with arbitrary precision, the maximum output admissible set for $q \in \mathcal{Q}^i$ is a polytope defined by a finite number of constraints

$$\mathcal{O}_\infty^i = \{(x, q) : H_{x\infty}^{\ i} x + H_{q\infty}^{\ i} q \le K_\infty^i\}. \tag{10}$$

We design an algorithm generating a feasible reference based on Result 1. The trajectory $\{q(hT_s^f)\}_h$ is viewed as a sequence of points $\{q(h)\}_h$, and we choose the reference among such points, $r(t) \in \{q(h)\}_h$ for all $t \in \mathbb{Z}_{0+}$. Let

$$\kappa(x, \mu, \{q(h)\}_h) = \max_{\eta \in \mathbb{Z}_{[0,M]}} \quad \eta + \phi \tag{11a}$$

$$\text{s.t.} \quad (x^i, q^i(\eta + \mu)) \in \mathcal{O}_\infty^i \tag{11b}$$

$$i \in \{\mathbf{x}, \mathbf{y}\}, \tag{11c}$$

and let $\mu(t) \in \mathbb{Z}_{0+}$ be the index of the last processed point within the t^{th} sampling interval, i.e., $r(t) = q(\mu(t))$, then

$$\mu(t) = \kappa(x(t), \mu(t-1), \{q(h)\}_h) \tag{12a}$$

$$r^i(t) = q^i(\mu(t)),\ i \in \{\mathbf{x}, \mathbf{y}\}. \tag{12b}$$

The reference governor (12) selects the reference by (11) that indicates how many points can be processed until the next sampling period without violating the constraints and while making sure that the selected reference can be maintained as target without violating the constraints. The maximum M is imposed due to the maximum number of points that can be tracked by the fast stage in the sampling period of the slow stage. In fact, in order to guarantee that (5) is satisfied, starting from $\{q(h)\}_h$ that enforces it, $\{r(t)\}_t$ should not be faster than $\{q(hT_s^f)\}_h$.

Theorem 1. Let $\{q(hT_s^f)\}_{h=0}^{\bar{h}}$ be a finite-time trajectory such that for all $h \in \mathbb{Z}_{[0,\bar{h}]}$ the steady state $x_e^i(q(h))$ for $q(h)$ satisfies $(x_e^i(q(h)), q^i(h+1)) \in \text{int}(\mathcal{O}_\infty^i)$, and let $x(0)$ be such that $(x^i(0), q^i(0)) \in \mathcal{O}_\infty^i$, for $i \in \{\mathbf{x}, \mathbf{y}\}$. Then, (11), is recursively feasible for (9), (12), and $\{r(tT_s^s)\}_t$ is such that if $u_s^i(t) = r^i(t)$, $i \in \{\mathbf{x}, \mathbf{y}\}$, (9d) is satisfied and there exists a finite $\bar{t} \in \mathbb{Z}^+$ with $r(\bar{t}T_s^s) = q(\bar{h}T_s^f)$.

Proof. Constraint satisfaction and recursive feasibility are obtained by induction based on the fact that at every $\tau \in \mathbb{Z}_{0+}$ (12) selects r based on (11) using $\mathcal{O}_\infty$. Due to

Definition 1 if $(x^i(t), r^i(t)) \in \mathcal{O}^i_\infty$, (9) is satisfied at time t, and for $t = 0$, $(x(0), r^i(0)) \in \mathcal{O}_\infty$ for $r^i(0) = q^i(0)$. For some $t \in \mathbb{Z}_{0+}$, let (11) have a feasible. Then $r(t)$ is such that $(x^i(t), r^i(t)) \in \mathcal{O}^i_\infty$. Since $\mathcal{O}^i_\infty$ is PI for (9), $(x^i(\tau), r^i(t)) \in \mathcal{O}^i_\infty$ for all $\tau \geq t$ and (11) is feasible for all $\tau \geq t$. Thus, $(x^i(\tau), r^i(\tau)) \in \mathcal{O}^i_\infty$ for all $\tau \in \mathbb{Z}_{0+}$. As regards finite termination, since $(x^i_e(q(h)), q^i(h+1)) \in \mathrm{int}(\mathcal{O}^i_\infty)$ there exists $\bar{\rho} > 0$ such that $(x^i_e(q(h)) \oplus \mathcal{B}(\rho), q^i(h+1))) \subseteq \mathrm{int}(\mathcal{O}^i_\infty)$. Let $(x(t), r(t)) \in \mathcal{O}_\infty$, and $r(t) = q(\bar{h})$, and assume that for all $\tau \in \mathbb{Z}_{0+}$, $(x(t+\tau), q(\bar{h}+1)) \notin \mathcal{O}_\infty$. Thus, for some $\rho > 0$, $x(t+\tau) \notin x^i_e(q(h)) \oplus \mathcal{B}(\rho)$ for all $\tau \in \mathbb{Z}_{0+}$. However, since A_s is asymptotically stable, for any $\rho \geq 0$ there exists $\tau \in \mathbb{Z}_{0+}$ such that $\|x^i(t+\tau) - x^i_e\| \leq \rho$, which contradicts the previous statement. Then, there must exist a finite $\tau \in \mathbb{Z}_{0+}$, $x(t+\tau) \in \{x^i_e(q(h))\} \oplus \mathcal{B}(\bar{\rho})$ and hence, $(x(t+\tau), q(\bar{h}+1)) \in \mathcal{O}_\infty$, i.e., τ is a finite time for moving to a new point. By repeating for every $q \in \{q(h)\}_{h=0}^{\bar{h}}$, where $\bar{h}$ is finite, the time to process all points is finite.

Remark 1. The reference governor (11), (12), preserve the geometry and satisfies (5) by: (i) selecting only points on the trajectory $\{q(h)\}_h$, and (ii) computing the reference for both axis at the same time, effectively coupling the axis through constraints The proposed reference governor operates a (nonlinear) transformation that "stretches" the time to reduce the processing speed when needed to guarantee constraint satisfaction.

Remark 2. The reference governor guarantees pointwise in time constraint satisfaction, and, this ensures constraint satisfaction for $q(h)$ such that $q(h) = r(t)$ for some $t \in \mathbb{Z}_{0+}$. For the remaining points, constraint satisfaction is not automatically guaranteed, but it can be achieved by either in two ways. A first approach is to including constraints on the intersampling points, or by tightening the constraints accounting for the maximum motion of the fast stage during one sampling period of the slow stage.

4. TRACKING MPC FOR MULTISTAGE MACHINES

Next we re-formulate (7) based on the spatial reference governor in Section 3 to guarantee recursive feasibility. Due to the tracking nature of the problem we formulate the dynamics in input incremental form,

$$\bar{x}(t+1) = \bar{A}\bar{x}(t) + \bar{B}v(t) \tag{13}$$

where $\bar{x} = [x' \ \nu]'$, ν is the 1-step delayed position command for the slow stage, i.e., $\nu(t) = u_s(t-1)$, and $\bar{A} = \left[\begin{smallmatrix} A & B \\ 0 & 1 \end{smallmatrix}\right]$, $\bar{B} = \left[\begin{smallmatrix} B \\ 0 \end{smallmatrix}\right]$, so that the input to (13) is the step-to-step change in the reference $v(t) = u_s(t) - u_s(t-1)$. In (13) we have dropped the superscript $i \in \{\mathbf{x}, \mathbf{y}\}$, as we will do in the rest of this section, for notational simplicity, and similarly we drop the subscript s, e.g., from u_s. When compared to (9), in (13) the reference is no longer part of the dynamics, but it will still be part of the constraints. Given the reference trajectory $R_t = [r_{0|t} \dots r_{N|t}]$, $F(\bar{x}, r) \geq 0$ for all x, r, and $F(\bar{x}, r) = 0$ if and only if $\bar{x} = x_e(r)$, i.e., it is an equilibrium such that $y = r$, $L(\bar{x}, r, v) \geq 0$ for all x, r, v, and $L(\bar{x}, r, v) = 0$ if $\bar{x} = x_e(r)$, and $v = 0$, $\Upsilon_t = [v_{k|t} \dots v_{N-1|t}]$, $\mathcal{X}_\nu \supseteq \mathcal{Q}$ are constraints on commands, $\eta \in \mathbb{R}_{[0,1]}$, the MPC finite horizon optimal control problem is

$$\mathcal{V}(x(t)) = \min_{\Upsilon_t} F(\bar{x}_{N|t}, r_{N|t}) + \sum_{k=0}^{N-1} L(\bar{x}_{k|t}, r_{k|t}, v_{k|t}) \tag{14a}$$

$$\text{s.t. } \bar{x}_{k+1|t} = \bar{A}\bar{x}_{k|t} + \bar{B}v_{k|t} \tag{14b}$$

$$HC_x x_{k|t} + HC_q r_{k|t} \leq K \tag{14c}$$

$$\nu_{k|t} \in \mathcal{X}_\nu \tag{14d}$$

$$x^i_{N|t} \in \eta\mathcal{O}^i_\infty(r^i_{N|t}) \tag{14e}$$

$$\bar{x}_{0|t} = \bar{x}(t). \tag{14f}$$

where $\Upsilon^*_t = [v^*_{k|t} \dots v^*_{N-1|t}]$ denotes the optimal solution. In what follows we denote by $U^*_t = U^*_t(\Upsilon_t, \nu(t))$ the optimal control input sequence at time t corresponding to the optimizer of (14).

Theorem 2. Consider the MPC controller that at any time $t \in \mathbb{Z}_{0+}$ solves (14), where $\eta = 1$, $r_{k|t} = r_{k+1|t-1}$ and $r_{N|t} = q(\mu_{N|t})$, $\mu_{N|t} = \kappa(x_{N|t-1}, \mu_{N|t-1}, \{q(h)\}_h)$. Let (14) be feasible at time $t \in \mathbb{Z}_{0+}$, then (14) is feasible at any $\tau \in \mathbb{Z}_{0+}$, $\tau \geq t$.

Proof. Let (14) be feasible at t and $U^*_t = [u^*_{0|t} \dots u^*_{N-1|t}]$ be the optimal input sequence. At $t+1$, since $x_{N|t} \in \mathcal{O}_\infty(r_{N|t})$, according to Theorem 1, (11) is feasible and $\mu_{N|t+1} = \kappa(x_{N|t}, \mu_{N|t}, \{q(h)\}_h)$, $r_{N|t+1} = q(\mu_{N|t+1})$. Then, $\tilde{U}_{t+1} = [\tilde{u}_{0|t+1} \dots \tilde{u}_{N-1|t+1}]$ where $\tilde{u}_{k|t+1} = u^*_{k+1|t}$ for $k \in \mathbb{Z}_{[0,N-2]}$, and $\tilde{u}_{N-1|t+1} = r_{N|t+1}$ is a feasible input sequence, since the corresponding state trajectory $[\tilde{x}_{0|t+1} \dots \tilde{x}_{N|t+1}]$ is such that $\tilde{x}_{k|t+1} = x^*_{k+1|t}$ for $k \in \mathbb{Z}_{[0,N-1]}$ and $(x^*_{N|t}, r_{N|t+1}) \in \mathcal{O}_\infty$ implies that $(\tilde{x}_{N|t+1}, r_{N|t+1}) \in \mathcal{O}_\infty$, hence (14e) is satisfied. Thus, $\tilde{\Upsilon}_{t+1}$ corresponding to $\tilde{U}_t$ is feasible for (14). The reasoning can be repeated thus completing the proof.

Theorem 3. Let $\{q(hT^f_s)\}_{h=0}^{\bar{h}}$ be a finite-time trajectory such that for all h the equilibrium $x_e(q(h))$ for $q(h)$ satisfies $(x_e(q(h)), q(h+1)) \in \mathrm{int}(\mathcal{O}_\infty)$, and let $(x(0), q(0)) \in \mathcal{O}_\infty$. Consider the MPC that at every iteration solves (14) where $\eta = 1$, $r_{k|t} = r_{k+1|t-1}$ and $r_{N|t} = q(\mu_{N|t})$, $\mu_{N|t} = \kappa(x_{N|t-1}, \mu_{N|t-1}, \{q(h)\}_h)$. Assume that for every $\bar{x}$, r such that $x \in \mathcal{O}_\infty(r)$, $\nu \in \mathcal{X}_\nu$ there exists v such that $F(\bar{A}\bar{x} + \bar{B}v, r) + L(\bar{x}, r, v) - F(\bar{x}, r) < 0$. Then, there exists a finite time when $r_{N|t} = q(\bar{h})$ and $(x_{N|t}, r_{N|t}) \in \mathcal{O}_\infty$.

The full proof is skipped due to limited space, and it is based on the MPC being asymptotically stabilizing and the equilibrium being in the interior of $\mathcal{O}_\infty$, which under the assumptions ensure a finite time transition to each following point, similarly to Theorem 1.

While commonly used for MPC Falugi and Mayne (2012), the assumption on the cost in Theorem 3 may be difficult to verify, as it relates to the existence of a control Lyapunov function (Rawlings and Mayne (2009)). Thus, the following may be preferred.

Theorem 4. Let $x(0)$ be such that $(x(0), q(0)) \in \mathcal{O}_\infty$ and let $\mathcal{O}_\infty(r)$ be λ-contractive, i.e., for all $x \in \mathcal{X}$, $r \in \mathcal{Q}$ such that $x \in \mathcal{O}_\infty(r)$, $Ax + Br \in \lambda\mathcal{O}_\infty(r)$, $0 < \lambda < 1$. For every $h \in \mathbb{Z}_{0+}$, let $q \in \mathcal{Q}$, and if $x \in \lambda\mathcal{O}_\infty(q(h))$, then $x \in \mathcal{O}_\infty(q(h+1))$. Consider the MPC that at every iteration solves (14) where $\eta = \lambda$, $r_{k|t} = r_{k+1|t-1}$ and $r_{N|t} = q(\mu_{N|t})$, $\mu_{N|t} = \kappa(x_{N|t-1}, \mu_{N|t-1}, \{q(h)\}_h)$.

Then, there exists a finite $t \in \mathbb{Z}_+$ when $r_{N|t} = q(\bar{h})$ and $(x_{N|t}, r_{N|t}) \in \mathcal{O}_\infty$, and $t \leq \bar{h}$.

The full proof is skipped due to limited space, and it is based on the contractivity of $\mathcal{O}_\infty$, and the assumptions on the reference.

4.1 Real-time numerical solver algorithm

When F, L are convex quadratic functions, (14) results in a convex parametric quadratic program (pQP)

$$\min_z \ \frac{1}{2} z' Q_p z + \theta' C_p' z + \frac{1}{2} \theta' \Omega_p \theta \qquad (15a)$$

$$\text{s.t. } G_p z \leq S_p \theta + W_p \ . \qquad (15b)$$

where $z = \Upsilon_t$, $\theta \in \mathbb{R}^{n_\theta}$ is the parameter vector and $\theta^i = [\bar{x}' \ R_t']'$. The dual of (15) is the nonnegative pQP

$$\min_U \ \frac{1}{2} \xi' Q_d \xi + \theta^{i\prime} S_d' \xi + W_d' \xi + \frac{1}{2} \theta' \Omega_d \theta \qquad (16a)$$

$$\text{s.t. } \xi \geq 0 \ , \qquad (16b)$$

where $Q_d = G_p Q_P^{-1} G_p'$, $S_d = (G_p Q_p^{-1} C_p + S_p)$, $W_d = W_p$, $\Omega_d = C_p' Q_p^{-1} C_p - \Omega_p$. From the optimal solution Y^* of (16), the solution of (15) is

$$z(\xi^*) = \Psi_{d2p}(\theta, \xi^*) = \Gamma_d \theta + \Xi_d \xi^*, \qquad (17)$$

where $\Xi_d = -Q_p^{-1} G_p'$, $\Gamma_d = -Q_p^{-1} C_p$.

In Di Cairano et al. (2013) it was shown that for solving (16) one can execute the iterations

$$[\xi_{(\ell+1)}]_i = \frac{[(Q_d^- + \phi)\xi_{(\ell)} + F_d^-]_i}{[(Q_d^+ + \phi)\xi_{(\ell)} + F_d^+]_i} [\xi_{(\ell)}]_i \qquad (18)$$

where $F_d = S_d \theta + W_d$, $M_d = \theta' \Omega_d \theta$, $\gamma_d = \Gamma_d \theta$, $K_p = S_p \theta + W_p$, until primal feasibility and zero duality gap are reached within appropriate tolerances. Due to the simple iteration, (18) can be easily implemented and verified even in embedded platforms.

Due to hard real-time requirements, it may be necessary to execute a fixed (possibly small) number $\bar{\ell} \in \mathbb{Z}_+$ of iterations (18), and hence the optimum may not achieved. Let $\xi^{(\bar{\ell})}$ be the candidate solution of (16), $\tilde{z}$ be the corresponding solution of (15) from (17), and $\tilde{U}_t$ be the corresponding control sequence. If $\tilde{U}_t$ is feasible, it is used, because the terminal constraint guarantees recursive feasibility. If $\tilde{U}_t$ is not feasible, we exploit the previous feasible solution and the current reference from (12).

Corollary 1. Let U_{t-1} be a feasible solution for $t \in \mathbb{Z}_+$. The solution $\bar{U}_t$ where $\bar{u}_{k|t} = u_{k+1|t-1}$ and $u_{N|t} = r_{N|t}$ is feasible for (14).

By Corollary 1 we can dimension set a maximum of $\bar{\ell}$ iterations, and then use (12) and Corollary 1 to build a backup feasible solution.

5. CASE STUDY: SIMULATIONS AND RESULTS

Next we show simulations for the dynamics of a real machine with stage time-scale separation of about 2 orders of magnitude, considering real microprocessor computing

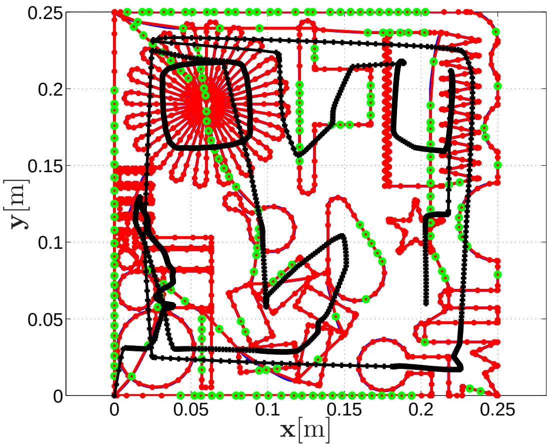

Fig. 2. Processed pattern (red) covering exactly the desired pattern, slow stage motion (black), and points where the reference governor slows the motion (green).

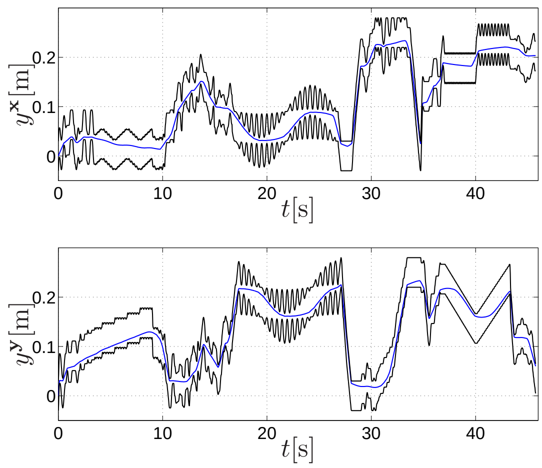

Fig. 3. Position of the slow stage **x**, **y** axes (blue) and constraints (black) due to the reference and fast stage range

capabilities and a real processing pattern.The pattern is obtained from a CAD design of multiple parts, with small and large features. The initial trajectory is generated by a standard CAM algorithm using the dynamics of the fast stage and the operating range of the slow stage, so that it represents an ideal trajectory and a lower bound to the processing time. We design the proposed control algorithm with a prediction horizon of $N = 20$ steps, a ratio of the stage sampling periods $M = 150$, $T_s^s = 30$ms, and a hard limit to $\bar{\ell} = 500$ iterations. The results are reported in Figures 2–5.

Figure 2 shows the processed pattern (which covers precisely the desired processing pattern within the allowed precision of 20μm), the motion of the slow stage obtained by the proposed method, and the points where the reference governor reduces the processed points per sample, to enforce constraints and to guarantee recursive feasibility.

Figure 3 shows the motion of the slow stage, for **x** and **y** coordinates, and the corresponding constraints related to the allowed distance from the filtered reference, which

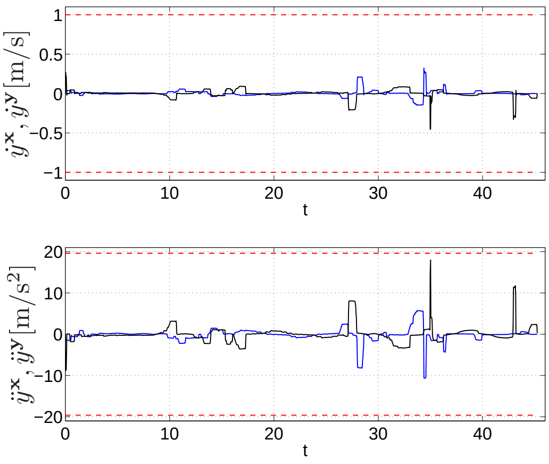

Fig. 4. Velocity and acceleration of the slow stage $\mathbf{x}$ axis (blue), $\mathbf{y}$ axis (black) and constraints (red).

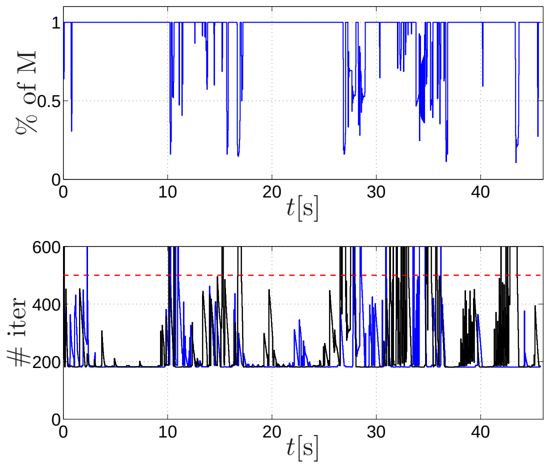

Fig. 5. Percentage with respect to M of points processed by the reference governor trajectory. MPC iterations ($\mathbf{x}$ axis blue, $\mathbf{y}$ axis black), where 600 indicates that Corollary 1 was used.

guarantees that, with the motion of the fast stage, the pattern is effectively tracked. While not shown explicitly, $r^i(t)$ is the average of the constraints on $y_s^i(t)$, $i \in \{\mathbf{x}, \mathbf{y}\}$. The motion of the slow stage is significantly smoother than the reference motion, as can be seen from the slow stage constraints. This reduces the energy consumption and the machine vibrations, increasing machining precision.

The smoothness of the motion of the slow stage can also be noted from Figures 4 showing velocity and acceleration profiles for the $\mathbf{x}$ and $\mathbf{y}$ axes of the slow stage. These are kept far from their actual constraints, despite the processing trajectory being less than 2s, i.e., 5%, longer than the initial ideal (unrealizable) reference trajectory.

Finally, Figure 5 shows the amount of processing allowed at each step by the reference governor , as percentage of the maximum points, $M = 150$, and the number of iterations executed by the QP solver, where the value 600 indicates that the algorithm did not find a feasible solution within the allowed 500 iterations and a feasible solution was obtained from Corollary 1.

6. CONCLUSIONS

We have presented the design of a MPC for controlling dual stage processing machines that demonstrates the potential of MPC in precision manufacturing. The approach is based on exploiting the timescale separation to formulate the problem as the control of a constrained system subject to reference-dependent constraints. We have proposed a spatial reference governor that modifies an ideal, usually infeasible reference to generate a feasible reference profile that preserve the spatial pattern for a tracking MPC. We have shown that such a tracking MPC guarantees constraint satisfaction, is recursively feasible, can guarantee finite-time processing of the spatial pattern, and allows for real-time implementation. We have also shown simulations for real machine parameters on a pattern generated by real CAD-CAM software validating the approach.

REFERENCES

Di Cairano, S. (2012). An industry perspective on MPC in large volumes applications: Potential Benefits and Open Challenges. In *Proc. 4th IFAC Nonlinear Model Predictive Control Conference*, 52–59. Noordwijkerhout, The Netherlands.

Di Cairano, S., Brand, M., and Bortoff, S.A. (2013). Projection-free parallel quadratic programming for linear model predictive control. *International Journal of Control*, 86(8), 1367–1385.

Falugi, P. and Mayne, D.Q. (2012). Tracking performance of Model Predictive Control. In *Proc. 51st IEEE Conf. on Decision and Control*, 2631 – 2636. Maui, HI.

Faulwasser, T. and Findeisen, R. (2009). Nonlinear model predictive path-following control. In *Nonlinear Model Predictive Control*, 335–343. Springer.

Ferramosca, A., Limon, D., Alvarado, I., Alamo, T., and Camacho, E. (2009). MPC for tracking of constrained nonlinear systems. In *Proc. 48th IEEE Conf. on Decision and Control*, 7978–7983. Shangai, China.

Gilbert, E. and Kolmanovsky, I. (2002). Nonlinear tracking control in the presence of state and control constraints: a generalized reference governor. *Automatica*, 38(12), 2063–2073.

Hrovat, D., Di Cairano, S., Tseng, H.E., and Kolmanovsky, I.V. (2012). The development of model predictive control in automotive industry: A survey. In *IEEE Conf. Control Applications*, 295–302.

Lam, D., Manzie, C., and Good, M.C. (2013). Model predictive contouring control for biaxial systems. *IEEE Transactions on Control Systems Technology*, 21(2), 552–559.

Limon, D., Alvarado, I., Alamo, T., and Camacho, E. (2008). MPC for tracking piecewise constant references for constrained linear systems. *Automatica*, 44(9), 2382 – 2387.

Rawlings, J.B. and Mayne, D.Q. (2009). *Model Predictive Control: Theory and Design*. Nob Hill, Madison, Wisconsin.

Staroselsky, S. and Stelson, K.A. (1988). Two-stage actuation for improved accuracy of contouring. In *American Control Conference.*