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 500kHz, 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:** \( \mathbb{R}, \mathbb{R}_+, \mathbb{Z} \), and \( \mathbb{Z}_0^+, \mathbb{Z}_+ \) are the sets of real, nonnegative real, positive real, and integer, nonnegative integer, positive integer numbers, and we use the notation \( \mathbb{Z}_0^+[a, b) = \{ z \in \mathbb{Z} : a \leq z < b \} \) to denote intervals. By \( [a, b) \), we denote the \( i \)-th component of \( a \), for \( a \in \mathbb{R}^n \), \( b \in \mathbb{R}^m \). \( (a, b) = [a' \ b']^T \in \mathbb{R}^{n+m} \) 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 \( A \) and \( (a, b) \in A \) we denote by \( A(B) \) the section of \( A \) in the coordinates of \( b \) at the values of \( b \). We denote the Minkowski set sum by \( \oplus \) and \( \eta A \) where \( \eta \in \mathbb{R}_+ \) is the scaling of \( A \) by \( \eta \). \( B(\rho) \), \( \rho \in \mathbb{R}_+ \) denotes the norm-ball of radius \( \rho \). 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 \( x_{t+k} \) denotes the predicted value of \( x \) at sample \( t + k \), i.e., \( x_{t+k} \), based on data at sample \( t \), where \( x_{0} = x_t \). The operator \( \ast \) 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 into finished parts by a specific worktool. The machine must process at a high rate and with high precision parts into finished parts by a specific worktool. The machine must process at a high rate and with high precision parts into finished parts by a specific worktool. The machine must process at a high rate and with high precision parts into finished parts by a specific worktool. The machine must process at a high rate and with high precision parts into finished parts by a specific worktool.

**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

\[
y_j^s(t) = G_j^s(t) \ast u_j^s(t), ~ j \in \{s, f \}, ~ i \in \{x, y \},
\]

where \( \ast \) 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 \( i \in \{x, y \} \) indexes the axis (\( x \) vs \( y \)). The stages are subject to constraints on range

\[
-\bar{y}_j^s \leq y_j^s \leq \bar{y}_j^s,
\]

and motion velocity and acceleration

\[
-\bar{y}_j^{\dot{}} \leq y_j^{\dot{}} \leq \bar{y}_j^{\dot{}} , \quad \bar{y}_j^{\ddot{}} \leq y_j^{\ddot{}} \leq \bar{y}_j^{\ddot{}} .
\] (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 \( \bar{y}_j^s \ll \bar{y}_j^f, \bar{y}_j^{\dot{s}} \gg \bar{y}_j^{\dot{f}} \).

The overall position of the worktool is the algebraic sum of positions of the two stages

\[
y_j^j(t) = G_j^j(t) \ast u_j^j(t) + G_j^s(t) \ast u_j^s(t), \quad i \in \{x, y \}.
\]

System (1) is controlled in discrete-time, and, due to the different bandwidths, the sampling periods for the fast and slow stage are different, \( T_s^j \ll T_s^f \). Specifically, \( T_s^f = M \cdot T_s^j \), 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, \( p(\sigma) = [p^x(\sigma) \ p^y(\sigma)]^T, \sigma \in \mathbb{R}_{[0,1]} \), the objective is to control (4) subject to (2), (3) such that

\[
||p^x(\sigma) - p^x(\sigma) - p^y(\sigma) - p^y(\sigma)|| \leq \varepsilon, \quad \forall \sigma \in \mathbb{R}_{[0,1]}.
\] (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 Findleisen (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 \( \{g(hT^j_f)\}_h \), \( h \in \mathbb{Z}_{0+} \), so that \( \tilde{y}(t) = T^j_f(t) \ast q(t) \) satisfies (3) for \( j = f, (2) \) for \( j = s, \) and (5) within the desired \( \varepsilon \in \mathbb{R}_{0+}^q. \) Thus, \( \{g(hT^j_f)\}_h \) is a trajectory for an ideal fast-and-large single stage machine. Next, we design a controller for (1), (3) such that

\[
-\mathcal{F} \leq y^s_i(t) - q^j(t) \leq \mathcal{F}, \quad i \in \{x, y\}. 
\]

For \( i \in \{x, y\} \), this amounts to solving at every sampling period \( T^*_s \), the receding horizon control problem

\[
\begin{align*}
\min_{u^s_k} & \quad F(y^s_{N|t}, q^s_{N|t}) + \sum_{k=0}^{N-1} L(g^s_{k|t}, u^s_{k|t}, q^s_{k|t}) \\
\text{s.t.} & \quad (1), (2), (3), \quad \text{where } s = j \quad (7a) \\
-\mathcal{F} & \leq y^s_{k|t} - q^j_k(t) \leq \mathcal{F}, \quad (7b) \\
\end{align*}
\]

where \( N \in \mathbb{Z}_{0+} \) is the prediction horizon, \( U^s = [u^s_{0|t} \ldots u^s_{N-1|t}] \), \( F, L \) 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 \( \{g(hT^j_f)\}_h \), \( h \in \mathbb{Z}_{0+} \) that satisfies (3) for \( j = f, (2) \) for \( j = s, \) and \( \tilde{y}(t) = T^j_f(t) \ast q(t) \) satisfies (5), compute a modified reference trajectory \( \{r(T^*_s)\}_s = \{r^s(T^j_f)^s, \ast r^j(T^j_f)^j\} \), such that \( q(t) = T^j_f(t) \ast r(t) \) satisfies (5) within \( \varepsilon \in \mathbb{R}_{0+}^q, \) 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 \( \{g(hT^j_f)\}_{h=0} \) 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. However, 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 \( z \in \mathcal{Z} \subseteq \mathbb{R}^q, \) a constraint admissible set \( S_x \subseteq \mathbb{R}^n \) is a set such that

\[
x(t) \in S_x \Rightarrow h(x(r)) \in \mathcal{Z}, \forall r \geq t. \tag{8}
\]

Any constraint admissible set \( S_x \) is positive invariant (PI) for \( x(t+1) = f(x(t)), \) that is, if \( x \in S_x, \) then

\[
f(x) \in S_x. \quad \text{The maximal constraint admissible set, } S_x, \quad \text{is the largest constraint admissible set, meaning that there exists no } x \in \mathbb{R}^n \text{ and constraint admissible set } S_x, \quad \text{such that } x \in S_x, \text{ and } x \notin S_x.
\]

For \( i \in \{x, y\} \) and \( j = s, \) consider the state space representation of (1), (2), (3), (6), where \( u^s_i(t) = q^j_i(t). \) By adding a constant reference dynamics \( q^j_i(t+1) = q^j_i(t), \) and defining \( z^j_i = C^s_i x^j_i + C^{q^j_i} q^j_i = [y^s_i, \tilde{y}^s_i, y^s_i, \tilde{y}^s_i], \)

\[
x^j_i(t+1) = A^j_i x^j_i(t) + B^j_i q^j_i(t) \tag{9a}
\]

\[
q^j_i(t+1) = q^j_i(t) \tag{9b}
\]

\[
z^j_i = C^s_i x^j_i(t) + C^{q^j_i} q^j_i(t) \tag{9c}
\]

\[
H^j_i z^j_i(t) \leq K_i. \tag{9d}
\]

**Result 1.** (Gilbert and Kolmanovsky (2002)). Consider (9), \( i \in \{x, y\}, \) where \( C_z = [C_z, C_q], A_z = [A^x_z, A^y_z], (C_z^2, A_z^2) \) is observable and \( Z = \{z : H^j_i z^j \leq K_i\} \) is a polytope, closed and bounded. Let \( Q^j_i \) be such that all \( q^j_i \in Q^j_i \) are steady state admissible, i.e., the corresponding equilibrium \( x^j_i(q^j_i) \) satisfies \( C_x x^j_i(q^j_i) + C^{q^j_i} q^j_i \in \text{int}(Z^j). \) Then, with arbitrary precision, the maximum output admissible set for \( q^j_i \) is a polytope defined by a finite number of constraints

\[
\mathcal{O}^\infty = \{(x, q) : H^j_i x + H^{q^j_i} q \leq K_i\}. \tag{10}
\]

We design an algorithm generating a feasible reference based on Result 1. The trajectory \( \{g(hT^j_f)\}_h \) 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

\[
\kappa(x, \mu, \{q(h)\}_h) = \max_{\eta \in [\mu, x]} \eta + \phi \tag{11a}
\]

s.t. \((x^j_i, q^j_i(\eta + \mu)) \in \mathcal{O}^\infty \tag{11b})

\[
i \in \{x, y\}. \tag{11c}
\]

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

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

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

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, \( r(t) \) should not be faster than \( \{g(hT^j_f)\}_h \).

**Theorem 1.** Let \( \{g(hT^j_f)\}_{h=0} \) be a finite-time trajectory such that for all \( h \in \mathbb{Z}_{0+} \) the steady state \( x^j_i(q(h)) \) for \( q(h) \) satisfies \( (x^j_i(q(h)), q^j_i(\mu(t) + 1)) \in \text{int}(\mathcal{O}^\infty), \) and let \( x(0) \) be such that \( (x(0), q(0)) \in \mathcal{O}^\infty \), for \( i \in \{x, y\}. \) Then, (11), is recursively feasible for (9), (12), and \( \{r(T^*_s)\}_s \) is such that if \( u^s_i(t) = r^j_i(t), i \in \{x, y\}, \) (9d) is satisfied and there exists a finite \( t \in \mathbb{Z}^+ \) with \( r(T^*_s) = q^j_i(T^*_s). \)

**Proof.** Constraint satisfaction and recursive feasibility are obtained by induction based on the fact that at every \( r \in \mathbb{Z}_{0+} \) (12) selects \( r \) based on (11) using \( \mathcal{O}^\infty. \) Due to
For some $\tau \in s$ satisfaction for time constraint satisfaction, and this ensures constraint dynamics in input incremental form, due to the tracking nature of the problem we formulate part of the constraints. Given the reference trajectory $t_r(t)$, $t_r(0) \in \Omega_{\infty}$ for $r'(0) = q'(0)$. For some $t \in \mathbb{Z}_{0+}$, let (11) have a feasible. Then $r(t)$ is such that $(x(t), r'(t)) \in \Omega_{\infty}$. Since $\Omega_{\infty}$ is PI for $(9), (x'(r), r'(t)) \in \Omega_{\infty}$ for all $\tau \geq t$ and $(11)$ is feasible for all $\tau \geq t$. Thus, $(x'(r), r'(t)) \in \Omega_{\infty}$ for all $r \in \mathbb{Z}_{0+}$. As regards finite termination, since $(x'_i(q(h)), q(h) + 1) \in \operatorname{int}(\Omega_{\infty})$ there exists $\rho > 0$ such that $x'_i(q(h)) \in B(\rho), q(h) + 1)) \subseteq \operatorname{int}(\Omega_{\infty})$. Let $(x(t), r(t)) \in \Omega_{\infty}$, and $r(t) = q(h)$, and assume that for all $r \in \mathbb{Z}_{0+}$,

$$t_r = \min_{\mathbb{Z}_{0+}} F(x_{Nt}, r_{Nt}) + \sum_{k=0}^{N-1} L(x_{k[t]}, r_{k[t]}, v_{k[t]})$$

$$\text{s.t. } \bar{x}_{k+1} = A \bar{x}_k + B u_{k[t]}$$

$$H C_q x_{k[t]} + H C q_{k[t]} \leq K$$

$$v_{k[t]} \in \mathcal{X}$$

$$\bar{x}_{Nt} = q_{Nt}$$

$$x_{Nt} = \bar{x}(t).$$

where $Y^*_t = [v^*_{k[t]} \ldots v^*_{Nt-1}]$ denotes the optimal solution. In what follows we denote by $U_{t}^* = U^*_t(\bar{Y}_t, \nu(t))$ 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 $r_{Nt} = q_{Nt}, \mu_{Nt} = \kappa(x_{Nt-1}, \mu_{Nt-1}, \{q(h)\}_h)$. Let (14) be feasible at time $t \in \mathbb{Z}_{0+}$, then (14) is feasible for any $t \in \mathbb{Z}_{0+}, \tau \geq t$.

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

**Theorem 3.** Let $(\{q(hT^*_p)\})^*_{p=0}$ be a finite-time trajectory such that for all $h$ the equilibrium $x_e(q(h))$ for $h$ satisfies $(x_e(q(h)), q(h) + 1) \in \operatorname{int}(\Omega_{\infty})$, and let $(x(0), q(0)) \in \Omega_{\infty}$. Consider the MPC that at every iteration solves (14) where $\eta = 1, r_{k[t]} = r_{k+1[t]-1}$ and $r_{Nt} = q_{Nt}, \mu_{Nt} = \kappa(x_{Nt-1}, \mu_{Nt-1}, \{q(h)\}_h)$. Assume that for every $r$, such that $x \in \mathcal{X}$, there exists $x'$ such that $F(\bar{A} \bar{x} + B \nu, L(e, r, v) - F(x, r, v) < 0$. Then, there exists a finite time when $r_{Nt} = q(h)$ and $(x_{Nt}, r_{Nt}) \in \Omega_{\infty}$.

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 $\Omega_{\infty}$, which under the assumptions ensure a finite time transition to each following point, similarly to Theorem 1.

While commonly used for MPC Fulgur 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 $(x(0), q(0)) \in \Omega_{\infty}$ and let $\mathcal{O}_{X}(r)$ be $\lambda$-contractive, i.e., for all $x \in X, r \in Q$ such that $x \in \mathcal{O}_{X}(r), A\lambda + B\lambda \in \mathcal{O}_{X}(r), 0 < \lambda < 1$. For every $h \in \mathbb{Z}_{0+}, l \in \mathcal{Q},$ and $x \in \mathcal{Q},$ then $x \in \mathcal{O}_{X}(q(h) + 1)).$ Consider the MPC that at every iteration solves (14) where $\eta = 1, r_{k[t]} = r_{k+1[t]-1}$ and $r_{Nt} = q_{Nt}, \mu_{Nt} = \kappa(x_{Nt-1}, \mu_{Nt-1}, \{q(h)\}_h).$
Then, there exists a finite \( t \in \mathbb{Z}_+ \) when \( r_{N[t]} = q(h) \) and \( (x_{N[t]}, r_{N[t]}) \in O_\infty \), and \( t \leq h \).

The full proof is skipped due to limited space, and it is based on the contractivity of \( O_\infty \), 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)

\[
\begin{align*}
\min_{\xi} & \quad \frac{1}{2} z' Q_d \xi + \phi' S_d \xi + W_d \xi + \frac{1}{2} \theta' \Omega_d \theta \\
\text{s.t.} & \quad G_d \xi \leq S_p \theta + W_p .
\end{align*}
\]

where \( z = \mathcal{T}_t \), \( \theta \in \mathbb{R}^n \) is the parameter vector and \( \theta^* = [\bar{x}' \ R_t]' \). The dual of (15) is the nonnegative pQP

\[
\begin{align*}
\min_{\xi} & \quad \frac{1}{2} z' Q_d \xi + \phi' S_d \xi + W_d \xi + \frac{1}{2} \theta' \Omega_d \theta \\
\text{s.t.} & \quad \xi \geq 0,
\end{align*}
\]

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 of (16), the solution of (15) is

\[ z(\xi^*) = \Psi_d \theta, \xi^* = \Gamma_d \theta + \Xi_d \xi^* , \]

where \( \Xi_d = -Q_p^{-1} C_p, \Gamma_d = -Q_p^{-1} C_p \).

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

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

where \( \gamma_d = \Gamma_d \theta, \gamma_d = \Gamma_d \theta, K_p = S_p \theta + W_p \), 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 \( \xi^{(\ell)} \) be the candidate solution of (16), \( \bar{z} \) be the corresponding solution of (15) from (17), and \( \bar{U}_t \) be the corresponding control sequence. If \( \bar{U}_t \) is feasible, it is used, because the terminal constraint guarantees recursive feasibility. If \( \bar{U}_t \) 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 \( \bar{u}_k(t) = u_{k+1}\ell_{t-1} \) 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 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 \( M = 150, T_s = 30 \text{ms} \), and a hard limit to \( \ell = 500 \) iterations. The results are reported in Figures 2–5.

Figure 2 shows the processed pattern (which covers precisely the desired pattern, slow stage motion (black), and points where the reference governor slows the motion (green)).

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
guarantees that, with the motion of the fast stage, the
pattern is effectively tracked. While not shown explicitly,
r^*(t) is the average of the constraints on \( y_i'(t), \ i \in \{x, y\} \).
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 \( x \) and \( y \) 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 \( M \) iter. 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 sys-
tem 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 pat-
tern, 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


in large volumes applications: Potential Benefits and
Open Challenges. In Proc. 4th IFAC Nonlinear Model
Predictive Control Conference, 52–59. Noordwijkerhout,
The Netherlands.

Projection-free parallel quadratic programming for linear
model predictive control. International Journal of
Control, 86(8), 1367–1385.

on Decision and Control, 2631 – 2636. Maui, HI.

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 Deci-
sion and Control, 7978–7983. Shangai, China.

Gilbert, E. and Kolmanovsky, I. (2002). Nonlinear track-
ing control in the presence of state and control con-
straints: 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

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 Predic-
tive Control: Theory and Design. Nob Hill, Madison,
Wisconsin.

actuation for improved accuracy of contouring. In American Control Conferne.