MITSUBISHI ELECTRIC RESEARCH LABORATORIES
https://www.merl.com

Control of Dual-stage Processing Machines by Bounded Tracking-error MPC

Di Cairano, Stefano; Goldsmith, Abraham

TR2016-042 July 06, 2016

Abstract

We consider a dual-stage precision manufacturing machine where a worktool is actuated via a motion system consisting of a "fast" stage with large bandwidth but small operating range, and a "slow" stage with smaller bandwidth but larger operating range. We design a controller based on a recently developed tracking method for constrained systems that guarantees enforcement of constraints and of an assigned bound on the tracking error. For satisfying the controller assumption, we design a reference trajectory generation algorithm that is simple and can also be executed offline. The proposed control system guarantees correct processing of the pattern and finite processing time, for which bounds can be easily computed.

American Control Conference (ACC)

Mitsubishi Electric Research Laboratories, Inc.
201 Broadway, Cambridge, Massachusetts 02139

Control of Dual-stage Processing Machines by Bounded Tracking-error MPC

S. Di Cairano, A. Goldsmith

***Abstract*— We consider a dual-stage precision manufacturing machine where a worktool is actuated via a motion system consisting of a "fast" stage with large bandwidth but small operating range, and a "slow" stage with smaller bandwidth but larger operating range. We design a controller based on a recently developed tracking method for constrained systems that guarantees enforcement of constraints and of an assigned bound on the tracking error. For satisfying the controller assumption, we design a reference trajectory generation algorithm that is simple and can also be executed offline. The proposed control system guarantees correct processing of the pattern and finite processing time, for which bounds can be easily computed.**

I. Introduction

New high performance processing machines feature complex electromechanical architectures, for instance with multiple actuation stages where actuators with different bandwidths and operating ranges are combined to process at high rate large workpieces. Dual-stage machines are equipped with two actuation stages, a "slow" stage with large operating range but small bandwidth and acceleration limits, and a "fast" stage with large bandwidth and acceleration limits but small operating range. For each axis, the overall position of the worktool is the sum of the positions of the two stages along such axis, see Figure 1. Thus, the machine can rapidly process small features of the workpiece by actuating the "fast" stage, and is still able to process large features by superimposing the motion of the "slow" stage.

Trajectory generation and control for dual-stage machines is significantly more complicated than for single stage ones, because it involves multiple input–single output systems subject to constraints on position, velocity, and acceleration. Thus, classical methods based on frequency separation [1] are clearly suboptimal. Instead, model predictive control (MPC) has been proven effective for constrained multivariable systems in several application domains [2], [3]. Nonlinear spatial MPC has been proposed for contouring control of single-stage machines, based on linearizing (implicitly or explicitly) the dynamics along the given path [4], [5]. However, these techniques have limited applicability for multistage machines because linearization causes errors, due to the trajectory not being uniquely defined, and because the time scale separation of the stages results in ill-conditioned numerical optimization problems. For dual-stage processing machines, [6], [7] propose to control only the slow stage subject to additional constraints that limit the distance between the processing path and the slow axis position to be within the range of the fast stage. For guaranteeing feasibility

The authors are with Mitsubishi Electric Research Laboratories, Cambridge, MA, e-mail: dicairano,goldsmith@merl.com

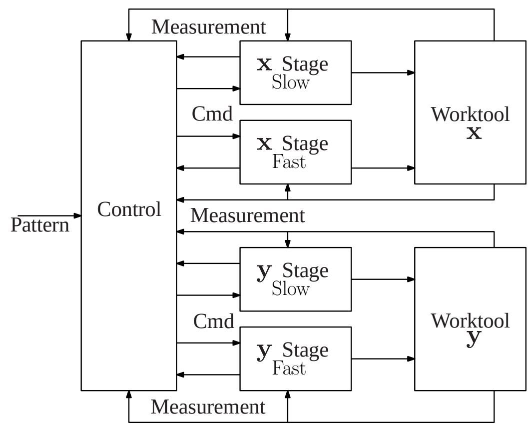

Fig. 1. Architecture of a dual-stage dual-axis processing machine.

of such reference-dependent constraints, a spatial reference governor is developed in [7].

While effective, the method in [7] has still three limitations: (i) the processing time and other performance metrics are suboptimal due to using a reference governor [8], which enforces future constraints by constant commands; (ii) the reference manipulation must be performed in real-time based on the predicted state at the end of the previous prediction horizon, thus adding computations to the real-time control algorithm; (iii) it is not possible to predict before execution the time needed to complete the processing, and finite-time termination may be difficult to enforce.

In this paper we propose a different method aimed at overcoming the above limitations. The method exploits a recently developed control design guaranteeing a tracking error bound for trajectories being the output of constrained non-autonomous linear systems [9]. The proposed method offers the following advantages. First, it is based on robust control invariant sets guaranteeing constraint satisfaction by varying commands, and hence it achieves faster processing because it is less conservative. Second, the reference generation can be done offline (or in parallel) to the trajectory generation and control, thus reducing the real-time computation time. Third, the termination time is known before execution, and a bound can be obtained even before trajectory generation, independently of the chosen cost function. The only drawback is that the robust control invariant set may increase the number of constraints in the MPC problem and hence the computations. This is mitigated by the fact that recursive feasibility is guaranteed for any prediction horizon.

Next, in Section II we describe the dual-stage dual-axis machine and the related tracking control problem, in

Section III we discuss the bounded tracking-error algorithm design for the dual-stage machine. In Section IV we synthesize the controller as an MPC strategy, and discuss its properties. Finally, in Section V we report simulations on a processing pattern obtained from a CAD–CAM software, and we discuss the improvements with respect to the method in [9]. We summarize our conclusions in Section VI.

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 notations like $\mathbb{Z}_{[a,b)} = \{z \in \mathbb{Z} : a \leq z < b\}$ to denote intervals. For $a \in \mathbb{R}^n$, $b \in \mathbb{R}^m$, by $[a]_i$ we denote the i-th component of a, $(a,b) = [a' \ b']' \in \mathbb{R}^{n+m}$ is the stacked vector, and I and 0 are the identity and the zero matrices of appropriate size. 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 $T_s t$, $x_{k|t}$ denotes the predicted value of x at sample $t+k$, i.e., x_{t+k}, based on data at sample t, and $x_{0|t} = x(t)$. We denote the time-domain convolution operator by $*$.

II. Dual-stage Processing Machines: Modeling and Control Architecture

The objective of the dual-stage dual-axis (i.e., 2D) processing machine is to process with a specific worktool blocks of raw material into finished parts. In order to process at high rate the small features, e.g., less than a millimeter, the worktool will be subject to large accelerations, up to several g. Due to the large features, e.g., up to a meter, and possibly multiple parts in a single block, the worktool must operate over a large range, e.g., some meters, and hence will have a large mass. For achieving high precision under large accelerations and with a large mass, the dual-stage machine combines a slow stage and a fast stage, as shown by the schematic in Figure 1.

For the machine in Figure 1, the operating range is the combination of the slow stage and fast stage ranges. The small features of the machined part can be processed by high acceleration movements of the fast stage, which has small mass and small range, while large features can be processed by superimposing movements of the slow stage, which has large range, large mass, and hence limited acceleration. The model of the actuators in closed-loop with their servocontrollers can be described as

$$y^i_j(t) = G^i_j(t) * u^i_j(t),\ j \in \{s,f\},\ i \in \{\mathbf{x},\mathbf{y}\}, \tag{1}$$

where y is the position, u is the position command, $j \in \{s,f\}$ is the index of the stage (slow vs fast), $i \in \{\mathbf{x},\mathbf{y}\}$ is the index of the axis ($\mathbf{x}$ vs $\mathbf{y}$) and G^i_j are the closed-loop transfer functions with dc-gain 1. For the machine architecture considered here, G^i_j are 3^{rd} order functions and the position of the worktool is the sum of the stage positions,

$$y^i(t) = y^i_f(t) + y^i_s(t) = G^i_f(t) * u^i_f(t) + G^i_s(t) * u^i_s(t),\ i \in \{\mathbf{x},\mathbf{y}\}. \tag{2}$$

The stages are subject to symmetric upper and lower bounds on operating ranges,

$$-\overline{y}^i_j \leq \quad y^i_j \quad \leq \overline{y}^i_j, \tag{3}$$

and on velocities and accelerations,

$$-\overline{\dot{y}}^i_j \leq \dot{y}^i_j \leq \overline{\dot{y}}^i_j, \qquad -\overline{\ddot{y}}^i_j \leq \ddot{y}^i_j \leq \overline{\ddot{y}}^i_j. \tag{4a}$$

The difference between the slow and fast stages are in the bandwidth of the transfer functions in (1), where $\mathrm{BW}(G^i_f) \gg \mathrm{BW}(G^i_s)$, $i \in \{\mathbf{x},\mathbf{y}\}$, and in the constraints in (3) and (4), where $\overline{y}^f_j \ll \overline{y}^s_j$, $\overline{\ddot{y}}^f_j \gg \overline{\ddot{y}}^s_j$. Instead, there is no particular relation between the bounds on the velocities of the slow and fast stage in (4) for the machine considered here. Finally, (1) is controlled in discrete-time, and, according to the bandwidths, the sampling period for the slow stage is much longer than that for the fast stage, $T^s_s = M \cdot T^f_s$, where $M \in \mathbb{Z}_+$ and $M \gg 1$.

A. Trajectory generation and control of dual-stage machines

The objective of the trajectory generation and control for dual-stage machines with dynamics (1), (2) and subject to constraints (3), (4) is to compute and make the worktool reproduce a spatial path such that

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

where $p(\sigma) = [p^{\mathbf{x}}(\sigma) \ p^{\mathbf{y}}(\sigma)]'$ is the spatial curve representing the pattern to be processed and $\sigma \in \mathbb{R}_{[0,1]}$ is the curve parametrization variable. Condition (5) requires the worktool to follow the spatial pattern within a small, e.g., micron-range, tolerance $\rho \in \mathbb{R}_+$.

As opposed to nonlinear spatial MPC [4], [5], we formulate a time-based control algorithm and we exploit the time-scale separation to reduce the computational load. By standard methods that are currently implemented in single stage machines, we can generate a trajectory $\{q(hT^f_s)\}_h = \{(q^{\mathbf{x}}(hT^f_s), q^{\mathbf{y}}(hT^f_s))\}_h$, $h \in \mathbb{Z}_{0+}$, so that $\tilde{y}^i(t) = G^i_f(t) * q^i(t)$, $i \in \{\mathbf{x},\mathbf{y}\}$ satisfy (4) for $j = f$, (3) for $j = s$, and (5) for the given $\rho \in \mathbb{R}_{0+}$. Here, $\{q(hT^f_s)\}_h$ is the trajectory of an ideal machine that has the strengths of both stages, the large range of the slow stage and the large bandwidth and high acceleration of the fast stage. Then, we generate a trajectory for the slow stage and ensure that the difference between the slow stage and the processing pattern can be covered by the fast stage. Due to the way by which $\{q(hT^f_s)\}_h$ has been generated and since $\mathrm{BW}(G^i_f) \gg \mathrm{BW}(G^i_s)$, it is enough to control the constrained slow stage such that

$$-\overline{y}^i_f \leq y^i_s(t) - q^i(t) \leq \overline{y}^i_f,\ i \in \{\mathbf{x},\mathbf{y}\}. \tag{6}$$

To this end, we solve with sampling period T^s_s, the receding horizon control problem

$$\min_{U_{st}} \quad F(y^i_{N|t}, q^i_{N|t}) + \sum_{k=0}^{N-1} L(y^i_{s\,k|t}, u^i_{s\,k|t}, q^i_{k|t}) \tag{7a}$$

$$\text{s.t.} \quad (1),\ (3),\ (4),\ \text{where } j = s \tag{7b}$$

$$-\overline{y}^i_f \leq y^i_{s\,k|t} - q^i_{k|t} \leq \overline{y}^i_f, \tag{7c}$$

$$\mathcal{G}(y^i_{s\ k|t}, \dot{y}^i_{s\ k|t}, \ddot{y}^i_{s\ k|t}, q^i_{k|t}, u^i_{k|t}) \leq 0 \tag{7d}$$

where $i \in \{\mathbf{x},\mathbf{y}\}$, $N \in \mathbb{Z}_{0+}$ is the prediction horizon, $U^i_{st} = [u^i_{s0|t} \dots u^i_{sN-1|t}]$, F, L are the terminal and stage cost, respectively, and $\mathcal{G}$ describes additional constraints. Since in (7) the constraints depend on the reference trajectory, recursive feasibility is in general not guaranteed. Here, we aim at solving the following:

Problem 1: Given $\{q(hT_s^f)\}_h$, $h \in \mathbb{Z}_{0+}$ that satisfies (4) for $j = f$, (3) for $j = s$, and $\tilde{y}^i(t) = G_f^i(t) * q(t)$, $i \in \{\mathbf{x}, \mathbf{y}\}$, 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 $y(t) = G_f^i(t) * r(t)$, $i \in \{\mathbf{x}, \mathbf{y}\}$, satisfies (5), and design $\mathcal{G}$ in (7) such that for any convex F, L, the resulting problem (7) where r is substituted for q is convex, recursively feasible, and any finite time reference $\{q(hT_s^f)\}_{h=0}^{\bar{h}}$ is processed in finite time with a known bound. □

Problem 1 involves simultaneous reference generation and tracking control, and has attracted considerable interest in recent years, see, e.g., [10]–[12]. The methods proposed in [10], [11] ensure recursive feasibility by deforming the reference, and hence (5) may not be satisfied. Here we guarantee that (5) is satisfied by basing our design on [9], which does not require the online modification of the reference signal, as long this is generated by satisfying certain assumptions.

III. Bounded Tracking Control Design for Dual-Stage Machines

First, we introduce the following useful definitions, see, e.g., [13] for more details.

Definition 1: Given $x(t+1) = f(x(t), u(t), w(t))$ where $x \in \mathcal{X} \subseteq \mathbb{R}^n$, $u \in \mathcal{U} \subseteq \mathbb{R}^m$ and $w \in \mathcal{W} \subseteq \mathbb{R}^d$ are the state, input and disturbance vectors, respectively, $\mathcal{C} \subseteq \mathcal{X}$ is said to be a robust control invariant (RCI) if

$$\forall x_t \in \mathcal{C}, \exists u_t \in \mathcal{U} : f(x_t, u_t, w_t) \subseteq \mathcal{C},\ \forall w_t \in \mathcal{W},\ \forall t \in \mathbb{Z}_{0+}.$$

The *maximal* RCI set $\mathcal{C}^\infty$ in $\mathcal{X}$ contains all other RCI sets in $\mathcal{X}$. Given the RCI set $\mathcal{C}$, the robustly admissible input (RAI) set for $x \in \mathcal{C}$ is

$$\mathcal{C}_u(x) = \{u \in \mathcal{U} : f(x, u, w) \in \mathcal{C},\ \forall w \in \mathcal{W}\}.$$

When $\mathcal{W} = \{0\}$, $\mathcal{C}$ and $\mathcal{C}_u$ are simply called control invariant and admissible input set, respectively. □

A. Bounded tracking

The objective of the bounded tracking control algorithm is to ensure that a given discrete-time linear plant system subject to constraints tracks within a pre-assigned tracking error bound a reference signal generated by a constrained linear reference system driven by an unknown bounded input. Such a reference system can provide a rich class of reference signals due to being driven by a general input signal. We consider a plant system

$$\begin{aligned} x(t+1) &= Ax(t) + Bu(t) \\ y(t) &= Cx(t), \end{aligned} \tag{8}$$

where $x \in \mathbb{R}^n$, $u \in \mathbb{R}^m$, $y \in \mathbb{R}^p$ are the state, input and output vectors, respectively. System (8) is subject to

$$x \in \mathcal{X},\ u \in \mathcal{U}. \tag{9}$$

We want (8) to track within a given error bound $\epsilon \in \mathbb{R}_+$ the reference signal $r(t)$ generated by the reference system

$$\begin{aligned} \eta(t+1) &= A^\eta \eta(t) + B^\eta \gamma(t) \\ r(t) &= C^\eta \eta(t), \end{aligned} \tag{10}$$

where $\eta \in \mathbb{R}^{n_r}$, $\gamma \in \mathbb{R}^{m_r}$, $r \in \mathbb{R}^p$ are the reference system state, input, and output vectors, respectively, subject to

$$\eta \in \mathcal{R},\ \gamma \in \Gamma. \tag{11}$$

In order to enforce the constraints (11), the input to (10) is selected by a *reference generator algorithm* (RGA). At every $t \in \mathbb{Z}_{0+}$, given $\eta(t) \in \mathcal{C}^\eta$, where $\mathcal{C}^\eta$ is a known CI set of (10), (11), the RGA enforces $\eta(t+1) \in \mathcal{C}^\eta \subseteq \mathcal{R}$ by selecting $\gamma(t) \in \mathcal{C}_\gamma(\eta(t)) \subseteq \Gamma$.

Note that the RGA is independent from the controller state. Furthermore, the RGA only guarantees that (10) will satisfy (11), but gives no guarantees on the satisfiability of the plant or the tracking constraints. In general, any specific RGA that satisfies the above properties can be used.

Problem 2: Consider (8) subject to (9), (10) subject to (11), and a given tracking error bound $\epsilon \in \mathbb{R}_+$. Let $R_t^N = [\eta'_{0|t}, \ldots, \eta(t)'_{N|t}]'$, $N \in \mathbb{Z}_{0+}$ be a predicted reference profile generated by RGA. Design a control law $u_t = \kappa(x_t, R_t^N)$ and a set $\mathcal{X}_0 \subseteq \mathbb{R}^n \times \mathbb{R}^{n_r}$ of initial states and references $(x_0, \eta_0) \in \mathcal{X}_0$ from which (8) in closed-loop with $\kappa(x_t, R_t^N)$ for all $t \in \mathbb{Z}_{0+}$ satisfies (9) and

$$\|y(t) - r(t)\|_\infty \le \epsilon, \tag{12}$$

for any $\eta(t) \in \mathcal{R}$ obtained from the RGA. □

Let $\mathcal{X}^{x,\eta} = \{(x, \eta) : x \in \mathcal{X},\ \eta \in \mathcal{C}^\eta, (Cx - C^\eta \eta) \in \mathcal{B}(\epsilon)\}$, and let $\mathcal{C}^{x,\eta} \subseteq \mathcal{X}^{x,\eta}$ be a RCI set for (8), (10) subject to (9), (11) such that

$$\begin{aligned} &(x, \eta) \in \mathcal{C}^{x,\eta} \Rightarrow \exists u \in \mathcal{U} : \\ &\quad (Ax + Bu, A^\eta \eta + B^\eta \gamma) \in \mathcal{C}^{x,\eta},\ \forall \gamma \in \mathcal{C}_\gamma(\eta). \end{aligned} \tag{13}$$

Indeed, if $(x(t), \eta(t)) \in \mathcal{X}_0 \subseteq \mathcal{C}^{x,\eta}$, there exists $u(t) \in \mathcal{U}$ such that $(x(t+1), \eta(t+1)) \in \mathcal{X}^{x,\eta}$ for every admissible $\eta(t+1)$.

The maximal RCI in (13) is in general non-convex [14] and very hard to compute, and to use in a control algorithms. However in [9] an algorithm for computing a *polyhedral* non-maximal $\mathcal{C}^{x,\eta}$ has been proposed, which is significantly easier to use in optimization-based control algorithms for dual-stage processing machines.

B. Bounded tracking design for dual-stage machine

In order to apply the technique that solves Problem 2 for solving Problem 1 we formulate plant and reference systems as in Section III-A, and design the polyhedral roust control invariant set for tracking. Then, we implement the reference generation algorithm that maximizes the execution speed without deforming the spatial pattern, in order to ensure (5). While the design process is applied to both processing axes, from now on, when possible, we omit the superscript indexing the axes, $i \in \{\mathbf{x}, \mathbf{y}\}$, for simplicity.

The slow stage dynamics is realized in state space form (8). Since for the slow stage considered (1) is of 3^{rd}

order, we choose $x = [y_s\ \dot{y}_s\ \ddot{y}_s]'$, and hence $\mathcal{X} = \{x : -[\bar{y}_s\ \bar{\dot{y}}_s\ \bar{\ddot{y}}_s]' \leq x \leq [\bar{y}_s\ \bar{\dot{y}}_s\ \bar{\ddot{y}}_s]'\}$. We model the bound on the difference between the reference and the slow stage position (6) by (12) with $\epsilon = \bar{y}_f$.

For generating the reference, we formulate a reference system (10) as the constrained integrator

$$\begin{aligned} \eta(t+1) &= \eta(t) + \gamma(t) \\ r(t) &= \eta(t). \end{aligned} \quad (14)$$

where $r = \eta \in \mathcal{R} = \{\eta : \ |\eta| \leq \bar{y}_s\}$, $\gamma \in \Gamma = \{\gamma : \ |\gamma| \leq \bar{\gamma}\}$, and $\bar{\gamma}$ is determined as described next. For this choice, $\mathcal{C}^{\eta\infty} = \mathcal{R}$ and $\mathcal{C}_{\gamma}^{\eta\infty}(\eta) = \{\gamma \in \Gamma : \ \eta + \gamma \in \mathcal{R}\}$. From now on we can interchange r and η since by (14) they are equal.

The value of $\bar{\gamma}$ represents the maximum rate of change of the reference. Thus, if it is too large, the reference may be too fast for the machine to follow, which results in an empty RCI set $\mathcal{C}^{x,\eta}$. On the other hand, if it is too small, the reference moves slowly, which results in loss of productivity. We choose $\bar{\gamma}$ by

$$\bar{\gamma} = \arg\min_{\bar{\gamma}} \quad \Psi(\bar{\gamma}, \mathcal{C}^{x,\eta}) \quad (15a)$$

$$\text{s.t.} \quad \mathcal{C}^{x,\eta} \neq \emptyset \quad (15b)$$

$$g_{\gamma}(\bar{\gamma}) \leq 0 \quad (15c)$$

where (15c) formulates additional constraints on $\bar{\gamma}$ and (15a) determines the objective to be optimized. For the dual-stage machine we consider $\Psi(\bar{\gamma}, \mathcal{C}^{x,\eta}) = -\bar{\gamma}$, which results in the largest bound on the reference rate, and hence the fastest reference motion. In general, (15) is nonconvex, but since there is only one variable, it can be solved gridding and bisection search. While this procedure may be time consuming, it is performed only once at design, and by computing the RCI as in [9] the solution is found in a relatively short time.

C. Reference trajectory generation

The final component for implementing the control strategy is the reference generation algorithm, RGA. As mentioned in Section II, from the spatial pattern $p(\sigma) = [p^{\mathbf{x}}(\sigma)\ p^{\mathbf{y}}(\sigma)]'$, $\sigma \in \mathbb{R}_{[0,1]}$, an ideal trajectory $\{q(hT_s^f)\}_h = \{(q^{\mathbf{x}}(hT_s^f), q^{\mathbf{y}}(hT_s^f))\}_h$, $h \in \mathbb{Z}_{0+}$ is generated for an ideal single stage machine that has the favorable features of both stages. Such an ideal trajectory will in general be infeasible for the actual machine that has a limited range fast stage and a slowly moving slow stage. Thus, the reference generation algorithm needs to slow down the ideal trajectory to make it feasible. For bounded tracking error control, a feasible reference trajectory satisfies (10), (11) for the reference dynamics described in Section III-B.

Let the function $\kappa(\mu, \{q(h)\}_h)$ be defined by

$$\kappa(\mu, \{q(h)\}_h) = \max_{\varsigma \in \mathbb{Z}_{[0,M]}} \quad \mu + \varsigma \quad (16a)$$

$$\text{s.t.} \quad |q^i(\varsigma + \mu) - q^i(\mu)| \leq \bar{\gamma} \quad (16b)$$

$$q^i(\varsigma + \mu) \in \mathcal{C}^{\eta\infty}, \quad (16c)$$

$$i \in \{\mathbf{x}, \mathbf{y}\},$$

where M is the maximum number of commands, i.e., processing points, that can be executed by the fast stage in a sampling period of the slow stage, and let $\mu(t) \in \mathbb{Z}_{0+}$ denote the index of the last processed point within the t^{th} sampling interval, i.e., $r(t) = q(\mu(t))$. We compute the reference at time t as

$$\mu(t) = \kappa(\mu(t-1), \{q(h)\}_h) \quad (17a)$$

$$r^i(t) = q^i(\mu(t)), \ i \in \{\mathbf{x}, \mathbf{y}\} \quad (17b)$$

Equation (17) selects as reference the point in the sequence that: (1) is less than M points away from the last point, (2) is at a distance smaller than $\bar{\gamma}$ from the previous point, in each axis, (3) satisfies the reference system constraints, and (4), maximizes the progress, i.e., the point that makes the counter μ grow larger. Conditions (1)–(4) provide as next reference the point that maximizes the progress in processing, due to (4), while being an admissible value for the reference system, due to (2) and (3), and ensuring that the fast stage can receive the required commands to track the points during the next slow stage sampling period, due to (1). Thus, (17), gives the fastest trajectory that can be processed by the dual-stage machine by the bounded tracking control method in Section III-A.

The optimization problem in (16) is trivial to solve by scanning the ideal trajectory backwards from M points ahead of the last processed points and verifying the satisfaction of the constraints. Verification of the constraints is also simple, because all the constraints are linear. Note that the references are chosen among the points in $\{q(h)\}_h$, which means that the reference generation process does not modify the positions of processing points, but only their timing. This is of key importance because it means that the spatial pattern is not deformed, only its processing speed is reduced to enforce constraints. Hence, starting from an ideal trajectory that satisfies (5), the reference trajectory will also satisfy (5).

Remark 1: It is important to notice that (17), is completely independent of the plant system state, which means that, as opposed to the reference generation method in [7], the RGA does not need to be executed during processing, but it can be performed even before the control algorithm starts.

IV. MPC for Bounded Tracking Control for Dual-Stage Processing Machines

Next, we formulate the control problem (7) using the robust control invariant for bounded tracking (13) designed for the dual-stage machine according to Section III-B, to track there reference generated according to (17).

We formulate the dynamics in input incremental form,

$$\bar{x}(t+1) = \bar{A}\bar{x}(t) + \bar{B}v(t) \quad (18)$$

where $\bar{x} = [x'\ \nu]'$, ν is the 1-step delayed position command for the slow stage, i.e., $\nu(t) = u_s(t-1)$. Thus, the input to (18) is the step-to-step change in the reference $v(t) = u_s(t) - u_s(t-1)$. Given the reference trajectory $R_t = [r_{0|t} \ldots r_{N|t}]$ generated by the RGA (17), the bounded tracking MPC finite horizon optimal control problem is

$$\mathcal{V}(x(t)) =$$

$$\min_{\Upsilon_t} \quad F(\bar{x}_{N|t}, r_{N|t}) + \sum_{k=0}^{N-1} L(\bar{x}_{k|t}, r_{k|t}, v_{k|t}) \quad (19a)$$

$$\text{s.t.} \quad \bar{x}_{k+1|t} = \bar{A}\bar{x}_{k|t} + \bar{B}v_{k|t} \quad (19b)$$

$$(x_{k|t}, r_{k|t}) \in \mathcal{C}^{x,\eta} \quad (19c)$$

$$\bar{x}_{0|t} = \bar{x}(t). \quad (19d)$$

where $F(\bar{x}, \eta) \geq 0$ for all x, r, and $L(\bar{x}, \eta, v) \geq 0$ are convex terminal and stage cost, respectively, $\Upsilon_t = [v_{0|t} \ldots v_{N-1|t}]$ is the optimizer, $\Upsilon_t^* = [v_{0|t}^* \ldots v_{N-1|t}^*]$ is the optimal solution, and $U_t^* = U_t(\Upsilon_t^*, \nu(t))$ is the optimal control input sequence at time t computed from $\nu(t)$ and Υ_t^*. Next, we discuss the properties of the proposed control design, where the proofs are omitted due to limited space.

Theorem 1: Consider the MPC controller that at any time $t \in \mathbb{Z}_{0+}$ solves (19), where $r_{k|t} = r_{k+1|t-1}$ and $r_{N|t} = q(\mu_{N|t})$, $\mu_{N|t} = \kappa(\mu_{N|t-1}, \{q(h)\}_h)$. If (19) is feasible at $t \in \mathbb{Z}_{0+}$, then (19) is feasible at any $\tau \in \mathbb{Z}_{0+}$, $\tau \geq t$. □

Theorem 2: Let $\{q(hT_s^f)\}_{h=0}^{\bar{h}}$ be a finite-time trajectory such that for all h, $q((h+1)T_s^f) - q(hT_s^f) \in \mathcal{C}_\gamma^{\eta\infty}(q(hT_s^f))$. Then, the total processing time obtained by (17), (19) is $\bar{T} \leq \bar{h}T_s^s$. Furthermore, if for all $h \in \mathbb{Z}_{[0,\bar{h}]}$ $\min_h |\mathcal{C}_\gamma^{\eta\infty}(q(hT_s^f))| \geq \varphi > 0$, then the total processing time is also bounded as $\bar{T} \leq T_s^s(\bar{h}/M + \mathcal{L}/\varphi)$, where $\mathcal{L} = \sum_{i \in \{\mathbf{x},\mathbf{y}\}} \mathcal{L}_i$, and $\mathcal{L}_i$ is the traveled distance along i^{th} axis for $\{q(hT_s^f)\}_{h=0}^{\bar{h}}$. A value of φ is $\varphi = \min\{\bar{\gamma}, \min_{i \in \{\mathbf{x},\mathbf{y}\}, \mathbf{h} \in \mathbb{Z}_{[0,\bar{\mathbf{h}}]}} \{|q_i(h) - \bar{y}_i^s|, |q_i(h) - \underline{y}_i^s|\}$. □

Theorem 1 follows from the properties of the RCI set and of the RGA. Theorem 2 follows from the RGA, from the properties of $\mathcal{C}_\gamma^{\eta\infty}(r)$, and provides a valid bound when $\{q(hT_s^f)\}_{h=0}^{\bar{h}}$ does not reach a border of the slow stage range. The bounds computed from Theorem 2 are conservative, but they apply even before the RGA is executed on the pattern, which can be done offline since the RGA does not use the plant system state. Note that for bounding the processing time, besides the pattern length, only the number of points $\bar{h}$ is needed. If the latter is available, the bound on the processing time can be computed from only the pattern $p(\sigma)$, even before the ideal trajectory $\{q(hT_s^f)\}_{h=0}^{\bar{h}}$ is generated. Based on Theorems 1, 2, the following holds directly.

Corollary 1: The control strategy based on (17), (19) solves Problem 1. □

V. SIMULATION RESULTS

The algorithm based on (17), (19) has been designed for a real machine with 2 orders of magnitude time-scale separation between the stages. The simulation pattern is obtained from a CAD design of multiple parts. As previously discussed, $\{q(hT_s^f)\}_h$ is generated by a standard CAM algorithm using the dynamics of the fast stage and the operating range of the slow stage. The algorithm based on (17), (19) is implemented with a prediction horizon of $N = 20$ steps, a ratio of the stage sampling period $M = 150$,

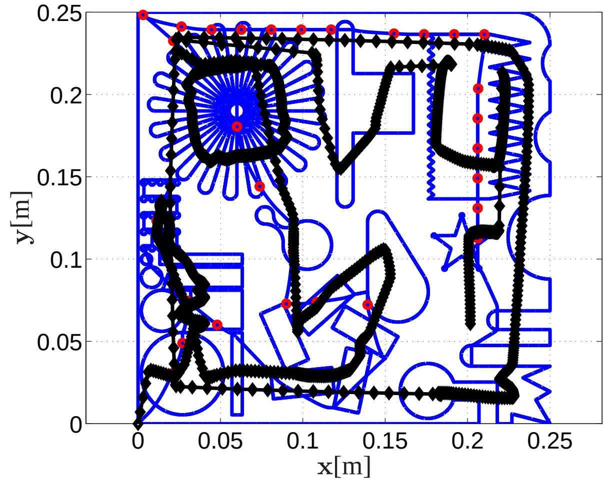

Fig. 2. Processed pattern (red) covering the desired pattern within ρ, slow stage motion (black), and points where the RGA imposes to process less than M=150 points (red).

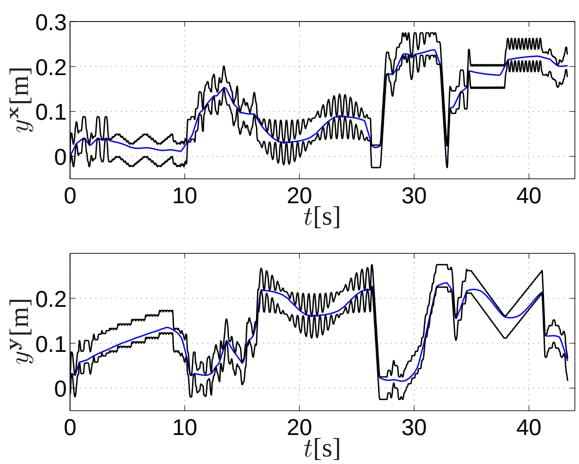

Fig. 3. Position of the slow stage $\mathbf{x}$, $\mathbf{y}$ axes (blue) and constraints (black) due to the reference and fast stage range

and $T_s^s = 30$ms. For the given machine where $\bar{y}_f = 25$mm, $\bar{y}_s = 1$m, $\bar{\dot{y}}_s = 1$m/s, $\bar{\ddot{y}}_s = 19.6$m/s^2, (15) results in $\bar{\gamma} = 0.0183$, giving a maximum reference speed of 611mm/s. The results are reported in Figures 2–5.

Figure 2 shows the processed pattern, which covers the desired pattern within $\rho = 50\mu$m, the motion of the slow stage obtained by the proposed method, and the points where the RGA reduces the processed points per sample to guarantee that (11) is satisfied. Based on Theorem 2, an upper bound to the processing time of 58.6s is obtained, while the actual processing time is 43.2 seconds. The bound is larger, due to the conservativeness of the computations, but still indicative.

Figure 3 shows the slow stage position for $\mathbf{x}$ and $\mathbf{y}$ axes, and the constraints related to the allowed distance from the RGA reference, which guarantees that the pattern is effectively processed by the fast stage.

Table I compares the results obtained by the method proposed here with those obtained by the method in [7],

Method	time$[s]$	max accel.[m/s^2]	mean accel. [m/s^2]	mean vel. [mm/s]	mean track error[mm]
RG-MPC($N = 20$)	45.42	$(10.64, 17.97)$	$(0.75, 0.96)$	$(19.13, 24.72)$	$(12.80, 14.96)$
BT-MPC($N = 20$)	43.20	$(11.14, 11.82)$	$(0.83, 1.05)$	$(21.52, 27.17)$	$(11.81, 13.40)$
BT-MPC($N = 1$)	43.20	$(19.52, 19.52)$	$(1.06, 1.23)$	$(27.26, 31.78)$	$(11.88, 13.71)$

TABLE I

RESULTS $(\mathbf{x}, \mathbf{y})$ FOR THE METHOD PROPOSED HERE (BT-MPC) FOR $N = 20$, $N = 1$ AND THE METHOD IN [7] (RG-MPC) FOR $N = 20$.

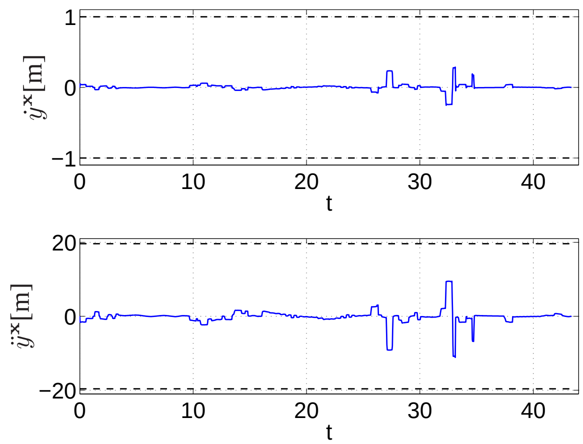

Fig. 4. Velocity and acceleration of the slow stage $\mathbf{x}$ axis (blue) and constraints (black).

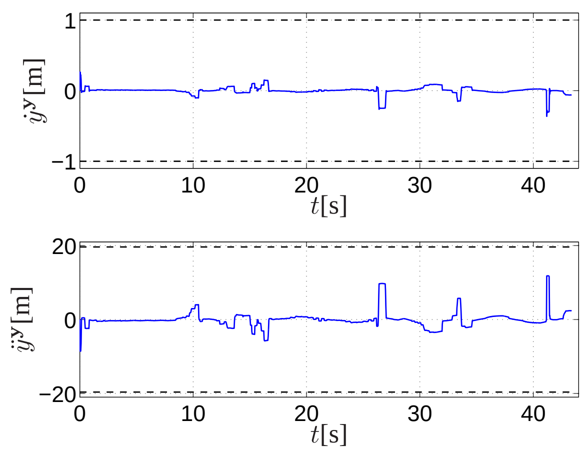

Fig. 5. Velocity and acceleration of the slow stage $\mathbf{y}$ axis (blue) and constraints (black)

both for $N = 20$, and with those obtained with the method proposed here with $N = 1$. When $N = 20$ is considered, the new method reduces the processing time by about 4.5%. Also, the mean tracking error of the slow stage is reduced. To achieve that, the mean acceleration and mean velocity of the slow stage are slightly higher, but still significantly lower than those that would be obtained by a single stage machine. When the horizon is reduced to $N = 1$, the processing time does not change, but the mean acceleration and velocity increase significantly, due to the shorter preview. Finally, it is worth mentioning that despite the RCI constraints introducing some additional effort, the controller implemented with the solver in [15] executes in real time.

VI. CONCLUSIONS

We have proposed a control design for dual-stage dual-axis processing machines for precision manufacturing based on RCI sets. With respect to previously proposed methods, the advantages are the possibility of executing the RGA offline, the use of less conservative constraints, and the guaranteed finite-time termination with an easily computable bound. These result in improved processing time as shown in a realistic case study.

REFERENCES

[1] S. Staroselsky and K. A. Stelson, "Two-stage actuation for improved accuracy of contouring," in *Proc. American Control Confernce*, 1988.

[2] D. Hrovat, S. Di Cairano, H. E. Tseng, and I. V. Kolmanovsky, "The development of model predictive control in automotive industry: A survey," in *Proc. IEEE Conf. Control Appl.*, 2012, pp. 295–302.

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

[4] T. Faulwasser and R. Findeisen, "Nonlinear model predictive path-following control," in *Nonlinear Model Predictive Control*. Springer, Berlin Heidelberg, 2009, pp. 335–343.

[5] D. Lam, C. Manzie, and M. C. Good, "Model predictive contouring control for biaxial systems," *IEEE Trans. Control Systems Technology*, vol. 21, no. 2, pp. 552–559, 2013.

[6] S. Haghighat, S. Di Cairano, D. Konobrytskyi, and S. Bortoff, "Coordinated control of a dual-stage positioning system using constrained model predictive control," in *ASME Dyn. Sys. Control Conf.*, 2014.

[7] S. Di Cairano, A. Goldsmith, and S. Bortoff, "Model predictive control and spatial governor for multistage processing machines in precision manufacturing," in *Proc. 5th IFAC Nonlinear Model Predictive Control Conference*, 2013, pp. 3800–3805.

[8] I. Kolmanovsky, E. Garone, and S. Di Cairano, "Reference and command governors: A tutorial on their theory and automotive applications," in *American Control Conf.* IEEE, 2014, pp. 226–241.

[9] S. Di Cairano and F. Borrelli, "Constrained tracking with guaranteed error bounds," in *Proc. 52nd IEEE Conf. Decision and Control*, 2013, pp. 3800–3805.

[10] D. Limon, I. Alvarado, T. Alamo, and E. Camacho, "MPC for tracking piecewise constant references for constrained linear systems," *Automatica*, vol. 44, no. 9, pp. 2382 – 2387, 2008.

[11] A. Ferramosca, D. Limon, I. Alvarado, T. Alamo, and E. Camacho, "MPC for tracking of constrained nonlinear systems," in *Proc. 48th IEEE Conf. Decision and Control*, 2009, pp. 7978–7983.

[12] P. Falugi and D. Q. Mayne, "Tracking performance of Model Predictive Control," in *Proc. 51st IEEE Conf. Decision and Control*, 2012.

[13] F. Blanchini and S. Miani, *Set-theoretic methods in control*. Springer Science & Business Media, 2007.

[14] S. Raković, E. Kerrigan, D. Mayne, and J. Lygeros, "Reachability analysis of discrete-time systems with disturbances," *IEEE Trans. Automatic Control*, vol. 51, no. 4, pp. 546–561, 2006.

[15] S. Di Cairano, M. Brand, and S. A. Bortoff, "Projection-free parallel quadratic programming for linear model predictive control," *International Journal of Control*, vol. 86, no. 8, pp. 1367–1385, 2013.