**49th IEEE Conference on Decision and Control**
**December 15-17, 2010**
**Hilton Atlanta Hotel, Atlanta, GA, USA**

# Model Predictive Contouring Control

Denise Lam, Chris Manzie and Malcolm Good

*Abstract*— **Biaxial contouring systems involve competing control objectives of maximising accuracy while minimising traversal time. In this paper, a model predictive controller for contouring systems is proposed where the control inputs are determined by minimising a cost function which reflects the trade-off between these competing objectives, subject to state and actuator constraints. To facilitate real-time implementation, a linear time-varying approach is proposed, and stability is guaranteed by introducing an additional contraction constraint. Simulation results for an XY table system demonstrate the effectiveness of the proposed contouring control scheme.**

## I. INTRODUCTION

Control of multi-axis contouring systems involves accurate, high speed tracking of a predetermined geometric path. Industrial applications include machine tool control and laser profiling. Such systems are often subject to actuator constraints which limit the acceleration capabilities of the machine.

In contouring applications, a control objective is to minimise contouring error, defined as the minimum distance between the current position and the desired path. Cross-coupling control is a technique which explicitly seeks to minimise contouring error by adding contour error compensation to the axis control inputs [1]. In traditional contouring systems, the desired path is converted to a time-dependent reference trajectory offline which is then tracked online using feedback controllers. It is desired to traverse the path at high speed to maximise productivity. However, due to the constraints and dynamics of the system, this may lead to reduced accuracy. As a result, time optimal planning of the reference trajectory is of significant interest.

A number of researchers have proposed adjusting the speed of the reference trajectory such that the contour follows a desired geometric path in minimum time. Offline trajectory optimisation routines based on acceleration and velocity constraints were proposed in [2], and later [3] with the addition of jerk constraints. In the context of robot motion control, the combined trajectory and control input optimisation is reformulated into a convex optimisation problem utilising a dynamic model of the system [4]. All of these offline methods are purely feedforward and cannot handle modelling errors and disturbances.

In contouring applications there is a trade-off between productivity and accuracy. For example, it is sometimes desirable to sacrifice contouring accuracy to allow the path to be traversed faster. In [5], the reference trajectory and the tuning parameters of the feedback controller were optimised together such that the tracking accuracy was below a specified tolerance, for a given contour. Naturally, as the tolerance is increased, the time to complete the trajectory decreases. In [6], optimal tracking for piezo-based nanopositioners was implemented by minimising a cost function representing the trade-off between control effort and accuracy. Hence, the system was allowed to deviate from the desired path resulting in reduced control effort, but with a fixed speed reference trajectory.

Path following control is a feedback control scheme where the controller determines the velocity of the reference trajectory as well as the control inputs online. However, path following approaches [7] do not take actuator constraints into account. Recently, a new path following control framework based on model predictive control (MPC) has been proposed [8]. It allows for optimisation of the reference trajectory and the system inputs online in a receding horizon fashion, subject to actuator and state constraints. The utilisation of feedback at each time step allows for modelling errors and disturbances to be rejected, under certain conditions. However, since nonlinear MPC is used in [8], finding a real-time solution to the optimisation problem is difficult.

In this work the framework of model predictive path-following control is extended to suit high speed biaxial contouring control applications. The model predictive structure allows for constraint handling and potential disturbance rejection. The trade-off between productivity and accuracy is addressed using the MPC cost function, allowing the system to deviate from the desired path in order to increase productivity. The weights in the cost function determine the relative importance of the competing control objectives. A linear time-varying formulation is used to reduce computational complexity.

## II. THE CONTOURING CONTROL PROBLEM

Consider the following linear discrete time system which describes the dynamics of a biaxial contouring system

$$\xi_{k+1} = A\xi_k + Bu_k,\ \xi_k = \begin{bmatrix} x_k & y_k & \eta_k \end{bmatrix}^T \qquad (1)$$

where $x_k \in \mathbb{R}$ and $y_k \in \mathbb{R}$ denote the $x$ and $y$ axis displacements at time $k$, $\eta_k \in \mathbb{R}^{n-2}$ denotes the system's internal states and $u_k \in \mathbb{R}^{n_u}$ is the system input. The system is subject to input and state constraints $u \in \mathscr{U}, \xi \in \mathscr{X}$, where $\mathscr{U}$ and $\mathscr{X}$ are closed convex polytopes containing the origin, and in addition $\mathscr{U}$ is bounded.

The objective is to steer $(x_k, y_k)$ along a continuously differentiable and bounded two-dimensional geometric path

Denise Lam, Chris Manzie and Malcolm Good are with the Department of Mechanical Engineering, The University of Melbourne, Victoria, Australia `d.lam@pgrad.unimelb.edu.au`, `manziec@unimelb.edu.au`, `mcgood@unimelb.edu.au`

**978-1-4244-7746-3/10/$26.00 ©2010 IEEE**

$(x_d(\theta), y_d(\theta))$:

$$x_d : [\theta^s, 0] \to \mathbb{R};\;\; y_d : [\theta^s, 0] \to \mathbb{R};\;\; \theta^s < 0 \tag{2}$$

The contouring error $\epsilon_k^c$ is defined as the normal deviation from the desired path [1], and can be expressed as

$$\begin{aligned} \epsilon_k^c &= \sin\phi(\theta_r)\,(x_k - x_d(\theta_r)) - \cos\phi(\theta_r)\,(y_k - y_d(\theta_r))\,, \\ \phi(\theta_r) &= \arctan\left(\frac{\nabla y_d(\theta_r)}{\nabla x_d(\theta_r)}\right), \end{aligned} \tag{3}$$

where $\theta_r(x, y)$ is the value of the path parameter where the distance between the point $(x_d(\theta_r), y_d(\theta_r))$ and $(x, y)$ is minimal, as per Fig. 1. The multi-objective control problem involves selecting the control input $u$ such that the solutions of (1) traverse near the desired geometric path, minimising contouring error while maximising path speed.

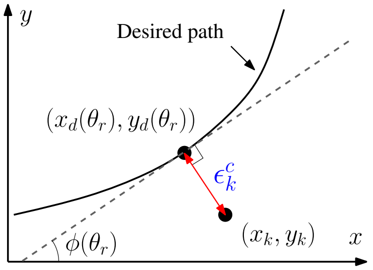


Fig. 1. Contouring error

## III. MODEL PREDICTIVE CONTOURING CONTROL

It is proposed to extend model predictive path-following control (MPFC) [8] to the contouring control problem. It is assumed that the desired path $(x_d(\theta), y_d(\theta))$ is parameterised by arc length, i.e. $ds/d\theta = 1$, where $s$ denotes the distance travelled along the path. Arc length parameterisation of general curves is nontrivial, however techniques exist in the literature for approximate arc length parameterisation of spline curves; see for example [9]. The system (1) is augmented with the following dynamics

$$\theta_{k+1} = \theta_k + v_k,\;\; v_k \in [0, v_{max}],\;\; v_{max} > 0, \tag{4}$$

where $v_k$ is a virtual input to be determined by the controller and $\theta_k$ denotes the value of the path parameter at time $k$. Since the path is parameterised by arc length, $v$ is directly proportional to the path speed. Also, non-reversal of the path is guaranteed, since $v_k \geq 0$.

It is proposed to use $\theta_k$, whose evolution is governed by (4), as an approximation to $\theta_r(x_k, y_k)$. The contouring error is then approximated by

$$\hat{\epsilon}^c(\xi_k,\theta_k) = \sin\phi(\theta_k)(x_k - x_d(\theta_k)) - \cos\phi(\theta_k)(y_k - y_d(\theta_k)) \tag{5}$$

Let $\epsilon^l$ denote the path distance that $(x_d(\theta_r), y_d(\theta_r))$ lags $(x_d(\theta_k), y_d(\theta_k))$ and approximate $\epsilon^l$ as

$$\hat{\epsilon}^l(\xi_k,\theta_k) = -\cos\phi(\theta_k)(x_k - x_d(\theta_k)) - \sin\phi(\theta_k)(y_k - y_d(\theta_k)) \tag{6}$$

Refer to Fig. 2 for a graphical interpretation of $\epsilon_c$, $\epsilon_l$ and their approximations.

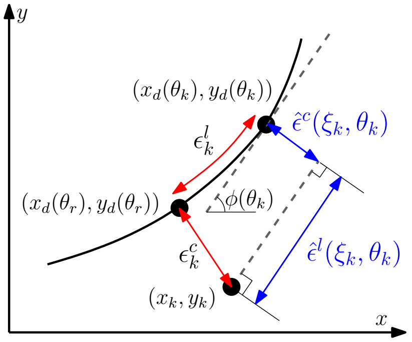

$y$
$(x_d(\theta_k), y_d(\theta_k))$
$\epsilon_k^l$
$\hat{\epsilon}^c(\xi_k, \theta_k)$
$(x_d(\theta_r), y_d(\theta_r))$
$\phi(\theta_k)$
$\epsilon_k^c$
$\hat{\epsilon}^l(\xi_k, \theta_k)$
$(x_k, y_k)$
$x$

Fig. 2. Contouring error, lag and their approximations

From Fig. 2, it can be observed that $\theta_k \approx \theta_r(x_k, y_k)$ if $\hat{\epsilon}^l(\xi_k, \theta_k) \approx 0$. Therefore, to aid in the problem formulation, it is desired to select $v_k$ such that $\hat{\epsilon}^l \approx 0$. Note that while $\theta_r$ in Figure 1 is not necessarily unique, the smooth evolution of $\theta_k$ enforced by the constraint on $v_k$ ensures that the system follows the path smoothly, provided $v_{max}$ is chosen to be sufficiently small.

Model predictive control involves minimisation of a cost function over a prediction horizon of $N$ time steps. The cost function represents the control objectives and their relative importance. In the context of contouring control, the competing objectives include minimising contouring error while maximising the path distance travelled at each time step in the horizon. In addition, to allow $\theta_k$ to be used as an approximation to $\theta_r$, it is desired that $\hat{\epsilon}^l(\xi_k, \theta_k) \approx 0$.

The cost function $J_k$ is therefore chosen to represent the trade-off between contouring accuracy and path speed, as well as penalising control input deviations:

$$\begin{aligned} J_k = \sum_{i=1}^{N} \Bigg( & \begin{bmatrix} \hat{\epsilon}^c(\xi_{k+i}, \theta_{k+i}) \\ \hat{\epsilon}^l(\xi_{k+i}, \theta_{k+i}) \end{bmatrix}^T Q \begin{bmatrix} \hat{\epsilon}^c(\xi_{k+i}, \theta_{k+i}) \\ \hat{\epsilon}^l(\xi_{k+i}, \theta_{k+i}) \end{bmatrix} - q_\theta \theta_{k+i} \\ & + \begin{bmatrix} \Delta u_{k+i} \\ \Delta v_{k+i} \end{bmatrix}^T R \begin{bmatrix} \Delta u_{k+i} \\ \Delta v_{k+i} \end{bmatrix} \Bigg) \end{aligned} \tag{7}$$

$$\begin{aligned} \text{where}\quad & \Delta u_k = u_k - u_{k-1},\; \Delta v_k = v_k - v_{k-1}, \\ & Q = \begin{bmatrix} q_c & 0 \\ 0 & q_l \end{bmatrix},\; q_c, q_l, q_\theta > 0,\; R \in \mathbb{R}^{(n_u+1)^2}, \end{aligned} \tag{8}$$

and $R$ is positive definite. The penalty weights $q_c$, $q_\theta$, $R$ are tuning parameters to be decided based on the relative importance of contouring accuracy, path speed, and control deviations, and $q_l$ is chosen to be large so that $\hat{\epsilon}^l(\xi_{k+i}, \theta_{k+i}) \approx 0$.

This leads to the following optimisation problem being posed:

$$\text{Minimise}\quad J_k, \tag{9}$$

$$\text{Subject to}\quad \left.\begin{aligned} & \xi_{k+i} = A\xi_{k+i-1} + Bu_{k+i-1} \\ & \theta_{k+i} = \theta_{k+i-1} + v_{k+i-1} \\ & u_{k+i-1} \in \mathscr{U},\; v_{k+i-1} \in [0, v_{max}] \\ & \xi_{k+i} \in \mathscr{X},\; \theta_{k+i} \in [\theta^s, 0],\; i = 1, \ldots, N \end{aligned}\right\} \tag{10}$$

The model predictive contouring controller is implemented by solving the optimisation (9)-(10) at each time step. The first element of the optimal control input $\mathbf{u}_k^*$ is applied to the plant, while the first element of the optimal path speed $\mathbf{v}_k^*$

is used to update $\theta_k$. The optimisation is then repeated in a receding horizon fashion.

The cost function (7) can easily be modified to incorporate other control objectives, such as the minimisation of control effort and jerk. Similarly, additional explicit constraints specific to certain applications can also be included.

In general, solving the optimisation (9)-(10) in real time is computationally difficult. Even though the system (1) is linear, the nonlinearity of the path function $(x_d(\theta), y_d(\theta))$ results in a nonlinear optimisation problem.

To reduce the computation time, a linear time-varying (LTV) approach is proposed which approximates the optimisation problem with a convex quadratic program (QP). Then, a contraction constraint is introduced, which under specified conditions guarantees closed loop stability in the presence of approximation errors introduced by the LTV implementation.

### *A. Linear time-varying implementation*

A number of LTV MPC schemes have been proposed in the literature (e.g. [10]), where a nonlinear plant model is linearised around one or more operating points at each time step. The LTV approach is applied to model predictive contouring control as follows.

*1) Approximate cost function:* Assume that $\hat{\mathbf{u}}_k^*$, $\hat{\mathbf{v}}_k^*$ are close approximations of the optimal predicted input trajectories, which are unknown at time $k$. Let $\hat{\xi}_k^* = \{\hat{\xi}_{k,k}^*, \ldots, \hat{\xi}_{k+N-1,k}^*\}$ and $\hat{\Theta}_k^* = \{\hat{\theta}_{k,k}^*, \ldots, \hat{\theta}_{k+N-1,k}^*\}$ denote the state trajectories obtained by applying $\hat{\mathbf{u}}_k^*$ to (1) and $\hat{\mathbf{v}}_k^*$ to (4) with initial conditions $\hat{\xi}_{k,k}^* = \xi_k$ and $\hat{\theta}_{k,k}^* = \theta_k$.

The contouring error and lag can be approximated by linear functions using a Taylor series expansion and neglecting higher order terms

$$\hat{\epsilon}^{a,c}_{k+i,k} = \hat{\epsilon}^c(\hat{\xi}^*_{k+i,k}, \hat{\theta}^*_{k+i,k}) + \nabla\hat{\epsilon}^c(\hat{\xi}^*_{k+i,k}, \hat{\theta}^*_{k+i,k}) \begin{bmatrix} \xi_{k+i} \\ \theta_{k+i} \end{bmatrix}, \quad (11)$$

$$\hat{\epsilon}^{a,l}_{k+i,k} = \hat{\epsilon}^l(\hat{\xi}^*_{k+i,k}, \hat{\theta}^*_{k+i,k}) + \nabla\hat{\epsilon}^l(\hat{\xi}^*_{k+i,k}, \hat{\theta}^*_{k+i,k}) \begin{bmatrix} \xi_{k+i} \\ \theta_{k+i} \end{bmatrix}. \quad (12)$$

The linearised predictions $\hat{\epsilon}^{a,c}$ and $\hat{\epsilon}^{a,l}$ are used to approximate the cost function. The estimated cost is

$$\begin{aligned} J_k^a = &\sum_{i=1}^{N} \Big( \begin{bmatrix} \hat{\epsilon}^{a,c}_{k+i,k} & \hat{\epsilon}^{a,l}_{k+i,k} \end{bmatrix} Q \begin{bmatrix} \hat{\epsilon}^{a,c}_{k+i,k} & \hat{\epsilon}^{a,l}_{k+i,k} \end{bmatrix}^T - q_\theta \theta_{k+i} \\ &+ \begin{bmatrix} \Delta u^T_{k+i} & \Delta v_{k+i} \end{bmatrix} R \begin{bmatrix} \Delta u^T_{k+i} & \Delta v_{k+i} \end{bmatrix}^T \Big). \end{aligned} \quad (13)$$

The LTV model predictive contouring controller is implemented by solving the following optimisation problem

$$\text{Minimise } J_k^a, \text{ Subject to (10).} \quad (14)$$

The optimisation (14) can be formulated as a convex quadratic program (QP), for which there exist efficient solution methods. A method for finding the approximated input trajectories $\hat{\mathbf{u}}_k^*$, $\hat{\mathbf{v}}_k^*$ is proposed in the following.

*2) Trajectory approximation:* Let $\mathbf{u}_{k-1}^*$ and $\mathbf{v}_{k-1}^*$ denote the optimal input trajectory obtained by solving the optimisation (14) at time $k-1$. Then the approximations $\hat{\mathbf{u}}_k^*$, $\hat{\mathbf{v}}_k^*$ can be obtained by truncating $\mathbf{u}_{k-1}^*$, $\mathbf{v}_{k-1}^*$ and appending feasible inputs $\hat{u}^*_{k+N-1}$, $\hat{v}^*_{k+N-1}$ (typically $\hat{u}^*_{k+N-1}, \hat{v}^*_{k+N-1} = 0$), so that $\hat{\mathbf{u}}_k^* = \{u^*_{k,k-1}, ..., u^*_{k+N-2,k-1}, \hat{u}^*_{k+N-1}\}$, $\hat{\mathbf{v}}_k^* = \{v^*_{k,k-1}, ..., v^*_{k+N-2,k-1}, \hat{v}^*_{k+N-1}\}$.

At the initial time step $k = 0$, $\hat{\mathbf{u}}_0^*$, $\hat{\mathbf{v}}_0^*$ can be computed via the following iterative procedure.

*Procedure 1:*

1) Initialise $\hat{\mathbf{u}}_0^{*j}$, $\hat{\mathbf{v}}_0^{*j}$ to $\hat{\mathbf{u}}_0^{*0}$, $\hat{\mathbf{v}}_0^{*0}$ (typically $\hat{\mathbf{u}}_{\mathbf{0}}^{*0}, \hat{\mathbf{v}}_0^{*0} = \mathbf{0}$).
2) Compute the LTV model (11)-(12) based on $\hat{\mathbf{u}}_0^{*j}$, $\hat{\mathbf{v}}_0^{*j}$
3) Solve the optimisation (14) to obtain $\mathbf{u}_0^{*j}$ and $\mathbf{v}_0^{*j}$
4) Set $\hat{\mathbf{u}}_0^{*j} = \mathbf{u}_0^{*j}$, $\hat{\mathbf{v}}_0^{j} = \mathbf{v}_0^{*j}$ and increment $j$.
5) Repeat steps 2-4 until $||\mathbf{u}_0^{*j} - \mathbf{u}_0^{*j-1}|| \leq \epsilon_u$ and $||\mathbf{v}_0^{*j} - \mathbf{v}_0^{*j-1}|| \leq \epsilon_v$, for some $\epsilon_u, \epsilon_v > 0$.

Procedure 1 is essentially a Newton-like approach to solving the nonlinear optimisation (9)-(10). General techniques for showing convergence of such approaches are described in [11]. Since the optimal trajectories are not expected to change much from one time step to another, $\hat{\mathbf{u}}_k^*$ and $\hat{\mathbf{v}}_k^*$ are good approximations of the (unknown) optimal trajectories $\mathbf{u}_k^*$ and $\mathbf{v}_k^*$ which can be used to calculate a linear time-varying approximation to the cost function.

### *B. Stability analysis*

In this section, a modified version of LTV model predictive contouring control is proposed, and conditions under which the modified approach ensures closed loop stability are developed.

To investigate the stability of the proposed control scheme, an augmented nonlinear system is formed, making use of the following assumption.

*Assumption 1:* There exists a continuously differentiable and bounded function $\eta_d : [\theta^s, 0] \to \mathbb{R}^{n-2}$ such that for all $\theta$, $(x_d(\theta), y_d(\theta), \eta_d(\theta)) \in \mathscr{X}$ and $(x_d(0), y_d(0), \eta_d(0))$ is an equilibrium point of (1).

Consider the following change of co-ordinates:

$$\tilde{\xi}_k = \xi_k - \xi_d(\theta_k), \;\; \xi_d(\theta) = \begin{bmatrix} x_d(\theta) & y_d(\theta) & \eta_d(\theta) \end{bmatrix}^T. \quad (15)$$

Then the following augmented nonlinear system can be defined

$$\begin{bmatrix} \tilde{\xi}_{k+1} & \theta_{k+1} \end{bmatrix}^T = \tilde{f}(\tilde{\xi}_k, \theta_k, u_k, v_k) \quad (16)$$

where $\tilde{f}(\tilde{\xi}, \theta, u, v)$ is defined from (1), (4) and $\xi_d(\theta)$. By Assumption 1, the origin of (16) is an equilibrium point. In the following, the aim is to enforce stability of the origin of (16).

In [8], the terminal set/penalty approach is used to guarantee closed loop stability of the system under model predictive path-following control. A terminal penalty of $E(\theta) = \frac{\epsilon}{2}\theta^2$ is used, and it is shown that closed loop stability is guaranteed if $\epsilon$ is chosen to be sufficiently large. However, in the context of model predictive contouring control, this choice of terminal penalty can have unwanted effects on the behaviour of the controller, since it results in an increased weighting on the path speed. In addition, due to the model/plant mismatch introduced by the linear time-varying implementation, this terminal penalty does not guarantee stability of the system under LTV model predictive contouring control.

Because of these issues, an alternative stabilising technique is employed for model predictive contouring control. The chosen technique is contractive MPC [12], where stability is enforced by introducing a contractive constraint. Contrary to other MPC stability approaches, the cost function is not employed as a Lyapunov function. Contractive MPC does not involve any additional terms in the cost function, and has been used to guarantee closed loop stability of linear time varying MPC [13], making it an attractive approach for this application.

When contractive MPC is employed the contraction constraint remains constant for groups of $N$ time steps [12]. It is therefore useful to express the time step $k$ in terms of indices $l$, $m$ and the horizon length $N$ as follows:

$$k = lN + m \tag{17}$$

where $l \in \{0, 1, ..., \infty\}$ and $m \in \{0, 1, ..., N-1\}$. Table I shows how $l$ and $m$ evolve with $k$.

TABLE I

VALUES OF $l$ AND $m$ FOR INCREASING $k$

| $k$ | 0 | 1 | ... | $N-1$ | $N$ | $N+1$ | ... |
|---|---|---|---|---|---|---|---|
| $l$ | 0 | 0 | ... | 0 | 1 | 1 | ... |
| $m$ | 0 | 1 | ... | $N-1$ | 0 | 1 | ... |

In contractive MPC, a norm of the state is constrained to decrease in order to enforce stability. The following norm is defined for $(\tilde{\xi}, \theta)$

$$||\tilde{\xi}, \theta||_V = \tilde{\xi}^T P \tilde{\xi} + p|\theta| \tag{18}$$

where $P \in \mathbb{R}^{n^2}$ is positive definite and $p > 0$. The contraction constraint is then defined in terms of $||\tilde{\xi}, \theta||_V$:

$$\begin{aligned} &||\tilde{\xi}_{(l+1)N}, \theta_{(l+1)N}||_V \\ &\qquad \le \begin{cases} \alpha||\tilde{\xi}_{lN}, \theta_{lN}||_V, & ||\tilde{\xi}_{lN}, \theta_{lN}||_V \le \tilde{\alpha}, \\ ||\tilde{\xi}_{lN}, \theta_{lN}||_V - \tilde{\alpha}, & ||\tilde{\xi}_{lN}, \theta_{lN}||_V > \tilde{\alpha}, \end{cases} \\ &\quad \alpha \in (0,1),\ \tilde{\alpha} > 0. \end{aligned} \tag{19}$$

The contraction constraint (19) depends on the value of the states at time $lN$. If $(\tilde{\xi}_{lN}, \theta_{lN}) \in B_{\tilde{\alpha}}$, then an exponential rate of decay is imposed. If the states are outside $B_{\tilde{\alpha}}$ then $||\tilde{\xi}, \theta||_V$ is constrained to decrease linearly. This type of contraction constraint is chosen to best reflect the desired behaviour: traverse the path linearly until the end is approached, and then converge exponentially to the end point. By choosing the constraint in this way, the effect of the contraction constraint on the behaviour of the controller is minimised while guaranteeing closed loop stability.

In order to retain the convex structure of the optimisation, the contraction constraint must be approximated using a linear time varying approach. The LTV optimisation problem becomes

$$\begin{aligned} &\text{Minimise } J^a_{lN+m} \\ &\text{Subject to (10),} \\ &||\tilde{\xi}^a_{(l+1)N,lN}, \theta_{(l+1)N}||_V \\ &\qquad \le \begin{cases} \alpha||\tilde{\xi}_{lN}, \theta_{lN}||_V, & ||\tilde{\xi}_{lN}, \theta_{lN}||_V \le \tilde{\alpha}, \\ ||\tilde{\xi}_{lN}, \theta_{lN}||_V - \tilde{\alpha}, & ||\tilde{\xi}_{lN}, \theta_{lN}||_V > \tilde{\alpha}, \end{cases} \\ &\text{where } \tilde{\xi}^a_{(l+1)N,lN} = \xi_{(l+1)N} - \xi^a_{d,(l+1)N,lN}\big(\theta_{(l+1)N}\big) \end{aligned} \tag{20}$$

and $\xi^a_{d,(l+1)N,lN}(\theta)$ is a Taylor series approximation of $\xi_d(\theta)$ around $\hat{\Theta}^*_{lN}$, neglecting higher order terms:

$$\begin{aligned} \xi^a_{d,(l+1)N,lN}(\theta) &= \xi_d(\hat{\theta}^*_{(l+1)N,lN}) \\ &\quad + \nabla\xi_d(\hat{\theta}^*_{(l+1)N,lN})(\theta - \hat{\theta}^*_{(l+1)N,lN}). \end{aligned} \tag{21}$$

Following [13], the contraction constraint remains constant over $N$ time steps, so the linearisation (21) for the contraction constraint occurs every $N$ time steps, in contrast to the linearisation of the cost function (11)-(12) which occurs at every time step. The overall contractive LTV model predictive contouring control algorithm is as follows:

*Control Algorithm 1:*

1) Set $l = 0$, $m = 0$ and calculate $\hat{\mathbf{u}}^*_0$, $\hat{\mathbf{v}}^*_0$ using Procedure 1
2) Linearise the prediction of $||\tilde{\xi}_{(l+1)N}, \theta_{(l+1)N}||_V$ using (21) and $\hat{\mathbf{u}}^*_{lN}$, $\hat{\mathbf{v}}^*_{lN}$
3) Linearise the predictions of $\hat{\epsilon}^c$ and $\hat{\epsilon}^l$ using (11)-(12) and $\hat{\mathbf{u}}^*_{lN+m}$, $\hat{\mathbf{v}}^*_{lN+m}$
4) Solve the constrained optimisation (20) to obtain $\mathbf{u}^*_{lN+m}$ and $\mathbf{v}^*_{lN+m}$
5) Apply the first element of $\mathbf{u}^*_{lN+m}$ to the plant and update $\theta_k$ with the first element of $\mathbf{v}^*_{lN+m}$
6) Calculate $\hat{\mathbf{u}}^*_{lN+m+1}$, $\hat{\mathbf{v}}^*_{lN+m+1}$ from $\mathbf{u}^*_{lN+m}$, $\mathbf{v}^*_{lN+m}$
7) If $m = N-1$, set $m = 0$, increment $l$ and return to Step 2. Otherwise, increment $m$ and return to Step 3.

The addition of the contraction constraint changes the structure of the problem to a convex quadratically constrained quadratic program (QCQP), which can be solved with conventional algorithms.

In the remainder, sufficient conditions for stability of the origin of (16) under Control Algorithm 1 will be established. The approximation of $||\tilde{\xi}_{(l+1)N}, \theta_{(l+1)N}||_V$ introduces a model/plant mismatch which can affect closed loop stability of the system. The following assumptions are required:

*Assumption 2:* There exists a constant $\beta > 0$ such that for all $l \in \{0, ..., \infty\}, m \in \{0, ..., N-1\}$ $||\tilde{\xi}_{lN+m}, \theta_{lN+m}||_V \le \beta||\tilde{\xi}_{lN}, \theta_{lN}||_V$.

*Remark 1:* Assumption 2 imposes a bound on the transient states between times $k = lN$ and $k = (l+1)N$. Since the system inputs are constrained and the path functions are bounded, Assumption 2 is always satisfied.

*Assumption 3:* The optimisation problem (20) is feasible for all $k$ from the initial state.

*Remark 2:* Unfortunately, satisfaction of Assumption 3 cannot be guaranteed *a priori*. A methodology for choosing $P$ and $p$ such that Assumption 3 is most likely satisfied is discussed in Remark 5.

*Assumption 4:* The function $\xi_d(\theta)$ is twice continuously differentiable and there exists $G \in \mathbb{R}$ such that $||{}^1\!/_2 \nabla^2 \xi_d(\theta)|| \leq G$, for all $\theta \in (\theta^s, 0)$, where $||.||$ is the Euclidean norm.

First, a bound on the difference between the approximated state prediction $\tilde{\xi}^a_{(l+1)N,lN}$ and the true state $\tilde{\xi}_{(l+1)N}$ is established. Then this bound will be used to develop conditions on $\alpha$ and $\tilde{\alpha}$ that will guarantee stability of the origin of the nonlinear system (16).

*Lemma 1:* Let Assumption 4 hold. Then there exists $\lambda_1, \lambda_2 > 0$ such that for all $l$

$$||\tilde{\xi}_{(l+1)N} - \tilde{\xi}^a_{(l+1)N,lN}, 0||_V \leq \min\{\lambda_1 ||\tilde{\xi}_{lN}, \theta_{lN}||_V, \lambda_2\} \quad (22)$$

*Proof:* By the Mean Value Theorem and Assumption 4,

$$\begin{aligned} &||\tilde{\xi}_{(l+1)N} - \tilde{\xi}^a_{(l+1)N,lN}|| \\ &\quad = ||\xi_d(\theta_{(l+1)N}) - \xi^a_{d,(l+1)N,lN}(\theta_{(l+1)N})|| \\ &\quad \leq G|\theta_{(l+1)N} - \hat{\theta}^*_{(l+1)N,lN}|^2. \end{aligned} \quad (23)$$

Since $\hat{\theta}^*_{lN,lN} = \theta_{lN}$, $\theta \in [\theta^s, 0]$ and $v \in [0, v_{max}]$,

$$|\theta_{(l+1)N} - \hat{\theta}^*_{(l+1)N,lN}|^2 \leq N v_{max} \min\{N v_{max}, |\theta_{lN}|\}. \quad (24)$$

Combining (23) and (24),

$$\begin{aligned} &||\tilde{\xi}(\theta_{(l+1)N}) - \tilde{\xi}^a_{(l+1)N,lN}(\theta_{(l+1)N}), 0||_V \\ &\quad \leq \sqrt{\sigma_{max}\{P\}}\, G N v_{max} \min\{N v_{max}, |\theta_{lN}|\} \end{aligned} \quad (25)$$

where $\sigma_{max}\{P\}$ denotes the maximum eigenvalue of $P$. Setting $\lambda_1 = {}^1\!/_p \sqrt{\sigma_{max}\{P\}} G N v_{max}$ and $\lambda_2 = \sqrt{\sigma_{max}\{P\}} G N^2 v^2_{max}$ concludes the proof. ■

Making use of Lemma 1, the following theorem establishes conditions under which Control Algorithm 1 guarantees asymptotic stability of (16).

*Theorem 1:* Let Assumptions 1-4 hold. If $\alpha, \tilde{\alpha}$ from (20) and $P, p$ from (18) satisfy

1) $\alpha - {}^1\!/_p \sqrt{\sigma_{max}\{P\}}\, G N v_{max} < 1, \quad \alpha > 0,$
2) $\tilde{\alpha} > \sqrt{\sigma_{max}\{P\}} G N^2 v^2_{max},$

then the origin of (16) under Control Algorithm 1 is asymptotically stable. Moreover, states starting in $B_{\tilde{\alpha}} := \{\xi, \theta : ||\xi, \theta||_V \leq \tilde{\alpha}\}$ will converge to the origin exponentially.

*Proof:* By the triangle inequality,

$$\begin{aligned} ||\tilde{\xi}_{(l+1)N}, \theta_{(l+1)N}||_V \leq\ & ||\tilde{\xi}_{(l+1)N} - \tilde{\xi}^a_{(l+1)N}, 0||_V \\ & + ||\tilde{\xi}^a_{(l+1)N,lN}, \theta_{(l+1)N}||_V \end{aligned} \quad (26)$$

The 0 appears because there is no linearisation error in predicting $\theta_{(l+1)N}$. In the remainder of the proof, two separate cases are considered.

**Case 1:** $||\tilde{\xi}_0, \theta_0||_V > \tilde{\alpha}$. From Lemma 1 and the contraction constraint,

$$\begin{aligned} ||\tilde{\xi}_{(l+1)N}, \theta_{(l+1)N}||_V &\leq \lambda_2 + ||\tilde{\xi}_{lN}, \theta_{LN}||_V - \tilde{\alpha}, \\ ||\tilde{\xi}_{lN}, \theta_{lN}||_V &> \tilde{\alpha}. \end{aligned} \quad (27)$$

Clearly, if

$$\tilde{\alpha} - \lambda_2 > 0, \quad (28)$$

the following hold:

1) $||\tilde{\xi}_{lN}, \theta_{lN}||_V \leq ||\tilde{\xi}_0, \theta_0||_V$.
2) There exists $l_f > 0$ such that $||\tilde{\xi}_{l_f N}, \theta_{l_f N}||_V \leq \tilde{\alpha}$,

From Assumption 2,

$$||\tilde{\xi}_{lN+m}, \theta_{lN+m}||_V \leq \beta ||\tilde{\xi}_0, \theta_0||_V. \quad (29)$$

Therefore if (28) holds, $||\tilde{\xi}_k, \theta_k||_V$ remains bounded for all $k$ and converges to $B_{\tilde{\alpha}}$.

**Case 2:** $||\tilde{\xi}_0, \theta_0||_V \leq \tilde{\alpha}$. From Lemma 1 and the contraction constraint,

$$||\tilde{\xi}_{(l+1)N}, \theta_{(l+1)N}||_V \leq \lambda_1 ||\tilde{\xi}_{lN}, \theta_{LN}||_V + \alpha ||\tilde{\xi}_{lN}, \theta_{LN}||_V \quad (30)$$

It follows from (30) and Assumption 2 that

$$||\tilde{\xi}_{lN+m}, \theta_{lN+m}||_V \leq \beta(\alpha + \lambda_1)^l ||\tilde{\xi}_0, \theta_0||_V. \quad (31)$$

It can be shown that if [12]

$$\alpha + \lambda_1 < 1, \quad (32)$$

there exist $\gamma_1 \geq 0$, $\gamma_2 \in (0, 1)$ such that $||\tilde{\xi}_k, \theta_k||_V \leq \gamma_1 ||\tilde{\xi}_0, \theta_0||_V\ \gamma_2^k$ Therefore, if (32) holds the origin of the system (16) is locally exponentially stable from $B_{\tilde{\alpha}}$. Combining the results from Case 1 and 2, it can be concluded that the origin of the system (16) is asymptotically stable if (28) and (32) hold. ■

*Remark 3:* It follows from Theorem 1 that a necessary condition for stability to hold is that ${}^1\!/_p \sqrt{\sigma_{max}\{P\}}\, G N v_{max} < 1$. This can always be satisfied by setting $p$ sufficiently large relative to $\sigma_{max}\{P\}$.

*Remark 4:* If the true contraction constraint (19) is used instead of the approximation, it follows from Theorem 1 that any $\alpha \in (0, 1)$ and $\tilde{\alpha} > 0$ will guarantee closed loop stability of the system. The effect of the linear time-varying approximation is that the contraction constraint must become more conservative to overcome the linearisation error.

*Remark 5:* As mentioned in Remark 2, satisfaction of Assumption 3 cannot be guaranteed *a priori*. However, examining the system constraints, it can be seen that choosing $p$ large relative to $\sigma_{max}\{P\}$ minimises the risk of feasibility issues.

## IV. SIMULATION RESULTS

To demonstrate the effectiveness of the proposed contouring controller, the algorithm was simulated for an XY table system. Each axis of the XY table is modelled as two rotational inertias connected by a flexible coupling. The XY table model is as follows

$$\begin{aligned} &\xi = \begin{bmatrix} x & y & \psi_x & \dot{\psi}_x & \dot{\varphi}_x & \psi_y & \dot{\psi}_y & \dot{\varphi}_y \end{bmatrix}^T, u = \begin{bmatrix} i_x & i_y \end{bmatrix}^T \\ &x_k = \tau \varphi_{x,k}, \ \ y_k = \tau \varphi_{y,k}, \\ &\varphi_{j,k+1} = \varphi_{j,k} + T_s \dot{\varphi}_{j,k}, \ \ \psi_{j,k+1} = \psi_{j,k} + T_s \dot{\psi}_{j,k} \\ &\dot{\varphi}_{j,k+1} = \dot{\varphi}_{j,k} + {}^{T_s}\!/_{J_l} (k(\phi_{j,k} - \varphi_{j,k}) + c(\dot{\psi}_{j,k} - \dot{\varphi}_{j,k}) \\ &\qquad - b_l \dot{\varphi}_{j,k}), \\ &\dot{\psi}_{j,k+1} = \dot{\psi}_{j,k} + {}^{T_s}\!/_{J_m} (K_t\, i_{j,k} + k(\varphi_{j,k} - \psi_{j,k}) \\ &\qquad + c(\dot{\varphi}_{j,k} - \dot{\psi}_{j,k}) - b_m \dot{\psi}_{j,k}), \end{aligned} \quad (33)$$

where $j = \{x, y\}$ and $\psi_j$, $\varphi_j$ represent the angular displacements of the motor and load respectively in each axis.

The simulation parameters are summarised in Table II. The system inputs are the servo motor currents $i_x$, $i_y$ in Amperes which are subject to the constraints $-1 \leq i_x, i_y \leq 1$.

TABLE II

SIMULATION PARAMETERS

| Parameter | Unit | Value |
|---|---|---|
| Motor inertia $J_m$ | kg m$^2$ | $1.9 \times 10^{-5}$ |
| Load inertia $J_l$ | kg m$^2$ | $1.93 \times 10^{-5}$ |
| Motor viscous friction $b_m$ | Nm s/rad | $1.37 \times 10^{-4}$ |
| Load viscous friction $b_l$ | Nm s/rad | $1.44 \times 10^{-5}$ |
| Motor torque constant $K_t$ | Nm/A | 0.5105 |
| Coupling stiffness $k$ | Nm/rad | 2.5450 |
| Coupling damping $c$ | Nm s/rad | 0.001 |
| Ball screw pitch $\tau$ | mm/rad | 0.7958 |
| Sample period $T_s$ | s | 0.001 |

The model predictive contouring controller was simulated for the XY table system using Control Algorithm 1 for the contour shown in Fig. 3. The chosen parameters were $N = 50$, $q_l = 10^3$, $R = diag\{5, 5, 2000\}$ and varying values of $q_c$ and $q_\theta$. The desired path is an arc length parameterised quintic spline generated using the method outlined in [9]. The average computation time per time step for the chosen horizon on a desktop PC with MATLAB was 0.19 seconds, indicating that further work is required before the controller can be implemented on a real machine.

The contraction constraint was chosen as $P = 10^{-5} I$ , $p = 100$, where $I$ is the identity matrix. For this simulation, it can be shown that choosing $\alpha = 0.997$, $\tilde{\alpha} = 4.138$ satisfies the requirements of Theorem 1. The contraction constraint was implemented at each time step by first solving the QP (14) without the contraction constraint, and subsequently solving the QCQP (20) only if the contraction constraint is violated. This approach reduced the additional computation time involved with incorporating the contractive constraint, as in most cases the constraint was satisfied already by solving (14). With the above parameters the QCQP solver was never required, indicating that the effect of the contraction constraint on the behaviour of the controller is minimal.

The contours for $q_c = 10$, $q_\theta = 10$ and $q_c = 1000$, $q_\theta = 0.1$ are shown in Fig. 3. It can be observed that when the contouring error penalty is lower, the path is traversed faster but at the cost of increased contouring error. Thus, by adjusting the penalty weights, the trade-off between productivity and contouring accuracy can be systematically addressed. Table III summarises the maximum contouring error and average path speed for each combination of weights.

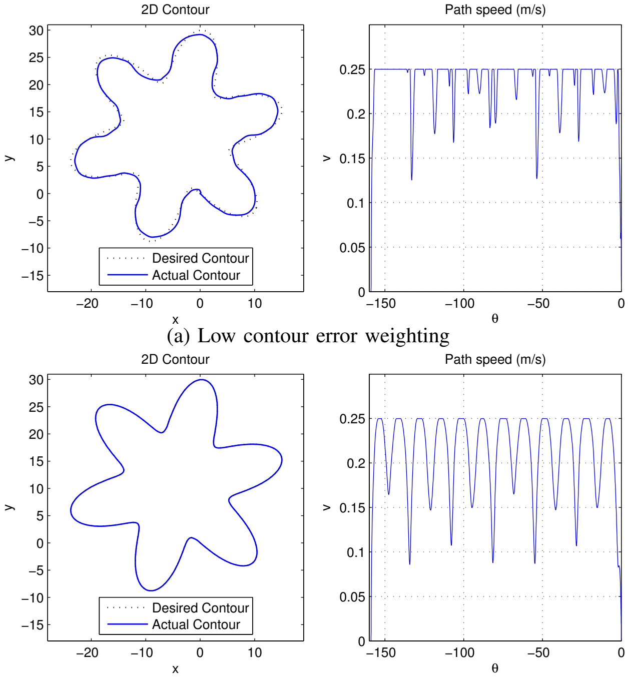
TABLE III

MAX. CONTOURING ERROR AND TRAVERSAL TIME

| Contour error weighting | Path speed weighting | Max. contouring error (mm) | Traversal time (s) |
|---|---|---|---|
| 10 | 10 | 0.932 | 0.759 |
| 1000 | 0.1 | 0.0074 | 0.832 |

## V. FURTHER WORK

Future work may include accounting for disturbances and modelling errors, and further reductions in computation time.



(a) Low contour error weighting

(b) High contour error weighting

Fig. 3. XY table simulation results

## VI. ACKNOWLEDGEMENT

The authors would like to thank ANCA Motion Pty Ltd for their ongoing support in this research.

## REFERENCES

[1] Y. Koren and C. C. Lo, “Advanced controllers for feed drives,” *CIRP Ann.*, vol. 41, no. 2, pp. 689–698, 1992.

[2] D. Renton and M. Elbestawi, “High speed servo control of multi-axis machine tools,” *Int. J. Mach. Tools Manuf*, vol. 40, pp. 539–559, 2000.

[3] J. Dong, P. Ferreira, and J. Stori, “Feed-rate optimization with jerk constraints for generating minimum-time trajectories,” *Int. J. Mach. Tools Manuf*, vol. 47, pp. 1941–1955, 2007.

[4] D. Verscheure, B. Demeulenaere, J. Swevers, J. D. Schutter, and M. Diehl, “Time-optimal path tracking for robots: A convex optimization approach,” *IEEE Trans. Autom. Control*, vol. 54, no. 10, pp. 2318–2327, 2009.

[5] F. Imamura and H. Kaufman, “Time optimal contour tracking for machine tool controllers,” *IEEE Control Syst. Mag.*, vol. 11, no. 3, pp. 11–17, 1991.

[6] D. Croft, S. Stilson, and S. Devasia, “Optimal tracking of piezo-based nanopositioners,” *Nanotechnology*, vol. 10, pp. 201–208, 1999.

[7] A. P. Aguiar, J. P. Hespanha, and P. V. Kokotivić, “Performance limitations in reference tracking and path following for nonlinear systems,” *Automatica*, vol. 4, no. 3, pp. 598–610, 2008.

[8] T. Faulwasser, B. Kern, and R. Findeisen, “Model predictive path-following for constrained nonlinear systems,” in *CDC*, 2009.

[9] K. Erkorkmaz and Y. Altintas, “Quintic spline interpolation with minimal feed fluctuation,” *J. Manuf. Sci. Eng.*, vol. 127, pp. 339–349, 2005.

[10] R. Sharma, D. Nesic, and C. Manzie, “Idle speed control using linear time varying model predictive control and discrete time approximations,” in *IEEE Multi-Conference on Systems and Control (MSC), Yokohama, Japan. (Submitted as an invited paper)*, 2010.

[11] B. T. Polyak, *Introduction to Optimization*. Optimization Software, 1987.

[12] S. L. De Oliveira Kothare and M. Morari, “Contractive model predictive control for constrained nonlinear systems,” *IEEE Trans. Autom. Control*, vol. 45, no. 6, pp. 1053–1071, 2000.

[13] S. L. De Oliveira, “Model predictive control (MPC) for constrained non-linear systems,” Ph.D. dissertation, California Institute of Technology, Pasadena, CA, 1996.