Application of Model Predictive Contouring Control to an X-Y Table

Denise Lam * Chris Manzie * Malcolm Good *

* *Department of Mechanical Engineering, The University of Melbourne, Australia*
{lamd,manziec,mcgood}@unimelb.edu.au

Abstract: Model predictive contouring control is a new control scheme based on minimisation of a cost function which reflects the trade-off between the competing objectives of accuracy and traversal time. In this paper, model predictive contouring control is implemented in real time on an X-Y table. The controller automatically adjusts the feed rate to maintain accuracy along the desired path. By varying the cost function weights, higher accuracy can be achieved by sacrificing productivity, or vice versa. Experimental results demonstrate that the new contouring control scheme achieves an improvement in performance compared to PI based and traditional MPC tracking controllers.

1. 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 traditional contouring systems, a path planning routine is used to convert the desired path to a time-dependent reference trajectory, which is then tracked using feedback controllers. 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 (Koren and Lo, 1992). 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 by Renton and Elbestawi (2000), and later Dong et al. (2007) with the addition of jerk constraints.

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. Imamura and Kaufman (1991) proposed optimising the reference trajectory and the tuning parameters of the feedback controller such that the tracking accuracy was below a specified tolerance, for a particular contour. Naturally, as the tolerance is increased, the time to complete the trajectory decreases. Erkorkmaz et al. (2006) used spline fitting techniques to smooth out sharp corners in the contouring path. The cornering feed rate was then tuned manually using a simulation model until a desired level of accuracy was achieved. These offline path planning approaches employ conventional feedback control to track the optimal reference trajectory.

In Verscheure et al. (2009), a dynamic model of the system is used to optimise the control inputs and path speed simultaneously for a robot manipulator. This approach is purely feedforward and does not compensate for modelling errors or disturbances. 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, and therefore has the capability to reject disturbances. However, path following approaches such as Aguiar et al. (2008) do not take constraints into account.

A path following control framework based on model predictive control (MPC) was proposed by Faulwasser et al. (2009). The model predictive path-following controller (MPFC) optimised 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 by Faulwasser et al. (2009), finding a real-time solution to the optimisation problem in contouring applications is difficult.

In previous work, model predictive contouring control (MPCC) was developed as an extension of the MPFC framework to contouring control (Lam et al., 2010). 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 (LTV) formulation was proposed to reduce

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

computational complexity. Simulations were conducted with a perfect plant model and no disturbances.

This paper extends the previous work to real-time implementation of model predictive contouring control (MPCC) on an X-Y table test rig. A velocity form state-space model is used to compensate for modelling errors and disturbances. Experimental results demonstrate how the controller automatically reduces the path velocity around tight curves. By varying the weights in the cost function, different levels of accuracy and productivity are achieved. A comparison between MPCC, cascaded PI and model predictive tracking control shows a significant improvement in accuracy with MPCC for similar traversal times.

2. TEST RIG DESCRIPTION

The X-Y table test rig is shown in Fig. 1. Each axis consists of a brushless AC servomotor, coupling, and precision linear table with lead screw. The motors are controlled using two servo drives which receive q-axis (torque producing) current commands $i_{c,x}$ and $i_{c,y}$ from the Target PC with a minimum sample period of 1 ms. The servo drives employ proportional-integral (PI) controllers to control the d-axis currents to zero and the q-axis currents i_x, i_y to the commands $i_{c,x}$, $i_{c,y}$ by applying the appropriate phase voltages to the motors. The current commands $i_{c,x}$, $i_{c,y}$ are subject to the constraints $|i_{c,x}|, |i_{c,y}| \leq 0.5\,\text{A}$. Position feedback is available via resolver shaft position sensors on the motors and linear encoders on the axes. The overall test rig architecture is shown in Fig. 2.

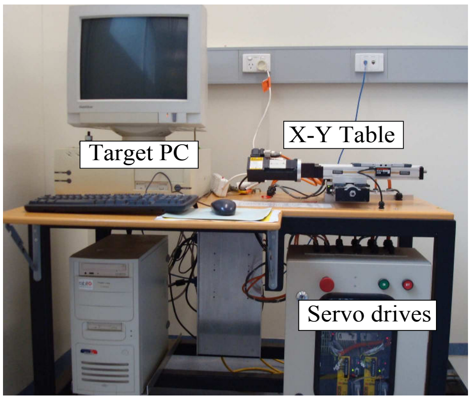

Fig. 1. X-Y table test rig

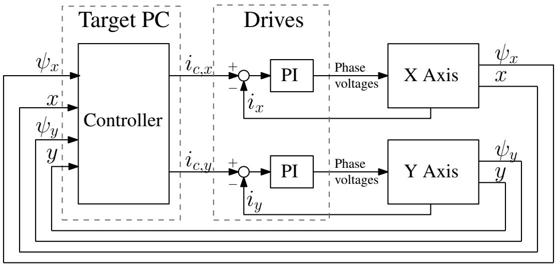

Fig. 2. X-Y table test rig architecture

3. X-Y TABLE MODEL

Since MPCC is a model-based control approach, a dynamic model of the X-Y table is required for the controller design. Each axis of the X-Y table is modelled as two rotational inertias connected by a flexible coupling, as shown in Fig. 3 for the X-axis. The continuous time equations of motion for the X-Y table are given in (1),

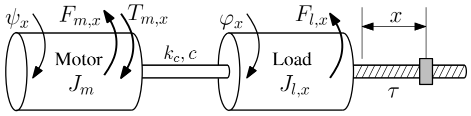

Fig. 3. X-axis two inertia model

$$
\begin{aligned}
\ddot{\psi}_x &= \frac{1}{J_m}\Big(T_{m,x}+k_c(\varphi_x-\psi_x)+c(\dot{\varphi}_x-\dot{\psi}_x)-F_{m,x}\Big),\\
\ddot{\varphi}_x &= \frac{1}{J_{l,x}}\Big(k_c(\psi_x-\varphi_x)+c(\dot{\psi}_x-\dot{\varphi}_x)-F_{l,x}\Big),\\
\ddot{\psi}_y &= \frac{1}{J_m}\Big(T_{m,y}+k_c(\varphi_y-\psi_y)+c(\dot{\varphi}_y-\dot{\psi}_y)-F_{m,y}\Big),\\
\ddot{\varphi}_y &= \frac{1}{J_{l,y}}\Big(k_c(\psi_y-\varphi_y)+c(\dot{\psi}_y-\dot{\varphi}_y)-F_{l,y}\Big),
\end{aligned}
\tag{1}
$$

where for each axis, T_m is the motor torque, F_m and F_l are the motor and load friction torques respectively, J_m and J_l are the motor and load inertias respectively, and ψ and φ are the angular positions of the motor and load respectively. Note that the motor and coupling characteristics are identical in both axes, while the load properties differ. The linear displacements x and y are related to the angular displacements by

$$x = \tau\varphi_x,\ y = \tau\varphi_y, \tag{2}$$

where τ is the lead screw pitch. The motor torque is modelled as a linear function of motor current, so that

$$T_{m,x} = K_t i_x,\ T_{m,y} = K_t i_y, \tag{3}$$

where K_t is the motor torque constant. The PI current controller is assumed to be sufficiently fast so that $i_x = i_{c,x}$ and $i_y = i_{c,y}$. This assumption is a potential source of modelling error not considered in Lam et al. (2010).

The friction torques are given by

$$
\begin{aligned}
F_{m,x} &= b_m\dot{\psi}_x,\ F_{l,x} = b_{l,x}\dot{\varphi}_x,\\
F_{m,y} &= b_m\dot{\psi}_y,\ F_{l,y} = b_{l,y}\dot{\varphi}_y,
\end{aligned}
\tag{4}
$$

where b_m, b_l are the co-efficients of viscous friction for the motor and load respectively. Since MPCC employs a linear model of the system, nonlinear friction is treated as a disturbance, and is compensated by augmenting the model with a constant output disturbance, as shown later in (6).

The continuous time equations (1) are discretised by applying a zero-order hold with sampling period h. The linear discrete time dynamic model can be expressed in state-space form

$$\xi_{k+1} = A\xi_k + Bu_k,\ \begin{bmatrix} x_k \\ y_k \end{bmatrix} = C\xi_k,\ u \in [u_{min}, u_{max}], \tag{5}$$

where $\xi = \begin{bmatrix}\psi_x & \dot{\psi}_x & \varphi_x & \dot{\varphi}_x & \psi_y & \dot{\psi}_y & \varphi_y & \dot{\varphi}_y\end{bmatrix}^T$, $u = [i_{c,x}\ i_{c,y}]^T$, u_{min}, u_{max} are the constraints on the command currents, and A, B, and C are the discrete time state matrices which can be obtained from (1)-(4) and h. The model parameters were identified using conventional system identification methods and are summarised in Fig. 4.

Implementation of model predictive contouring control requires the solution of a constrained optimisation at each time step, which requires some time to compute. As

Parameter	Unit	Values	
		x	y
Motor inertia J_m	$\mathrm{kg \cdot m^2}$	2.3×10^{-5}	
Table inertia J_l	$\mathrm{kg \cdot m^2}$	2.10×10^{-5}	2.33×10^{-5}
Motor viscous friction b_m	$\mathrm{Nm \cdot s/rad}$	1.39×10^{-4}	1.17×10^{-4}
Table viscous friction b_l	$\mathrm{Nm \cdot s/rad}$	0.001	7.25×10^{-4}
Motor torque constant K_t	$\mathrm{Nm/A}$	0.4364	
Coupling stiffness k_c	$\mathrm{Nm/rad}$	2.6682	2.3017
Coupling damping c	$\mathrm{Nm \cdot s/rad}$	5.0×10^{-4}	0.0011
Lead screw pitch τ	$\mathrm{mm/rad}$	0.7958	

Fig. 4. X-Y table model parameters

a result, a 1-sample computational delay is introduced, which must be accounted for in the prediction model in order to maintain performance. Therefore, the X-Y table model (5) is modified to include a unit delay in the input.

Furthermore, due to parameter error and unmodelled dynamics such as nonlinear friction, it is expected that the X-Y table model (5) will exhibit significant modelling errors. These errors, if not compensated, will cause deterioration in performance. Hence, it is necessary to modify (5) by incorporating a disturbance model. There are a variety of techniques available for incorporating disturbance modelling in MPC (Muske and Badgwell, 2002). The method chosen for the X-Y table is to convert (5) to a velocity form state-space model, where the new state and input represent the change in the original state and input, and augmenting the state vector with the system outputs x and y. The new model incorporating both the unit delay and disturbance model is expressed as

$$\begin{bmatrix} \Delta\xi_{k+1} \\ \Delta u_k \\ x_{k+1} \\ y_{k+1} \end{bmatrix} = \begin{bmatrix} A & B & 0 \\ 0 & 0 & 0 \\ CA & CB & I \end{bmatrix} \begin{bmatrix} \Delta\xi_k \\ \Delta u_{k-1} \\ x_k \\ y_k \end{bmatrix} + \begin{bmatrix} 0 \\ I \\ 0 \end{bmatrix} \Delta u_k,$$

$$\begin{bmatrix} x_k \\ y_k \end{bmatrix} = [0 \ 0 \ I] \begin{bmatrix} \Delta\xi_k \\ \Delta u_{k-1} \\ x_k \\ y_k \end{bmatrix}, \text{where} \begin{array}{l} \Delta\xi_k = \xi_k - \xi_{k-1}, \\ \Delta u_k = u_k - u_{k-1}, \end{array} \tag{6}$$

and I is the identity matrix.

The augmented model (6) is then used to compute predictions of x and y for use in the control algorithm. This has the effect of introducing a constant output disturbance model in the MPC formulation (Muske and Badgwell, 2002). Alternatively, the augmented model (6) can be viewed as adding "integral action" to the controller (Wang, 2009). In the controller implementation, the optimal control input u_k is not applied to the plant until time $k+1$, so that the latency in the test rig matches the unit delay in the model (6).

4. MPCC OF X-Y TABLE

It is proposed to implement model predictive contouring control (Lam et al., 2010) on an X-Y table test rig using the model (6). First, the contouring control task and objectives are identified. Then, the model predictive contouring control formulation is introduced. Finally, the real-time implementation of MPCC is discussed.

4.1 Control task and objectives

The control task is to steer the X-Y table along a continuous two-dimensional geometric path $(x_d(\theta), y_d(\theta))$, parameterised by a path parameter θ:

$$x^d : [\theta^s, 0] \to \mathbb{R}; \;\; y^d : [\theta^s, 0] \to \mathbb{R}; \;\; \theta^s < 0, \tag{7}$$

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.

The contouring error ϵ_k^c is defined as the normal deviation from the desired path (Koren and Lo, 1992), and can be expressed as

$$\epsilon_k^c = \sin\phi(\theta^r)\left(x_k - x^d(\theta^r)\right) - \cos\phi(\theta^r)\left(y_k - y^d(\theta^r)\right),$$
$$\phi(\theta^r) = \arctan\left(\frac{\nabla y^d(\theta^r)}{\nabla x^d(\theta^r)}\right), \tag{8}$$

where $\nabla(.) = \frac{d(.)}{d\theta}$, and $\theta^r(x,y)$ 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. 5.

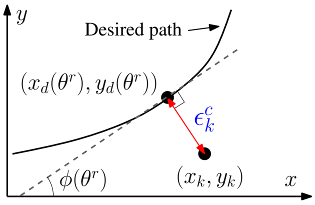

Fig. 5. Contouring error

4.2 Cost function formulation

Following Lam et al. (2010), the MPCC cost function is derived as follows. The following dynamics are introduced for the path parameter θ:

$$\theta_{k+1} = \theta_k + v_k, \; v_k \in [0, v_{max}], \; v_{max} > 0, \; \theta_k \in [\theta^s, 0] \tag{9}$$

where v_k is a virtual input to be determined by the controller and θ_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 $v_k \geq 0$.

The path parameter θ_k, whose evolution is governed by (9), is used as an approximation to $\theta^r(x_k, y_k)$.

Assumption 1. θ_k is sufficiently close to $\theta^r(x_k, y_k)$.

Remark 1. Let ϵ^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 ϵ^l as

$$\hat{\epsilon}^l(\xi_k, \theta_k) = -\cos\phi(\theta_k)\left(x_k - x^d(\theta_k)\right) - \sin\phi(\theta_k)\left(y_k - y^d(\theta_k)\right) \tag{10}$$

Refer to Fig. 6 for a graphical interpretation of ϵ_c, ϵ_l and their approximations. It can be observed that for most paths, $\theta_k \to \theta^r(x_k, y_k)$ as $\hat{\epsilon}^l(\xi_k, \theta_k) \to 0$. Therefore for practical purposes, Assumption 1 can be enforced by including an appropriate penalty on $\hat{\epsilon}^l$ in the cost function (12).

By Assumption 1, the contouring error can be approximated by

$$\hat{\epsilon}^c(\xi_k, \theta_k) = \sin\phi(\theta_k)\left(x_k - x^d(\theta_k)\right) - \cos\phi(\theta_k)\left(y_k - y^d(\theta_k)\right), \tag{11}$$

and θ_k can be used as an approximation of how far along the path the system has travelled.

The cost function J_k represents the trade-off between contouring accuracy and path speed over a horizon of N time steps, as well as penalising control input deviations:

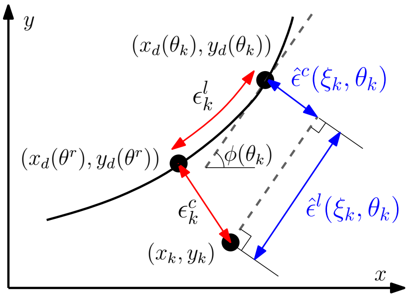
y
x
$(x_d(\theta_k), y_d(\theta_k))$
$\hat{\epsilon}^c(\xi_k, \theta_k)$
ϵ^l_k
$(x_d(\theta^r), y_d(\theta^r))$
$\phi(\theta_k)$
ϵ^c_k
$\hat{\epsilon}^l(\xi_k, \theta_k)$
(x_k, y_k)

Fig. 6. Contouring error ϵ^c, lag ϵ^l and their approximations $\hat{\epsilon}^c$ and $\hat{\epsilon}^l$

$$J_k = \sum_{i=1}^{N} \left(\begin{bmatrix} \hat{\epsilon}^c(\xi_{k+i}, \theta_{k+i}) \\ \hat{\epsilon}^l(\xi_{k+i}, \theta_{k+i}) \end{bmatrix}^T Q \begin{bmatrix} \hat{\epsilon}^c(\xi_{k+i}, \theta_{k+i}) \\ \hat{\epsilon}^l(\xi_{k+i}, \theta_{k+i}) \end{bmatrix} - q_\theta \theta_{k+i} + \begin{bmatrix} \Delta u_{k+i} \\ \Delta v_{k+i} \end{bmatrix}^T R \begin{bmatrix} \Delta u_{k+i} \\ \Delta v_{k+i} \end{bmatrix} \right), \tag{12}$$

where $\Delta v_k = v_k - v_{k-1}$, $Q = \begin{bmatrix} q_c & 0 \\ 0 & q_l \end{bmatrix}$, $q_c, q_l, q_\theta > 0, R \in \mathbb{R}^3$, and R is positive definite. The penalty weights q_c, q_θ, 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 sufficiently large to satisfy Assumption 1.

4.3 QP approximation

In general, minimisation of the cost function (12) is computationally difficult. To reduce the computation time, a linear time-varying (LTV) approach was proposed by Lam et al. (2010) which approximates the optimisation problem with a convex quadratic program (QP). A similar LTV approximation is employed in this paper to facilitate real-time implementation on the X-Y table, and is described below.

Assume that $\hat{\Theta}^*_k = \{\hat{\theta}^*_{k,k}, \ldots, \hat{\theta}^*_{k+N-1,k}\}$ is a close approximation of the optimal predicted path state trajectory, which is unknown at time k. The desired path functions $x_d(\theta), y_d(\theta)$ are linearised across the prediction horizon using a Taylor series expansion around $\hat{\Theta}^*_k$ and neglecting higher order terms

$$\begin{aligned} x^{a,d}_{k+i,k}(\theta) &= x^d(\hat{\theta}^*_{k+i,k}) + \nabla x^d(\hat{\theta}^*_{k+i,k})(\theta - \hat{\theta}^*_{k+i,k}), \\ y^{a,d}_{k+i,k}(\theta) &= y^d(\hat{\theta}^*_{k+i,k}) + \nabla y^d(\hat{\theta}^*_{k+i,k})(\theta - \hat{\theta}^*_{k+i,k}). \end{aligned} \tag{13}$$

Approximations for the contouring error and $\hat{\epsilon}^l$ can then be formed using (13),

$$\begin{aligned} \hat{\epsilon}^{a,c}_{k+i,k}(\xi_{k+i}, \theta_{k+i}) &= \sin\phi(\hat{\theta}^*_{k+i,k})(x_{k+i} - x^{a,d}k+i,k(\theta_{k+i})) \\ &\quad - \cos\phi(\hat{\theta}^*_{k+i,k})(y_{k+i} - y^{a,d}_{k+i,k}(\theta_{k+i})), \\ \hat{\epsilon}^{a,l}_{k+i,k}(\xi_{k+i}, \theta_{k+i}) &= -\cos\phi(\hat{\theta}^*_{k+i,k})(x_{k+i} - x^{a,d}_{k+i,k}(\theta_{k+i})) \\ &\quad - \sin\phi(\hat{\theta}^*_{k+i,k})(y_{k+i} - y^{a,d}_{k+i,k}(\theta_{k+i})). \end{aligned} \tag{14}$$

The approximated cost function is formed from $\hat{\epsilon}^{a,c}$ and $\hat{\epsilon}^{a,l}$ as follows:

$$J^a_k = \sum_{i=1}^{N} \Big(\left[\hat{\epsilon}^{a,c}_{k+i,k} \ \hat{\epsilon}^{a,l}_{k+i,k}\right] Q \left[\hat{\epsilon}^{a,c}_{k+i,k} \ \hat{\epsilon}^{a,l}_{k+i,k}\right]^T - q_\theta \theta_{k+i} + \left[\Delta u^T_{k+i} \ \Delta v_{k+i}\right] R \left[\Delta u^T_{k+i} \ \Delta v_{k+i}\right]^T \Big). \tag{15}$$

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

$$\begin{aligned} &\text{Minimise} \quad J^a_k \\ &\text{Subject to (6),} \\ &\qquad \theta_{k+i} = \theta_{k+i-1} + v_{k+i-1}, \\ &\qquad u_{k+i-1} \in [u_{min}, u_{max}], \\ &\qquad v_{k+i-1} \in [0, v_{max}], \\ &\qquad \theta_{k+i} \in [\theta^s, 0], \ i = 1, ..., N. \end{aligned} \tag{16}$$

The optimisation (16) can be formulated as a convex quadratic program (QP) of the following form

$$\begin{aligned} &\text{Minimise} \ \frac{1}{2} z^T_k \mathcal{H}_k z_k + z^T_k \mathcal{G}_k, \\ &\text{Subject to} \ E_k z_k \leq F_k, \end{aligned} \tag{17}$$

where $z_k = [v_k, \ \ldots \ v_{k+N-1}, \ \Delta u_k, \ \ldots \ \Delta u_{k+N-1}]^T$, $\mathcal{H}_k$ and $\mathcal{G}_k$ are computed by combining the approximate cost function (15) with the plant model (6), and E_k and F_k are derived from the inequality constraints of (16). Refer to Maciejowski (2002) for a detailed description. The quadratic program (17) can be solved using conventional optimisation techniques.

Calculation of the linearised path functions (13) requires an estimate of the path state trajectory $\hat{\Theta}^*_k$. The state trajectory is estimated using the optimal virtual input trajectory from the previous time step, as described in Lam et al. (2010). Since the optimal trajectories are not expected to change much from one time step to the next, $\hat{\Theta}^*_k$ is a good approximation of the (unknown) optimal state trajectory Θ^*_k, which can be used to calculate a linear time-varying approximation to the cost function.

The overall linear time-varying MPCC scheme is summarised in Fig. 7.

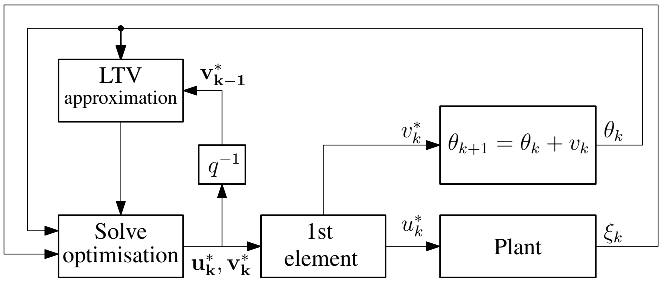
LTV
approximation
$\mathbf{v}^*_{\mathbf{k-1}}$
q^{-1}
v^*_k
$\theta_{k+1} = \theta_k + v_k$
θ_k
Solve
optimisation
$\mathbf{u}^*_\mathbf{k}, \mathbf{v}^*_\mathbf{k}$
1st
element
u^*_k
Plant
ξ_k

Fig. 7. Linear time-varying MPCC architecture

5. EXPERIMENTAL RESULTS

The model predictive contouring controller was implemented on the X-Y table test rig using MATLAB and xPC Target. The controller was programmed using Simulink and then compiled and downloaded to the Target PC for real-time execution. Linear time-varying MPCC was run with a sample period of $h = 4\,\text{ms}$ and horizon length $N = 7$. Hildreth's active set algorithm (Wang, 2009) was used to solve the constrained QP at each time step, with input constraints $|i_{c,x}|, |i_{c,y}| \leq 0.5$ and $0 \leq v \leq 1$, corresponding to current command saturation at $\pm 0.5\,\text{A}$ and a maximum path velocity of $0.25\,\text{m/s}$.

The desired contour path is shown in Fig. 8 and is represented by an arc-length-parameterised quintic spline

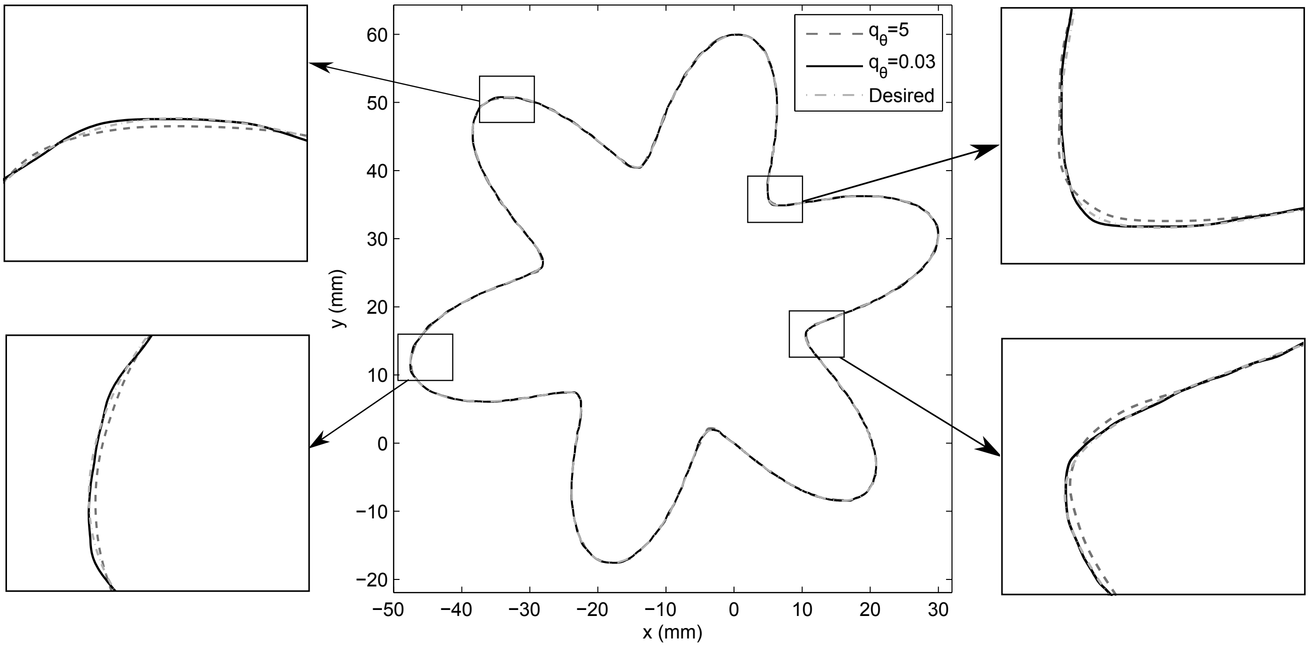

Fig. 8. MPCC Contour for $q_\theta = 0.03$ and $q_\theta = 5$

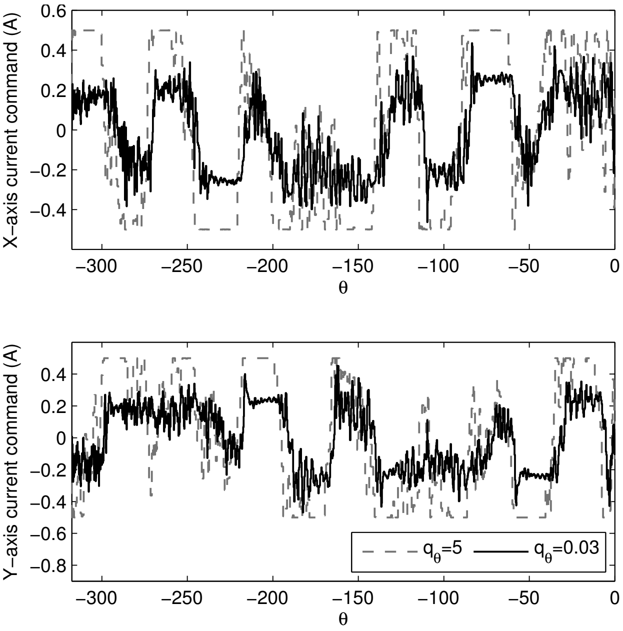

Fig. 9. Current commands vs. path parameter for $q_\theta = 0.03$ and $q_\theta = 5$

generated using the method proposed in Erkorkmaz and Altintas (2005). Tests were conducted for various values of the path speed weighting q_θ, while keeping the other weightings constant with $q_l = 200$, $q_c = 100$ and $R = diag(60, 60, 200)$.

Figs. 8, 9, 10 and 11 show plots of the contour, command currents, path speed, and contouring error respectively for $q_\theta = 0.03$ and $q_\theta = 5$. Fig. 10 demonstrates how the controller adjusts the path speed automatically to maintain accuracy, reducing the path speed around the tight curves on the contour. The effect of the cost function weights can also be observed from Figs. 10 and 11. For

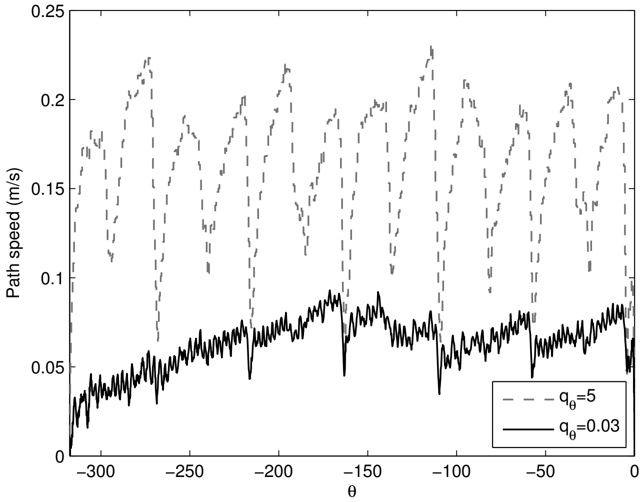

Fig. 10. Path speed vs. path parameter for $q_\theta = 0.03$ and $q_\theta = 5$

$q_\theta = 0.03$, the contouring error is lower, but with a lower path speed compared to $q_\theta = 5$.

The performance of MPCC was compared against the industry standard cascaded PI control scheme as well as an advanced scheme, model predictive tracking control (MPTC), at the same sample period. Additionally, the cascaded PI controller was tested at a faster sample period of 1 ms. For the tracking controllers, the desired path $(x^d(\theta), y^d(\theta))$ was converted to a reference trajectory (x^d_k, y^d_k) by applying a constant path velocity. This is in contrast to MPCC, where the controller sets the path velocity automatically. The cascaded PI control scheme is shown in Fig. 12. The integrator state is restricted to the saturation limits to prevent integral windup. The model predictive tracking controller minimises the following cost function in a receding horizon fashion, subject to the system constraints:

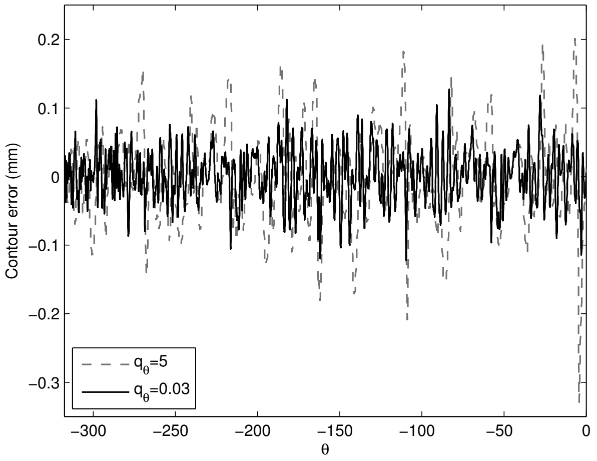

Fig. 11. Contour error vs. path parameter for $q_\theta = 0.03$ and $q_\theta = 5$

$$J_k^T = \sum_{i=1}^{N}\left(\begin{bmatrix} x_{k+i}-x^d_{k+i}\\ y_{k+i}-y^d_{k+i}\end{bmatrix}^T Q_t \begin{bmatrix} x_{k+i}-x^d_{k+i}\\ y_{k+i}-y^d_{k+i}\end{bmatrix} + \Delta u_{k+i}^T R_t \Delta u_{k+i}\right) \tag{18}$$

where Q_t and R_t are weighting matrices representing the relative importance of tracking accuracy and control deviations. For the tracking controllers, tests were conducted where the constant path velocity used to generate the reference trajectory was varied from 0.05m/s to 0.15 m/s.

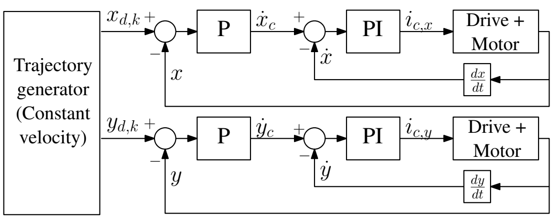

Fig. 12. Cascaded PI tracking control scheme

Fig. 13 shows a plot of the root mean square (RMS) contouring error versus traversal time for the MPCC, MPTC and cascaded PI controllers, with contouring error expressed as a percentage of the maximum radius of the contour shape. As expected, for all controllers the contouring accuracy improves for longer traversal times. The MPCC performs similarly to the 1 ms PI controller at low speeds, but achieves much better contouring accuracy at high speeds, despite the fact that the PI controller is operating four times faster. Comparing MPCC, MPTC and cascaded PI control at the same sample rate, MPCC outperforms the other controllers over the range of traversal times tested.

6. CONCLUSION

It has been demonstrated that model predictive contouring control can be implemented in real-time on a biaxial contouring system with a 4 ms sampling period, with significant performance improvements over tracking control schemes. However, servo controllers on industrial machines often operate at much faster sampling periods, in the order of 100μs. The development of a model predictive contouring controller suitable for implementation in industrial contouring systems is the subject of ongoing work.

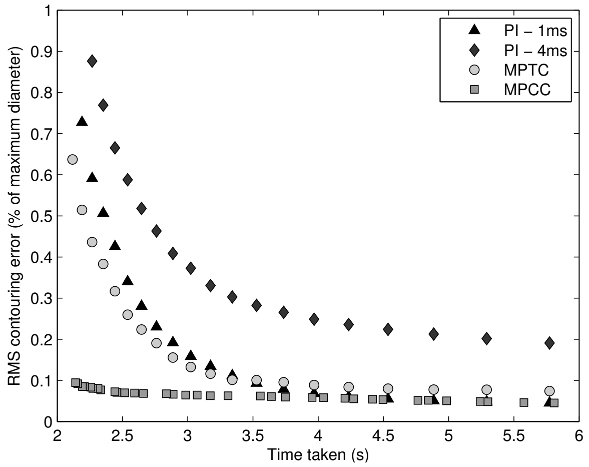

Fig. 13. RMS contouring error versus traversal time for cascaded PI, MPTC and MPCC

REFERENCES

Aguiar, A. P., Hespanha, J. P., Kokotivić, P. V., 2008. Performance limitations in reference tracking and path following for nonlinear systems. Automatica 4 (3), 598–610.

Dong, J., Ferreira, P., Stori, J., 2007. Feed-rate optimization with jerk constraints for generating minimum-time trajectories. Int. J. Mach. Tools Manuf. 47, 1941–1955.

Erkorkmaz, K., Altintas, Y., 2005. Quintic spline interpolation with minimal feed fluctuation. J. Manuf. Sci. Eng. 127, 339–349.

Erkorkmaz, K., Yeung, C.-H., Altintas, Y., 2006. Virtual CNC system. Part II. High speed contouring application. Int. J. Mach. Tools Manuf. 46, 1124–1138.

Faulwasser, T., Kern, B., Findeisen, R., 2009. Model predictive path-following for constrained nonlinear systems. In: Decision and Control, 48th IEEE Conference on. pp. 8642 –8647.

Imamura, F., Kaufman, H., 1991. Time optimal contour tracking for machine tool controllers. IEEE Control Syst. Mag. 11 (3), 11–17.

Koren, Y., Lo, C. C., 1992. Advanced controllers for feed drives. CIRP Ann. 41 (2), 689–698.

Lam, D., Manzie, C., Good, M., 2010. Model predictive contouring control. In: Decision and Control (CDC), 49th IEEE Conference on. pp. 6137–6142.

Maciejowski, J., 2002. Predictive Control with Constraints. Pearson Education Ltd.

Muske, K. R., Badgwell, T. A., 2002. Disturbance modeling for offset-free linear model predictive control. J. Process Control 12, 617–632.

Renton, D., Elbestawi, M., 2000. High speed servo control of multi-axis machine tools. Int. J. Mach. Tools Manuf. 40, 539–559.

Verscheure, D., Demeulenaere, B., Swevers, J., Schutter, J. D., Diehl, M., 2009. Time-optimal path tracking for robots: A convex optimization approach. IEEE Trans. Autom. Control 54 (10), 2318–2327.

Wang, L., 2009. Model Predictive Control System Design and Implementation Using MATLAB. Springer-Verlag.