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
{l amd, 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...
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 encoder 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.

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).

\[
\begin{bmatrix}
\dot{\psi}_{x} \\
\dot{\phi}_{x} \\
\dot{\psi}_{y} \\
\dot{\phi}_{y}
\end{bmatrix}
= \begin{bmatrix}
\frac{1}{J_{m,x}}(T_{m,x} + k_c(\psi_x - \dot{\psi}_x) + c(\dot{\phi}_x - \dot{\psi}_x) - F_{i,x}) \\
\frac{1}{J_{l,x}}(k_c(\psi_x - \dot{\phi}_x) + c(\dot{\phi}_x - \dot{\psi}_x) - F_{l,x}) \\
\frac{1}{J_{m,y}}(T_{m,y} + k_c(\psi_y - \dot{\psi}_y) + c(\dot{\phi}_y - \dot{\psi}_y) - F_{m,y}) \\
\frac{1}{J_{l,y}}(k_c(\psi_y - \dot{\phi}_y) + c(\dot{\phi}_y - \dot{\psi}_y) - F_{l,y})
\end{bmatrix},
\]

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 \( \psi \) and \( \phi \) 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

\[
x = \tau \phi_x, \quad y = \tau \phi_y,
\]

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

\[
T_{m,x} = K_l i_x, \quad T_{m,y} = K_l i_y,
\]

where \( K_l \) 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

\[
F_{m,x} = b_m \dot{\psi}_x, \quad F_{l,x} = b_t \dot{\phi}_x, \\
F_{m,y} = b_m \dot{\psi}_y, \quad F_{l,y} = b_t \dot{\phi}_y,
\]

where \( b_m, b_t \) 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

\[
\xi_{k+1} = A \xi_k + B u_k, \quad \begin{bmatrix} x_k \\ y_k \end{bmatrix} = C \xi_k, \quad u \in [u_{\text{min}}, u_{\text{max}}],
\]

where \( \xi = [\psi_x, \dot{\psi}_x, \phi_x, \dot{\phi}_x, \psi_y, \dot{\psi}_y, \phi_y, \dot{\phi}_y]^T \), \( u = [i_{c,x}, i_{c,y}]^T \), \( u_{\text{min}}, u_{\text{max}} \) are the constraints on the command currents, and \( A, B, C \) 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
Table 1: X-Y table model parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Unit</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>Motor inertia $J_m$</td>
<td>kg·m²</td>
<td>$2.3 \times 10^{-5}$</td>
</tr>
</tbody>
</table>
| Table inertia $J_l$        | kg·m² | $2.10 \times 10^{-5}$  
|                            |       | $2.33 \times 10^{-5}$  |
| Motor viscous friction $b_m$ | Nm/s/rad | $1.30 \times 10^{-3}$  
|                            |       | $1.17 \times 10^{-3}$  |
| Motor viscous friction $b_l$ | Nm/s/rad | $0.001$  
| Motor torque constant $K_t$ | Nm/A | $0.4364$  
| Coupling stiffness $K_c$  | Nm/rad | $2.6682$  
| Coupling damping $c$      | Nm/s/rad | $5.0 \times 10^{-4}$  
| Lead screw pitch $p$      | mm    | $0.7958$  

Fig. 4. X-Y table model parameters

Table 2: Parameters for the control algorithms

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Unit</th>
<th>Values</th>
</tr>
</thead>
</table>
| Motor torque constant $K_t$ | Nm/A | $0.4364$  
| Coupling stiffness $K_c$  | Nm/rad | $2.6682$  
| Coupling damping $c$      | Nm/s/rad | $5.0 \times 10^{-4}$  
| Lead screw pitch $p$      | mm    | $0.7958$  

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 task control 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 $\theta$:

$$x^d : [\theta^s, 0] \rightarrow \mathbb{R}; \ y^d : [\theta^s, 0] \rightarrow \mathbb{R}; \ \theta^s < 0,$$ (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 $\epsilon_c$ is defined as the normal deviation from the desired path (Koren and Lo, 1992), and can be expressed as

$$\epsilon_c = \sin(\theta^r) \left(x_k - x^d(\theta^r)\right) - \cos\phi(\theta^r) \left(y_k - y^d(\theta^r)\right),$$

$$\phi(\theta^r) = \arctan\left(\frac{v_y(\theta^r)}{v_x(\theta^r)}\right),$$ (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.

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 $\theta$:

$$\dot{\theta}_{k+1} = \theta_{k+1} + \Delta \theta_k, \ \theta_k \in [0, \theta_{max}], \ \theta_{max} > 0, \ \theta_k \in [\theta^s, 0]$$ (9)

where $\theta_k$ 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 $\theta_k \geq 0$.

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

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

Remark 1. Let $\epsilon_l$ denote the path distance that $(x^d(\theta^r), y^d(\theta^r))$ lags $(x^d(\theta_l), y^d(\theta_l))$ and approximate $\epsilon_l$ as

$$\epsilon_l(\xi_k, \theta_k) = -\cos(\theta_k)(x_k - x^d(\theta_k)) - \sin(\theta_k)(y_k - y^d(\theta_k))$$ (10)

Refer to Fig. 6 for a graphical interpretation of $\epsilon_c$, $\epsilon_l$ and their approximations. It can be observed that for most paths, $\epsilon_l(\theta_k) \rightarrow 0$. Therefore, for practical purposes, Assumption 1 can be enforced by including an appropriate penalty on $\epsilon_l^2$ in the cost function (12).

By Assumption 1, the contouring error can be approximated by

$$\epsilon^2(\xi_k, \theta_k) = \sin(\theta_k)(x_k - x^d(\theta_k)) - \cos(\theta_k)(y_k - y^d(\theta_k)),$$ (11)

and $\theta_k$ 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:

$$J_k = \sum_{k=0}^{N-1} \left[ \alpha^2 \left(\epsilon(\xi_k, \theta_k)^2\right) + \beta \left(\Delta u_k^2\right) \right].$$ (12)
The desired contour path is shown in Fig. 8 and is represented by an arc-length-parameterised quintic spline approximation of the optimal predicted path state trajectory, $\Theta_k^*$, which can then 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.

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 $h = 4$ 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 $|v_c|, |v_q| \leq 5$ A and $0 \leq v \leq 1$, corresponding to current command saturation at $\pm 0.5$ A and a maximum path velocity of 0.25 m/s.

The desired contour path is shown in Fig. 8 and is represented by an arc-length-parameterised quintic spline approximation of the optimal predicted path state trajectory, $\Theta_k^*$, which can then 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.

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

Minimise $J_k^*$

Subject to (6),

\[
\begin{align*}
\theta_{k+1} &= \theta_{k+1} + v_{k+1}, \\
u_{k+1} &\in [\text{min}, \text{max}], \\
v_k &\in [0, v_{\text{max}}], \\
\theta_{k+1} &\in [\theta_{\text{min}}, \theta_{\text{max}}],
\end{align*}
\]

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

Minimise $\frac{1}{2} e_k^T H_k e_k + z_k^T G_k$.

Subject to $E_k z_k \leq F_k$, (17)

where $z_k = [v_k, \ldots, v_k, \Delta u_k, \ldots, \Delta u_k]$, $H_k$ and $G_k$ 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 $\Theta_k^*$, 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.

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

Minimise $J_k^*$

Subject to (6),

\[
\begin{align*}
\theta_{k+1} &= \theta_{k+1} + v_{k+1}, \\
u_{k+1} &\in [\text{min}, \text{max}], \\
v_k &\in [0, v_{\text{max}}], \\
\theta_{k+1} &\in [\theta_{\text{min}}, \theta_{\text{max}}],
\end{align*}
\]

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

Minimise $\frac{1}{2} e_k^T H_k e_k + z_k^T G_k$.

Subject to $E_k z_k \leq F_k$, (17)

where $z_k = [v_k, \ldots, v_k, \Delta u_k, \ldots, \Delta u_k]$, $H_k$ and $G_k$ 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 $\Theta_k^*$, 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.

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

Minimise $J_k^*$

Subject to (6),

\[
\begin{align*}
\theta_{k+1} &= \theta_{k+1} + v_{k+1}, \\
u_{k+1} &\in [\text{min}, \text{max}], \\
v_k &\in [0, v_{\text{max}}], \\
\theta_{k+1} &\in [\theta_{\text{min}}, \theta_{\text{max}}],
\end{align*}
\]

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

Minimise $\frac{1}{2} e_k^T H_k e_k + z_k^T G_k$.

Subject to $E_k z_k \leq F_k$, (17)

where $z_k = [v_k, \ldots, v_k, \Delta u_k, \ldots, \Delta u_k]$, $H_k$ and $G_k$ 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 $\Theta_k^*$, 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.
generated using the method proposed in Erkorkmaz and Altintas (2005). Tests were conducted for various values of the path speed weighting \( q_\theta \), while keeping the other weightings constant with \( q_l = 200 \), \( q_c = 100 \) and \( R = \text{diag}(60, 60, 200) \).

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 \( q_\theta = 0.03 \), 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 \((x_k^d, y_k^d)\) 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 \( q_\theta = 5 \)

\[
J_k^T = \sum_{i=1}^{N} \begin{bmatrix} x_{k+i} - x_{d,k+i} \\ y_{k+i} - y_{d,k+i} \end{bmatrix}^T Q_i \begin{bmatrix} x_{k+i} - x_{d,k+i} \\ y_{k+i} - y_{d,k+i} \end{bmatrix} + \Delta u_{k+i}^T R_i \Delta u_{k+i} 
\]

(18)

where \( Q_i \) and \( R_i \) 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 0.05 m/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 100 \( \mu \)s. The development of a model predictive contouring controller suitable for implementation in industrial contouring systems is the subject of ongoing work.

REFERENCES


