

# Model-based Control Input Compensation for Parallel Kinematic Machine Tools

Michael Hadorn\*,

Institute of Machine Tools and Manufacturing (IWF-ETH), Zurich, Switzerland<sup>†</sup>

## Abstract

This paper presents a new model-based approach for control input compensation leading to better performing machine tools with nonlinear behavior. An optimization algorithm has been developed and implemented for the parallel kinematic machine tool (PKM) *Hexaglide*, developed at the IWF in Zurich. With the developed algorithm the contour feed along a given path is maximized offline according to the limited maximum actuator speed and force leading to minimization of execution time. In addition to the capabilities of similar algorithms the presented solution supports automatic optimization of an entire NC program. Some basic considerations are presented for the dynamics of PKMs. The optimization algorithm is also described. Results of optimized trajectories are shown. Their execution time is compared with traditionally created trajectories using constant limits.

**Keywords:** Optimization, Control, CNC.

## 1 Introduction

The treatment of control input is called *Control Input Compensation (CIC)* (see [1]). Several areas and methods for CIC have been examined in the past, e.g. time or frequency-based filtering of the controller inputs; geometric modulation of the path within certain tolerances, e.g. prevention of corners and discontinuous curvature by smoothing the path [2]; contour feed control along a given path [3]/[1]. In this paper, *contour feed control* along a given path is regarded. Hereby, appropriate criteria have to be kept. Typical criteria are *maximum actuator speed*  $\dot{q}_{max}$  and *maximum actuator force*  $\tau_{max}$ . Traditional algorithms introduce velocity profiles (e.g. trapezoidal or  $\sin^2$  profiles) where discontinuities in the feed at corners or changing feed rate occur. In general, the machine tool is assumed to behave linearly. Mainly in the field of PKMs - built for high dynamic applications - special attention has to be paid to CIC. The sig-

nificant nonlinearities of a PKM lead to strong dependencies of the dynamic capabilities on the actual position and orientation of the tool. Therefore traditional algorithms for the CIC described above clearly fail to fully utilize the capabilities of PKMs.

### 1.1 Some examples of nonlinear behavior

The following figures show examples of nonlinear behavior. The first example illustrates *direction dependency* of the contour feed limit, whereas the second example illustrates *position dependency* of acceleration limits.

Figure 2 shows maximum contour feed dependency on its direction in the Y-Z plane for the PKM *Hexaglide* (see figure 1). The Hexaglide is a 6-DOF milling PKM with constant strut length and moving base-points ("glide"-principle). Its equations of motion are significantly nonlinear.

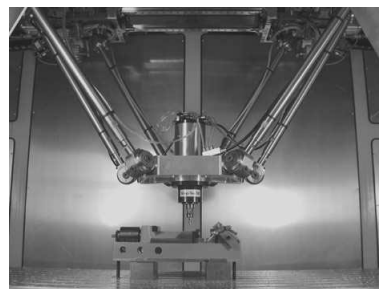


Figure 1: PKM Hexaglide, developed at the IWF Zurich. In the middle of the figure, the spindle is shown. On the top of the figure, the actuators (linear motors) are shown.

Limits regarding maximum actuator speed and force are plotted. The resulting area of allowed feed is compared with a square area defined by conventional CIC (hatched area). Thereby, constant feed limits for the Y and Z direction are used (see also section 5). As it can be seen, the area defined by conventional CIC is considerably smaller than the model-based one.

\*hadorn@iwf.bepr.ethz.ch

<sup>†</sup>Tannenstrasse 3, CH-8092 Zuerich, iwf.bepr.ethz.ch

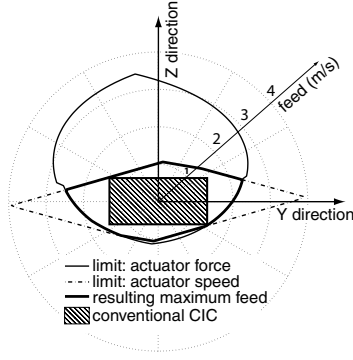


Figure 2: Polar plot for direction dependent maximum contour feed in the Y-Z plane. Computed for the PKM *Hexaglide*. Tool center point position is the approximate working volume center:  $X=0$ ,  $Y=0$ ,  $Z=-1$  m.

The position dependent maximum acceleration in +Y direction from standstill is shown in figure 3. Maximum acceleration varies from near 0 to  $25m/s^2$  in the working volume. Conventional CIC would have to reduce the acceleration in +Y direction to a very low limit that is valid for the whole working volume. Therefore the machine tool would move with acceleration far from possible in most of the working volume.

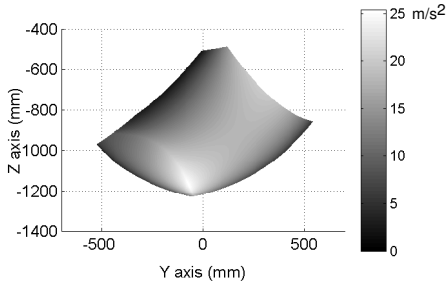


Figure 3: Maximum acceleration in +Y direction in the working volume for the Y-Z plane ( $X=0$ ). Computed for the PKM *Hexaglide*.

## 2 Task formulation

### 2.1 Tool path formulation

Paths for machine tools are typically given by *NC programs*. Thereby, geometric commands define the nominal contour. Process requirements define the feed along the path during machining, which has to be expressed in the NC program by programming of

the *feed rate*. In contrary, movement during *rapid traverse* should be executed as fast as possible (e.g. tool or workpiece change).

For each geometric command  $i$  (block index  $i$ ) of an NC program, the path is given analytically (e.g. a circular arc). The 6-tuple  $\mathbf{x}$  containing the position and orientation of the tool in the workpiece coordinate system is given as a function of a path parameter  $s$  (e.g. arc length of the path):

$$\mathbf{x}^i = \mathbf{x}^i(s), \quad s_0^i \leq s \leq s_1^i \quad (1)$$

1<sup>st</sup> and 2<sup>nd</sup> time derivatives:

$$\dot{\mathbf{x}}^i(s(t)) = (\mathbf{x}^i)'(s) \dot{s} \quad (2)$$

$$\ddot{\mathbf{x}}^i(s(t)) = (\mathbf{x}^i)''(s) \dot{s}^2 + (\mathbf{x}^i)'(s) \ddot{s} \quad (3)$$

### 2.2 Machine model

According to [4], the equations of motion in the form of the inverse dynamic problem of general mechanical multi-body systems can be expressed as the *configuration space equation*

$$\begin{aligned} \tau &= \mathbf{M}(\mathbf{x})\ddot{\mathbf{x}} + \mathbf{B}_1(\mathbf{x})\kappa_1 + \mathbf{B}_2(\mathbf{x})\kappa_2 \\ &\quad + \mathbf{C}(\mathbf{x})\dot{\mathbf{x}} + \mathbf{g}(\mathbf{x}), \\ \kappa_1 &:= (\dot{x}_1\dot{x}_2 \quad \dot{x}_1\dot{x}_3 \quad \cdots \quad \dot{x}_{n-1}\dot{x}_n)^T, \\ \kappa_2 &:= (\dot{x}_1^2 \quad \dot{x}_2^2 \quad \cdots \quad \dot{x}_n^2)^T. \end{aligned} \quad (4)$$

Combination of equations 1-4 results in

$$\tau = \mathbf{m}(s^i)\ddot{s}^i + \mathbf{b}(s^i)(\dot{s}^i)^2 + \mathbf{c}(s^i)\dot{s}^i + \mathbf{g}(s^i) \quad (5)$$

for NC block  $i$ .

With equation 5, the equation of motion ("along the path") is expressed in terms of the path parameter  $s^i$  and its time derivatives. This accords to a *reduction* of the information to the *effectively needed* information to solve the problem of contour feed control. This fact is expressed in the reduction of the system matrices  $\mathbf{M}$ ,  $\mathbf{B}$ , etc. in equation 4 to vectors  $\mathbf{b}$ ,  $\mathbf{c}$ , etc in equation 5.

In equation 5, most of the relevant physical effects on multi-body-systems are modelled. But the important part of Coulomb friction in the actuators is still neglected (see [5]). Therefore, equation 5 has been completed by the following term representing Coulomb friction:

$$\begin{aligned} \tau_c &= \mathbf{w}(s), \\ \mathbf{w}(s) &:= \begin{cases} 0 & \dot{s} = 0, \\ \text{sign}\{\mathbf{J}^{-1}(\mathbf{x})\mathbf{x}'(s)\} \mathbf{W}_c & \text{otherwise.} \end{cases} \quad (6) \\ \mathbf{W}_c &:= \text{diag}\{(w_{c_1} \quad \cdots \quad w_{c_n})\} \end{aligned}$$

whereby  $\mathbf{W}_c$  is a diagonal matrix, containing the Coulomb friction forces and  $\mathbf{J}$  is the Jacobian.

## 2.3 Optimization Problem

The following optimization problem is formulated for one NC block  $i$  (one-block subproblem). The transition conditions between the blocks are expressed in the boundary conditions for the one-block subproblem (such that no steps in the contour feed between the blocks occur).

**Given:** A path according to equation 1 and a machine model according to equations 5 and 6.

**Solve for:**  $\dot{s}(s)$ , such that

$$I := \int_{t_0}^{t_1} dt \rightarrow \min. \quad (7)$$

The minimization of time  $t$  in equation 7 accords to a *maximization of contour feed  $\dot{s}$*  along the path [6]. Thus the following algorithms simply try to maximize  $\dot{s}(s)$  without looking at the cost function 7.

**Constraints:** The constraint are limited tool feed, actuator feed, actuator force and boundary conditions.

## 3 Algorithm

To solve the optimization problem over all the NC blocks and the one-block subproblem described in section 2.3, two steps are necessary. They are described in sections 3.1 and 3.2.

### 3.1 Overall routine

Transitions between path segments defined in the form of equation 1 are at least  $C^0$ -continuous (no steps occur for  $\mathbf{x}$ ). Thus, "corners" and transitions with discontinuous curvature may often appear. To prepare transition conditions for the following one-block subproblem the overall routine has two choices:

- Reducing feed while passing the discontinuity to a low value defined by the programmer,
- substituting parts of adjoining NC blocks by smoothing blocks.

Both strategies are realized in the system described in this paper. Nevertheless, the description of these would go beyond the scope of this paper.

### 3.2 One-block subproblem algorithm

The algorithm solving the one-block subproblem described in section 2.3 is divided into two parts (according to [6])<sup>1</sup>:

1. Finding the maximum allowed *boundary feed*,  $\dot{s}_{max}(s_k)$  by setting  $\ddot{s} = 0 \forall s_k$ ,
2. connecting the areas of increasing and decreasing  $\dot{s}_{max}(s_k)$  by maximum allowed *boundary acceleration*,  $\ddot{s}_{max}(s_k)$ .

Hereby, a sampled trajectory  $\dot{s}(s_k)$  with sampling index  $k$  is regarded.

**Step 1: Finding boundary feed.** The computation of  $\dot{s}_{max}$  is based on the constraints in section 2.3, since all these constraints are functions of  $\dot{s}$ . Thereby, the actuator force constraint is reduced by setting  $\ddot{s} = 0 \forall s_k$ . Note that there is no requirement on  $C^0$ -continuity of  $\dot{s}_{max}(s)$ , since  $C^0$ -continuity will be reached after having done step 2. Especially when actuators have to change their direction of motion, Coulomb friction will change its sign. This may lead to  $C^0$ -discontinuities of  $\dot{s}_{max}(s)$ .

**Step 2: Finding boundary acceleration.** After having computed  $\dot{s}_{max}$ , boundary acceleration  $\ddot{s}_{max}(s_k)$  has to be found. Thereto,  $\dot{s}_{max}$  is corrected such that the actuator force constraint is achieved. In this step, an increasing  $\dot{s}_{max}(s_k)$  is optimized forwardly, decreasing  $\dot{s}_{max}(s_k)$  backwardly.

For illustration, figure 4 shows an example. At position (1), the trajectory starts with boundary acceleration, until it reaches boundary feed at (2). The next interval of increasing feed starts at (3). Thus, optimization goes on at this point until it reaches boundary feed at (4). No more interval of increasing feed can be found in this example. Therefore, backward optimization is done by starting at (5), until it reaches the (corrected) boundary feed at (6), etc.

## 4 Interface to the machine's NC

The computed feed is first divided into adequate parts. These parts are then approximated by a 5<sup>th</sup> order polynomial with a least-squares procedure using weighting and damping such that:

$$\dot{s}_{opt}(s) = \vartheta_5 s^5 + \vartheta_4 s^4 + \vartheta_3 s^3 + \vartheta_2 s^2 + \vartheta_1 s + \vartheta_0. \quad (8)$$

The coefficients  $\vartheta_i$  are then stored in a modified NC program together with the given geometric commands.

<sup>1</sup>In the following explanations the block index  $i$  is omitted.

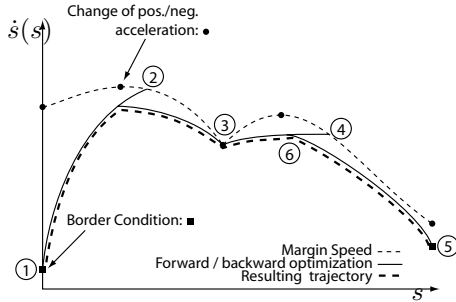


Figure 4: Example for optimization algorithm.

## 5 Example

The following example has been computed for the *IWF-Hexaglide*. The given tool path is shown in figure 5. The tool first moves in its *distinct* direction

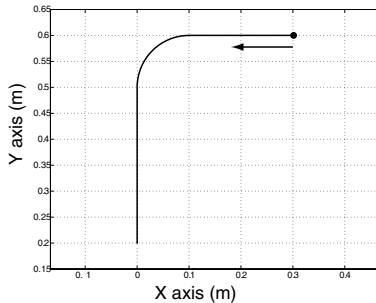


Figure 5: Path example

(-X direction), where the dynamic capacity of the machine does not change<sup>2</sup>. It then moves on a quarter circular arc and moves on in -Y direction.

The resulting feed is shown in figure 6, whereby the dash-dotted curve is the boundary feed computed by step 1 of the one-block subproblem algorithm. Step 2 leads to the solid curve with boundary acceleration, which is the resulting time optimal feed (see section 3.2). In this example no feed reduction over corners or corner rounding is necessary because the tool path is  $C^1$  continuous<sup>3</sup>.

For comparison the feed for the trajectory in figure 5 has been optimized with both conventional CIC methods (i.e. limitation of speed and acceleration in the tool and actuator coordinate system). The results are shown in figure 7.

<sup>2</sup>The existence of a distinct direction is typical for PKMs with glide-principle.

<sup>3</sup> $C^2$  discontinuities appear in this path, but they are not yet treated in this work.

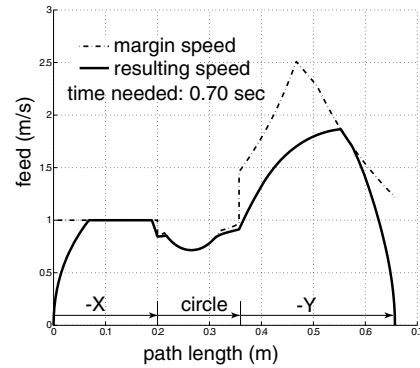


Figure 6: Computed optimal contour feed along the given path.

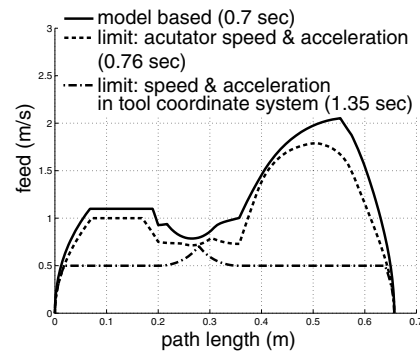


Figure 7: Comparison of 3 methods to get an optimal contour feed.

## References

- [1] Bernhard Siegrist. *Lagegeregelte Vorschubantriebe mit Führungsgrößen-Kompensation höherer Ordnung*. PhD thesis, ETH Zürich, IWF, Tannenstrasse 3, 8092 Zürich, Switzerland, 1990.
- [2] Guang Yu. *Optimale Steuerung der Bewegung und der Geschwindigkeit für das drei- und fünfachsige Fräsen*. PhD thesis, ETH Zürich, IWF, Tannenstrasse 3, 8092 Zürich, Switzerland, 1996.
- [3] Matthias Fauser. *Steuerungstechnische Massnahmen für die Hochgeschwindigkeits-Bearbeitung*. PhD thesis, RWTH Aachen, 1997.
- [4] John J Craig. *Adaptive Control of Mechanical Manipulators*. Addison-Wesley, 1988.
- [5] Michael Hadorn. Concept and application of a model-based cic for pkms. In Reimund Neugebauer, editor, *The 3rd Chemnitz PK Seminar*, pages 351–369. IWU, Chemnitz, 2002.
- [6] Rainer Johanni. *Optimale Bahnplanung bei Industrierobotern*. VDI Verlag, Düsseldorf, 1988.