Accuracy and efficiency improvements for CNC machines based on advanced toolpath descriptions and motion control algorithms

Rida T. Farouki

Department of Mechanical & Aerospace Engineering,
University of California, Davis

— synopsis —

- real–time CNC interpolators — PH curves vs. G codes
- variable feedrates for constant material removal rate
- inverse dynamics problem for minimization of path error
- high-speed cornering under axis acceleration bounds
- exact contour error computation for cross-coupled control
- optimal orientations for contour machining of surfaces

3-axis “open architecture” CNC mill

- MHO Series 18 Compact Mill
- 18” $\times$ 18” $\times$ 12” work volume
- Yaskawa brushless DC motors
- zero-backlash precision ball screws
- linear encoders, ± 0.001” accuracy
- MDSI OpenCNC control software
- custom real-time interpolators

G codes – traditional tool path specification

approximate general curved paths by many short linear/circular moves

`G01` = linear move, `G02/G03` = clockwise/anti-clockwise circular move

`X,Y` = target point, `I,J` = offsets from current location to circle center

```
N01 G01 X0 Y0 F37200
N02 G01 X-41 Y87
N03 G01 X-62 Y189
N04 G02 X-23 Y478 I654 J0
N05 G01 X474 Y1015
  ... etc.
```

- accurate path specification $\Longrightarrow$ voluminous part programs
- block look-ahead problem for acceleration/deceleration management
- aliasing effects in HSM, when G code length comparable to $V\Delta t$

real-time CNC interpolators

- computer numerical control (CNC) machine: digital control system
- within each sampling interval ($\Delta t \sim 10^{-3}$ sec) of servo system, compares actual position (measured by encoders on machine axes) with reference point (computed by real-time interpolator)
- real-time CNC interpolator algorithm — given parametric curve $\mathbf{r}(\xi)$ and speed (feedrate) function V, compute reference-point parameter values $\xi_1, \xi_2, \ldots$ in real time:

$$\int_0^{\xi_k} \frac{|\mathbf{r}'(\xi)|\,\mathrm{d}\xi}{V} = k\Delta t\,, \quad k = 1, 2, \ldots$$

- general parametric curve — compute ξ_k by Taylor series expansion
- Pythagorean-hodograph (PH) curves — analytic reduction of "interpolation integral" $\Rightarrow$ accurate & efficient real-time interpolator

Taylor series expansions

Suppose $\xi(t)$ specifies time variation of the parameter along $\mathbf{r}(\xi)$ when traversed with (constant or variable) feedrate V.

Reference–point parameter value ξ_{k+1} obtained from preceding value ξ_k by Taylor–series expansion

$$\xi_{k+1} = \xi_k + \dot{\xi}(t_k)\Delta t + \tfrac{1}{2}\ddot{\xi}(t_k)\,(\Delta t)^2 + \cdots$$

of $\xi(t)$ about $t = t_k = k\Delta t$, where dots denote time derivatives

$\Longrightarrow$ need expressions for time derivatives $\dot{\xi}(t), \ddot{\xi}(t), \ldots$ of $\xi(t)$.

For a given curve $\mathbf{r}(\xi)$, the parametric speed σ and feedrate V are defined in terms of cumulative arc length s along $\mathbf{r}(\xi)$ by

$$\sigma = |\mathbf{r}'(\xi)| = \frac{\mathrm{d}s}{\mathrm{d}\xi}, \qquad V = \frac{\mathrm{d}s}{\mathrm{d}t}$$

Time derivatives can be converted to parametric derivatives using

$$\frac{\mathrm{d}}{\mathrm{d}t} = \frac{\mathrm{d}s}{\mathrm{d}t}\frac{\mathrm{d}\xi}{\mathrm{d}s}\frac{\mathrm{d}}{\mathrm{d}\xi} = \frac{V}{\sigma}\frac{\mathrm{d}}{\mathrm{d}\xi}$$

Successive applications of $\mathrm{d}/\mathrm{d}t$ give

$$\dot{\xi} = \frac{V}{\sigma}, \qquad \ddot{\xi} = \frac{\sigma V' - \sigma' V}{\sigma^2}\,\dot{\xi}\,,$$

$$\dddot{\xi} = \frac{\sigma V' - 3\sigma' V}{\sigma^2}\,\ddot{\xi} + \frac{\sigma V'' - \sigma'' V}{\sigma^2}\,\dot{\xi}^2\,, \quad \mathrm{etc.}\,,$$

where primes indicate derivatives with respect to ξ. Derivatives of the parametric speed can be expressed recursively as

$$\sigma' = \frac{\mathbf{r}' \cdot \mathbf{r}''}{\sigma}, \qquad \sigma'' = \frac{\mathbf{r}' \cdot \mathbf{r}''' + |\mathbf{r}''|^2 - \sigma'^2}{\sigma}, \quad \mathrm{etc.}$$

For variable feedrate, must express $V', V'', \ldots$ in terms of derivatives with respect to variable that V is specified as a function of

time-dependent feedrate: $V(t)$ — acceleration/deceleration rates

$$V' = \frac{\sigma}{V}\frac{\mathrm{d}V}{\mathrm{d}t}\,, \qquad V'' = \frac{\sigma'}{V}\frac{\mathrm{d}V}{\mathrm{d}t} - \frac{\sigma^2}{V^3}\left(\frac{\mathrm{d}V}{\mathrm{d}t}\right)^2 + \frac{\sigma^2}{V^2}\frac{\mathrm{d}^2V}{\mathrm{d}t^2}$$

arc-length-dependent feedrate: $V(s)$ — distance along trajectory

$$V' = \sigma\,\frac{\mathrm{d}V}{\mathrm{d}s}\,, \qquad V'' = \sigma'\,\frac{\mathrm{d}V}{\mathrm{d}s} + \sigma^2\,\frac{\mathrm{d}^2V}{\mathrm{d}s^2}$$

curvature-dependent feedrate: $V(\kappa)$ — control material removal rate

$$V' = \sigma\,\frac{\mathrm{d}\kappa}{\mathrm{d}s}\frac{\mathrm{d}V}{\mathrm{d}\kappa}\,, \qquad V'' = \left(\sigma'\,\frac{\mathrm{d}\kappa}{\mathrm{d}s} + \sigma^2\,\frac{\mathrm{d}^2\kappa}{\mathrm{d}s^2}\right)\frac{\mathrm{d}V}{\mathrm{d}\kappa} + \left(\sigma\,\frac{\mathrm{d}\kappa}{\mathrm{d}s}\right)^2\frac{\mathrm{d}^2V}{\mathrm{d}\kappa^2}$$

$V(\kappa)$ case requires arc–length derivatives of curvature:

$$\kappa = \frac{(\mathbf{r}' \times \mathbf{r}'') \cdot \mathbf{z}}{\sigma^3}, \qquad \frac{\mathrm{d}\kappa}{\mathrm{d}s} = \frac{(\mathbf{r}' \times \mathbf{r}''') \cdot \mathbf{z} - 3\sigma^2\sigma'\kappa}{\sigma^4},$$

$$\frac{\mathrm{d}^2\kappa}{\mathrm{d}s^2} = \frac{(\mathbf{r}'' \times \mathbf{r}''' + \mathbf{r}' \times \mathbf{r}'''') \cdot \mathbf{z} - 3\sigma(2\sigma'^2 + \sigma\sigma'')\kappa - 7\sigma^3\sigma'(\mathrm{d}\kappa/\mathrm{d}s)}{\sigma^5}.$$

problems with Taylor series interpolators

- finite # of terms in Taylor series $\Longrightarrow$ unknown truncation error
- coefficients of higher–order terms very complicated & costly to compute $\Longrightarrow$ incompatible with real–time computing
- several papers give erroneous coefficients for Taylor interpolators

Pythagorean-hodograph (PH) curves

$\mathbf{r}(\xi) =$ PH curve in $\mathbb{R}^n$ $\iff$ components of hodograph $\mathbf{r}'(\xi)$

are elements of a Pythagorean $(n+1)$–tuple of polynomials

PH curves incorporate special algebraic structures in their hodographs

- rational offset curves $\mathbf{r}_d(\xi) = \mathbf{r}(\xi) + d\,\mathbf{n}(\xi)$
- polynomial parametric speed $\sigma(\xi) = |\mathbf{r}'(\xi)| = \dfrac{\mathrm{d}s}{\mathrm{d}\xi}$
- polynomial arc-length function $s(\xi) = \displaystyle\int_0^\xi |\mathbf{r}'(\xi)|\,\mathrm{d}\xi$
- energy integral $E = \displaystyle\int_0^1 \kappa^2\,\mathrm{d}s$ has closed-form evaluation
- real-time CNC interpolators, rotation-minimizing frames, etc.

Planar & spatial Pythagorean-hodograph curves

$$x'^2(t) + y'^2(t) = \sigma^2(t) \quad \Longleftrightarrow \quad \left\{ \begin{array}{l} x'(t) = u^2(t) - v^2(t) \\ y'(t) = 2\,u(t)v(t) \\ \sigma(t) = u^2(t) + v^2(t) \end{array} \right.$$

choose complex polynomial $\mathbf{w}(t) = u(t) + \mathrm{i}\,v(t)$

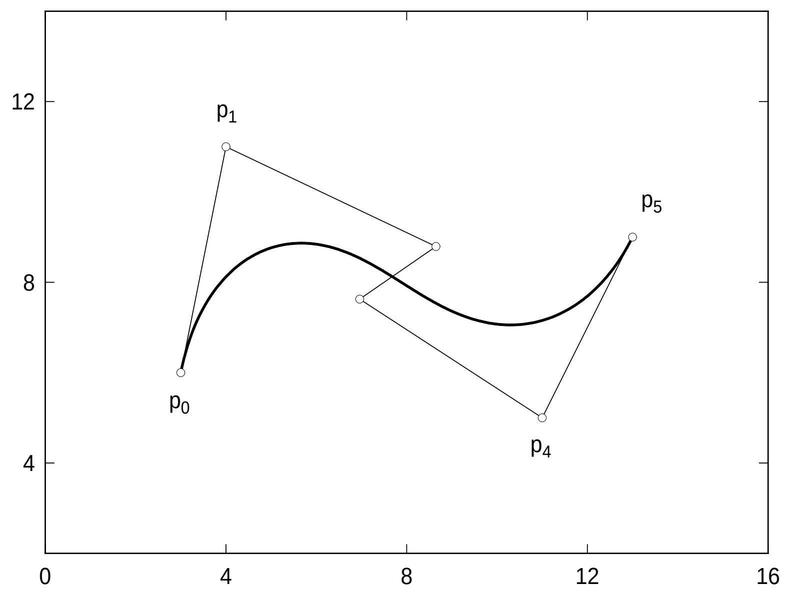
planar Pythagorean hodograph — $\mathbf{r}'(t) = (x'(t), y'(t)) = \mathbf{w}^2(t)$

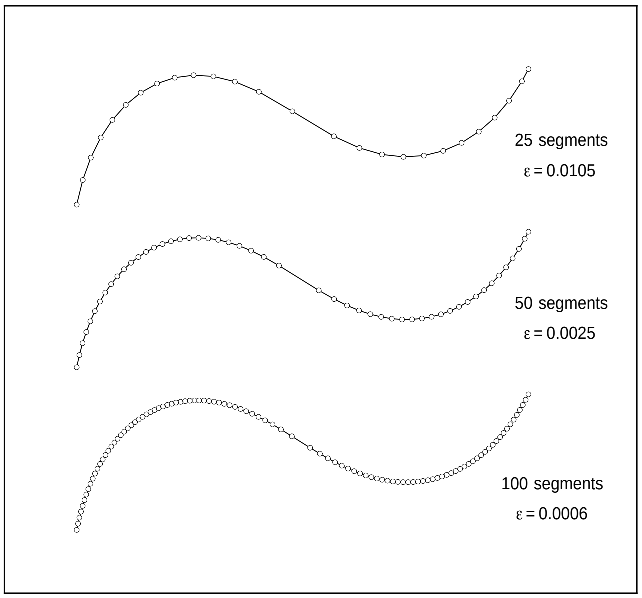
$$x'^2(t) + y'^2(t) + z'^2(t) = \sigma^2(t) \quad \Longleftrightarrow \quad \left\{ \begin{array}{l} x'(t) = u^2(t) + v^2(t) - p^2(t) - q^2(t) \\ y'(t) = 2\,[\,u(t)q(t) + v(t)p(t)\,] \\ z'(t) = 2\,[\,v(t)q(t) - u(t)p(t)\,] \\ \sigma(t) = u^2(t) + v^2(t) + p^2(t) + q^2(t) \end{array} \right.$$

choose quaternion polynomial $\mathcal{A}(t) = u(t) + v(t)\,\mathbf{i} + p(t)\,\mathbf{j} + q(t)\,\mathbf{k}$

spatial Pythagorean hodograph — $\mathbf{r}'(t) = (x'(t), y'(t), z'(t)) = \mathcal{A}(t)\,\mathbf{i}\,\mathcal{A}^*(t)$

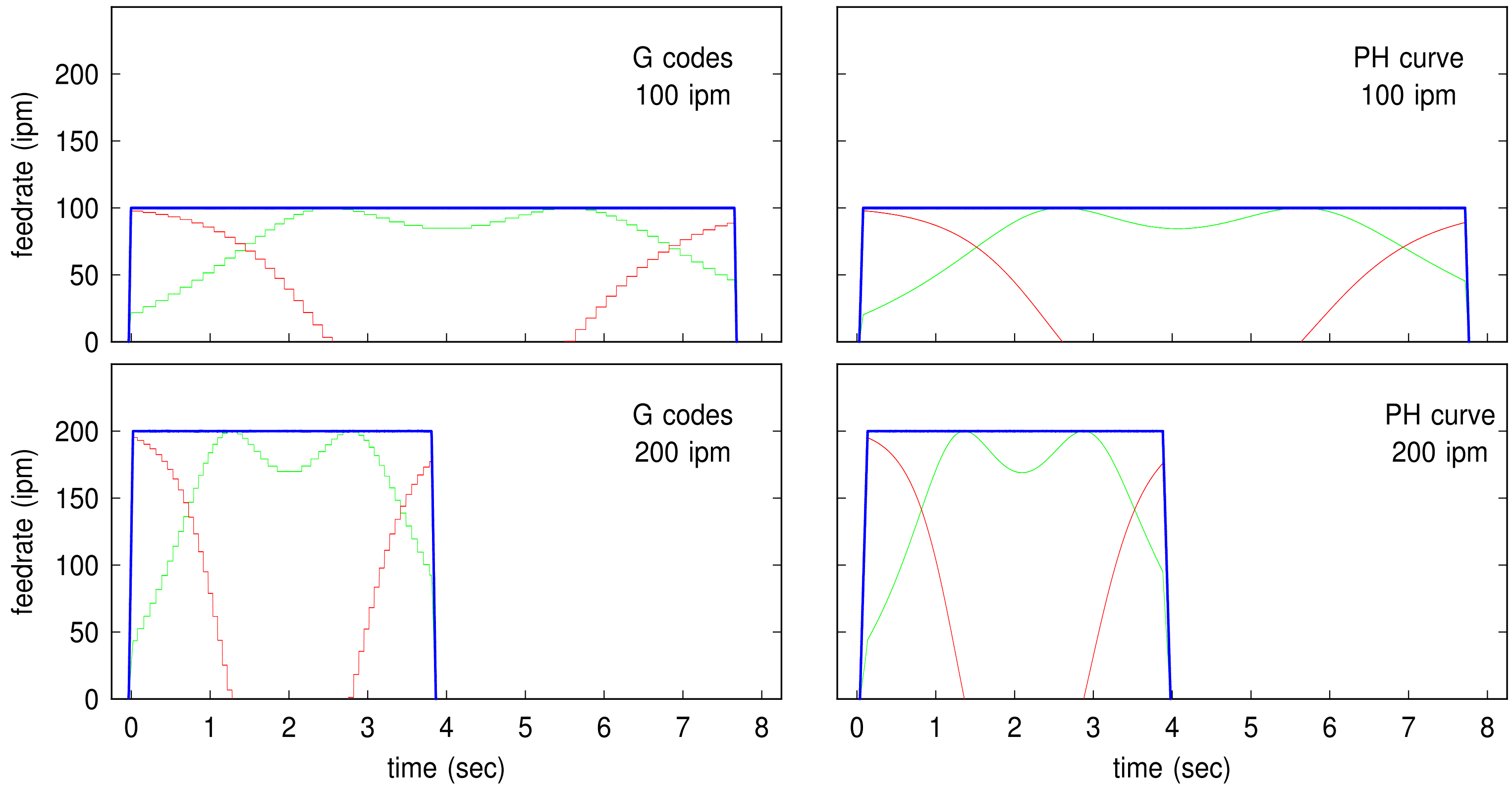
real-time CNC interpolators for Pythagorean-hodograph (PH) curves





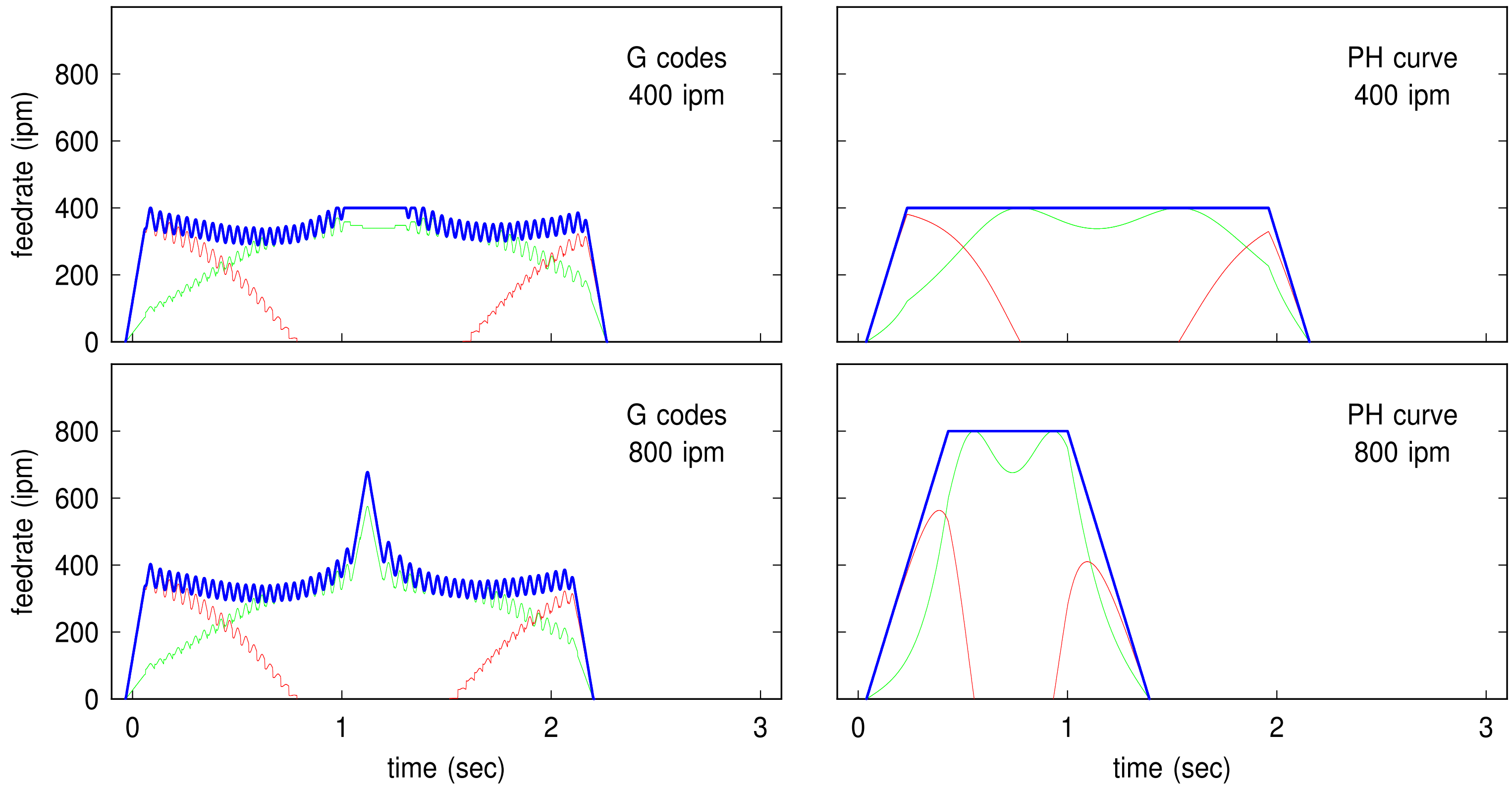
Left: analytic tool path description (quintic PH curve). Right: approximation of path to various prescribed tolerances using piecewise-linear G codes.

comparative feedrate performance: 100 & 200 ipm



The G code and PH curve interpolators both give excellent performance (red) at 100 and 200 ipm. The “staircase” nature of the x and y feedrate components (blue and green) for the G codes indicate faithful reproduction of the piecewise-linear path, while the PH curve yields smooth variations.

comparative feedrate performance: 400 & 800 ipm



At 400 & 800 ipm, the PH curve interpolator continues to yield impeccable performance – but the performance of the G code interpolator is severely degraded by "aliasing" effects, incurred by the finite sampling frequency and discrete nature of the piecewise-linear path description.

repertoire of feedrate variations for PH curves

real-time CNC interpolator: given parametric curve $\mathbf{r}(\xi)$ and feedrate variation V, compute reference-point parameter values $\xi_1, \xi_2, \ldots$ from

$$\int_0^{\xi_k} \frac{|\mathbf{r}'(\xi)|\,\mathrm{d}\xi}{V} = k\Delta t$$

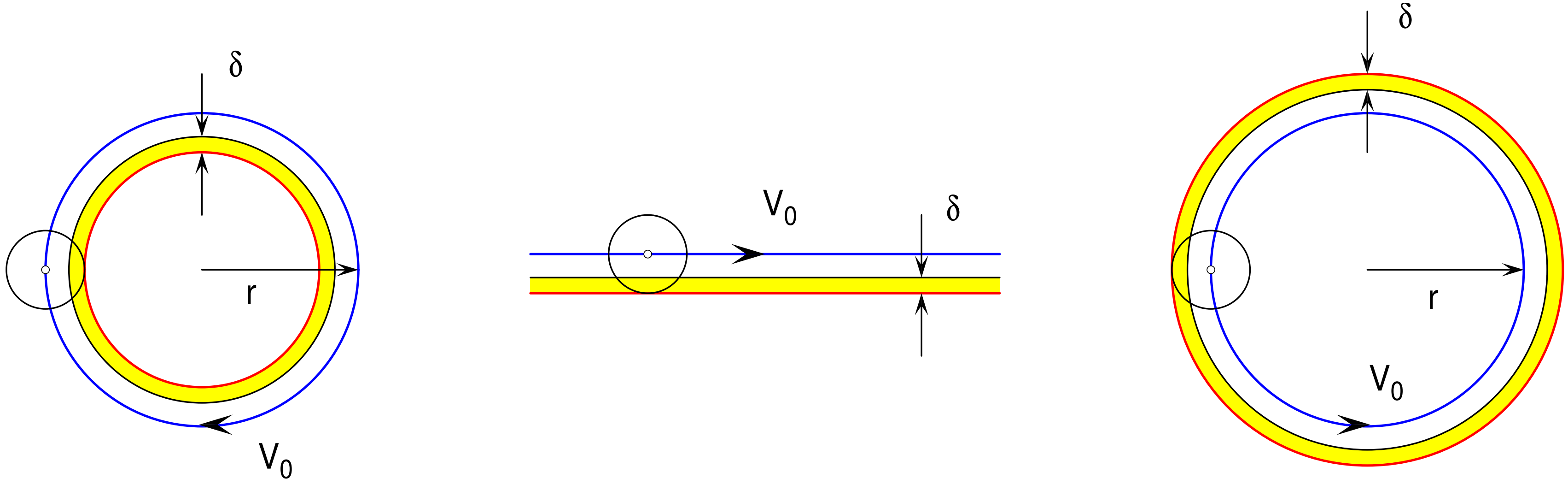
instead of Taylor series expansion, PH curves admit analytic reduction of the interpolation integral, for many feedrate variations of practical interest:

◦ constant V, linear or quadratic dependence $V(s)$ on arc length s

◦ time-dependent feedrate, for any easily integrable function $V(t)$ — useful for acceleration and deceleration management

◦ curvature-dependent feedrate for constant material removal rate (MRR) at fixed depth of cut δ — $V(\kappa) = V_0\,[\,1 + \kappa(d - \frac{1}{2}\delta)\,]^{-1}$

material removal rate (MRR) as function of curvature

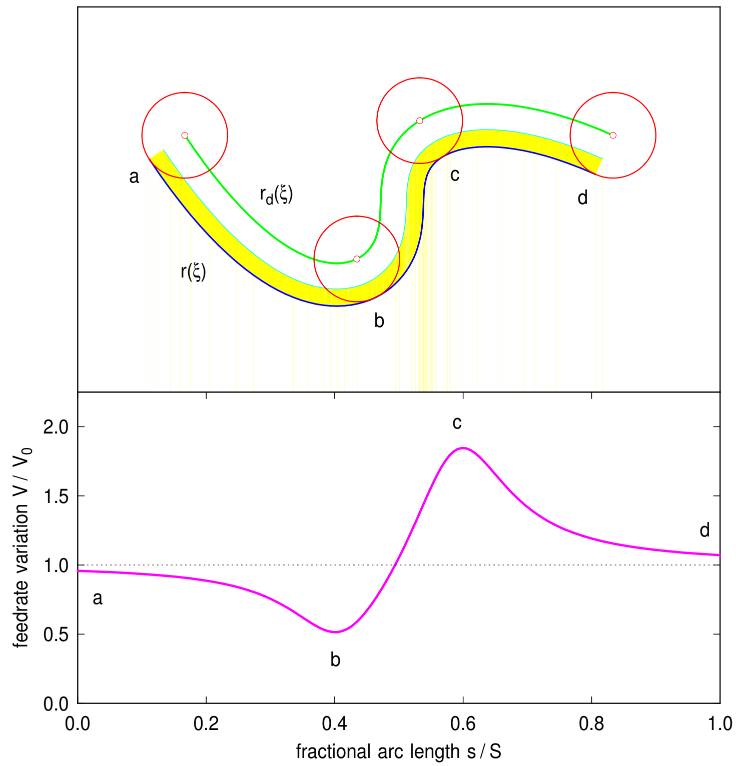


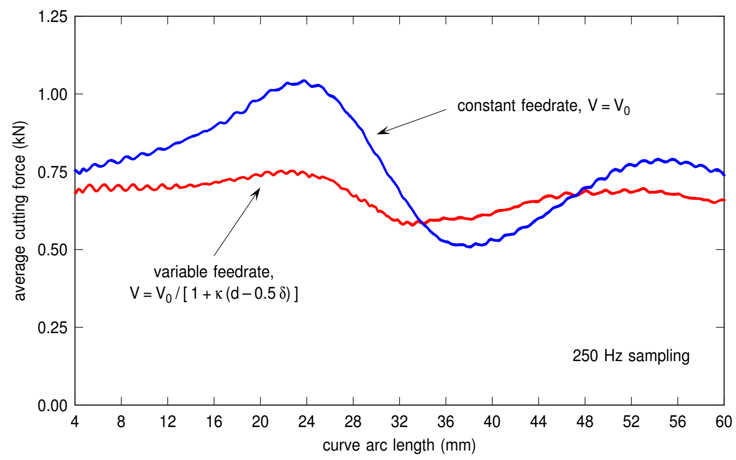
clockwise, $\kappa = -1/r$ linear, $\kappa = 0$ anticlockwise, $\kappa = 1/r$

Volume removed (yellow area) by a cylindrical tool of radius d moving with feedrate V_0 through depth of cut δ for: a clockwise circular path of radius r (left); a linear path (center); and an anti–clockwise circular path of radius r (right). In each case, the MRR can be expressed in terms of the curvature as $V_0\,\delta\,[\,1+\kappa(d-\frac{1}{2}\delta)\,]$. Hence, to maintain constant MRR, one should use the curvature–dependent feedrate

$$V(\xi) \,=\, \frac{V_0}{1\,+\,\kappa(\xi)\,(d-\frac{1}{2}\delta)}\,.$$

curvature-dependent feedrate for constant MRR





G codes for PH curve tool paths

`G05` = quintic PH curve, `A,B,C,P,Q,R` = coefficients of $u(t), v(t)$

`X,Y` = target point, `F & U,V,W` = variable feedrate type & parameters

```
N05 G05 H5 F0 U37200
N10 G05 X1092 Y-294 A-31.026 B-38.537 C-31.481 P16.934 Q-16.436 R13.062
N15 G05 X1470 Y-1386 A-31.481 B-24.426 C-28.476 P13.062 Q42.560 R2.794
N20 G05 X2226 Y-294 A-28.476 B-32.526 C-36.896 P2.794 Q-36.972 R-14.887
N25 G05 X3444 Y504 A-36.896 B-41.267 C-32.579 P-14.887 Q7.197 R-25.365
N30 G05 X2226 Y1722 A-32.579 B-23.892 C8.378 P-25.365 Q-57.927 R-36.389
N35 G05 X2100 Y504 A8.378 B40.647 C21.987 P-36.389 Q-14.851 R-28.356
N40 G05 X1134 Y252 A21.987 B3.326 C-13.761 P-28.356 Q-41.861 R-28.525
N45 G05 X756 Y1050 A-13.761 B-30.849 C-3.668 P-28.525 Q-15.189 R-32.746
N50 G05 X0 Y0 A-3.668 B23.514 C31.026 P-32.746 Q-50.304 R-16.934
```

R. T. Farouki, J. Manjunathaiah, G–F. Yuan, G codes for the specification of Pythagorean–hodograph tool paths and associated feedrate functions on open–architecture CNC machines, *International Journal of Machine Tools & Manufacture* **39**, 123–142 (1999)

allows combination of traditional (linear/circular) G codes and PH quintics, with variable feedrates — dependent on time, arc length, or curvature

inverse dynamics problem for path error minimization

C. A. Ernesto & R. T. Farouki (2010), Solution of inverse dynamics problems for contour error minimization in CNC machines, *International Journal of Advanced Manufacturing Technology* **49**, 589–604

inertia (resistance to motion) and damping (frictional energy dissipation) of CNC machine axes prevent exact execution of commanded motion

develop dynamic model of machine/controller system, expressed in terms of linear ordinary differential equations

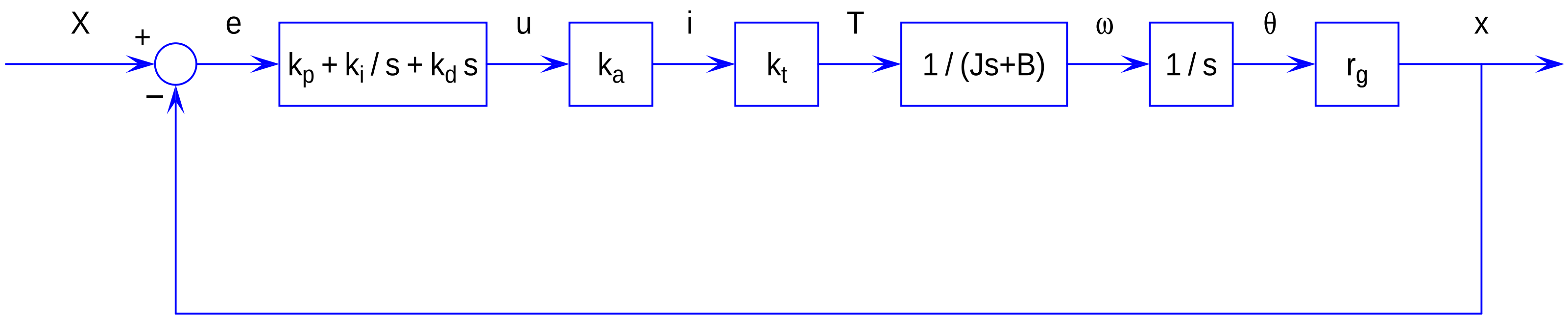
transform independent variable from the time t to the curve parameter ξ : constant coefficients $\rightarrow$ polynomial coefficients

revert differential equations: swap input & output dependent variables

solve reverted differential equations for modified input path that, subject to machine dynamics, exactly yields desired output path

for brevity, consider only x–axis motion (same principles for y, z axes)

block diagram of CNC machine x–axis drive with PID controller



X = commanded position from real–time interpolator
x = actual position as measured by position encoders
$e = X - x$ = instantaneous x–axis position error
k_p, k_i, k_d = proportional, integral, derivative gains
u = output voltage from controller
i = current from current amplifier
T = torque from DC electric motor
J, B = x–axis inertia and damping
ω, θ = motor shaft angular speed & position
r_g = transmission ratio (angular $\rightarrow$ linear conversion)

machine/controller dynamics

CNC machine executes actual path $(x(t), y(t))$ determined from commanded path $(X(t), Y(t))$ by differential equations of the form

$$
\begin{aligned}
a_x \dddot{x} + b_x \ddot{x} + c_x \dot{x} + x &= d_x \ddot{X} + e_x \dot{X} + X \,, \\
a_y \dddot{y} + b_y \ddot{y} + c_y \dot{y} + y &= d_y \ddot{Y} + e_y \dot{Y} + Y \,,
\end{aligned}
$$

where dots indicate time derivatives, constant coefficients $a_x, b_x, \ldots$ depend on the machine/controller physical parameters.

But commanded path $(X(\xi), Y(\xi))$ specified by general parameter ξ, rather than time t.

Transform independent variable: time $t \rightarrow$ curve parameter ξ

$$
\frac{\mathrm{d}}{\mathrm{d}t} = \frac{\mathrm{d}s}{\mathrm{d}t} \frac{\mathrm{d}\xi}{\mathrm{d}s} \frac{\mathrm{d}}{\mathrm{d}\xi} = \frac{V}{\sigma} \frac{\mathrm{d}}{\mathrm{d}\xi} \,,
$$

where $\sigma = \mathrm{d}s/\mathrm{d}\xi =$ parametric speed, and $V = \mathrm{d}s/\mathrm{d}t =$ feedrate.

If $\xi(t)$ specifies time variation of parameter when $(X(\xi), Y(\xi))$ is traversed with (constant or variable) feedrate V, its time derivatives are

$$\dot{\xi} = \frac{V}{\sigma}\,, \quad \ddot{\xi} = \frac{\sigma V' - \sigma' V}{\sigma^2}\,\dot{\xi}\,, \quad \dddot{\xi} = \frac{\sigma V' - 3\sigma' V}{\sigma^2}\,\ddot{\xi} + \frac{\sigma V'' - \sigma'' V}{\sigma^2}\,\dot{\xi}^2\,, \quad \text{etc.}$$

and we have

$$\frac{\mathrm{d}}{\mathrm{d}t} = \dot{\xi}\,\frac{\mathrm{d}}{\mathrm{d}\xi}\,, \quad \frac{\mathrm{d}^2}{\mathrm{d}t^2} = \dot{\xi}^2\,\frac{\mathrm{d}^2}{\mathrm{d}\xi^2} + \ddot{\xi}\,\frac{\mathrm{d}}{\mathrm{d}\xi}\,, \quad \frac{\mathrm{d}^3}{\mathrm{d}t^3} = \dot{\xi}^3\,\frac{\mathrm{d}^3}{\mathrm{d}\xi^3} + 3\,\dot{\xi}\,\ddot{\xi}\,\frac{\mathrm{d}^2}{\mathrm{d}\xi^2} + \dddot{\xi}\,\frac{\mathrm{d}}{\mathrm{d}\xi}\,, \quad \text{etc.}$$

Hence transformed differential equations become

$$\begin{aligned}
\alpha_x(\xi)\,x''' + \beta_x(\xi)\,x'' + \gamma_x(\xi)\,x' + \delta_x(\xi)\,x \;&=\; \lambda_x(\xi)\,X'' + \mu_x(\xi)\,X' + \nu_x(\xi)\,X\,, \\
\alpha_y(\xi)\,y''' + \beta_y(\xi)\,y'' + \gamma_y(\xi)\,y' + \delta_y(\xi)\,y \;&=\; \lambda_y(\xi)\,Y'' + \mu_y(\xi)\,Y' + \nu_y(\xi)\,Y\,,
\end{aligned}$$

where primes denote derivatives with respect to ξ, and $\alpha_x(\xi), \beta_x(\xi), \ldots$ are polynomials in ξ if $(X(\xi), Y(\xi))$ is a PH curve.

Now revert the differential equations — solve "backwards" to find input required to produce desired output.

Input = modified path $(\hat{X}(\xi), \hat{Y}(\xi))$, output = desired path $(X(\xi), Y(\xi))$

$$
\begin{aligned}
\lambda_x(\xi)\hat{X}'' + \mu_x(\xi)\hat{X}' + \nu_x(\xi)\hat{X} &= \alpha_x(\xi)X''' + \beta_x(\xi)X'' + \gamma_x(\xi)X' + \delta_x(\xi)X, \\
\lambda_y(\xi)\hat{Y}'' + \mu_y(\xi)\hat{Y}' + \nu_y(\xi)\hat{Y} &= \alpha_y(\xi)Y''' + \beta_y(\xi)Y'' + \gamma_y(\xi)Y' + \delta_y(\xi)Y.
\end{aligned}
$$

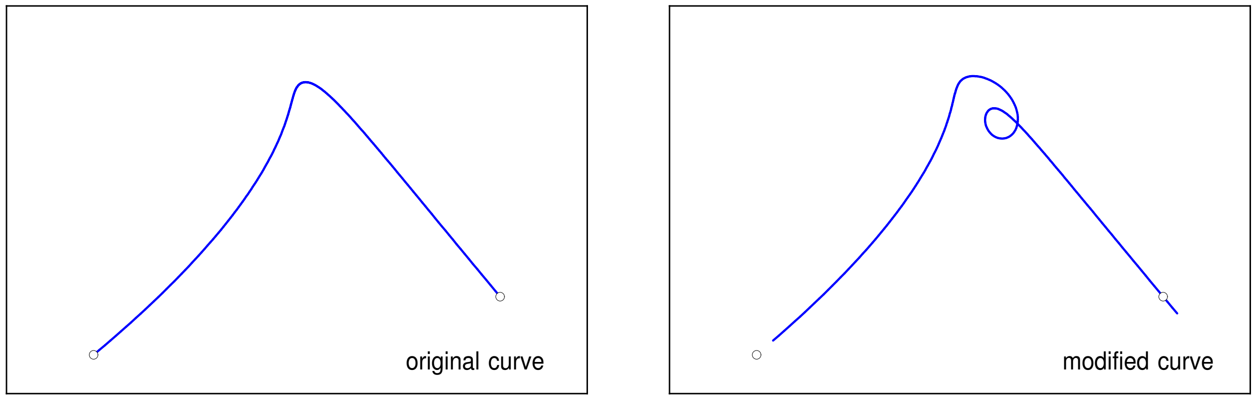
Must solve initial value problem for linear ODEs in $\hat{X}(\xi)$, $\hat{Y}(\xi)$ with known polynomials in ξ as coefficients and right–hand sides.

Simplest case: P controller with $k_i = k_d = 0 \Rightarrow \lambda_x(\xi) = \mu_x(\xi) \equiv 0$ and $\lambda_y(\xi) = \mu_y(\xi) \equiv 0$. Can solve exactly for $\hat{X}$, $\hat{Y}$ as

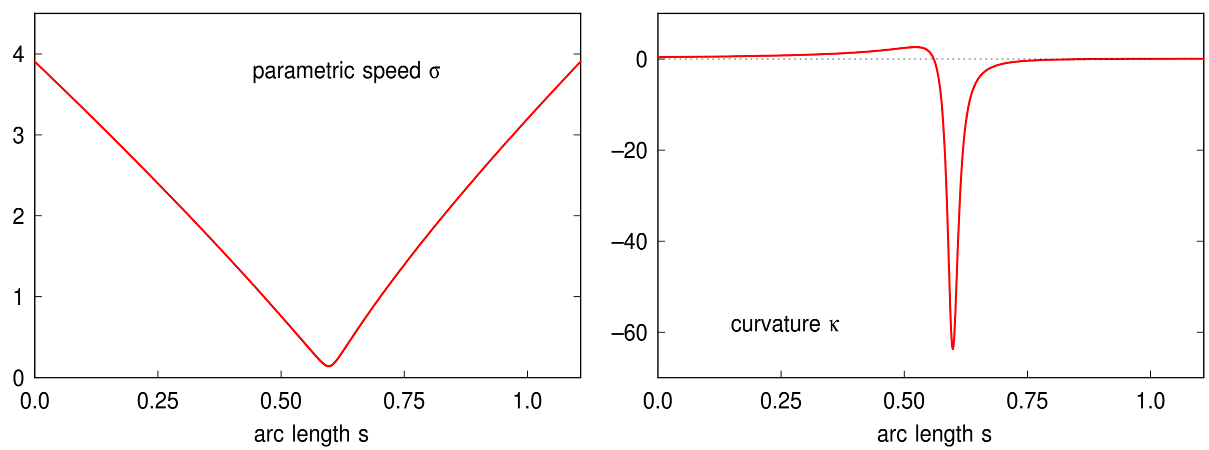
$$
\begin{aligned}
\hat{X} &= \frac{b_x\sigma V^2 X'' + V\,[\,b_x(\sigma V' - \sigma' V) + c_x\sigma^2\,]\,X' + \sigma^3 X}{\sigma^3}, \\
\hat{Y} &= \frac{b_y\sigma V^2 Y'' + V\,[\,b_y(\sigma V' - \sigma' V) + c_y\sigma^2\,]\,Y' + \sigma^3 Y}{\sigma^3}.
\end{aligned}
$$

Modified path <u>exactly</u> determined as higher–order rational Bézier curve!

example: quintic PH curve and P controller



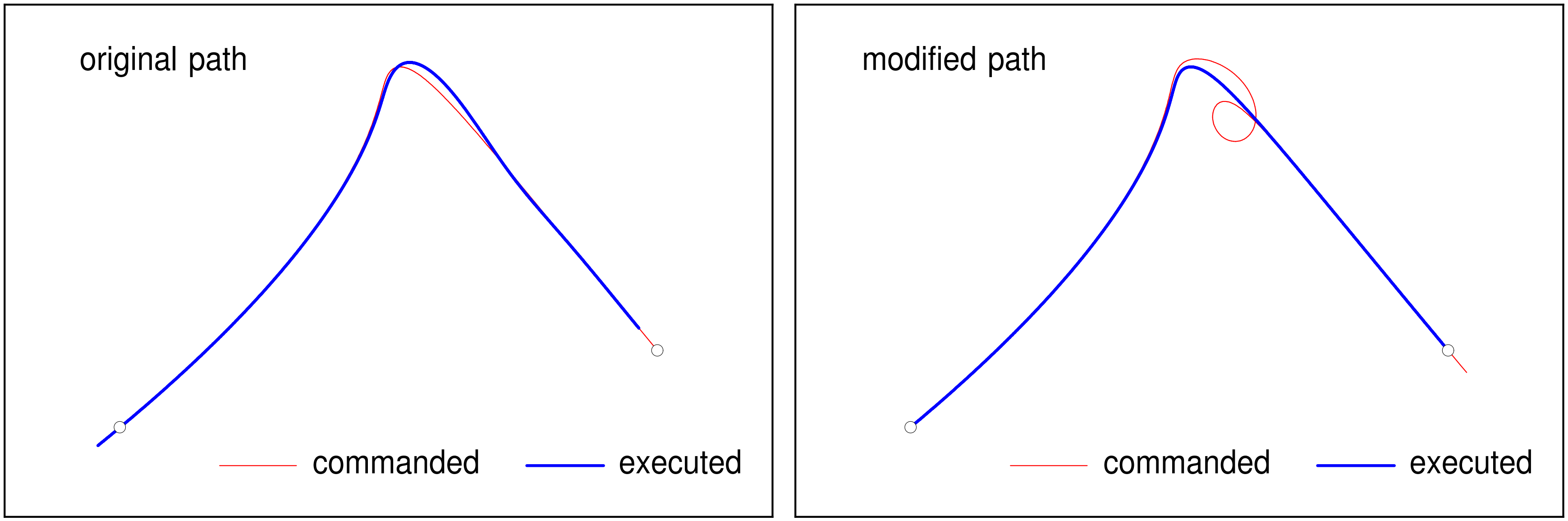
Left: quintic PH curve defining desired path $(X(\xi), Y(\xi))$. Right: modified path $(\hat{X}(\xi), \hat{Y}(\xi))$ that compensates for the machine/controller dynamics, for a P controller with gain $k_p = 10$ and constant feedrate $V = 0.12$ m/s.



Extreme variation of parametric speed and curvature on quintic PH curve.

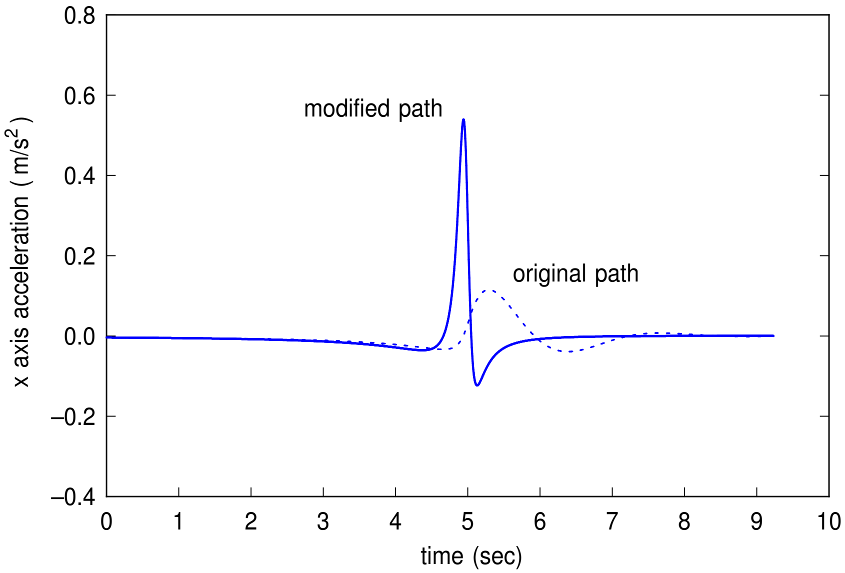
comparison of output for original and modified paths

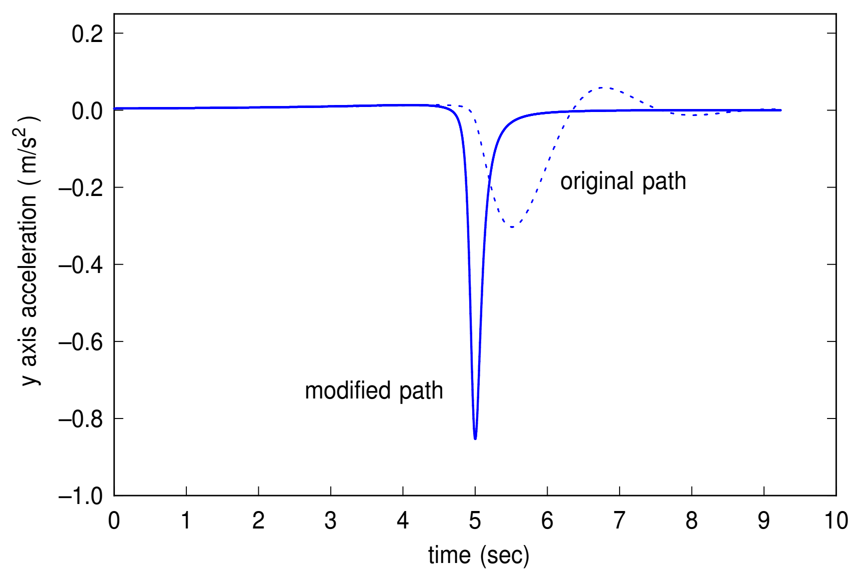
P controller with gain $k_p = 10$ and constant feedrate $V = 0.12$ m/s



Comparison of commanded and executed motions for original (left) and modified (right) paths. In the former case, the executed motion deviates significantly from the desired path. In the latter case, the executed motion is essentially indistinguishable from the original commanded path.

axis accelerations for original and modified paths





Comparison of x and y axis accelerations for original and modified paths. The modified path incurs greater peak accelerations, requiring a higher motor torque capacity.

efficient high-speed cornering

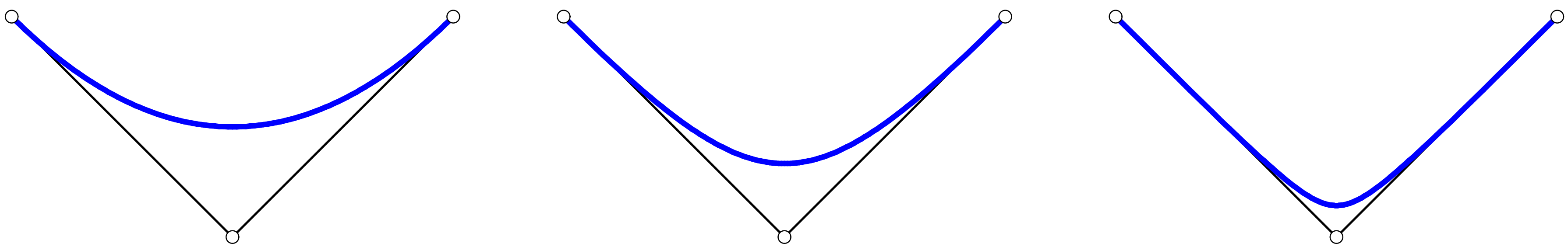
C. A. Ernesto & R. T. Farouki (2011), High-speed cornering by CNC machines under prescribed bounds on axis accelerations and toolpath contour error, *International Journal of Advanced Manufacturing Technology*, to appear

- exact traversal of sharp corner in toolpath requires zero feedrate — high deceleration & acceleration rates increase execution time, may incur large contour errors
- round corner with G^1 Bézier conic "splice" segment with deviation satisfying prescribed geometrical tolerance ϵ
- specify square of feedrate on conic segment as a Bernstein-form polynomial, to be determined optimization problem with point-wise constraints arising from machine axis acceleration bounds
- applying constraints to Bernstein coefficients, optimization problem can be solved to any desired accuracy by a monotonically convergent sequence of linear programming problems, using subdivision methods

tolerance-based conic splice segments

standard-form rational quadratic Bézier curve

$$\mathbf{r}(\xi) \;=\; \frac{\mathbf{p}_0\,(1-\xi)^2 \;+\; w_1\mathbf{p}_1\,2(1-\xi)\xi \;+\; \mathbf{p}_2\,\xi^2}{(1-\xi)^2 \;+\; w_1\,2(1-\xi)\xi \;+\; \xi^2}$$



left: ellipse ($w_1 < 1$); center: parabola ($w_1 = 1$); right: hyperbola ($w_1 > 1$)

monotone relationship between weight w_1 and geometrical tolerance ϵ

given $\mathbf{r}(\xi) = (x(\xi), y(\xi))$ minimize $T = \int_0^1 \frac{|\mathbf{r}'(\xi)|}{V(\xi)}\,\mathrm{d}\xi$

with respect to coefficients $c_0, \ldots, c_n$ of squared feedrate

$$V^2(\xi) = \sum_{k=0}^{n} c_k \binom{n}{k} (1-\xi)^{n-k} \xi^k$$

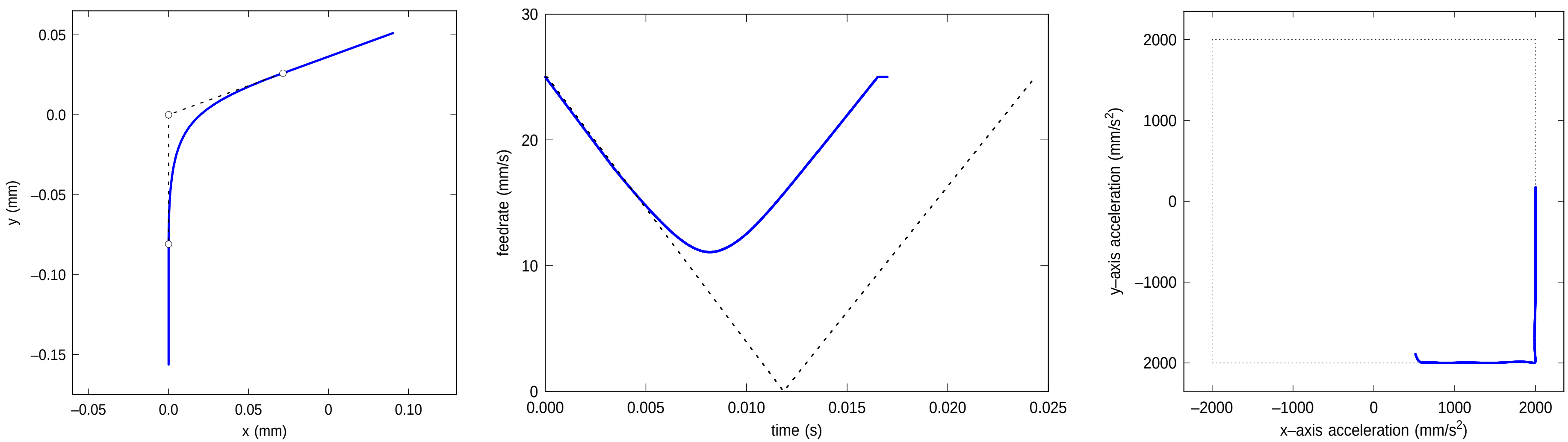
subject to point-wise A_x, A_y axis acceleration constraints

$$\begin{aligned}
(x'^2 + y'^2)(x''V^2 + x'VV') - (x'x'' + y'y'')x'V^2 - A_x(x'^2 + y'^2)^2 &\le 0\,, \\
(x'^2 + y'^2)(x''V^2 + x'VV') - (x'x'' + y'y'')x'V^2 + A_x(x'^2 + y'^2)^2 &\le 0\,, \\
(x'^2 + y'^2)(y''V^2 + y'VV') - (x'x'' + y'y'')y'V^2 - A_y(x'^2 + y'^2)^2 &\le 0\,, \\
(x'^2 + y'^2)(y''V^2 + y'VV') - (x'x'' + y'y'')y'V^2 + A_y(x'^2 + y'^2)^2 &\le 0\,.
\end{aligned}$$

by subdivision $\rightarrow$ convergent sequence of linear programming problems

high-speed cornering: computed example

tolerance: $\epsilon = 0.015$ mm,   acceleration bounds: $A_x = A_y = 2000$ mm/s^2



left: smoothed corner; center: feedrate function; right: acceleration plot

achieves $\sim$ 30% reduction in overall cornering time

cross-coupled control of CNC machines

CNC machines have traditionally used independent axis controllers.
Contouring accuracy can be improved with cross–coupled controllers
(communicate information about path deviation between axis controllers)

At time $t_k = k\Delta t$, let

$\mathbf{p}$ = actual machine position (measured by encoders)

$\mathbf{r}(\xi_k)$ = reference point (commanded position) on curve

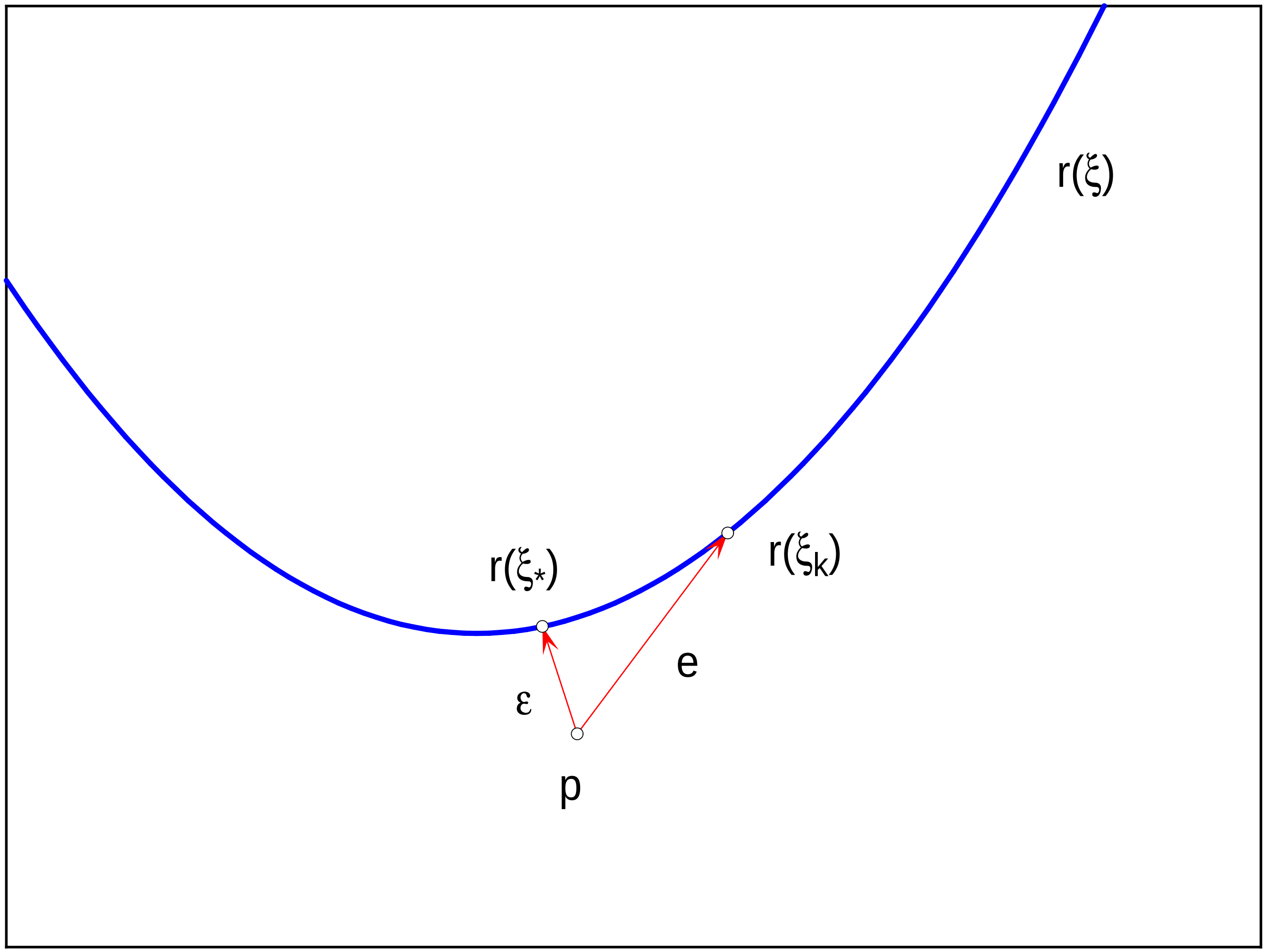
$\mathbf{e} = (e_x, e_y) = \mathbf{r}(\xi_k) - \mathbf{p}$ = position error vector

$\mathbf{r}(\xi_*)$ = footpoint (closest point) to $\mathbf{p}$ on $\mathbf{r}(\xi)$

$\boldsymbol{\epsilon} = (\epsilon_x, \epsilon_y) = \mathbf{r}(\xi_*) - \mathbf{p}$ = contour error vector

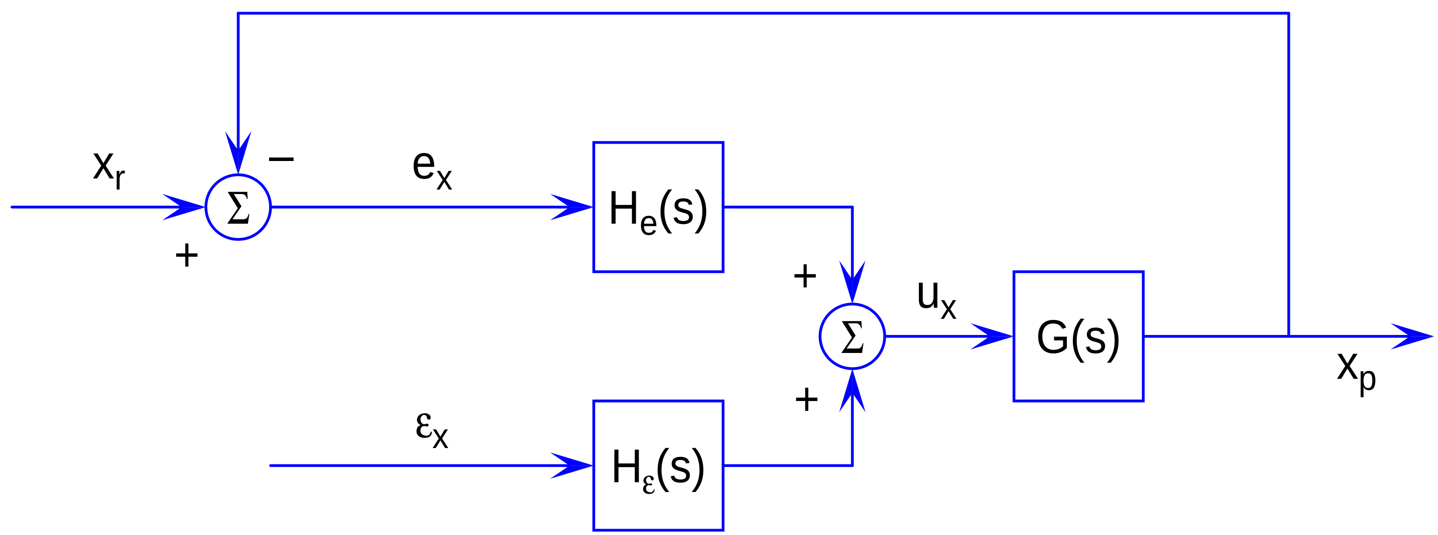
$\epsilon = |\boldsymbol{\epsilon}|$ = contour error of $\mathbf{p}$ with respect to $\mathbf{r}(\xi)$

write $\mathbf{e} = \mathbf{e}_{\parallel} + \mathbf{e}_{\perp}$ where “normal deviation” $\mathbf{e}_{\perp} = \boldsymbol{\epsilon}$ = contour error
(limits accuracy of the machined part), while “tangential deviation”
$\mathbf{e}_{\parallel} = \mathbf{e} - \mathbf{e}_{\perp}$ = feed error (only affects the overall machining time)



$\mathbf{r}(\xi_k) =$ reference point (commanded position) and $\mathbf{p} =$ actual position (measured by encoders) at $t_k = k\,\Delta t$. Then position error $\mathbf{e} = \mathbf{r}(\xi_k) - \mathbf{p}$ and contour error $\boldsymbol{\epsilon} = \mathbf{r}(\xi_*) - \mathbf{p}$, where $\mathbf{r}(\xi_*)$ is the *footpoint* of $\mathbf{p}$ on $\mathbf{r}(\xi)$.

cross-coupled controller block diagram

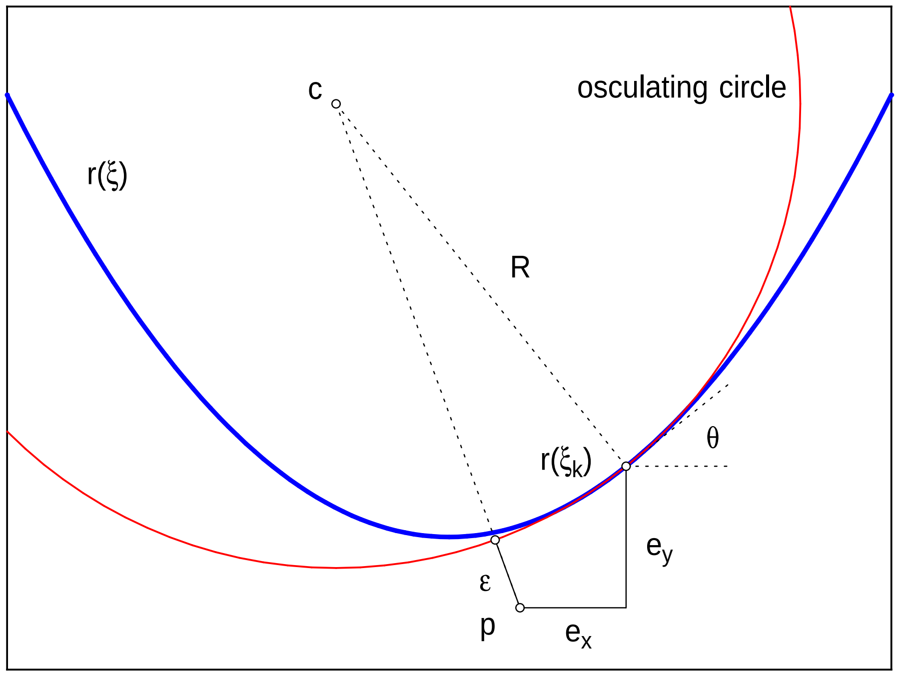


actuating signal for the x–axis is a combination of the position error e_x and contour error ϵ_x components, as modulated by individual (e.g., PID) controller transfer functions $H_e(s)$ and $H_\epsilon(s)$

need accurate & efficient algorithms for real–time computation of contour error ϵ w. r. t. curved path $\mathbf{r}(\xi)$ at $\sim$ 1–10 kHz servo sampling frequencies

contour error estimation using osculating circle

Y. Koren and C. C., Lo (1991), Variable–gain cross–coupling for contouring, *CIRP Annals* **40**, 371–374.



quasi–linear contour error estimate $\epsilon \approx -C_x e_x + C_y e_y$ by approximation of the curve with circle of curvature at $\mathbf{r}(\xi_k)$ — where the "variable gains"

$$C_x = \text{sign}(\rho) \left[\sin\theta - \frac{e_x}{2\rho} \right], \quad C_y = \text{sign}(\rho) \left[\cos\theta + \frac{e_y}{2\rho} \right]$$

depend on tangent angle θ and (signed) radius of curvature ρ $(R = |\rho|)$

accuracy of osculating-circle approximation

Taylor series expansion of $\mathbf{r}(\xi)$ for arc–length increment Δs

$$\Delta\mathbf{r} \;=\; \Delta s\;\mathbf{t} \;+\; \tfrac{1}{2}(\Delta s)^2\,\kappa\,\mathbf{n} \;+\; \tfrac{1}{6}(\Delta s)^3\,(\dot{\kappa}\,\mathbf{n} - \kappa^2\mathbf{t}) \;+\; \cdots$$

osculating circle $=$ first two terms, deviation $=$ cubic and higher terms

$$\frac{|\,\frac{1}{6}(\Delta s)^3\,(\dot{\kappa}\,\mathbf{n} - \kappa^2\mathbf{t})\,|}{|\,\Delta s\;\mathbf{t} \;+\; \frac{1}{2}(\Delta s)^2\,\kappa\,\mathbf{n}\,|} \;=\; \tfrac{1}{6}(\kappa\Delta s)^2\sqrt{\frac{1+\dot{\kappa}^2/\kappa^4}{1+\frac{1}{4}(\kappa\Delta s)^2}}$$

osculating circle is poor approximation of $\mathbf{r}(\xi)$ if above ratio not $\ll 1$
— occurs when curvature has large magnitude or varies rapidly

more importantly, $\Delta s \approx |\,\mathbf{r}(\xi_k) - \mathbf{p}\,|$ may be relatively large if controller tolerates significant steady–state error (e.g., P controller)
— method uses “wrong” circle of curvature if κ varies rapidly

exact contour error measurement

J. R. Conway, C. A. Ernesto, R. T. Farouki, and M. Zhang (2011), Performance analysis of cross–coupled controllers for CNC machines based upon precise real–time contour error measurement, *International Journal of Machine Tools and Manufacture*, to appear

$\mathbf{p} = (x_p, y_p)$ and degree n polynomial curve $\mathbf{r}(\xi) = (x(\xi), y(\xi))$, $\xi \in [\,0,1\,]$

$$\epsilon \;=\; \min_{\xi\,\in\,[\,0,1\,]} |\,\mathbf{p}-\mathbf{r}(\xi)\,| \;=\; \min_{0\le i\le N+1} |\,\mathbf{p}-\mathbf{r}(\xi_i)\,|$$

$\xi_0 = 0$, $\xi_{N+1} = 1$ and $\xi_1,\ldots,\xi_N$ are odd–multiplicity roots on $(0,1)$ of

$$F(x_p,y_p,\xi) \;=\; [\,x_p - x(\xi)\,]\,x'(\xi) \;+\; [\,y_p - y(\xi)\,]\,y'(\xi)$$

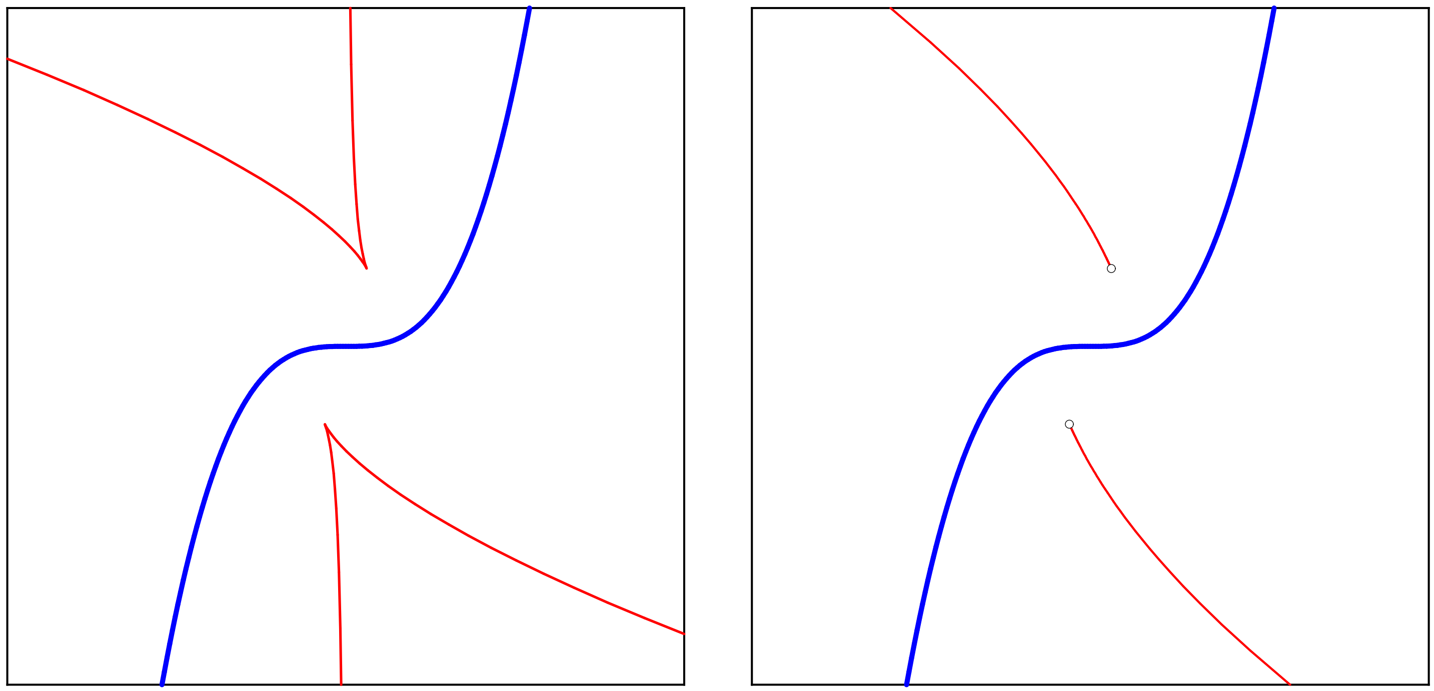
$F(x_p,y_p,\xi)$ is of odd degree $2n-1$ in ξ, so it has at least one real root

$\mathbf{r}(\xi_m)$ is called a footpoint of $\mathbf{p}$ on $\mathbf{r}(\xi)$ if $|\,\mathbf{p}-\mathbf{r}(\xi_i)\,|$ is minimum for $i=m$

if footpoint is unique, analytic continuation (e.g., predictor–corrector) method can be used to update it as $\mathbf{p}$ moves in small increments $\Delta\mathbf{p}$

not possible for locations of $\mathbf{p}$ with multiple footpoints, namely, $\mathbf{p}$ lies on evolute or self–bisector of $\mathbf{r}(\xi)$ — change in “identity” of footpoint occurs

locations of $\mathbf{p}$ with multiple footpoints



evolute (left) & self–bisector (right) for the cubic $\mathbf{r}(\xi) = (\xi, \xi^3)$

no analytic continuation of footpoint as $\mathbf{p}$ crosses these loci

no simple closed–form equation for the self–bisector of $\mathbf{r}(\xi)$

tracking all roots of $F(x_p, y_p, \xi) = 0$

simpler to track all (real & complex) roots as $\mathbf{p}$ moves, and select the real root ξ_m that minimizes $|\,\mathbf{p} - \mathbf{r}(\xi_k)\,|$

compute roots for initial location $\mathbf{p} = (x_0, y_0)$ then update by cubically–convergent Laguerre iteration as $\mathbf{p} \to \mathbf{p} + \Delta\mathbf{p}$

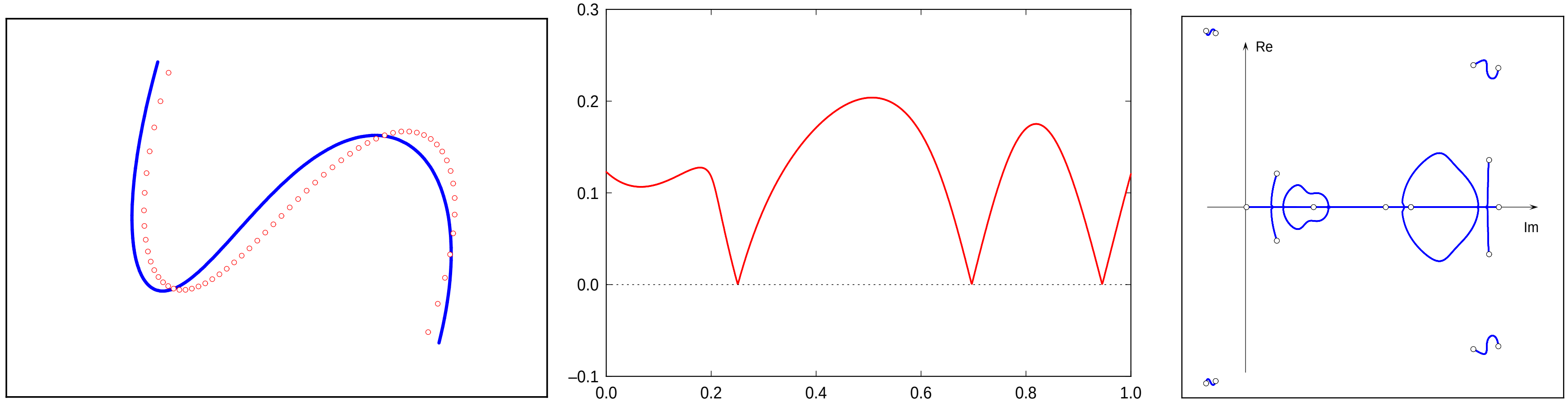
$$\xi^{(r+1)} \;=\; \xi^{(r)} \;-\; \frac{m f(\xi^{(r)})}{f'(\xi^{(r)}) \pm \sqrt{g(\xi^{(r)})}}$$

where $f(\xi) = F(x_p, y_p, \xi)$ and

$$g(\xi) \;=\; (m-1)[\,(m-1)f'^2(\xi) \;-\; m f(\xi) f''(\xi)\,]$$

for degree 5 curve, runs in real time on 300 MHz processor

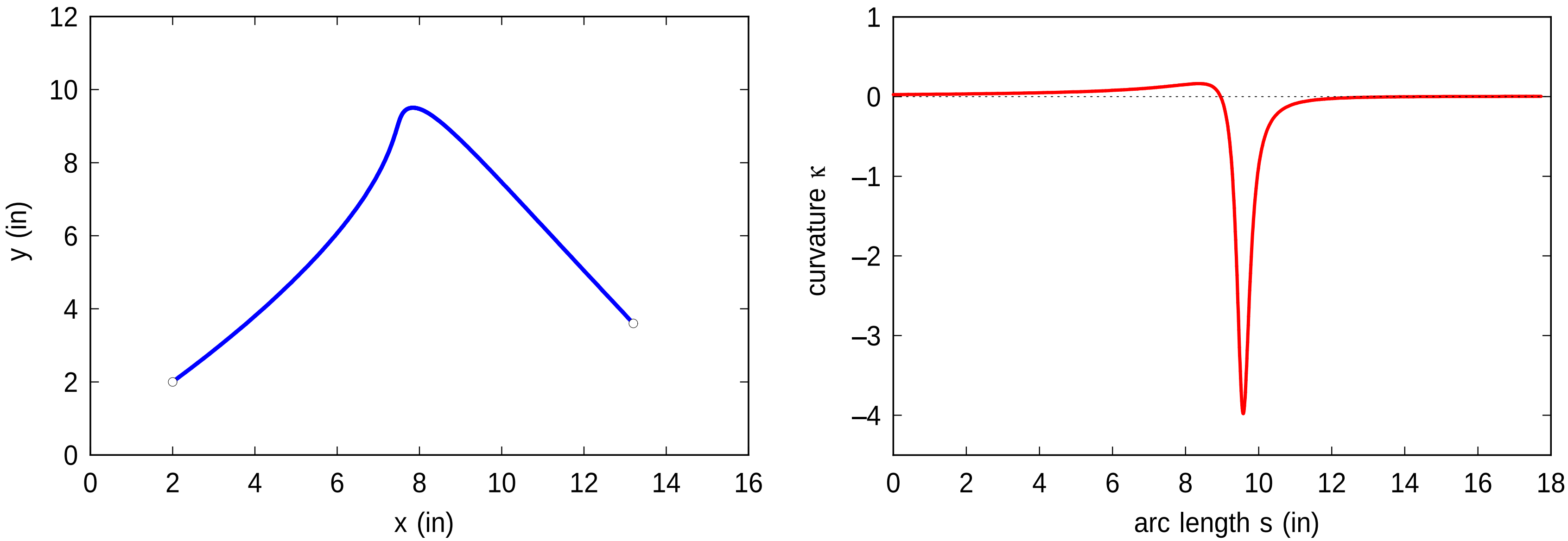
real-time contour error computation by Laguerre iteration



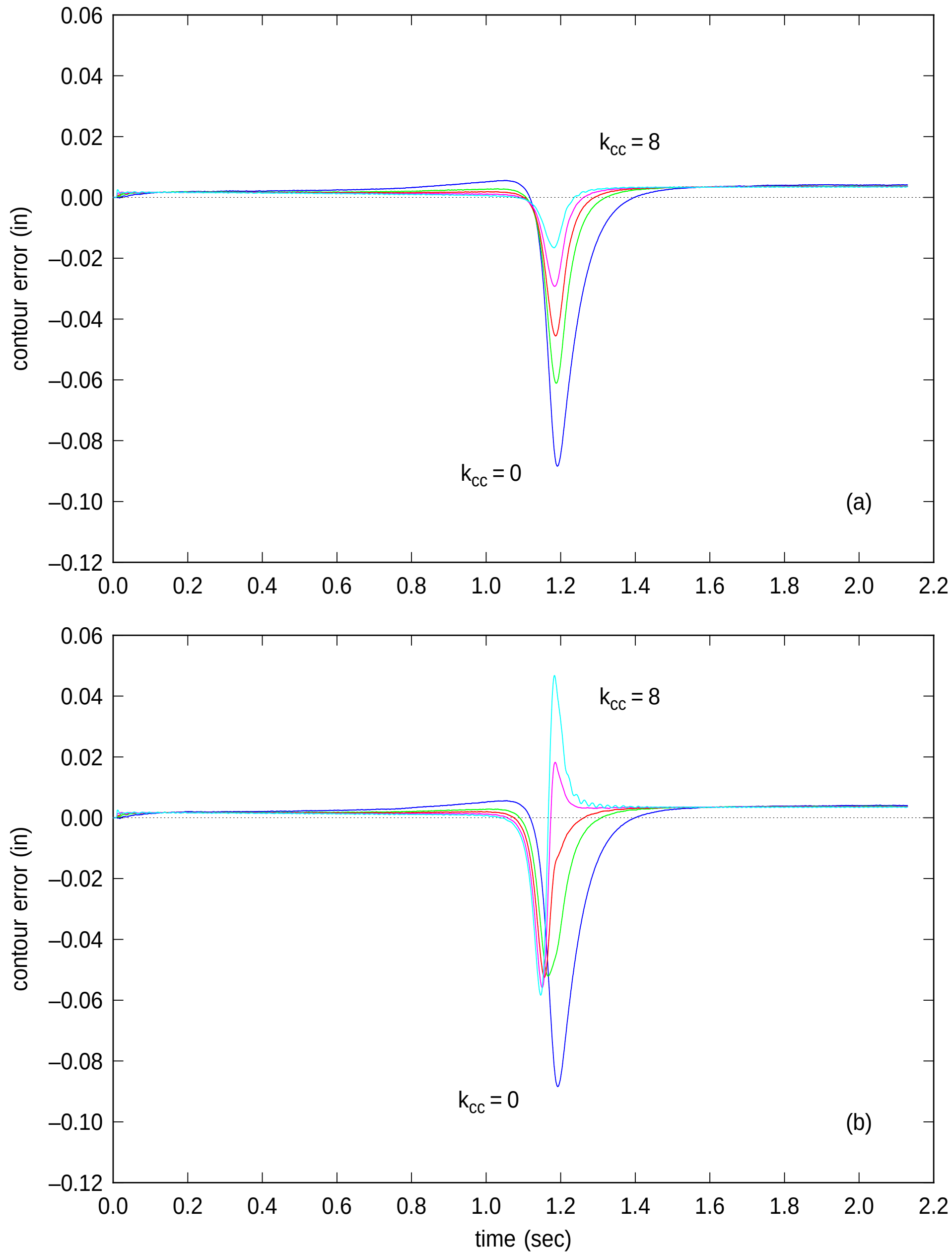
left: quintic curve $\mathbf{r}(\xi) = (x(\xi), y(\xi))$ with discrete sampling of machine positions $\mathbf{p} = (x_p, y_p)$; center: contour error of machine positions $\mathbf{p}$ with respect to $\mathbf{r}(\xi)$; right: variation of roots of the polynomial $F(x_p, y_p, \xi) = 0$ in the complex plane as $\mathbf{p}$ moves.

cross-coupled controller implemented using $H_\epsilon(s) = k_{cc}\, H_e(s)$, where $H_e(s)$ is a standard P or PI type controller

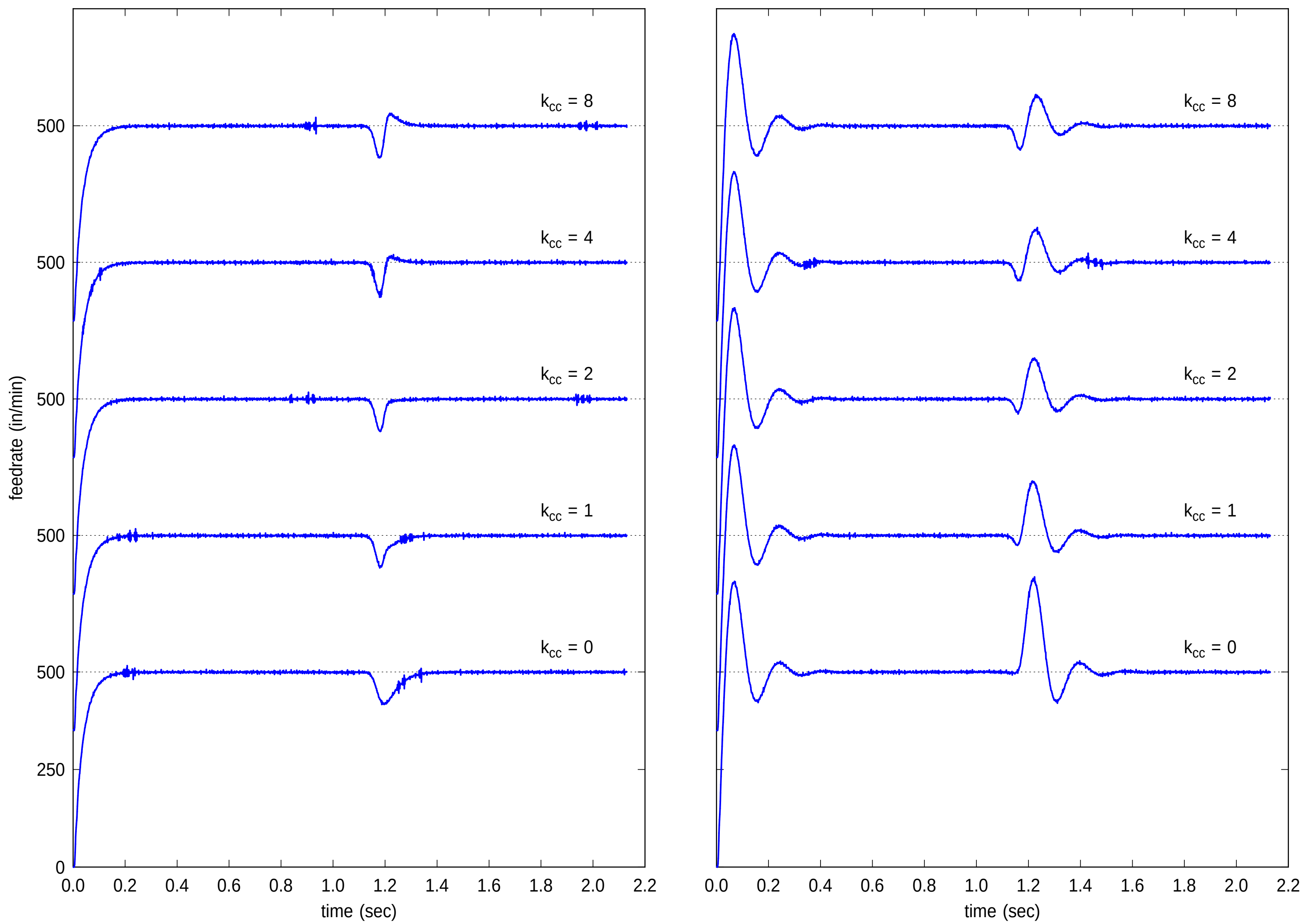
test curve for cross-coupled controller



left: PH quintic test curve used in cross–coupled controller experiments;
right: curvature profile — showing sharp curvature "spike" — of test curve



measured contour errors using exact computation (upper) and osculating circle approximation (lower) with cross–coupling gains $k_{cc} = 0, 1, 2, 4, 8$.



measured feedrate (for a 500 in/min command rate) with cross-coupled P (left) and PI (right) controllers, using exact contour error computation

conclusions for cross-coupled control

- feasibility of exact real-time contour error computation with modest (300 MHz) cpu & 1 kHz sampling frequency

- for P controller, exact computation gives systematic reduction of contour error relative to osculating-circle approximation, as relative gain k_{cc} is increased

- improvement for PI controller is less significant, since steady-state position error is largely suppressed, but feedrate fluctuations are much larger

- improvement in tracking accuracy most pronounced for paths with large curvature and rapid curvature variation

contour machining of free-form surfaces

Given parametric surface $\mathbf{r}(u,v)$ for $(u,v) \in [\,0,1\,] \times [\,0,1\,]$ and set Π of parallel planes with normal $\mathbf{N}$ and equidistant spacing Δ, planar sections of $\mathbf{r}(u,v)$ by planes of Π are the surface contours.

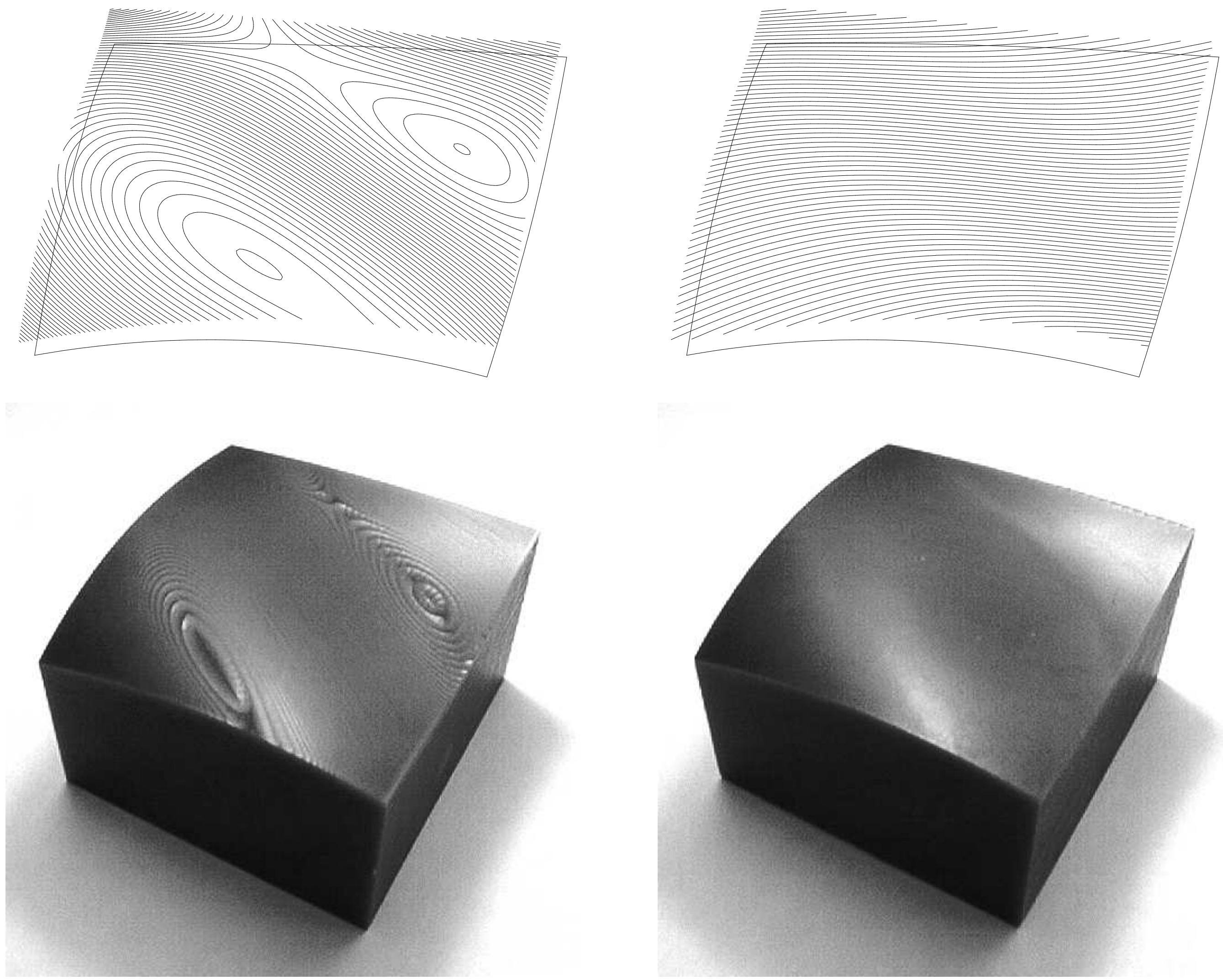
In contour machining, these surface contours define the contact curves of a spherical cutter with $\mathbf{r}(u,v)$. Spacing between adjacent contours is $\ell \approx \Delta/\sqrt{1-(\mathbf{N}\cdot\mathbf{n})^2}$, where surface normal $\mathbf{n}$ is defined by

$$\mathbf{n} \;=\; \frac{\mathbf{r}_u \times \mathbf{r}_v}{|\mathbf{r}_u \times \mathbf{r}_v|}\,, \qquad (u,v) \in [\,0,1\,] \times [\,0,1\,]\,.$$

For "best quality" contours — that minimize scallop height of machined surface between adjacent tool paths — we need to find orientation $\mathbf{N}$ of the planes Π that minimizes $\mathbf{N}\cdot\mathbf{n}$ for $(u,v) \in [\,0,1\,] \times [\,0,1\,]$.

In other words, $\mathbf{N}$ should be "as far as possible" from the set $\{\,\mathbf{n}\,\}$ of all normals to the surface $\mathbf{r}(u,v)$.

optimal section-plane orientation

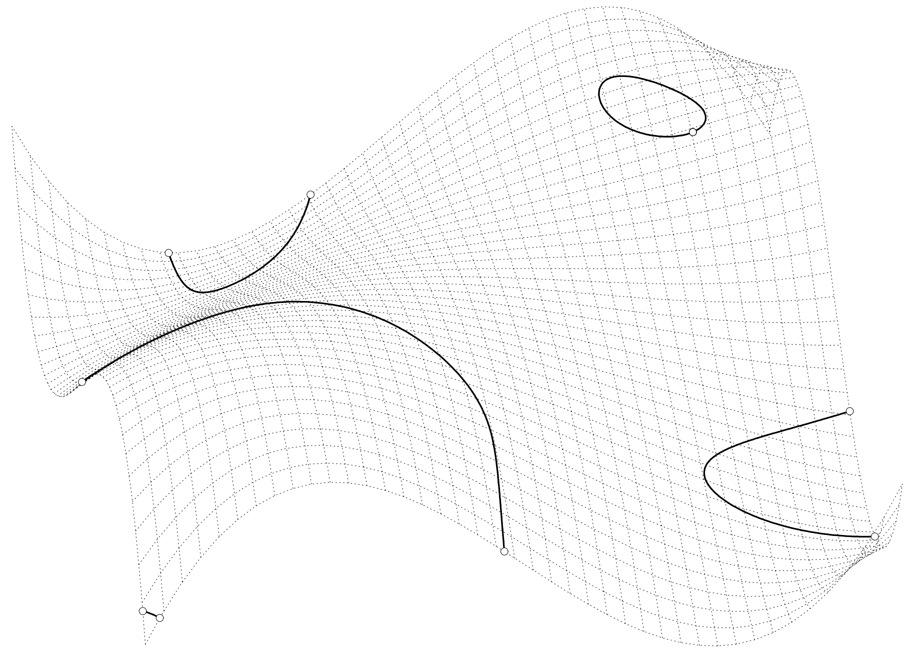


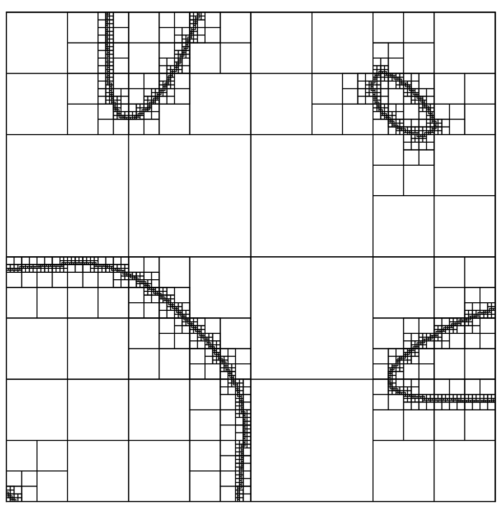
strategy for optimal orientation $\mathbf{N}$ of section planes

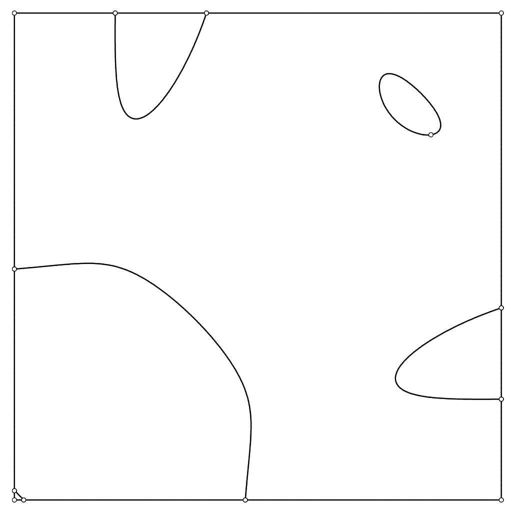
- set of normals $\{\mathbf{n}\}$ = Gauss map of surface, on unit sphere S^2
- Gauss map boundary $=$ subset of images of the parabolic lines (zero Guassian curvature) and patch boundaries on S^2
- symmetrize Gauss map by identifying opposed normals $\mathbf{n}, -\mathbf{n}$
- perform stereographic projection of Gauss map from S^2 to $\mathbb{R}^2$
- compute medial axis transform on *complement* of Gauss map
- center of largest circle in medial axis transform identifies vector $\mathbf{N}$ furthest from all normals $\mathbf{n}$ to $\mathbf{r}(u,v)$
- also applies to rapid prototyping / layered manufacturing processes

T. S. Smith, R. T. Farouki, M. al–Kandari, and H. Pottmann, Optimal slicing of free–form surfaces, *Computer Aided Geometric Design* **19**, 43–64 (2002)

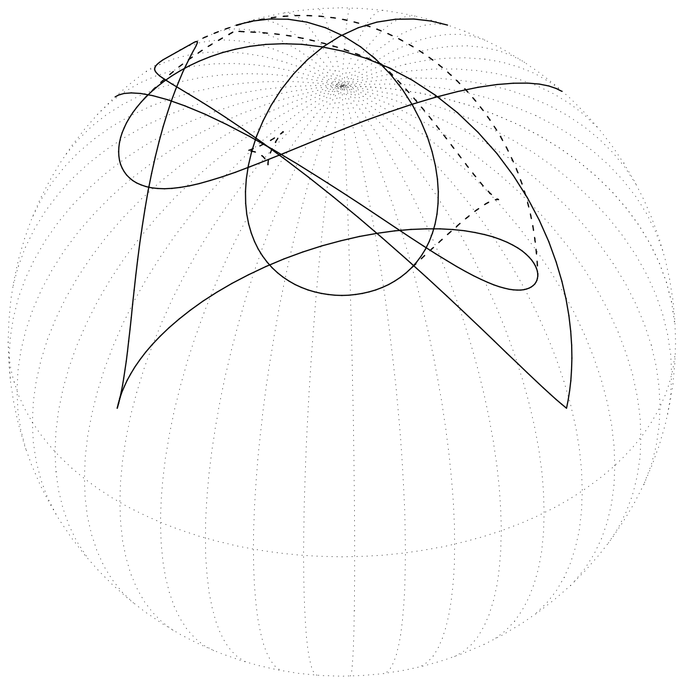
parabolic lines on free-form surfaces

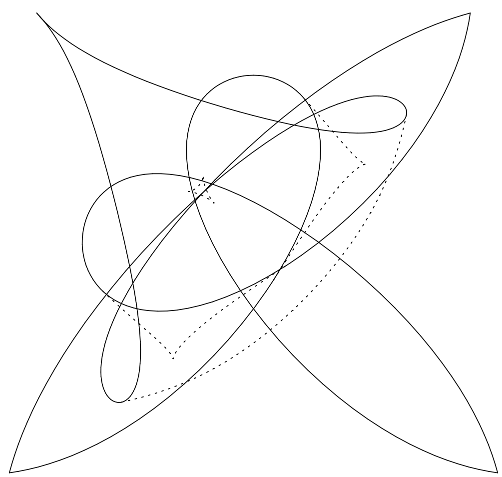


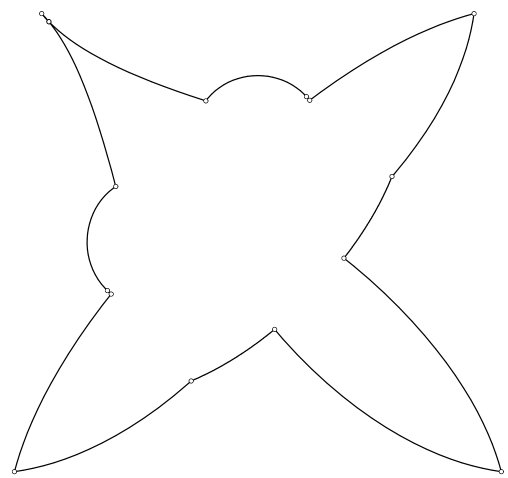




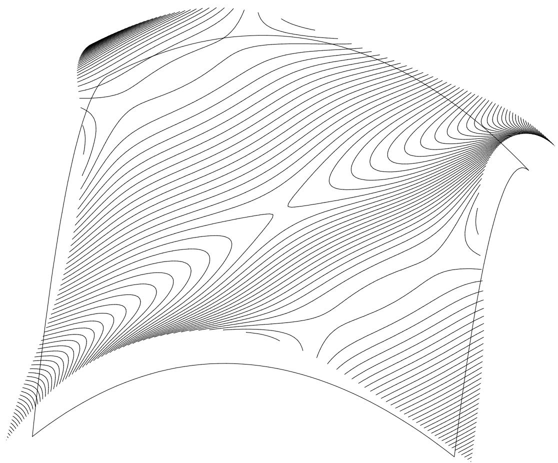
Gauss map computation for free-form surfaces

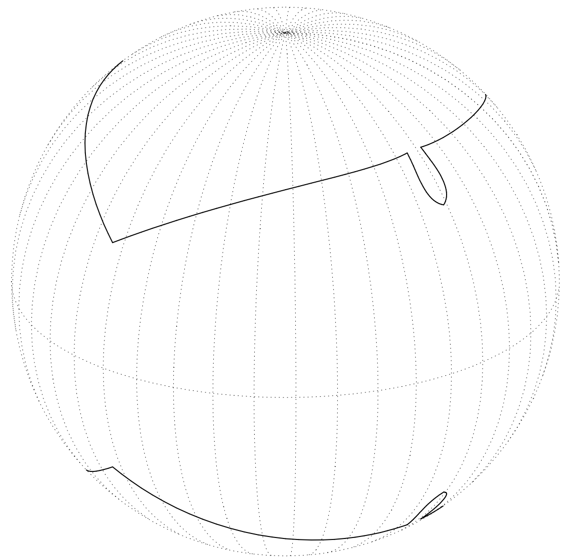


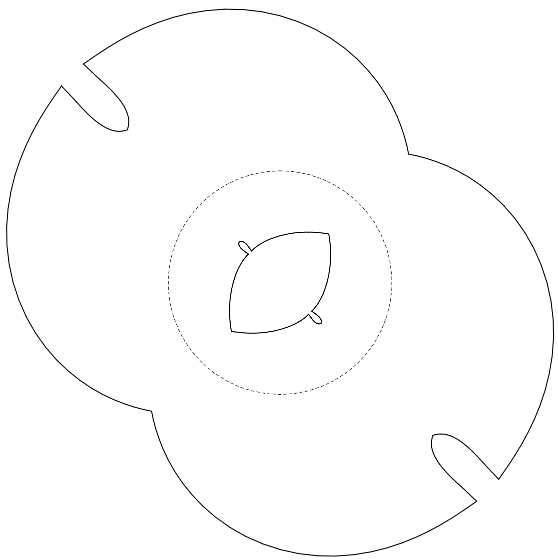


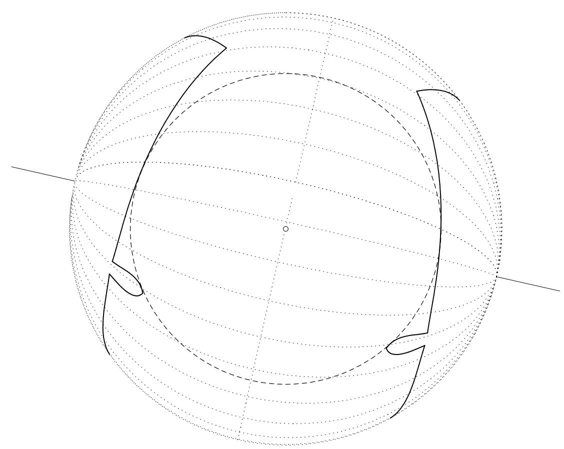


medial axis transform for complement of Gauss map









closure

- most CNC machines significantly under-perform in practice — control software, not hardware, is usually the limiting factor

- Pythagorean-hodograph curves ideally suited to CNC machining with feedrates dependent upon time, arc length, or curvature — analytic curve interpolators offer smoother and more accurate realization of high feedrates and acceleration rates than G codes

- high–speed cornering subject to axis acceleration bounds

- PH curves amenable to solution of inverse dynamics problems to compensate for inertia & damping of machine axes

- cross–coupled control based on exact real–time contour error computation improves tracking accuracy with P controller

- optimal orientation for contour machining of free–form surfaces