Int J Adv Manuf Technol (2005) 26: 1388–1396
DOI 10.1007/s00170-004-2119-2

ORIGINAL ARTICLE

Chih-Wei Lue · Yuan-Ming Cheng · Jih-Hua Chin

System structure and contour tracking for a hybrid motion platform

Received: 6 October 2003 / Accepted: 28 January 2004 / Published online: 8 December 2004

Abstract This work investigates the spatial contour tracking on a new kind of motion platform that is constructed by extending the parallel platform with a proven planar motion table. Multi-axis cross-coupled tracking control schemes were developed on the basis of this hybrid structured Stewart platform, which formed a 5-axis working machine. A framework for the functions of this system was established. Roadmap for further development that leads to a highly versatile and efficient precision motion platform was made.

Keywords 5-axis machine tool · Contouring · Cross-coupled tracking · Stewart platform · Trajectory tracking

C.-W. Lue · Y.-M. Cheng · J.-H. Chin (✉)
Department of Mechanical Engineering,
National Chiao Tung University,
1001, Tashue Road, Hsinchu 300, Taiwan, R.O.C.
E-mail: jhchin@mail.nctu.edu.tw
Tel.: +886-3-5712121 ext. 55105
Fax: +886-3-5727485

1 Introduction

A parallel machine tool is featured by its high structural rigidity and precision, while traditional serial machine tool is proven for its mature performance of continuous path contouring. But both suffer from inherited disadvantages. Serial structure has long load-bearing chains and hence limited stiffness and dynamics, while the parallel structure of a Stewart platform only prevails in a limited motion space that contains threatening singular points.

The purpose of this work is to pioneer a novel hybrid platform that integrates the advantages of both parallel and serial machines. The framework for the conceived hybrid motion platform was established and the multi-axis 3-D precision contour tracking was investigated. Figure 1 shows the novel system studied in this work.

In the trajectory tracking, Koren [1] first proposed the cross-coupled biaxial control, which has become the state-of-the-art in precision contour tracking [2–7]. Chin and Lin [8] enhanced the cross-coupling operation with a velocity precompensation. The

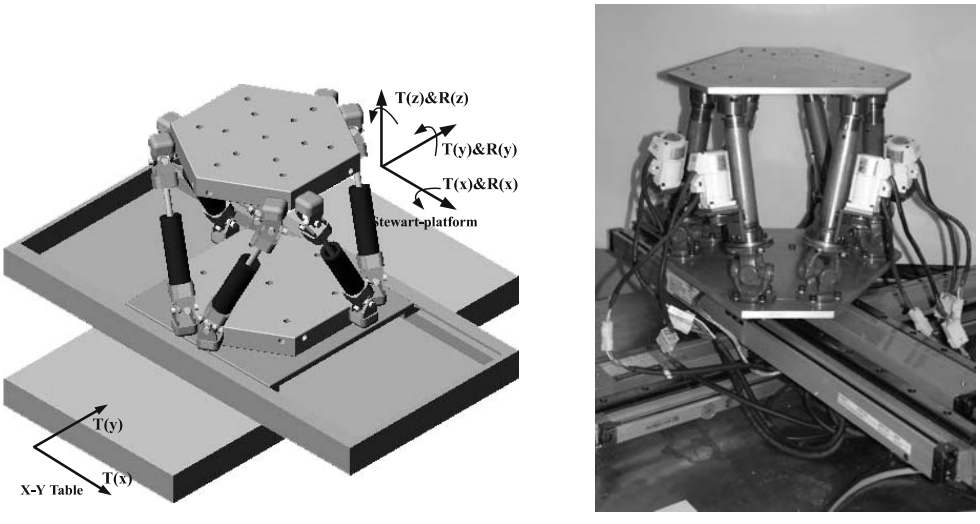


Fig. 1. Proposed hybrid motion platform. *Left*: computer model, *right*: prototype

precompensated cross-coupled control proved to have the best contour precision in circumstances of high feed rates and great trajectory curvatures [9, 10].

In the field of Stewart platform, mechanism design and analysis [11–13], accuracy analyses [14–17], workspace and singular point [18–21] were investigated. Portman et al. [22] constructed a 6-DOF parallel platform, which enabled displacements in the micron and submicron ranges.

Since the proposed novel system works like a 5-axis machine, some latest development in 5-axis concerning kinematics [23], path generation [24–26] and real-time interpolation [27, 28] were of interest.

2 System construction

Figure 2 shows the system block diagram of the constructed motion platform. The upper part of Fig. 2 that produces (x_a, y_a) represents the control of x-y table while the lower part that pro-

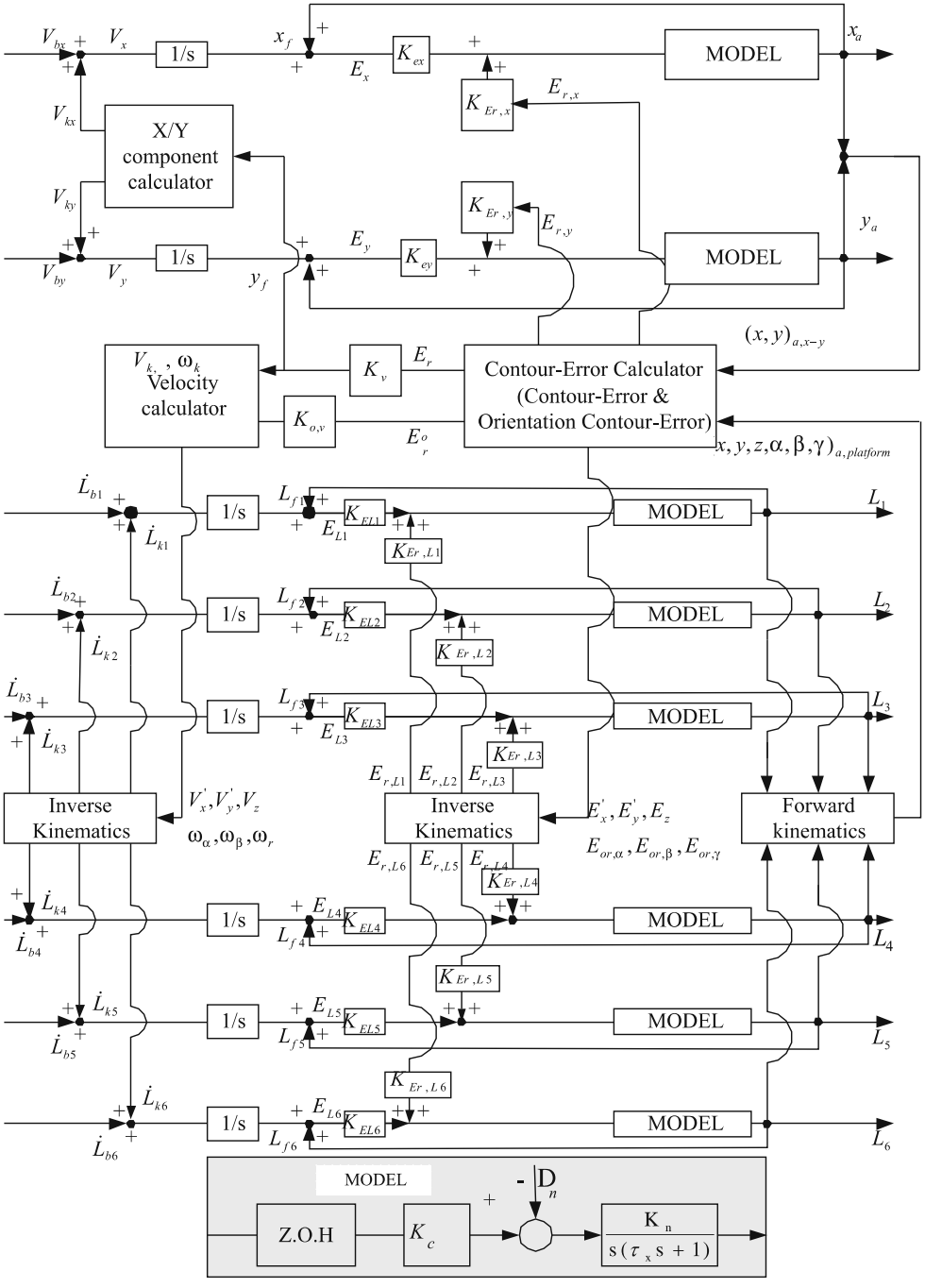


Fig. 2. Block diagram of the MCCPM controlled hybrid motion platform

duces ($L_1 \cdots L_6$) represents the control of leg lengths of parallel platform. The cross-coupled, precompensation control (CCPM), mainly planar techniques, was chosen and extended to multi-axis CCPM (MCCPM) in this work.

2.1 Six-DOF Stewart platform

2.1.1 Inverse kinematics

Figure 3 shows the upper and the lower platform of a Stewart platform. The six joint locations of the upper platform in the base coordinates can be expressed as:

$$\left[{}^{b}\mathrm{E}_i\right] = \left[{}^{b}\mathrm{T}_e\right] * \left[{}^{e}\mathrm{E}_i\right], \quad i = 1 \sim 6, \tag{1}$$

where $[{}^{e}\mathrm{E}_i] = [L_e C\phi_i, L_e S\phi_i, 0, 1]^T$, $i = 1 \sim 6$ and $[{}^{b}\mathrm{T}_e]$ is transformation matrix.

The six joint locations of base platform in the base coordinates are

$$\left[{}^{b}\mathrm{B}_i\right] = [L_b C\theta_i, L_b S\theta_i, 0]^T, \quad i = 1 \sim 6. \tag{2}$$

The leg lengths can be obtained by calculating:

$$|L_i| = \left|{}^{b}\mathrm{E}_i - {}^{b}\mathrm{B}_i\right|, \quad i = 1 \sim 6. \tag{3}$$

2.1.2 Forward kinematics

Contrary to the simple calculation for leg lengths, six nonlinear equations are to be solved to get the required centroid $(x, y, z; \alpha, \beta, \gamma)$ for a given set of $L_{i=1\sim6}$:

$$F_i\,(x, y, z, \alpha, \beta, \gamma) = \left[\left({}^{b}\mathrm{E}_i - {}^{b}\mathrm{B}_i\right)_x^2 + \left({}^{b}\mathrm{E}_i - {}^{b}\mathrm{B}_i\right)_y^2 + \left({}^{b}\mathrm{E}_i - {}^{b}\mathrm{B}_i\right)_z^2\right] - L_i^2. \tag{4}$$

2.1.3 Singular points and work space

The advantages of parallel platform are won at the cost of sophisticated determination and exclusion of singular points. This becomes more difficult when a continuous spatial trajectory is to be tracked. The finding of singular points can be performed by differentiating equations $F_i(x, y, z, \alpha, \beta, \gamma)$ as follows:

$$\frac{dF_i}{dt} = \frac{\partial F_i}{\partial L}\frac{\partial L}{\partial t} + \frac{\partial F_i}{\partial P}\frac{\partial P}{\partial t} = W(L)\,\dot{L} + S(P)\,\dot{P} = 0. \tag{5}$$

Case 1: Inverse kinematics singularities

When $\det W(L) = 0$, $\dot{P} = 0$ satisfies the equation $dF_i/dt = 0$, which corresponds to the workspace boundaries.

Case 2: Forward kinematics singularities

When $\det S(P) = 0$, $\dot{L} = 0$ satisfies the equation $dF_i/dt = 0$, which corresponds to the singular points.

2.2 Cross-coupled precompensation method (CCPM)

CCPM [8–10] was an extended version of cross-coupled control. The motion platform shown in Fig. 2 is already in CCPM control, which can be identified by the blocks "contour error calculator" and the "velocity calculator."

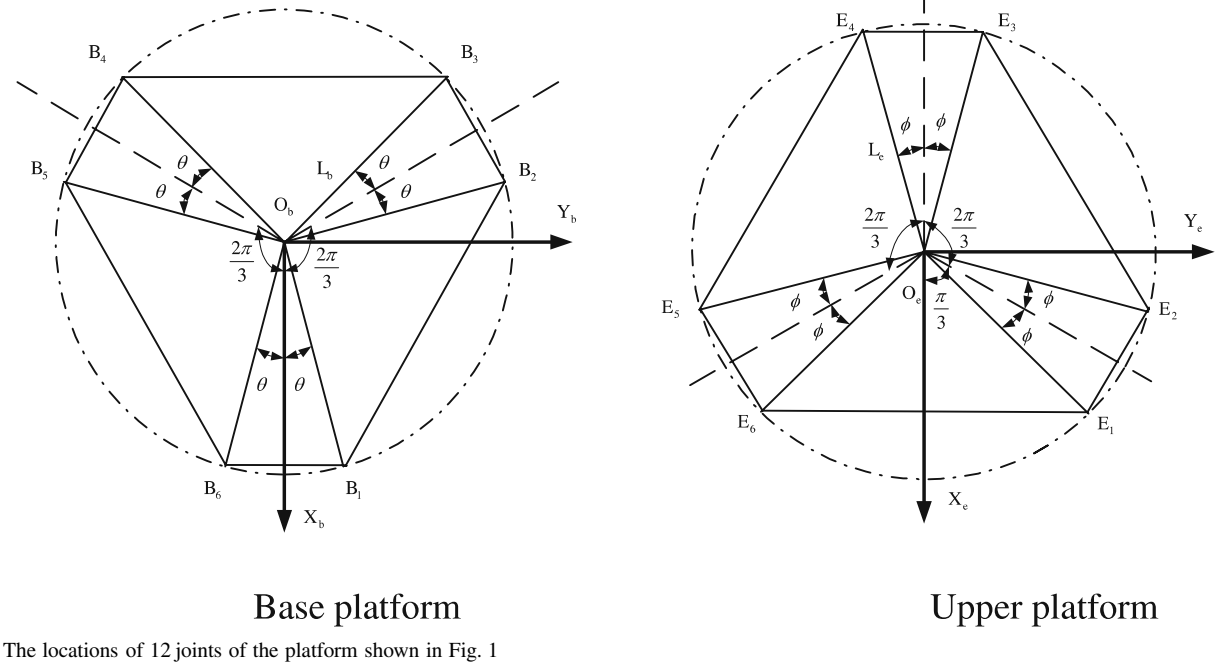


Fig. 3. The locations of 12 joints of the platform shown in Fig. 1

The speed precompensation requires that the feeding velocity be modified as:

$$\overrightarrow{V} = \overrightarrow{V_t} + \overrightarrow{V_k} . \tag{6}$$

3 New system developments

Based upon the knowledge stated above, the following function groups were developed.

3.1 Trajectory planning in the sense of 5-axis motion

3.1.1 Trajectory construction

A continuous spatial surface $S(u, v)$ is shown in Fig. 4, upon which a trajectory $P_i(u, v_c)$ can be constructed by setting v constant and u variable.

If bi-cubic patch [29] is chosen to construct the surface, then

$$S(u, v) = \sum_{i=0}^{3}\sum_{j=0}^{3} a_{ij} u^i v^j , \quad (0 \leqslant u \leqslant 1, 0 \leqslant v \leqslant 1) . \tag{7}$$

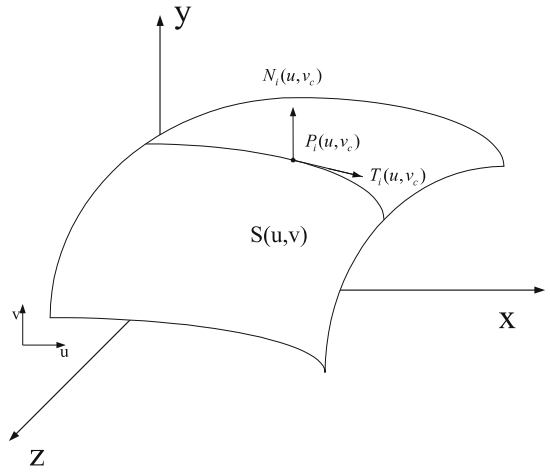


Fig. 4. A trajectory on surface

The normal to the surface at the machining point can be obtained by:

$$\left[n_x, n_y, n_z\right] = \left(\frac{\partial S}{\partial v} \times \frac{\partial S}{\partial u}\right) / \left(\left|\frac{\partial S}{\partial v} \times \frac{\partial S}{\partial u}\right|\right) .$$

The tool position and orientation $(P_t, O_t)_M^T$ correspond to $(x_t, y_t, z_t; 0, 0, -1)^T$ in base coordinates. The trajectory and its normal are denoted by $[S(u, v); n(u, v)]_{effector}^T$, which can be transformed to base coordinates as follows:

$$[S(u, v); n(u, v)]_M^T = [T.M.]_M^{base} [S(u, v); n(u, v)]_{base}^T , \tag{8}$$

where $[S(u, v)]_{base}^T = [T.M.]_{base}^{effector} [S(u, v)]_{effector}^T ,$

and $[n(u, v)]_{base}^T = [R.M.]_{base}^{effector} [n(u, v)]_{effector}^T .$

The transformation leads to α, β and γ as well as the following position relationships:

$$\begin{aligned} x_{xy} + x_c &= x_t - \left(S_x n_1 + S_y o_1 + S_z a_1\right) , \\ y_{xy} + y_c &= y_t - \left(S_x n_2 + S_y o_2 + S_z a_2\right) , \\ z_c &= z_t - \left(S_x n_3 + S_y o_3 + S_z a_3\right) . \end{aligned} \tag{9}$$

The demand leg lengths for a Stewart platform can be obtained by inverse kinematics:

$$L_i = 1 \sim 6 = \text{Inv-kinematics}\ (x_c, y_c, z_c, \alpha, \beta, \gamma) . \tag{10}$$

3.1.2 Trajectory generator (real-time interpolator)

Equations 7, 8 are too complicated to be used in real-time path generation. Instead, the trajectory information $[S(u, v); n(u, v)]$ could be mapped onto two separate sets of position $P_i(u, v_c)$ and orientation $O_i(u, v_c)$ by Hermite curves and be stored in memory as a database. Figure 5 describes the mapping of centroid of the upper platform.

Hermite curves:

$$P(x) = [F_i(x)] \begin{bmatrix} P_0 & P_1 & P_0' & P_1' \end{bmatrix}^T , \quad (0 \leqslant x \leqslant 1) . \tag{11}$$

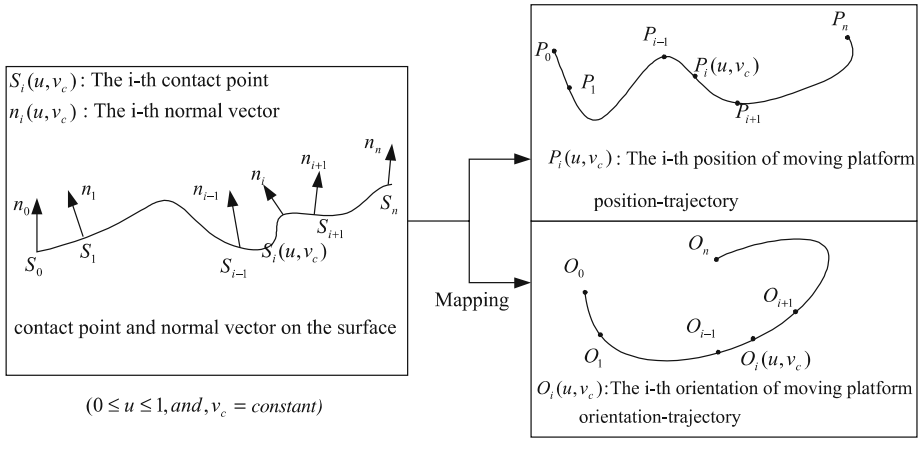


Fig. 5. The mapping of trajectory to position and orientation sets

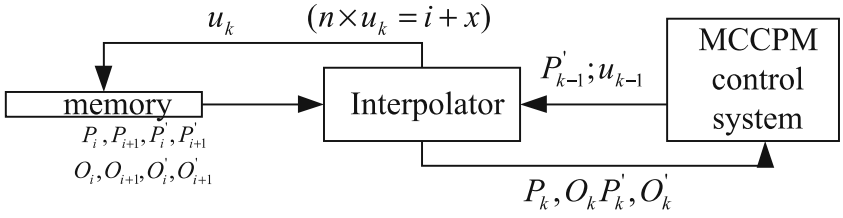


Fig. 6. The flowchart of real-time trajectory generation

Tangent vector is obtained as

$$P_k'(x) = \left[\frac{\partial F_k(x)}{\partial x}\right]\begin{bmatrix} P_0 & P_1 & P_0' & P_1' \end{bmatrix}^T, \quad (0 \leqslant x \leqslant 1). \quad (12)$$

Figure 6 describes the operation of the real-time trajectory generation. At each sampling time the MCCPM control provides P'_{k-1} and u_{k-1} to interpolator, which calculates parameter u_k according to Eq. 13 [10]. Information P_i, P_{i+1}, P'_i, P'_{i+1}, Q_i, Q_{i+1}, Q'_i, Q'_{i+1} corresponding to u_k are then recalled from the database and are used by interpolator to produce the trajectory information P_k, O_k, P'_k, O'_k according to Eqs. 11, 12:

$$u_k = u_{k-1} + \frac{V_b T}{\sqrt{{x'_{k-1}}^2 + {y'_{k-1}}^2 + {z'_{k-1}}^2}}, \quad (13)$$

where V_b is the feedrate.

3.2 Multi-axis trajectory tracking

3.2.1 Contour errors

Position control is the basics in CNC tracking. When tracking continuous contour the contour error become a primary target to be diminished.

In Fig. 7, the distance E between the actual position P_a and the desired position P_i is position error. The shortest distance between the actual position and the desired contour is contour error E_r, which can be found to be the following:

$$E_r = \begin{bmatrix} E_x, E_y, E_z \end{bmatrix}^T = \left[E - \left(E \cdot \overline{V}\right)\overline{V}\right] * \left(\vec{i}, \vec{j}, \vec{k}\right)^T. \quad (14)$$

Analogously to the trajectory contour, as shown in Fig. 8, an "orientation contour" E_{or} can be built in α, β, γ coordinates:

$$E_{or} = \begin{bmatrix} E_{or,\alpha}, E_{or,\beta}, E_{or,\gamma} \end{bmatrix}^T = \left[E_o - \left(E_o \cdot \overline{\omega}\right)\overline{\omega}\right] * \left(\vec{\alpha}, \vec{\beta}, \vec{\gamma}\right)^T. \quad (15)$$

The trajectory and orientation contour error, respectively, can be obtained as

$$\begin{pmatrix} E_{rx} \\ E_{ry} \\ E_{rz} \end{pmatrix} = \begin{pmatrix} \overline{V_y}\left(E_x\overline{V_y} - E_y\overline{V_x}\right) + \overline{V_z}\left(E_x\overline{V_z} - E_z\overline{V_x}\right) \\ \overline{V_x}\left(E_y\overline{V_x} - E_x\overline{V_y}\right) + \overline{V_z}\left(E_y\overline{V_z} - E_z\overline{V_y}\right) \\ \overline{V_x}\left(E_z\overline{V_x} - E_x\overline{V_z}\right) + \overline{V_y}\left(E_z\overline{V_y} - E_y\overline{V_z}\right) \end{pmatrix} \quad (16)$$

$$\begin{pmatrix} E_{or,\alpha} \\ E_{or,\beta} \\ E_{or,\gamma} \end{pmatrix} = \begin{pmatrix} \overline{\omega_\beta}\left(E_{o,\alpha}\overline{\omega_\beta} - E_{o,\beta}\overline{\omega_\alpha}\right) + \overline{\omega_\gamma}\left(E_{o,\alpha}\overline{\omega_\gamma} - E_{o,\gamma}\overline{\omega_\alpha}\right) \\ \overline{\omega_\alpha}\left(E_{o,\beta}\overline{\omega_\alpha} - E_{o,\alpha}\overline{\omega_\beta}\right) + \overline{\omega_\gamma}\left(E_{o,\beta}\overline{\omega_\gamma} - E_{o,\gamma}\overline{\omega_\beta}\right) \\ \overline{\omega_\alpha}\left(E_{o,\gamma}\overline{\omega_\alpha} - E_{o,\alpha}\overline{\omega_\gamma}\right) + \overline{\omega_\beta}\left(E_{o,\gamma}\overline{\omega_\beta} - E_{o,\beta}\overline{\omega_\gamma}\right) \end{pmatrix} \quad (17)$$

Among these six error components E_{rx}, E_{ry} shall be distributed to the x-y table and Stewart platform according to some algorithms or optimization process to be developed in the future. The Stewart platform determines the six error components, $\left(E'_{rx}, E'_{ry}, E_{rz}, E_{or,\alpha}, E_{or,\beta}, E_{or,\gamma}\right)$, such that these components must be the inverse of their leg-length-referred counterparts:

$$E_{L_i=1\sim6} = \text{Inv-Kinematics}\left(E'_{rx}, E'_{ry}, E_{rz}, E_{or,\alpha}, E_{or,\beta}, E_{or,\gamma}\right), \quad (18)$$

$$E_{rx(X-Y)} = E_{rx} - E'_{rx}; \quad E_{ry(X-Y)} = E_{ry} - E'_{ry}. \quad (19)$$

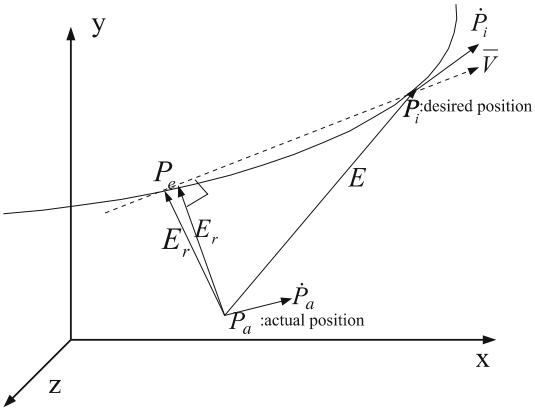


Fig. 7. Position error and contour error

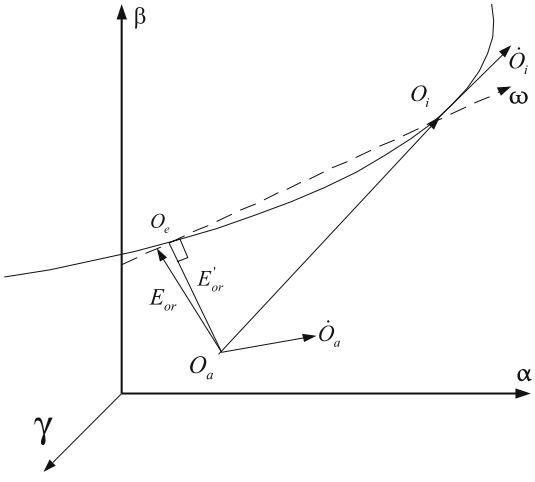


Fig. 8. "Orientation contour" error

3.2.2 Precompensation

The concept of precompensation is a velocity modification, which gives the machine a better adaptation to higher trajectory curvature.

Based on the experience in [9], better results could be achieved by using

$$\overrightarrow{V} = \begin{bmatrix} V_x & V_y & V_z \end{bmatrix}^T = V_t \overrightarrow{t} + K_v \overrightarrow{E_r} + K_i \int_0^t \overrightarrow{E_r} dt \,. \tag{20}$$

Similarly, the precompensation of angular velocity can be extended to

$$\overrightarrow{\omega} = \begin{bmatrix} \varpi_\alpha & \varpi_\beta & \varpi_\gamma \end{bmatrix}^T = \omega_b \overrightarrow{t0} + K_{o,v} \overrightarrow{E_{o,r}} + K_{o,i} \int_0^t \overrightarrow{E_{o,r}} dt \,. \tag{21}$$

The velocity components V_x, V_y shall be distributed to x-y table ($V_{x(X-Y)}$, $V_{y(X-Y)}$) and Stewart platform (V'_x, V'_y). In order to execute the required velocity precompensation by Stewart platform, the following inverse kinematics is necessary:

$$\dot{L}_{i=1\sim6} = \text{Inv-Kinematics} \left(V'_x, V'_y, V_z, \omega_a, \omega_b, \omega_\gamma \right) . \tag{22}$$

3.2.3 Coordination between the Stewart platform and the x-y table

The x-y table provides additional degrees of freedom that can be used to detour singular points or pursue optimization of motion. Assume that at $(n-1)$ discrete time point the centroid of platform is at $P_{a,n-1} = [x, y, z, \alpha, \beta, \gamma]_{a,n-1}$, and the command feed at time n is ΔP_n among which the components $\mu_x \Delta x_n$, $\mu_y \Delta y_n$ shall be assigned to the Stewart platform where μ_x, μ_y are the ratios of distribution. This makes the distributed feed of the Stewart platform and x-y table, respectively, at time n to be as follows:

$$\Delta P_{s,n} = \begin{bmatrix} \mu_x \Delta x_n - x_{a,n-1}, \mu_y \Delta y_n - y_{a,n-1}, \Delta z_n, \\ \Delta\alpha_n, \Delta\beta_n, \Delta\gamma_n \end{bmatrix}, \tag{23}$$

$$\Delta P_{xy,n} = \begin{bmatrix} (1-\mu_x)\,\Delta x_n + x_{a,n-1}, \left(1-\mu_y\right)\Delta y_n + y_{a,n-1} \end{bmatrix} . \tag{24}$$

The feeds of Stewart platform in terms of axis movement are

$$\begin{bmatrix} \Delta L_{i,n} \end{bmatrix} = \left[J \left(P_{a,n-1} + \frac{\Delta P_{s,n}}{2} \right) \right] * \begin{bmatrix} \Delta P_{s,n} \end{bmatrix}_{i=1\sim6} ,$$

$$\text{and } P_{s,n} = P_{a,n-1} + \Delta P_{s,n} \,. \tag{25}$$

Equation 25 issues the command feed of each axis for the Stewart platform:

$$\begin{aligned} \Delta L_{i,n} = {} & (a_i x + b_i) \left(\mu_x \Delta x_n - x_{a,n-1} \right) \\ & + (c_i y + d_i) \left(\mu_y \Delta y_n - y_{a,n-1} \right) + e_i \Delta z_n \\ & + (f_i y + g_i)\,\Delta\alpha_n + (h_i x + i_i y + j_i)\,\Delta\beta_n \\ & + (k_i x + l_i y + m_i)\,\Delta\gamma_n \,, \end{aligned} \tag{26}$$

where $x = x_{a,n-1} + (\mu_x \Delta x_n - x_{a,n-1})/2$;
$y = y_{a,n-1} + (\mu_y \Delta y_n - y_{a,n-1})/2$.

It gives

$$\Delta L_{i,n} = A_i \mu_x^2 + B_i \mu_y^2 + C_i \mu_x + D_i \mu_y + E_i \,, \quad (i = 1 \sim 6) \,. \tag{27}$$

The stroke of motion axes of Stewart platform can be written as

$$\sum_{i=1}^{6} L_{i,n} = A\mu_x^2 + B\mu_y^2 + C\mu_x + D\mu_y + E \,, \tag{28}$$

$$\sum_{i=1}^{6} L_{i,n} = A\,(\mu_x + C/2A) + B\left(\mu_y + D/2B\right) + E' \geqslant E' \,. \tag{29}$$

Substituting $\mu_x = -C/2A$, $\mu_y = -D/2B$ into Eqs. 28, 29 yields the respective command feed.

3.2.4 Multi-axis CCPM

The extension of CCPM to multi-axis (MCCPM) is featured by generating tracking command for x-y axis and the axes of Stewart platform as follows:

$$x_{f(X-Y)}(n) = x_{f(X-Y)}(n-1) + TV_{x(X-Y)} \,, \tag{30}$$

$$y_{f(X-Y)}(n) = y_{f(X-Y)}(n-1) + TV_{y(X-Y)} \,, \tag{31}$$

$$L_{f,i}(n) = L_{f,i}(n-1) + T\dot{L}_i \,, \quad i = 1 \sim 6 \,. \tag{32}$$

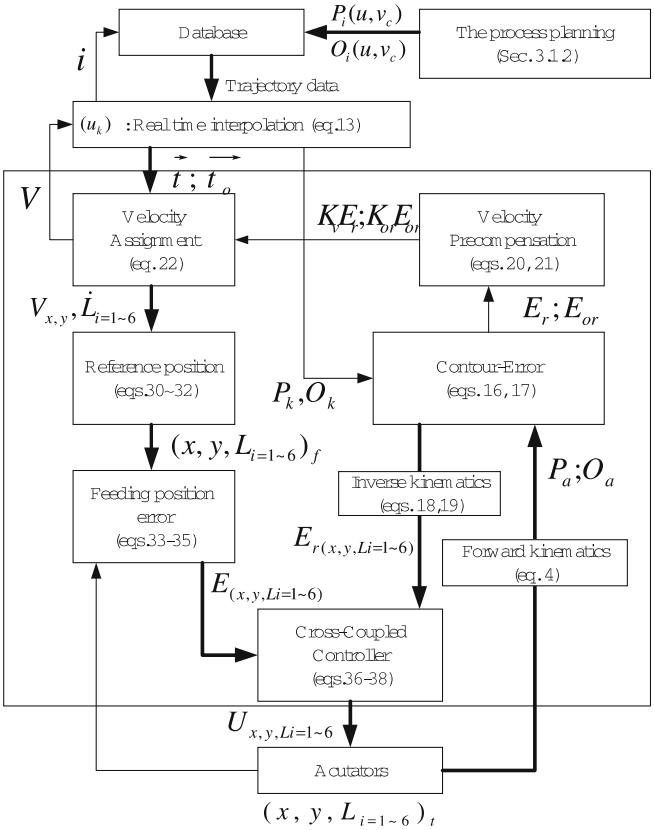


Fig. 9. The hybrid platform under multi-axis CCPM

The feeding errors of respective motion axis are calculated by

$$E_{x(X-Y)}(n) = E_{x(X-Y)}(n-1) + \left[x_{f(X-Y)}(n) - X_{t(X-Y)}(n)\right], \quad (33)$$

$$E_{y(X-Y)}(n) = E_{y(X-Y)}(n-1) + \left[y_{f(X-Y)}(n) - y_{t(X-Y)}(n)\right], \quad (34)$$

$$E_{L,i}(n) = E_{L,i}(n-1) + \left[L_{f,i}(n) - L_{t,i}(n)\right], \quad i = 1 \sim 6. \quad (35)$$

The control signals for respective motion axis in MCCPM are generated as follows

$$U_{x(X-Y)}(n+1) = K_{ex}E_{x(X-Y)}(n) + K_{\varepsilon r,x}E_{r,x(X-Y)}(n), \quad (36)$$

$$U_{y(X-Y)}(n+1) = K_{ey}E_{y(X-Y)}(n) + K_{\varepsilon r,y}E_{r,y(X-Y)}(n), \quad (37)$$

$$U_{L,i}(n+1) = K_{eL,i}E_{L,i}(n) + K_{\varepsilon r,Li}E_{r,Li}(n), \quad i = 1 \sim 6. \quad (38)$$

Figure 9 describes the operation of the proposed hybrid platform under multi-axis CCPM control.

4 Simulation for the hybrid platform under MCCPM

4.1 Virtual axes

Despite its simple appearance the proposed system (Figs. 1, 2) is highly sophisticated. Inverse kinematics shall be performed three times before the platform can do anything in terms of pursuing a contour (Eqs. 10, 18, 22).

An approach using a virtual axis is proposed and shown in Fig. 10 to simplify the whole control process. In Fig. 10 P represents virtual axes X, Y, Z and O represents virtual axes α, β, γ.

The inverse kinematics calculation is only performed at the final stage to find the control signals: $U_{X-Y\,\&\,Li=1\sim6} =$ Inv-kinematics(U_P, U_O), where U denotes control signals.

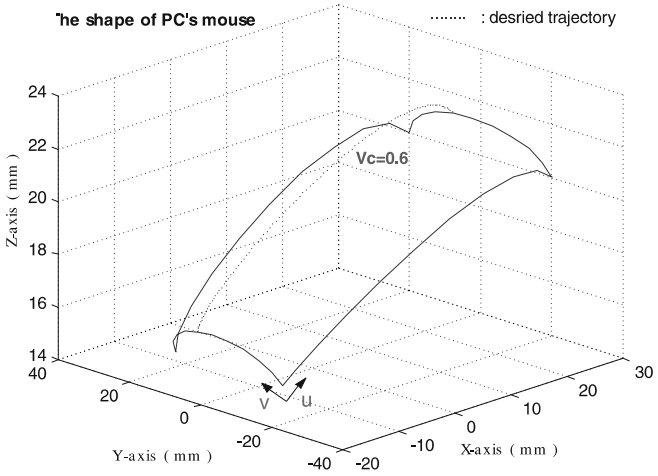


Fig. 11. The trajectory of free-form surface ($S(u, v)$, $v_c = 0.6$)

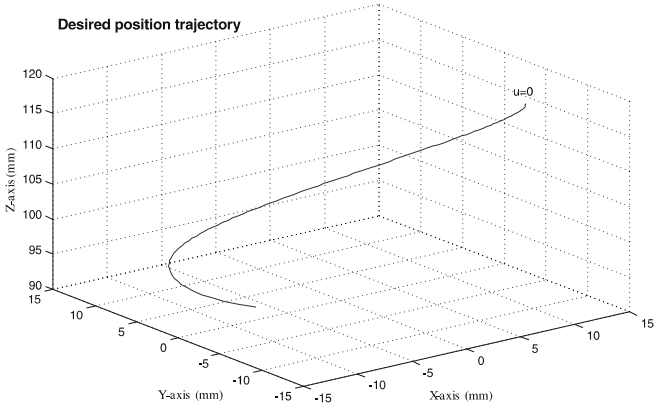
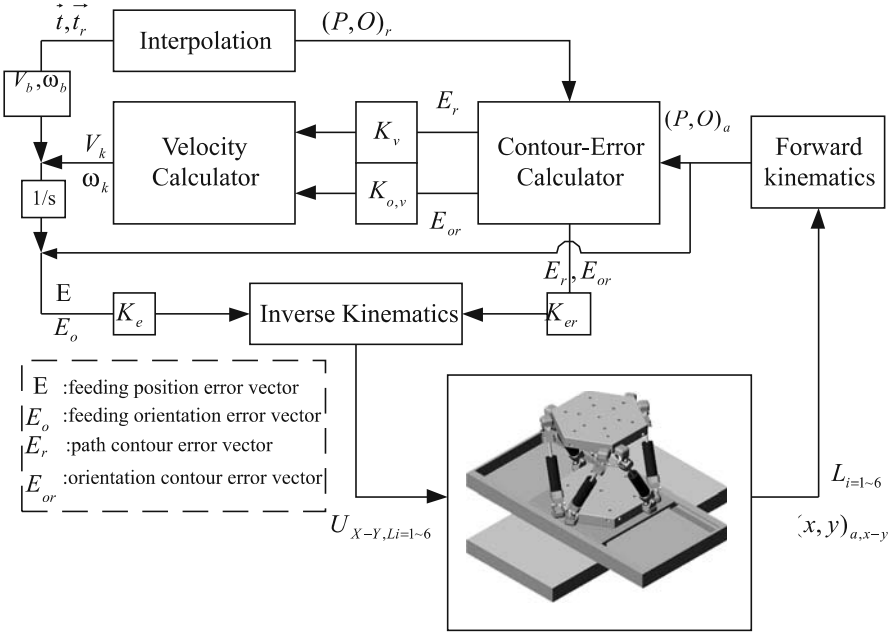


Fig. 12. Desired position trajectory $P_i(u, v_c)$ transformed from $S(u, v_c)$ of Fig. 11

Fig. 10. Control process using "virtual axes"



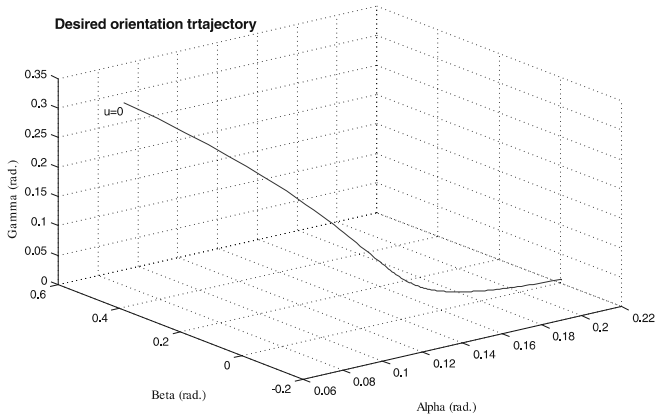


Fig. 13. Desired position trajectory $O_i(u, v_c)$ transformed from $S(u, v_c)$ of Fig. 11

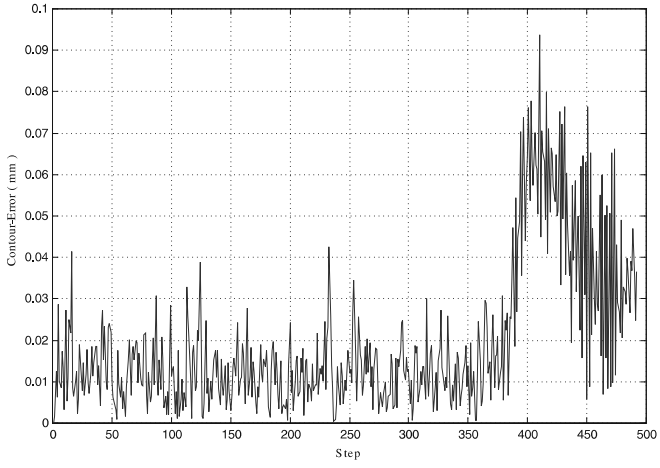


Fig. 14. The position contour error (MCCPM)

4.2 System simulation

Figure 11 shows the shape of a computer mouse for which the trajectory $S(u, v_c)$ is to be tracked. The hardware specifications are: $\phi = 15°$, $\theta = 15°$, $L_e = 50$, $L_b = 70$, $O_b = [0, 0, 20]$, $P_t = [0, 0, 150]$ (unit: mm) (see Fig. 3). The trajectory is first transformed to position and orientation information according to Sect. 3.1.2, as shown in Figs. 12 and 13, respectively, and then the tangent vector is computed. The results of overall tracking are shown in Fig. 14. The position IAE for the case in Fig. 14 is 0.019 mm, and the IAE orientation is 0.0004 rad. It is seen that the proposed hybrid motion platform is capable of precision tracking of continuous 3-D trajectory.

5 Conclusion and future work

This work pioneered a novel hybrid motion platform, which integrated the precision and the stiffness of parallel structure and the continuous contour tracking ability of serial machine. Framework for the multi-axis cross-coupled tracking control schemes based on this hybrid motion platform was established. Topics including trajectory construction, trajectory generation, forward and inverse kinematics, coordination of serial table and parallel platform, multi-axis tracking and system simulation were addressed.

In order to do real-time interpolation, the trajectory, which was derived from surface model, was mapped into position and orientation sets and stored as database. Real-time trajectory generation was performed by the calculation of trajectory parameter and retrieval of information from database. In order to cope with the five-axis feature, the planar CCPM scheme was extended to include both position and orientation precompensation. And the motion coordination between the x-y table and Stewart platform was explored. Finally, a simulation system was developed in which virtual axes were used to avoid repeated computation of inverse kinematics.

It was shown that this new kind of motion platform is capable of doing precision 3-D contour tracking under the proposed system structure. Extensive implementation of the system structure on a prototype platform is currently underway and new aspects of the technological advantages will be reported in future analyses.

References

1. Koren Y (1980) Cross-coupled biaxial computer control for manufacturing systems. ASME J Dyn Syst Meas Control 102(4):265–272
2. Koren Y, Lo Ch-Ch (1991) Variable gain cross coupling control for contouring. Ann CIRP 40:371–374
3. Yeh Z-M (1998) A cross-coupled bistage fuzzy controller for biaxis servomechanism control. Fuzzy Sets Syst 97:265–275
4. Chuang H-Y, Liu C-H (1991) Cross-coupled adaptive feedrate control for multiaxis machine tool. ASME J Dyn Syst Measure Control 113:451–457
5. Tarng YS, Chuang HY, Hsu WT (1999) Intelligent cross-coupled fuzzy federate controller design for CNC machine tools based on genetic algorithms. Int J Mach Tools Manuf 39:1673–1692
6. Srinivasan K, Kulkarni PK (1989) Cross-coupled control of biaxial feed drive servomechanisms. ASME J Eng Ind 111:140–148
7. Shieh Y-S, Lee A-C, Chen C-S (1996) Cross-coupled biaxial step control for CNC EDM. Int J Mach Tools Manuf 36(12):1363–1383
8. Chin J-H, Lin T-C (1997) Cross-coupled precompensation method for the contouring accuracy of computer numerically controlled machine tools. Int J Mach Tools Manuf 37(7):947–967
9. Chin J-H, Lin S-T (1997) The path precompensation method for flexible arm robot. Robot Comput Integr Manuf 13(3):203–215
10. Chin J-H, Lin H-W (1999) The algorithms of the cross-coupled precompensation method for generating the involute-type scrolls. ASME J Dyn Syst Meas Control 121:96–104
11. Zhang C-d, Song S-M (1993) Forward kinematics of a class of parallel (Stewart) platforms with closed-form solutions. Proceedings of the IEEE Int. Conf. on Robots and Automation, pp 2676–2681
12. Geng ZJ, Haynes LS (1994) A 3-2-1 kinematics configuration of a Stewart platform and its application to six degree of freedom pose measurements. Robot Comput Integr Manuf 11(1):23–34
13. Zhao X, et al. (2000) A successive approximation algorithm for the direct position analysis of parallel manipulators. Mech Mach Theory 35:1095–1101
14. Huynh P, et al. (1997) Maximum velocity analysis of parallel manipulators. IEEE Int. Conf. on Robotics and Automation, pp 3268–3273
15. Ropponen T, et al. (1995) Accuracy analysis of a modified Stewart platform manipulator. IEEE Int. Conf. on Robotics and Automation, pp 521–525

16. Wang J, et al. (1993) On the accuracy of a Stewart platform-I. The effect of manufacturing tolerances. IEEE Int. Conf. on Robotics and Automation, pp 725–731
17. Masory O, et al. (1993) On the accuracy of a Stewart platform-II. Kinematic calibration and compensation, IEEE Int. Conf. on Robotics and Automation, pp 114–120
18. Wang Z, et al. (2001) A study on workspace, boundary workspace analysis and workpiece positioning for parallel machine tools. Mech Mach Theory 36:605–622
19. Bhattacharya S, et al. (1998) Comparison of an exact and an approximate method of singularity avoidance in platform type parallel manipulators. Mech Mach Theory 33(7):965–974
20. Khalil W, Murareci D (1996) Kinematics analysis and singular configurations of a class of parallel robots. Math Comput Simul 41:377–390
21. Portman V (2000) Deterministic metrology of platform-type machine tools. Int J Mach Tools Manuf 40:1423–1442
22. Portman VT, Sandle B-Z, Zahavi E (2001) Rigid 6-DOF parallel platform for precision 3-D micromanipulation. Int J Mach Tools Manuf 41:1229–1250
23. Bohez ELJ (2002) Five-axis milling machine tool kinematic chain design and analysis. Int J Mach Tools Manuf 42:505–520
24. Pi J, Red E, Jensen G (1998) Grind-free tool path generation for five-axis surface machining. Comput Integr Manuf Syst 11(4):337–350
25. Kim J-H, et al. (2001) Development of a trajectory generation method for a five-axis NC machine. Mech Mach Theory 36:983–996
26. Lo C-C (1999) Efficient cutter-path planning for five-axis surface machining with a flat-end cutter. Comput-Aided Des 31:557–566
27. Lo C-C (1999) Real-time generation and control of cutter path for 5-axis CNC machining. Int J Mach Tools Manuf 39:471–488
28. Koren Y (1995) Five-axis surface interpolators. Ann CIRP 44(1): 379–382
29. Lee K (1999) Principles of CAD/CAM/CAE system. Addison-Wesley, Boston