1. Introduction

Parallel robot manipulators comprise a mobile platform connected to a fixed base through three or more articulated links and are used extensively throughout industry for such diverse applications as high-precision positioning systems, fiber alignment, welding, robotic manipulators, automatic inspection systems, and so forth. Therefore, planning a trajectory to perform a specific task is one of the most important class of problems in the applications of the parallel robot manipulators.

However, while moving along a specified trajectory, due to the limits on the workspace and existence of the force singularities, the parallel robot manipulators may not oppose forces or moments at some configurations. As a consequence, the manipulator gains some degrees of freedom, and becomes uncontrollable. Even the manipulator is very near to a singular manifold, the leg forces will increase violently to reach their allowable limits. Therefore, it is meaningful to plan a path without crossing a singular manifold on any operation for the parallel robot manipulators.

Compared to the vast researches on the path-programming of serial manipulators, studies on the singularity-free path programming of the parallel robot manipulators are relatively few. (Bhattacharya et al., 1998) developed an exact and an approximate on-line singularity avoidance method to restructure a path in the vicinity of a singular manifold for platform type parallel manipulators. (Dasgupta & Mruthyunjaya, 1998) proposed an algorithm to obtain a singularity-free trajectory for given two end-poses. The continuous paths are constructed through well-conditioned via points by examining the condition number at discrete steps on the corresponding straight line segment. (Sen et al., 2003) used a variational approach based on a Lagrangian to plan singularity-free paths with the actuators lengths remaining within their allowable limits. (Dash et al., 2005) used local routing method based on Grassmann’s line geometry to avoid isolated singularities inside the reachable workspace of parallel manipulators.

However, at present, most researches on the singularity-free path programming only deal with the kinematics of the parallel robot manipulators. When repetitive tasks in industrial applications are considered, some of physical operating costs, such as actuating forces or energy consumption, will become significantly important. Consequently, these effects should be further taken into account for a singularity-free path programming.
In this chapter, the dynamics formulation of a general parallel robot manipulator with an arbitrary geometry and inertia distribution according to the Boltzmann-Hamel-d’Alembert formulation will be first described. Then, the developed dynamics model in terms of the task-space coordinates are used to implement the singularity-free path programming for given two end poses of the parallel robot manipulator by minimizing some cost functions such as actuating forces, travel time and energy expenditure.

2. The Boltzmann-Hamel-d’Alembert formulation

2.1 Dynamic equations in terms of hybrid space coordinates

When planning a singularity avoidance path with the minimization of a cost function, e.g., actuating forces, travel time and energy consumption, etc., the dynamic equations of the parallel robot manipulators should be developed for this purpose. Several methods such as the Newton-Euler formulation (Do & Yang, 1998), (Dasgupta & Mruthyunjaya, 1998), (Dasgupta & Mruthyunjaya, 1998), (Nakamura & Ghodoussi, 1989), (Nakamura & Yamane, 2000), virtual work principle (Zhang & Song, 1993), (Wang & Gosselin, 1998), (Tsai, 2000), Kane’s method (Liu et al., 2000), kinematic influence coefficient theory (Wang et al., 2003) and the traditional Lagrangian formulation in generalized coordinates (Lebret et al., 1993), (Pang & Shahinpoor, 1994) are proposed for modeling and simulation of the dynamics of parallel manipulators. Comparing these approaches each other, it is obvious that the Lagrangian formulation has more physical insight because it utilizes the kinetic and potential energies to generate a well-structured form of equations of motion. However, if generalized coordinates are selected to express rotations of a structure in space, the partial derivatives of the Lagrangian by Lagrangian formulation in terms of generalized coordinates are quite tedious due to more intensive symbolic computations on time-varying inertia matrix. Therefore, such a direct formulation in all generalized coordinates is so arduous and cumbersome that the dynamic equations were developed most only under some simplifying assumptions regarding the geometry, configuration, structure and inertia distribution of manipulators. Rather than using all of generalized coordinates for the dynamics formulation, here the angular velocities as the time derivatives of a set of quasi-coordinates instead of generalized coordinates are utilized. Such angular velocities expressions are so-called quasi-velocities. The formulated kinetic and potential energies are expressed in matrix forms as function of quasi-velocities and rotation matrices, and then the Boltzmann-Hamel-d’Alembert formulation is employed to derive closed-form dynamic equations of the parallel robot manipulators with a completely general architecture and inertia distribution.

Fig. 1 illustrates the geometric structure of the general parallel robot manipulators considered in the present analysis. As shown, the mechanism has a fixed base and a moving platform connected to the base through six extensible legs with a spherical joint Bi at the top of the leg i (i = 1, ..., 6) and a universal joint Ai at the base end. Each leg is composed of two segments, the lower segment 1 is a fixed part and the upper segment 2 is a moving part, both are joined together by a prismatic joint, and each prismatic joint is driven by a linear actuator.
When planning a singularity avoidance path with the minimization of a cost function, Fig. 1 illustrates the geometric structure of the general parallel robot manipulators.

2.1 Dynamic equations in terms of hybrid space coordinates

Given two end poses of the parallel robot manipulator by minimizing some cost functions, task-space coordinates are used to implement the singularity-free path programming for a parallel manipulator. Comparing these approaches each other, it is obvious that the Kane's method (Liu et al., 2000), kinematic influence coefficient theory (Wang et al., 2003), virtual work principle (Zhang & Song, 1993), (Wang & Gosselin, 1998), (Tsai, 2000), the Newton-Euler formulation (Ding & Yang, 1998), (Dasgupta & Mruthyunjaya, 1998), (Nakamura & Ghodoussi, 1989), (Nakamura & Yamane, 1989) are proposed for modeling and simulation of the dynamics of parallel manipulators. Rather than using all of generalized coordinates for the formulation, here the angular velocities as the time derivatives of a set of quasi-coordinates instead of generalized coordinates are utilized. Such angular velocities expressed in matrix forms as function of quasi-velocities and rotation matrices, and then the formulated kinetic and potential energies are expressed in closed form. However, if some simplifying assumptions regarding the geometry, configuration, structure and inertia distribution of manipulators. Rather than using all of generalized coordinates for the formulation, here the angular velocities as the time derivatives of a set of quasi-coordinates instead of generalized coordinates are utilized. Such angular velocities expressed in matrix forms as function of quasi-velocities and rotation matrices, and then the formulated kinetic and potential energies are expressed in closed form.

Fig. 1. Schematic diagram of a general 6-UPS parallel robot manipulator

Consider one of the six legs and the moving platform for a dynamic analysis as shown in Fig. 2, an inertial coordinate system N-frame, X_NY_NZ_N, is fixed at the base and another coordinate system B-frame, X_bY_bZ_b, is attached to the moving platform with its origin at the gravity center B of the moving platform. Two parallel local coordinate systems, i1-frame and i2-frame with their corresponding origins A_i1 and A_i2, are attached to the respective lower segment 1 and upper segment 2 of the leg i. With the definitions of coordinate systems, the position vector of the gravity center of the moving platform with respect to the N-frame is denoted as p, the position vector of A_i1 with respect to the N-frame is represented as a_i, the position vector of the ball joint B_i with respect to the B-frame is defined as b_i, and the length vector of leg i from the universal joint A_i1 to the ball joint B_i.

Fig. 2. Coordinate systems for one leg and moving platform
with respect to the $i_{i}$-frame is expressed as $l_i$. Because each leg's length vector is equal to a total of the length vector, $l_{1i}$, of the lower segment 1 from $A_{1i}$ to $A_{1z}$ and the constant length vector, $l_{2i}$, of the upper segment 2 from $A_{1z}$ to $B_{i}$, the length vector of leg $i$ can be written as

$$l_i = l_is_i = l_{1i} + l_{2i} = (l_{1i} + l_{2i})s_i$$

where $s_i$ is a unit vector along the $i$th leg axis and expressed in the $i_{i}$-frame; $l_i$ is the length magnitude of the leg $i$; $l_{1i}$ and $l_{2i}$ are the respective length magnitudes of the lower segment 1 and the upper segment 2 of the leg $i$.

With multiple closed-loop chain mechanisms in a parallel robot manipulator, the geometric constraint equation on the basis of a vector loop relation can be written as

$$p + \sum_{i}^{N} Rb_i = a_i + \sum_{i}^{N} Rl_i$$

In Eq. (2), the rotation matrix $^N_iR$ with the subscript $L \in \{B, i = 1, ..., 6\}$ represents a transformation from a local L-frame to the inertial N-frame. The time derivative of a rotation matrix can be obtained by

$$^N_i\dot{R} = ^N_iR\tilde{\omega}_L$$

in which $\tilde{\omega}_L$ denotes a skew symmetric matrix formed from the angular velocity $\omega_L = [\omega_x, \omega_y, \omega_z]^T$ with respect to a local body-fixed L-frame such as

$$\tilde{\omega}_L = \begin{bmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{bmatrix}$$

Moreover, the angular velocity expressed in a body-fixed frame can be defined in a set of quasi-coordinates $\beta_L$ that are only defined meaningfully in angular quasi-velocities such as

$$\omega_L = \frac{d\beta_L}{dt}$$

Differentiating the constraint equation, Eq. (2), yields the constraint equation in terms of velocities, the results is

$$\dot{p} + \sum_{i}^{N} R\tilde{\omega}_ib_i = \sum_{i}^{N} Rl_i + \sum_{i}^{N} R\tilde{\omega}_il_i$$

where $l_i = \dot{l}_{1i}$ implies that each leg’s sliding velocity is equivalent to the relative sliding velocity between these two segments of the leg.
Since each leg is connected to the base with an universal joint; therefore, no spin is allowed about its longitudinal axis, it gives

$$\omega_i \cdot s_i = 0$$  \hspace{1cm} (7)

By pre-multiplying Eq. (6) respectively by \( (N_i R s_i)^T \) and by \( \hat{s}_i \), the magnitude of the sliding velocity as well as the angular velocity of leg \( i \) with respect to the \( i_1 \)-frame can be calculated as (Chen, 2003)

$$\dot{l}_i = \dot{l}_i = \left( N R s_i \right)^T \dot{\rho} + \left( N R s_i \right)^T N R \hat{\omega}_b b_i \hspace{1cm} (8)$$

$$\omega_i = \frac{1}{l_i} \hat{s}_i \left( N R^T \left( \dot{\rho} + N R \hat{\omega}_b b_i \right) \right) \hspace{1cm} (9)$$

Because Eq. (8) relates the velocities of the \( i \)th active joint to the linear velocities and angular velocities of the moving platform, the Jacobian matrix can be formulated by writing six times for each \( i \) from 1 to 6 and assembling in a matrix form as

$$\begin{bmatrix} \dot{l}_1 \\ \vdots \\ \dot{l}_6 \end{bmatrix} = Jacob \begin{bmatrix} \dot{\rho} \\ \omega_b \end{bmatrix} \hspace{1cm} (10)$$

where

$$Jacob = \begin{bmatrix} \left( N R s_1 \right)^T & \left( N R s_2 \right)^T & \left( N R s_3 \right)^T & \left( N R s_4 \right)^T & \left( N R s_5 \right)^T & \left( N R s_6 \right)^T \end{bmatrix} \hspace{1cm} (11)$$

Using the Boltzmann-Hamel-d’Alembert formulism (Chen & Chi, 2008), (Chen & Liao, 2008), the dynamic equation of the parallel robot manipulator can be formulated in terms of the task space coordinates (i.e. \( \ddot{x} = \left[ \dot{\rho}^T \ \dot{\gamma}_b^T \right]^T \), where \( \dot{\gamma}_b \) is defined by a set of Euler angles specifying the orientations of the moving platform) as follows:

$$\begin{bmatrix} \ddot{M}_p \ddot{x} + \dddot{D}_p \dot{x} + \dddot{G}_p \end{bmatrix} = \ddot{f}_p \hspace{1cm} \hspace{1cm} (12)$$

where \( \dddot{f}_p \) is a function of the output forces of the linear actuators, \( f = \left[ f_1 \ \ldots \ f_6 \right] \). The matrices and vectors in Eq. (12) are fully derived in the Appendix.

For the forward dynamics analysis, the trajectories of the manipulator can be solved numerically as a function of the actuating forces and initial states. For the inverse dynamics, the actuating forces are given as a function of the positions, velocities and accelerations of the moving platform. I.e.,

$$f = B^{-1} \left( \dddot{M}_p \ddot{x} + \dddot{D}_p \dot{x} + \dddot{G}_p \right) \hspace{1cm} (13)$$
where \( \overline{B} = C_i^T (Jacob) \)

It can be seen from Eq. (13) that a very large actuating force is required if the Jacobian matrix is rank-deficient or the value of the determinant of the Jacobian matrix approaches a singular manifold.

### 2.2 Fundamental structure properties

Although some fundamental structure properties of dynamic equations have been derived for open-chain manipulators in robotics, these derivations are of general nature based on the direct Lagrangian generalized coordinates formulation in an index form. In the sequel, these fundamental structural properties for the dynamics of the general parallel manipulators will be validated in a straight proof.

**Lemma**

For the task space dynamic equations, Eq. (12),

1. The inertia matrix \( \overline{M}_p \) is symmetric and positive definite.
2. \( \dot{M}_p - 2\overline{C}_p \) is a skew-symmetric matrix.

**Proof:**

Since the sub-matrices \( M_1 \), \( M_2 \) and \( M_p \) are symmetric, (see Appendix), the symmetry property of \( \overline{M}_p \) is easily seen by taking transpose on \( \overline{M}_p \) such that \( \overline{M}_p^T = \overline{M}_p \). The kinetic energy of the parallel robot manipulator is equal to a summation of the kinetic energy of the moving platform and the six legs. Thus,

\[
K.E. = \frac{1}{2} \dot{x}^T \overline{M}_p \dot{x} = \frac{1}{2} \dot{\phi}_1^T M_1 \dot{\phi}_1 + \frac{1}{2} \dot{\phi}_2^T M_2 \dot{\phi}_2
\]

(14)

Therefore, the positive definiteness of \( \overline{M}_p \) is guaranteed by the definition that the kinetic energy of the system is zero only if it has a zero velocity.

To prove that \( \dot{M}_p - 2\overline{C}_p \) is a skew-symmetric matrix, we first show that \( D_2 = \begin{bmatrix} 0 & 0 \\ 0 & D_{22} \end{bmatrix} \) and

\[
\dot{M}_2 - 2D_2 = \begin{bmatrix} 0 & -2D_{24} \\ -2D_{45} & M_{44} - 2D_{44} \end{bmatrix}
\]

are all skew-symmetric. It implies that \( D_{22} \) and \( \dot{M}_{44} - 2D_{44} \) are required to be skew-symmetric matrices as well as \( D_{22}^T = -D_{22} \) and \( D_{44}^T = -D_{44} \) are quite clear from the notation of these sub-matrices, (see Appendix). We then calculate

\[
\ddot{M}_{44} - 2D_{44} = \text{diag}[E_1, \ldots, E_6]
\]

(15)

where \( E_i = m_i \dot{I}_{11} (\ddot{S}_{12} - \ddot{d}_{12} \dddot{S}_{12}) - 2(I_{11} + \dot{I}_{11}) \omega_{12}^T - 2m_i \dot{I}_{11} (\omega_{12}^T \dddot{d}_{12}) \delta_{i1}, \quad i = 1, \ldots, 6 \)

After taking transpose it is shown that \((\dot{M}_{44} - 2D_{44})^T = -(M_{44} - 2D_{44})\). So it is concluded that both \( D_1 \) and \( \dot{M}_2 - 2D_2 \) are symmetric matrices.
Next, it will be shown that \( \dot{M}_p - 2D_p \) is skew-symmetric. We obtain

\[
\dot{M}_p - 2D_p = -2D_1 + J_1^T J_2^T (\dot{M}_2 - D_2) J_2 J_1 + \frac{d}{dt} (J_1^T J_2^T) M_2 (J_2^T J_1) - J_1^T J_2^T M_2 \frac{d}{dt} (J_2^T J_1)
\]  

(16)

Now that the first two terms of the right hand side are the sum of skew-symmetric matrices, by taking transpose and using symmetry of \( M_2 \), \( (\dot{M}_p - 2D_p)^T = -(\dot{M}_p - 2D_p) \) is proved.

Finally, \( \tilde{M}_p - 2\tilde{C}_p \) is calculated as

\[
\tilde{M}_p - 2\tilde{C}_p = C_1^T (\dot{M}_p - 2D_p) C_1 + \dot{C}_1 M_p C_1^T - C_1^T M_p \dot{C}_1
\]  

(17)

Because the first term of the right hand side has been proven as a skew-symmetric matrix, after taking transpose and inverting the sign, the skew-symmetric property of \( \tilde{M}_p - 2\tilde{C}_p \) for the task space dynamic equations is verified.

### 3. Singularity-Free Paths Programming

#### 3.1 Cost function and constraints

The objective here is to determine a path completely within the workspace with the following cost functions to be minimized while moving from the configuration \( x_o \) to the final configuration \( x_f \) during the time interval from \( t_0 \) to \( t_f \):

1. **Minimum actuating force**

\[
G = \int_{t_0}^{t_f} \left( \sum_{i=1}^{6} |f_i| \right) dt
\]  

(18)

2. **Time optimum**

\[
G = \Delta t = t_f - t_0
\]  

(19)

3. **Energy efficiency**

\[
G = \int_{t_0}^{t_f} \left| f^T \dot{t} \right| dt
\]  

(20)

4. **Mixed cost function**

A cost function can be formulated by mixing Eqs. (18) & (20), or Eqs. (19) & (20) using a weighting coefficient whose value is in the range of zero and one. The weighting coefficient weights these two functions according to the relative importance.

In addition, the considered trajectories in the task space must satisfy the constraints and conditions as follows:

1. **Initial and final conditions:**

\[
x(t_0) = x_o, \quad x(t_f) = x_f
\]

with the associated boundary conditions.
\[ \dot{x}(t_o) = \dot{x}(t_f) = \ddot{x}(t_o) = \ddot{x}(t_f) = 0 \]  

(2) Leg length constraints

\[ l_{\min} \leq l_i(x) \leq l_{\max} \quad \text{for } i=1,\ldots,6 \text{ and } t_o \leq t \leq t_f \]  

(3) Leg’s linear velocity constraints

\[ \left| \dot{l}(x, \dot{x}) \right| \leq \dot{l}_{\max} \quad \text{for } i=1,\ldots,6 \text{ and } t_o \leq t \leq t_f \]  

(4) Leg’s linear acceleration constraints

\[ \left| \ddot{l}(x, \dot{x}, \ddot{x}) \right| \leq \ddot{l}_{\max} \quad \text{for } i=1,\ldots,6 \text{ and } t_o \leq t \leq t_f \]  

(5) Actuating force constraints

\[ |f_i(x, \dot{x}, \ddot{x})| \leq f_{\max} \]  

Note that Eq. (25) restricts the output forces so that the planned path is implicitly singularity-free.

Typically, the singularity-free path programming with the minimum cost function is a constrained optimization problem. A suitable path should be selected so that the considered cost function subject to the conditions and constraints, Eqs. (21)-(25), is minimized.

### 3.2 Geometric path representation

A smooth spatial path can be generated by control points of path function. Because the B-spline function provides a simple method to create curves between the defined points, therefore, it is used for the path function. The trajectory \( x \) at each time sub-interval \( t_i \leq t \leq t_{i+1} \) \((i=0,1,\ldots,n-1)\) can be interpolated as

\[ x(t(u)) = U N \hat{X}_i \]  

where \( t(u) = t_i + u \Delta t, \quad 0 \leq u \leq 1 \)

\[ \Delta t = (t_f - t_o) / n \]

\[ U = \begin{bmatrix} u^3 & u^2 & u^1 & 1 \end{bmatrix} \]

\[ \hat{X}_i = \begin{bmatrix} \hat{x}^r_i & \hat{x}^r_{i+1} & \hat{x}^r_{i+2} \end{bmatrix}^T \]

and \( \hat{x}^r_k \) \((k=-1, 0, \ldots, n, n+1)\) are called the control points and total \((n+3)\) points. \( u \) is the independent spline parameter, \( N \) defines the shape of spline, being expressed for the cubic B-spline as
The associated velocities and accelerations along the trajectory are obtained by subsequent derivatives of \( x \) as

\[
\dot{x}(t(u)) = U^N \ddot{X}_i / \Delta t
\]

\[
\ddot{x}(t(u)) = U^N \dddot{X}_i / (\Delta t)^2, \ t_i \leq t \leq t_{i+1}
\]

where the prime is denoted as the derivative with respect to the spline parameter \( u \). Also, from the boundary condition, Eq. (21), the first three control points and the last three are fixed as

\[
\begin{align*}
\dot{x}_1 &= \dot{x}_0 = \dot{x}_1 = x_0 \\
\dot{x}_{n-1} &= \dot{x}_n = \dot{x}_{n+1} = x_f
\end{align*}
\]

Thus, the undetermined control points are reduced to \( \dot{x}_2, \ldots, \dot{x}_{n-1} \). By substituting the terms \( x, \dot{x} \) and \( \ddot{x} \) into Eq. (13), then into the cost function as well as the constraint Eqs. (22)-(25), the constraint optimization problem can be recast into an unconstrained problem by adding weighted penalty functions to form a pseudo cost function, the pseudo cost function can be expressed as

\[
\bar{G} = \sum_{i=0}^{n-1} \Delta t_i \int F(u, \dot{X}_i) du + \sum_{j=1}^{6} (w_1 \text{TRUE}(l_{\text{min}} - l_{\text{max}}(\dot{x}, u)) + w_2 \text{TRUE}(l_{\text{max}}(\dot{x}, u) - l_{\text{max}}) \\
+ w_3 \text{TRUE}(l_{\text{max}}(\ddot{x}, u) - \dot{l}_{\text{max}})) + w_4 \text{TRUE}(l_{\text{max}}(\dddot{x}, u) - \ddot{l}_{\text{max}}) + w_5 \text{TRUE}(l_{\text{max}}(\dddot{x}, u) - \dddot{l}_{\text{max}}))
\]

in which the kernel function \( F(u, \dot{X}_i) \) is associated with the cost function. In the chapter, \( F = \sum_{j=1}^{6} |f_j| \), \( F = 1 \) and \( F = \|f^T\| \) are respectively for the minimum actuating force problem, the time optimal problem, and the energy efficiency problem. The constraints are also mapped into the function of the spline parameter \( u \) and the set of control points \( \dot{x}_k \) \( (k=1, 0, \ldots, n, n+1) \). In addition, \( \text{TRUE} \) is a logical operational function with the value one if the statement is true and zero if the statement is false; \( w_1, \ldots, w_5 \) are the weighting factors that determine the magnitude of the penalty and should be taken large enough so that any constraint violation will be penalized with large cost.

By the transformation of the constrained optimization problem, a singularity-free path programming is equivalent to the determination of a set of control points for the minimization of the pseudo cost function. Although sufficient control points can generate any smooth spatial curve, the required computational load for the optimization will be increased dramatically.
3.3 Global search based on PSOA

In order to solve the highly nonlinear optimization problem described by the pseudo cost function, an evolutionary computation based optimization technique, PSOA mimicking the food-searching behavior of birds is employed (Kennedy & Eberhart, 1995). The process of PSOA is described as following:

1. Randomly initialize position \( \hat{X}_p = [\hat{x}_{p,1}^T, \ldots, \hat{x}_{p,n}^T]^T \), namely, undetermined control points, and velocity \( \hat{V}_p = [\hat{v}_{p,1}^T, \ldots, \hat{v}_{p,n}^T]^T \), \( p = 1, \ldots, P \), in which \( P = 40 \) particles is set.

2. Formulate the actuating forces through solving the inverse dynamics Eq. (13), and then a numerical integral based on the Simpson’s rule is used to calculate the integral of the pseudo cost function \( \overline{G}_p \) in Eq. (31) for each particle.

3. Compare the current fitness value \( \overline{G}_p \) with the particle’s previous best value \( \overline{G}_{p,b} \), if \( \overline{G}_p < \overline{G}_{p,b} \), then \( P_{p,b} = X_p \), \( p = 1, \ldots, P \), in which \( P_{p,b} = [p_{b,1}^T \ldots p_{b,n}^T]^T \) is the position of each particle with its personal best fitness value \( \overline{G}_p \).

4. Compare the current fitness value \( \overline{G}_p \) with the group’s previous best value \( \overline{G}_{g,b} \), if \( \overline{G}_p < \overline{G}_{g,b} \), then \( G_b = X_p \), \( p = 1, \ldots, P \), in which \( G_b = [g_{b,1}^T \ldots g_{b,n}^T]^T \) is the only one that has the global best fitness value \( \overline{G}_{g,b} \) thus far.

5. Update each particle according to PSOA,

\[
\begin{align*}
\dot{v}_{pj} &= v_{pj} + c_1 \cdot \text{rand}(\cdot) \left( \left(p_{b,j} - \hat{x}_{pj}\right) + c_2 \cdot \text{Rand}(\cdot) \left( g_{b,j} - \hat{x}_{pj} \right) \right), \quad j = 1, \ldots, 6 \cdot (n-3) \quad (32) \\
\dot{x}_{pj} &= \hat{x}_{pj} + v_{pj} \quad (33)
\end{align*}
\]

where \( v_{pj} \) is the rate of the position change for particle \( p \) with respect to the \( j \)th dimension and clamped to the range \( [-v_{\text{max}}, v_{\text{max}}] \) with \( v_{\text{max}} = 0.8 \) to prevent the likelihood of the particle from leaving the search space; \( c_1 \) and \( c_2 \) are two positive constants, in which \( c_1 = c_2 = 1.5 \) are set to control the step size that a particle will move in the direction of best position; \( \text{rand}(\cdot) \) and \( \text{Rand}(\cdot) \) are two normal distribution functions. \( \hat{x}_{pj} \) is the position of the \( p \)th particle with respect to the \( j \)th dimension.

6. If a convergence criterion is met, it means that all the particles have been toward the global best and its own individual best. Otherwise, repeat the steps starting from step (2) until it has reached the maximum allowable iteration number. The maximum iteration number is 350. Furthermore, the relative change in the group’s best pseudo objective function \( \overline{G}_{g,b} \) for two consecutive iterations being less than 0.01 is defined as the convergence criterion for the optimization process.

4. Numerical Simulations

4.1 Kinematic Parameters

The optimal singularity-free path programming with the minimum actuating force for the parallel robot manipulator is implemented in MATLAB routines. The kinematic parameters
used for the simulations are given below.

The platform points in the B-frame:

\[
[B_1 \ B_2 \ B_3 \ B_4 \ B_5 \ B_6] = \begin{bmatrix}
0.3 & 0.3 & 0.0 & 0.2 & 0.15 & 0.15 \\
0.0 & 0.2 & 0.3 & 0.1 & 0.2 & 0.15 \\
0.1 & 0.0 & 0.0 & 0.1 & 0.05 & 0.05 \\
\end{bmatrix}
\]

The base points in the N-frame:

\[
[A_1 \ A_2 \ A_3 \ A_4 \ A_5 \ A_6] = \begin{bmatrix}
0.6 & 0.1 & -0.3 & -0.3 & 0.2 & 0.5 \\
0.2 & 0.5 & 0.3 & -0.4 & -0.3 & -0.2 \\
0.0 & 0.1 & 0.0 & 0.0 & -0.05 & 0.0 \\
\end{bmatrix}
\]

All legs are considered to be identical. The masses of lower segment 1 and upper segment 2 of each leg are respective \(m_{i_1} = 2\) kg and \(m_{i_2} = 1\) kg, and the mass of the moving platform is \(m_p = 32\) kg. The unit vector \(s_i\) expressed in the \(i_i\)-frame is given by \(s_i = [1 \ 0 \ 0]^T\). The gravity centers of part 1 and part 2 of each leg \(d_{i_1} = d_{i_2} = [0.05 \ 0.001 \ 0.001]^T\) m. The constant length of the upper segment 2 is \(l_{i_2} = 1.0\) m. The moments of inertia of these two segments of each leg as well as the moving platform about their respective gravity centers in local frame are \(c I_{i_1} = \text{diag} [0.001, 1, 1] \text{kg} \cdot \text{m}^2\), \(c I_{i_2} = \text{diag} [0.0001, 0.4, 0.4] \text{kg} \cdot \text{m}^2\), \(I_b = \text{diag} [2, 2, 4] \text{kg} \cdot \text{m}^2\). The moments of inertia of the segments 1 and 2 of each leg about the respective joint points \(A_{i_1}\) and \(A_{i_2}\) in local frame can be calculated using parallel axes theorem as \(I_{i_1} = c I_{i_1} + m_{i_1} d_{i_1}^T d_{i_1}\), and \(I_{i_2} = c I_{i_2} + m_{i_2} d_{i_2}^T d_{i_2}\). The vector of gravity is \(g = [0 \ 0 \ -9.81]^T \text{m} / \text{sec}^2\).

The Euler angles vector \(\gamma = [\psi \ \theta \ \phi]^T\) is used to specify the orientation of the platform and each leg, which means that the inertial frame is rotated about the \(Z\) axis with \(\psi\) radians first, resulting in the primed frame; then about the \(\gamma'\) axis with \(\theta\) radians, resulting in the double primed frame; and finally about the \(\psi''\) axis with \(\phi\) radians, resulting in the local frame. Therefore, the rotation matrix of the local frame relative to the inertial frame is written by

\[
R = \begin{bmatrix}
c \psi c \theta & -s \psi c \phi + c \psi s \theta s \phi & s \psi s \phi + c \psi s \theta c \phi \\
c \psi s \theta & s \psi c \phi + c \psi s \theta s \phi & -c \psi s \phi + s \psi s \theta c \phi \\
-s \theta & c \theta s \phi & c \theta c \phi \\
\end{bmatrix}
\]  
(34)

where \(c(\cdot)\) is a shorthand notation for \(cos(\cdot)\) and \(s(\cdot)\) for \(sin(\cdot)\). The velocity transformation matrix \(C_L\) converting the angular velocity in terms of the time derivative of the Euler angles to the one with respect to a body-fixed frame is given as
where the subscript \( L \in \{ B, i = 1, \ldots, 6 \} \).

The path planning algorithm presented in the above steps is demonstrated by the following simulations. The desired initial pose \( x_0 \) and final pose \( x_f \) for the parallel robot manipulator are

\[
\begin{bmatrix}
    1.6 \\ -0.2 \\ -0.6 \\ 0.0 \\ 0.0 \\ 0.0
\end{bmatrix}^T, 
\begin{bmatrix}
    -0.8 \\ 0.1 \\ -0.4 \\ 0.0 \\ 0.0 \\ 0.0
\end{bmatrix}^T.
\]

The straight path connecting these two end-poses and keeping constant orientation is through a singular manifold as is shown by the zero determinant of the Jacobian in Fig 3. Then, optimal singularity-free paths will be sought between \( x_0 \) and \( x_f \) during the time interval subject to each leg’s length constraints, \( l_{\text{min}} = 0.2 \text{m} \), \( l_{\text{max}} = 3.0 \text{m} \), \( \dot{l}_{\text{max}} = 0.8 \text{m/sec} \), \( \ddot{l}_{\text{max}} = 1.0 \text{m/sec}^2 \) and \( f_{\text{max}} = 2500 \text{N} \).

Fig. 3. Determinant in straight path under constant orientation

Also, the weighting factors \( w_1, \ldots, w_5 \) serving as the magnitude of the penalty should be chosen appropriately only when the constraints are violated; because a small value may yield major constraint violations, but a large value will result in a very poor optimization problem. Thus, \( w_1, w_2 \) and \( w_3 \) are taken as \( 5 \times 10^5 \) and \( w_4 = w_5 = 5 \times 10^4 \) by a few trial simulations.

Finally, singularity-free paths are programmed by eight control points, i.e., two undetermined control points among them. More control points could imply a more flexible search for the optimal singularity-free paths, but with increased control points, it will cost more computation time for convergence.

4.2 Results and Discussions

According to the aforementioned parameters, the following examples will be illustrated to determine the optimal singularity-free paths:

\[
C_L = \begin{bmatrix}
-s\theta & 0 & 1 \\
-c\theta \phi & c\phi & 0 \\
-c\theta c\phi & -c\phi & 0
\end{bmatrix}
\]
(1) Minimum actuating force
The paths programmed for the minimum actuating forces considering the constant orientations and varied orientations of the moving platform during the time interval \( t_f = 20 \text{sec} \) are shown in Fig. 4. Fig. 5 is the orientation variations of the moving platform while traveling along the planned path with varied orientations. It is seen that these two planned paths are quite distinct.

![Fig. 4. Singularity-free planned paths for minimum actuating forces](image)

![Fig. 5. Orientation of moving platform along planned path with varied orientations](image)

Fig. 4 depicts the variations of the determinant along the corresponding paths. As is shown, the determinants are never equal to zero for the paths, the obtained paths based on PSOA successfully avoid singularities. Moreover, the difference of the evaluated values along the two obtained paths is of one order. The result is reflected on the time history plot of each leg’s actuating forces as shown in Figs. 7(a), (b). It is observed that the actuating forces for the actuators 2 and 3 reach their largest values at \( t = 6 \text{ sec} \) along the path with a constant orientation, this implies that the robot manipulator is the nearest to a singular manifold for the pose.
Although the actuating forces can be decreased by varying the orientations of the moving platform, the required maximum leg lengths increase as indicated in Figs. 8(a), (b). It means that a larger task space is necessary to accommodate the planned singularity-free path.

(2) Time optimum
For this problem, the travel time \( t_f \) is to be determined. Based on the singularity-free path planning algorithm, the planned trajectory is shown in Fig.9 (i.e. the line for \( \mu =1 \)) with the corresponding minimal travel time \( t_f =5.85 \) sec.

Fig. 6. Variations of determinant along corresponding planned paths

Fig. 7. Actuating forces along planned paths with (a) constant orientation and (b) varied orientations

(3) Energy efficiency
Fig. 9 also shows the minimal-energy trajectory with the corresponding travel time \( t_f =20.01 \) sec (i.e. the line for \( \mu =0 \)). Compared with the time optimal trajectory planning, reduction in the travel time is at the expense of a greater consumed energy, a poorer fitness value, and a larger force.

(4) Mixed cost function
The cost function is defined as

$$G = \mu \lambda (\Delta t) + (1-\mu) \int_0^t \left| f \right|^2 dt$$  \hspace{1cm} (36)$$

The optimal singularity free trajectories for \( \mu = 0.3, 0.6 \) and 0.8 with the corresponding determined travel time \( t_f = 7.758, 6.083 \) and 6.075 sec are also respectively shown in Fig. 9.

Fig. 8. Leg lengths along planned paths with (a) constant orientation and (b) varied orientations

5. Conclusions

In this chapter, a numerical technique is presented to determine the singularity-free trajectories of a parallel robot manipulator. The required closed-form dynamic equations for the parallel manipulator with a completely general architecture and inertia distribution are
developed systematically using the new structured Boltzmann-Hamel-d’Alembert approach, and some fundamental structural properties of the dynamics of parallel manipulators are validated in a straight proof.

In order to plan a singularity-free trajectory subject to some kinematic and dynamic constraints, the parametric path representation is used to convert a planned trajectory into the determination of a set of undetermined control points in the workspace. With a highly nonlinear expression for the constrained optimal problem, the PSOA needing no differentiation is applied to solve for the optimal control points, and then the corresponding trajectories are generated. The numerical results have confirmed that the obtained singularity-free trajectories are feasible for the minimum actuating force problem, time optimal problem, energy efficient problem and mixed optimization problem. The generic nature of the solution strategy presented in this chapter makes it suitable for the trajectory planning of many other configurations in the parallel robot manipulator domain and suggests its viability as a problem solver for optimization problems in a wide variety of research and application fields.

![Fig. 9. Planned paths for time optimum, energy efficiency and mixed cost function](image)

### 6. References

Bhattacharya, S.; Hatwal, H. & Ghosh, A. (1998). Comparison of an exact and an approximate method of singularity avoidance in platform type parallel manipulator. *Mech. Mach. Theory*, **33**, 965-974.

Chen, C.T. (2003). A Lagrangian formulation in terms of quasi-coordinates for the inverse dynamics of the general 6-6 Stewart platform manipulator. *JSME International J. Series C*, **46**, 1084-1090.

Chen, C.T. & Chi, H.W. (2008). Singularity-free trajectory planning of platform-type parallel manipulators for minimum actuating effort and reactions. *Robotica*, **26**(3), 357-370.

Chen, C.T. & Liao, T.T. (2008). Optimal path programming of the SPM using the Boltzmann-Hamel-d’Alembert dynamics formulation model. *Adv Robot*, **22**(6), 705-730.

Dasgupta, B. & Mruthyunjaya, T.S. (1998). Singularity-free planning for the Stewart platform manipulator. *Mech. Mach. Theory*, **33**, 771-725.
Dasgupta, B. & Mruthyunjaya, T.S. (1998). Closed-form dynamic equations of the general Stewart platform through the Newton-Euler approach. *Mech. Mach. Theory, 33*, 993-1012.

Dasgupta, B. & Mruthyunjaya, T.S. (1998). A Newton-Euler formulation for the Inverse dynamics of the Stewart platform manipulator. *Mech. Mach. Theory, 33*, 1135-1152.

Do, W. & Yang, D. (1998). Inverse dynamic analysis and simulation of a platform type of robot. *J. Robot. Syst.*, 5, 209-227.

Dash, A.K.; Chen, I.; Yeo, S. & Yang, G. (2005). Workspace generation and planning singularity-free path for parallel manipulators. *Mech. Mach. Theory, 40*, 776-805.

Kennedy, J. & Eberhart, R. (1995). Particle swarm optimization, in *Proc. of IEEE Int. Conf. on Neural Network*, pp. 1942-1948, Australia.

Lebret, G.; Liu, K. & Lewis, F.L. (1993). Dynamic analysis and control of a Stewart platform manipulator. *J. Robot. Syst.*, 10, 629-655.

Liu, M.J.; Li, C.X. & Li, C.N. (2000). Dynamics analysis of the Gough-Stewart platform manipulator. *IEEE Trans. Robot. Automat.*, 16, 94-98.

Nakamura, Y. & Ghodoussi, M. (1989). Dynamics computation of closed-link robot mechanisms with nonredundant and redundant actuators, *IEEE Transactions on Robotics and Automation, 5*, 294-302.

Nakamura, Y. & Yamane, K. (2000). Dynamics computation of structure-varying kinematic chains and its application to human figures. *IEEE Trans. Robotics and Automation*, 16, 124-134.

Pang, H. & Shahinpoor, M. (1994). Inverse dynamics of a parallel manipulator. *J. Robot. Syst.*, 11, 693-702.

Sen, S.; Dasgupta, B. & Bhattacharya, S. (2003). Variational approach for singularity-free path-planning of parallel manipulators. *Mech. Mach. Theory, 38*, 1165-1183.

Tsai, L.W. (2000). Solving the inverse dynamics of a Stewart-Gough manipulator by the principle of virtual work. *Trans. ASME J. Mech. Design, 122*, 3-9.

Wang, S.C.; Hikita, H.; Kubo, H.; Zhao, Y.S.; Huang, Z. & Ifukube, T. (2003). Kinematics and dynamics of a 6 degree-of-freedom fully parallel manipulator with elastic joints. *Mech. Mach. Theory, 38*, 439-461.

Wang, J. & Gosselin, C.M. (1998). A new approach for the dynamic analysis of parallel manipulators. *Multibody System Dynamics, 2*, 317-334.

Zhang, C. & Song, S. (1993). An efficient method for inverse dynamics of manipulators based on the virtual work principle. *J. Robot. Syst.*, 10, 605-627.

**Acknowledgments**

This work is supported by the National Science Council of the ROC under the Grant NSC 98-2221-E-212 -026 and NSC 98-2221-E-269-015.

**Appendix**

- Dynamic equation of general parallel robot manipulators

\[
\ddot{\vec{M}}_p \ddot{\vec{x}} + \vec{D}_p \dot{\vec{x}} + \vec{G}_p = \vec{f}_p
\]

where \( \vec{M}_p = C_1^t (M_1 + J_1^t M_2 J_1) C_1 = C_1^t M_p C_1 \)
\[
\tilde{D}_p = C_i^T \left( (M_1 + J_i^T M_2 J_i) \dot{C}_i + (D_1 + J_i^T D_2 J_i + J_i^T M_2 J_i^T) C_i \right) = C_i^T \left( (M_1 + J_i^T M_2 J_i) \dot{C}_i + D_p C_i \right)
\]

\[
\tilde{G}_p = C_i^T (G_1 + J_i^T G_2)
\]

\[
\tilde{f}_p = C_i^T (Jacobian)^T f
\]

The actuating forces vector \( f = [f_1 \ldots f_6]^T \).

In (A1), the velocity transformation matrix, \( C_1 \), is defined as

\[
C_1 = \begin{bmatrix}
I_{3 \times 3} & 0 \\
0 & C_\omega
\end{bmatrix}
\]

where \( C_\omega \) is the angular velocity transformation of the moving platform. In addition, \( I_1 \) and \( J_2 \) are the sub-matrices appropriately partitioned while developing the equations of motion, and are defined as

\[
J = [J_1 \mid J_2] = \begin{bmatrix}
\left( ^N R_{s_1} \right)^T & \left( ^N R_{s_2} \right)^T \frac{1}{l_1} R b_1^T \\
\vdots & \vdots \\
\left( ^N R_{s_6} \right)^T & \left( ^N R_{s_7} \right)^T \frac{1}{l_6} R b_6^T \\
\frac{1}{l_1} \tilde{s}_1 \frac{1}{l_2} R^{T} & \frac{1}{l_4} \tilde{s}_4 \frac{1}{l_6} R^{T} \\
\vdots & \vdots \\
\frac{1}{l_6} \tilde{s}_6 \frac{1}{l_6} R^{T} & \frac{1}{l_8} \tilde{s}_8 \frac{1}{l_6} R^{T}
\end{bmatrix} - I_{2 \times 24}
\]

in which \( I_{24 \times 24} \) is an \( n \times n \) unitary matrix such that \( J_2 = -I_{2 \times 24} \).

- Inertia matrix, Coriolis and centrifugal terms, gravity vector

\[
M_1 = \begin{bmatrix}
M_{11} & 0 \\
0 & M_{22}
\end{bmatrix}, \quad M_2 = \begin{bmatrix}
M_{33} & M_{34} \\
M_{43} & M_{44}
\end{bmatrix}
\]

where \( M_{11} = m_\beta I_{3 \times 3} \)

\[
M_{22} = I_\beta
\]

\[
M_{33} = diag\left[ m_{i_{12}}, \ldots, m_{i_{62}} \right]
\]

\[
M_{34} = diag\left[ m_{i_{12}} \tilde{s}_{i_{12}}, \ldots, m_{i_{62}} \tilde{s}_{i_{62}} \right]
\]

\[
M_{43} = diag\left[ m_{i_{12}} \tilde{d}_{i_{12}}, \ldots, m_{i_{62}} \tilde{d}_{i_{62}} \right]
\]

\[
M_{44} = diag\left[ I_{1}, \ldots, I_{6} \right]
\]

and \( I_i = I_\beta + I_{i_{12}} - m_{i_{12}} \tilde{s}_{i_{12}}^2 + m_{i_{12}} l_{i_{12}} \left( \tilde{s}_{i_{12}} \tilde{d}_{i_{12}} + \tilde{d}_{i_{12}} \tilde{s}_{i_{12}} \right), \quad i = 1, \ldots, 6 \)
\[ D_1 = \begin{bmatrix} 0 & 0 \\ 0 & D_{22} \end{bmatrix}, \quad D_2 = \begin{bmatrix} 0 & D_{43} \\ D_{43} & D_{44} \end{bmatrix} \]  

(A5)

where \( D_{22} = \overline{I}_b \omega_B^T \)

\[
D_{44} = \text{diag}\left[ m_{i2}\omega_l^T \bar{s}_i (l_1 \bar{s}_1 + \bar{d}_{12}) \right., \ldots, \left. m_{i2}\omega_l^T \bar{s}_i (l_6 \bar{s}_6 + \bar{d}_{62}) \right]
\]

\[
D_{43} = \text{diag}\left[ m_{i2}(l_1 \bar{s}_1^T + \bar{d}_{12}^T) \bar{s}_i \omega_i \right., \ldots, \left. m_{i2}(l_6 \bar{s}_6^T + \bar{d}_{62}^T) \bar{s}_i \omega_i \right]
\]

\[
D_{44} = \text{diag}\left[ h_i, \ldots, h_i \right]
\]

and \( h_i = m_{i2} \bar{I}_i (l_i \bar{s}_i^T + \bar{d}_{i2}^T) \bar{s}_i + (I_{i1} + I_{i2}) \omega_i^T + m_{i2} \bar{I}_i (\omega_i^T \bar{d}_{i2}^T) \bar{s}_i^T, \quad i = 1, \ldots, 6 \)

The tildes over the matrix-vector products in \( D_{22} \) and \( h_i \) denote a skew-symmetric matrix formed from the matrix-vector product:

\[
G_1 = \begin{bmatrix} G_{11} \\ G_{21} \end{bmatrix}, \quad G_2 = \begin{bmatrix} G_{31} \\ G_{41} \end{bmatrix}
\]  

(A6)

where \( G_{11} = -m_i \bar{g} \)

\[
G_{21} = 0_{3x1}
\]

\[
G_{31} = \begin{bmatrix} -m_{i2} \bar{g}_1^T \bar{R}_1 s_1 \quad \ldots \quad -m_{i6} \bar{g}_6^T \bar{R}_6 s_6 \end{bmatrix}^T
\]

\[
G_{41} = \begin{bmatrix} -\bar{g}_1^T \bar{R}_1 (m_{i1} \bar{d}_{i1} + m_{i2} \bar{l}_1 \bar{s}_1 + m_{i2} \bar{d}_{12})^T \quad \ldots \quad -\bar{g}_6^T \bar{R}_6 (m_{i6} \bar{d}_{i6} + m_{i6} \bar{l}_6 \bar{s}_6 + m_{i6} \bar{d}_{62})^T \end{bmatrix}^T
\]
The purpose of this volume is to encourage and inspire the continual invention of robot manipulators for science and the good of humanity. The concepts of artificial intelligence combined with the engineering and technology of feedback control, have great potential for new, useful and exciting machines. The concept of eclecticism for the design, development, simulation and implementation of a real time controller for an intelligent, vision guided robots is now being explored. The dream of an eclectic perceptual, creative controller that can select its own tasks and perform autonomous operations with reliability and dependability is starting to evolve. We have not yet reached this stage but a careful study of the contents will start one on the exciting journey that could lead to many inventions and successful solutions.

How to reference
In order to correctly reference this scholarly work, feel free to copy and paste the following:

Chun-Ta Chen and Te-Tan Liao (2010). On the Optimal Singularity-Free Trajectory Planning of Parallel Robot Manipulators, Advances in Robot Manipulators, Ernest Hall (Ed.), ISBN: 978-953-307-070-4, InTech, Available from: http://www.intechopen.com/books/advances-in-robot-manipulators/on-the-optimal-singularity-free-trajectory-planning-of-parallel-robot-manipulators
