Dynamic modeling and robust control of a 3-PRC translational parallel kinematic machine

Yangmin Li *, Qingsong Xu

Department of Electromechanical Engineering, Faculty of Science and Technology, University of Macau, Av. Padre Tomás Pereira, Taipa, Macao SAR, China

A R T I C L E   I N F O

Article history:
Received 12 September 2007
Received in revised form 7 May 2008
Accepted 20 May 2008

Keywords:
Parallel robots
Inverse dynamics
Robust control
Tracking control

A B S T R A C T

The dynamic modeling and robust control for a three-prismatic-revolute-cylindrical (3-PRC) parallel kinematic machine (PKM) with translational motion have been investigated in this paper. By introducing a mass distribution factor, the simplified dynamic equations have been derived via the virtual work principle and validated on a virtual prototype with the ADAMS software package. Based upon the established model, three dynamics controllers have been attempted on the 3-PRC PKM. The intuitive co-simulations with the combination of MATLAB/Simulink and ADAMS show that the control performance of neither inverse dynamics control nor robust inverse dynamics control is satisfactory in the presence of parametric uncertainties in PKM dynamics. On the contrary, the controller based on the passivity-based robust control scheme is more suitable for tracking control of the PKM in terms of both control performances and controller design procedures. The results presented in the paper provide a sound base for both the mechanical system design and control system design of a 3-PRC PKM.

© 2008 Elsevier Ltd. All rights reserved.

1. Introduction

In recent years, the research and development activities on parallel kinematic machines (PKMs) have been accelerated along with the evolution on both hardware and software of computer technology since the latter provides a strong support for researchers and engineers in robotics community. A PKM generally consists of a moving platform that is connected to a fixed base through more than one limbs together. Comparing with a traditional serial manipulator that is constituted with rigid-body links and joints connected in serial, a PKM possesses a more rigid structure and better payload carrying ability. Thus, it is more suitable for situations where high precision, stiffness, velocity, and heavy load-carrying are required within a restricted workspace [1].

Up to now, most of the investigations can be found on kinematics issues of PKM [2–5], while relatively few researches can be referred to the dynamics control of PKM. Ordinarily, there are two problems in PKM dynamics, namely the inverse and forward dynamics, where the former solves the actuation forces of actuators once the trajectories (in either joint space or task space) are planned, and the latter deals with the output motion of the PKM when the actuation forces are given. The inverse dynamics may be used for the design of a dynamics controller, whereas the forward dynamics may be adopted for dynamics simulation of a PKM which can also be conducted by resorting to effective dynamics software packages such as ADAMS, DADS, and RecurDyn, etc. As far as the approaches to generate the PKM inverse dynamic model are concerned, most traditional methods are used, such as Newton–Euler formulation [6], Lagrangian formulation [7,8], virtual work principle [9,10], and some other methods [11]. With respect to the real-time control of a PKM, the goal of dynamic modeling is to establish an inverse dynamic model which is simple yet accurate enough to represent the PKM system.

In general, once a PKM is designed and developed, its manipulation accuracy can be guaranteed by designing a proper controller. The inverse dynamics control (IDC) can produce a nice control performance provided that a full dynamic model of the PKM is used with all dynamic parameters known exactly [12]. Nevertheless, there will always exist uncertainties in the dynamic model due to the difficulty to accurately identify the model parameters, or the existence of unmodeled dynamics arising from the simplifications in modeling process. Hence, the ideal control performance of the IDC method will be degraded. To solve this problem, the robust control [13–16] and adaptive control [17–20] schemes can be adopted. Generally speaking, the robust controller is a fixed controller designed to meet the performance requirements over a given range of uncertainties, whereas an adaptive controller utilizes the manner of parameter estimation to adapt the computational model to the dynamic model on line. However, an adaptive controller that performs well in the presence of parametric uncertainties may not perform ideally in the case...
of other types of uncertainties in terms of unmodeled dynamics or external disturbances. From this point of view, a robust controller can compensate more uncertainties than an adaptive one. Therefore, we employ a simplified dynamic model in conjunction with a robust controller to seek for an effective approach to deal with the control issues for a PKM in this research.

In previous works of the authors, a 3-PRC (three-prismatic-revolute-cylindrical) PKM with relative simple structure was presented in [21] with its kinematic problems solved in details. After that, the stiffness modeling was performed, which showed that the overconstrained 3-PRC PKM could be converted into a non-overconstrained 3-CRC (three-cylindrical-revolute-cylindrical) PKM without any influences on its mobility and kinematics [22]. The objective of the current research is to establish the inverse dynamic model and implement the robust control for a 3-PRC PKM in case of dynamics uncertainties. With the addition of validating the dynamics model in ADAMS, we accomplish a co-simulation on a virtual prototype by combining MATLAB with ADAMS software to verify the designed controllers [23]. The virtual prototype provides a test bed via computer technology so as to verify control algorithm and enhance the design timely just before the development of a prototype. Besides, the co-simulation allows not only a visual view of the animation for the PKM behavior induced by the controller but also an inspection of whether the designed prototype and controllers are satisfactory or not for real developments. So, it is more intuitive than most of the existing simulations purely based on control algorithms.

In the remainder of the paper, after a brief review of the 3-PRC PKM architecture and its kinematics in Section 2, the dynamic model is established and validated in Section 3. Afterwards, the IDC scheme is applied on the PKM in Section 4, and a robust IDC for the PKM is adopted in Section 5. However, it has been illustrated that the designed robust controller is not clearly superior to the former one, which motivates the design of a more effective one in Section 6 along with the performances validated through co-simulations. Finally, some concluding remarks are summarized in Section 7.

2. Architecture and kinematics description

The virtual prototype and schematic diagram for a 3-PRC PKM are shown in Figs. 1 and 2, respectively. The PKM consists of a moving platform, a fixed base, and three limbs with identical kinematic structure. Each limb connects the moving platform to the base by a P (prismatic) joint, a R (revolute) joint, and a C (cylindrical) joint in sequence, where the P joint is driven by a linear actuator assembled on the fixed base. Thus, the moving platform is attached to the base by three identical PRC linkages. Since the kinematics and mobility problems of the 3-PRC PKM have already been resolved in [21,22], we only review some useful results below which provide a base for the present research.

To facilitate the analysis, as shown in Figs. 2 and 3, we assign a fixed Cartesian frame O(x, y, z) at the centered point O of the fixed base, and a moving frame P(u, v, w) on the triangle moving platform at the centered point P, along with the y- and v-axes perpendicular to the platform, and the x- and z-axes parallel to the u- and w-axes, respectively. Both \( \Delta A_i A_2 A_3 \) and \( \Delta B_1 B_2 B_3 \) are assigned to be equilateral triangles so as to obtain a symmetric workspace of the manipulator. In addition, the ith limb \( C_i B_i \) (i = 1, 2, 3) with the length of l is connected to the moving platform at \( B_i \) which is a point on the axis of the ith C joint. \( B_i \) denotes the point on the moving platform that is coincident with the initial position of \( B_i \), and the three points \( B_i \) lie on a circle of radius b. The three rails \( M_i N_i \) intersect one another at point D and intersect the x–z plane at points \( A_1, A_2, \) and \( A_3 \) that lie on a circle of radius a. The sliders of P joints \( C_i \) are restricted to move along the rails between \( M_i \) and \( N_i \). Moreover, the axes of the R and C joints within the ith limb are parallel to each other to ensure only translational motion of the platform. Angle \( \alpha \) is measured from the fixed base to
rails $M_iN_i$ and is defined as the layout angle of actuators. Without loss of generality, let the z-axis point along $A_iO_i$ and the w-axis along $B_iP_i$. Additionally, let $d_{max}$ and $s_{max}$ denote the maximum stroke of linear actuators and passive C joints, respectively.

The purpose of the inverse kinematics is to solve the actuated variables from a given position of the moving platform. Given a set of actuation inputs, the position of the moving platform can be solved by the forward kinematic analysis. With reference to Fig. 3, a vector-loop equation can be written for the $i$th limb as

$$l_{i0} = l_i - d_i d_0,$$  
(1)

with the notation of

$$l_i = p_i + s_i s_0 - a_i,$$  
(2)

where $l_{i0}$ is the unit vector along $C_iB_i$, $d_i$ represents the linear displacement of the $i$th actuated joint, $d_0$ is the unit vector directing along rail $M_iN_i$, $s_i$ is the stroke of the $i$th C joint, and $s_0$ denotes the unit vector parallel to the axes of the C and R joints of limb $i$.

Substituting Eq. (2) into Eq. (1) and dot-multiplying both sides of the expression by $s_i$ allows the derivation of $s_i$, i.e.,

$$s_i = -s_{i0}^T p_i.$$  
(3)

Additionally, differentiating both sides of Eq. (1) with respect to time along with a necessary calculation allows the derivation of the velocity equation:

$$q = J x,$$  
(4)

where $q = [d_1, d_2, d_3]^T$ is a vector of actuated joint rates, $x = [p_1, p_2, p_3]^T = [v_1, v_2, v_3]^T$ denotes the vector of linear velocities for the moving platform, and $J = J_0 J$ is defined as the Jacobian matrix of a 3-PRC PKM relating output velocities to the actuated joint rates, where

$$J_q = \begin{bmatrix} I_{10} d_{10} & 0 & 0 \\ 0 & I_{20} d_{20} & 0 \\ 0 & 0 & I_{30} d_{30} \end{bmatrix}_{3 \times 3}, \quad J_x = \begin{bmatrix} I_{10} \\ I_{20} \\ I_{30} \end{bmatrix}.$$  
(5)

3. Dynamic modeling and validation

The main object of dynamic analysis for a PKM is to develop an inverse dynamic model, which enables the computation of the required actuator forces/torques when a desired trajectory is given. At the same time, for a real-time implementation of a dynamics controller, the model should be simple and accurate enough to represent the machine dynamics. In what follows, by introducing a mass distribution factor, we perform the dynamic modeling for a 3-PRC PKM using the virtual work principle and validate it through the dynamics simulation in ADAMS environment.

3.1. Simplification hypotheses

Concerning a 3-PRC PKM, the complexity of its dynamics partly comes from the three moving legs. Since the legs can be manufactured with light materials, we can simplify the dynamics problem by defining a mass distribution factor $w$ ($0 < w < 1$) for the legs. That is, the mass of each leg is divided into two portions and placed at its two extremities, i.e., one part with the proportion of $w$ at its lower extremity (moving platform), and the other part $(1 - w)$ at its upper extremity (the slider), and hence the rotational inertias of legs are neglected.

Let $m_p$, $m_s$, and $m_l$ denote the masses for the moving platform, one slider, and one leg, respectively. Then, the equivalent masses of the moving platform and the slider become

$$m_p = m_p + \frac{2}{3} w m_l,$$  
(6)

$$m_s = m_s + \frac{1}{3} (1 - w) m_l.$$  
(7)

The simplification method for other types of PKM was reported for the dynamic modeling of the DELTA parallel robot [24], where it was directly assigned that one third of the passive leg mass was attached to the moving platform, i.e., $w = \frac{1}{3}$. In what follows, we will show that it is a different case for a 3-PRC PKM and a proper value of $w$ will be determined based on dynamics simulation.

3.2. Dynamic modeling

Assume that $f = [f_1, f_2, f_3]^T$ is a vector of actuator forces, $\delta q = [\delta d_1, \delta d_2, \delta d_3]^T$ and $\delta x = [\delta p_1, \delta p_2, \delta p_3]^T$ are the vectors of virtual linear displacements for the sliders and the moving platform, respectively. Besides, let $\delta s = [\delta s_1, \delta s_2, \delta s_3]^T$ be a vector for the virtual displacements of C joints with respect to the moving platform.

Applying the virtual work principle allows the derivation of the following equation by neglecting the friction forces in passive R and C joints and assuming that there are no external forces exerted:

$$f^T \delta q + G_q^T \delta q - F_q^T \delta q + G_p^T \delta x - (F_p^T \delta x - F_p^T \delta s) = 0,$$  
(8)

where $G_q = [m_0 g, m_0 g, m_0 g]^T$ is the vector of gravity forces of sliders with $g$ denoting the gravity acceleration, $G_p = [0, -m_p g, 0]^T$ is the gravity force vector of the moving platform, $F_q = [m_0 d_1, m_0 d_2, m_0 d_3]^T$ describes a vector for the inertial forces of sliders, $F_p = [m_p p_1, m_p p_2, m_p p_3]^T$ represents the vector of inertial forces of the moving platform, and $F_s = [(m_0)/2s_1, (m_0)/2s_2, (m_0)s_3]^T$ is the vector of inertial forces for the legs of lower parts, respectively.

In view of Eq. (3), we can derive that

$$\delta s = s_o \delta x.$$  
(9)

where

$$s_o = \begin{bmatrix} s_{i0}^T \\ s_{j0}^T \\ s_{k0}^T \end{bmatrix}_{3 \times 3}.$$  

In addition, with reference to Eq. (4), we can obtain that

$$x = J^{-1} q,$$  
(10)

which leads to

$$\delta x = J^{-1} \delta q.$$  
(11)

Substituting Eqs. (9) and (11) into Eq. (8), yields

$$(f^T + G_q^T - F_q^T + G_p^T J^{-1} - F_p S_0 J^{-1}) \delta q = 0.$$  
(12)

Since Eq. (12) holds for any virtual displacements $\delta q$, we have

$$f = F_s + J^{-T} F_p - J^{-T} S_0 F_i - G_i - J^{-T} G_p.$$  
(13)

Next, substituting the inertial forces into Eq. (13), results in the computed forces:

$$f = M_i q + J^{-T} M_p x - J^{-T} S_0 M_s x + G_s - J^{-T} G_p.$$  
(14)
where
\[
\begin{bmatrix}
M_s & 0 & 0 \\
0 & M_s & 0 \\
0 & 0 & M_s \\
\end{bmatrix}
\]
\[
\begin{bmatrix}
M_p & 0 & 0 \\
0 & M_p & 0 \\
0 & 0 & M_p \\
\end{bmatrix}
\]
\[
\begin{bmatrix}
m_s/2 & 0 & 0 \\
0 & m_s/2 & 0 \\
0 & 0 & m_s/2 \\
\end{bmatrix}
\]
In addition, differentiating Eqs. (4) and (10) with respect to time, respectively, leads to
\[
\ddot{q} = J\ddot{x} + J\dot{\dot{q}},
\]
\[
x = J^{-1}\ddot{q} + J^{-1}\dot{q}.
\]
And then, substituting Eq. (16) into Eq. (14), allows the generation of dynamic model for a 3-PRC PKM expressed in joint space:
\[
f = M(q,q)\ddot{q} + C(q,q)\dot{q} + G(q),
\]
where
\[
M(q) = M_s + J^{-1}M_pJ^{-1} - J^{-1}s_l^T M_s s_l J^{-1},
\]
\[
C(q,q) = (J^{-1}M_p - J^{-1}s_l^T M_s s_l )J^{-1},
\]
\[
G(q) = -G_0 - J^{-1}G_p.
\]
On the other hand, substituting Eq. (15) into Eq. (14), results in the dynamic model of a 3-PRC PKM described in task space:
\[
f_x = J^T f = M_t(x)\ddot{x} + C_t(x,\dot{x})\dot{x} + G_t(x),
\]
with
\[
M_t(x) = J^T M_s J + M_p - s_l^T M_s s_l,
\]
\[
C_t(x,\dot{x}) = J^T M_s s_l,
\]
\[
G_t(x) = -J^T G_0 - G_p,
\]
where \(x \in \mathbb{R}^3\) denotes the controlled variables, \(M_t(x) \in \mathbb{R}^{3 \times 3}\) is a inertia matrix, \(C_t(x,\dot{x}) \in \mathbb{R}^{3 \times 3}\) represents the matrix of centrifugal and Coriolis forces, and \(G_t(x) \in \mathbb{R}^3\) is the vector of gravity forces.

In view of Eqs. (18) and (20), it is observed that the dynamic model in task space is less complicated than that in joint space, hence the model of Eq. (19) is adopted for the following dynamics analysis and control purposes. It can be shown that the dynamic model possesses two noticeable features, i.e., the inertial matrix \(M_t\) is symmetric positive definite and \(M_t - 2C_t\) satisfies the skew symmetry property.

The main problem in practical implementation of task space control for a PKM comes from the acquisition of the output posture of the PKM moving platform. For real implementation, employing redundant sensors in measurement will be easy to realize nowadays. In the current simulation studies, the forward kinematics for a 3-PRC PKM is solved on-line by resorting to the Newton–Raphson method thanks to the high performance of the computer hardware.

3.3. Validation on dynamic model

In ADAMS, a virtual prototype for a 3-PRC PKM with kinematic and dynamic parameters described in Table 1 has been created. The established dynamic model for the PKM has been verified by employing the ADAMS software package via a simulation study carried out below.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>(a)</td>
<td>0.336 m</td>
<td>(x)</td>
<td>30.0°</td>
</tr>
<tr>
<td>(b)</td>
<td>0.152 m</td>
<td>(d_{\text{max}})</td>
<td>0.310 m</td>
</tr>
<tr>
<td>(l)</td>
<td>0.400 m</td>
<td>(s_{\text{max}})</td>
<td>0.250 m</td>
</tr>
<tr>
<td>(m_p)</td>
<td>3.982 kg</td>
<td>(m_t)</td>
<td>0.906 kg</td>
</tr>
<tr>
<td>(m_s)</td>
<td>1.297 kg</td>
<td>(g)</td>
<td>9.807 m/s²</td>
</tr>
</tbody>
</table>

![Fig. 4. Actuator forces obtained by dynamic formulation and ADAMS simulation in case of \(w = 0.5\).](image)

Let the moving platform track a helical trajectory along the \(y\)-axis in the reference frame, i.e.,
\[
p_x = 0.05 \sin \left( \frac{\pi}{2} t \right) \cos \left( \frac{3}{2} \pi t \right),
\]
\[
p_y = h_0 - 0.025 \left[ 1 - \cos \left( \frac{\pi}{2} t \right) \right],
\]
\[
p_z = 0.05 \cos \left( \frac{\pi}{2} t \right) \sin \left( \frac{3}{2} \pi t \right),
\]
where \(t\) is the time variable in unit of second, and \(p_x\), \(p_y\), and \(p_z\) are in units of meters, with \(h_0 = -0.3552\) m denoting the home position height of the moving platform. For the simulation, the input displacements of linear actuators are solved via the inverse kinematics, and then exported to ADAMS to drive the PKM following the trajectory in Eq. (21). The actuation forces of the three actuators for \(w = 0.5\) are illustrated in Fig. 4, which can be observed that there are deviations between the two approaches due to the introduced hypotheses.

Since the mass distribution factor \(w\) varies from 0 to 1, it is necessary to determine its value to get an optimal dynamics simplification. With the variation of the factor \(w\), the error range distributions for the actuator forces obtained by the dynamic model with comparison to the ADAMS output are illustrated in Fig. 5. It is observed that the distribution of the actuation error range of the 3-PRC PKM is the best in case of \(w = 0.58\) since the actuation force error reaches to the minimum value and the errors are almost symmetrically distributed about the zero mean value. Therefore, the optimal factor \(w = 0.58\) rather than \(w = 1\) is designed for the 3-PRC PKM in this research. Under such a case, the deviations of the computed actuation forces with respect to the ADAMS simulation results are plotted in Fig. 6, which reveals that the computed force errors of the dynamic model are within only \(\pm 1.25\%\).
optimal values for the mass distribution factor \( w \). Due to the simple architecture of the established dynamic model, it can be adopted for dynamics control of the 3-PRC PKM. Due to the small variations on velocities and accelerations, the mass factor is also leg dependent, which is different from leg to leg even for the same trajectory. This phenomenon can be observed by an insightful view of Fig. 6, which indicates that the optimal mass distribution factor \( w = 0.58 \) for the PKM is actually optimized for leg 3 in terms of error distributions around corresponding mean values. For a more detailed study, different mass factors should be considered for different legs to obtain a more accuracy result, which is remained for our future research.

4. Inverse dynamics control

As an important basis for dynamics control, the IDC scheme is based upon the transformation of the nonlinear dynamic equations into equivalent linear and decoupled second-order systems [12].

4.1. IDC algorithm

The dynamics control in task space utilizing the IDC method is implemented for a 3-PRC PKM in what follows. Firstly, the task space dynamic model in Eq. (19) can be rewritten into the form:

\[
\mathbf{f}_\varepsilon = \mathbf{M}_\varepsilon (\mathbf{x}) \dot{\mathbf{x}} + \mathbf{H}_\varepsilon (\mathbf{x}, \dot{\mathbf{x}}),
\]

where \( \mathbf{H}_\varepsilon (\mathbf{x}, \dot{\mathbf{x}}) = \mathbf{C}_\varepsilon (\mathbf{x}, \dot{\mathbf{x}}) + \mathbf{G}(\mathbf{x}) \).

Fig. 7 depicts a block diagram of the IDC scheme with proportional-derivative (PD) feedback. Assuming that there are no external disturbances, then the manipulator is actuated by the actuation forces described in the task space:

\[
\mathbf{f}_\varepsilon = \mathbf{M}_\varepsilon (\mathbf{x}) \mathbf{u} + \mathbf{H}_\varepsilon (\mathbf{x}, \dot{\mathbf{x}}),
\]

where \( \mathbf{u} \) is an input signal vector in the form of acceleration.

Combining Eq. (22) with Eq. (23) results in the following linear second-order system:

\[
\ddot{\mathbf{x}} = \mathbf{u},
\]

which indicates that the system of Eq. (23) under control in Eq. (24) is linear and decoupled with respect to the input vector \( \mathbf{u} \). And the reference signal can be defined according to the following algorithm:

\[
\mathbf{r} = \dot{\mathbf{x}}_d + \mathbf{K}_p \mathbf{x}_d + \mathbf{K}_d \ddot{\mathbf{x}}_d,
\]

where \( \mathbf{x}_d \) denotes the desired trajectory for the moving platform, and the symmetric positive definite feedback gain matrices are chosen as the diagonal form:

\[
\mathbf{K}_p = \text{diag}(\omega_p^2), \quad \mathbf{K}_d = \text{diag}(\zeta \omega_n^2),
\]

which indicates that the \( i \)th (\( i = 1, 2, \) and 3) component of reference \( \mathbf{r} \) influences only the \( i \)th DOF motion of the PKM with a natural frequency \( \omega_n \) and a damping ratio \( \zeta \).

The acceleration input signal in Eq. (24) then becomes

\[
\ddot{\mathbf{x}} = \ddot{\mathbf{x}}_d + \mathbf{K}_p (\mathbf{x}_d - \mathbf{x}) + \mathbf{K}_d (\mathbf{x}_d - \mathbf{x}).
\]

Substituting Eq. (28) into Eq. (24), leads to a homogeneous second-order differential equation of errors:

\[
\ddot{\mathbf{e}} + \mathbf{K}_p \ddot{\mathbf{e}} + \mathbf{K}_d \mathbf{e} = 0,
\]

where

\[
\mathbf{e} = \mathbf{x}_d - \mathbf{x}
\]
is the vector of displacement tracking errors. It has been shown that the error system in Eq. (29) is asymptotically stable along with the positive definite matrices \( \mathbf{K}_p \) and \( \mathbf{K}_d \).

![Fig. 5. Distribution of the actuator force error range versus the mass factor w.](image)

![Fig. 6. Deviations of computed actuation forces with respect to simulation results from ADAMS in case of \( w = 0.58 \).](image)

![Fig. 7. Block diagram of inverse dynamics control scheme in task space.](image)
4.2. Simulation results and discussions

A co-simulation on the virtual prototype of the 3-PRC PKM has been accomplished by combining MATLAB/Simulink with ADAMS. During the simulation, the control algorithm is executed under MATLAB to generate the command forces, which are then exported to ADAMS environment and applied to the actuators of the virtual prototype at each cycle of time. The resulted outputs of the virtual prototype (position and velocity of the moving platform) are measured by “sensors” in ADAMS and then fed back to the controller in MATLAB for calculation of the next command signal. In such a communication manner, the co-simulation proceeds until the end of time, and the input and output variables for the virtual prototype are illustrated in Fig. 8.

The manipulator is commanded to track a trajectory defined in Eq. (21). In order to make the system response critically damped, the damping ratio is chosen as \( \zeta = 1.0 \) to eliminate overshooting of tracking errors. Moreover, to avoid exciting the structural oscillation and resonance of the PKM system, the undamped natural frequency \( \omega_n \) may be designed no more than one-half of the structural resonant frequency \( \omega_s \) of the PKM, i.e., \( \omega_n \leq \omega_s / 2 \).

Assume that the mechanical resonant frequency of the 3-PRC PKM is 6Hz. Then, the natural frequency of the control system can be designed as: \( \omega_n = 2\pi \times 3 \text{Hz} = 18.85 \text{rad/s} \), which allows the determination of the feedback gains:

\[
K_p = 355.3 \quad \text{and} \quad K_d = 37.7. \tag{31}
\]

The simulation results show that the above IDC approach results in almost zero steady-state errors. However, in real situation, the payload and dynamic parameters may not be exactly known. With the normal dynamic parameters of the 3-PRC PKM deviated 20% from the exact values, the control results are illustrated in Fig. 9. With the forward kinematics solved on-line by the Newton–Raphson iterative method, the simulation time is about 10 s running on a personal computer with Pentium-4 3.00-GHz CPU.

The oscillation ranges of position and velocity control errors are, respectively, defined as the tolerances of steady-state position errors \( e_{tol}(e) \) and velocity errors \( e_{tol}(\dot{e}) \) after the settling time \( t_s \) in this paper. We use both the maximum control errors \( e_{max}(e) \) and \( e_{max}(\dot{e}) \) and the tolerances of steady-state errors to describe the control performances of a designed controller. From Fig. 9, we can observe that the maximum control errors are \( e_{max}(e) = 4.8 \text{mm}, e_{max}(\dot{e}) = 5.6 \text{mm/s} \), and the tolerances of steady-state errors are \( e_{tol}(e) = 5.8 \text{mm}, e_{tol}(\dot{e}) = 10.8 \text{mm/s} \), respectively, after \( t_s = 0.3 \text{s} \).
Although the IDC control performance can be enhanced by increasing the feedback gains, the gains are restricted by the natural frequency of the PKM. Thus the control performance is limited, which provides a motivation for the design of a robust controller in the following section.

5. Robust IDC scheme

5.1. Robust inverse dynamics control (R IDC) design

Referring to the dynamic equation (22), in the presence of uncertainties including modeling errors, unknown loads, and parameters measurement, the manipulator is actuated with the following joint forces expressed in the task space:

\[ f_i = \dot{M}_i(x)u + \ddot{H}_i(x, \dot{x}), \]

where \( u \) is a new input vector to be determined, \( \dot{M}_i(x) \) and \( \ddot{H}_i(x, \dot{x}) \) denote the estimators of the inertial matrix \( \dot{M}_i(x) \) and the nonlinear coupling matrix \( \ddot{H}_i(x, \dot{x}) \) implemented in the controller, respectively. The errors of the estimates, i.e., the uncertainties, can be expressed by

\[ \dot{M}_i = \dot{M}_i - \dot{M}_i, \quad \ddot{H}_i = \ddot{H}_i - \ddot{H}_i, \]

Taking Eq. (32) as a nonlinear control law gives

\[ \dot{M}_i(x)\ddot{x} + \ddot{H}_i(x, \dot{x}) = \dot{M}_i(x)u + \ddot{H}_i(x, \dot{x}), \]

which allows the generation of

\[ \ddot{x} = u + (\dot{M}_i^{-1}\dot{M}_i - I)u + \dot{M}_i^{-1}\dddot{H}_i = u - \eta, \]

where \( I \) denotes a \( 3 \times 3 \) identity matrix and

\[ \eta = (I - \dot{M}_i^{-1}\dot{M}_i)u + \dot{M}_i^{-1}\dddot{H}_i. \]

In view of the position error expressed by Eq. (30), the velocity error can be written as \( e = x_d - x \), and the acceleration error can be calculated as follows with the consideration of Eq. (35).

\[ \ddot{e} = \ddot{x}_d - \ddot{x} + \eta \]

In order to compensate for the uncertainties, the following input vector is chosen:

\[ u = \ddot{x}_d + K_P\dot{e} + K_Ie + \sigma. \]

By setting

\[ E = \begin{bmatrix} e \\ \dot{e} \end{bmatrix} \]

as the state of the system, Eq. (38) can be written as

\[ \dot{E} = AE + B(\eta - \sigma), \]

where

\[ A = \begin{bmatrix} 0 & 1 \\ -K_P & -K_I \end{bmatrix}, \quad B = \begin{bmatrix} 0 \\ 1 \end{bmatrix}. \]

Since the matrices \( K_P \) and \( K_I \) are chosen so that \( A \) in Eq. (40) is a Hurwitz matrix, i.e., \( A \) has all its eigenvalues in the open left half of the complex plane, any symmetric positive definite matrix \( P \) can be chosen to give a unique solution \( Q \) satisfying the relationship:

\[ A^TQ + QA = -P. \]

where \( Q \) is symmetric positive definite as well.

Defining the control law for \( \sigma \) in accordance with [15]:

\[ \sigma = \begin{cases} \rho(e, \dot{e})/||\mu|| & \text{if } ||\mu|| > \varepsilon, \\ \rho(e, \dot{e}) / \varepsilon & \text{if } ||\mu|| \leq \varepsilon, \end{cases} \]

where \( \mu = B^TQE \) and \( \varepsilon > 0 \), it follows that the control input in Eq. (38) is continuous and the Lyapunov function candidate

\[ V(E) = E^TQE > 0, \quad \forall E \neq 0, \]

satisfies \( V < 0 \) along the trajectories of the error system of Eq. (40). The proof procedure is similar to that in [12] and hence is omitted here.

5.2. Simulations and discussions

In order to implement the R IDC scheme presented above, we have to determine several control parameters in terms of \( K_P, K_I, P, \varepsilon, \) and \( \rho \). Generally, the greater the uncertainty is, the larger the positive scaler \( \rho \) is, and the constant \( \varepsilon \) can be chosen as large as necessary to reduce or eliminate chattering. Additionally, it is observed that \( \rho \) is a function of tracking errors \( (e, \dot{e}) \) and the calculation of \( \rho \) requires the determination of bounds associated with the desired tracking trajectory and the uncertainties including \( x_M, M_M, \kappa, \) and \( \gamma \).

A co-simulation is carried out to command the 3-PRC PKM tracking the same trajectory described by Eq. (21). With the normal dynamic parameters offset 20% from the real values of the PKM, the trajectory and uncertainty bounds can be computed as: \( x_M = 1.24, M_M = 0.23, \kappa = 30.06, \) and \( \gamma = 0.85 \). In addition, the feedback gains are designed according to the natural frequency of the system as described in Eq. (31), and other control parameters are chosen as: \( P = \text{diag}(10) \) and \( \varepsilon = 0.2 \).

The simulation results are illustrated in Fig. 11, which shows that after \( t_s = 0.8 \text{s} \), the control performances are \( e_{\text{max}}(e) = 4.5 \text{mm}, e_{\text{max}}(\dot{e}) = 3.6 \text{mm/s}, e_{\text{cal}}(e) = 5.3 \text{mm}, \) and \( e_{\text{cal}}(\dot{e}) = 6.8 \text{mm/s} \), respectively. Comparing the control results with those of the IDC, we can conclude that the reduction of control errors is not very obvious, while the settling time is increased by almost 2 times. So, the control performance of the designed robust controller is not clearly superior to the former inverse dynamics controller.

By changing the values of design parameters, we can observe that the increasing values of some parameters may lead to less steady-state errors. For example, an increase of \( M_M \) by 5 times, i.e., \( M_M = 1.38, \) leads to the control performances of \( e_{\text{cal}}(e) = 3.9 \text{mm}, e_{\text{cal}}(\dot{e}) = 3.2 \text{mm}, e_{\text{cal}}(e) = 4.7 \text{mm}, \) and \( e_{\text{cal}}(\dot{e}) = 4.6 \text{mm/s}, \)
respectively, as depicted in Fig. 12. Thus, the maximum control errors on displacement and velocity tracking have been greatly reduced by 19% and 43%, and the error tolerances have been reduced by 19% and 57%, respectively, with respect to the IDC results. However, the control errors are reduced at the expense of a long settling time \( t_s = 1.6 \text{s} \) to reach the steady state for all of the tracking errors, that is more than 4 times longer than the settling time in the IDC method. A further increase of control parameters will reduce the control errors even more at the sacrifice of longer settling time to arrive at the steady state, and induce more severe initial oscillations of the control errors at the same time. Hence, the design parameters cannot be increased infinitely, and the performance of the RIDC controller is limited.

Besides, an overview of the design procedure for the robust controller reveals that there are as many as eight control parameters and uncertainty bounds to be determined in advance. This motivates us to seek a more simple controller for an effective control of a 3-PRC PKM in the presence of parameters uncertainties in the following section.

6. Passivity-based robust control scheme

In this section, we attempt an alternative robust controller for a 3-PRC PKM based on the passivity or skew symmetry property of the dynamic equations. Making use of both the skew symmetry property and linearity in the parameters, the adopted control algorithm allows the design of a controller with reduced burden on the determination of uncertainty bounds.

6.1. Passivity-based robust control (PBRC) design

In view of the linear parametrization property of the dynamic model of a manipulator, the dynamic equation (19) can be linearized as

\[
M_1(x)\ddot{x} + C_1(x, \dot{x})\dot{x} + G_1(x) = Y(x, x, \dot{x})\theta = f_c,
\]

where \( \theta \) is a \( m \times 1 \) vector of constant parameters and \( Y \) denotes a \( n \times m \) matrix which is a regressor function of position, velocity and acceleration for the moving platform of the 3-PRC PKM. By
setting
\[ \theta = [\bar{m}_s - \bar{m}_p, m||]^T. \] (46)
then a 3 × 3 regressor matrix Y can be found, whose components are all functions of x and its derivatives do not contain any dynamic parameters.

For the sake of tracking the desired trajectory x_d, the control input is designed as follows:
\[ f_x = \bar{M}_q(x)a + \bar{C}_q(x, \dot{x})v + \bar{G}_q(x) + Kr. \] (47)
where the quantities a, v, and r are defined as
\[ v = \dot{x}_d + \Lambda e, \] (48)
\[ a = v = \dot{x}_d + \Lambda e, \] (49)
and \( r = v - \dot{x} = \dot{e} + \Lambda e, \) (50)
with K and \( \Lambda \) representing diagonal constant matrices of positive gains.

Taking into consideration the linearity property of the dynamic model, the control in Eq. (47) then becomes
\[ f_x = Y(x, x, a, v)\hat{\theta} + Kr, \] (51)
where \( \hat{\theta} \) denotes the estimate for the parameter vector \( \theta \) and can be described by
\[ \hat{\theta} = \theta_0 + \xi, \] (52)
with \( \theta_0 \) denoting a vector of fixed normal parameters and \( \xi \) being a robust term to be determined.

Combining Eq. (47) with Eq. (45) allows the derivation of
\[ \bar{M}_q(x)\dot{r} + \bar{C}_q(x, x)\dot{r} + Kr = Y(\hat{\theta} - \theta). \] (53)

Assign \( \hat{\theta} = \theta_0 - \theta \) be a constant vector describing the parametric uncertainty in the system, then, Eq. (53) becomes
\[ \bar{M}_q(x)\dot{r} + \bar{C}_q(x, x)\dot{r} + Kr = Y(\hat{\theta} - \theta). \] (54)
In the case that the uncertainty is bounded by a constant \( \delta > 0 \), i.e.,
\[ ||\theta|| = ||\theta_0 - \theta|| \leq \delta, \] (55)
the robust term \( \xi \) can be designed as follows [13]:
\[ \xi = \begin{cases} -\frac{\delta}{||\lambda||} & \text{if } ||\lambda|| > \varepsilon, \\ -\frac{\delta}{\varepsilon} & \text{if } ||\lambda|| \leq \varepsilon, \end{cases} \] (56)
where \( \lambda = Y^T r \) and \( \varepsilon > 0 \).

Defining a Lyapunov function candidate for the system in Eq. (54) as
\[ V = \frac{1}{2}r^T M_q(x)r + e^T \Lambda e, \] (57)
it can be demonstrated that the closed-loop system is uniformly ultimately bounded [13] under control law of Eq. (51).

The block diagram for the passivity-based robust control is illustrated in Fig. 13. The guideline for the design parameter of \( \varepsilon \) lies in that a smaller \( \varepsilon \) will lead to smaller tracking errors while the elimination or reduction of chattering requires a larger \( \varepsilon \). Hence, a design tradeoff should be made to yield a better control performance.

6.2. Simulation results and discussions

From the above design process of the control scheme, it can be seen that there are only four control parameters to be designed, which are the constants K, \( \Lambda \), \( \delta \), and \( \varepsilon \). A comparison of the PBRC controller with the former designed RIDC one reveals that finding a constant bound \( \delta \) for the constant vector \( \hat{\theta} \) is much simpler than seeking for a time-varying bound \( \rho \) for \( \eta \) in Eq. (36). Because the bound \( \delta \) relies only on the inertia parameters of the PKM, while \( \rho \) depends on the state error, the desired trajectory, with the addition of some assumptions on the estimated matrices \( \bar{M}_q \) and \( \bar{C}_q \).

For the simulation study, with the normal dynamic parameters of the 3-PRC PKM deviated 20% from the exact values, it is easy to calculate:
\[ ||\theta|| = \sqrt{(0.2\bar{m}_p)^2 + (0.2\bar{m}_p)^2 + (0.2\bar{m}_p)^2} \leq 1.17, \] (58)

hence the uncertainty bound is designed as \( \delta = 1.17 \). In addition, the gain matrices are selected as: \( K = \text{diag}(150) \) and \( \Lambda = \text{diag}(200) \), respectively. The parameter \( \varepsilon \) is chosen as 0.5, which results in almost no chattering and a quick response with the maximum control parameters of \( \varepsilon_{\text{max}}(\varepsilon) = 1.4 \text{ mm/s} \) and \( \varepsilon_{\text{max}}(\varepsilon) = 3.7 \text{ mm/s} \), and steady-state error tolerances of \( e_{\text{tol}}(\varepsilon) = 1.5 \text{ mm} \).
and \( \epsilon_{\text{tol}} = 7.4 \text{ mm/s} \), respectively, after \( t_s = 0.1 \text{ s} \), as shown in Fig. 14.

For the convenience of comparison, the control results from three types of controllers are elaborated in Table 2. Comparing the PBRC control results with those of the IDC, we can see that the maximum control errors for the displacement and velocity tracking have been reduced by 71% and 34%, respectively. Comparing the current results with those from the RIDC with increased control parameters, it is observed that the settling time has been significantly shortened, and both the maximum control error and steady-state error tolerance for the displacement trajectory have been significantly reduced by 64% and 68%, respectively, although the velocity tracking errors have been increased by certain magnitudes. In most cases, given the desired trajectory of the moving platform of the PKM, the control errors for the displacement tracking are the most concerned factor. Overall, the designed PBRC is more practical than the RIDC controller in virtue of simple design procedures and good control performances.

Besides, in view of Table 2, we can observe that the maximum displacement tracking errors from the dynamics control are a few millimeters, which are relatively larger than conventional kinematics-based motion control approach in a general sense. However, the control errors are obtained by a relatively large offset of 20% from the true parameters of the PKM dynamics actually. Under such a worse situation, the kinematics-based controller may probably lead to an even larger control error. Generally, the smaller the uncertainties of dynamic parameters are, the smaller the control errors are.

Furthermore, since the implemented PBRC controller allows certain tolerance on the accuracy of the dynamic parameters, it is interesting to have a knowledge concerning the effects of mass distribution factor \( (w) \) on the control performance in addition to the actuator forces. For instance, as the increasing of the mass factor from 0 to 1, the control performances for the aforementioned trajectory in terms of the maximum and tolerance of position errors are shown in Fig. 15. A comparison of the variation tendencies as shown in Figs. 5 and 15 reveals that the mass factor has similar tendency of effects on actuator forces and control performances. However, as reflected by the variation range of the maximum and tolerance of position errors \( (\epsilon_{\text{max}} \text{ and } \epsilon_{\text{tol}}) \) within 0.1 mm, the influence of mass factor on control performances is trivial than that on the actuator forces. This in part demonstrates the effectiveness of the implemented robust control scheme. Even so, from application point of view, it is necessary to optimize the mass factor with respect to control performances for various trajectories within the workspace of the PKM. Such a meaningful study is expected and planned in the next step of our future research.

7. Conclusions

In this paper, the inverse dynamics modeling and robust controllers design for a 3-PRC PKM have been conducted. By optimizing the mass distribution factor concerning the legs, the simplified dynamic model of the PKM has been established and
validated on a virtual prototype created in ADAMS environment. Based upon the derived model, three types of controllers in terms of inverse dynamics control (IDC), robust inverse dynamics control (RIDC), and passivity-based robust control (PBRC) in task space have been implemented on the virtual prototype through the co-simulation with MATLAB/Simulink and ADAMS.

The simulation results show that the control performances of IDC are degraded if the normal parameters of the PKM are some degree of offsets from the exact values. It has been revealed that the performance of RIDC controller is not obviously superior to the IDC one, and the improvement of its control performance is at the cost of long settling time, which is not preferred for practical applications. In contrast, the control performances of PBRC controller have been significantly improved with comparison to the two former controllers, which are reflected by the fact that the maximum control errors of displacement tracking have been reduced by 71% and 64% with respect to the IDC and RIDC controllers, respectively, along with a shortened settling time. Moreover, the design process of a PBRC controller is greatly simplified with comparison to the RIDC controller. The analysis procedure also exhibits that a controller cannot be arbitrarily applied to a PKM without a careful selection.

The main contribution of this paper is the establishment of simplified inverse dynamic model for a 3-PRC PKM and application of a suitable robust controller for trajectory tracking in the presence of dynamics uncertainties. The obtained results are helpful for the mechatronic design and development of a 3-PRC PKM. In the future work, the treatment of actuator’s dynamics and frictions in passive joints will be conducted once a physical prototype is developed. Moreover, the design and validation methodology presented in this paper can be extended to other types of PKM as well.

Acknowledgments

The authors appreciate the fund support from the research committee of University of Macau under Grant no.: RG065/06-07S/08T/LYM/FST and Macao Science and Technology Development Fund under Grant 069/2005/A.

References


