Teleoperation by using non-isomorphic mechanisms in the master-slave configuration for speed control

Amit Shukla, Member, IEEE, Hamad Karki, Member, IEEE, Laxmidhar Behera, Member, IEEE, Mo Jamshidi, Fellow, IEEE,

Abstract—This paper presents modeling, simulation and control of a novel teleoperated mechanism, where two non-isomorphic manipulators are used in one integrated system, with the purpose of usage in Oil and Gas industry in the future. Overall integration of the teleoperated system has been done in the master-slave configuration, where a stuart type 6-DOF parallel manipulator is used as a master robot and a 6-DOF serial manipulator is used as a slave robot. Since, the parallel manipulator has a close loop structure and serial manipulator is an open loop structure their work spaces are of completely different nature in terms of overall shape and size. A novel task-space mapping mechanism has been proposed in this work to integrate these two completely non-identical manipulators. Damped-least square method is used for overcoming the singularity of proposed task-space mapping matrix. To capture the dynamic response of the teleoperator device, detailed dynamic modeling of the parallel and serial manipulator has been presented. Multilevel control architecture with local actuator-space and joint-space controllers, for the parallel and serial manipulators respectively, has also been presented in this work.

Index Terms—Parallel Manipulator, Serial Manipulator, Teleoperation, Master-slave mechanism, Singularity

I. INTRODUCTION

With ever increasing consumption, easy resources of petroleum products are depleting very fast, whereas new resources situated in extreme conditions are posing serious challenges to safety of human lives and environment. Apart from this, Oil and Gas industry has to cope-up with profitability of business model, optimal usage of resources, and finally increasing production and recovery rate to satisfy soaring global demand [1]. Strategy to meet these critical challenges require radical innovations and optimal usage of available state of art technologies. Oil and Gas industry has lot to learn from successful implementation of robotics and automation for repetitive, dirty and dangerous tasks of manufacturing industry. Tragic events like Deep Horizon oil spill in the Gulf of Mexico [2] has now literally forced Oil and Gas Industry to put automation high on their agenda [3]. Since, materials involved in Oil and Gas industry are highly sensitive and very different in nature compared to manufacturing industry, usage of fully autonomous robotic technology is risky and still farfetched solution. Therefore, humanly supervised semi-autonomous robotic technology like teleoperation finds perfect match of expectations for this industry. Intelligent drilling, under water exploration, smart inspection and manipulation of pipes and tanks, and automated operations for final production are some of the key areas where usage of teleoperated mechanism will not only increase safety standards but also help to increase production as well. Keeping all these issues in mind, present work describes a novel teleoperator mechanism to be used in Oil and Gas industry.

Teleoperation is one of the oldest branches of robotics. Teleoperation word itself means ‘operation at distance’ where brain of the mechanism is located at distance from the site of operation. A close loop teleoperator is a mechanism which has two spatially separated ports interacting with human operator and remote environment simultaneously, while being connected via communication channel [4]. In 1952, Goertz [5] was the first person to propose such tele-operated devices for handling of radioactive materials from behind the shielded wall. This was the first robotic device working on the principle of the master-slave configuration [6]. The local manipulator from which operator is giving command, is called master robot, whereas remote manipulator at working site is called slave robot. Teleoperation mechanism has been already developed to be used for various research and commercial purposes such as space exploration [7]–[9], military operation [10], waste management [11], nuclear environment [12]–[14], explosive material disposal [15]–[17], offshore and undersea operations [18], [19], rescue operation [20]–[22], entertainment [23], [24] and robot assisted surgery [25]–[29].

Mechanically integrated teleoperators of the early days had limitations of distance with poor performance [6], and then emergence of electrically coupled teleoperators overcame both the problems [30]. But with increasing distances came problem of time delay, which led to destabilization of systems [31] and delayed force-feedback [32], therefore subsequently lot of research work had gone in to that direction [33]. Later on introduction of the internet, around mid 1990, has completely removed the distance problem but brought completely new challenges in the form of variable time delays, packet-loss and disconnections [34]. To overcome these challenges passivity theory and scattering approaches has been suggested by researchers [35]. In most of the teleoperated devices, the master-slave manipulators have isomorphic structures [36] and liner dynamic similarity like combination of serial-serial [37] and parallel-parallel [38], therefore master and slave manipulators
always have scalable relationship between their workspaces. Though there are some devices where non-isomorphic manipulators are also put in to master-slave configuration for teleoperation but not much work has been done for teleoperation control of a 6-DOF serial manipulator by a 6-DOF parallel manipulator with full haptic feedback.

Integration of two 6-DOF manipulators, having completely different workspace and structure, in master-slave configuration requires novel solution for workspace mapping. Most of the time proposed solution for such integration is scaling up of master manipulator’s work space to the task space of the slave manipulator. This concept involves direct position to position mapping with clutching [39]. In this mechanism when operator hits the boundary in the task space of the master manipulator, teleoperation is suspended and master end-effector is relocated just like a computer mouse. Though direct task space mapping is intuitive but process of clutching and relocation is highly frustrating and inefficient. Workspace drift control has been proposed by Conti and Khatib in [40] but this technique can be used only when task space of slave manipulator is already known in detail. Space mapping have other problems such as underutilization of the task-space of the slave manipulator as whole space can not be utilized efficiently. This research work presents joint-space mapping for teleoperation, rather than commonly used task-space mapping. When in teleoperation master and slave manipulators are integrated by mapping of joint-space every manipulation trajectory in the task space of the slave manipulator will always have corresponding trajectory in the task space of the master manipulator without ever hitting boundary line. Such joint-space mapping will automatically create a nonlinear mapping between task-space of the master and slave manipulators without any scaling factor even though master and slave task spaces are of completely different shapes and sizes. This novel mapping mechanism will be useful in some of the teleoperation based manipulation tasks of oil and gas industry, such as there are many sensitive operations in oil and gas fields where getting clear visual feedback from task space of the slave manipulator is not possible or remain challenging such as underground or underwater conditions with very low visibility. In such conditions haptic feedback along with full access of the manipulation space becomes very important. The biggest benefit of using joint-space mapping is the fullest access of complete cartesian task space of both the master and slave manipulators without any scaling factor in cartesian space. Simple Jacobian matrices are used to derive the final form of the task-space mapping matrix and further damped least square method is used to solve singularity problem of the matrix inversion. To the best of our knowledge, derivation of such mapping matrix for the teleoperation based control of the manipulator speed has been proposed for the first time. Therefore, to test validity of the integration mechanism and efficacy of its control structure, operator and remote sides (both are placed in same room) are connected directly without any internet and only speed control has been tested though there is capability of haptic feedback as well. This paper is divided in four sections, first and second section will describe modeling and control of the parallel and serial manipulator respectively, third section will describe integration of these two non-isomorphic manipulators by novel mapping technique and overall control mechanism and fourth section will present analysis over tracking results.

II. Parallel Manipulator

First physically realizable parallel mechanism was proposed by Gough and Whitehall [41] and later in 1965 Stewart has proposed a manipulating structure [42] which has 6-DOF and can have general motion in space. From 1960s till now parallel mechanisms have got many applications such as flight simulators, machine tool, parallel robots, precise positioning, VR technologies and medical treatment fields etc [43]. Force-output-to-manipulator-weight ratio of parallel manipulator is increased substantially by using its close-loop and multiple parallel links architecture [44]. Close-loop structure also increases parallel manipulator’s stiffness and accuracy, by averaging effect of end-effector error, in comparison to most of the serial manipulators in-spite of having significantly low inertia. All the methods proposed in literature for solution of dynamics of the parallel manipulator can be placed in three main categories first Lagrangian methods [45]–[47], second Newtown-Euler method [48]–[50] and third Kane’s method [51]. Though Euler-Lagrangian formulation is well structured and can be expressed in close form but requires large amount of symbolic and numeric computations to derive and evaluate partial derivatives of the Lagrangian [52]. Newton-Euler approach requires computations of all the constraints and moments between the links though these calculations are not involved in the control of manipulator [51]. Whereas Kane’s method based derivations for dynamic model is a kind of combination of both Lagrange and Newton-Euler method, which inherits all the advantages of these methods but still remains free from above mentioned disadvantages [53].

A. Dynamic analysis of the Parallel Manipulator

In the parallel manipulator also known as stewart platform, base platform and end-effector (movable platform) are connected by multiple linear mechanisms. Fig. 1(b) presents schematic diagram of a 6-DOF parallel manipulator having a fixed base, a movable platform and a extendable actuator. To fully describe the motion of movable platform, an inertial reference frame is fixed at origin shown in Fig. 1(b). Actuators are used to closely connect movable platform and fixed base frame. Upper and lower parts of the actuators are connected to movable platform and base frame respectively with coordinate points \( s_i \) and \( B_i \). Where parameter subscript \( i \) stands for \( i^{th} \) actuator. When actuators change their lengths in synchronized way there is a change in position of the movable platform. This change can be calculated by knowing new joint coordinates of the actuators and movable platform.

Coordinates of the new position of actuator’s upper joint is \( s_i’ = (s_{ix}, s_{iy}, s_{iz}) \) after parallel displacement \((x, y, z)\) and rotation \((\phi, \theta, \psi)\) along and around \(x, y, z\) axes, respectively from its initial position \( s_i = (s_{ix}, s_{iy}, s_{iz}) \). It can be written in
terms of rotation matrix as follows:

\[
\begin{pmatrix}
\frac{\partial x_i}{\partial a_{f,i}} \\
\frac{\partial y_i}{\partial a_{f,i}} \\
\frac{\partial z_i}{\partial a_{f,i}}
\end{pmatrix} = R \begin{pmatrix}
\frac{\partial x}{\partial a_{f,i}} \\
\frac{\partial y}{\partial a_{f,i}} \\
\frac{\partial z}{\partial a_{f,i}}
\end{pmatrix}
\]  

(1)

Where \( R \) is the rotation matrix (54) and index \( i = 1, 2, ..., 6 \).

\[
R = \begin{pmatrix}
c\psi c\theta & c\psi s\theta s\phi - s\psi c\phi & c\psi s\theta c\phi + s\psi s\phi \\
c\psi s\theta & s\psi s\theta s\phi + c\psi c\phi & s\psi s\theta c\phi - c\psi s\phi \\
-s\theta & c\theta s\phi & c\theta c\phi
\end{pmatrix}
\]

Where \( c = \cos \) and \( s = \sin \)

Fig. 1. (a) Six-DOF parallel manipulator used in present research work, (b) Vector representation of manipulator with one actuator and (c) 6-DOF serial manipulator PUMA 560.

This parallel manipulator has six independent links to enable six freedom of motion \((x, y, z, \phi, \theta, \psi)\) for the movable platform. Fig. 1(a) presents a picture of 6-DOF parallel manipulator used in present work.

\[
\dot{q} = \left[ \dot{x}, \dot{y}, \dot{z}, \dot{\phi}, \dot{\theta}, \dot{\psi} \right]^T
\]

(3)

Where \( \dot{q} = [\dot{x}, \dot{y}, \dot{z}]^T \) is translational velocity of the end effector and \( \omega \) is the generalized rotational velocity of the end effector. Relationship between \( \omega \), Euler angles \((\phi, \theta, \psi)\) and is defined as follows:

\[
\omega = \begin{pmatrix}
\omega_x \\ \omega_y \\ \omega_z
\end{pmatrix} = R \begin{pmatrix}
1 & 0 & -s\theta \\
0 & c\phi & c\theta s\phi \\
0 & -s\phi & c\theta c\phi
\end{pmatrix} \begin{pmatrix}
\dot{\phi} \\ \dot{\theta} \\ \dot{\psi}
\end{pmatrix}
\]  

(4)

Actuator vector is defined by following expression:

\[
l_i = L_i \cdot a_{i,n} = u + R s_i^T - B_i,
\]

(5)

Where \( a_{i,n} \) and \( L_i \) are the unit vector and actuator length along the \( i \)-th actuator respectively, \( u \) is the end-effector vector, \( s_i^T \) is the position vector of joint with the \( i \)-th actuator in upper platform frame and \( B_i \) is the vector representing joint of the \( i \)-th actuator in the base frame.

Relationship between infinitesimal displacements of the movable platform and the actuators can be derived by taking derivative of (5):

\[
\delta L_i \cdot a_{i,n} + L_i \cdot \delta a_{i,n} = \delta u + \delta R \cdot s_i^T + R \cdot \delta s_i^T - \delta B_i \]  

(6)

Where \( \delta B_i = \delta s_i^T = 0 \) and \( \delta R \cdot s_i^T \) can be expressed as:

\[
\delta R \cdot s_i^T = \hat{\Omega} \cdot R \cdot s_i^T = \begin{pmatrix}
0 & -\delta\Omega_y & \delta\Omega_z \\
\delta\Omega_z & 0 & -\delta\Omega_x \\
-\delta\Omega_y & \delta\Omega_x & 0
\end{pmatrix} \cdot R \cdot s_i^T = \delta\Omega \times R \cdot s_i^T
\]  

(7)

Where \( \delta\Omega = (\delta\phi, \delta\theta, \delta\psi) \) is the infinitesimal rotation of the movable platform.

Now (6) can be written as follows:

\[
\delta L_i \cdot a_{i,n} + L_i \cdot \delta a_{i,n} = \delta u + \delta R \cdot s_i^T - \delta B_i
\]  

(8)

By taking inner product of (8) with unit vector \( a_{i,n} \):

\[
a_{i,n} \cdot a_{i,n} \cdot \delta L_i + a_{i,n} \cdot \delta a_{i,n} \cdot L_i = a_{i,n} \cdot \delta u + a_{i,n} \cdot (\delta \Omega \times R \cdot s_i^T)
\]

(9)

Where \( a_{i,n} \cdot a_{i,n} = 1 \) and \( a_{i,n} \cdot \delta a_{i,n} = 0 \) and once again (9) can be simplified as:

\[
\delta L_i = a_{i,n} \cdot \delta u + (R \cdot s_i^T \times a_{i,n}) \cdot \delta \Omega
\]

(10)

Unit vector along the actuator length can be described as follows:

\[
a_{i,n} = \begin{pmatrix}
(s_i^{f_x} - B_{ix})/L_i \\
(s_i^{f_y} - B_{iy})/L_i \\
(s_i^{f_z} - B_{iz})/L_i
\end{pmatrix}
\]

(11)

Where \( B_i \) is a constant vector as joints at the base frame are fixed.

Equation (10) can be further expanded for all the six actuator links in a more compact and useful form as follows:

\[
\delta X_p = [J_p]^{-1}\delta X_p
\]

(12)

Where \( \delta X_p = (\delta u^T, \delta \Omega T)^T \) is the infinitesimal movement of the platform and \( \delta L_p = (\delta L_1, ..., \delta L_6)^T \) is the infinitesimal movement of the six actuators. This relationship will be further used in mapping the task-spaces of the parallel and serial manipulators for which more details will be given in control section of this paper.

There is a relationship between every upper joint velocity \((v_{s,i})\) and generalized velocity \((\dot{q})\) of the end-effector and it can be described in terms of Jacobian as follows:

\[
v_{s,i} = \dot{u} + \omega \times (Rs_i) = J_{s,i} \dot{q},
\]

(13)
B. Dynamics of the actuators

Upper platform of the manipulator is supported by six actuator links and these actuators are driven by the core-less DC-motors. Every actuator in its configuration has two parts, the first is a lower part which is a lead ball screw connected to the motor rotor by a synchronous belt and the second is a upper part, which is a hollow cylinder with groves at inner surface. With every rotation of lower screw part (rotating part), upper part (actuating part) of the actuator moves forward or backward depending on the direction of rotation. Rotational speed of the motor rotor and related actuator velocity has the following relationship:

\[ v_{a,i} = \frac{L \cdot \eta}{60} \cdot \omega_{m,i}, \quad (14) \]

Where \( v_{a,i} \) is the translational velocity of the actuator along the actuator length, \( \eta \) is the rotational efficiency, \( \gamma \) is the gear reduction ratio and \( \omega_{m,i} \) is the rotational speed of the motor.

Translational velocity of the actuator can also be described in terms of generalized platform velocity:

\[ v_{a,i} = a_{i,n}^T \cdot v_{s,i} = J_{\theta_{i,q}} \dot{q}_{i}, \quad (15) \]

Where \( J_{\theta_{i,q}} \) is the Jacobian between actuator velocity and generalized velocity of the end-effector.

Angular acceleration \( \alpha_{m,i} \) of the motor rotor can be defined further as:

\[ \alpha_{m,i} = \dot{\omega}_{m,i} = K(v_{a,i} + \dot{a}_{i,n}^T v_{s,i}) \]
\[ = K(a_{i,n}^T \dot{v}_{s,i} + a_{i,n}^T v_{s,i}) \frac{(I - a_{i,n} a_{i,n}^T)}{||l_i||} v_{s,i}, \quad (16) \]

Where \( \frac{(I - a_{i,n} a_{i,n}^T)}{||l_i||} = P_{a,n} \) is the projection to the plane with normal vector \( a_{i,n} \) and \( K \) is the transmission ratio between motor rotor angular velocity \( (\omega_{m,i}) \) and actuator’s translational velocity \( (v_{a,i}) \).

The acceleration of the upper actuator joint can be described by further differentiating (13):

\[ \ddot{v}_{a,i} = \ddot{u} + \dot{\omega} \times (R \dot{s}_i) + \omega \times (\omega \times (R \dot{s}_i)) \]
\[ = J_{a,i,q} \ddot{q} - ||\omega||^2 P_{\omega} a_i, \quad (17) \]

Where \( P_{\omega} = (I - \omega, \omega) \).

Angular velocity of the actuator perpendicular to the actuator length can be described as follows:

\[ \omega_{s,i} = a_{i,n} \cdot \frac{v_{s,i}}{||l_i||}, \quad (18) \]

If rotating part of the actuator has center of gravity (COG) at \( rc \) and the actuating part has COG at \( ac \), velocities of these COG’s can be described, in terms of upper joint velocity \( v_{s,i} \), as follows:

\[ v_{rc,i} = \omega_{s,i} \times (r_{rc} \cdot a_{i,n}) = J_{rc,s,i} v_{s,i}, \quad (19) \]
\[ v_{ac,i} = v_{s,i} + \omega_{s,i} \times (-r_{ac} \cdot a_{i,n}) = J_{ac,s,i} v_{s,i}, \quad (20) \]

Since acceleration of the COGs of the actuator parts also generate inertial forces there calculation is important and should be written in terms of platform motion:

\[ \ddot{v}_{ac,i} = \frac{d}{dt} v_{ac,i} = \frac{d}{dt} (J_{ac,s,i} v_{s,i}) = J_{ac,s,i} \ddot{v}_{s,i} + J_{ac,s,i} \dot{v}_{s,i}, \quad (21) \]

Differentiation of the Jacobian \( J_{ac,s,i} \) can be calculated as follows:

\[ J_{ac,s,i} = \frac{d}{dt}(I - \frac{a_{i,n}}{||l_i||} P_{a,n}) = \frac{r_{ac}}{||l_i||^2} (v_{s,i} a_{i,n} P_{a,n}) \]
\[ + \frac{r_{ac}}{||l_i||^2} (P_{a,n} v_{s,i} a_{i,n}^T + a_{i,n} v_{s,i}^T P_{a,n}), \quad (22) \]

When inertia of the load torque is high in comparison to light actuators, inertial forces due to actuators can be neglected but when both are comparative, effect of the actuator forces can not be ignored anymore, as in present case of the experimental manipulator setup shown in Fig. 1(a). The gimbals are assumed to rotate frictionlessly and inertia of actuator around actuating axis is negligible. There are two basic parameters to define both the parts of the actuator, the first is the mass and the inertia around the axis orthogonal to the actuating direction of the actuator. For rotating and actuating part these parameters will be \((m_{rc}, I_{rc})\) and \((m_{ac}, I_{ac})\), respectively.

The inertial forces of the actuators mainly have three parts, the inertial mass forces, the influence of the inertia and overall gravitational force. Finally all these forces are projected in the direction of the generalized velocities of the upper joints of the actuators with the platform to be accommodated in the overall dynamics of the system.

\[ f_{as,i} = J_{ac,s,i}^T m_{a} \ddot{v}_{ac,i}, \quad \]
\[ = J_{ac,s,i}^T m_{a} J_{ac,s,i} \ddot{v}_{s,i} + J_{ac,s,i}^T m_{a} J_{ac,s,i} \dot{v}_{s,i}, \]
\[ = M_{as,i} \ddot{v}_{s,i} + C_{as,i} v_{s,i}, \quad (23) \]

Now total inertial forces at upper gimbal points due to movement of inertias of rotating and actuating parts can be derived.
by using Lagrangian method and resultant relationship can be described as follows [55]:
\[
f_{s,i,r,i} = \frac{\left( a_i \right) \left( \| a_i \| \right)^2}{\| a_i \|} \left[ P_{a,i,n} v_{s,i} + \left( a_i \right)^T \left( \| a_i \| \right) P_{a,i,n} v_{s,i} \right], \\
= M_{s,i} \dot{v}_{s,i} + C_{s,i} v_{s,i}, \\
(24)
\]
At last the gravitational forces on the actuating part and rotating part are calculated and they are projected along the generalized velocities of the platform:
\[
f_{a,g,i}^s = J_{ac,i} m a g, \\
f_{r,g,i}^s = J_{irc,i} m g, \\
(25, 26)
\]

C. Influence of the motor systems’ dynamics

All the actuators are actively driven by lead-ball screw, which in turn driven by the core-less DC-motors. Rotating part of the actuator is connected to the DC-motor by a synchronous belt. When motor rotor rotates around its axis it also forces screw to rotate around its own axis. Since, motor is fixed on the actuator, so along with this self rotation, motor also rotates with rotating actuator part. For the first kind of rotation combined inertia of the motor rotor and screw (snail) is defined as \( i_{ms} \) in motor’s rotor reference frame, for the second kind of motion total mass and inertia are defined as \( i_{sm} \) and \( i_{sm} \) in the reference frame of screw respectively.

The inertial torque generated by the motion of the motor and actuator can be described as follows:
\[
T_{ms,i} = i_{ms} \alpha_{ms,i}, \\
= i_{ms} K \left( a_{i,n}^T v_{s,i} + v_{s,i}^T \left( I - a_{i,n} a_{n}^T \right) \| a_{i,n} \| \right) v_{s,i}, \\
= M_{ms,i} \dot{v}_{s,i} + C_{ms,i} v_{s,i}, \\
(27)
\]
Where \( M_{ms,i} = i_{ms} K \left( a_{i,n}^T a_{i,n} \right) \| a_{i,n} \| \) and \( C_{ms,i} = i_{ms} K \cdot v_{s,i}^T \left( I - a_{i,n} a_{n}^T \right) a_{i,n} \| a_{i,n} \| \). This inertial torque can be projected along the generalized velocities of the platform as follows:
\[
f_{ms,i}^s = J_{ac,i}^T T_{ms,i} = J_{ac,i}^T M_{ms,i} \dot{v}_{s,i} + J_{ac,i}^T C_{ms,i} v_{s,i}, \\
(28)
\]
The inertial forces due to combined movement of inertias of motor and actuator can be described at the upper joint of the platform as follows:
\[
f_{ms,i}^s = \frac{i_{ms}}{\| i_{ac} + i_{rc} \|} \left( M_{ia,ir,i} \dot{v}_{s,i} + C_{ia,ir,i} v_{s,i} \right), \\
= M_{ims,i} \dot{v}_{s,i} + C_{ims,i} v_{s,i}, \\
(29)
\]
Finally, calculation of inertial force due to gravity can be defined as follows:
\[
f_{ms,g,i}^s = J_{irc,s,i}^T m_{ms,i} g, \\
(30)
\]

D. Multi-body dynamics of the Parallel Manipulator using Kane’s equations

The principle of virtual work is an efficient method for developing dynamic equations for inverse dynamics of the parallel manipulator as shown in [56], [57]. Every external force applied on the body can be decomposed into a force passing through the center of gravity of the system and a moment around it. And then by applying principle of virtual work, Kane’s equation for the parallel manipulator has been derived, where sum total of active forces/torques and inertial forces/torques are zero [58], [59].
\[
F_{ac} + F_{in}^s = 0, \\
M_{ac} + M_{in} = 0 \\
(31, 32)
\]
Where \( F_{ac} \) and \( F_{in}^s \) are the generalized active forces and inertial forces, respectively applied on the manipulator’s platform in the direction of generalized velocities. This can be further elaborated as follows [53]:
\[
F_{ac} = A_n \cdot F_a + m_p \cdot G \\
F_{in} = -m_p \cdot \ddot{u} \\
(33, 34)
\]

For angular motion of the platform active moment and inertial moment are calculated in the base frame:
\[
M_{ac} = R S^p \times A_n \cdot F_a \\
M_{in} = -I_p \cdot \dot{\omega} - \omega \times I_p \cdot \omega \\
(35, 36)
\]
By combining all the equations from (31) to (36) complete dynamic model of the 6-DOF parallel manipulator can be derived as follows:
\[
\begin{bmatrix}
A_n \\
R S^p \times A_n
\end{bmatrix} F_a = \begin{bmatrix}
m_p I_6 & 0 \\
0 & I_p
\end{bmatrix} \begin{bmatrix}
\ddot{u} \\
\dot{\omega}
\end{bmatrix} \\
\begin{bmatrix}
0 & 0 \\
0 & \omega \cdot I_p
\end{bmatrix} \begin{bmatrix}
\ddot{u} \\
\dot{\omega}
\end{bmatrix} + \begin{bmatrix}
m_p G
\end{bmatrix} \\
(37)
\]
In short, (37) can also be written in following form:
\[
J q = M(p(\chi) \dot{q} + C(p(\chi), \dot{q}) q + G(p, \dot{q}) \\
(38)
\]
Where \( F_a \) is the active force vector applied on the platform by the six actuators, \( R \) is the rotation matrix, \( A_n = (a_{1,n}, a_{2,n}, ..., a_{6,n}) \) is the coordinate matrix of the normal actuator vectors, \( S^p \) is the coordinate matrix of the actuator joints on the platform \( S^p = (s_1, s_2, ..., s_6) \), \( M(p) \) is the mass matrix of the movable platform in the base frame, \( m_p \) is the mass of the upper platform, \( \dot{\omega} \) is the skew symmetric matrix related to angular velocity \( \omega \), \( I_p = I_{R p} R_p^T \) is the inertia matrix transformation with \( I_p \) as the inertia matrix in the platform frame, \( C(p, \dot{q}) \) contains centripetal and Coriolis force components involved in the platform movement and \( G(p) \) is the overall gravitational force on the moving platform.

E. Complete dynamics of the parallel manipulator including actuators

Total inertial forces generated by the motion of actuators and motors can be described as follows:
\[
f_i^s = \sum_{i=1}^{6} J_{i,s,i} q \left( f_{a,i,s,i}^s + f_{a,i,ir,i}^s + f_{a,g,i}^s + f_{r,g,i}^s \right) \\
+ f_{ms,i}^s + f_{ims,i}^s + \sum_{i=1}^{6} f_{ms,i}^s \\
(39)
\]
It can be observed from (38) that left-hand-side of the equation has all the active forces and right-hand-side of the equation...
has inertial forces. After combining all the inertial forces, generated due to motion of the actuators and motor systems, total inertial forces applied on the parallel manipulator can be written as follows [59]:

\[ f^* = f^*_l + M_p(\chi)\dot{q} + C_p(\chi, \dot{q})\ddot{q} + G_p. \]  

Final equation describing the dynamics of the parallel manipulator can be described as follows:

\[ J_{\theta, i}(\chi)F_a = M_{tp}(\chi)\dot{q} + C_{tp}(\chi, \dot{q})\ddot{q} + G_{tp}, \]  

Where \( F_a, M_{tp}, C_{tp} \) and \( G_{tp} \) are defined as follows:

\[ F_a = \frac{2\pi}{T_p} \cdot \gamma \cdot \eta T \cdot T_{em}, \]

\[ M_{tp} = M_p + \sum_{i=1}^{6} \left( J_{s, i}^T C_{a, i} + M_{a, i} + M_{ms, i} \right) J_{s, i}, \]

\[ + \sum_{i=1}^{6} J_{s, i}^T M_{ims} J_{s, i}, \]  

\[ C_{tp} = C_p(\chi, \dot{q})\dot{q} + \sum_{i=1}^{6} J_{s, i}^T C_{a, i} + C_{a, i} + C_{ms, i} J_{s, i}, \]

\[ + \sum_{i=1}^{6} J_{s, i}^T C_{ims} J_{s, i} \dot{q} - ||\omega||^2 \sum_{i=1}^{6} J_{s, i}^T M_{ims} P_\omega s_i \]

\[ - ||\omega||^2 \sum_{i=1}^{6} \left( J_{s, i}^T \left( M_{a, i} + M_{ia, i} + M_{ms, i} \right) \right) P_\omega s_i, \]

\[ G_{tp} = G_p + \sum_{i=1}^{6} J_{s, i}^T (f_{s, i}^* + f_{s, i}), \]

Where \( \tau \) is the joint torque vector, \( M(q) \) is the inertia matrix also called as positive definite inertia matrix, \( q \) is the joint-space-variable vector (six rotational joints of the PUMA 560), \( \dot{q} \) is the joint acceleration vector, \( B(q) \) is the matrix of Coriolis torques, \( C(q) \) is the matrix of Centrifugal torque, \( \dot{q}\ddot{q} = [\dot{q}_1 \cdot \dot{q}_2 \cdot \dot{q}_3 \cdot \ldots \cdot \dot{q}_n \cdot \ddot{q}_2 \cdot \ddot{q}_3 \cdot \ldots \cdot \ddot{q}_n]^T \) is vector of joint velocity, \( \ddot{q}^2 = [\dot{q}_1^2 \cdot \dot{q}_2^2 \cdot \dot{q}_3^2 \cdot \ldots \cdot \dot{q}_n^2]^T \) and \( G(q) \) is the gravity torque vector.

In an actuator driven manipulator, actuator torques are the inputs signals and outputs are the actual joint displacements, therefore solution of \( 46 \) can be described in the following form:

\[ \dot{q} = M^{-1}(q)(\tau - B(q)(\dot{q}\ddot{q}) - C(q)\dot{q}^2 - G(q)) \]  

There are three important assumptions, which have been made for the purpose of simplifying the calculation of matrices \( M(q), B(q), C(q) \) and \( G(s) \) and these assumptions are as follows: the rigid body assumption, sixth link has been assumed to be symmetric, that is \( I_{xx} = I_{yy} \) and only the mass moments of inertia are considered, that is \( I_{xx}, I_{yy} \) and \( I_{zz} \). Complete expressions of these matrices have been described in [61].

IV. OVERALL CONTROL ARCHITECTURE OF THE TEOLEOPERATED MECHANISM

In general every teleoperated mechanism has three main components, first operator side, second remote side and third communication channel between these two distant ports. On the operator side, human operator interacts with master manipulator to give commands to the slave manipulator through communication channel. Using internet as a communication channel creates various other challenging problems such as instability due to network delay. Therefore, in present work these ports are connected directly without any internet to demonstrate working principle of newly proposed teleoperated device, where involved manipulators have completely non-identical structure and fundamentally different operating principle.

Before integrating, parallel and serial manipulator in the master-slave configuration, PID controller based multi staged control mechanism has been implemented for both the manipulators individually. And efficacy of the control mechanisms have been tested by supplying them with suitable trajectories in the task-space and joint-space, respectively. Validation of the standalone trajectory-tracking performances have been done by comparing the obtained results with already published data for each manipulator. Since, in both the manipulators all the actuators are driven by the DC-motors, suitable control mechanism for the DC-motor has been given in the next section. Subsequently, standalone control of the parallel and serial manipulator have been described in successive sections and at last overall control of the integrated teleoperated device has been discussed in detail. Fig. 13 and 14 presents schematic diagrams of the parallel manipulator, serial manipulator and finally integrated teleoperator system, respectively, along with their multi-stage controllers for the actuators.
A. Control mechanism for the DC motor

Due to speed controllability and well behaved torque characteristics, DC-servo motors are the most widely used actuators for automated devices. Every actuator of the parallel and serial manipulator is driven by a DC-motor. Electromagnetic torque $T_{em}$ of the motor is produced by circulating armature current $i_a$ in its stator windings and they are related as follows:

$$T_{em} = K_e i_a$$  \hspace{1cm} (48)

Similarly, induced emf $e_a$ depends only on the motor speed $\omega_m$ and is related to each other by voltage-constant $K_e$:

$$e_a = K_e \omega_m$$  \hspace{1cm} (49)

When $v_a$ is applied on the motor windings, it overcomes induced emf $e_a$ and causes current $i_a$ to flow in the motor windings. By applying Kirchhoff’s voltage law following equation can be written:

$$v_a = e_a + R_a i_a + L_a \frac{di_a}{dt}$$  \hspace{1cm} (50)

On the mechanical side, electromagnetic torque $T_{em}$, produced by the motor, overcomes the mechanical torque $T_L$ to produce acceleration:

$$\frac{d\omega_m}{dt} = \frac{1}{J_{eq}} (T_{em} - T_L)$$  \hspace{1cm} (51)

Where $J_{eq}$ is the equivalent inertia of the DC-motor and the mechanical load. The lengths of the actuators keep on changing with motions of the actuator motors, therefore the equivalent inertia $J_{eq}$ will be keep on varying with time and which is very difficult to continuously calculate in real time.

B. Control of the parallel manipulator

Actuators driven by the DC-motors are sole source of active forces and torques applied on the upper platform of the parallel manipulator. Since, in the motor driven parallel manipulator, calculation of time varying equivalent inertia $J_{eq}$ in (51) is very difficult in real time, therefore, instead of directly calculating $J_{eq}$ in the DC-motor frame, torque-force balance equation is written in the parallel manipulator’s reference frame and electromagnetic torque, produced by the DC-motors, is just taken as input vector in to the dynamic equation (11) derived for the parallel manipulator.

Overall control architecture of the parallel manipulators can be described in three main stages, position control of the motor rotor position ($\theta_{m}$), speed control of the motor rotor ($\omega_m$) and current control in the motor windings ($i_a$). Fig. 3 presents schematic diagram of the parallel manipulator along with its three stage controllers for the actuators.

At first a desired cartesian trajectory with surge, sway, heave, roll, pitch and yaw is defined. Then by using inverse kinematics of the parallel manipulator this trajectory given in the task space of the manipulator is converted in to the joint space of the manipulator. Here joint space of the manipulator is actuators’ translational speeds in the direction of their actuation. From speed of the actuator, speed of the corresponding motor rotor is calculated which is further integrated to calculate the angular position of the motor rotor. This desired position of the motor rotor ($\theta_d$) is then compared against actual position of the motor rotor ($\theta_m$). The difference between desired and actual rotor position ($\delta\theta = \theta_d - \theta_m$) is passed through a PID based position controller to calculate desired speed ($\omega_d$) of the motor rotor:

$$\omega_d = (K_{p1} + K_{i1} \int dt + K_{d1} \frac{d}{dt}) \{\theta_d - \theta_m\}$$  \hspace{1cm} (52)

This desired speed ($\omega_d$) is compared against actual speed ($\omega_m$) of the motor rotor and again a PID based speed controller is used to calculate the desired current in the motor windings.

$$i_d = \frac{T_{ref}}{K_t} = (K_{p2} + K_{i2} \int dt + K_{d2} \frac{d}{dt}) \{\omega_d - \omega_m\}$$  \hspace{1cm} (53)

This desired current ($i_d$) is compared against the actual current ($i_a$) flowing in the motor windings. A PID based current controller is operated on the current difference to generate
reference voltage to be applied on the windings to generate required current.

\[ v_d = (K_{p3} + K_{i3} \int dt + K_{d3} \frac{d}{dt}) (i_d - i_a) \]  

(54)

When this current \( i_a \) flows through motor windings, it generates suitable electromagnetic torque \( T_{em} \), which is further supplied to the dynamic equation of the manipulator. Each actuator has three control loops, and there are total six actuators so there are total eighteen control loops. Complete process of the control is explained schematically in Fig. 3 and mathematically in equations (41 - 45). Since, complete dynamics of the 6-DOF parallel manipulator is highly non-linear along with strong coupling terms and time varying parameters, PID controllers by themselves were unable to mitigate tracking errors. Therefore, to improve performance of the tracking controllers, Su, et al., [62] have proposed usage of nonlinear PID control based technique. With same purpose in [63] researchers have proposed novel feedback linearization mechanism leading to some improvement, but still overall results were not very satisfactory. In [63], model based improved controller has been used, which essentially modifies required torque reference to the PMSM motors. In contrast to these approaches, in present work, one lag-compensator for each actuator leg has been proposed in addition to the three staged PID controllers, to drastically improve tracking performance.

By using lag-compensators [65], desired rotational speed of the motor (\( \omega_d \)), as seen in Fig. 3, is modified as follows:

\[ \omega_d = \frac{1 + T_{1s}}{1 + T_{2s}} (K_{p1} + K_{i1} \int dt + K_{d1} \frac{d}{dt}) (\theta_d - \theta_m) \]  

(55)

Usage of lag-compensators, to modify the desired rotational speed of the DC-motors, has significantly improved the tracking performance of the controller with minimization of tracking error. This novel approach gives even better results in comparison to the approach presented in [64].

C. Control mechanism of the PUMA 560

There are mainly two categories of trajectory tracking schemes for the serial-manipulator, task-level servo schemes and joint-level servo schemes. In the present work for the local control of serial manipulator joint-level servo scheme has been used because both the manipulators, master and slave (PUMA 560) have 6-DOF joint-space and it will be very easy to integrate them via their joint-spaces. The biggest benefit of using this kind of integrated control structure is that it also allows us to bypass the difficulty of solving inverse kinematic of the serial manipulator in the master-slave teleoperation.

PUMA 560 has six rotatory joints and all the joints are driven by DC-motors of different parameters, which has been described in [66] and presented here in the Table I. Modeling and control of the DC-motors has been already described in Section IV-A. The joint-space control for the PUMA 560 is implemented here with three layers of PID controllers. First of all a joint-space trajectory, in the form of joint-angle variations with respect to time, will be supplied to the manipulator control. At the stage of position control, the difference between actual joint angle (\( \theta_{sa} \)) and desired joint angle (\( \theta_{sd} \)) will be calculated and this difference will be passed through a PID controller to further generate desired speed reference.

\[ \omega_{sd} = (K_{sp1} + K_{si1} \int dt + K_{sd1} \frac{d}{dt}) (\theta_{sd} - \theta_{sa}) \]  

(56)

Now difference between desired (\( \omega_{sd} \)) and actual speed (\( \omega_{sa} \)) of the joints will be calculated and this difference will be further pass through a PID controller to generate desired current reference.

\[ i_{sd} = \frac{T_{ref}}{K_{st}} = (K_{sp2} + K_{si2} \int dt + K_{sd2} \frac{d}{dt}) (\omega_{sd} - \omega_{sa}) \]  

(57)

Once again actual current (\( i_{sa} \)) will be measured and will be compared with desired current reference (\( i_{sd} \)) and this difference will be passed through last PID controller to generate required voltage \( v_s \) which in turn will generate suitable electromagnetic torque (\( T_{se} \)) for tracking given desired joint-space trajectory.

\[ v_s = (K_{sp3} + K_{si3} \int dt + K_{sd3} \frac{d}{dt}) (i_{sd} - i_{sa}) \]  

(58)

Similar three staged controller will be implemented for all the six rotatory joints of the serial manipulator. Next section will describe integration and over all control of the parallel and serial manipulator in the master-slave configuration.

D. Integrated control of the teleoperated device

In the present work, overall integrated teleoperator mechanism has three broad layers of control architecture, which are described schematically in Fig. 5. Two main components of the teleoperation device, namely the master and slave manipulator, have their own stand-alone control mechanisms. These local control loops are further divided in three stages, position control, speed control and finally torque control (actually current control) as described in previous sections.

The master manipulator, which is a parallel manipulator, has six actuators in the form of six links, whereas the slave manipulator, which is a serial manipulator, has serially connected six-rotatory joints. To drive serial manipulator from commands of the parallel manipulator in the master-slave configuration basically requires integration of their joint spaces with each other. In principle, there can be two ways to integrate the joint-spaces of these two manipulators:

- Each actuator of the parallel manipulator is directly mapped with one rotatory joint of the serial manipulator and in return mapping between dissimilar cartesian workspaces of the master and slave manipulators will be calculated indirectly.
Cartesian work-spaces of these two manipulators are directly mapped on each other while relation between the joint-space variables of both the manipulators will be calculated indirectly.

Both the master and slave manipulators have six degrees of freedom but they have completely non-identical work-spaces (cartesian task-spaces) and also have different types of singularities (which has to be avoided). Therefore, direct mapping of these work-spaces will not be an easy working solution. Another problem with such direct cartesian-space mapping is that task-space of the slave manipulator can not be utilized optimally as task space of the master manipulator is very different in terms of overall shape and size. Whereas in the first method, where each actuator of the parallel manipulator has been mapped on the one rotatory joint of the serial manipulator, will have two following important benefits:

- The first benefit is that slave manipulator will be able to access its complete cartesian task space and during the operation slave manipulator will never hit its boundary so frequent disengagements between master and slave manipulator will not be required anymore.
- The second benefit is that this method of integration will be computationally simple and calculation of inverse kinematics of the serial manipulator will be avoided.

The basic integration mapping mechanism can be described by following expression:

$$\partial L_p = [J_p]_{6 \times 6}^{-1} \partial X_p$$

(60)

Where joint-space vector $L_p$ of the parallel manipulator is combination of six actuators’ lengths and described as $L_p = \{L_1, L_2, L_3, L_4, L_5, L_6 \}$ $L_p$ and joint-space of the serial manipulators is described as a vector of six joint angles in the form of $\theta_s = \{\theta_1, \theta_2, \theta_3, \theta_4, \theta_5, \theta_6 \}$. Both of these spaces are mapped on each other by a diagonal matrix $[K]_{6 \times 6}$. Elements of this matrix can be calculated easily by establishing a relationship between the maximum possible variations of each actuator’s length of the parallel manipulator and respective variations of the joint-angles of the serial manipulator.

Once both of these manipulators are integrated as one system, a desired trajectory is given in the task-space of the serial manipulator, which then will be converted in to a trajectory in the task-space of the parallel manipulator. For successful tracking of every given trajectory in the serial-task-space $X_s = \{x_s, y_s, z_s, \alpha, \beta, \gamma \}$ there should be always an unique corresponding trajectory in the parallel-task-space $X_p = \{x_p, y_p, z_p, \phi, \theta, \psi \}$. This calculated trajectory in the parallel-task-space is used as an input to the control mechanism of the parallel manipulator. By using inverse kinematics of the parallel manipulator, variations in the actuators’ lengths can be obtained by following expression:

$$\partial X_s = [J_s]_{6 \times 6} \partial \theta_s$$

(62)

The Jacobian matrix $[J_s]_{6 \times 6}$ relates the linear and angular velocities of the manipulator tip to the angular velocities of the joints.

$$\begin{pmatrix} v \\ \omega \end{pmatrix} = [J_s]_{6 \times 6} \begin{pmatrix} J_1(\theta_s), J_2(\theta_s),...,J_6(\theta_s) \end{pmatrix} \dot{\theta}_s(t)$$

(63)

For the PUMA 560 manipulator, which is a 6R type robot, $[J_s]_{6 \times 6}$ can be defined as follows:

$$J_s(\theta_s) = \begin{bmatrix} z_0 & z_1 & \cdots & z_5 \\ z_0 \times p_{0,6} & z_1 \times p_{1,6} & \cdots & z_5 \times p_{5,6} \end{bmatrix}$$

(64)

Column vectors of the $[J_s]_{6 \times 6}$ can be defined as follows:

$$J_1(\theta_s) = [S_1(C_{23}C_{45}S_{56} + S_{23}(d_4 + C_5d_6) + a_2C_2) - C_1(S_4S_5d_6 + d_2), ~ C_1(C_{23}C_{45}S_{56} + S_{23}(d_4 + C_5d_6) + a_2C_2) - S_1(S_4S_5d_6 + d_2), ~ 0, ~ 0, ~ 0, ~ 1]^T$$

(65)
Many control algorithms fails because physically in such near singular means having zero determinant. Near singularities, one very widely used solution to overcome this difficulty is to implement the singularity avoidance technique into a low-level control algorithm. Among many such methods, for overcoming singularity problem, are pseudo-inverse, damped least-square (DLS) [68] and selectively damped least square (SDLS) etc. Out of all, damped least-square (DLS) method is the most effective and computationally efficient to deal with singularity. In general manipulator reaches to the singularity configuration, if the manipulator motion lies completely in the feasible direction at the singularity. But usage of damped least-square inverse technique always produces a new trajectory which deviates from the original path near singularity.

The damped least-square inverse method, minimizes the error of the joint velocities solution and magnitude of joint velocities [69], i.e. 
\[
\| [J_s]_{6 \times 6} \partial \theta_s - \partial X_s \|^2 + \lambda \| \partial \theta_s \|^2
\] (75)
The joint velocities are given as the solution of the normal equation:
\[
\partial \theta_s = [J_s]^*_{6 \times 6} \partial X_s
\] (76) 
Where
\[
[J_s]^*_{6 \times 6} = ([J_s]_{6 \times 6}^T [J_s]_{6 \times 6} + \lambda^2 [I]_{6 \times 6})^{-1} [J_s]^T_{6 \times 6}
\] (77)
and [J_s]^*_{6 \times 6} is the damped least-square inverse Jacobian which should be used in the place of normal inverse Jacobian [J_s]_{6 \times 6} to produce non-singular solution for inverse kinematics of the manipulator. \( \lambda \) is the damping factor, which represents the relationship between the joint velocity and velocity error. The lower the value of \( \lambda \), closer the dynamics of the manipulator to the real system due to [J_s]_{6 \times 6} \approx [J_s]^*_{6 \times 6}^{-1} \) which can lead to near singularity situation and the higher the value of \( \lambda \), the more deviation error is found between requested and actual end effector velocity. Therefore, an optimized value of \( \lambda \) should be used so that system can maintain the feasibility of joint velocity and simultaneously deviation error should also be minimized [70].

Final expression for the modified mapping matrix will be:
\[
[J_s]_{6 \times 6} = [J_p]_{6 \times 6} [K]_{6 \times 6} [J_s]_{6 \times 6}^{-1}
\] (78)
\[
[S]_{6 \times 6} = [J_p]_{6 \times 6} [K]_{6 \times 6} [J_s]_{6 \times 6}^{-1}
\] (79)

V. Simulation Results

In the proposed teleoperated mechanism, the master robot is a 6-DOF parallel manipulator and the slave robot is a 6-DOF serial manipulator. To evaluate the accuracy of the complete model and efficacy of its control architecture as one integrated system, several trajectories have been given for the tracking in cartesian space of the serial manipulator. Generating a desired trajectory in the task space of the serial manipulator, where at every moment of time all the six parameters \( X_s = \{ x_s, y_s, z_s, \alpha, \beta, \gamma \} \) of the serial manipulator’s end effector is defined, is a serious challenge. Therefore, an indirect way has been devised to overcome this challenge, according to which first the parallel and serial manipulator are connected without any kind of mapping of the task-spaces of the both the manipulator. Now in this configuration the parallel manipulator is directly driving the serial manipulator. Defining a task-space trajectory for the parallel manipulator is quite easy and it is
shown in (80). This desired trajectory in the task-space of the parallel manipulator has been fed to the controller of the parallel manipulator. Now the tracking controllers will set the actuators of the parallel manipulator in a desired motion which in turn will set desired joint angle motion for the serial manipulator due to mapping between joint-spaces of both the manipulator as shown in [59]. The controllers of the serial manipulator will make sure that actual joint angles of the serial manipulator must track the desired joint angle motion calculated by (59). Finally by knowing the joint angle trajectory (θs), task-space trajectory (Xs) for the serial manipulator can be calculated by (62). This actual task-space trajectory of the serial manipulator is recorded and supplied to the integrated teleoperator system as a desired trajectory in the task-space of the serial manipulator, where all the six parameters of the end-effector are known. Fig. 5 describes the whole technique of generating a desired task-space trajectory for the serial manipulator from feeding known trajectory in the task-space of the parallel manipulator.

The maximum allowed displacement for the parallel manipulator platform in cartesian space is as follows

\[
X_{max} = 0.02 m, Y_{max} = 0.02 m, Z_{max} = 0.017 m, \phi_{max} = 10^\circ, \theta_{max} = 10^\circ, \psi_{max} = 20^\circ.
\]

The desired cartesian trajectories given to platform is described as follows:

\[
S(t) = \begin{cases} 
\text{Surge : } X = \chi_1 = 0.015 \sin(2\pi \times 0.2t) \text{m}, \\
\text{Sway : } Y = \chi_2 = 0.015 \sin(2\pi \times 0.2t) \text{m}, \\
\text{Heave : } Z = \chi_3 = 0.012 \sin(2\pi \times 0.2t) \text{m}, \\
\text{Roll : } \phi = \chi_4 = 5^\circ \sin(2\pi \times 0.2t), \\
\text{Pitch : } \theta = \chi_5 = 5^\circ \sin(2\pi \times 0.2t), \\
\text{Yaw : } \psi = \chi_6 = 15^\circ \sin(2\pi \times 0.2t), 
\end{cases}
\]

Parameters for both the manipulators are described in Tables II and III. Fig. 6 shows the comparison of desired trajectory and actually achieved trajectory, and this is done in the task-space so that comparison can be made in realistically visible space. Though, overall tracking performance is very good still here are some visible deviations between desired and actual speed during whole trajectory as shown in Fig. 7(a), 7(c) and 7(e) and this deviation is basically due to introduction of damped least square Jacobian to avoid singularity. From Fig. 7(b), 7(d) and 7(f) it is clearly evident that tracking of angular speed is extremely good. There are some oscillations in actual task-space trajectory around desired trajectory but these oscillations get settled very soon as tracking goes forward. These oscillation are due to various reasons such has high gain values of the PID controllers and mismatch of task-space mapping at initial condition of operation but all these incompatibility and uncertainties get settled in time less than one second.

### Table II

<table>
<thead>
<tr>
<th>Parameter Values of the Parallel Manipulator</th>
<th>Symbols</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total height of the manipulator</td>
<td>H</td>
<td>0.227 m</td>
</tr>
<tr>
<td>Upper joint radius</td>
<td>r_u</td>
<td>0.110 m</td>
</tr>
<tr>
<td>Lower joint radius</td>
<td>r_b</td>
<td>0.345 m</td>
</tr>
<tr>
<td>Upper joint spacing</td>
<td>L_u</td>
<td>37°</td>
</tr>
<tr>
<td>Lower joint spacing</td>
<td>L_b</td>
<td>16°</td>
</tr>
<tr>
<td>Middle actuator length</td>
<td>l_0</td>
<td>0.250 m</td>
</tr>
<tr>
<td>Upper platform mass</td>
<td>m_p</td>
<td>1.4235 kg</td>
</tr>
<tr>
<td>Actuator’s actuating part mass</td>
<td>m_a</td>
<td>0.37 kg</td>
</tr>
<tr>
<td>Actuator’s rotating part mass</td>
<td>m_b</td>
<td>0.06 kg</td>
</tr>
<tr>
<td>Actuator’s actuating part inertia</td>
<td>i_a</td>
<td>0.00055 kgm²</td>
</tr>
<tr>
<td>Actuator’s rotating part inertia</td>
<td>i_b</td>
<td>0.00009 kgm²</td>
</tr>
<tr>
<td>Reduction ratio from actuator to motor</td>
<td>g</td>
<td>16</td>
</tr>
<tr>
<td>Lead of ball screw</td>
<td>L_0</td>
<td>0.002 m</td>
</tr>
</tbody>
</table>

### Table III

<table>
<thead>
<tr>
<th>Parameter Values of the Coreless-DC Motor</th>
<th>Symbols</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rated output power</td>
<td>P_r</td>
<td>5.3 W</td>
</tr>
<tr>
<td>Rated speed</td>
<td>\omega_n</td>
<td>8200 rpm</td>
</tr>
<tr>
<td>Torque constant</td>
<td>K_t</td>
<td>0.01177 Nm/A</td>
</tr>
<tr>
<td>Back-emf constant</td>
<td>K_e</td>
<td>0.00123 V/rpm</td>
</tr>
<tr>
<td>Armature resistance</td>
<td>R_s</td>
<td>3.6 Ω</td>
</tr>
<tr>
<td>Armature inductance</td>
<td>L_a</td>
<td>0.00022 mH</td>
</tr>
<tr>
<td>Maximum current</td>
<td>I_a,max</td>
<td>3.4 A</td>
</tr>
<tr>
<td>Rotor magnetic flux</td>
<td>\lambda_f</td>
<td>0.1252 Wb</td>
</tr>
<tr>
<td>Moment of inertia</td>
<td>J</td>
<td>1.85 kgm²</td>
</tr>
</tbody>
</table>

### Table IV

<table>
<thead>
<tr>
<th>Hartenberg-Denavit Parameters of the PUMA 560</th>
<th>Symbols</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>First joint</td>
<td>\alpha_1</td>
<td>-160° : 160°</td>
</tr>
<tr>
<td>Second joint</td>
<td>\alpha_2</td>
<td>-225° : 45°</td>
</tr>
<tr>
<td>Third joint</td>
<td>\alpha_3</td>
<td>-45° : 225°</td>
</tr>
<tr>
<td>Fourth joint</td>
<td>\alpha_4</td>
<td>-110° : 170°</td>
</tr>
<tr>
<td>Fifth joint</td>
<td>\alpha_5</td>
<td>-100° : 100°</td>
</tr>
<tr>
<td>Sixth joint</td>
<td>\alpha_6</td>
<td>-266° : 266°</td>
</tr>
</tbody>
</table>
the actuators, Kane’s method based modeling technique is used. By establishing force-torque balance in manipulator’s reference frame, calculation of time varying equivalent inertia of the motor is successfully avoided. Since, over all system of parallel manipulator is highly non-linear in nature with coupling terms and varying parameters, only PID controller alone could not reduce this resultant tracking error, even after best tuning. One lag compensator for each actuator has been designed in combination with PID controller, which resulted in error improved tracking performance for parallel manipulator.

Accuracy of the trajectory tracking in the task-space of the serial manipulator has demonstrated validity of the modeling procedure and efficacy of the overall control architecture proposed in this work. After getting successful results from modeling and simulation, experimental implementation of the overall procedure is underway in a larger project, where usage of this technology in the form of telepresence in the Oil and Gas industry is the main aim.

REFERENCES


Fig. 7. Speed tracking performance along and around the X-axis (a)-(b), Y-axis (c)-(d), Z-axis (e)-(f) respectively where solid blue lines represent desired speed trajectory and dotted red lines represent actual speed trajectory.


