ABSTRACT

Inverse dynamic analysis can be used in designing controllers, but the computational requirements may prevent its use in real time. A combined piecewise linearization and off-line inverse dynamic analysis approach has been exploited in this paper to achieve the required computational efficiency with the convenience of traditional linear controllers, coupled with the accuracy and adaptability of inverse dynamic analysis.

The paper simulates a three-degree-of-freedom RRR planar mechanism to demonstrate the techniques. An inverse dynamic analysis of this mechanism is carried out for different desired motions and for different operating speeds. A piecewise linearization approach is introduced to represent the system as a multi-input multi-output linear system with motion and speed dependent coefficients.

A separate linearization method is developed to determine the error dynamics off-line. A standard linear-quadratic regulator is applied to the linearized model for the design of a feedforward and feedback controller. The robustness against external disturbances of the proposed adaptive linearized feed-forward and feedback controller is examined by simulated examples. Most of the computationally demanding inverse dynamic and linearization calculations are carried out off-line, therefore the technique offers many potential applications involving highly complex systems.

Nomenclature

(·) denotes estimate
α Speed parameter
\( \bar{\nu} \) Time average value of the motor voltage
\( \Delta R^2 \) Margin used to detect break points in the piecewise linearization
\( \Delta x \) Movement of \( x \)
\( \hat{v} \) Estimated motor voltage value
\( \lambda_j \) Lagrange multiplier
0 Matrix or vector of zeros in the appropriate dimension
\( \lambda \) Vector of Lagrange multipliers
\( \theta_m \) Vector of motor angles, their first and second derivatives
\( A \) \( N \times N \) array of \( a_{ij} \) functions
\( B \) Control input coefficient matrix
b Vector of parameters in the estimation
C1 Coefficient matrix of function of \( q \) and \( t \)
C2 Matrix of function of \( q \), \( \dot{q} \) and \( t \)
\( D_1, D_2 \) Vectors of \( q, \dot{q} \) and \( t \)
F Constraint Jacobian matrix
Q Vector of generalized inputs
q Vector of generalized coordinates
Q, R LQR weighting matrices
U Control input vector
u Control vector
V Vector of voltage values
X Matrix of independent variables in the estimation
y Desired motion specifying coordinates
z State vector

*Address all correspondence to this author.
\( \theta_c \) Absolute angle of the platform.

\( \theta_i \) Angular position of the \( i \)th motor

\( a_{i,j} \) Functions of generalized coordinates and time

\( b \) Linearized coefficients

\( f_j \) Constraint equation

\( i, j, k \) Integers

\( J \) Cost function

\( K \) Number of degrees of freedom of the required motion

\( L \) Lagrangian function

\( M \) Number of degrees of freedom

\( N \) Number of generalized coordinates

\( n \) Number of data points in the estimation

\( P \) Covariance matrix for the recursive least square estimator

\( Q_i \) Generalized input

\( q_i \) Generalized coordinate

\( R^2 \) Goodness of fitness measure

\( t \) Time

\( T_s(\cdot) \) Settling time (percentage in parenthesis)

\( u \) Normalized time

\( v_i \) Voltage input to the \( i \)th motor

\( x \) Any coordinate

\( x_0 \) Initial condition of \( x \)

\( x_0, y_c \) Cartesian coordinates of the center of platform.

**Introduction**

Parallel manipulators for high speed applications offer many advantages compared to serial manipulators because closed loop kinematic mechanisms lead to high stiffness and good dynamics properties [1]. In addition, accuracy and reliability are also improved by the elimination of cable transmissions and placing the drive motors on the ground [2, 3]. There is increasing research activity in the field of three or less degrees-of-freedom (DOF) Planar Parallel Manipulators (PPM). Although kinematic problems with different combinations of joints and legs can be successfully tackled [4–6], dynamic analysis is complicated because of the existing of multiple closed loop chains.

Achieving accurate motion control at high speeds with high dynamic performance requires accurate dynamic modeling of multi-physics systems and nonlinear controllers to handle complex dynamic interactions between system components and other connected systems. Various modeling techniques can be used [7–9], but the analytical development of equations of motion is tedious and prone to errors even with the help of symbolic programming tools such as MATLAB [10], and MAPLE [11]. A constraint Lagrangian based approach [12] is used in this paper, where the equations are automatically developed by the software package Dysim [13, 14] from topological data. This is an object oriented approach suitable for multi-physics systems. The model developed can be used as part of other simulation environments such as MATLAB and SIMULINK.

Controllers based on linear system theory do not always function well, and various adaptive control algorithms have been developed to improve performance. Due to the highly nonlinear behavior of these systems, it is difficult to design a controller to perform well for different motions and at different speeds. Inverse dynamic analysis tools do however offer high performance and take into account nonlinear system dynamics [14–17]. They have been used to predict the controller inputs in order to achieve a required motion, and various feed-forward controllers have been suggested in the literature. Inverse dynamic analysis can also be utilized in feedback controllers, but the computational requirements usually prevent real-time implementation.

Previous work [18] introduced a linearization approach for the system and error dynamics of a three DOF PPM by using the inverse dynamics analysis off-line. It was shown that accurate linear models could be obtained for a specified motion that can be used in real time. It resulted in three decoupled second order equations, and the robustness under different motion speeds was tested. Control issues using such linear models were discussed in [19] and simulation results were presented. In this paper, multistage recursive estimation is introduced by utilizing statistical data to obtain more reliable and accurate piecewise linear models for a specified motion. The same method is also applied for the error model. The linearized error model is implemented with a Linear Quadratic Gaussian (LQR) control algorithm to ensure stability and improve performance.

**System Dynamics**

The Lagrangian equations of motion are [12]

\[
\frac{d}{dt} \left( \frac{\partial L}{\partial \dot{q}_i} \right) - \frac{\partial L}{\partial q_i} + \sum_{j=1}^{N-M} \lambda_j \frac{\partial f_j}{\partial q_i} = Q_i, \quad i = 1 \cdots N
\]  

(1)

The symbols are defined in the nomenclature. The Lagrangian function represents the total kinetic energy minus the total potential energy of the system, and can be written in the following general form in terms of the generalized coordinates:

\[
L = \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} a_{i,j} \ddot{q}_i \dot{q}_j + \sum_{i=1}^{N} a_{i,0} \dot{q}_i + a_{0,0}
\]  

(2)

The functions \( a_{i,j} \)s are functions of generalized coordinates and time. Due to the use of superfluous coordinates, \( (N - M) \) constraint equations are needed.

\[
f_j = 0, \quad j = 1 \cdots (N - M)
\]  

(3)
Inserting (2) into (1) and double differentiating (3) gives the following \( 2N - M \) algebraic differential equations:

\[
\begin{bmatrix} A & F^T & B \\ F & 0 & 0 \\ C_1 & 0 & 0 \end{bmatrix} \begin{bmatrix} \ddot{q} \\ \lambda \end{bmatrix} = \begin{bmatrix} D_1 \\ D_2 \end{bmatrix} + \begin{bmatrix} Q \\ 0 \end{bmatrix}
\]

This can be solved for \( 2N - M \) unknowns, namely the second derivatives of generalized coordinates and Lagrange multipliers, and then the second derivatives can be integrated twice to obtain the motion of the system. The time history of Lagrange multipliers is automatically calculated and can be used to calculate the forces of constraints.

In inverse dynamics analysis, control inputs are calculated in order to achieve a desired motion, which is specified in terms of the second derivatives of the generalized coordinates. Assuming that the required motion has \( K \) (\( \leq M \)) degrees of freedom, it can be represented as follows:

\[
\ddot{y} = C_1 \dddot{q} - C_2
\]

The control input vector \( U \) of dimension \( K \) can be added to the generalized input vector with a coefficient matrix of \( B \) specifying the location of the control action. There are various ways of formulating an inverse dynamics model as discussed in [14], but the general formulation can be written as follows by moving the unknown control input vector to the left hand side in (4) and using the desired motion as additional constraint equations:

\[
\begin{bmatrix} A & F^T & B \\ F & 0 & 0 \\ C_1 & 0 & 0 \end{bmatrix} \begin{bmatrix} \dddot{q} \\ \lambda \\ U \end{bmatrix} = \begin{bmatrix} D_1 + Q \\ D_2 \\ \ddot{y} + C_2 \end{bmatrix}
\]

The above algebraic differential equations can be solved for the second derivatives of the generalized coordinates, which can then integrated twice to obtain the motion of the system. The required control input vector and the Lagrange multipliers are automatically calculated during this process.

The in-house simulation package Dysim [14] is used in this paper. The program automatically develops the equation of motions for the forward and inverse dynamic analysis from physical data. It is suitable for multi-physics system, and its object oriented properties allow the models to be used by other simulation environments. Matlab/Simulink [10] environment is used to generate the results in this paper.

### Simulated System

The PPM shown in Fig. 1 is consist of two equilateral triangles, and a moving platform \( B_1B_2B_3 \) connected to the ground by three independent kinematic chains \( A_iB_iC_i, \ i = 1 \cdots 3 \). Each chain has three independent revolute joints, and the actuators are placed on the ground and connected to the manipulator at \( A_1,A_2,\text{and}A_3 \). The origin of the earth fixed coordinate system is defined as the center of the triangle \( A_1A_2A_3 \). The positions of \( A_1, A_2, \text{and}A_3 \) are \( (-0.15\sqrt{3}, -0.15) \) m, \( (0.15\sqrt{3}, -0.15) \) m, and \( (0, 0.3) \) m respectively. Other relevant data for the manipulator are shown in Table 1. The three angles \( \theta_1, \theta_2, \text{and} \theta_3 \) are driven by three similar dc-motors. Data for each motor are given in Table 2.

The mechanical manipulator consists of 7 objects, namely six connecting rods and the central platform. The properties of each object and the connection information are fed to Dysim. The program selects 21 generalized coordinates (three for each object) for the mechanism and 3 generalized coordinates (charge for each motor) for the motors. The total number of degrees of freedom is 6; 3 mechanical and 3 electrical. The Lagrangian function and the dynamic equations of motion including the 18 constraint equations are automatically developed. The program also calculates the initial conditions of the superfluous coordin-

![Three degrees of freedom PPM](image)

**Table 1. Manipulator Data**

<table>
<thead>
<tr>
<th>Object</th>
<th>Length (m)</th>
<th>Diameter (m)</th>
<th>Mass (kg)</th>
<th>Inertia (gm²)</th>
</tr>
</thead>
<tbody>
<tr>
<td>( A_iC_i )</td>
<td>0.17</td>
<td>0.02</td>
<td>0.144</td>
<td>0.3617</td>
</tr>
<tr>
<td>( C_iB_i )</td>
<td>0.15</td>
<td>0.02</td>
<td>0.127</td>
<td>0.2513</td>
</tr>
<tr>
<td>( B_1B_2B_3 )</td>
<td>0.13\sqrt{3}</td>
<td>-</td>
<td>0.331</td>
<td>0.1663</td>
</tr>
<tr>
<td>Center Load</td>
<td>-</td>
<td>0.03</td>
<td>1.000</td>
<td>0.09</td>
</tr>
</tbody>
</table>

Copyright © 2006 by ASME
The velocity and acceleration can be written as

\[ x = x_0 + \Delta x \left( 1 - e^{-(\alpha t)^3} \right) \]  \hspace{1cm} (7)

This is a smooth and twice differentiable continuous function for all values of \( t \geq 0 \). Its first and second derivatives are zero at the start and at the end of the motion. In addition, the motion is controlled by a single parameter which gives complete control of the motion speed. The speed parameter \( \alpha \) determines the motion speed, and related to the 1% and 5% settling times as follows:

\[ T_s(0.01) = \frac{1.66}{\alpha}, \quad T_s(0.05) = \frac{1.44}{\alpha} \]  \hspace{1cm} (8)

The velocity and acceleration can be written as

\[ \dot{x} = \alpha \Delta x (3u^2) e^{-u^3} \]  \hspace{1cm} (9)

\[ \ddot{x} = \alpha^2 \Delta x (6u - 9u^4) e^{-u^3} \]  \hspace{1cm} (10)

where \( u = \alpha t \) is the normalized time.

A straight line point-to-point motion of the platform is selected as the desired motion. Only the acceleration-time history is required for the inverse dynamic analysis. The Cartesian coordinates of the center of gravity of the platform is initially set to (-0.025, -0.025) m with an angle of \( \theta_c = 0 \) deg. The final position is specified as (0.025, 0.025) m with an angle of 10 deg. The motion duration is selected as 0.25 s. Equation (7) is applied to all three coordinates of the platform, \( x_c, y_c \) and \( \theta_c \) to specify this desired motion to the inverse dynamic modeling.

**Piecewise Linearization**

If the motor input voltages are selected as control inputs, inverse dynamic analysis will generate the time history of control voltages and the motor output coordinates (angles and their first derivatives). Considering a decoupled linear model for one of the motors, say the \( i \) th motor, the control voltage can be written in the following form as described in [18, 19]:

\[ v_i(t) = b_{0,i} + b_1, \dot{\theta}_i + b_2, \ddot{\theta}_i + b_3, \theta_i \]  \hspace{1cm} (11)

If PPM is moving on a horizontal plane, the required motor voltages at the end of the motion should approach to zero giving the following constraint on the linearized parameters:

\[ b_{0,i} = -b_3, \theta_i(T_s) \]  \hspace{1cm} (12)

if the data from the simulation are stored in a discrete form, they can be arranged in a matrix form as follows:

\[ \mathbf{V} = \begin{bmatrix} v_1(1) \\ v_1(2) \\ \vdots \\ v_1(n) \end{bmatrix}, \quad \mathbf{X} = \begin{bmatrix} \theta_i(1) & \dot{\theta}_i(1) & \ddot{\theta}_i(1) \\ \theta_i(2) & \dot{\theta}_i(2) & \ddot{\theta}_i(2) \\ \vdots & \vdots & \vdots \\ \theta_i(n) & \dot{\theta}_i(n) & \ddot{\theta}_i(n) \end{bmatrix} \]  \hspace{1cm} (13)

This gives the following Least Square Estimator (LSE) of the parameter vector \( \mathbf{b} \) [22]:

\[ \mathbf{\hat{b}} = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{V} \]  \hspace{1cm} (14)

The estimation process is illustrated in Fig. 2, where the vector \( \theta_m \) contains motor angles, and their first and second derivatives. Also the statistical data from the LSE, such as the goodness of fit measure \( R^2 \) and the standard errors of each estimated coefficients can be utilized to assess the accuracy of the linearized model and significance of the estimated parameters. The goodness of fit measure \( R^2 \), corrected by the degrees of freedom of
the estimation, is defined as follows:

\[
R^2 = 1 - \frac{n - 1}{n - 3} \left( 1 - \sum_{k=1}^{n} \frac{(v_i(k) - \bar{v}_i)^2}{\sum_{k=1}^{n} (v_i(k) - \bar{v}_i)^2} \right)
\]  \hspace{1cm} (15)

where \(n\) is the number of data points. If a high \(R^2\) value (close to 1) is obtained then the linearized model can be used in real time implementations instead of the nonlinear inverse dynamic analysis [18, 19] for the specified motion.

This paper extends the method to cases where the estimation produces poor \(R^2\) by breaking the motion duration into two or more linearized sections. For this a recursive version of the LSE is needed so that the estimates are updated at each time interval. The recursive estimation of coefficient matrix at the \(k\)th time interval, \(\hat{b}(k)\) can be formulated as [23]:

\[
\hat{b}(k) = \hat{b}(k-1) + P(k-1)X(k)[1 + X^T(k)P(k-1)X(k)]^{-1} \times [V(k) - X^T(k)\hat{b}(k-1)]
\]  \hspace{1cm} (16)

where the covariance matrix \(P\) is updated as follows:

\[
P(k) = [P(k-1)^{-1} + X(k)X^T(k)]^{-1}
\]  \hspace{1cm} (17)

The initial values of the coefficient estimates are taken as zero, and the initial value of the \(P\) matrix is set to a very large diagonal matrix.

The piecewise linearization break points are calculated by an algorithm based on \(R^2\) measurements at each time step. The \(R^2\) value should initially be increasing with each sampled data until a peak value is achieved. If the subsequent measurements of \(R^2\) drops by a predefined margin \(\Delta R^2\) below the current maximum value of \(R^2\), then the algorithm defines this point as the break point and stops the estimation. The recursive estimator is then initialized and started again until the next break point or end of the motion is reached. This enables the linearized controllers to be used in a wide range of applications involving highly nonlinear systems and high speed motions. This process is demonstrated by using a numerical example.

**Numerical example:** A sampling time interval of 0.0005 s (sampling frequency of 2 kHz) is selected, which gives 501 sampling times including the initial values for the duration of the motion. Straight LSE results by using the complete data set for the simulated motion are shown in Table 3. Although a good fit is obtain for all three motors, the \(R^2\) value for Motor 3 is relatively low. The calculated and linearized motor control voltages are shown in Fig. 3. The lower \(R^2\) for the Motor 3 can be observed graphically from the figure.

Figure 4 shows \(R^2\) values when RLSE is applied to the same data. As it can clearly be seen from the graphs that the \(R^2\) value for the Motor 3 is decreasing. A slight decrease for Motor 2 can also be seen towards the end of the motion. When the piecewise linearization option is selected with \(\Delta R^2 = 0.01\), no break point was detected for Motor 1, but one break point for Motor 2 at 339th sampling interval, and two break points for Motor 3 at 118th, 317th sampling intervals were detected. The calculated
Figure 4. \( R^2 \) values calculated at each sampling interval by the RLSE

and piecewise linearized control voltages for all three motors are shown in Fig. 5. The estimated parameters and the goodness of fit values for Motor 2 and Motor 3 are given in table 4. A significant improvement on the Motor 3 results is apparent from this figure with a change of \( R^2 \) from 0.9251 to 1.0.

No attempt has been made to ensure continuity of the linearized voltages between regions. This can be done by introducing another constraint to the linearization process, but this would effectively reduce the number of coefficients in the estimation thus reducing the goodness of fit. The problem due to the jump discontinuity could also be handled by the error dynamic loop as discussed later.

**Error Dynamics**

After successfully linearizing the inverse dynamic model to provide the feed-forward control input to the system, it is necessary to add feedback control to take into account the modeling errors, unknown or unmeasured external disturbances, and measurement noise. It has been shown previously that the error dynamics for a nonlinear system can be significantly different than the system dynamics [18]. A linearization process is applied to the simulated errors, and proposed to be used in the feedback for real-time implementation. A multi-frequency disturbance is introduced at each motor sequentially, and the required changes in the motor input voltages are calculated as shown in Fig. 6 and used in a RLSE algorithm to calculate the coefficients of the linearized error model. A Schroeder Phased Harmonic Sequence (SPHS) [24] is used to generate a low-peak-factor multifrequency signal with controllable power spectrum to represent disturbances. A coupled linear model is considered for the error dynamics. The disturbance levels on motor accelerations were selected as 10% of the desired acceleration. When a disturbance is introduced to the motion of each motor one at a time, the inverse dynamics would produce variations of control voltages in all three motors. Therefore, a total of 9 coefficients would be estimated from each test. This would give a total of 27 coefficients for the error model. Figure 7 shows the calculated voltage variations in order to compensate for the motion errors introduced to three motors, and compares these with the results obtained from a linearized model. The goodness of values \( R^2 \) for 3 motors are 0.8667, 0.9626 and 0.8214. These are not as good as the \( R^2 \) values obtained for linearizing the system in the previous section, however the error models are not used directly in the feedback,
but to design a LQR regulator. Although this is a path tracking problem, the error dynamics can be treated as a regulator problem as constant zero output is desired. Most of the nonlinearities are compensated by the feed-forward path.

**Linear Quadratic Regulator**

A Linear Quadratic regulator is selected to achieve a stable and optimum performance of the linearized error model. The LQG regulator consists of an optimal state-feedback gain, which minimizes a quadratic cost function as follows [25]:

$$J(z,u) = \int_0^\infty (z^TQz + u^TRu) \, dt$$

(18)

where the state vector $z$ corresponds to motion errors in angular displacements, velocities and accelerations of three motors, and the control vector $u$ corresponds to corrections to the control voltages. The diagonal weight matrices $Q$ and $R$ are selected by using Bryson’s rule according to the maximum acceptable values of the elements of the $z$ and $u$ vectors. The linearized error model is used to form the LQR.

The overall control system is shown in Fig. 8. The controller is tested by adding SPHS disturbance forces on the platform along the $x$ and $y$ directions. The results of the simulation are shown in Fig. 9. The tracking errors in terms of $x_c$, $y_c$ and $\theta_c$ are shown as a function of time during the motion. The results are also compared with the case where there is no LQR feedback loop. The effectiveness of the suggested LQR control based on the linearized error dynamics can be clearly seen from the figure. Various simulation runs (not presented here) with different disturbance levels were also made, and showed the ability of the LQR controller to produce stable and accurate results when operating under varying conditions.

**Conclusions**

In this paper, a numerical dynamic model for a three DOF RRR PPM is built by using Lagrangian equations and the software *Dysim*, which utilizes a numerical coding system. Both forward and inverse dynamics models are used in a Simulink environment to design linear feed-forward and feedback controllers for real-time implementations. The numerical linearization is used to form the LQR.

---

**Table 4. Piecewise RLSE results for Motors 2 and 3**

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Region 1</th>
<th>Region 2</th>
<th>Region 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Motor 2:</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Data Range</td>
<td>1-339</td>
<td>340-501</td>
<td></td>
</tr>
<tr>
<td>$b_{2,1}$</td>
<td>0.0252</td>
<td>0.0335</td>
<td></td>
</tr>
<tr>
<td>$b_{2,2}$</td>
<td>0.6518</td>
<td>-0.0797</td>
<td></td>
</tr>
<tr>
<td>$b_{2,3}$</td>
<td>0.2569</td>
<td>-38.38</td>
<td></td>
</tr>
<tr>
<td>$R^2_2$</td>
<td>0.9898</td>
<td>0.9999</td>
<td></td>
</tr>
<tr>
<td>Motor 3:</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Data Range</td>
<td>1-118</td>
<td>119-317</td>
<td>318-501</td>
</tr>
<tr>
<td>$b_{3,1}$</td>
<td>-0.0682</td>
<td>-0.0295</td>
<td>0.0063</td>
</tr>
<tr>
<td>$b_{3,2}$</td>
<td>-0.6596</td>
<td>-1.0297</td>
<td>1.0327</td>
</tr>
<tr>
<td>$b_{3,3}$</td>
<td>-12.982</td>
<td>-19.901</td>
<td>41.328</td>
</tr>
<tr>
<td>$R^2_3$</td>
<td>0.9903</td>
<td>0.9896</td>
<td>0.9969</td>
</tr>
</tbody>
</table>
Figure 9. Tracking errors with (solid line) and without (dashed line) LQR under SPH disturbance forces on the table.

An off-line strategy is also developed to obtain a coupled linearized error model from the nonlinear simulations. The error dynamics are significantly different to the system dynamics. The path following requirements are converted to a regulator problem in the error model, and a LQR controller is designed to ensure stability of the feedback loop and to optimize the trajectory tracking performance. The designed regulator-based controller is tested by applying external unknown disturbance forces to the platform. The simulation results demonstrate good control performance. The suggested linear controller not only deals with the system nonlinearities, it is also robust against unknown disturbances for high speed path following problems. This approach performs all the computationally demanding calculations off-line, and synthesizes an optimized linear controller to be implemented online.

REFERENCES


