FRACTIONAL ORDER CONTROL OF THE 3-CPU PARALLEL KINEMATICS MACHINE

Luca Bruzzone, Pietro Fanghella
University of Genova - DIME
Via Opera Pia 15a, 16145 Genova (GE), Italy
bruzzone@dimec.unige.it, pietro.fanghella@unige.it

ABSTRACT
The paper deals with the application of a fractional-order position control scheme to a parallel kinematics machine, the 3-CPU platform. In particular, the PDD$^{1/2}$ control scheme, which exploits the half-derivative term in combination with the proportional and derivative ones, is applied to control the positions of the three linear actuators. A nondimensional approach is used for sake of generality. The results of the multibody simulations show that this control scheme can reduce the tracking error with respect to the classical PD scheme using the same settling energy.

KEY WORDS
Control, Mathematical Modelling and Simulation, Fractional Calculus, Parallel Kinematics Machine.

1. Introduction

Even though the concept of Fractional Calculus, which considers non-integer derivatives and integrals, dates back to the end of the seventeenth century, its first practical applications are relatively recent; important application fields are electronics, signal processing, bioengineering, control system design [1-2].

Undoubtedly, most real phenomena can be correctly modelled by means of integer order (IO) differential equations; nevertheless, there are examples of physical systems that can be better modelled using fractional order (FO) derivatives [3-6]. In physics, in the field of variational principles, it is possible to replace IO derivatives with FO ones, obtaining fractional formulations of the Euler-Lagrange equations, of the Hamilton equations and of the Dirac equations [7-9]. The dynamic and vibrational behaviour of some viscoelastic polymers and elastomers can be better modelled by fractional order differential equations [10, 11].

Even if fractional order systems are relatively rare both in nature and in technology, Fractional Calculus can be applied in the area of control system design to control integer-order plants [12-15]; this is probably the use of Fractional Calculus with the widest range of potential applications. The usual approach to synthesize fractional-order control consists in the PI'$^D$' scheme, which generalizes the PID scheme by using derivatives and integrals to an arbitrary order [16]; the two orders $\lambda$ and $\mu$ provide additional degrees of freedom to tune the system behaviour.

In [17-20] the PDD$^{1/2}$ scheme is proposed and compared to the PD scheme; this alternative approach to fractional-order control consists in using both the half derivative term and the first-order derivative term in combination (only the behaviour in the transient state is analysed, therefore the integral term, which eliminates the steady state error, is not considered). In particular, in [20] the assessment of the influence of the half-derivative term is performed introducing the dimensionless settling energy, physically representing the effort of the control system to drive the controlled system to steady state; the considered controlled system is a second-order purely inertial linear system, and different combinations of the derivative and half-derivative terms are compared keeping constant the dimensionless settling energy. The results show that a proper combination of derivative and half-derivative terms allows to reduce the settling time and the rise time, with a limited increase of the overshoot.

In the present paper the same combination of integer-order and fractional-order derivative terms are applied to position control of a three-degree-of-freedom 3-CPU parallel kinematics machine, to verify the effectiveness of the proposed control scheme in case of nonlinear and coupled MIMO controlled system.

2. The PDD$^{1/2}$ control

Fig. 1 shows the block-scheme of a second-order translational system with a PDD$^{1/2}$ control; $m$ is the translating mass; $x$ is the mass displacement and $F$ is the control output force; the transfer function of the PDD$^{1/2}$ control is expressed by the following equation:

$$G(s) = K_p + K_d s + K_{hd} s^{1/2}$$

where $K_p$ is the proportional gain, $K_d$ is the derivative gain and $K_{hd}$ is the half-derivative gain.

For the numerical simulation of the system of Fig. 1, the discrete-time computation of the half-derivative for the PDD$^{1/2}$ control is based on the following equation [21]:

$$G_{d}(s) = K_p + K_d s + K_{hd} s^{1/2}$$
\[ D^\alpha x(t) = \frac{1}{h^\alpha} \sum_{k=0}^{[\frac{t}{kh}]} w_k^\alpha x(t - kh) \] (2)

where \( h \) is the sampling time, \( \alpha \) is the derivative order (1/2 for the half-derivative) and the recursive terms \( w_k \) are evaluated recursively from:

\[ w_0^\alpha = 1 \]
\[ w_k^\alpha = \left( 1 - \frac{\alpha + 1}{k} \right) w_{k-1}^\alpha, \quad k = 1, 2, ... \] (3)

The response of the dimensionless closed-loop system in case of unit step input is discussed in [20], with a systematic comparison of the effects of the derivative and of the half-derivative terms on the settling time (2% band), the rise time from 10% to 90% of the final value, the overshoot (%), and the settling energy, defined according to the following equation:

\[ E_{x,ad} = \int_0^\infty F_{ad}^2 dt_{ad} \] (9)

While the increase of derivative term reduces remarkably the rise time but not significantly the settling time, the increase of half-derivative term reduces both the rise time and the settling time; on the other hand, the derivative term has a positive effect on the overshoot, while the half-derivative doesn’t influence significantly this parameter. It is interesting to compare different combinations of derivative and half-derivative terms, characterized by the same settling energy; in particular, Fig. 3 shows the time histories of \( x_{ad} \) in seven cases, corresponding to the seven parameter sets of Tab. 1.

### Table 1

<table>
<thead>
<tr>
<th>Parameter set</th>
<th>( \zeta )</th>
<th>( \psi )</th>
<th>Overshoot (%)</th>
<th>Dimensionless settling time</th>
<th>Dimensionless rise time</th>
</tr>
</thead>
<tbody>
<tr>
<td>a (PD)</td>
<td>1.2</td>
<td>0</td>
<td>10.42</td>
<td>5.61</td>
<td>0.67</td>
</tr>
<tr>
<td>b (PDD(1/2))</td>
<td>1</td>
<td>1.851</td>
<td>14.34</td>
<td>3.26</td>
<td>0.48</td>
</tr>
<tr>
<td>c (PDD(1/2))</td>
<td>0.8</td>
<td>2.193</td>
<td>17.39</td>
<td>2.87</td>
<td>0.48</td>
</tr>
<tr>
<td>d (PDD(1/2))</td>
<td>0.6</td>
<td>2.223</td>
<td>21.02</td>
<td>2.72</td>
<td>0.51</td>
</tr>
<tr>
<td>e (PDD(1/2))</td>
<td>0.4</td>
<td>2.073</td>
<td>26.04</td>
<td>2.71</td>
<td>0.55</td>
</tr>
<tr>
<td>f (PDD(1/2))</td>
<td>0.2</td>
<td>1.799</td>
<td>33.99</td>
<td>4.22</td>
<td>0.61</td>
</tr>
<tr>
<td>g (PD(1/2))</td>
<td>0</td>
<td>1.375</td>
<td>49.64</td>
<td>8.46</td>
<td>0.69</td>
</tr>
</tbody>
</table>

In Table 1 the settling time, the rise time and the overshoot are compared. While the overshoot is lowered by increasing values of \( \zeta \), the settling time has a minimum: in the first three cases settling is delayed by the high damping, while in the last three cases settling is slowed down by the oscillations; also the rise time has a minimum, but for higher values of \( \zeta \). The parameter sets c to d represent a good compromise of the different performance indices.

In the following of the work the seven parameter sets a to g will be applied to position control of the 3-CPU parallel mechanism to verify if the benefits of this combination of derivative and half-derivative terms are confirmed also in case of a strongly nonlinear and coupled
MIMO system, and in presence of a reference different
form the step.

3. The 3-CPU parallel kinematics machine

The 3-CPU translating parallel manipulator is made up of
three legs with serial CPU kinematics (Cylindrical,
Prismatic and Universal joints) that connect the fixed base
to the moving platform [22-23].

Figure 4 shows the sketch of a single leg: O(x,y,z) is
the global Cartesian frame; P(u,v,w) is the frame located
on the moving platform; the reference point P is placed at
the intersection of the inner revolute axes of the universal
joints (t).  

Fig. 5 shows a simplified kinematic scheme of the
complete robot. The points B_i represent the mobile
platform universal joint centres; the points A_i represent
the positions of cylindrical joints; a_i=(A_i-O) is a vector
with the direction of axis of the cylindrical joint of the i-th
leg.

In general the moving platform is characterized by a
complex spatial motion; nevertheless, a motion of pure
translation is achieved under the following geometrical
constraints [22]:

- the three legs are identical and symmetrically
disposed with respect to the centre of the fixed base;
- the axes of the 3 cylindrical joints (a_i) intersect at a
common point;
- in each leg, the cylindrical joint axis (a_i) is parallel
to the axis of the inner revolute pair of the universal
joint (t_i), and they are both normal to the prismatic
pair axis (d_i);
- in each leg, the outer revolute pair of the universal
joint is normal to the inner one.

Under these assumptions, the two frames O(x,y,z) and
P(u,v,w) maintain the same orientation during the
machine’s motion.

To fulfill these conditions, and to obtain an axially
symmetric machine, we adopt the following values of the
angles α_i and φ_i (Fig. 4): α_1 = α_2 = α_3 = arccos((2/3)0.5) =
35.26°; φ_1 = 0°; φ_2 = 120°; φ_3 = 240°. Let us note that the
prismatic axis d_i can lie in the same plane containing t,
and a_i (s_i = 0) or in a parallel plane (s_i = 0); however, s_i
does not influence the kinematics; in the following we
will assume s_i =0.
If the platform only translates, the outer revolute pairs of the universal joints are actually idle, i.e. no relative rotation at all occurs in these pairs [22]; however, these joints are not blocked to avoid over-constraint conditions.

The actuation of the platform is obtained by driving the linear motion of the cylindrical pairs; therefore the vector of the robot internal coordinates is composed of the magnitudes of the three vectors \( \mathbf{a_i} \), (Fig. 4), while the robot external coordinates are the three components of the vector \( \mathbf{p} = (P-O) \).

Under these geometric conditions, the direct and inverse position kinematics are expressed by the following equations [22]:

\[
\begin{bmatrix}
    p_x \\
    p_y \\
    p_z \\
\end{bmatrix} =
\begin{bmatrix}
    2/\sqrt{6} & -1 & -1 \\
    0 & 1/\sqrt{2} & -1/\sqrt{2} \\
    1/\sqrt{3} & 1/\sqrt{3} & 1/\sqrt{3} \\
\end{bmatrix}
\begin{bmatrix}
    a_1 \\
    a_2 \\
    a_3 \\
\end{bmatrix} -
\begin{bmatrix}
    0 \\
    0 \\
    \sqrt{3}t \\
\end{bmatrix}
\] 

\[\text{(10)}\]

\[
\begin{bmatrix}
    a_1 \\
    a_2 \\
    a_3 \\
\end{bmatrix} =
\begin{bmatrix}
    \sqrt{2/3} & 0 & \sqrt{1/3} \\
    -\sqrt{1/6} & \sqrt{1/2} & \sqrt{1/3} \\
    -\sqrt{1/6} & -\sqrt{1/2} & \sqrt{1/3} \\
\end{bmatrix}
\begin{bmatrix}
    p_x \\
    p_y \\
    p_z \\
\end{bmatrix} +
\begin{bmatrix}
    t \\
    t \\
    t \\
\end{bmatrix}
\] 

\[\text{(11)}\]

where \( t \) is the distance between the point \( P \) and the platform universal joints centres.

Even if the Jacobian matrix is evidently constant and the platform is purely translating, the robot dynamics is nonlinear, due to the presence of the centrifugal and Coriolis terms related to the leg links; the inverse and direct dynamic models of the 3-CPU are discussed in [24].

4. Model of the closed-loop system and control assumptions

A multibody model of the 3-CPU parallel robot controlled in closed-loop has been developed in SimMechanics, within the Matlab/Simulink environment. The control layout is shown in Fig. 6: the trajectory generator originates a reference trajectory in the external coordinates, which is transformed in the internal coordinates by means of the inverse kinematics equations (11). The three errors, differences between the reference positions \( (a_{1,ref}, a_{2,ref}, a_{3,ref}) \) and the actual positions \( (a_1, a_2 \) and \( a_3) \) of the actuators, are the inputs of three SISO controllers, which can be characterized by the seven parameter sets of Tab. 1. The controller outputs, saturated at a maximum value \( F_{max} \), are the input forces of the multibody model of the machine.

The robot geometry considered in the simulations is defined by the coordinates of the main robot points in the initial configuration, which is in the centre of the workspace; these coordinates are listed in Tab. 2, with \( t = 0.1\sqrt{3}/2 \text{ m} = 0.1224 \text{ m}, l = 0.5 \text{ m} \) and:

\[
h_p = \frac{3}{\sqrt{2}} l - \sqrt{3}t
\]

\[\text{(12)}\]

\[
h_b = l \left( \sqrt{2} + 1/\sqrt{2} \right) - \sqrt{4/3}t
\]

\[\text{(13)}\]

Fig. 6. Layout of the 3-CPU PKM position control

The robot inertial features considered in the simulations are collected in Tab. 3. Since the end-platform translates without rotations, the elements of its inertia tensor don’t influence the dynamic behaviour. All the leg links rotate only about the directions of the vectors \( (A_i-O) \); therefore only the corresponding elements of the inertia tensor are reported in Tab. 3; the centres of mass of the first leg links are considered located in \( A_1 \); the centres of mass of the second leg links, placed between the prismatic and the universal joints, are located in the middle of \( A_i \) and \( B_i \) in the initial configuration. The saturation force for each actuator is \( +/-500 \text{ N} \).

<table>
<thead>
<tr>
<th>point</th>
<th>( x )</th>
<th>( y )</th>
<th>( z )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( P )</td>
<td>0</td>
<td>0</td>
<td>( h_0 )</td>
</tr>
<tr>
<td>( A_1 )</td>
<td>-1/2</td>
<td>( \sqrt{2} l/2 )</td>
<td>( l/\sqrt{2} )</td>
</tr>
<tr>
<td>( A_2 )</td>
<td>( -1/2 )</td>
<td>( -\sqrt{2} l/2 )</td>
<td>( l/\sqrt{2} )</td>
</tr>
<tr>
<td>( A_3 )</td>
<td>( \sqrt{2}/3t )</td>
<td>0</td>
<td>( h_0 )</td>
</tr>
<tr>
<td>( B_1 )</td>
<td>( -t/\sqrt{3} )</td>
<td>( t/\sqrt{2} )</td>
<td>( h_0 )</td>
</tr>
<tr>
<td>( B_2 )</td>
<td>( -t/\sqrt{3} )</td>
<td>( t/\sqrt{2} )</td>
<td>( h_0 )</td>
</tr>
<tr>
<td>( B_3 )</td>
<td>( -t/\sqrt{3} )</td>
<td>( t/\sqrt{2} )</td>
<td>( h_0 )</td>
</tr>
</tbody>
</table>

Table 2

Coordinates of the main points of the 3-CPU mechanical model in the initial configuration

| end platform | mass | \( 5 \text{ kg} \) |
| first leg link (between cylindrical and prismatic joint) | mass | \( 4 \text{ kg} \) |
| moment of inertia around \( (A_i-O) \) | \( 1.1e-3 \text{ kgm}^2 \) |
| second leg link (between prismatic and universal joint) | mass | \( 4 \text{ kg} \) |
| moment of inertia around the barycentric axis parallel to \( (A_i-O) \) | \( 1.6e-1 \text{ kgm}^2 \) |

In order to compare the different combinations of derivative and half-derivative terms discussed in
section 2, simulations have been performed selecting the gains in the following way:

- the value of the proportional gain \( K_p \) is selected in order to obtain a proper “equivalent natural frequency”: with an heuristic approach, we consider the overall robot moving mass \( m_r \) and we multiply the gain \( K_p \) for \( \sqrt{3} \) because three linear actuators are orthogonal, and the magnitude of the vector sum of three orthogonal unit vectors is \( \sqrt{3} \):

\[
\omega_n = \frac{\sqrt{3} K_p}{m_r}
\]  

(14)

- \( K_p = 2 \times 10^4 \) N/m is obtained considering \( m_r = 29 \) kg and imposing an adequate equivalent natural frequency (35 rad/s)

- starting from \( K_p \), \( m_r \) and \( \omega_n \) the derivative and half-derivative gains are evaluated using eqs. (4) and (5)

In the simulations the half-derivative of the error is computed using eq. (2) with sampling time \( h = 10^{-3} \) s, considering all the time history (absolute memory).

5. Simulation results

In the simulations, nine rectilinear displacements with length \( l_{disp} = \sqrt{3}/10 \) m =0.173 m starting from the central position defined in Tab. 2 are considered; the nine displacement directions, collected in Tab. 4, are in the half-space defined by the \( zx \) plane of the fixed reference frame with positive \( y \) (only half workspace is considered for symmetry).

Table 4
Simulation displacement directions
(unit vector components)

<table>
<thead>
<tr>
<th>Displacement number</th>
<th>x</th>
<th>y</th>
<th>z</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>-1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>5</td>
<td>0</td>
<td>0</td>
<td>-1</td>
</tr>
<tr>
<td>6</td>
<td>( \sqrt{3}/3 )</td>
<td>( \sqrt{3}/3 )</td>
<td>( \sqrt{3}/3 )</td>
</tr>
<tr>
<td>7</td>
<td>-( \sqrt{3}/3 )</td>
<td>( \sqrt{3}/3 )</td>
<td>( \sqrt{3}/3 )</td>
</tr>
<tr>
<td>8</td>
<td>( \sqrt{3}/3 )</td>
<td>( \sqrt{3}/3 )</td>
<td>-( \sqrt{3}/3 )</td>
</tr>
<tr>
<td>9</td>
<td>-( \sqrt{3}/3 )</td>
<td>( \sqrt{3}/3 )</td>
<td>-( \sqrt{3}/3 )</td>
</tr>
</tbody>
</table>

The reference motion planned by the trajectory generator along these rectilinear segments is characterized by a trapezoidal speed law with maximum speed \( v_{disp} = l_{disp} \) / 0.5 s = 0.346 m/s and acceleration/deceleration \( a_{disp} = v_{disp} \) / 0.1 s = 3.46 m/s\(^2\), with these values the acceleration and deceleration phases are characterized by a duration of 0.1 s, while the duration of the constant speed phase is 0.4 s.

Figures 7 to 10 show the percentage tracking error for the displacements 1, 3, 4 and 6, expressed by the following expression:

\[
\% = 100 \sqrt{\left( p_x - p_{x,ref} \right)^2 + \left( p_y - p_{y,ref} \right)^2 + \left( p_z - p_{z,ref} \right)^2}
\]

(15)
It is possible to note that the gain sets have similar influence on the time histories of the tracking error, independently of the displacement direction. This is generalized by the three-dimensional graph of Fig. 11, which shows the maximum tracking error for all the nine trajectories of Tab. 4 and the seven gain sets of Tab. 1: the combination of both the proportional and half-derivative terms (PDD$^{1/2}$) reduces the maximum tracking error with respect to the pure PD or PD$^{1/2}$.

The three-dimensional graph of Fig. 12 compares the settling energy for the same grid of Fig. 11; as a matter of fact, the seven gain sets a-g are characterized by the same adimensional settling energy defined by eq. (9) for the step response of a second-order purely inertial linear system, as discussed in section 2; therefore it is necessary to verify if the energy consumptions are similar also for the considered nonlinear MIMO controlled system, in presence of a straight displacement with trapezoidal speed law. The dimensional settling energy for the position-controlled 3-CPU is:

$$E_s = \sum_{i=1}^{3} \int_0^\infty F_i^2 \, dt$$  \hspace{1cm} (14)

where $F_i$ are the three actuator forces.

The graph of Fig. 12 shows that the energy consumption is quite constant for the different gain sets, especially if we exclude the last (pure PD$^{1/2}$), while it is influenced by the trajectory direction. This demonstrates that the adoption of the PDD$^{1/2}$ allows to reduce remarkably the trajectory tracking error with respect to the PD (-38.5% for the gain set c, mean value for the nine considered trajectories, Tab. 5) with the same maximum control output and with a negligible increase of the energy consumption (+4.3% for the gain set c, mean value for the nine considered trajectories, Tab. 5).

6. Conclusion

In the control of a second-order purely inertial system, the combination of proportional, derivative and half-derivative terms (PDD$^{1/2}$) shows some advantages over a classic PD scheme: in the transient state the settling time and the rise time for a step response are lower, in a comparison with the equal settling energy (Bruzzone and Fanghella, 2012). In the present paper, different
combinations of derivative and half derivative terms have been compared in different conditions:
- the controlled system is not a linear SISO, but a coupled and nonlinear MIMO, in particular a 3-CPU parallel kinematics machine;
- the control output is limited;
- the reference motion set-point is not a step, but a rectilinear displacement with variable direction and trapezoidal speed law as usual in real mechatronic applications.

The performed simulation have shown that the same $\zeta$-$\psi$ combinations which reduce the settling time and the rise time in the step response of a second-order linear system are appropriate also for the application in the three control loops of the robot actuators; in particular, the maximum tracking error (distance between the reference and actual end-effector position) is reduced significantly (up to -38% with $\zeta = 0.8$ and $\psi = 2.193$) with a comparable settling energy (+4%).

These results show that the PDD$^{1/2}$ scheme (or the corresponding PIDD$^{1/2}$ scheme, adding the integral term to reduce the steady-state error) are a good option to control also nonlinear MIMO mechatronic system; in the following of the research these fractional-order schemes will be applied to existing PKM prototypes.

References