Fractional order position control of SCARA robots

Luca Bruzzone, Giorgio Bozzini

Department of Mechanics and Design of Machines (DIMEC),
University of Genoa, Italy
E-mail: {bruzzone, bozzini}@dimec.unige.it

Keywords: Fractional calculus, position control, SCARA robot.

SUMMARY. Fractional calculus (FC) considers derivatives and integrals of a non-integer order. In this paper the application of fractional calculus to robot control is considered, in particular with reference to the $PDD^{1/2}$ scheme, which is characterized by the proportional, derivative and half-derivative terms. The PD and $PDD^{1/2}$ schemes are compared with reference to position control of a 4-degree-of-freedom SCARA robot. The results show that the adoption of the $PDD^{1/2}$ control is an interesting option to minimize the tracking error in serial robots and other mechatronic devices.

1 INTRODUCTION

Fractional calculus (FC) is a possible extension of classical mathematics; it considers derivatives and integrals of a non-integer order (real or even complex) [1-3]. This mathematical problem was investigated in the eighteenth and nineteenth centuries by many mathematicians, such as Euler and Liouville; nevertheless, the first practical applications are recent; for instance, application fields are physics, biology, electronics, control system design [4-6]. In this paper, attention is focused on fractional-order control of robots.

The application of FC to control system design is usually based on the $PI^{\lambda}D^{\mu}$ scheme, in which the derivative and the integral terms are generalized to a non-integer order; with this scheme, there are five parameters (three gains and two orders) to tune the system behaviour, and this may lead to benefits [7].

The proposed approach is different: the derivative term is maintained and the *half-derivative term* (derivative of order 1/2) is added, giving rise to the $PIDD^{1/2}$ scheme [8, 9, 10]. The main justification of this approach is practical: the well-known and trustworthy PID scheme is unlikely to be abandoned by industrial robot designers; on the other hand, the addition of the half-derivative term is more easy to be accepted by designers and end-users if its benefits are verified.

In the paper, the influence of the integral term on the steady state error is not considered, and the attention is focused on the comparison between the PD and the $PDD^{1/2}$ schemes in transient state. These two schemes are compared with reference to position control of a 4-degree-of-freedom SCARA robot; in particular, three possible position control schemes are considered [11]:

- the *joint-based control*
- the *inverse Jacobian control*
- the *transpose Jacobian control*

2 THEORETICAL DEFINITION OF HALF-DERIVATIVE AND IMPLEMENTATION OF THE PDD$^{1/2}$ DISCRETE-TIME CONTROL SCHEME

There are different possible definitions of fractional order derivatives; for control system design, the Letnikov definition has some theoretical advantages that lead to a compact discrete-time implementation, expressed in the *z*-domain by the following transfer function [12]:

$$D^{\mu}\left(z^{-1}\right) \cong \frac{1}{T^{\mu}} \sum_{k=0}^{\infty} (-1)^{k} \frac{\Gamma(\mu+1)}{k!\Gamma(\mu-k+1)} z^{-k} \quad (1)$$

In (1) T is the sampling period and Γ is the *gamma function*, which generalizes the factorial function to real and complex numbers. Equation (1) is characterized by an infinite number of terms, but it can be truncated to the sixth order with negligible error. The sixth order truncation of the half-derivative ($\mu = 1/2$) is expressed in the *z*-domain by the following expression:

$$D^{1/2}\left(z^{-1}\right) = \sqrt{\frac{1}{T}}\left(1-\frac{1}{2}z^{-1}-\frac{1}{8}z^{-2}-\frac{1}{16}z^{-3}-\frac{5}{128}z^{-4}-\frac{7}{256}z^{-5}-\frac{21}{1024}z^{-6}\right) \quad (2)$$

Using equation (2), it is possible to implement the discrete-time PDD$^{1/2}$ control system. Figure 1 shows the scheme of the application of this control to a purely inertial SISO system.

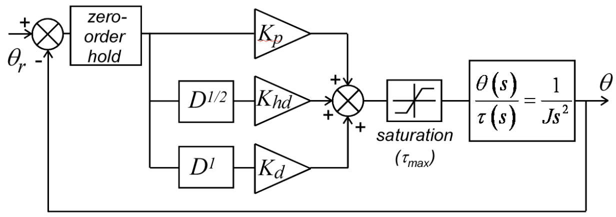

Figure 1: PDD$^{1/2}$ control with saturation of a second-order linear system.

The error $e = \theta_r - \theta$ is processed by a zero-order hold with sampling time T; besides the usual proportional and derivative terms (K_p and K_d gains), the half-derivative term is added (K_{hd} gain); the $D^{1/2}$ discrete-time transform is calculated by means of equation (2), while the D^1 discrete-time transform is the following well-known equation:

$$D^{1}\left(z^{-1}\right) = \frac{1-z^{-1}}{T} \quad (3)$$

The maximum absolute value of the control output is limited by a saturation function.

The purely inertial second order linear system is in continuous time; it is characterized by the mass moment of inertia J; its dynamic behaviour is represented by the Laplace transfer function:

$$\frac{\theta(s)}{\tau(s)} = \frac{1}{Js^2} \quad (4)$$

Figure 2 shows the comparison of the discrete-time PD and $PDD^{1/2}$ schemes in the control of such system. The proportional gain K_p of both the PD and the $PDD^{1/2}$ controls is defined as function of the desired natural angular frequency ω_n:

$$K_p = J\omega_n^2 \tag{5}$$

The output saturation value (τ_{max}) of both the PD and $PDD^{1/2}$ controls is defined starting from the maximum angular acceleration α_{max}:

$$\tau_{\max} = J\alpha_{\max} \tag{6}$$

Once selected the value of K_{hd} (null for the PD control) the derivative gain K_d is determined choosing its minimum value that provides a stabilization without overshoot in presence of a step of θ_r [10].

The entity of the gains K_d and K_{hd} can be expressed by two nondimensional coefficients ξ and γ [10]:

$$\xi = \frac{K_d}{2\sqrt{JK_p}} \tag{7}$$

$$\gamma = \frac{K_{hd}}{K_p^{3/4} J^{1/4}} \tag{8}$$

In Figure 2 five cases are considered, characterized by five different values of ξ and γ, according to Table I; the two nondimensional coefficient are reciprocally constrained by the hypothesis of stabilization without overshoot in presence of step.

As discussed in [10], the influence of the half-derivative term (which increases in the gain sets from 0 to 4) allows to reduce the settling time under the same limitations of null overshoot and maximum control output. This is due to the fact that if the half-derivative term increases the control output tends to the one of a bang-bang control (Figure 2, right), which minimizes the settling time of a second-order linear system. The adoption of the $PDD^{1/2}$ control algorithm allows to lower remarkably the settling time to within 2%, up to about –40%, using the same maximum torque.

Table I: values of the nondimensional coefficients ξ and γ for the gain sets 0 to 4.

Gain Set	ξ	γ
0 (PD)	0.96	0
1 ($PDD^{1/2}$)	1.02	0.24
2 ($PDD^{1/2}$)	1.10	0.59
3 ($PDD^{1/2}$)	1.20	1.18
4 ($PDD^{1/2}$)	1.49	2.36

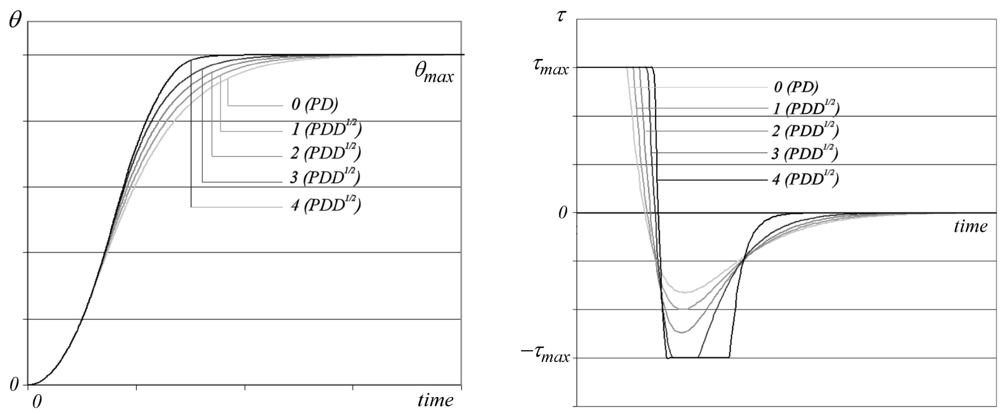

Figure 2: Influence of the half-derivative term on the system behaviour: angular position θ (left) and control output τ (right) in presence of a commanded step from 0 to $\theta_{\max}$.

3 PDD$^{1/2}$ CONTROL OF A SCARA ROBOT

Figure 3 shows the kinematic scheme and the geometric parameters of a SCARA robotic architecture; the robot kinematics is fully defined by the four parameters l_0, l_1, l_2, l_3; l_{G1} and l_{G2} represent the positions of the centres of gravity G_1 and G_2 of the links 1 and 2.

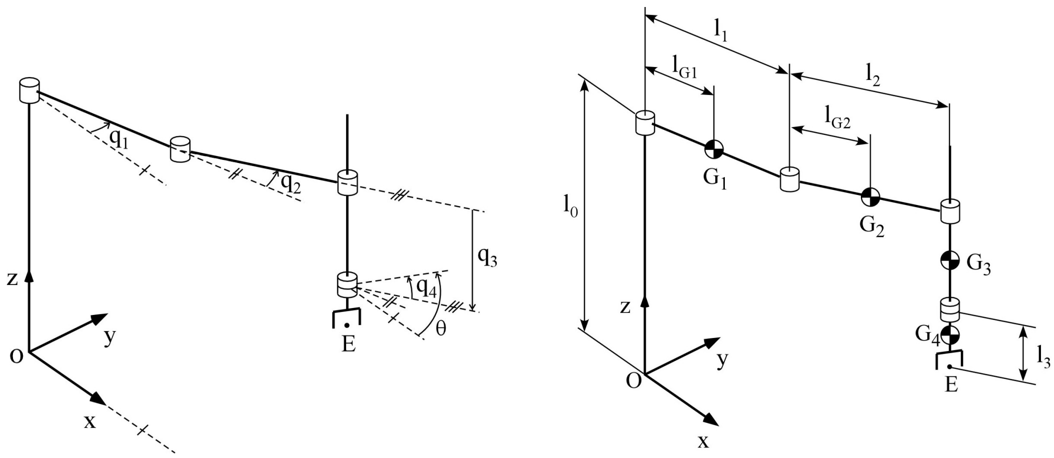

Figure 3: Kinematic scheme and the geometric parameters of the SCARA robot.

The dynamic model of a SCARA robot without friction is expressed by the following equation [11]:

$$\boldsymbol{\tau} = \mathbf{H}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q},\dot{\mathbf{q}}) + \mathbf{G}(\mathbf{q}) - \mathbf{J}^T(\mathbf{q})\mathbf{F} \tag{9}$$

In equation (9):

- $\mathbf{q} = [q_1, q_2, q_3, q_4]^T$ is the vector of the internal coordinates, composed of three angles, q_1, q_2, and q_4, and one displacement, q_3 (see Figure 3);
- $\boldsymbol{\tau} = [\tau_1, \tau_2, \tau_3, \tau_4]^T$ is the vector of the actuator generalized forces, composed of three torques, τ_1, τ_2, and τ_4, and one force, τ_3;
- $\mathbf{H}(\mathbf{q})$ is the inertia matrix;
- $\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})$ is the vector of the centrifugal and Coriolis terms;
- $\mathbf{G}(\mathbf{q})$ is the vector of the gravity terms;
- $\mathbf{J}(\mathbf{q})$ is the Jacobian matrix;
- $\mathbf{F} = [F_x, F_y, F_z, M_z]^T$ is vector of the generalized forces that the environment applies to the end-effector, which correspond to the external coordinates $\mathbf{x}$.

The external degrees of freedom are represented by the vector of the external coordinates $\mathbf{x} = [x, y, z, \theta]^T$, composed of the coordinates of the point E in the fixed reference frame and of the end-effector angle $\theta = q_1 + q_2 + q_4$ (Figure 3).

The position control of a generic serial robot can be performed according to different schemes [11]. The simplest approach is the *joint-based control*, which requires the inverse kinematics transform because trajectory planning is generally available in terms of external coordinates.

An alternative approach is the *Cartesian-based control*, in the external coordinates; the Cartesian-based control can be realized in two ways: the *inverse Jacobian control* and the *transpose Jacobian control*. In the following these three approaches will be compared with reference to the SCARA architecture.

Figures 4 to 6 show respectively the joint-based PDD$^{1/2}$ position control scheme, the inverse Jacobian PDD$^{1/2}$ position control scheme and the transpose Jacobian PDD$^{1/2}$ position control scheme of the SCARA.

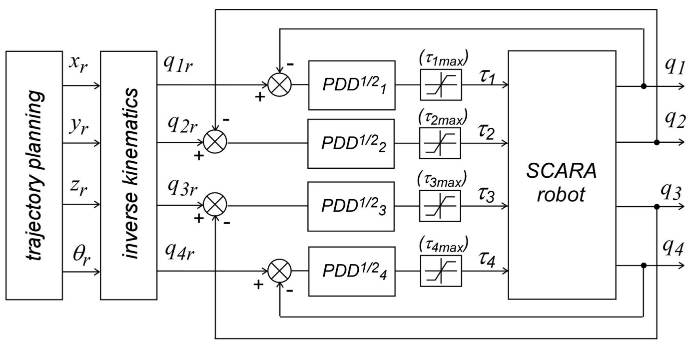

Figure 4: Joint-based PDD$^{1/2}$ position control of the SCARA robot.

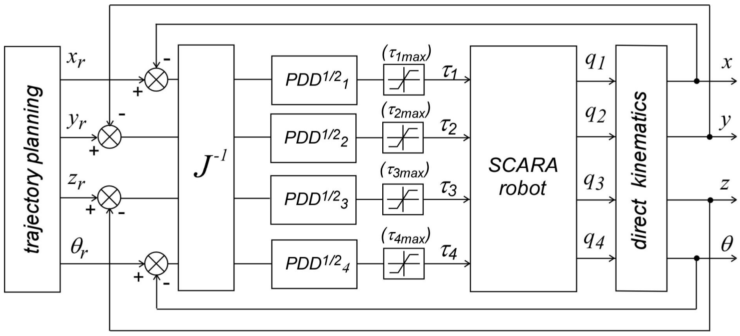

Figure 5: Inverse Jacobian $PDD^{1/2}$ position control of the SCARA robot.

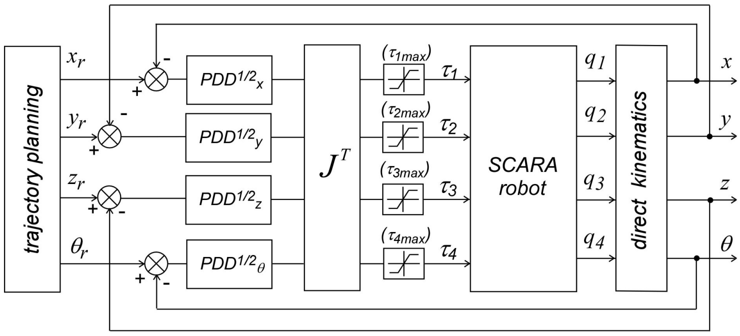

Figure 6: Transpose Jacobian $PDD^{1/2}$ position control of the SCARA robot.

It is possible to demonstrate that the joint-based and the inverse Jacobian control laws are theoretically equivalent out of singularities, if the gains are tuned according to the same criteria [9]; the only differences between the two control systems are due to the implementation in the real hardware; therefore in the following only the simulation results of the inverse Jacobian control and of the transpose Jacobian control will be reported, because the simulation results of the joint-based control are equal to the ones of the inverse Jacobian control.

4 SIMULATION RESULTS

In the simulations the trajectory has been defined following the ISO standards for the test trajectories of industrial robots: the maximum cube inscribed in the robot workspace (considering the inner workspace radius equal to $l_1/2$) is represented in Figure 7; one of its diagonal planes passes through the vertices A, B, C, D; the simulation trajectory A'C'B'D' connects the vertices of the square with the same diagonals of ABCD and side A'B' = 0.8·AB. The trajectory is followed with constant end-effector orientation θ; the reference end-effector speed v is constant to emphasize the influence of the control loops on the tracking error; even if this motion planning is not profitable from a practical point of view (a classical trapezoidal speed law for each line segment reduces remarkably the tracking error peaks in the vertices), it is used to highlight the effects of the half-derivative term on the trajectory tracking capability. The robot geometrical and inertial parameters and the trajectory coordinates are summarized in Table II.

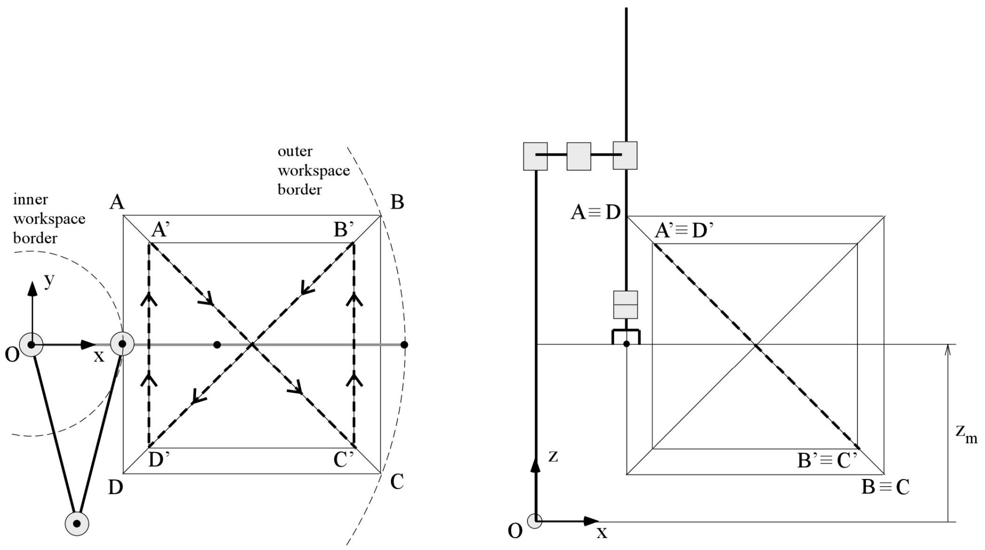

Figure 7: Simulation trajectory.

Table II: Geometrical and inertial parameters and trajectory points.

Geometrical Parameters			
Length of link 0 (l_0)			1 m
Length of link 1 (l_1)			0.5 m
Length of link 2 (l_2)			0.5 m
Length of link 4 (l_3)			0.1 m
Distance of the c.o.m G_1 from the first joint (l_{G1})			0.25 m
Distance of the c.o.m G_2 from the second joint (l_{G2})			0.25 m
Inertial Parameters			
Mass of link 1 (m_1)			20 kg
Mass of link 2 (m_2)			10 kg
Mass of link 3 (m_3)			3.5 kg
Mass of link 4 (m_4)			1.5 kg
Moment of inertia of link 1 around its barycentric vertical axis (I_{G1})			0.467 kg·m^2
Moment of inertia of link 2 around its barycentric vertical axis(I_{G2})			0.230 kg·m^2
Moment of inertia of link 3 around its barycentric vertical axis(I_{G3})			0.010 kg·m^2
Moment of inertia of link 4 around its barycentric vertical axis(I_{G4})			0.005 kg·m^2
Trajectory points	*x* coordinate	*y* coordinate	*z* coordinate
A'	*0.319 m*	*0.276 m*	*0.752 m*
C'	*0.871 m*	*-0.276 m*	*0.200 m*
B'	*0.871 m*	*0.276 m*	*0.200 m*
D'	*0.319 m*	*-0.276 m*	*0.752 m*

For the inverse Jacobian and the joint-based schemes, the comparison between the PD and the PDD$^{1/2}$ is carried out following the same criteria used in section 2 for the one-degree-of-freedom rotational system (same maximum force/torque, derivative gain tuned to its minimum value that assures null overshoot in presence of a step) [10]. It is possible to tune the gains using a derived approach with the SCARA architecture, even if it is a coupled non-linear system, with these assumptions:

- for actuator 1 we consider the moment of inertia of the links 1 to 4 with q_2 and q_4 constantly zero;
- for actuator 2 we consider the moment of inertia of the links 2 to 4 with q_4 constantly zero;
- for actuator 3 we consider the translating mass (m_3+m_4);
- for actuator 4 we consider the moment of inertia of link 4.

If the inertial terms are considered constant, the gains of the four actuators can be tuned as for four decoupled second-order linear systems. According to this approach, in the simulations:

- the proportional gains K_{p1} to K_{p4} are obtained by equation (5) with ω_n=10 rad/s;
- the derivative gains K_{d1} to K_{d4} and the half-derivative gains K_{hd1} to K_{hd4} are tuned according to the null-overshoot criterion, obtaining the couples of ξ and γ coefficients of table III;
- the saturation values are obtained by equation (6) with α_{max} = 10π rad/s^2 for actuators 1, 2 and 4 and a_{max} = $\alpha_{max}\cdot(l_1+l_2)$ for actuator 3;
- the sampling time is T= 0.01 s.

The six gain sets *a* to *f* have different levels of half-derivative term (null for the gain set *a*). The tracking error with the six gain sets is shown in Figures 8 and 9 for two different end-effector speeds (0.25 m/s and 0.5 m/s).

The simulation results show that the introduction of the half-derivative gain allows to reduce the tracking error under the same limitation of maximum actuator moments/forces.

Table III: Control parameters.

Gain set	*a* (PD)	*b* (PDD$^{1/2}$)	*c* (PDD$^{1/2}$)	*d* (PDD$^{1/2}$)	*e* (PDD$^{1/2}$)	*f* (PDD$^{1/2}$)
γ	0	0.5	1	2	3	4
ξ	0.92	0.94	0.92	0.84	0.75	0.8

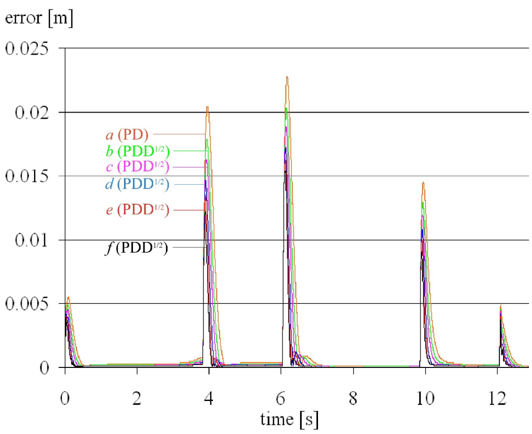

Figure 8: Tracking error with end-effector speed v = 0.25 m/s.

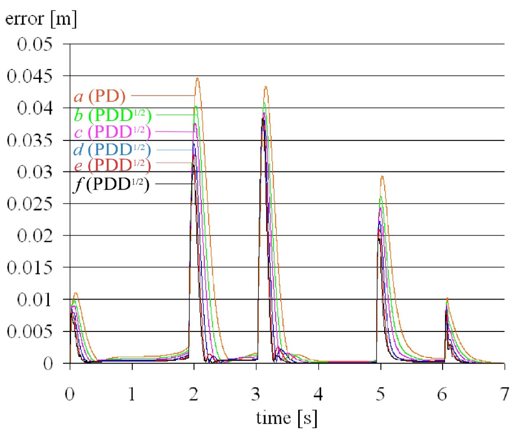

Figure 9: Tracking error with end-effector speed v = 0.5 m/s.

Adopting the transpose Jacobian control, it is not possible to tune the gains according to the same criteria used for second-order single d.o.f. system and extended to the joint-based and inverse Jacobian controls, because the PDD$^{1/2}$ gains are applied to errors in Cartesian space, and it is impossible to identify equivalent constant masses/inertias. Therefore, the effect of the half-derivative gain is assessed by measuring extensively the maximum tracking error with different combinations of proportional, derivative and half-derivative gains.

In order to have a homogeneous behaviour we assume $K_p = K_{px} = K_{py} = K_{pz}$, $K_d = K_{dx} = K_{dy} = K_{dz}$, and $K_{hd} = K_{hdx} = K_{hdy} = K_{hdz}$. Let us note that the gains $K_{p\theta}$, $K_{d\theta}$ and $K_{hd\theta}$ are not influential since the simulation trajectory is with constant end-effector orientation and friction is negligible; therefore the torque of actuator 4 is constantly null. The 3D plot of Figure 10 shows the values of maximum tracking error along the trajectory A'C'B'D', with constant speed v = 0.25 m/s, for K_p ranging from 200 to 800 N/m and K_d ranging from 50 to 200 Ns/m. The four surfaces are related to different values of K_{hd}: 0 Ns$^{1/2}$/m (PD), 25 Ns$^{1/2}$/m, 50 Ns$^{1/2}$/m, 100 Ns$^{1/2}$/m. It is possible to note that the introduction of the half-derivative term reduces the maximum tracking error.

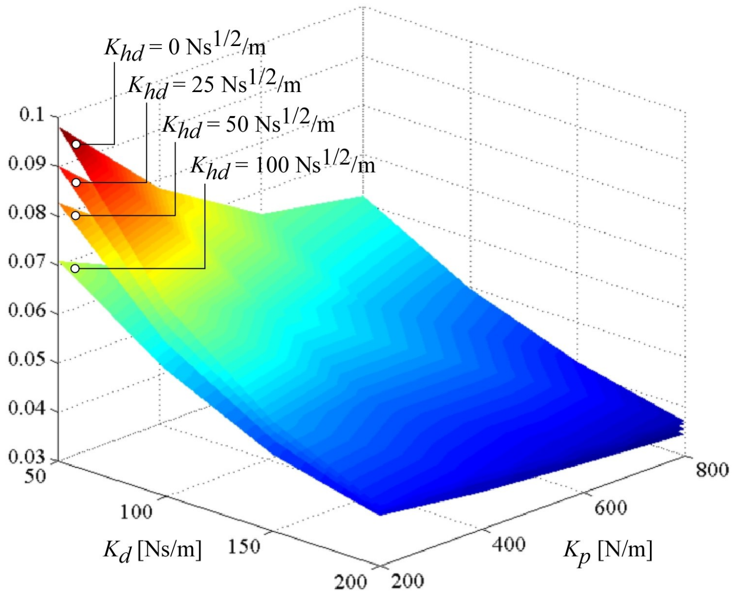

Figure 10: Maximum tracking error as function of the control gains with v = 0.25 m/s.

5 CONCLUSIONS

In the proposed work, the PDD$^{1/2}$ control scheme has been applied to position control of a common industrial robot, the SCARA. The PDD$^{1/2}$ algorithm has been introduced in three different position control schemes: the *Cartesian-based control*, the *inverse Jacobian control*, the *transpose Jacobian control*. The behaviour of the SCARA robot has been assessed by simulation with reference to the trajectory prescribed by the ISO standards for the performance test of industrial robots. The results show that the introduction of the half-derivative term reduces the tracking error. The PDD$^{1/2}$ algorithm and its possible PIDD$^{1/2}$ extension seem to be interesting options for position control of industrial robots and other mechatronic devices, which do not completely revolutionize the well-known and accepted PD-PID schemes, but integrate them. In the following of the work several issues will be investigated, such as the effects of the introduction of the integral term and the influence of the half-derivative term in presence of friction.

References

[1] Miller, K.S. and Ross, B., *An Introduction to the Fractional Calculus and Fractional Differential Equations,* Wiley, New York (1993).

[2] Oldham, K.B., and Spanier, J., *The fractional calculus: theory and application of differentiation and integration to arbitrary order,* Academic Press, New York (1974).

[3] Podlubny, I., *Fractional Differential Equations,* Academic Press, San Diego (1999).

[4] Hilfer, R., *Applications of Fractional Calculus in Physics*, World Scientific, Singapore (2000).

[5] Anastasio, T.J., "The fractional-order dynamics of brainstem vestibulo-oculomotor neurons", *Biological cybernetics,* **72**, 69-79 (1994).

[6] Oustaloup, A., "Fractional order sinusoidal oscillators: optimization and their use in highly linear FM modulation", *IEEE Transactions on Circuits and Systems*, **28**(10), 1007-1009 (1981).

[7] Podlubny, I., "Fractional-order systems and $PI^{\lambda}D^{\mu}$-controllers", *IEEE Transactions on Automatic Control,* **44**(1), 208–213 (1999).

[8] Bruzzone, L., and Bozzini, G., "Application of fractional calculus to robotics: the $PDD^{1/2}$ control", in *Proc. of RAAD 2008, 17th International Workshop on Robotics in Alpe-Adria-Danube Region*, Ancona, Italy, September 15-17, 2008, CD (2008).

[9] Bruzzone, L., and Bozzini, G., "Application of the $PDD^{1/2}$ algorithm to position control of serial robots", in *Proc. of the 28th IASTED Intl. Conf. On Modelling, Identification and Control MIC 2009*, Innsbruck, Austria, February 16-18, 2009, 225-230 (2009).

[10] Bruzzone, L., and Bozzini, G., "Fractional-order derivatives and their application to the position control of robots", *International Journal of Mechanics and Control*, **10**(1), 39-44 (2009).

[11] Craig, J.J., *Introduction to robotics: Mechanics and Control (3rd Edition),* Prentice Hall, Upper Saddle River, NJ (2005).

[12] Machado, J.A.T., "Fractional-order derivative approximations in discrete-time control systems", *Journal of Systems Analysis-Modelling-Simulation,* **34**, 419-434 (1999).