Using Fractional Calculus for Lateral and Longitudinal Control of Autonomous Vehicles

J. I. Suárez[1], B. M. Vinagre[1], A. J. Calderón[1], C. A. Monje[1], and Y. Q. Chen[2]

[1] EII, Universidad de Extremadura, Badajoz - Spain
`{jisuarez, bvinagre, ajcalde, cmonje}@unex.es`
[2] CSOIS, Utah State University, Logan, Utah - USA
`yqchen@ece.usu.edu`

Abstract. Here it is presented the use of Fractional Order Controllers (FOC) applied to the path-tracking problem in an autonomous electric vehicle. A lateral dynamic model of a industrial vehicle has been taken into account to implement conventional and Fractional Order Controllers. Several control schemes with these controllers have been simulated and compared . First, different controllers with similar parameters have been implemented and then they have been improved by using optimization methods. The preliminary results are presented here.

1 Introduction

Path-tracking problems in autonomous vehicles and mobile robotics have been investigated for the last two decades. Some methods proposed can be basically divided into *temporal* (based on the application of control theory) and *spatial* controllers (based on geometric methods such as *pure-pursuit*) (see [1] and [2] for additional references). In this work, a spatial path tracking method, called the *ϵ-controller*, is applied which was first proposed in [3]. This path tracking method computes the normal distance from the vehicle to the desired path, ϵ, and generates a desired velocity vector for the vehicle to follow the path.

To improve its performance, several regulation schemes using the fractional-order control (FOC) idea [4] have been investigated here. FOC is based on "fractional-order calculus". Recent books [5–8] provide a good source of references on fractional-order calculus. However, applying fractional order calculus to dynamic systems control is just a recent focus of interest [9–14]. For pioneering works, we cite [4, 15–17]. For the latest development of fractional calculus in automatic control and robotics, we cite [18]. As for path tracking problems, the first experiences in path-tracking applied to XY cutting tables and mobile robotics can be found in [19] and [20].

In this paper we present some preliminary results of the use of FOC in path-tracking problems applied to an autonomous electric vehicle by using an ϵ-controller on path-tracking basic algorithm.

The rest of the paper is organized as follows: in Sec. 2 an introduction to fractional calculus is made. In Sec. 3 the lateral dynamic model of the vehicle is shortly described. In Sec. 4 the ϵ-controller is briefly introduced. Section 5 presents several control schemes (P, PI, PI$^{\alpha}$,...) with a special attention to FOC. Section 6 shows some of our simulated results. Finally, in Sec. 7, some conclusions are outlined.

2 Introduction to Fractional Calculus

Even though the idea of fractional order operators is as old as the idea of the integer order ones is, it has been in the last decades when the use of fractional order operators and operations has become more and more popular among many research areas. The theoretical and practical interest of these operators is nowadays well established, and its applicability to science and engineering can be considered as emerging new topics. Even if they can be thought of as somehow ideal, they are, in fact, useful tools for both the description of a more complex reality, and the enlargement of the practical applicability of the common integer order operators. Among these fractional order operators and operations, the fractional integro-differential operators (fractional calculus) are specially interesting in automatic control and robotics.

2.1 Fractional Order Operators

Fractional calculus is a generalization of integration and differentiation to non-integer (fractional) order fundamental operator ${}_a\mathrm{D}_t^{\alpha}$, where a and t are the limits and α, $(\alpha \in \mathbb{R})$ the order of the operation. The two definitions used for the general fractional integro-differential are the Grünwald-Letnikov (GL) definition and the Riemann-Liouville (RL) definition. The GL definition is that

$${}_a\mathrm{D}_t^{\alpha} f(t) = \lim_{h\rightarrow 0} h^{-\alpha} \sum_{j=0}^{[\frac{t-a}{h}]} (-1)^j \binom{\alpha}{j} f(t-jh) \tag{1}$$

where $[\cdot]$ means the integer part, while the RL definition is

$${}_a\mathrm{D}_t^{\alpha} f(t) = \frac{1}{\Gamma(n-\alpha)} \frac{\mathrm{d}^n}{\mathrm{d}t^n} \int_a^t \frac{f(\tau)}{(t-\tau)^{\alpha-n+1}} \mathrm{d}\tau \tag{2}$$

for $(n-1 < r < n)$ and where $\Gamma(\cdot)$ is the Euler's *gamma* function.

For convenience, Laplace domain notion is usually used to describe the fractional integro-differential operation. The Laplace transform of the RL fractional derivative/integral (2) under zero initial conditions for order α, $(0 < \alpha < 1)$ is given by

$$\mathcal{L}\{{}_a\mathrm{D}_t^{\pm\alpha} f(t); s\} = s^{\pm\alpha} F(s). \tag{3}$$

2.2 Fractional Order Control Systems

In theory, the control systems can include both the fractional order dynamic system to be controlled and the fractional-order controller. A fractional order plant to be controlled can be described by a typical n-term linear FODE in time domain

$$a_n \, \mathrm{D}_t^{\beta_n} y(t) + \cdots + a_1 \, \mathrm{D}_t^{\beta_1} y(t) + a_0 \, \mathrm{D}_t^{\beta_0} y(t) = 0 \tag{4}$$

where $a_k (k = 0, 1, \cdots, n)$ are constant coefficients of the FODE; β_k, (k=0, 1, 2, ···, n) are real numbers. Without loss of generality, assume that $\beta_n > \beta_{n-1} > \cdots > \beta_1 > \beta_0 \geq 0$. Consider a control function which acts on the FODE system (4) as follows:

$$a_n \, \mathrm{D}_t^{\beta_n} y(t) + \cdots + a_1 \, \mathrm{D}_t^{\beta_1} y(t) + a_0 \, \mathrm{D}_t^{\beta_0} y(t) = u(t). \tag{5}$$

By Laplace transform, we can get a fractional transfer function :

$$G_p(s) = \frac{Y(s)}{U(s)} = \frac{1}{a_n s^{\beta_n} + \cdots + a_1 s^{\beta_1} + a_0 s^{\beta_0}}. \tag{6}$$

In general, a fractional-order dynamic system can be represented by a transfer function of the form:

$$G_p(s) = \frac{Y(s)}{U(s)} = \frac{b_m s^{\alpha_m} + \cdots + b_1 s^{\alpha_1} + b_0 s^{\alpha_0}}{a_n s^{\beta_n} + \cdots + a_1 s^{\beta_1} + a_0 s^{\beta_0}} \tag{7}$$

However, in control practice, more common is to consider the fractional order controller. This is due to the fact that the plant model may have already been obtained as an integer order model in classical sense. In most cases, our objective is to apply the fractional order control (FOC) to enhance the system control performance. Taking conventional PID-controller as an example, its fractional order version, $\mathrm{PI}^{\lambda}\mathrm{D}^{\mu}$ controller, was studied in time domain and in frequency domain. The time domain formula is that

$$u(t) = K_p e(t) + T_i \mathrm{D}_t^{-\lambda} e(t) + T_d \mathrm{D}_t^{\mu} e(t). \qquad (\mathrm{D}_t^{(*)} \equiv_0 \mathrm{D}_t^{(*)}). \tag{8}$$

It can be expected that $\mathrm{PI}^{\lambda}\mathrm{D}^{\mu}$ controller (8) may enhance the systems control performance due to more tuning knobs introduced.

3 Vehicle Dynamic Model

The vehicle used in our work is a Citroën Berlingo with an Ackerman steering system [21]. For modeling the lateral dynamics of the vehicle a body-fixed coordinate systems (BFCS) is fixed to its center of gravity (CG) and the roll, pitch, bounce and deceleration dynamics are neglected. A linear model can be obtained by solving the dynamic equations and further simplifications are done [22]. We assume for simplicity that the both front wheel turn the same amount of angle and hence each wheel produces the same steering forces. The resulting model is known as the bicycle dynamics model. Figure 1 shows a diagram where the main parameters and variables are depicted. They are the following ones:

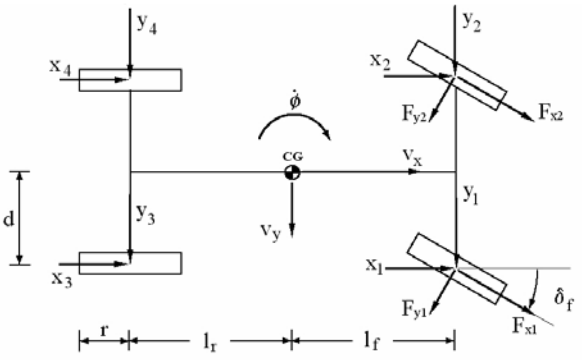

Fig. 1. Acerkman steered vehicle and its related forces for the lateral dynamic model

- v_x, v_y: longitudinal and lateral velocity, respectively.
- δ_f: front wheel steering angle
- $\dot{\phi}$: yaw rate
- m: vehicle mass
- I_z: moment of inertia about the z-axis
- c_f, c_r: front and rear wheel cornering stiffness
- l_f, l_r: distances of the front and rear axles from the CG
- d: distance from car centerline to each wheel (half-track)
- r: wheel radius

The coordinate system uses the convention of the Society of Automotive Engineers with the z-axis pointing into the road and the positive yaw direction as depicted in Fig.1. The longitudinal velocity v_x is supposed to be approximately constant. Choosing $\dot{\phi}$ and v_y as state variables the vehicle model can be expressed by the following state equations:

$$\begin{bmatrix} \dot{v}_y \\ \ddot{\phi} \end{bmatrix} = \begin{bmatrix} -2\frac{a_1}{mv_x} & -v_x - 2\frac{a_2}{mv_x} \\ -2\frac{a_3}{I_z v_x} & -2\frac{a_4}{I_z v_x} \end{bmatrix} \cdot \begin{bmatrix} v_y \\ \dot{\phi} \end{bmatrix} + \begin{bmatrix} b_1 \\ b_2 \end{bmatrix} \delta_f \tag{9}$$

where $a_1 = c_f + c_r$; $a_2 = c_f l_f - c_r l_r = a_3$; $a_4 = c_f l_f^2 + c_r l_r^2$; $b_1 = 2\frac{c_f}{m}$; $b_2 = 2\frac{c_f l_f}{I_z}$.

Parameters for the Citroën Berlingo used in our simulations are: $m = 1466Kg$; $I_z = 28000Nm^2$; $c_f = c_r = 60000N/rad$; $l_f = 1.12m$; $l_r = 1.57m$

4 Path-tracking Algorithm

The path-tracking problem was accomplished by using the scalar ϵ-controller [3]. Basically, it is a regulator, with the cascade control architecture shown in Fig.2, that operates on the vehicle normal deviation ϵ from the desired path. The ϵ-controller (C_ϵ) generates a desired velocity vector $\mathbf{V}_I^*$ which depends on the lateral deviation from the path. If the vehicle is near the path the ϵ-controller

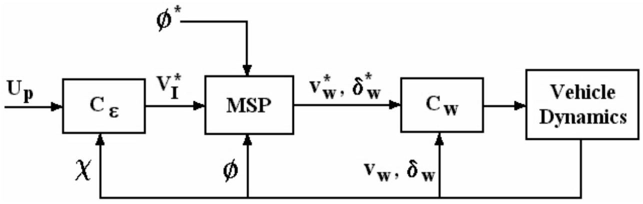

Fig. 2. ϵ-Controller cascade control achitecture

generates a velocity vector tangent to the path and permits the vehicle to travel at its maximum speed. However, when the vehicle is far from the desired path, the direction of $\mathbf{V}_I^*$ points to the closest point on the path in the radial direction and then the velocity of the vehicle is reduced.

The *MakeSetPoint* (MSP) algorithm converts the desired velocity vector $\mathbf{V}_I^*$ into body-fixed longitudinal velocity (v_x^*) and steering angle (δ_w^*) setpoints. This is accomplished by rotating the velocity vector, given in an Inertial Cartesian coordinate System (ICS), into the vehicle-fixed coordinate system. Then the longitudinal velocity and the steering angle setpoints are obtained from the vehicle geometry. The low-level controllers (C_w) track the actuator-level setpoints. A detailed description of the algorithm can be found in [3] and [21].

5 Regulation schemes

The control law of the ϵ-controller turns the two-dimensional path tracking problem into a scalar regulation. It is a nonlinear controller operating on the scalar ϵ, then, several control schemes (P, PI, ...) can be implemented. In the following subsections three different strategies of regulation are presented and compared each other: P, PI and fractional PI$^\alpha$ controllers.

5.1 P Controller

In this regulation method the control signal is proportional to the lateral distance from the path, then

$$u(t) = K_P \epsilon(t) \tag{10}$$

This is a simple regulator, but presents a stationary error. Hence, by increasing the proportional gain (K_P) the vehicle can be stabilized, but the error can not be driven to zero. Furthermore, the bigger the proportional gain, the faster the vehicle turns to path. For these reasons, it is necessary to use other regulators.

This controller has been implemented as a digital P regulator, with a sample period $T = 0.1$ seconds. The transfer function is:

$$G_P(z) = K_P \tag{11}$$

5.2 PI and PI$^{\alpha}$ Controller

In these kind of controllers the control law can be obtained as follows

$$u(t) = K_P \epsilon(t) + K_I I^{\alpha} \epsilon(t) \tag{12}$$

being

$$I^{\alpha} f(t) \equiv \frac{1}{\Gamma(\alpha)} \int_0^t (t-\tau)^{\alpha-1} f(\tau)\, d\tau,\ t > 0,\ \alpha \in \Re^{+} \tag{13}$$

With $1 < \alpha < 2$ we have a fractional controller. However, when $\alpha = 1$, (12) becomes

$$u(t) = K_P \epsilon(t) + K_I \int_0^t \epsilon(\tau)\, d\tau \tag{14}$$

which is the control law corresponding to the classical PI controller. In this case, the proportional gain must be lowered to reduced the oscillation and the integral term can be used to drive the error to zero.

The PI controller has been implemented with a digital integrator using the trapezoidal rule. The transfer function is:

$$G_{PI}(z) = K_P + K_I \frac{T(z+1)}{2(z-1)} \tag{15}$$

In order to preserve the zero stationary error the PI$^{\alpha}$ controllers have been implemented with two cascade integrators, the first one corresponding $\frac{1}{s}$ and the second one corresponding $\frac{1}{s^{1-\alpha}}$, $\alpha > 1$. The first integrator has been discretized using the Tustin transformation $s = \frac{2(z-1)}{T(z+1)}$. The fractional order integrator $\frac{1}{s^{1-\alpha}}$ has an infinite terms representation and must be discretized using a finite dimensional IIR filter. For our case (see [23]) the method of the Continued Fraction Expansion (CFE) has been used for obtaining a finite dimensional approximation of the fractional power of the Tustin discrete equivalent.

6 Simulations

6.1 Simulink Model

The high-level Simulink model shown in Fig. 3 simulates the control architecture depicted in Fig. 2. The *Behavior Generator* subsystem outputs the parameters of the path geometry which are used for the *Sensor Motion Scheduler (SMS)* to provide all the quantities required by the ϵ-controller, in order to keep the vehicle on the desired path. The SMS is implemented together with the ϵ-controller and the MSP algorithm in the second subsystem, whose outputs are the desired velocity and steering angle. The next subsystem includes the steering system, that is modeled as a first order system with a steering angle limitation of 30^{o}, and a low-level PI controller. The two next blocks are the dynamic lateral vehicle

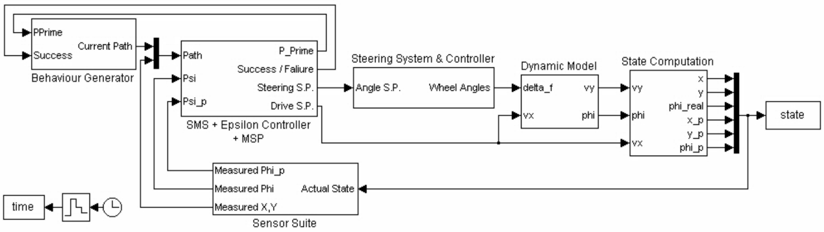

Fig. 3. Simulink model of the vehicle with the ϵ-controller

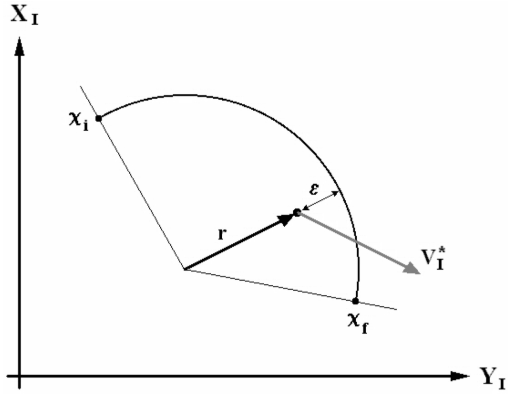

Fig. 4. ϵ-Controller parameters for an arc segment

model and a block that computes the vehicle state. Finally, in the feedback path a sensor suite provides only the necessary state variables to the path-tracking controllers.

6.2 First results

The firsts simulations have been done to compare the perfomance of several kind of controllers. In these simulations the vehicle has been supposed to describe a semicircular path. First, the input vector to C_ϵ block $\mathbf{U}_P = \left[\chi_i \; \chi_f \; r \; V_d\right]^T$ shown in Fig.2 must be established, where χ_i and χ_f are the initial and final points, respectively, r is the radius of the path and V_d is desired vehicle velocity (Fig.4). The parameters chosen in our simulations are: $\chi_i = (0, 0)$, $\chi_f = (0, 50)$, $r = 25m$ and $V_d = 10m/s$. The starting point of the vehicle is $(0, 0)$, then the vehicle describes, at the maximum speed of 10m/s, a semicircular path with radius of 25 metres. In the present case, velocity is not important, for this reason when the vehicle is far from the path the ϵ-controller will decrease the velocity and when it is on the path the vehicle will travel at its maximum speed (10m/s).

Table 1. Controllers Parameters

Controllers	K_P	K_I	α
P	20	0	0
PI	20	5	1
$PI^{1.5}$	20	5	1.5
$PI^{1.25}\,(1)$	20	5	1.25
$PI^{1.25}\,(2)$	22	5	1.25

Table 2. Computed ISE

P	PI	$PI^{1.5}$	$PI^{1.25}\,(1)$	$PI^{1.25}\,(2)$
0.0752	0.0478	0.0558	0.0541	**0.0436**

Five control schemes have been simulated: P, PI, and three FOC, one PI$^{1.5}$ and two PI$^{1.25}$ controllers with different values of K_P. Table 1 shows the parameters used in our simulations for the different controllers. Note that K_P is the same for all the controllers, except for the last one, in which the proportional gain has been increased to show the effects of varying K_P in the fractional controllers. Moreover, all the PI controllers have the same integral gain K_I.The aim of these simulations is to demonstrate that with a controller with an additional tunning parameter a better result can be obtained. It can be thought that a better performance can be reached with a PI controller, with two tunning parameters, than with a P controller with only one tunning parameter. The same consideration can be done with a PI$^{\alpha}$ controller an PI controller.

Figure 5 shows the responses of three tested controllers: P, PI and PI$^{1.25}$ with $K_P = 22$. A positive value of ϵ means the vehicle lies inside the desired path, whereas a negative one means the vehicle is outside the arc. Note the steady state error of the P controller. As explained before, the integral part of the PI controllers tends to eliminate this error.

Figure 6 shows the behavior for the three fractional controllers. For classical PI controller we only can vary two parameters (K_P and K_I). Note that in this case, for FOC an additional parameter (α) can be varied to obtain different responses.

With the aim of comparing the regulation schemes we have computed the integral squared error (ISE) given by:

$$e = \sum_{i=1}^{N} \epsilon_i^2 \tag{16}$$

being this error the distance between the desired path and the actual path that the vehicle travels. Table 2 shows the computed ISE for the different controllers. Note the smaller area for the PI controllers; as it was obvious, by reducing the steady state error, the ISE will be reduced too. PI and PI$^{\alpha}$ controllers present a better performance than P controller and the best one is the PI$^{1.25}$(2).

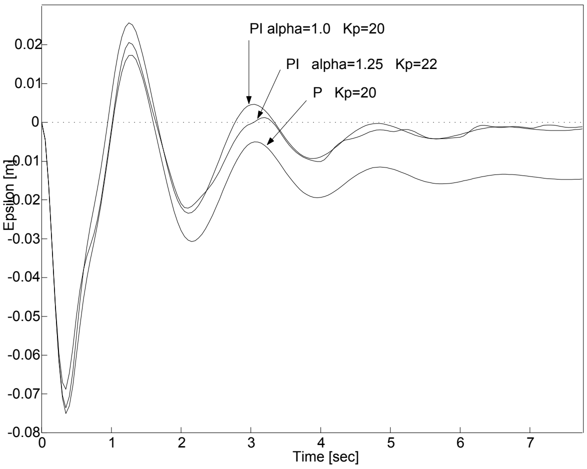

Fig. 5. Deviation ϵ along the path for P, PI and $\mathrm{PI}^{1.25}$ controllers

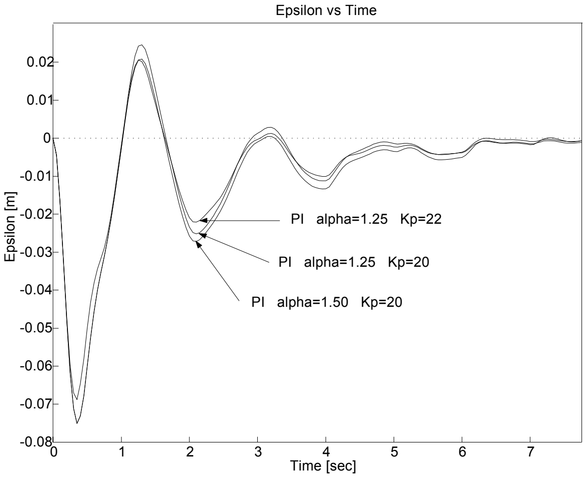

Fig. 6. Deviation ϵ along the path for different FOC

Table 3. Controllers Optimized Parameters

Controllers	K_P	K_I	α
P	42.6934	0	0
PI	38.9035	21.6490	1
PI^α	40.0267	20.1579	1.1217

Table 4. Comparison Parameters

Controllers	ISE	Ov	t_{set}	t_{zc}
P	0.017573	−0.0387	4.0535	0.7822
PI	0.011303	−0.0409	2.0535	0.7767
PI^α	0.011258	−0.0403	2.0101	0.7730

6.3 Optimized results

In the next simulations the previous controllers (P, PI and PI$^\alpha$) have been optimized and four comparison parameters have been stablished to test the behavior of the controllers. They are:

- the *integral squared error* (ISE), named before,
- the *overshoot*, that it is the maximun deviation from the desired path,
- the *settling time*, defined as the time spent to reach and keep inside 0.01 meters from the path,
- and the *first zero-crossing time*, the time to reach to the first zero-crossing or also the first time to reach $\epsilon = 0$.

The optimization of the parameters of the controllers has been done by minimizing the ISE. The optimized parameters are shown in Table 3.

In Table 4 it can been observed that the best performance corresponds to the fractional order controller. It has the least ISE, settling time and first zero-crossing time. As it was expected, with an additional parameter the best results corresponds to the controller with more number of parameters. It can be observed in the fractional controller that with a greater α, the proportional gain increases and the integral gain decreases. These results are depicted in Fig. 7. The PI and PI$^\alpha$ controllers have a similar behavior, but the fractional controller is better.

7 Conclusions

Firstly, several controllers have been compared to show their performance in path-tracking problems using a dynamic model of a Citroën Berlingo vehicle. A novel and simple regulation scheme, fractional PI$^\alpha$ controller, was simulated with better results than the other regulations schemes (P and PI controllers). A sintonization method for the controllers has been presented with the aim of

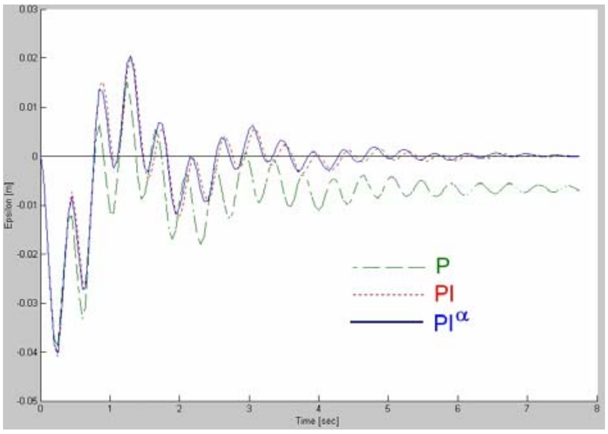

Fig. 7. Controllers responses with their optimal values

improving their performances and obtaining the optimal values for the parameters K_P, K_I and α. The fractional calculus has been demonstrated to be another useful tool for controllers design. New sintonization methods will be looked for in future works, and also low level fractional order controllers will be implemented to improve the vehicle performace.

8 Acknowlegdments

This work has been partially supported by Research Grant DPI 2002-04064C05-03 (MCYT).

References

1. Ollero, A.: Robotica. Manipuladores y Robots Moviles. Marcombo (2001)
2. Wit, J.S.: Vector Pursuit Path Tracking for Autonomous Ground Vehicles. PhD thesis, University of Florida (2000)
3. Davidson, M., Bahl, V.: The Scalar ϵ-Controller: A Spatial Path Tracking Approach for ODV, Ackerman, and Differentially-Steered Autonomous Wheeled Mobile Robots. In: Proceedings of the IEEE International Conference on Robotics and Automation. (2001) 175–180
4. Manabe, S.: The Non-Integer Integral and its Application to Control Systems. JIEE (Japanese Institute of Electrical Engineers) Journal **80** (1960) 589–597
5. Oldham, K.B., Spanier, J.: The Fractional Calculus. Academic Press, New York (1974)
6. Samko, S.G., Kilbas, A.A., Marichev, O.I.: Fractional Integrals and Derivatives and Some of Their Applications. Nauka i technika, Minsk (1987)
7. Miller, K.S., Ross, B.: An Introduction to the Fractional Calculus and Fractional Differential Equations. Wiley, New York (1993)

8. Podlubny, I.: Fractional Differential Equations. Volume 198. Academic Press (1999)
9. Lurie, B.J.: Three-Parameter Tunable Tilt-Integral-Derivative (TID) Controller. US Patent US5371670 (1994)
10. Podlubny, I.: Fractional-Order Systems and $\mathrm{PI}^{\lambda}\mathrm{D}^{\mu}$-Controllers. IEEE Trans. Automatic Control **44** (1999) 208–214
11. Oustaloup, A., Mathieu, B., Lanusse, P.: The CRONE Control of Resonant Plants: Application to a Flexible Transmission. European Journal of Control **1** (1995)
12. Oustaloup, A., Moreau, X., Nouillant, M.: The CRONE Suspension. Control Engineering Practice **4** (1996) 1101–1108
13. Raynaud, H., ZergaInoh, A.: State-Space Representation for Fractional Order Controllers. Automatica **36** (2000) 1017–1021
14. Vinagre, B.M., Petrás, I., Podlubny, I., Chen, Y.Q.: Using Fractional Order Adjustment Rules and Fractional Order Reference Model in Model-Reference Adaptative Control. Nonlinear Dynamics **29** (2002) 269–279
15. Manabe, S.: The non-Integer Integral and its Application to Control Systems. ETJ of Japan **6** (1961) 83–87
16. Oustaloup, A.: Fractional Order Sinusoidal Oscilators: Optimization and Their Use in Highly Linear FM Modulators. IEEE Transactions on Circuits and Systems **28** (1981) 1007–1009
17. Axtell, M., Bise, E.M.: Fractional calculus applications in control systems. In: Proceedings of the IEEE 1990 Nat. Aerospace and Electronics Conf., New York, USA (1990) 563–566
18. Vinagre, B.M., Chen, Y.: Lecture Notes on Fractional Calculus Applications in Automatic Control and Robotics. The 41st IEEE CDC2002 Tutorial Workshop #2, `http://mechatronics.ece.usu.edu /foc/cc02tw/cdrom/lectures/book.pdf`, or, `http://eii.unex.es/isa/LasVegas/` (2002)
19. Orsoni, B., Melchior, P., Oustaloup, A.: Davidson-Cole Transfer Function in Path Tracking Design. In: Proceedings of the 6th European Control Conference, Porto, Portugal (2001) 1174–1179
20. Orsoni, B., Melchior, P., Oustaloup, A., Badie, T., Robin, G.: Fractional motion control: Application to an xy cutting table. Nonlinear Dynamics **29** (2002) 297–314
21. Bahl, V.: Modeling and Control of a Class of Autonomous Wheeled Mobile Robots. Master's thesis, Department of Electrical and Computer Engineering, Utah State University, Logan, Utah (2002)
22. Brennan, S.: Modeling and control issues associeted with scaled vehicles. Master's thesis, University of Illinois (1999)
23. Vinagre, B.M., Podlubny, I., Hernandez, A., Feliu, V.: Some Approximations of Fractional Order Operators Used in Control Theory and Applications. Fractional Calculus and Applied Analysis **3** (2000) 231–248