

# Spatial Path Tracking of An Autonomous Industrial Vehicle Using Fractional Order Controllers

**J. I. Suárez**  
Univ. of Extremadura  
Badajoz - Spain  
jmarcelo@unex.es

**B. M. Vinagre**  
Univ. of Extremadura  
Badajoz - Spain  
bvinagre@unex.es

**Y. Q. Chen**  
CSOIS, Utah State University  
Logan, Utah - USA  
yqchen@ece.usu.edu

## Abstract

The aim of this paper is to present the use of Fractional Order Controllers (FOC) applied to the path-tracking problem in an autonomous electric vehicle. Conventional and Fractional Order Controllers have been simulated and compared. 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. To achieve a good path tracking performance, many 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 paper, a spatial path tracking method, called the  $\epsilon$ -controller, is applied which was first proposed in [3] where the proposed  $\epsilon$ -controller computes the normal distance from the vehicle to the desired path,  $\epsilon$ , and generates a desired velocity vector for the vehicle to follow the path.

To improve the performance of  $\epsilon$ -controller, several regulation schemes using the fractional-order control (FOC) idea [4] have been investigated in this paper. FOC is based on "fractional-order calculus". Recent books [5, 6, 7, 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, 10, 11, 12, 13]. For pioneering works, we cite [4, 14, 15, 16]. For the latest development of fractional calculus in automatic control and robotics, we cite [17]. As for path tracking problems, the first experiences in path-tracking applied to XY cutting tables and mobile robotics can be found in [18] and [19].

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  $\epsilon$ -controller on path-tracking basic algorithm.

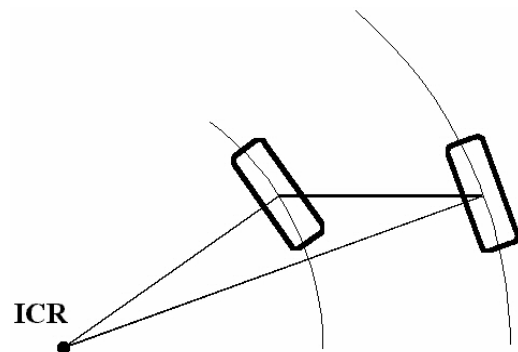


Figure 1: Vehicle with Ackerman system.

The rest of the paper is organized as follows: in Sec. 2 the lateral dynamic model of the vehicle is shortly described. In Sec. 3 the  $\epsilon$ -controller is briefly introduced. Section 4 presents several control schemes (P, PI, PI <sup>$\alpha$</sup> ,...) with a special attention to FOC. Section 5 shows some of our simulated results. Finally, in Sec. 6, some conclusions and guidelines for future investigations are outlined.

## 2 Vehicle Dynamic Model

The vehicle used in our work is a Citroën Berlingo with an Ackerman steering system [20]. This type of system allows the front wheels to travel different distances and describe their respective arcs when cornering (Fig.1). If the front wheels were turned the same angle, then the inner wheel would slide and the effectiveness of the steering would be reduced (Fig.2). With the sake of reduce this unwanted effect the inner wheel must turn a greater angle than the outer one. So, when cornering the both front wheels are perpendicular to the line that connects them with the instantaneous center of rotation (ICR).

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

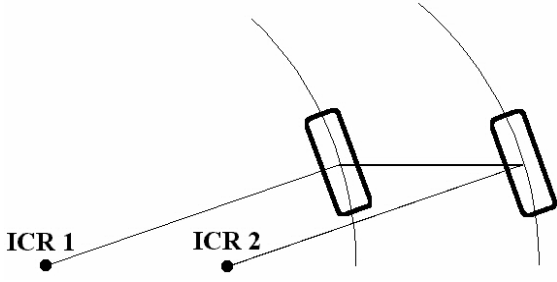


Figure 2: Vehicle without Ackerman system

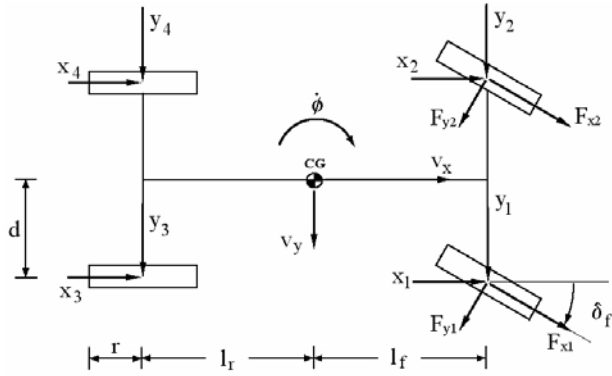


Figure 3: Ackerman steered vehicle and its related forces for the lateral dynamic model

can be obtained by solving the dynamic equations and further simplifications are done. For instance, the Dugof tyre model is used, which states that the force exerted by a tyre when cornering is proportional to the slip angle of the tyre [21]. This is only true if small steering angles are considered. Also the model is simplified by assuming 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, because it models a bicycle whose movements are constrained to in-plane motion.

Figure 3 shows a diagram where the main parameters and variables are depicted. They are the following ones:

- $v_x, v_y$ : longitudinal and lateral velocity, respectively.
- $\delta_f$ : 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.3. The longitudinal velocity  $v_x$  is supposed to be approximately constant. Based on the coordinate system frames shown in Fig.3, the dynamic equations can be obtained from the Newton's law by computing the forces and torques acting on the vehicle:

$$m(\dot{v}_y + v_x \dot{\phi}) = y_f + y_r \quad (1)$$

$$I_z \ddot{\phi} = l_f y_f - l_r y_r \quad (2)$$

where  $y_f = y_1 + y_2$  and  $y_r = y_3 + y_4$ .

Choosing  $\dot{\phi}$  and  $v_y$  as state variables the vehicle model can be expressed as the following state equations:

$$\begin{bmatrix} \dot{v}_y \\ \dot{\phi} \end{bmatrix} = \begin{bmatrix} -\frac{a_1}{mv_x} & -v_x - 2\frac{a_2}{mv_x} \\ -2\frac{a_3}{I_z v_x} & -2\frac{a_4}{I_z v_x} \end{bmatrix} \cdot \begin{bmatrix} v_y \\ \dot{\phi} \end{bmatrix} + \begin{bmatrix} b_1 \\ b_2 \end{bmatrix} \delta_f \quad (3)$$

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

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

### 3 Path-tracking Algorithm

The path-tracking problem was accomplished by using the scalar  $\epsilon$ -controller [3]. Basically, it is a regulator, with the cascade control architecture shown in Fig.4, that operates on the vehicle normal deviation  $\epsilon$  from the desired path. The  $\epsilon$ -controller ( $C_\epsilon$ ) 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  $\epsilon$ -controller 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 ( $\delta_w^*$ ) setpoints.

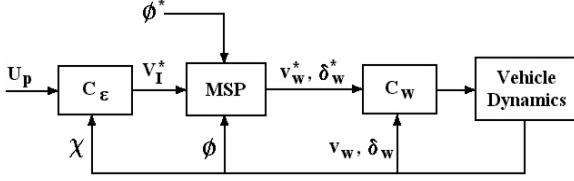


Figure 4: Cascade control architecture of the  $\epsilon$ -controller

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 [20].

#### 4 $C_\epsilon$ -controller regulation schemes

The control law of the  $\epsilon$ -controller turns the two-dimensional path tracking problem into a scalar regulation. It is a nonlinear controller operating on the scalar  $\epsilon$ , 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.

##### 4.1 P Controller

In this regulation method the control law is given by

$$u(t) = K_P \epsilon(t) \quad (4)$$

then, the control signal is proportional to the lateral distance from the path. 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:

$$G_P(z) = K_P \quad (5)$$

##### 4.2 PI and $PI^\alpha$ Controller

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

$$u(t) = K_P \epsilon(t) + K_I I^\alpha \epsilon(t) \quad (6)$$

being

$$I^\alpha f(t) \equiv \frac{1}{\Gamma(\alpha)} \int_0^t (t-\tau)^{\alpha-1} f(\tau) d\tau, \quad t > 0, \quad \alpha \in \mathbb{R}^+ \quad (7)$$

We have chosen  $\alpha = 1.25$  and  $\alpha = 1.5$  and hence we have obtained two fractional controllers. These values have been chosen to show that if we have an additional parameter, in this case  $\alpha$ , and the other parameter remains constant, we can obtain better results or, in the worse case, the same result. However, when  $\alpha = 1$ , (6) becomes

$$u(t) = K_P \epsilon(t) + K_I \int_0^t \epsilon(\tau) d\tau \quad (8)$$

which is the well known output of a 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:

$$G_{PI}(z) = K_P + K_I \frac{T(z+1)}{2(z-1)} \quad (9)$$

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  $\frac{1}{s^{1-\alpha}}$ . The first integrator has been discretized using the Tustin transformation  $s = \frac{2(z-1)}{T(z+1)}$ . The fractional order integrator  $\frac{1}{s^\alpha}$  has an infinite terms representation and must be discretized using a finite dimensional IIR filter. For our case (see [22]) 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. With  $\alpha = 1.5$  and  $\alpha = 1.25$  the approximations for the fractional part of the integrator are respectively:

$$\frac{1}{s^{0.5}} = \frac{7.155z^5 + 3.578z^4 - 7.155z^3 - 2.683z^2 + 1.342z + 0.2236}{32z^5 - 16z^4 - 32z^3 + 12z^2 + 6z - 1}$$

$$\frac{1}{s^{0.25}} = \frac{28.489z^5 + 7.122z^4 - 30.863z^3 - 5.490z^2 + 6.417z + 0.473}{60.235z^5 - 15.059z^4 - 65.255z^3 + 11.608z^2 + 13.568z - 1}$$

## 5 Simulations

### 5.1 Simulink Model

The high-level Simulink model shown in Fig. 5 simulates the control architecture depicted in Fig. 4. 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  $\epsilon$ -controller, in order to keep the vehicle on the desired path. The SMS is implemented together with the  $\epsilon$ -controller and the MSP algorithm

Table 1: Controllers Parameters

	$K_P$	$K_I$	$\alpha$
$P$	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

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  $30^\circ$ , and a low-level PI controller. The two next blocks are the dynamic lateral vehicle 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.

## 5.2 Controllers

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  $K_I$ .

## 5.3 Simulations Results

In the simulations the vehicle has been supposed to describe a semicircular path. First, the input vector to  $C_\epsilon$  block  $\mathbf{U}_P = [\chi_i \ \chi_f \ r \ V_d]^T$  shown in Fig.4 must be established, where  $\chi_i$  and  $\chi_f$  are the initial and final points, respectively,  $r$  is the radius of the path and  $V_d$  is desired vehicle velocity (Fig.6). 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  $\epsilon$ -controller will decrease the velocity and when it is on the path the vehicle will travel at its maximum speed (10m/s).

Figure 7 shows the responses of three tested controllers: P, PI and  $PI^{1.25}$  with  $K_P = 22$ . A positive value of  $\epsilon$  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 8 shows the behavior for the three fractional controllers. For classical PI controller we only can

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	<b>0.0436</b>

vary two parameters ( $K_P$  and  $K_I$ ). Note that in this case, for FOC an additional parameter ( $\alpha$ ) 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:

$$e = \sum_{i=1}^N \epsilon_i^2 \quad (10)$$

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).

## 6 Concluding Remarks

In this paper we have compared several controllers to show their performance in path-tracking problems. A novel and simple regulation scheme, fractional  $PI^\alpha$  controller, was simulated with good results. In future works we will look for sintonization methods for the fractional controllers with the aim of improving the performance by using such controllers by appropriate or optimal tuning of the parameters  $K_P$ ,  $K_I$  and  $\alpha$ . Also, we will test the controllers in the Citroën Berlingo vehicle.

## Acknowledgments

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

## References

- [1] A. Ollero, *Robotica. Manipuladores y Robots Mviles*. Marcombo, 2001.
- [2] J. S. Wit, *Vector Pursuit Path Tracking for Autonomous Ground Vehicles*. PhD thesis, University of Florida, 2000.
- [3] M. Davidson and V. Bahl, "The Scalar  $\epsilon$ -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*, pp. 175–180, May 21–26 2001.

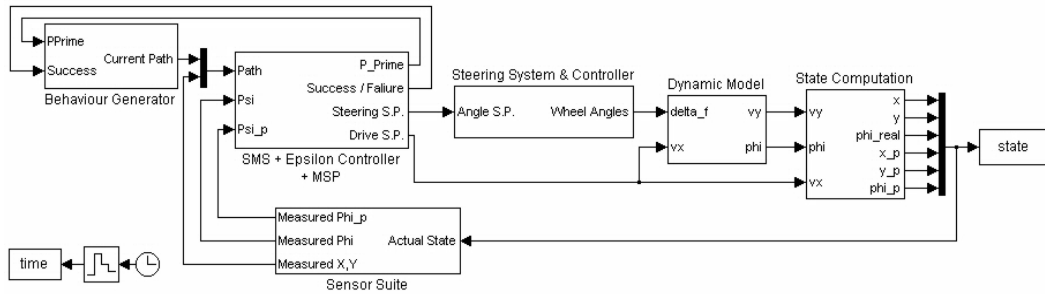


Figure 5: Simulink model of the  $\epsilon$ -controller used in the simulations

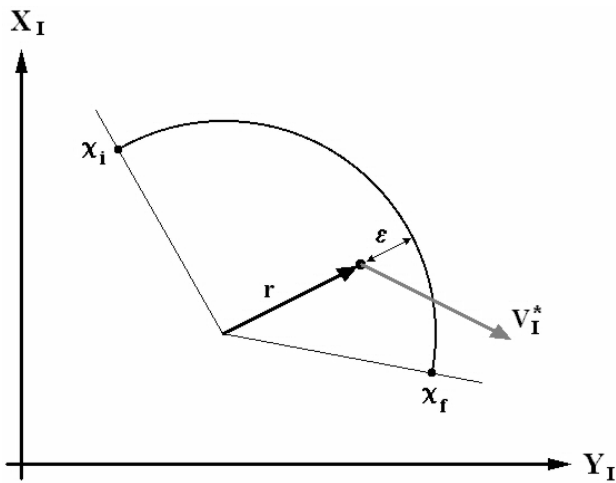


Figure 6:  $\epsilon$ -Controller parameters for an arc segment.

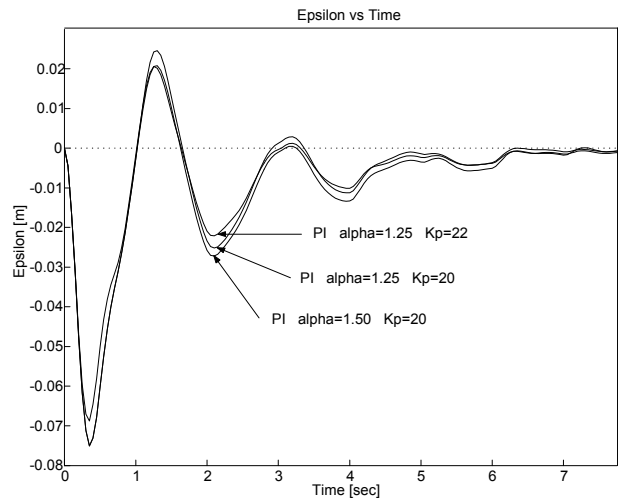


Figure 8: Deviation  $\epsilon$  along the path for different FOC.

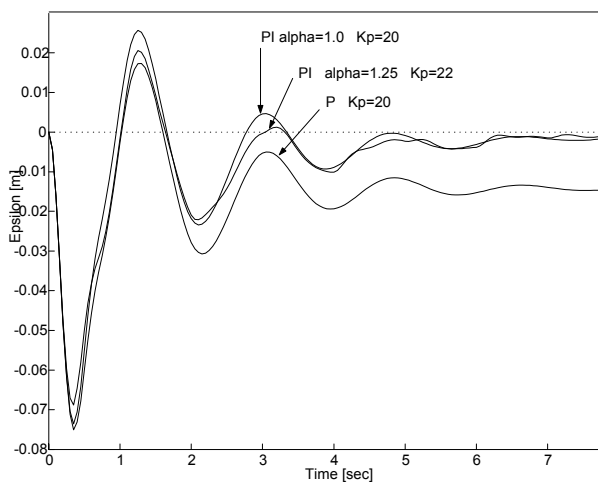


Figure 7: Deviation  $\epsilon$  along the path for P, PI and  $PI^{1.25}$  controllers

- [4] S. Manabe, "The Non-Integer Integral and its Application to Control Systems," *JIEE (Japanese Institute of Electrical Engineers) Journal*, vol. 80, no. 860, pp. 589–597, 1960.
- [5] K. B. Oldham and J. Spanier, *The Fractional Calculus*. New York: Academic Press, 1974.
- [6] S. G. Samko, A. A. Kilbas, and O. I. Marichev, *Fractional Integrals and Derivatives and Some of Their Applications*. Minsk: Nauka i technika, 1987.
- [7] K. S. Miller and B. Ross, *An Introduction to the Fractional Calculus and Fractional Differential Equations*. New York: Wiley, 1993.
- [8] I. Podlubny, "Fractional-Order Systems and Fractional-Order Controllers," Tech. Rep. UEF-03-94, Slovak Academy of Sciences. Institute of Experimental Physics, Department of Control

- Engineering. Faculty of Mining, University of Technology. Kosice, November 1994.
- [9] B. J. Lurie, “Three-Parameter Tunable Tilt-Integral-Derivative (TID) Controller,” *US Patent US5371670*, 1994.
- [10] I. Podlubny, “Fractional-Order Systems and  $PI^{\lambda}D^{\mu}$ -Controllers,” *IEEE Trans. Automatic Control*, vol. 44, no. 1, pp. 208–214, 1999.
- [11] A. Oustaloup, B. Mathieu, and P. Lanusse, “The CRONE Control of Resonant Plants: Application to a Flexible Transmission,” *European Journal of Control*, vol. 1, no. 2, 1995.
- [12] A. Oustaloup, X. Moreau, and M. Nouillant, “The CRONE Suspension,” *Control Engineering Practice*, vol. 4, no. 8, pp. 1101–1108, 1996.
- [13] H. Raynaud and A. ZergaInoh, “State-Space Representation for Fractional Order Controllers,” *Automatica*, vol. 36, pp. 1017–1021, 2000.
- [14] S. Manabe, “The non-Integer Integral and its Application to Control Systems,” *ETJ of Japan*, vol. 6, no. 3-4, pp. 83–87, 1961.
- [15] A. Oustaloup, “Fractional Order Sinusoidal Oscillators: Optimization and Their Use in Highly Linear FM Modulators,” *IEEE Transactions on Circuits and Systems*, vol. 28, no. 10, pp. 1007–1009, 1981.
- [16] M. Axtell and E. M. Bise, “Fractional calculus applications in control systems,” in *Proceedings of the IEEE 1990 Nat. Aerospace and Electronics Conf.*, (New York, USA), pp. 563–566, 1990.
- [17] B. M. Vinagre and Y. Chen, *Lecture Note on Fractional Calculus Applications in Automatic Control and Robotics*.  
<http://mechatronics.ece.usu.edu/foc/cc02tw/cdrom/lectures/book.pdf>, or,  
<http://eii.unex.es/isa/LasVegas/>: The 41st IEEE CDC2002 Tutorial Workshop # 2, December 2002.
- [18] B. Orsoni, P. Melchior, and A. Oustaloup, “Davidson-Cole Transfer Function in Path Tracking Design,” in *Proceedings of the 6th European Control Conference*, (Porto, Portugal), pp. 1174–1179, July 2001.
- [19] B. Orsoni, P. Melchior, A. Oustaloup, T. Badie, and G. Robin, “Fractional motion control: Application to an xy cutting table,” *Nonlinear Dynamics*, vol. 29, no. 1-4, pp. 297–314, 2002.
- [20] V. Bahl, “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.
- [21] S. Brennan, “Modeling and control issues associated with scaled vehicles,” Master’s thesis, University of Illinois, 1999.
- [22] B. M. Vinagre, I. Podlubny, A. Hernandez, and V. Feliu, “Some Approximations of Fractional Order Operators Used in Control Theory and Applications,” *Fractional Calculus and Applied Analysis*, vol. 3, no. 3, pp. 231–248, 2000.
- [23] S. Brennan and A. Alleyne, “Using a Scale Testbed: Controller Design and Evaluation,” *IEEE Control Systems Magazine*, vol. 21, no. 3, pp. 15–26, 2001.