UDC 531.3

Trajectory Tracking Control of Programmed Motion in Second Order Nonholonomic Systems

C. T. Deressa

Department of Theoretical Physics and Mechanics
Peoples' Friendship University of Russia
6, Miklukho-Maklaya str., Moscow, 117198, Russia

The D'Alembert–Lagrange principle in general stands for all ideal holonomic and nonholonomic constraints of arbitrary order. But in practice the application of the principle is restricted to ideal holonomic and linear first order nonholonomic constraints. In recent years the direct application of this famous principle is made to model dynamic equation of acceleration level constrained systems. This paper uses the dynamic equation developed to establish a theoretical framework for trajectory tracking control of programmed motion with acceleration level constraints. The concept of dividing constraints based on their sources into natural and programmed constraints is employed. The trajectory tracking control is accomplished by two models called Reference Control Model constructed using both the programmed and natural constraints and a Dynamic Control Model developed by considering the natural constraints only. The Reference control model is used to plan the required trajectory based on a given acceleration or lower level programmed constraint. The Dynamic Control Model is utilized to control and stabilize the trajectory tracking process. Finally, to verify the effectiveness of the framework developed in the paper, a practical example is provided and simulation results are depicted.

Key words and phrases: programmed constraint, natural constraint, programmed motion, reference control model, dynamic control model, trajectory tracking, stability.

1. Introduction

The discovery of nonholonomic systems was made by Euler while studying rolling of rigid bodies, whereas the term *nonholonomic system* was coined by Hertz in 1894 [1–3]. Hertz was the first person to make clear distinction between holonomic and nonholonomic systems.

The classification of constraints into holonomic and nonholonomic doesn't include all the constraints in real world. Moreover many dynamic equations uses holonomic and first order linear nonholonomic systems with the exception of Appell equation that can be applied to systems with second order constraints [4].

In response to the above paragraph, recently [4,5] a dynamic division of constraints based on their sources is made. Indeed, if conditions are imposed by nature or environment then the constraints are called Natural Constraints. *Natural constraints* are based on the assumption that, nonholonomic constraints arise when two or more bodies are in contact with each other and roll without slipping. A constraint may be imposed on velocities, accelerations or any other feature of the system such as performance and design. These restrictions are said to be *Programmed Constraints.* Programmed constraints, like Natural constraints, can have forms including higher derivatives of the coordinates. Imposing tasks to be performed by a dynamic system are examples of Programmed constraint [4, 6].

Robots for instance, are designed to perform [4] many requirements that may be described by programmed constraints. Programmed constraints that can be specified by algebraic or higher order differential equations can be put as a task to be performed by them. This is why studying programmed constraints of higher orders are of interest.

Motion tracking includes tracking of a planned motion described by algebraic or differential equation of constraints. In nonlinear control theory, motion tracking is the same [1] as trajectory tracking. There are two types [1] of models for accomplishment of Trajectory tracking in nonlinear control: a *kinematic model* in which the control input is velocity of the system and the *dynamic model* of the system in which the control

Received 26^{th} June, 2014.

inputs can be forces and torques. Kinematic models highly exploited for trajectory tracking control of programmed constraints of lower levels. In this paper we focus on the use of dynamic model for trajectory tracking control of programmed motion in higher order constraints.

Trajectory tracking of first order nonholonomic systems is achieved using dynamic models in a reduced state form [7]. Control objective other than trajectory tracking of nonholonomic systems of first order constraints can't be achieved using only dynamic models in reduced state forms [2].

Motion tracking includes tracking of a planned motion described by algebraic or differential equation of constraints. In nonlinear control theory, motion tracking is the same [6] as trajectory tracking. There are two types of models for accomplishment of Trajectory tracking in nonlinear control: a kinematic model in which the control input is velocity of the system and the dynamic model of the system in which the control inputs can be forces and torques.

Trajectory tracking of first order nonholonomic systems is achieved using dynamic models in a reduced state form [8]. Control objective other than trajectory tracking of nonholonomic systems of first order constraints can't be achieved using only dynamic models in reduced state forms [2, 6].

Trajectory Tracking Control of Programmed Motion in Higher Order Nonholonomic Systems is a strategy for tracking control of programmed motion given by equations of second order constraints is established in this paper.

The strategy uses a Reference Control model and a Dynamic Control model of a system. A Reference Control model is developed based on both Natural and programmed constraints and is used for generating a dynamically possible trajectory of a given programmed constraint. A Dynamic Control model is used for stabilization and selection of an appropriate control input torque for the purpose of tracking the planned trajectory obtained from the Reference Control model.

2. Dynamic Modeling of First and Second Order Nonholonomic Systems

The state of [7, 9, 10] representative point $(q_j, \dot{q}_j)$ of a system with generalized coordinates $\mathbf{q} = (q_1, q_2, \ldots, q_n)$, Lagrangian L is given by the solution of:

$$\frac{\mathrm{d}}{\mathrm{d}t}\frac{\partial L}{\partial \dot{q}_j} - \frac{\partial L}{\partial q_j} = Q_j^{ex} + Q_j^c, \quad j = 1, 2, 3, \ldots, n, \tag{1}$$

where Q_j^{ex} is an external force, Q_j^c is the unknown force which constrain the system. In case Q_j^c is a constraint force of ideal constraint, then (1) reduces to the form:

$$\left(\frac{\mathrm{d}}{\mathrm{d}t}\frac{\partial L}{\partial \dot{q}_j} - \frac{\partial L}{\partial q_j} - Q_j^{ex}\right)\delta q_j = Q_j^c \delta q_j = 0, \quad j = 1, 2, 3, \ldots, n. \tag{2}$$

Equation (2) is the famous d'Alembert–Lagrange principle, a fundamental principle of Analytical dynamics developed by Lagrange.

Although the principle is meant for all ideal holonomic and nonholonomic constraints, it is widely applied to linear first order velocity constraints for a long time. But recently, application of d'Alembert–Lagrange to general nonholonomic systems of higher orders is done by M. R. Flannery in 2011 [10]. The dynamic equations and transposition relations between $\delta(\dot{q}_j)$ and $\frac{\mathrm{d}}{\mathrm{d}t}\left(\frac{\delta q_j}{\mathrm{d}t}\right)$ for velocity constraints and between $\delta\ddot{q}_j$ and $\frac{\mathrm{d}}{\mathrm{d}t}(\delta\dot{q}_j)$ for acceleration constraints are established based on d'Alembert–Lagrange principle (2). In this section the main results are revised without their proofs. For further details refer to [10].

1. For constraints of the form:

$$\Phi_k(\mathbf{q}, \dot{\mathbf{q}}, t) = 0. \tag{3}$$

   (a) The virtual work done by the constraints are given as:

$$\frac{\partial \Phi_k}{\partial \dot{q}_j}\delta q_j = 0. \tag{4}$$

   (b) The dynamic equation is proved to be:

$$\frac{\mathrm{d}}{\mathrm{d}t}\frac{\partial L}{\partial \dot{q}_j} - \frac{\partial L}{\partial q_j} = Q^{ex} + \lambda_k \frac{\partial \Phi}{\partial \dot{q}_j}. \tag{5}$$

   (c) The transposition relation is given by:

$$\delta \Phi_k - \frac{\mathrm{d}}{\mathrm{d}t}\left[\frac{\partial \Phi}{\partial \dot{q}_j}\delta q_j\right] = 0. \tag{6}$$

Indeed, from constraints of the form (3) we obtain:

$$\dot{\Phi}_k = \frac{\partial \Phi_k}{\partial \dot{q}_j}\ddot{q}_j + \frac{\partial \Phi_k}{\partial q_j}\dot{q}_j + \frac{\partial \Phi_k}{\partial t} = 0 \tag{7}$$

This directly leads to:

$$\delta \Phi_k = \frac{\partial \Phi_k}{\partial \dot{q}_j}\delta \dot{q}_j + \frac{\partial \Phi_k}{\partial q_j}\delta q_j = 0 \tag{8}$$

Denote the m-independent and the c-dependent coordinates in$\{q_j\}$ by $q_i, i \leq m$ and p_s respectively.Then (7)(6a) decomposes into:

$$\dot{\Phi}_k = M_{ks}\ddot{p}_s + \left[\frac{\partial \Phi_k}{\partial \dot{q}_i}\ddot{q}_i + \frac{\partial \Phi_k}{\partial q_j}\dot{q}_j + \frac{\partial \Phi_k}{\partial t} = 0\right], \tag{9}$$

where $M_{ks} = M_{ks}(q, \dot{q}, p_s, \dot{p}_s, t) = \frac{\partial \Phi_k}{\partial \dot{p}_s}$ are elements of matrix$M = \{M_{ks}\}$ which is assumed to be positive definite and $q_j = \{q_i, p_s\}$. The solution for the dependent acceleration in (9) is given by:

$$\ddot{p}_s = -\overline{M}_{sr}\left[\frac{\partial \Phi_k}{\partial \dot{q}_i}\ddot{q}_i + \frac{\partial \Phi_k}{\partial q_j}\dot{q}_j + \frac{\partial \Phi_k}{\partial t}\right], \tag{10}$$

where matrices $\overline{M}_{sr}$ and M_{ks} are inverses of each other. For the dependent displacement we have:

$$\delta p_s = \left(\frac{\partial p_s}{\partial q_i}\right)\delta q_i = \left(\frac{\partial \dot{p}_s}{\partial \dot{q}_j}\right)\delta q_i = \left(\frac{\partial \ddot{p}_s}{\partial \ddot{q}_i}\right)\delta q_i. \tag{11}$$

This leads to(based on equation (9))

$$\delta p_s = -\overline{M}_{sr}\left(\frac{\partial \Phi_k}{\partial \dot{q}_i}\right)\delta q_i \tag{12}$$

After multiplying (12) by M_{ks} we obtain:

$$\frac{\partial \Phi_k}{\partial \dot{q}_j}\delta q_j = \frac{\partial \Phi_k}{\partial \dot{p}_s}\delta p_s + \frac{\partial \Phi_k}{\partial \dot{q}_i}\delta q_i = 0 \tag{13}$$

Since (7) and (12) are each zero the quantity

$$\delta \Phi_k - \frac{\mathrm{d}}{\mathrm{d}t}\left[\frac{\partial \Phi_k}{\partial \dot{q}_i}\delta q_j\right] = 0 \tag{6g}$$

provides a transpositional relation given by equation (6) above.

2. For acceleration level constraints of the form:

$$\Psi_k(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}, t) = 0, \quad k = 1, 2, \ldots, d. \tag{14}$$

(a) The virtual work done by the constraint force is given by:

$$\frac{\partial \Psi_k}{\partial \ddot{q}_j}\delta q_j = 0. \tag{15}$$

(b) The dynamic equation is proved to be:

$$\frac{\mathrm{d}}{\mathrm{d}t}\frac{\partial L}{\partial \dot{q}_j} - \frac{\partial L}{\partial q_j} = Q^{ex} + \lambda_k \frac{\partial \Psi}{\partial \ddot{q}_j}, \tag{16}$$

where $\lambda = \lambda_k$ is Lagrangian multiplier.

(c) The transposition relation which can be found in the same way done for equation (6) is given by:

$$\delta \Psi_k - \frac{\mathrm{d}^2}{\mathrm{d}t^2}\left[\frac{\partial \Psi_k}{\partial \ddot{q}_j}\delta q_j\right] = 0. \tag{17}$$

3. Control Models for Trajectory Tracking of Programmed Motion

In this section control models that are used for trajectory tracking of programmed motion are constructed. The control models are developed based on the concept of the null space and constrained dynamic model for higher level constraints discussed in section 2. Particularly, we focus on acceleration level constraints of the form (14) since it includes first order constraints of the form (3).

In this paper a **programmed constraint** is a non-material constraint and any requirement put on a physical system motion specified by an algebraic or differential equation of any order. **A programmed motion** is a system motion that satisfies a programmed constraint. **A natural constraint** is the usual holonomic and non-holonomic constraints that are not programmed.

Remark 1. a) Programmed constraints can also be holonomic or nonholonomic having the form as the classical holonomic and nonholonomic constraints.

b) In this paper constraints of the form (14) are assumed to include both programmed and natural constraints.

The constrained dynamic model (16) can be written in the form:

$$\begin{cases} M(\mathbf{q})\ddot{\mathbf{q}} + C(\mathbf{q}, \dot{\mathbf{q}}) + D(\mathbf{q}) = J^T(\mathbf{q}, \dot{\mathbf{q}})\lambda + Q^{ex}\,, \\ \psi_k(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}, t) = 0, \quad (k = 1, 2, \ldots, d). \end{cases} \tag{18}$$

Where $M(q)$ is an $(n \times n)$ positive definite symmetric matrix, λ is a d-dimensional vector of Lagrange's multipliers, $J^T(\mathbf{q}, \dot{\mathbf{q}}) = \frac{\partial \Psi_k}{\partial \ddot{q}_j}$ is a full rank of size $(d \times n)$ matrix, $C(\mathbf{q}, \dot{\mathbf{q}})$ is an $n \times 1$ matrix containing vectors of centripetal and Coriolis forces, $D(\mathbf{q})$ is an n-vector of gravitational force and Q^{ex} is an external force.

For trajectory tracking control of programmed motion, the dynamic model needs to be transformed to the reduced-state form. The Null space concept is used to eliminate the Lagrange multiplier from (18).

Equation (15) can be written in the form:

$$A(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} = 0, \tag{19}$$

where $A = \frac{\partial \Psi_k}{\partial \ddot{q}_j}$, and $\mathbf{q} = (q_1, q_2, \ldots, q_n)$. Let S be an $n \times (n-d)$ full rank matrix made from the basis vectors of the null space of A in (19) such that:

$$S(\mathbf{q}) = [g_1(\mathbf{q}), g_2(\mathbf{q}), \ldots, g_{n-d}(\mathbf{q})].$$

Where $g_1(\mathbf{q}), g_2(\mathbf{q}), ..., g_{n-d}(\mathbf{q})$ are column basis vectors of the null space of A.

Then there exists velocity vector $\mathbf{v}(t) = [v_1, v_2, ., ., .v_{n-d}]^T$ such that:

$$\dot{\mathbf{q}} = S(\mathbf{q})\mathbf{v}(t). \tag{20}$$

Substituting $\ddot{q}$ in the first equation of (18) and multiplying it by S^T we obtain:

$$\overline{M}\dot{\mathbf{v}}(t) + \overline{F}(\mathbf{q}, \dot{\mathbf{q}}) + \overline{D}(\mathbf{q}) = \overline{E}, \tag{21}$$

where $\overline{M} = S^T(\mathbf{q})M(\mathbf{q})S(\mathbf{q}), \overline{F}(\mathbf{q}, \dot{\mathbf{q}}) = S^T(\mathbf{q})[M(\mathbf{q})\dot{S}(\mathbf{q})\mathbf{v}(t) + C(\mathbf{q}, \dot{\mathbf{q}})], \overline{E} = S^T(\mathbf{q})Q^{ex}$ and $\overline{D}(\mathbf{q}) = S^T(\mathbf{q})D(\mathbf{q})$.

The constrained dynamic equation (16) is now transformed to **Reference Control Model for programmed motion** given by:

$$\begin{cases} \overline{M}\dot{\mathbf{v}}(t) + \overline{F}(\mathbf{q}, \dot{\mathbf{q}}) + \overline{D}(\mathbf{q}) = \overline{E}, \\ \Psi_k(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}, t) = 0. \end{cases} \tag{22}$$

Note that, $\Psi_k(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}, t) = 0$, $(k = 1, \ldots, d)$ in (22) includes both natural and programmed constraints.

The advantage of (22) is that, it doesn't include the constraint force and hence is convenient for tracking control. Equation (22) is said to be Reference Control Model. This is because it is used for planning a dynamically possible trajectory of a given dynamic system based on a given programmed constraint. An example is provided to make the concepts of reference control model clear.

Example 1. Consider a Differential Derive Mobile Robot (DDMR) shown in Fig. 1.

The mobile base is located with respect to the fixed reference frame denoted by $\{X_r, Y_r\}$ and by the body fixed frame at A denoted by $\{x_r, y_r\}$. The origin of the Robot fixed frame is defined to be the mid-point A on the axis between the wheels. The center of mass of the DDMR is located at a distance of $d \geqslant 0$ units form A on the axis of symmetry of the Robot. Let us fix the kinematic parameters and notation used to describe the mobile base in Table 1.

The natural constraints of the system are given by:

$$\begin{cases} -\dot{x}_a \sin\vartheta + \dot{y}_a \cos\theta = 0\,, \\ \dot{x}_a \cos\vartheta + \dot{y}_a \sin\vartheta + \dot{\vartheta}L = R\dot{\varphi}_R\,, \\ \dot{x}_a \cos\vartheta + \dot{y}_a \sin\vartheta - \dot{\vartheta}L = R\dot{\varphi}_R\,. \end{cases} \tag{23}$$

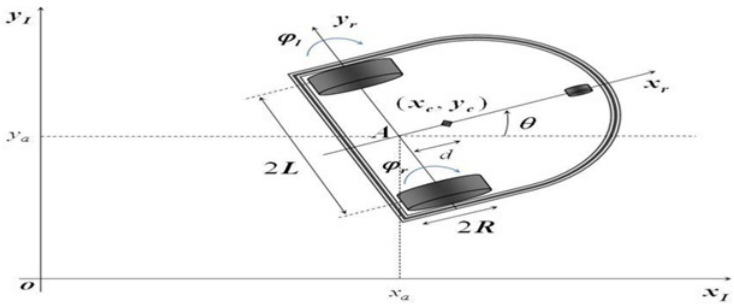

Figure 1. Differential Derive Mobile Robot

Table 1

Parameters and Notation

Parameters	Description
L	Distance from a wheel to A
d	Distance from wheel axis to center of mass
ϑ	Absolute rotation angle of the DDMR
(x_c, y_c)	Absolute position of center of mass
(x_a, y_a)	Absolute coordinates of A
φ_L, φ_R	Angular positions of the left and right driving wheels respectively
R	Wheel radius

Let us add a programmed constraint to the DDMR that we require it to move along a plane curve whose curvature is 5. That is:

$$\dot{x}_a \ddot{y}_a - \dot{y}_a \ddot{x}_a - 5[\dot{x}_a^2 + \dot{y}_a^2]^{3/2} = 0. \tag{24}$$

Now the total constraints of the DDMR are both (23) and (24). Observe that the programmed constraint has degree 2 and hence we have a higher order nonholonomic system. The purpose of this example is to apply (22) in obtaining the required trajectory of the DDMR, given the programmed constraint (24).

Using equation (22), the total constraints of the system given by (23) and (24), the null space concept we used to develop equation (22) and the dynamic equation of DDMR we obtain:

$$\dot{y}_a \ddot{y}_a + \dot{x}_a \ddot{x}_a = 0. \tag{25}$$

We need to write equation (25) in terms of linear velocity u and angular velocity ω of the DDMR given by $\dot{x}_a = u \cos\vartheta$ and $\dot{y}_a = u \sin\vartheta$. Substituting these values into (25) we obtain: $u = b$ where b is a non-zero constant. As a result $\dot{x}_a = b\cos\vartheta$ and $\dot{y}_a = b \sin\vartheta$. Moreover, these new results has to satisfy the programmed constraint and this leads to $\vartheta = 5bt$. Note that we have assumed $\vartheta(0) = x_a(0) = y_a(0) = 0$. Finally the required trajectory(dynamically possible trajectory), denoted by $\mathbf{x}_f = (x_f, y_f, \vartheta)^T$, is given by:

$$x_f = \frac{1}{5b}\sin(5bt), \quad y_f = \frac{-1}{5b}\cos(5bt), \quad \vartheta = 5bt. \tag{26}$$

The **Dynamic Control Model** is developed by considering only the natural constraints (without the programmed constraint) and has the following from:

$$\begin{cases} M(\mathbf{q})\ddot{\mathbf{q}} + C(\mathbf{q}, \dot{\mathbf{q}}) + D(\mathbf{q}) = B^T(\mathbf{q}, \dot{\mathbf{q}})\lambda + Q^{ex}, \\ B\dot{\mathbf{q}} = 0. \end{cases} \tag{27}$$

Where B is the usual Jacobian matrix obtained from the natural constraints, given by $B = \frac{\partial \Psi_k}{\partial \dot{\mathbf{q}}}$.

Using the concept of Null space (27) can be written in the form:

$$\begin{cases} \widehat{M}\dot{v} + \widehat{C}v = \widehat{E}\tau, \\ B\dot{\mathbf{q}} = 0, \end{cases} \tag{28}$$

where τ is the control input torque.

For the purpose of trajectory tracking control of programmed motion, we use both equations (22) and (28). The planned trajectory is obtained from (22) based on the given programmed constraint and the tracking control is performed by (28).

Moreover, for stabilization purpose we use the following method which is obtained by improving Baugarte's method of constraint stabilization.

A control input torque may be defined [6] as:

$$\tau = \tau(\widehat{M}(\mathbf{q}), \widehat{C}(\mathbf{q}, \dot{\mathbf{q}}), \widehat{N}(\mathbf{q}), \mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}, \mathbf{x}_f, \dot{\mathbf{x}}_f, \ddot{\mathbf{x}}_f).$$

Where $\widehat{N}(\mathbf{q})$ includes gravity terms. To cancel all nonlinearities and apply exactly the torque needed to overcome the inertia of the actuator the input torque can be defined as:

$$\widehat{M}(\mathbf{q})\ddot{\mathbf{x}}_f + \widehat{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{x}}_f + \widehat{N}(\mathbf{q}) = \tau. \tag{29}$$

Substituting this control law into the first equation of (28), we obtain:

$$\widehat{M}(\mathbf{q})\ddot{x}_f = \widehat{M}(\mathbf{q})\dot{v}. \tag{30}$$

Since $M(\mathbf{q})$ is positive definite in $\mathbf{q}$, we have:

$$\ddot{\mathbf{x}}_f = \ddot{\mathbf{x}},$$

where

$$\dot{v} = \ddot{\mathbf{x}}.$$

Hence, if the initial position and velocity of the DDMR matches the desired position and velocity, the DDMR will follow the desired trajectory. But obviously this control law will not correct for any initial condition errors which are present. This works in fever of modification to a method that may correct any initial condition error. This can be achieved by replacing $\ddot{\mathbf{x}}$ with:

$$\ddot{\mathbf{x}} = \ddot{\mathbf{x}}_f - \mathbf{K}_D\dot{\mathbf{e}} - \mathbf{K}_P\mathbf{e}.$$

Where, $\mathbf{e} = \mathbf{x} - \mathbf{x}_f$, and the corresponding control law becomes:

$$\widehat{M}(\mathbf{q})[\ddot{\mathbf{x}}_f - K_D\dot{\mathbf{e}} - K_p\mathbf{e}] + \widehat{C}(\mathbf{q}, \dot{\mathbf{q}})v + \widehat{N}(\mathbf{q}) = \tau, \tag{31}$$

where, $\mathbf{e} = \mathbf{x} - \mathbf{x}_f$, K_D, K_P are $(n-m) \times (n-m)$ constant positive definite gain matrices. Substitute (31) into equation (28), we obtain asymptotically stable error dynamics:

$$\ddot{\mathbf{e}} + K_D\dot{\mathbf{e}} + K_P\mathbf{e} = 0. \tag{32}$$

Where K_D and K_P are constant, positive definite and symmetric matrices. Equation (32) is a linear differential equation which governs the error between the **actual and planned (desired) trajectories**.

Equation (32) can be written in a state space form in terms of $[\Theta^T, \dot{\Theta}^T]$ as:

$$\frac{\mathrm{d}}{\mathrm{d}t}\begin{pmatrix}\Theta\\ \dot{\Theta}\end{pmatrix} = \begin{pmatrix}\dot{\Theta}\\ -K_D\dot{\Theta} - K_P\Theta\end{pmatrix} = \begin{pmatrix}0 & I\\ -K_P & -K_D\end{pmatrix}\begin{pmatrix}\Theta\\ \dot{\Theta}\end{pmatrix}, \tag{33}$$

where, I is the identity matrix of size m.

Let us investigate the stability of the origin in (33): The immediate Lyapunaov function candidate is:

$$\begin{aligned} V(\Theta,\dot{\Theta}) &= \frac{1}{2}\begin{pmatrix}\Theta\\ \dot{\Theta}\end{pmatrix}^T \begin{pmatrix}K_p + \epsilon K_D & \epsilon I\\ \epsilon I & I\end{pmatrix}\begin{pmatrix}\Theta\\ \dot{\Theta}\end{pmatrix} = \\ &= \frac{1}{2}\left(\dot{\Theta} + \epsilon\Theta\right)^T\left(\dot{\Theta} + \epsilon\Theta\right) + \frac{1}{2}\Theta^T[K_P + \epsilon K_D - \epsilon^2 I]\Theta. \end{aligned} \tag{34}$$

Where the constant ϵ satisfies: $K_D - \epsilon I > 0$, $K_P + \epsilon K_D - \epsilon^2 I > 0$. Evaluating the total time derivative of $V(\Theta,\dot{\Theta})$ we obtain:

$$\dot{V}(\Theta,\dot{\Theta}) = -\begin{bmatrix}\Theta\\ \dot{\Theta}\end{bmatrix}\begin{bmatrix}\epsilon K_P & 0\\ 0 & K_D - \epsilon I\end{bmatrix}\begin{bmatrix}\Theta\\ \dot{\Theta}\end{bmatrix}. \tag{35}$$

Equation (35) is globally negative definite and as a result, we conclude that $(\Theta,\dot{\Theta}) = (0,0)$ is globally asymptotically stable by Lyapunov's direct method of stabilization.

In summary we develop an algorithm for a trajectory tracking control of a programmed motion.

1. Obtain a dynamically possible trajectory x_f from the reference control model (22).
2. Obtain the first torque from equation (29).
3. Obtain actual trajectory x using the torque obtained in (2) and equation (28).
4. Observe the error by comparing the result from (3) and (1).
5. Keep on improving the error by obtaining an improved torque from (31) using different values for entries of the gain matrices K_D and K_P.
6. Substitute the torque obtained from (5) in (28) to get new actual trajectories.
7. Go to step 4.
8. Repeat these steps until you get sufficiently good input torque so that $|x - x_f| \leqslant \epsilon$ for a small positive number ϵ.

Example 2. This example is a continuation of Example 1. The purpose of this example is to find a torque that constrains the motion of the dynamics towards the required trajectory $\mathbf{x}_f$ (trajectory tracking). The result is described by simulation on MATLAB 2012a.

Substituting $b = \frac{1}{10}$ in equation (26) the required trajectory becomes:

$$x_f = 2\sin(0.5t), \quad y_f = -2\cos(0.5t), \quad \vartheta_f = 0.5t.$$

Simulation I. The algorithm developed above is used for simulation.

K_D and K_p are taken diagonal matrices. Let us start with $K_d = K_p = 0$. The new trajectory becomes: $x = \sin(0.5t)$, $y = -\cos(0.5t)$, $\vartheta = 0.5t$.

The simulation for the absolute value of the error in simulation I is shown in the Fig. 2. Error in x is denoted by $e(x)$ and error in y is denoted by $e(y)$. The error in ϑ is zero and not included in the figures.

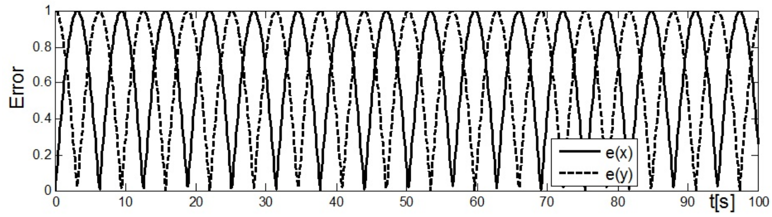

Figure 2. Simulation I graph

The portraits of actual trajectory and the required trajectory in simulation I, are shown in Fig. 3.

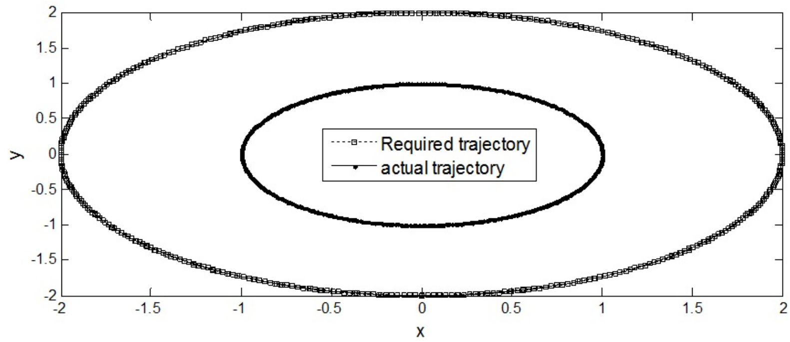

Figure 3. Simulation I portrait

Simulation II. After several experimentation on MATLAB, the torque obtained with small diagonal entries of the gain matrices $k_{d1} = 0.0001$, $k_{d2} = 0.0001$, $k_{p1} = 0.0001$, $k_{p2} = 0.0001$, $k_{d3} = 0.0001$, $k_{p3} = 0.0001$ seem to give a good result on tracking the programmed motion. In Fig. 4 the Error simulation is displayed.

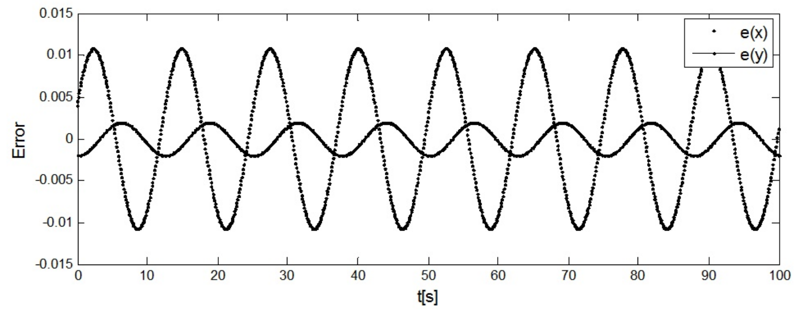

Figure 4. Error graph for simulation II

In simulation II, the obtained actual trajectory is given by:

$$x = 1.99\sin(0.5t) - 0.004\cos(0.5t), \quad y = -0.008\cos(0.5.t) - 1.99\cos(0.5t).$$

The portrait of the actual and the required curves are shown in Fig. 5.

Remark 2. In the simulation of the above demonstration it was observed that when the entries of the gain matrices increase (greater than 1) the trajectory tracking becomes highly violated. Taking the values smaller and smaller guarantees the asymptotic stability of the tracking. The values can be taken to be equal or different from each other. This simulation experiment is performed using symbolic maths

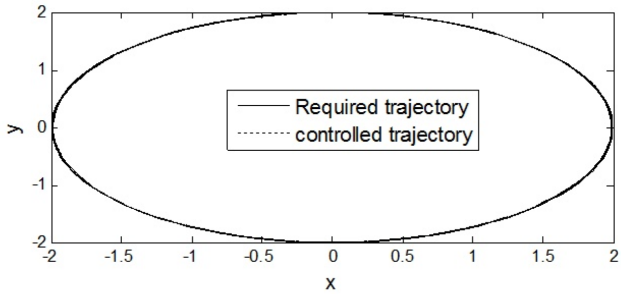

Figure 5. Simulation II portrait

in MATLAB. The expressions for the torque, for instance are too long and it may be too expensive to do it by paper and pencil work.

4. Conclusion

In this article trajectory tracking control of programmed acceleration level constraints are detailed. The underlying structure of the tracking control includes reference control model and dynamic control model used for planning and controlling the tracking process respectively. Although the framework developed in this paper is discussed in terms of higher order constraints, it can effectively be used for trajectory tracking control of the usual holonomic and first order linear nonholonomic constraints.

References

1. A. M. Bloch, J. E. Marsdeny, D. V. Zenkovz, Notices of the American Mathematical Society, Nonholonomic Dynamics (52) (2005) 324–333.
2. M. de León, A Historical Review on Nonholonomic Mechanics, RACSAM (106) (2012) 191–224, dOI 10.1007/s13398-011-0046-2.
3. A. V. Borisov, I. S. Mamaev, On the History of the Development of the Nonholonomic Mechanics, Regular and Chaotic Dynamics 7 (1) (2002) 43–47.
4. E. Jarzebowska, Dynamics Modeling of Nonholonomic Mechanical Systems: Theory and Applications, Nonlinear Analysis 63 (5–7) (2005) e185–e197.
5. E. Jarzebowska, Model-Based Control Strategies for Systems with Constraints of the Program Type, Communications in Nonlinear Science and Numerical Simulation 11 (5) (2006) 606–623.
6. R. Kelly, V. Santibáñez, A. Loría, Control of Robot Manipulators in Joint Space, Springer-Verlag, London, 2005.
7. I. V. Abramov, R. G. Mukharlyamov, Z. K. Kirgizbayev, Control of Dynamic Systems with Programmed Constraints, NVSU, Nizhnevartovsk, 2013, in Russian.
8. A. C. Galliulin, R. G. Mukharlyamov, I. A. Mukhametzyanov, Control of Dynamic Systems with Programmed Constraints, NVSU, Nizhnevartovsk, 2013, in Russian.
9. T. D. Chernet, Constructing Dynamic Equations of Constrained Mechanical Systems, Bulletin of PFUR. Series “Mathematics. Information Science. Physics” (3) (2013) 92–104.
10. M. R. Flannery, D’Alembert–Lagrange Analytical Dynamics for Nonholonomic Systems, Journal of Mathematical Physics (52).

УДК 531.3

Управление программным движением неголономной системы второго порядка вдоль траектории

Ч. Т. Дересса

Кафедра теоретической физики и механики
Российский университет дружбы народов
ул. Миклухо-Маклая, д. 6, Москва, Россия, 117198

Принцип Даламбера–Лагранжа позволяет построить уравнения динамики голономных и неголономных систем произвольного порядка. На практике использование этого принципа ограничивается идеальными голономными и линейными неголономными связями первого порядка. В последние годы этот известный принцип непосредственно используется для построения уравнений динамики системы со связями, зависящими от ускорений. В данной работе предлагается аналитическое решение задачи управления программным движением по траектории, зависящей от ускорения. Связи в зависимости от источника воздействия делятся на естественные и программируемые. Управление траекторией слежения осуществляется посредством использования модели планирования управляемого движения, построенного с учетом программируемых и естественных ограничений, и модели динамического управления, разработанной с учетом только естественных ограничений. Управление модели планирования движения по траектории используется для планирования траектории, определяемой ускорениями точек системы или ограничениями, соответствующими программе движения. Для управления движением по траектории и стабилизации используется динамическая модель управления. Наконец, для подтверждения эффективности предлагаемого в работе подхода приводится пример. Результаты моделирования изображены на графике.

Ключевые слова: программные связи, естественные ограничения, связи, программное движение, управление, динамические модели управления, траектория, стабилизация
.

Литература

1. *Bloch A. M., Marsdeny J. E., Zenkovz D. V.* Notices of the American Mathematical Society // Nonholonomic Dynamics. — 2005. — No 52. — Pp. 324–333.
2. *de León M.* A Historical Review on Nonholonomic Mechanics // RACSAM. — 2012. — No 106. — Pp. 191–224. — DOI 10.1007/s13398-011-0046-2.
3. *Borisov A. V., Mamaev I. S.* On the History of the Development of Nonholonomic Mechanics // Regular and Chaotic Dynamics. — 2002. — Vol. 7, No 1. — Pp. 43–47.
4. *Jarzebowska E.* Dynamics Modeling of Nonholonomic Mechanical Systems: Theory and Applications // Nonlinear Analysis. — 2005. — Vol. 63, No 5–7. — Pp. e185–e197.
5. *Jarzebowska E.* Model-Based Control Strategies for Systems with Constraints of the Program Type // Communications in Nonlinear Science and Numerical Simulation. — 2006. — Vol. 11, No 5. — Pp. 606–623.
6. *Kelly R., Santibáñez V., Loría A.* Control of Robot Manipulators in Joint Space. — London: Springer-Verlag, 2005.
7. *Абрамов Н. В., Мухарлямов Р. Г., Киргизбаев Ж. К.* Управление динамикой систем с программными связями. — Нижневартовск: НВГУ, 2013. — 162 с.
8. Построение систем программного движения / А. С. Галлиулин, И. А. Мухаметзянов, Р. Г. Мухарлямов, В. Д. Фурасов. — Москва: Наука, 2013.
9. *Deressa C. T.* Constructing Dynamic Equations of Constrained Mechanical Systems // Bulletin of PFUR. Series “Mathematics. Information Science. Physics”. — 2013. — No 3. — Pp. 92–104.
10. *Flannery M. R.* D’Alembert–Lagrange Analytical Dynamics for Nonholonomic Systems // Journal of Mathematical Physics. — 2011. — No 52.