A Nonlinear programming method for time-optimal control of an omni-directional mobile robot

Shi-Min Wang, Chia-Ju Wu, Jia-Yan Wei

Abstract

The time-optimal control problem of a three-wheeled omni-directional mobile robot is addressed in this paper. Different from usual cases, in which the Pontryagin’s Minimum Principle (PMP) is used, an iterative procedure is proposed to transform the time-optimal problem into a nonlinear programming (NLP) one. In the NLP problem, the count of control steps is fixed initially and the sampling period is treated as a variable in the optimization process. The optimization object is to minimize the sampling period such that it is below a specific minimum value, which is set in advance considering the accuracy of discretization. To generate initial feasible solutions of the formulated NLP problem, genetic algorithms (GAs) are adopted. Since different initial feasible solutions can be generated, the optimization process can be started from different points to find the optimal solution. In this manner, one can find a time-optimal movement of the omni-directional mobile robot between two configurations. To show the feasibility of the proposed method, simulation results are included for illustration.

Keywords: Time-optimal control, nonlinear programming, omni-directional robots.

Shi-Min Wang, Lecturer, Department of Electrical Engineering, Hsiuping University of Science and Technology.
Chia-Ju Wu,Professor, Department of Electrical Engineering,
National Yunlin University of Science and Technology.
Jia-Yan Wei, Lecturer, Department of Electrical Engineering, Hsiuping University of Science and Technology.
Received 12 January 2011; accepted 28 March 2011

可全方位運動機器人之非線性規劃最佳時間控制法

王世民、吳佳儒、魏嘉延

摘　要

本論文為探討一部三輪可全方位運動機器人的時間最佳化控制問題，這裏所提出的方法為 Pontryagin 的最小原則（PMP）。所使用的迭代法為非線性規劃（NLP）方法的時間最佳化題型，NLP 問題的初始值在控制過程中為常數，而取樣週期在做最佳化過程中為變數，最佳化的目的是希望取樣週期要比設定的最小值更低，如此一來才可確保它的準確度。本論文所制定的NLP問題初始可行解可由遺傳演算法(GAs)來求得，因為初始可行解可求得，所以時間最佳化問題便可進行計算而得到最佳解。在這種模式下，可全方位運動機器人在空間移動時就可以找到最佳的時間運動方式。本論文所提出的方法可經由模擬結果來作說明得到驗證。

關鍵詞：時間最佳化控制，非線性規劃，全方位機器人

王世民：修平科技大學電機系講師
吳佳儒：國立雲林科技大學電機工程系教授
魏嘉延：修平科技大學電機系講師
投稿日期：2011 年 1 月 12 日　接受刊登日期：2011 年 3 月 28 日

1. Introduction

In recent years, mobile robots have been used widely in many occasions [1]. Among several kinds of mobile robot, the omni-directional ones have attracted much attention since they have the ability to move simultaneously and independently in translation and rotation [2]. A typical application of omni-directional mobile robots is the annual international Robocup competition [3], in which omni-directional mobile robots are used to play soccer-like games.

Many researchers have studied omni-directional mobile robots and most research has been focused on the mechanical design and dynamic analysis. Pin and Killough [2] presented the concepts for a family of holonomic wheeled platforms that feature full omni-directionality with simultaneous and independent controlled rotational and translational capabilities. Jung *et al*. [4] developed an omni-directional mobile robot, derived its kinematic and dynamic models, and used a fuzzy logic controller for the shooting action control. Kalmar-Nagy *et al.* [5] proposed an innovative method to generate near-optimal trajectories for an omni-directional robot. This method provided an efficient method for path planning and allowed a large number of possible scenarios to be explored in real time. William II *et al*. [6] presented a dynamic model for omni-directional wheeled mobile robots, considering the occurrence of slip between the wheels and motion surface. Chen *et al.* [7] presented an off-road omni-directional robot, which can run on an uneven road and obstacles. They also designed a position and velocity control system for the robot such that the robot can be automatically controlled to run in an optional direction and to track an orbit. With the same kind of omni-directional robot in [7], Chen *et al*. [8] developed an intelligent genetic programming method to search for an optimum route leading the robot to given destination and avoiding obstacles. Liu *et al*. [9] designed a nonlinear controller for an omni-directional mobile robot utilizing the so-called linearization control method such that robust stability and performance can be provided. In [10,11], the dynamic model of an omni-directional mobile robot is developed, and several control strategies are discussed based on linear control methods while the robot dynamics is nonlinear. A resolved-acceleration control with PI and PD feedback is developed in [10] and PID control, self-tuning PID control, and fuzzy control of the omni-directional mobile robot are introduced in [11].

From the robot testing and the competition experience of Robocup games, it is realized that a time-optimal control method for the mobile robots between configurations can improve their performance significantly. In the past few years, the time-optimal problem of mobile robots has attracted the attention of several researchers [12-14]. However, to the best knowledge of the authors, previous researchers have not addressed the time-optimal control problem of an omni-directional mobile robot yet. This motivates the research in this paper and a NLP method will be proposed to carry out the motion maneuver of an omni-directional mobile robot between two configurations in minimum-time.

The time-optimal motion-planning (TOMP) problem for an omni-directional mobile robot is to find the time-optimal motion in a smooth flat surface between two configurations, where the initial and final velocities are zero. Usually, this TOMP problem leads to the utilization of the PMP [15], in which one needs to solve a set of differential equations. Since these equations are usually nonlinear and highly coupled, one will have two-point boundary value problems, which are intractable in numerical computation.

Recently, a NLP method that does not utilize the PMP was developed by one of the authors of this paper to solve the time-optimal control problem of linear systems [16]. The basic idea of this method is that instead of considering a fixed sampling period, the count of control steps is fixed initially and the sampling period is treated as a variable in the optimization process. The optimization object is to minimize the sampling period such that it is below a specific minimum value, which is set in advance considering the accuracy of discretization. With this approach, the optimization procedure requires only two iterations in most linear cases, thereby reducing the computation time dramatically.

Extending the concept in [16] to nonlinear systems, this paper shows the generation of time-optimal motion between two configurations for an omni-directional mobile robot with three independently driven individual wheels. In the beginning, dynamical equations of the omni-directional mobile robot are introduced and an iterative procedure will be proposed to transform the time-optimal problem into a NLP one. However, since the dynamics of the omni-directional robot is highly nonlinear, it is a difficult task to find a feasible solution for the formulated NLP problem. Therefore, a GA-based approach is proposed to generate feasible solutions for the formulated NLP problem. In this manner, since feasible solutions can be obtained easily, the optimization process can be started from many different starting points to find the optimal solution. Simulation examples are given to verify the feasibility of the proposed method.

The rest of this paper is as follows. In Section 2, dynamical equations of the omni-directional mobile robot are derived. Then the TOMP problem between two configurations of the omni-directional mobile robot is formulated as a NLP one by an iterative procedure in Section 3. In Section 4, GAs are used to generate initial feasible solutions of the NLP problem. Problem solution and simulation results are shown in Sections 5 and 6, respectively. Finally, conclusions and discussion are given in Section 7.

2. Dynamic Equations of the Omni-directional Robot

In this section, it is assumed that the omni-directional robot consists of the orthogonal-wheel assembly mechanism proposed in [2] and a schematic diagram to illustrate the motion of the omni-directional robot is given as shown in Figure 1. In the working space of the robot, a world-frame $[x_w, y_w]^T$ and a moving-frame $[x_m, y_m]^T$ are defined as shown in Figure 2. The world-frame denotes a frame that everything discussed can be referenced and the moving-frame is a frame attached to the center of the gravity of the robot. The transformation between these two frames is described by

$$\begin{bmatrix} \dot{x}_w \\ \dot{y}_w \end{bmatrix} = \begin{bmatrix} \cos\phi & -\sin\phi \\ \sin\phi & \cos\phi \end{bmatrix} \begin{bmatrix} \dot{x}_m \\ \dot{y}_m \end{bmatrix} \tag{1}$$

where ϕ is the angle between these two frames.

With the transformation in (1) and according to the Newton's Second Law of Motion, one can obtain

$$M(\ddot{x}_m - \dot{y}_m\dot{\phi}) = -\frac{1}{2}D_1 - \frac{1}{2}D_2 + D_3 \tag{2}$$

$$M(\ddot{y}_m + \dot{x}_m\dot{\phi}) = \frac{\sqrt{3}}{2}D_1 - \frac{\sqrt{3}}{2}D_2 \tag{3}$$

$$I_v\ddot{\phi} = (D_1 + D_2 + D_3)L \tag{4}$$

where M is the mass of the robot, I_v is the moment of inertia of the robot, L is the distance between any wheel and the center of gravity of the robot, and D_i $i = 1,2,3$, are the driving forces of the wheels.

In addition, the driving system property for each wheel is assumed to be given by [17]

$$I_w\ddot{\theta}_i + c\dot{\theta}_i = ku_i - RD_i, \quad i = 1,2,3 \tag{5}$$

where c is the viscous friction factor of the wheel, R is the radius of the wheel,

I_ω is the moment of inertia of the wheel around the driving shaft, k is the driving gain factor, and u_i is the driving input torque.

From (1) through (5), and the geometrical relationships among variables $\dot{\phi}, \dot{x}_m, \dot{y}_m$, and $\dot{\theta}_i$, it is found that

$$\ddot{x}_m = a_1\dot{x}_m + a_2\dot{y}_m\dot{\phi} - b_1(u_1 + u_2 - 2u_3), \quad (6)$$

$$\ddot{y}_m = a_1\dot{y}_m - a_2\dot{x}_m\dot{\phi} + \sqrt{3}b_1(u_1 - u_2), \quad (7)$$

$$\ddot{\phi} = a_3\dot{\phi} + b_2(u_1 + u_2 + u_3), \quad (8)$$

where

$$a_1 = -3c/(3I_\omega + 2MR^2) \quad (9)$$

$$a_2 = 2MR^2/(3I_\omega + 2MR^2) \quad (10)$$

$$a_3 = -3cL^2/(3I_\omega L^2 + I_v R^2) \quad (11)$$

$$b_1 = kR/(3I_\omega + 2MR^2) \quad (12)$$

$$b_2 = kRL/(3I_\omega L^2 + I_v R^2) \quad (13)$$

From (6) through (13), and the transformation between the world-frame and the moving-frame, the dynamical equations of the omni-directional robot are given as

$$\frac{d}{dt}\begin{bmatrix}\dot{x}_w\\ \dot{y}_w\\ \dot{\phi}\end{bmatrix} = \begin{bmatrix}a_1 & -a_4\dot{\phi} & 0\\ a_4\dot{\phi} & a_1 & 0\\ 0 & 0 & a_3\end{bmatrix}\begin{bmatrix}\dot{x}_w\\ \dot{y}_w\\ \dot{\phi}\end{bmatrix} + \begin{bmatrix}b_1\beta_1 & b_1\beta_2 & 2b_1\cos\phi\\ b_1\beta_3 & b_1\beta_4 & 2b_1\cos\phi\\ b_2 & b_2 & b_2\end{bmatrix}\begin{bmatrix}u_1\\ u_2\\ u_3\end{bmatrix} \quad (14)$$

where

$$a_4 = 3I_\omega/(3I_\omega + 2MR^2) \quad (15)$$

$$\beta_1 = -\sqrt{3}\sin\phi - \cos\phi \quad (16)$$

$$\beta_2 = \sqrt{3}\sin\phi - \cos\phi \quad (17)$$

$$\beta_3 = \sqrt{3}\cos\phi - \sin\phi \quad (18)$$

$$\beta_4 = -\sqrt{3}\cos\phi - \sin\phi \quad (19)$$

3. TOMP between Two Configurations

3.1 Problem Formulation

The TOMP problem of the omni-directional mobile robot between two configurations is to find the control inputs that will move the system from an initial configuration to a desired final configuration while minimizing the traveling time. With the dynamics in (14) through (19), the TOMP problem can be formulated as follows:

PROBLEM 1: For the omni-directional mobile robot described in (14) through (19), assuming that the initial configuration is given as

$$(x_w(0), y_w(0), \phi(0)) = (x_0, y_0, \phi_0) \quad (20)$$

$$(\dot{x}_w(0), \dot{y}_w(0), \dot{\phi}(0)) = (0, 0, 0) \quad (21)$$

determine the control inputs $u_1(t)$, $u_2(t)$, and $u_3(t)$ for $t \in [0, t_f]$ to minimize

$$J = t_f \tag{22}$$

subject to

$$(x_w(t_f), y_w(t_f), \phi(t_f)) = (x_f, y_f, \phi_f) \tag{23}$$

$$(\dot{x}_w(t_f), \dot{y}_w(t_f), \dot{\phi}(t_f)) = (0, 0, 0) \tag{24}$$

and

$$u_{i,\min} \le u_i(t) \le u_{i,\max} \quad \text{for}$$

$$t \in [0, t_f];\ i = 1,2,3 \tag{25}$$

where (x_f, y_f, ϕ_f) is the desired final configuration.

It is obvious that Problem 1 is a very difficult problem due to the nature of the nonlinear and coupled relation of the omni-directional mobile robot. To cope with the difficulty, Problem 1 will be formulated and solved in the discrete-time domain by numerical methods. By extending the concept in [16], it will be shown how to determine the time-optimal movement of an omni-directional mobile robot between configurations. The first step is to divide the interval $[0, t_f]$ into N equal time intervals, where N is the number of control steps [16]. That is

$$t_i - t_{i-1} = \Delta t = t_f / N \qquad \text{for } i = 1, 2, \cdots, N \tag{26}$$

If the acceleration is assumed to be constant for each sub-interval, then one obtains

$$\begin{bmatrix} \dot{x}_w(i) \\ \dot{y}_w(i) \\ \dot{\phi}(i) \end{bmatrix} = \begin{bmatrix} \dot{x}_w(i-1) + \ddot{x}_w(i-1)\cdot \Delta t \\ \dot{y}_w(i-1) + \ddot{y}_w(i-1)\cdot \Delta t \\ \dot{\phi}_w(i-1) + \ddot{\phi}_w(i-1)\cdot \Delta t \end{bmatrix}$$
$$= \begin{bmatrix} \dot{x}_w(0) + \sum_{k=0}^{i-1} \ddot{x}_w(k)\cdot \Delta t \\ \dot{y}_w(0) + \sum_{k=0}^{i-1} \ddot{y}_w(k)\cdot \Delta t \\ \dot{\phi}(0) + \sum_{k=0}^{i-1} \ddot{\phi}(k)\cdot \Delta t \end{bmatrix} \tag{27}$$

for $i = 1,2,\cdots,N$

$$\begin{bmatrix} x_w(i) \\ y_w(i) \\ \phi(i) \end{bmatrix} =$$
$$\begin{bmatrix} x_w(i-1) + 0.5\times(\dot{x}_w(i) + \dot{x}_w(i-1))\cdot \Delta t \\ y_w(i-1) + 0.5\times(\dot{y}_w(i) + \dot{y}_w(i-1))\cdot \Delta t \\ \phi(i-1) + 0.5\times(\dot{\phi}(i) + \dot{\phi}(i-1))\cdot \Delta t \end{bmatrix} \tag{28}$$
$$= \begin{bmatrix} x_w(0) + 0.5\times\sum_{k=1}^{i} (\dot{x}_w(k) + \dot{x}_w(k-1))\cdot \Delta t \\ y_w(0) + 0.5\times\sum_{k=0}^{i-1} (\dot{y}_w(k) + \dot{y}_w(k-1))\cdot \Delta t \\ \phi(0) + 0.5\times\sum_{k=0}^{i-1} (\dot{\phi}(k) + \dot{\phi}(k-1))\cdot \Delta t \end{bmatrix}$$

for $i = 1,2,\cdots,N$

where $(x_w(i), y_w(i), \phi(i))$ and $(\dot{x}_w(i), \dot{y}_w(i), \dot{\phi}(i))$ are used to denote $(x_w(i\cdot\Delta t), y_w(i\cdot\Delta t), \phi(i\cdot\Delta t))$ and $(\dot{x}_w(i\cdot\Delta t), \dot{y}_w(i\cdot\Delta t), \dot{\phi}(i\cdot\Delta t))$, respectively, for notational simplicity.

If $(u_1(0), u_2(0), u_3(0))$ are substituted into (14), with the given $(x_w(0), y_w(0), \phi(0))$ and $(\dot{x}_w(0), \dot{y}_w(0), \dot{\phi}(0))$, then the values of $(x_w(1), y_w(1), \phi(1))$ and $(\dot{x}_w(1), \dot{y}_w(1), \dot{\phi}(1))$ can be obtained from (27) and (28). Applying input torques to (14) sequentially and repeatedly using (27) and (28), the final configuration of the robot can be expressed as functions of $(x_w(0), y_w(0), \phi(0))$, $(\dot{x}_w(0), \dot{y}_w(0), \dot{\phi}(0))$, the input variables $(u_1(0), u_2(0), u_3(0)), \cdots, (u_1(N-1), u_2(N-1), u_3(N-1))$, and the sampling period Δt. This means that

$$x_w(N) = f_1(x_w(0), y_w(0), \phi(0), \dot{x}_w(0), \dot{y}_w(0), \dot{\phi}(0), u_1, u_2, u_3, \Delta t) \quad (29)$$

$$y_w(N) = f_2(x_w(0), y_w(0), \phi(0), \dot{x}_w(0), \dot{y}_w(0), \dot{\phi}(0), u_1, u_2, u_3, \Delta t) \quad (30)$$

$$\phi(N) = f_3(x_w(0), y_w(0), \phi(0), \dot{x}_w(0), \dot{y}_w(0), \dot{\phi}(0), u_1, u_2, u_3, \Delta t) \quad (31)$$

$$\dot{x}_w(N) = f_4(x_w(0), y_w(0), \phi(0), \dot{x}_w(0), \dot{y}_w(0), \dot{\phi}(0), u_1, u_2, u_3, \Delta t) \quad (32)$$

$$\dot{y}_w(N) = f_5(x_w(0), y_w(0), \phi(0), \dot{x}_w(0), \dot{y}_w(0), \dot{\phi}(0), u_1, u_2, u_3, \Delta t) \quad (33)$$

$$\dot{\phi}(N) = f_6(x_w(0), y_w(0), \phi(0), \dot{x}_w(0), \dot{y}_w(0), \dot{\phi}(0), u_1, u_2, u_3, \Delta t) \quad (34)$$

where $\mathbf{u_1} = (u_1(0), u_1(1), \cdots, (u_1(N-1))$, $\mathbf{u_2} = (u_2(0), u_2(1), \cdots, (u_2(N-1))$, and $\mathbf{u_3} = (u_3(0), u_3(1), \cdots, (u_3(N-1))$. A flowchart to illustrate the derivation of (29) through (34) is shown in Figure 3. With (29) through (34), Problem 1 is now turned into a standard constrained NLP problem as follows:

PROBLEM 2: Given the initial configuration in (20) and (21), determine the values of $u_1(0), u_1(1), \cdots, u_1(N-1)$, $u_2(0), u_2(1), \cdots, u_2(N-1)$, $u_3(0), u_3(1), \cdots, u_3(N-1)$, and Δt to minimize

$J = N \cdot \Delta t$ (35)

subject to

$\Delta t > 0$ (36)

$(x_w(N), y_w(N), \phi(N)) = (x_f, y_f, \phi_f)$ (37)

$(\dot{x}_w(N), \dot{y}_w(N), \dot{\phi}(N)) = (0,0,0)$ (38)

$u_{i,\min} \le u_i(j) \le u_{i,\max}$ for $i = 1,2,3;\ j = 0,1,\cdots, N-1$ (39)

where $(x(N), y(N), \phi(N))$ and

$(\dot{x}(N), \dot{y}(N), \dot{\phi}(N))$ are defined in (29) through (34).

3.2 Choice of Control Steps and Sampling Period

Although the TOMP problem of an omni-directional mobile robot can be formulated as shown in Problem 2, there still exist several difficulties to be solved. One difficulty is the choice of the value of control steps N. It is obvious that a larger value of N gives more freedom for the input variables. However, this also means more computation burden for Problem 2. For linear system without constraints on the input variables, it has been shown that the initial choice of N must be greater than the dimension of state variables [16]. Though no similar rules can be followed for nonlinear systems, an integer that is large than the dimension of state variables will be chosen as an initial value of N in this paper.

Another difficulty is the choice of the sampling period. From the viewpoint of discretization accuracy, it is obvious that smaller sampling period value will result in a more accurate model. Therefore, a limitation of the sampling period, say Δt_{limit}, should be chosen. If the value of Δt obtained in Problem 2 is greater than Δt_{limit}, then a new value of control steps will be chosen according to

$$N_{\text{new}} > \frac{N \cdot \Delta t}{\Delta t_{\text{limit}}} \tag{40}$$

4. Initial Feasible Solutions

Most NLP algorithms usually need an initial feasible solution to start the optimization process. In Problem 2, an initial feasible solution means a set of $u_1(0), u_1(1), \cdots, u_1(N-1)$, $u_2(0), u_2(1), \cdots, u_2(N-1)$, $u_3(0), u_3(1), \cdots, u_3(N-1)$, and Δt satisfying the constraints in (36) through (39). It is obvious that these solutions are not easy to be found since the constraints are highly nonlinear and coupled. Therefore, an approach based on GAs is developed to generate initial feasible solutions.

The theoretical basis of GAs is that chromosomes (solutions) better suited to the environment (evaluation) will have greater chance of survival and better chance of producing offspring. The evolutionary process is based primary on the mutation and crossover operators. The crossover operator combines the features of two parents to form two offspring. The mutation operator arbitrarily alters one or more genes of a selected chromosome, which increases the variability of the population. These two operators can further be divided into static and dynamic, where static ones do not change over the life of the population while dynamic ones are functions of time.

In the evolutionary process to generate initial feasible solutions of Problem 2, genetic operators such as real number encoding, arithmetical crossover and non-uniform mutation will be implemented. Moreover, dynamic mutation and crossover, enlarged sampling space and ranking mechanism will also be used to expedite the convergence of the evolutionary process.

4.1 Chromosome Representations

How to encode a solution of the problem into a chromosome is a key issue for GAs. In this paper, since the parameters to be determined are all real, real number encoding technique will be used. Once the real-code chromosomes are used, the next step is to determine the number of genes in a chromosome. If the number of control steps is *N*, then the chromosomes will contains (3*N*+1) genes, which denote $u_1(0), u_1(1), \cdots, u_1(N-1)$, $u_2(0), u_2(1), \cdots, u_2(N-1)$, $u_3(0), u_3(1), \cdots, u_3(N-1)$, and Δt, respectively. For a chromosome $\mathbf{x} = [x_1, x_2, x_{3N+1}]$, one can find that the first 3*N* genes are within the ranges $[u_{i,\min}, u_{i,\max}]$ for $i = 1, 2, 3$, and the lower bound of the last gene is greater than zero.

4.2 Crossover and Mutation Operations [18]

Arithmetical crossover and non-uniform mutation will be introduced in this section. For two real-coded chromosomes $\mathbf{x}_1$ and $\mathbf{x}_2$, the operation of arithmetical crossover is defined as follows:

$$\mathbf{x}_1' = \lambda \mathbf{x}_1 + (1-\lambda)\mathbf{x}_2 \tag{41}$$

$$\mathbf{x}_2' = \lambda \mathbf{x}_2 + (1-\lambda)\mathbf{x}_1 \tag{42}$$

where $\lambda \in (0,1)$.

For a given parent $\mathbf{x}$, if a gene x_k of it is selected for mutation, then the resulting offspring will be randomly selected from one of the following two choices.

$$x_k' = x_k + (x_k^U - x_k) \cdot r \cdot \left(1 - \frac{gen}{G}\right)^b \tag{43}$$

$$x_k' = x_k - (x_k - x_k^L) \cdot r \cdot \left(1 - \frac{gen}{G}\right)^b \tag{44}$$

where x_k^U and x_k^L are the upper and lower bounds of x_k; r is a random number from $[0,1]$; gen is the generation number; G is the maximal generation number, and b is a parameter determining the degree of non-uniformity.

In addition to arithmetical crossover and non-uniform mutation, dynamic crossover and mutation probability rates will also be used for fast convergence. The crossover and mutation rates are defined as follows:

$$\text{crossover rate} = \exp\left(-\frac{gen}{G}\right) \tag{45}$$

$$\text{mutation rate} = \exp\left(-\frac{gen}{4G}\right) - 1 \tag{46}$$

4.3 Enlarge Sampling Space

To generate good offspring, a method for selection of parents will be necessary. For selection methods that are developed based on regular sampling space, parents are replaced by their offspring soon after they give birth. In this manner, some fitter chromosomes will be worse than their parents. To cope with this problem, the selection method in this paper will be performed in enlarged sampling space, in which both parents and offspring have the same chance of competition for survival. Moreover, since more random perturbation is allowed in enlarged sampling space, high crossover and mutation will be allowed in the evolutionary process.

4.4 Ranking Mechanism

In proportional selection procedure, the selection probability of a chromosome is proportional to its fitness. This scheme exhibits some undesirable properties such as a few super chromosomes will dominate the process of selection in early generations. Moreover, competition among chromosomes will be less strong and a random search behavior will emerge in later generations. Therefore, the ranking mechanism is used in this paper to mitigate these problems, in which the chromosomes are selected proportionally to their ranks rather than actual evaluation values. This means that the fitness will be an integer number from 1 to P, where P is the population size. The best chromosomes will have a fitness value equal to P and the worst one will have a fitness value equal to 1.

5. Problem Solution

The details of the proposed method can be summarized as follows:

Algorithm A: (Generating an initial feasible solution)

Step 1: Define the fitness function.

Step 2: Determine the population size, the crossover rate according to (45), and the mutation rate according (46).

Step 3: Produce an initial generation in a random way.

Step 4: Evaluate the fitness for each member of generation.

Step 5: With the crossover rate in Step 2, generate offspring according to (41) and (42), in which the ranking mechanism is used for selection of chromosomes.

Step 6: With mutation rate in Step 2, generate offspring according to (43) and (44).

Step 7: Select the members of the new generation from the parents in the old generation and the offspring in Step 5 and Step 6 according to their fitness values.

Step 8: Repeat the procedure in Step 5 through Step 7 until the number of generations reaches a prescribed value.

Algorithm B：(Solution of Problem 2)

Step 1: Choose a value of Δt_{limit} and an integer N.

Step 2: Formulate the TOMP problem as a NLP problem as shown in Problem 2 with the chosen value N.

Step 3: Use Algorithm A to find an initial feasible solution of Problem 2.

Step 4: Use any NLP algorithm to determine the minimum value of Δt in Problem 2 based on the initial feasible solution obtained in Step 3.

Step 5: If $\Delta t > \Delta t_{\text{limit}}$, then choose a new value of N according to (40) and go to Step 2. Otherwise, continue.

Step 6: $N \cdot \Delta t$ is the minimal traveling time.

6. Simulation Results

In this simulation example, the omni-directional mobile robot is to be moved from the initial configuration

$$(x_w(0), y_w(0), \phi(0)) = (0\text{ m}, 0\text{ m}, 0^\circ) \quad (47)$$

$$(\dot{x}(0), \dot{y}(0), \dot{\phi}(0)) = (0\text{ m}, 0\text{ m}, 0^\circ) \quad (48)$$

to the desired final configuration

$$(x_w(N), y_w(N), \phi(N)) = (1\text{ m}, 0\text{ m}, 180^\circ) \quad (49)$$

$$(\dot{x}(N), \dot{y}(N), \dot{\phi}(N)) = (0\text{ m}, 0\text{ m}, 0^\circ) \quad (50)$$

in a time-optimal manner.

For convenience, the dynamical equations used in this example are the same as those in [10,11]. This means that the parameters of the mobile robot are chosen as $M = 9.4$ kg, $L = 0.178$ m, $I_v = 11.25\text{ kg}\cdot\text{m}^2$, $I_\omega = 0.02108\text{ kg}\cdot\text{m}^2$, $c = 5.983\times 10^{-6}\text{ kg}\cdot\text{m}^2/\text{s}$, $R = 0.0245$ m, and $k = 1$, respectively. Meanwhile, the constraints on the input torques are assumed to be

$$-10\text{Nm} \le u_1 \le 10\text{Nm} \tag{51}$$

$$-10\text{Nm} \le u_2 \le 10\text{Nm} \tag{52}$$

$$-10\text{Nm} \le u_3 \le 10\text{Nm} \tag{53}$$

In applying Algorithm A to generate an initial feasible solution, the fitness function is defined as

$$fitness = \frac{1}{1+e^2+\dot{e}^2} \tag{54}$$

where

$$e^2 = (x_f - x_w(N))^2 + (y_f - y_w(N))^2 + (\phi_f - \phi(N))^2 \tag{55}$$

and

$$\dot{e}^2 = (\dot{x}_w(N))^2 + (\dot{y}_w(N))^2 + (\dot{\theta}(N))^2 \tag{56}$$

In applying GAs, the population size and the maximal generation number are chosen to be 50 and 100, respectively. During the simulation, the MATLAB Optimization Toolbox will be used, and the value of Δt_{limit} and the initial value of N are chosen to be 0.05 (sec.) and 11, respectively.

Applying Algorithm B with *N*=11, the values of Δt and $N \cdot \Delta t$ are found to be 0.0985 (sec.) and 1.0835 (sec.), respectively. Since $\Delta t > \Delta t_{\text{limit}}$, the value of N will be updated according to (40), and the new value of N is chosen to be 22. Applying Algorithm B with *N*=22, the values of Δt and $N \cdot \Delta t$ are found to be 0.0475 (sec.) and 1.0461 (sec.), respectively, and the simulation results are shown in Figure 4.

7. Conclusions and Discussion

This paper presented a novel method to solve the TOMP problem of a three-wheeled omni-directional mobile robot. The first step is to transform the problem into a NLP problem by an iterative procedure. Then a GA-based method is proposed for generation of initial feasible solutions since an initial feasible solution is usually needed in solving a NLP problem. Different from the methods that utilizing the PMP, the major advantage of the proposed method is that one does not need to solve a set of highly nonlinear differential equations.

In the proposed method, one may ask why the optimal solution cannot be obtained by applying the GAs directly. From theoretical point of view, this task is possible to be done. However, in practice, the major difficulty is that the feasibility of the solution is very easy to be violated during the evolutionary process. This explains why the time-optimal solution cannot be obtained by applying the GAs directly.

It can be proved that the solution obtained satisfying the Kuhn-Tucker condition [19], which is a criterion used to check a local minimum. In addition, from the simulation results in Figure 4, one also can find that at least one of the four control inputs saturated at any time instant. This means that the solution is in the form of bang-bang control [20]. If a solution does not satisfy the Kuhn-Tucker condition or not in the form of bang-bang control, then one can conclude that the solution is not a global minimum. However, since the solution obtained meets both criterions simultaneously, it will be hard to determine whether the solution is globally optimal or not. More effort will be needed if one is interested in this issue.

References

[1] R. D. Schraft & G. Schmierer, *Service Robots* (A K Peters, Ltd., 2000).

[2] F. G. Pin & S. M. Killough, “A new family of omnidirectional and holonomic wheeled platforms for mobile robots,” *IEEE Trans. on Robotics and Automation*, *10*, 1994, 480-489

[3] http://www.robocup.org/

[4] M. J. Jung, H. S. Shim, H. S. Kim, & J. H. Kim, “Omni-directional mobile base OK-II,” *Proc. of the IEEE International Conference on Robotics and Automation*, 2000, 3449-3454.

[5] T. Kalmar-Nagy, P. Ganguly, & R. D’Andrea, “Real-time trajectory generation for omnidirectional vehicles,” *Proc. of the American Control Conference,* 2002, 286-291.

[6] R. L. Williams, II, B. E. Carer, P. Gallina, & G. Rosati, “Dynamic Model with slip for wheeled omnidirectional robots,” *IEEE Trans. on Robotics and Automation*, *18*, 2002, 285-293.

[7] P. Chen, S. Mitsutake, T. Isoda, & T. Shi, “Omni-directional robot and adaptive control metod for off-road running,” *IEEE Trans. on Robotics and Automation*, *18*, 2002, 251-256.

[8] P. Chen, S. Koyama, S. Mitsutake, & T. Isoda, “Automatic running planning for omni-directional robot using genetic programming,” *Proc. of the IEEE International Symposium on Intelligent Control*, 2002, 485-489.

[9] Y. Liu, X. Wu, J. J. Zhu, & J. Lew, “Omni-directional mobile robot controller design by trajectory linearization,” *Proc. of the American Control Conference*, 2003, 3423-3428.

[10] K. Watanabe, “Control of an omnidirectional mobile robot,” *Proc. of the Second International Conference on Knowledge-Based Intelligent Electronic Systems,* 1998, 51-60.

[11] K. Watanabe, Y. Shiraishi, & S. G.

Tzaffestas, "Feedback control of an omnidirectional autonomous platform for mobile serve robots," *Journal of Intelligent and Robotic Systems, 20*, 1998, 315-330.

[12] P. Soueres & J. P. Laumond, "Shortest paths synthesis for a car-like robot," *IEEE Trans. on Automatic Control, 41*, 1996, 672-688.

[13] W. Wu, H. Chen, & P. Y. Woo, "Optimal motion planning for a wheeled mobile robot," *Proc. of the IEEE International Conference on Robotics and Automation*, 1999, 41-46.

[14] Y. Zheng & P. Moore, "The design of time-optimal control for two-wheel driven carts tracking a moving target," *Proc. of the IEEE International Conference on Decision and Control*, 1995, 3831–3836.

[15] L. S. Pontryagin, V. G. Boltyanskii, R. V. Gamkrelidze, & E. F. Mishchenko, *The Mathematical Theory of Optimal Processes* (New York: Gordon and Beach, 1986).

[16] T. S. Chung & C. J. Wu, "A computationally efficient numerical algorithm for the minimum-time control problem of continuous systems," *Automatica*, *28*, 1992, 841–847.

[17] M. Saito & T. Tsumura, "Collision avoidance among multiple mobile robots – A local approach based on nonlinear programming, *Trans. of the Institute of Systems, Control, and Information Engineers*, *3*, 1990, 252-260.

[18] M. Gen & R. Cheng, *Genetic Algorithm and Engineering Design* (New York: Wiley, 1997).

[19] D. G. Luenberger, *Linear and Nonlinear Programming* (Addison-Wesley, 1973).

F. L. Lewis, *Optimal Control* (New York: Wiley, 1986).

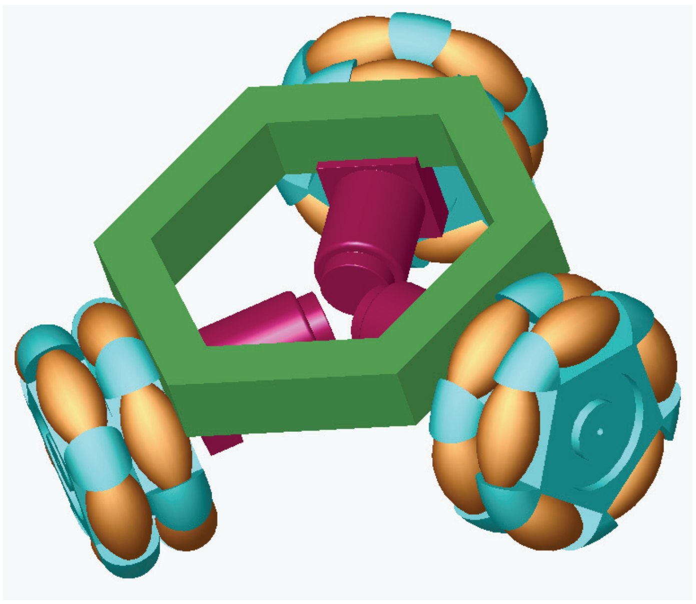

Figure 1. A schematic diagram of the omni-directional robot.

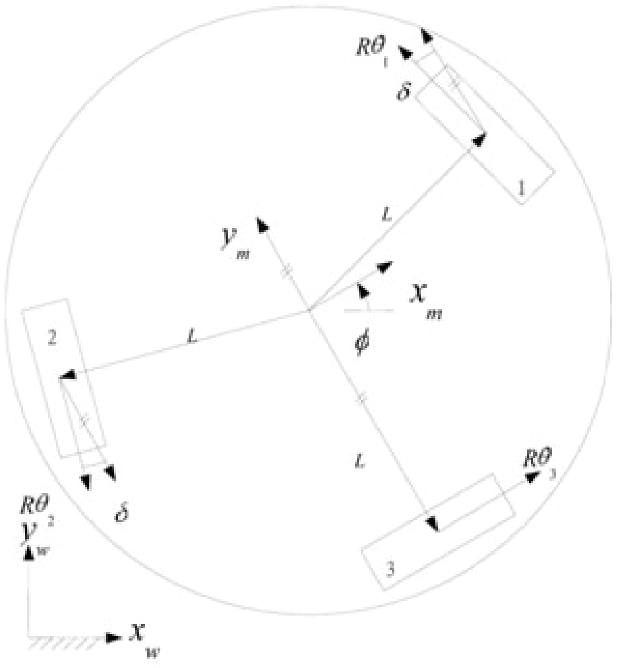

Figure 2. Definitions of the word-frame $[x_w, y_w]^T$ and the moving-frame $[x_m, y_m]^T$.

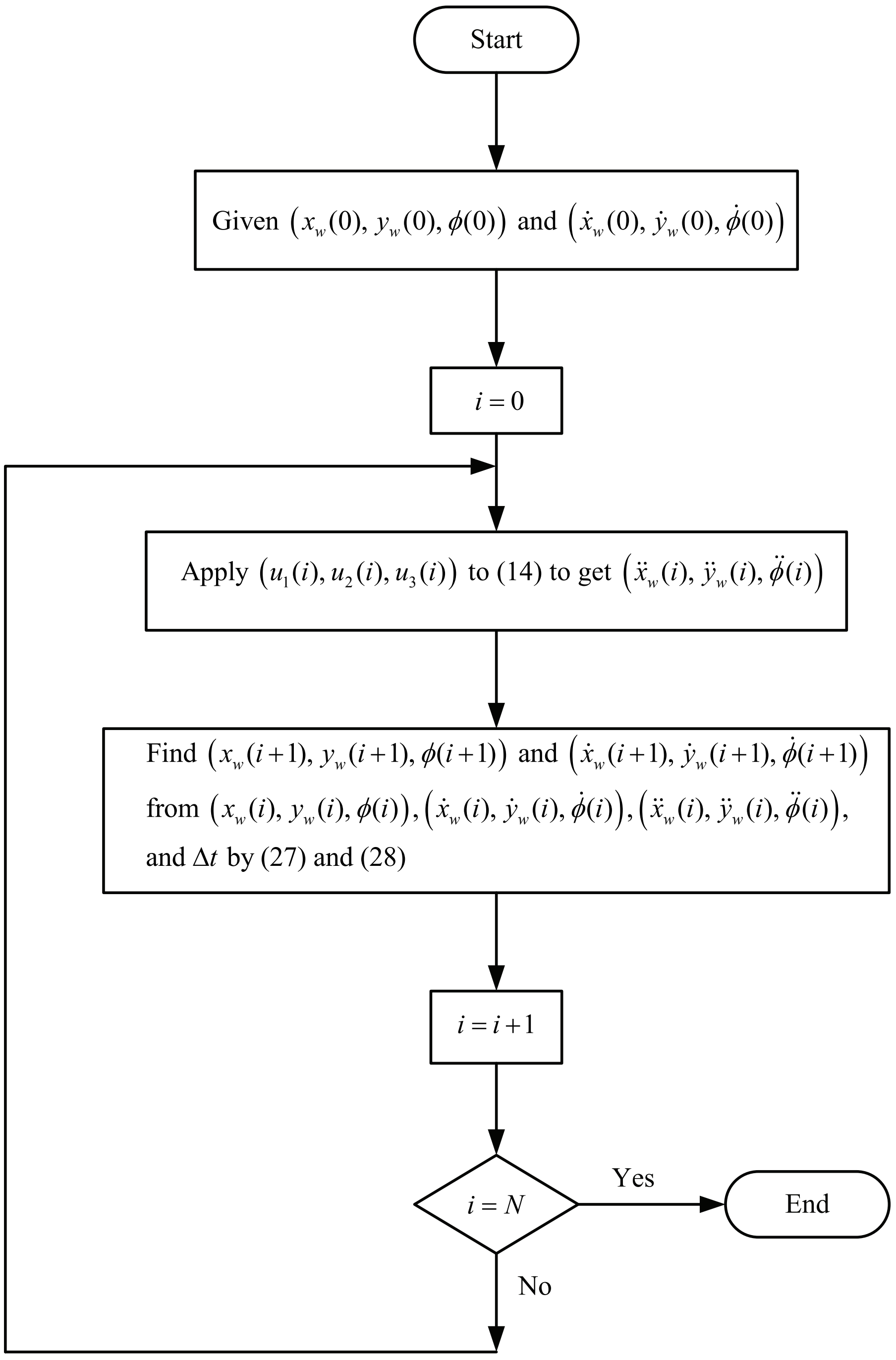
Start
Given $(x_w(0), y_w(0), \phi(0))$ and $(\dot{x}_w(0), \dot{y}_w(0), \dot{\phi}(0))$
$i = 0$
Apply $(u_1(i), u_2(i), u_3(i))$ to (14) to get $(\ddot{x}_w(i), \ddot{y}_w(i), \ddot{\phi}(i))$
Find $(x_w(i+1), y_w(i+1), \phi(i+1))$ and $(\dot{x}_w(i+1), \dot{y}_w(i+1), \dot{\phi}(i+1))$ from $(x_w(i), y_w(i), \phi(i)), (\dot{x}_w(i), \dot{y}_w(i), \dot{\phi}(i)), (\ddot{x}_w(i), \ddot{y}_w(i), \ddot{\phi}(i))$, and Δt by (27) and (28)
$i = i + 1$
$i = N$
Yes
End
No

Figure 3. A flowchart to illustrate the derivation of equations (29) through (34).

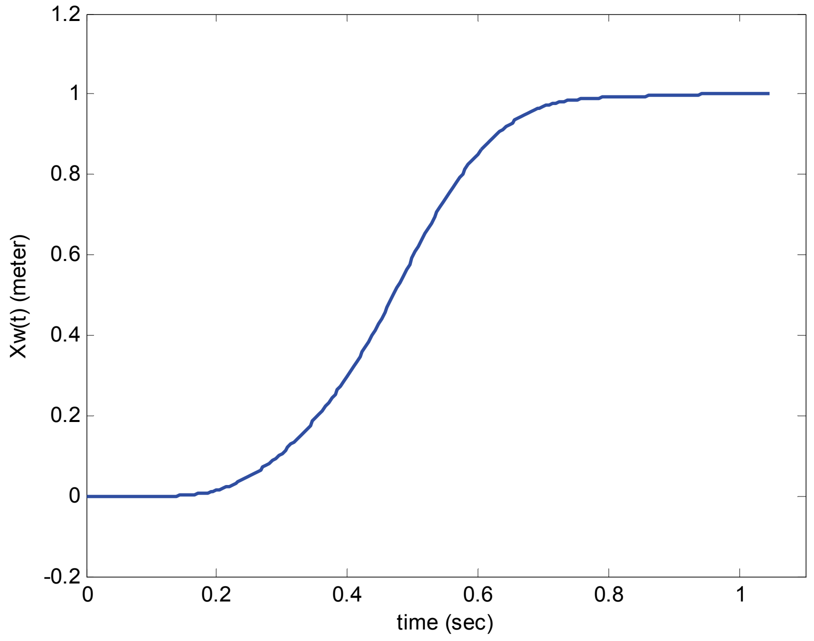

Figure 4(a). Plot of $x_w(t)$ for $N = 22$.

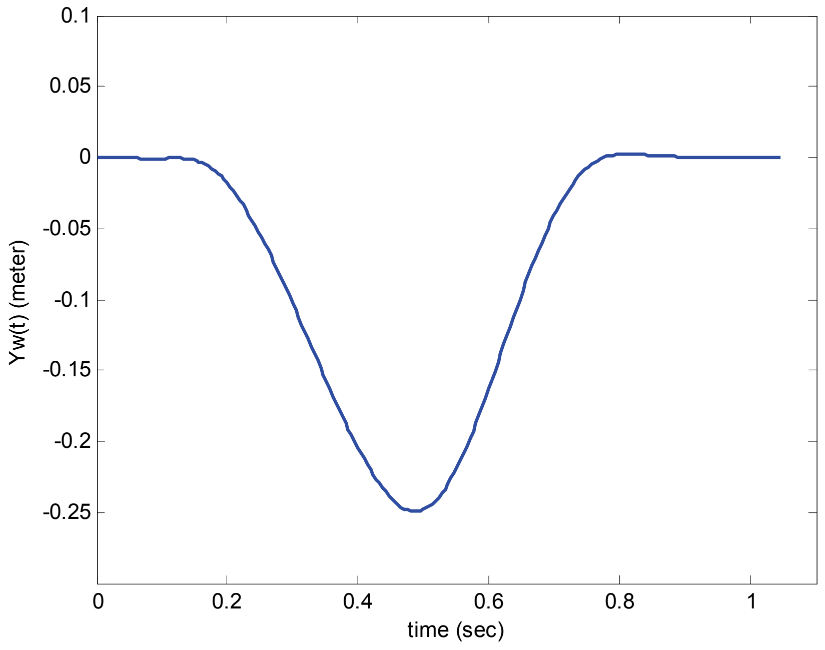

Figure 4(b). Plot of $y_w(t)$ for $N = 22$.

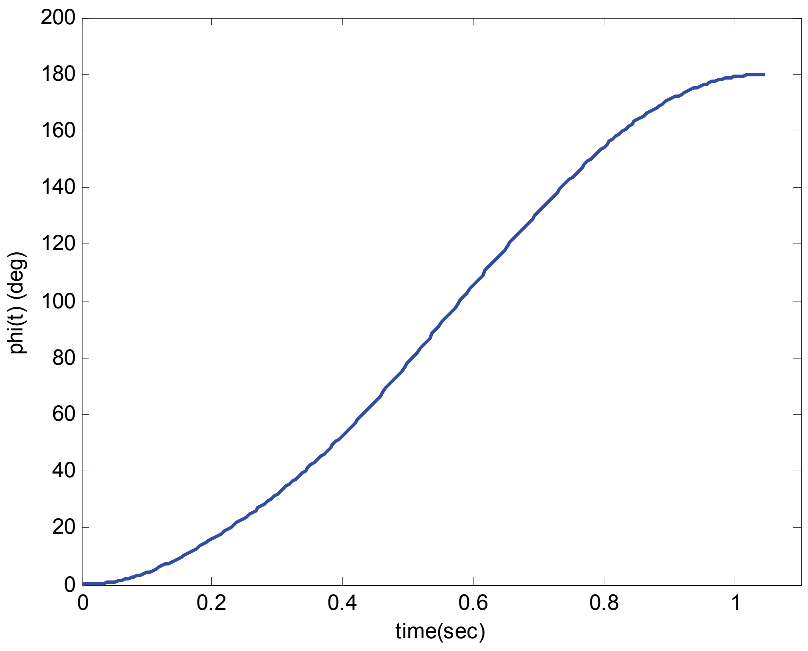

Figure 4(c). Plot of $\phi(t)$ for $N = 22$.

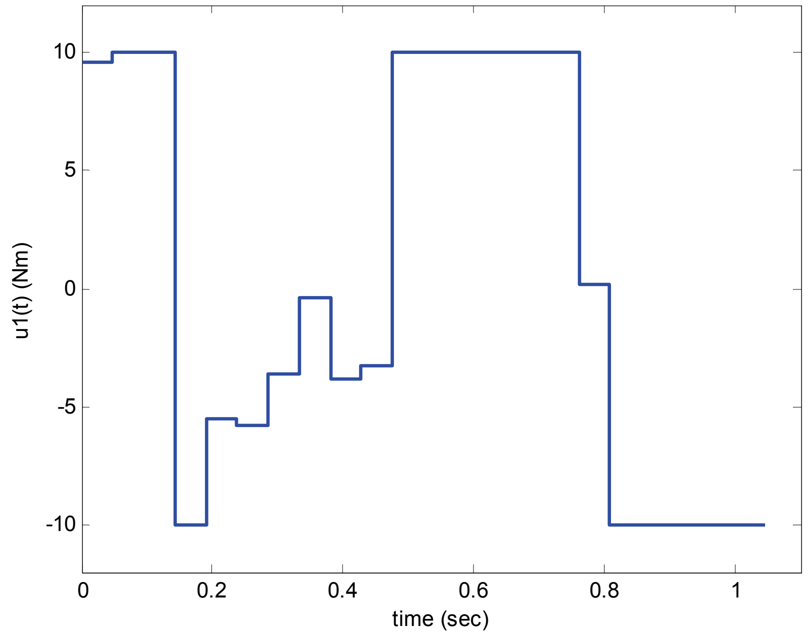

Figure 4(d). Plot of $u_1(t)$ for $N = 22$.

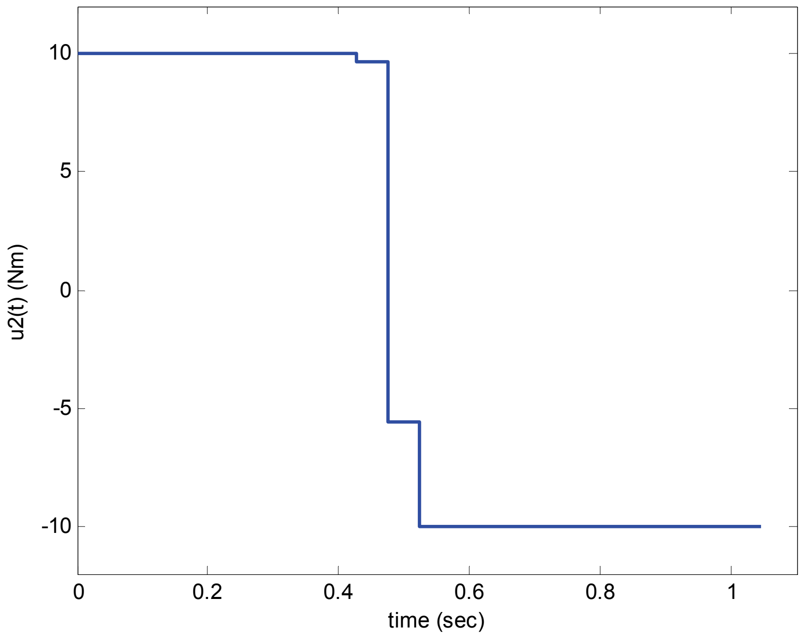

Figure 4(e). Plot of $u_2(t)$ for $N = 22$.

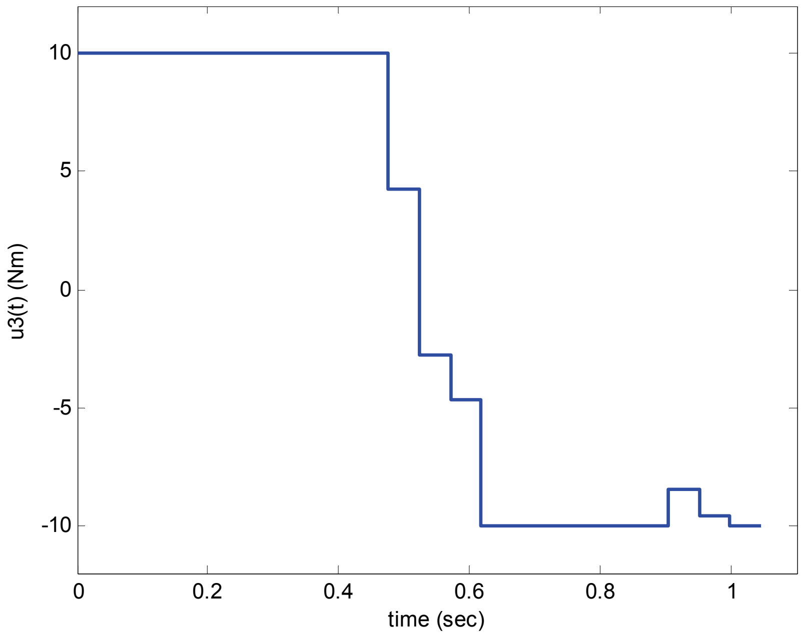

Figure 4(f). Plot of $u_3(t)$ for $N = 22$.