Trajectory tracking and obstacle avoidance for cable driven robot by Nonlinear Model Predictive Control

Halim Merabti*, Bilel Bouchemal*, Kamel Boucherma*, Khaled Belarbi**

* 1: Welding and NDT Research Centre (CSC). BP 64 CHERAGA-ALGERIA.

** Ecole polytechnique de Constantine, Campus Université de Constantine 3, Nouvelle ville Ali Mendjel Constantine Algeria kbelarbi@yahoo.com

Abstract:

In this paper, nonlinear model predictive control is proposed to solve the tracking trajectory and obstacle avoidance of the cable based robot. The application of the nonlinear model predictive control scheme in the field of robotics was limited by the computational burden associated with optimization algorithms to be solved at each sampling. The Particle Swarm Optimization algorithm is used for the solution of the optimization problem arising in the nonlinear MBPC applied for the cable robot tracking trajectory with fixed obstacle avoidance. Quality of the solution and computation time shows that NMPC-PSO can be an alternative for real time applications.

Keywords: cable robot, nonlinear model predictive control, particle swarm optimization, tracking trajectory.

1. Introduction:

Cable robots are a type of robotic manipulators which not only use rigid links in their structures but also utilize cables to deliver the desired motion. These robots are appealing because of their structural simplicity, large workspace, high stiffness, and its reconfigurability and transportability [1]. A major disadvantage of cable robots is that a cable can only exert tension. The cable robots can be used in: Manipulation, Acquisition, re-education, Integration on buildings, Assistance and relief and other applications.

A number of control schemes were applied to control the cable robots. In [2] and [3] a proportional derivative (PD) controller was implemented for tracking circular, triangular and sinusoidal trajectories. The Sliding mode control is used in [4]. Constrained linear model predictive control of the redundant cable robots were proposed in [5].

In this work, we propose the use of the nonlinear model predictive control (NMPC) scheme to solve the problem of tracking reference trajectory and obstacle avoidance in space for a cable based robot. The main drawback for the application of NMPC schemes to the control of mobile robots, in particular, was related to the computational burden associated with optimization algorithms to be solved at each sampling period. This has limited its applications to sufficient slow dynamic systems. On the other hand, a number of fast solutions methodologies for linear MBPC have appeared with improved computational [6 7]. In this work, we present an application of the Particle Swarm Optimization algorithm, PSO for the solution of the optimization problem arising in the nonlinear MBPC applied for the cable robot tracking trajectory with fixed obstacle avoidance.

This paper is organized as follows: the second section introduces the problem formulation of cable based robot tracking trajectory, the third section describe the general formulation of non linear model predictive control, the fourth section present the particle swarm optimization heuristic, in section V we give the simulation results.

2. Cable based driven robot:
This section presents the inverse and forward translational position and velocity kinematics analysis for space cable direct driven robot.

The cable length $L_i$ is the Euclidean norm between the moving point $X = \{x \ y \ z\}$ and each fixed link vertex $A_i$:

$$L_i = \sqrt{(x - A_{ix})^2 + (y - A_{iy})^2 + (z - A_{iz})^2} \quad i=1,\ldots,n$$  \(1\)

The position of the end-effector can be expressed as:

$$\{x \ y \ z\} = \{A_{ix} + L_i \cos(\theta_i)\sin(\phi_i) \ A_{iy} + L_i \sin(\theta_i)\sin(\phi_i) \ A_{iz} + L_i \cos(\phi_i)\}^T$$  \(2\)

Where $\theta$ is the polar angle and $\phi$ is the azimuthal angle.

The time derivative gives:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix} = \begin{bmatrix} \cos(\theta_i)\sin(\phi_i) & -L_i \sin(\theta_i)\sin(\phi_i) & L_i \cos(\theta_i)\cos(\phi_i) \\ \sin(\theta_i)\sin(\phi_i) & L_i \cos(\theta_i)\sin(\phi_i) & L_i \sin(\theta_i)\cos(\phi_i) \\ \cos(\phi_i) & 0 & -L_i\sin(\phi_i) \end{bmatrix} \begin{bmatrix} \dot{L}_i \\ \dot{\theta}_i \\ \dot{\phi}_i \end{bmatrix} \quad i=1,\ldots,n$$  \(3\)

The problem is to find the control law defined by $L_i, \theta_i,$ and $\phi_i$ that allows the cable based robot to track a given reference trajectory defined by $\{x_r(t) \ y_r(t) \ z_r(t)\}$ using nonlinear model based predictive control.

3. Nonlinear Model predictive control:

Consider a non linear system described by the discrete state space model:

$$x(k+1) = f(x(k), u(k))$$  \(4\)

where $x(k)$ is the state, $u(k)$ the control signal and $f$ is a continuous mapping.

The control signal $u(k)$ is such that:

$$u(k) \in \mathbb{U} \subset \mathbb{R}^m$$  \(5\)

$\mathbb{U}$ is a compact convex set with $0 \in \text{int}(\mathbb{U})$ and $f(0,0)=0$. Moreover the state may be constrained to stay into a convex and closed set:

$$x(k) \in \mathbb{X}$$  \(6\)

The problem solved by the non linear model predictive control is to regulate the state to the origin by solving the following optimization problem:

$$\min_{u} J_{N}(x,k,\mathbb{U})$$  \(7\)

With (4), (5) and (6) Where

$$J_{N}(x,k,\mathbb{U}) = F(x(k+N)) + \sum_{i=k}^{k+N-1} L(x(i), u(i))$$  \(8\)

Where $N$ is the optimization horizon. $F(x(k+N))$ is a weight on the final state. Moreover, the final state may be constrained to be in a final region.
The weight $F$ and the final region are introduced to guarantee stability of the non linear MPC.

The solution gives the control sequence up to $N$

$$U = [u(k) \ u(k+1) \ \ldots \ u(k+N-1)] \in \mathbb{R}^N$$

and only $u(k)$ is applied at sampling instant $k$. The procedure is repeated at each sampling instant.

The optimization problem (7), (8) is generally non convex. The straightforward algorithm for solving this problem is the sequential quadratic programming, SQP, an extension of the active set method used for solving quadratic program [8]. This method is difficult to code and time consuming. A number of algorithms have been proposed for solving this problem in a reasonable amount of time such that the multiple shooting method [9] and nonlinear sum of squares [10]. In this work, Particle Swarm Optimization algorithm[11], PSO, is applied to the solution of this problem.

4. Particle Swarm Optimization

The particle swarm optimization (PSO) algorithm is a population-based search algorithm, developed by Kennedy and Eberhart in 1995 [11], inspired by the social behavior of birds within a flock. The particle swarm concept originated as a simulation of a simplified social system.

In a PSO algorithm each potential solution to the problem is called a particle and the population of solutions is called a swarm. Each particle keeps track of its coordinates in the problem space which are associated with the best solution, $p_{best}$, (fitness) it has achieved so far. Another best value that is tracked by the global version of the particle swarm optimizer is the overall best value, and its location, obtained so far by any particle in the population. This location is called $g_{best}$. At each step, the PSO concept consists of changing the velocity of each particle toward its $p_{best}$ and $g_{best}$ locations. The velocity is weighted by a random term, with separate random numbers being generated for acceleration toward $p_{best}$ and $g_{best}$ locations.

The original process for implementing the global version of PSO is as follows:

1- Initialize a population of particles with random positions and velocities in the problem space
2- For each particle, evaluate the desired optimization fitness function.
3- Compare particle’s fitness evaluation with particle’s $p_{best}$. If current value is better than $p_{best}$, then set $p_{best}$ value equal to the current value and $p_{best}$ location equal to the current location.
4- Compare fitness evaluation with the population’s overall previous best. If current value is better than $g_{best}$, then reset $g_{best}$ to the current particle’s array index value.
5- Change the velocity and position of the particle according to the equations (8) and (9), respectively:

$$v_{k+1}^j = v_k^j + c_1 r_1 (p_{k}^j - x_k^j) + c_2 r_2 (p_{k}^g - x_k^j)$$

$$x_{k+1}^j = x_k^j + v_{k+1}^j$$

6- Loop to step 2 until a criterion is met.

Where: $x_k^j$ : Particle position, $v_k^j$ : Particle velocity, $p_{k}^j$ : Local best position

$p_{k}^g$ : Global best position, $-c_1, c_2$ : Constants, $-r_1, r_2$ : Random numbers between 0 and 1.

5. Simulation results:

We consider two mobile robots, one real and one virtual, the real robot track the virtual one. The problem is to find the control law defined by $L_i, \hat{\theta}_i, \text{and} \phi_i, i = 1, ..., 8,$ that allows the cable based robot to track a given reference trajectory. In this work, only $L_1, \hat{\theta}_1,$ and $\phi_1$ are defined solving the optimisation problem described by (7) and (8). The other cables lengths are defined using (1).

Simulation parameters are: iterations $=25$; $c_1=c_2=1.2$; swarm size $=25$; $Nu=2$; $Np=7$; links vertexes are: A1(0; 0; 2); A2(2; 0; 2); A3(2; 2; 2); A4(0; 2; 2); A5(0; 0; 0); A6(2; 0; 0);
A7(2; 2; 0); A8(0; 2; 0). All computations are achieved on Intel® Core i3- 3.3 GHz with 4Go RAM. The Cartesian velocities are constrained to:

\[-0.17 \text{ (m/s)} \leq v_x \leq 0.17 \text{ (m/s)}\]
\[-0.17 \text{ (m/s)} \leq v_y \leq 0.17 \text{ (m/s)}\]
\[-0.17 \text{ (m/s)} \leq v_z \leq 0.17 \text{ (m/s)}\]

Constraints satisfaction is ensured by adding a penalty to the cost function. The reference trajectory (position of the virtual robot to track) is given by:

\[x_r(t) = 1 + \cos(\omega_0 t) ; y_r(t) = 1 + \sin(\omega_0 t) ; z_r(t) = 1 + \cos(2\omega_0 t) ; \omega_0 \approx 0.02 \text{rad/s}\]

a. First scenario: tracking trajectory:

![Fig.1 (a) cable robot trajectory](image1.png)

![Fig.1 (b) Eight cable lengths](image2.png)

![Fig.1 (c) Cartesians velocities](image3.png)
The results of the first scenario are shown in figure 2(a, b, c, d). The presented simulation results demonstrate the good tracking of the reference trajectory. Variations in cables lengths are very smooth which allow a good controlling of pulleys after the conversion of the cable length to angle rotations. The constraints on the Cartesian velocities are always satisfied as shown in figure 2 (c).

b. Second scenario: tracking trajectory and obstacles avoidance:

Fig2. (a) Cable robot trajectory for the tracking trajectory and obstacle avoidance
The results of this scenario are shown in figures 3(a, b, c, d). It can be seen that the cable based robot track the reference trajectory and ovoid all obstacles. The obstacle is avoided by adding a penalty to the cost function when the distance between the robot and the obstacle is less than a safe distance defined previously. The constraints on the Cartesian velocities are always satisfied as shown in figure 3 (c). The computation time is less than 5ms which is very encouraging for real time applications.

6. Conclusion:

In this work, a cable robot was controlled to track a reference trajectory and avoid collision with fixed obstacles. The nonlinear MPBC scheme was applied to control this system and the particle swarm optimization algorithm was used for the solution of the optimization problem arising in the non linear MBPC. Two scenarios was proposed: the first is a simple tracking and the second is a tracking and avoidance fixed obstacles. Obtained results are encouraged for real time applications.

References:


