Single and Multi Objective Predictive Control of Mobile Robots

H. Merabti[1(✉)], K. Belarbi[2], and I. Bouchachi[1]

[1] Research Center in Industrial Technologies Crti, Cheraga, P.O.Box 64, 16014 Algiers, Algeria
h.merabti@csc.dz, fingroo@yahoo.fr

[2] Ecole Nationale Polytechnique de Constantine Campus, Constantine 3, Nouvelle ville Ali Mendjeli, Constantine, Algeria
kbelarbi@yahoo.com

Abstract. In this work, we present a comparison between the use of a simple and multi objective MBPC in robots control for tracking trajectories and obstacle avoidance. Two cases were considered, in the first each robot has its own MPC controller where in the second a single two- objectives MPC controller is used for both robots. In the second case; two approaches were proposed to solve the multi objective optimization problem arising in the MOMPC: the multi objective Particle Swarm Optimization (MOPSO) and weighted sum method. The simulation results show that the robots movement is more stable by the MOPSO-NMPC than the PSO-NMPC. Computation times as expected are shorter PSO-NMPC; however MOPSO-NMPC although more time consuming is still feasible.

Keywords: Model predictive control · Metaheuristics · Multiobjective optimization

1 Introduction

Model based predictive control (MBPC) is based on the use of a model for predicting the future behavior of the system over a finite future horizon. The current control action is obtained by solving on-line, at each sampling instant, a finite horizon optimal control problem, using the current state of the plant as the initial state [1]. The optimization yields an optimal control sequence and the first control in this sequence is applied to the plant. The solution of the optimization problem depends on the nature of the model and constraints.

Multi objective model based predictive control (MOMBPC) has been proposed by a number of authors with improved performance. For example, in [2], the authors use multi objective optimization to tune nonlinear model predictive controllers based on a weighted sum objective function and in [3] authors shown that it is possible to compute a Pareto optimal solution as an explicit piecewise affine function after recasting the optimization problem associated with the multi objective MPC as a multi parametric multi objective linear or quadratic program.

M. Chadli et al. (eds.), *Recent Advances in Electrical Engineering and Control Applications*,
Lecture Notes in Electrical Engineering 411, DOI 10.1007/978-3-319-48929-2_6

In this work, we present a comparison between the use of a simple and multi objective MBPC in robots control for tracking trajectories and obstacle avoidance. Each robot tracks a specified reference trajectory. In the first case, each robot has its own MPC controller and the Particle Swarm Optimization algorithm PSO is used for the solution of the optimization problem arising in the MPC. In the second case, a single two- objectives MPC controller is used for both robots. This last problem was solved using two approaches: the first is the use of multi-objectives PSO for the solution of the multi objective optimization problem arising in MOMPC, the second is to transform the multi objective optimization problem to simple optimization one using the weighted sum method.

The paper is organized as follows: Sect. 2 gives the formulation of simple and Multi objective nonlinear model predictive control, Sect. 3 provides the description of the MOPSO and the weighted sum method, Sect. 4 simulation results are given.

2 Model Predictive Control

Consider a nonlinear system described by the discrete state space model:

$$x(k+1) = f(x(k), u(k)) \tag{1}$$

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

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

$$u(k) \in \mathbb{U} \subset \mathbb{R}^m \tag{2}$$

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

$$x(k) \in \mathbb{X} \tag{3}$$

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

$$\min_{\mathcal{U}} J_N(x, k, \mathcal{U}) \tag{4}$$

With (1), (2) and (3) Where

$$J_N(x, k, \mathcal{U}) = F(x(k+N)) + \sum_{i=k}^{k+N-1} L(x(i), u(i)) \tag{5}$$

Where N is the optimization horizon. *F(x(k + N))* is a weight of the final state. Moreover, the final state may be constrained to be in a final region:

$$x(k+N) \in X_f \subset \mathbb{X} \tag{6}$$

The weight F and the final region are introduced to guarantee stability of the nonlinear MPC.

The solution gives the control sequence up to N

$$\mathcal{U} = \left[u(k) \quad u(k+1) \quad \cdots \quad u(k+N-1) \right] \in \mathbb{U}^N$$

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

The optimization problem (5), (6) 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 [4]. 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 [5] and nonlinear sum of squares [6, 7]. In this work, Particle Swarm Optimization algorithm [8], PSO, is applied to the solution of this problem.

The problem of multi objective model predictive control is to minimize, at each sampling time, the l follows functions cost:

$$\begin{aligned} J_i(U, x) = {} & F_i(x(k+N)) \\ & + \sum\nolimits_{k=j}^{k+N-1} L_i(x(j), u(j))\, with\, i = 1, \cdots, l \end{aligned} \tag{7}$$

$F_i(x(k+N))$ is a weight on the final state. Moreover, the final state may be constrained to be in a final region $(k+N) \in X_f \subset X$. The weight F_i and the final region are introduced to guarantee stability of the nonlinear MPC.

The solution gives the set of Pareto front and only one Pareto optimal solution is selected and applied at sampling instant k. The procedure is repeated at each sampling time. In this work we use Multi Objective Particle Swarm Optimization, MOPSO [9], to generate a set of approximately Pareto-optimal solutions in a single run and we use a weighted sum approach to convert the multi objective MPC to a single one.

3 Solution of the Multi Objective Predictive Control Problem

1. The MOPSO

Particle swarm optimization is an evolutionary computation technique developed by Kennedy and Eberhart in 1995 [8]. The particle swarm concept originated as a simulation of a simplified social system. The success of the particle swarm optimization algorithm motivated the researchers to apply it to multi-objective optimization problems. The MOPSO [9] is one of the algorithms proposed to solve the multi objective optimization problem using particle swarm optimization algorithm. The MOPSO maintains two archives, one for storing the global non-dominated solutions and the other for storing the individual best solutions attained by each particle. Basically, the updating of the particle is performed as follows:

$$V(t+1) = w * V(t) + R_1 * (p_{best}(t) - p(t)) + R_2 * (Rep(h) - p(t)) \quad (8)$$

$$p(t+1) = p(t) + V(t+1) \quad (9)$$

Where V is the particle velocity, *p(t)* is the current position of the particle, w is a constant, R_1 and R_2 are random numbers in [0 1]. *REP* is a repository where are stored the positions of the non dominated particles and h is an index in the repository that is introduced to ensure some fitness sharing [10].

REP is updated by inserting the currently non dominated positions and dominated positions are eliminated. The size of the repository being limited when it becomes full and particles in less populated areas are given priority over those highly populated regions.

2. Weighted sum method for multi-objective optimization

This method is the simplest and widely used classical method. It allows the transformation of the objective functions vector into a single-objective function. The single criterion is obtained by the sum of the weighted criteria as follows:

$$\min \sum_{i=1}^{k} w_i J_i(x) \text{ With } w_i \geq 0 \quad (10)$$

w_i: is the affected weight to the objective i where:

$$\sum_{i=1}^{k} w_i = 1$$

Weights Value depends on the relative importance of each objective. In this work, objectives have the same weight.

4 Application

We consider four mobile robots, two real and two virtual, the real robots track the virtual ones. It is assumed that there is a pure rolling. The contact between the wheels and the ground is supposed to be frictionless. The kinematic model of the real robots is given by:

$$\dot{x}_i(t) = \frac{v_{ri}(t) + v_{li}(t)}{2} \cos\theta_i(t) \quad (11)$$

$$\dot{y}_i(t) = \frac{v_{ri}(t) + v_{li}(t)}{2} \sin\theta_i(t) \quad (12)$$

$$\dot{\theta}_i(t) = \frac{v_{ri}(t) - v_{li}(t)}{b} \quad (13)$$

$$\omega_i = (v_{ri} - v_{li}/b) \quad (14)$$

Where i = *1, 2*, $v_{ri} \in R$ and $v_{li} \in R$ are the right and the left wheels linear velocities of the real robot *i*, $b \in R$ is the distance between the wheel centers. θ_i are the robots orientation and ω_i are the angular velocities. The objective is to find a control law defined by $v_{ri}(t)$, $v_{li}(t)$ (i = *1, 2*) that allows the robots to:

- track a given reference trajectories defined by:$[x_{ri}(t) y_{ri}(t)], i = 1, 2$, respectively
- Avoid fixed obstacles on the trajectories
- Avoid collisions

This problem is set as a single objective MPCs and multi-objective MPC with constraints. The optimization problem arising in the MPCs is solved by the particle swarm optimization meta-heuristic, PSO. The optimization problem arising in MOMPC is solved using two different approaches. The first consists in converting the multi objective optimisation problem to single one where the resulting objective function is a weighted sum of the two objective functions. The second approach consists of using the multi objective particle swarm optimisation algorithm (MOPSO) to generate a set of optimal Pareto solutions.

The first reference trajectory is given by:

$\boldsymbol{x_{r1}(t) = cos(\omega_0 t)}$; $\boldsymbol{y_{r1}(t) = sin}(2 * \boldsymbol{\omega_0 t})$; $\omega_0 = 0.02$ *rad/s* is the signal pulsation, the second reference trajectory is given by: $\boldsymbol{x_{r2}(t) = cos(\omega_0 t + \varphi)}$; $\boldsymbol{y_{r2}(t) = sin(2 * \omega_0 t + \varphi)}$;

The control signals are constrained to:

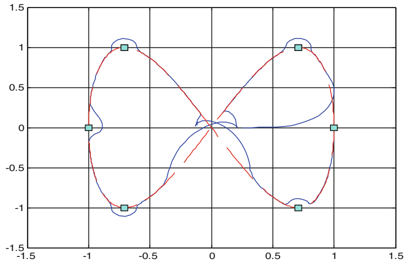
$$\boldsymbol{-0.7 \leq v_{ri} \leq 0.7 - 0.7 \leq v_{li} \leq 0.7} \quad (15)$$

and the angular velocity is constrained to:

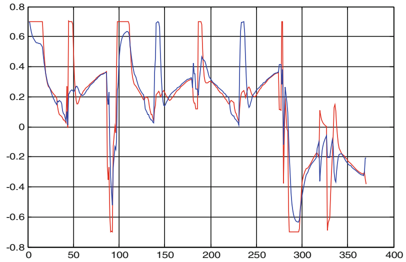
$$\boldsymbol{-5 \leq \omega_i \leq 5} \quad (16)$$

These algorithms are run until a satisfactory response is obtained. The collision point of the robots is m(0,0). The robots start from their initial positions (0.25, 0), (−0.75, 0) and track their own trajectories. It is observed that both have good tracking and avoid all fixed obstacles on their trajectories. However, the behavior of the robots in the collision point is different for each algorithm where:

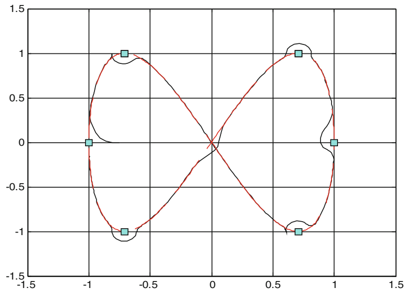
- For the first and the second algorithms (PSO-NMPCs and MOPSO-NMPC), it is observed that the second robot continues its tracking while the first avoids it by decreasing its velocity to keep a safe distance. Then, they continue their tracking and avoid the obstacles encountered. One can also observe that their movement is more stable by the MOPSO-NMPC than the PSO-NMPC (Fig. 1).
- For the last algorithm, the WS-MOMPC, it is observed that each robot avoids the collision with the other. Then, they continue their tracking and avoid fixed obstacles encountered. We also observe a dynamic movement of the robots.



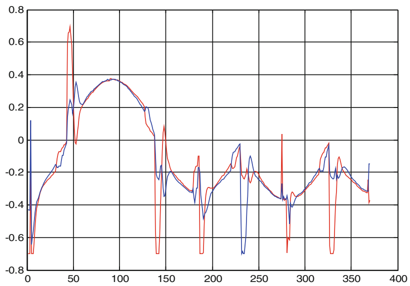
(a) Robot 1 trajectory



(b) Robot1 velocities

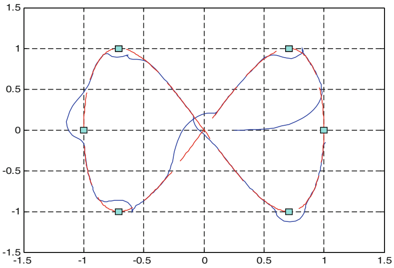


(c) Robot2 trajectory

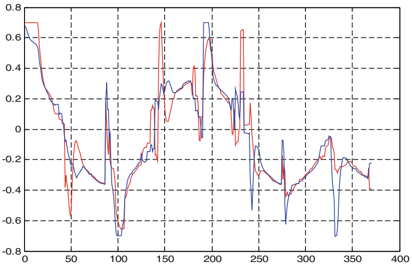


(d) Robot2 velocities

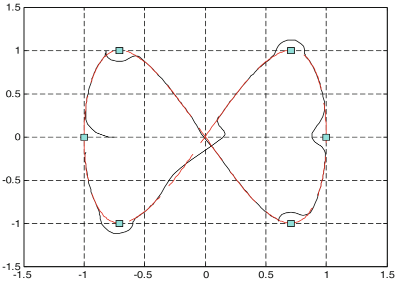
Fig. 1. Results with Two PSO-NMPC controllers. (a) Robot 1 trajectory (b) Robot1 velocities (c) Robot2 trajectory (d) Robot2 velocities



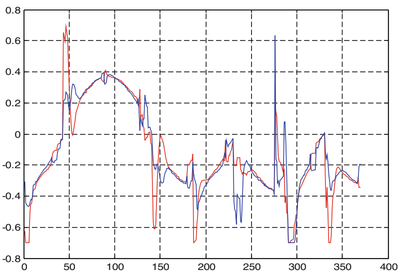
(a) Robot1 trajectory



(b) Robot1 velocities

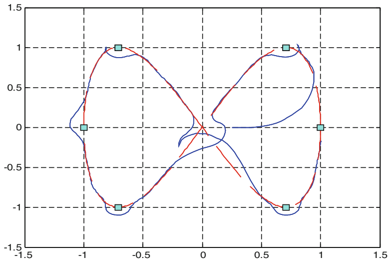


(c) The second robot trajectory

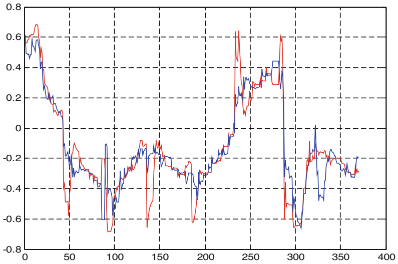


(d) The second robot velocities

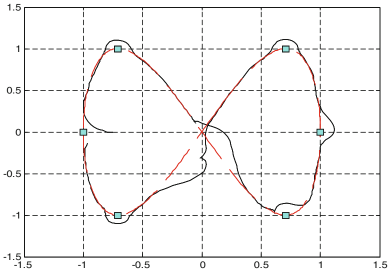
Fig. 2. Results with the Pareto: MOPSO-NMPC (a) Robot1 trajectory (b) Robot1 velocities (c) The second robot trajectory (d) The second robot velocities



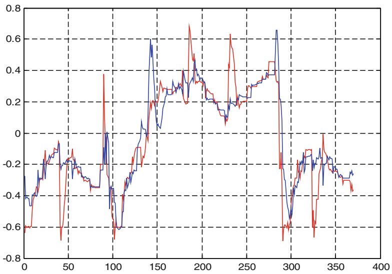
(a) Robot1 trajectory



(b) Robot1 velocities



(c) Robot2 trajectory



(d) Robot2 velocities

Fig. 3. Results with WS-MOMPC (a)Robot1 trajectory (b)Robot1 velocities. (c) Robot2 trajectory (d) Robot2 velocities

Table 1. COMPUTATION TIMES

	PSO-NMPC	MOPSO-NMPC	WS-MOMPC
Computation time	$\leq$ 4 ms	$\leq$ 11 ms	$\leq$ 12 ms

Computation times are given in Table 1, where it can be seen that PSO-NMPC outperforms the other algorithms. The constraints on the control signals are always satisfied as shown in Figs. 2 and 3.

5 Conclusion

In this work, we have developed a comparison between single objective and multi objective MBPC in robots control for tracking trajectories and obstacle avoidance. Each robot tracks a specified reference trajectory. In the first case, each robot has its own MPC controller and the Particle Swarm Optimization algorithm PSO is used for the solution of the optimization problem arising in the MPC. In the second case, a single two- objectives MPC controller is used for both robots. This last problem was solved using two approaches: the first is the use of multi-objectives PSO for the solution of the multi objective optimization problem arising in MOMPC, the second is to transform the multi objective optimization problem into a single objective optimization using the weighted sum method. The simulation results show that the robots movement is more stable by the MOPSO-NMPC than the PSO-NMPC. Computation times as expected are shorter PSO-NMPC, however MOPSO-NMPC although more time consuming, is still feasible.

References

1. Maciejowski, J.M.: Predictive Control with Constraints. Prentice Hall, Upper Saddle River (2001)
2. De Vito, D., Scattolini, R.: A receding horizon approach to the multiobjective. In: Proceedings of the 46th IEEE Conference on Decision and Control New Orleans, LA, USA, 12–14 December 2007
3. Bemporad, A., de la Peña, D.M.: Multiobjective model predictive control. Automatica **45**, 2823–2830 (2009)
4. Biegler, L.T., Rawlings, J.B.: Optimization approaches to nonlinear model predictive control. In: Arkun, Y., Ray, W.H. (eds.) CPC IV, The CACHE Corp., Austin, Texas (1991)
5. Diehl, M., Bock, H., Leineweber, D., Schlöder, J.: Efficient direct multiple shooting in nonlinear model predictive control. In: Keil, F., Mackens, W., Vos, H., Werther, J. (eds.) Scientific Computing in Chemical Engineering II, vol. 2. Springer, Berlin (1999)
6. Casavola, A., Famularo, D., Franzè, G.: Predictive control of constrained nonlinear systems via LPV linear embeddings. Int. J. Robust Nonlinear Control **13**(3–4), 281–294 (2003)
7. Franzè, G.: A nonlinear sum-of-squares model predictive control approach. IEEE Trans. Autom. Control **55**(6), 1466–1471 (2010)

8. Eberhart, R.C., Shi, Y.: Particle swarm optimization: developments, applications and resources. In: Proceedings of the Congress on Evolutionary Computation 2001, Seoul, Korea. Piscataway, NJ. IEEE Service Center (2001)(b)
9. Coello, C., Lechuga, M.: MOPSO: a proposal for multiple objective particle swarm optimization. In: Proceedings of the Congress on Evolutionary Computation, part of the 2002. IEEE World Congress on Computational Intelligence, Hawaii, pp. 1051–1056. IEEE (2002)
10. Li, X.: A non-dominated sorting particle swarm optimizer for multiobjective optimization. In: Cantú-Paz, E., et al. (eds.) GECCO 2003. LNCS, vol. 2723, pp. 37–48. Springer, Heidelberg (2003). doi:10.1007/3-540-45105-6_4