Advanced Robot Control
Model Predictive Control

Wojciech Domski

Chair of Cybernetics and Robotics,
Wrocław University of Science and Technology

Presentation compiled for taking notes during lecture

Wrocław University
of Science and Technology

Outline

Wrocław University
of Science and Technology

PID (1/2)

Before we dive into MPC let us take a moment and consider a well know control strategy – PID.

PID (2/2)

PID controller formula

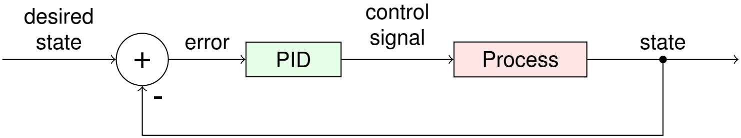
$$u = K_p e(t) + K_i \int_0^t e(\tau)\, d\tau + K_d \dot{e}(t) \qquad (1)$$

where $e = x - x_d$, $\dot{e} = \dot{x} - \dot{x}_d$.

PID control algorithm is based on error value, thus it can be seen as reactive control [1]. It does not relay on a model – it is model-less.

Finally, PID is a SISO algorithm. It has a single input to the system (e, error) and a single output (u, control signal).

Process overview



Model Predictive Control (1/1)

What is Model Predictive Control?

Model Predictive Control (1/4)

Model Predictive Control characterizes with following:

- is based on a process model,
- uses numerical optimization,
- is considered a MIMO system.

Model Predictive Control (2/4)

Model based

During the calculations a mathematical representation of a model is used to calculate control signals.

Numerical optimization

To calculate control signal or trajectory process called numerical optimization is considered. The purpose of this actions is to find the optimal control signals based on model reaction and its state evolution.

MIMO system

A (Multiple Inputs Multiple Outputs) MIMO system is a complex system which consumes multiple inputs and produces multiple outputs during a single process step.

Model Predictive Control (3/4)

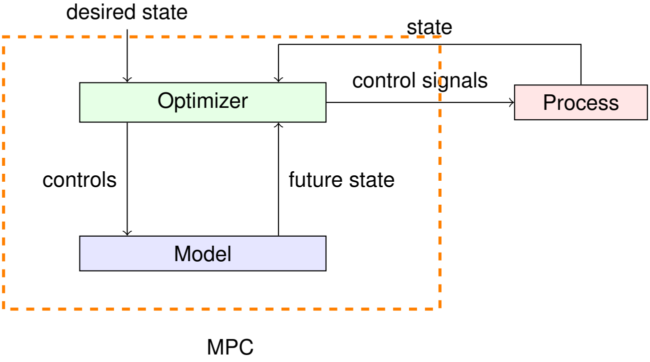
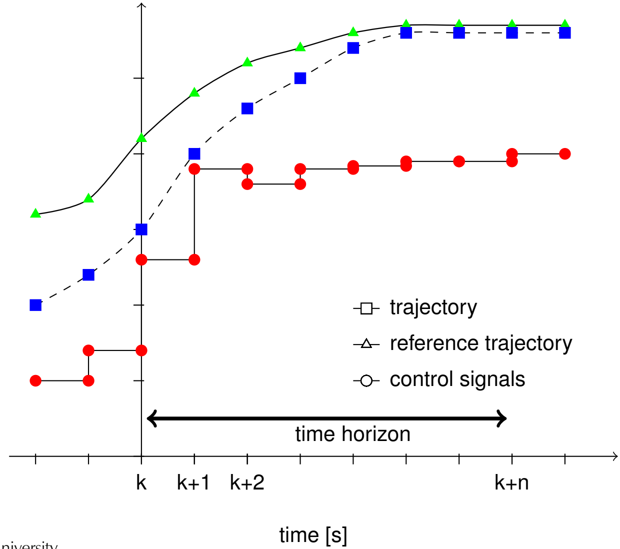


Figure: MPC signal flows

Model Predictive Control (4/4)

Model Predictive Control is considered a composition of two main subsystems – optimizer and model [2].

1. Desired state is fed to the optimizer.
2. Through a series of numerical calculations where a model is involved, a optimal control signal is derived.
3. Control signal is being calculated on a time horizon.
4. After optimization process is done only a control signal for next step is taken, the rest is discarded.
5. The control signal is passed to the control object.
6. The algorithm repeats from step **1** while increasing time, thus moving time horizon forward.


trajectory
reference trajectory
control signals
time horizon
k
k+1
k+2
k+n
time [s]

Two approaches to MPC

MPC could be applied in two different ways.

Direct control

As shown in the Fig. 1 MPC could be directly used to control an object.

Indirect control

MPC is used to generate control signals as long as we reach the goal (or local minimum). In this process the current state of an object is not updated from process but from model. Here, the output is the bypass product of the numerical optimization – a trajectory.

PID vs MPC

PID	MPC
reactive	model based
SISO	MIMO
error defines control signal	solution is derived through optimization
computationally stable	might be computationally unstable

Cost function (1/3)

MPC uses an optimizer in order to find the best control signals. However, to perform this operation a criteria has to be defined.

Cost function

The cost function allows to estimate the quality of a solution. Therefore, in each iteration of optimization routine the value of this function is examined.

Since the minimum value is searched optimizer produces different signals in order to find the best control signals which allow to achieve the smallest value of the cost function.

Cost function (2/3)

$$J(u) = \sum_{i=0}^{n} \left(f(x_i, x_g) + f(u_i) + f(x_i, \Phi) + g(x_i, u_i, \Phi) \right) \qquad (2)$$

where k is given control iteration, n is number of step across some time horizon. $f()$ are functions which define if a given goal was reached. These functions should be decreasing and have a single minimum (preferably).

Cost function (3/3)

Let us consider task where a robot needs to reach a goal pose.

- $f(x_i, x_g)$ represents distance to the goal, both e.g. euclidean distance and angular deviation,
- $f(u_i)$ energy consumption,
- $f(x_i, \Phi)$ distance to obstacles Φ,
- $g(x_i, u_i, \Phi)$ other terms to optimize.

Minimization problem

For given $J(u)$ cost function we propose a minimization problem

$$\min_u J(u), \tag{3}$$

subject to:

$$\begin{aligned} c_j(u) \geq 0, \quad & j \in \mathrm{I}, \\ lb_i \leq u_i \leq ub_i, \quad & i = 1, \dots, N. \end{aligned} \tag{4}$$

$c_j(u)$ represents constraints imposed on cost function $J(u)$ while I is considered to be set of inequality constraints. *lb* and *ub* define lower and upper bound respectively while N is the dimension of the problem, thus size of the control vector u.

Hard and Soft constraints (1/3)

The optimization problem (3) can be constraint in two different ways – bounds and constraints.

Bounds

Bounds are used to assume strict range for the control vector u which is subject to minimization (3). Bounds cannot be broken and if they are the solution is infeasible. Moreover, bound can be used to limit maximum and minimum control signals in the problem of trajectory planning with MPC.

Hard and Soft constraints (2/3)

The second group includes hard and soft constraints.

Hard constraints

Similarly to bounds, hard constraints $c_j(u)$ cannot be broken. They can be used to limit solution space in order to only look for a feasible solution.
Unless hard constraints $c_j(u)$ are broken the solution is feasible. Furthermore, they can be used to further constrain solution space. For example, they can be used to avoid obstacles in trajectory generating task.

Hard and Soft constraints (3/3)

Soft constraints

Soft constrains are expressed as $g()$ in cost function (2). Unlike hard constraints soft constraints can be violated or broken since they are part of cost function $J(u)$ which is being optimized. As an example obstacle avoidance can be given. However, it is important to introduce other techniques like safety zones for obstacles in order to give a margin for optimization.

Unicycle model (1/3)

Kinematic model for unicycle is given as

$$\begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{pmatrix} = \begin{bmatrix} \cos\theta & 0 \\ \sin\theta & 0 \\ 0 & 1 \end{bmatrix} \begin{pmatrix} v \\ \omega \end{pmatrix} \tag{5}$$

where v and ω are linear velocity and angular velocity, respectively.

Unicycle model (2/3)

An alternative unicycle model can be expressed as

$$\begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \\ \dot{\Phi}_1 \\ \dot{\Phi}_2 \end{pmatrix} = \begin{bmatrix} \cos\theta & \cos\theta \\ \sin\theta & \sin\theta \\ \frac{1}{L} & -\frac{1}{L} \\ \frac{2}{R} & 0 \\ 0 & \frac{2}{R} \end{bmatrix} \begin{pmatrix} \eta_1 \\ \eta_2 \end{pmatrix} \tag{6}$$

where $\eta_1 = \frac{R}{2}\dot{\Phi}_1$ and $\eta_2 = \frac{R}{2}\dot{\Phi}_2$ are scaled wheels' speeds. R is wheel radius and L is half distance between wheels.

Unicycle model (3/3)

Linear and angular velocities can be expressed using η as

$$v = \eta_1 + \eta_2, \tag{7}$$

$$\omega = \frac{1}{L}(\eta_1 - \eta_2) \tag{8}$$

or using $\dot{\Phi}$ as

$$v = \frac{R}{2}\left(\dot{\Phi}_1 + \dot{\Phi}_2\right), \tag{9}$$

$$\omega = \frac{R}{2L}\left(\dot{\Phi}_1 - \dot{\Phi}_2\right). \tag{10}$$

Ackerman model (1/1)

Kinematic for Ackerman model is given as

$$\begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{pmatrix} = \begin{pmatrix} u_1 \cos\theta \\ u_1 \sin\theta \\ \frac{u_1}{l} \tan u_2 \end{pmatrix} \tag{11}$$

where u_1 and u_2 are vehicle speed and steering angle. l is longitudinal distance between wheels.

Minimization solver (1/2)

SciPy, Python package, provides multiplicity of different optimization solvers, however it is important to consider only these which provide inequality constraints and bounds.

One of such is Sequential Least SQuares Programming (**SLSQP**) Algorithm.

Minimization solver (2/2)

SLSQP allows to solve following minimization problem

$$\min_{x} f(x), \tag{12}$$

subject to:

$$\begin{aligned} c_j(x) &= 0, & j &\in \mathrm{E}, \\ c_j(x) &\geq 0, & j &\in \mathrm{I}, \\ lb_i \leq x_i &\leq ub_i, & i &= 1, \ldots, N. \end{aligned} \tag{13}$$

Where E is a set of equality constraints, while I is a set of inequality constraints.

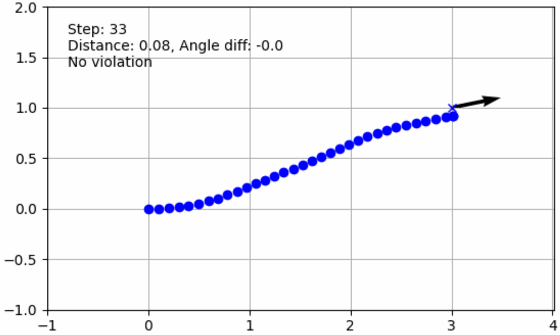


Figure: Unicycle model with no obstacles

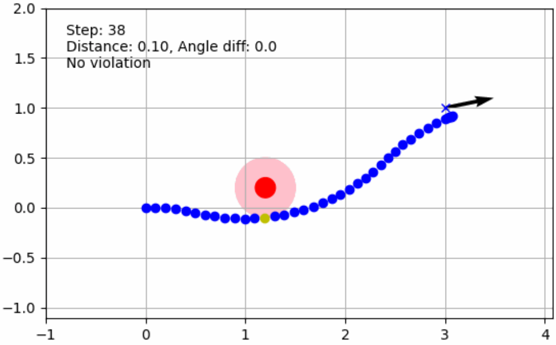


Figure: Unicycle model with no obstacles

Wrocław University
of Science and Technology

Quiz (1/1)

Calculate group number as the rest from dividing the Student ID number by 4.

Example

Student ID number is 123456, thus the group is 0.
Take last 2 digits from Student ID number (56) and calculate the rest from dividing by 4 (56 % 4 = 0).

Write down your name, Student ID number and group.

Literature (1/1)

- K. J. Åström and T. Hägglund.
  PID Controllers: Theory, Design and Tuning.
  Instrument Society of America, 2 edition.
- M. Ławryńczuk.
  Nonlinear Predictive Control Using Wiener Models.
  Springer Cham, 2022.