Driver assistance system design A

Nonlinear model predictive control

Carlo Novara

Politecnico di Torino
Dip. Elettronica e Telecomunicazioni

Outline

Introduction

- Nonlinear model predictive control (NMPC) is a general and flexible approach to nonlinear system control.
- **Approach.** At each time step:
  - A prediction over a given time horizon is performed, using a model of the plant.
  - The command input is chosen as the one yielding the "best" prediction (i.e., the prediction closest to the desired behavior) by means of some on-line optimization algorithm.
- NMPC allows us to deal with input/state/output constraints and to manage systematically the trade-off performance/command effort.
- NMPC is a nonlinear finite-horizon version of LQR.
- Applications: automotive systems, aerospace systems, chemical processes, robotics, biomedical devices, etc.

Nonlinear model predictive control

- Consider the MIMO nonlinear system

$$\begin{aligned}\dot{x} &= f(x,u)\\ y &= h(x,u)\end{aligned} \tag{1}$$

where $x \in \mathbb{R}^n$ is the state, $u \in \mathbb{R}^{n_u}$ is the command input and $y \in \mathbb{R}^{n_y}$ is the output.

- The generalization to time-varying systems is straightforward.
- Suppose that the state is measured in real-time, with a sampling time T_s. The measurements are

$$x(t_k), \quad t_k = T_s k, \; k = 0, 1, \ldots$$

- If the state is not measured, an observer has to be employed or a model in input-output form.
- NMPC is based on two key operations: <u>prediction</u> and <u>optimization</u>.

Nonlinear model predictive control

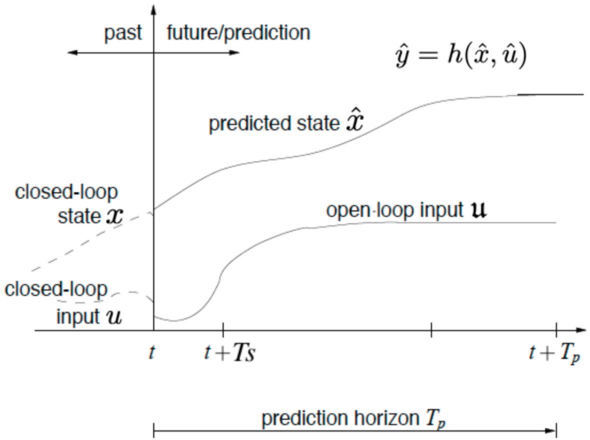
- At each time $t = t_k$, the system state and output are predicted over the time interval $[t, t + T_p]$.
  - The prediction is obtained by integration of (1) (or a model of it).
  - $T_p \geq T_s$ is called the *prediction horizon*.
- At any time $\tau \in [t, t + T_p]$, the predicted output $\hat{y}(\tau)$ is a function of the "initial" state $x(t)$ and the input signal:

$$\hat{y}(\tau) \equiv \hat{y}(x(t), \mathfrak{u}(t : \tau))$$

where $\mathfrak{u}(t : \tau)$ denotes a generic input signal in the interval $[t, \tau]$.

Nonlinear model predictive control

- In the time interval $[t, t + T_p]$, $\mathfrak{u}(\tau)$ is an open-loop input, in the sense that it does not depend on $x(\tau)$.



Nonlinear model predictive control

- At each time $t = t_k$, we look for an input signal $\mathfrak{u}(t:\tau) = u^*(t:\tau)$, such that the prediction

$$\hat{y}\left(x(t), u^*(t:\tau)\right) \equiv \hat{y}\left(u^*(t:\tau)\right)$$

  has the desired behavior for $\tau \in [t, t+T_p]$.

- The concept of desired behavior is formalized by defining the *objective function*

$$J\left(\mathfrak{u}(t:t+T_p)\right) \doteq \int_t^{t+T_p} \left(\|\tilde{y}_p(\tau)\|_Q^2 + \|\mathfrak{u}(\tau)\|_R^2\right) d\tau + \|\tilde{y}_p(t+T_p)\|_P^2$$

  where $\tilde{y}_p(\tau) \doteq r(\tau) - \hat{y}(\tau)$ is the predicted tracking error, $r(\tau) \in \mathbb{R}^{n_y}$ is a reference to track, $\|\cdot\|_X$ are weighted vector norms and their integrals are square signal norms.

- The input signal $u^*(t:t+T_p)$ is chosen as one minimizing the objective function $J\left(\mathfrak{u}(t:t+T_p)\right)$.

Nonlinear model predictive control

- Objective function:

$$J\left(\mathfrak{u}(t:t+T_p)\right) \doteq \int_t^{t+T_p} \left(\|\tilde{y}_p(\tau)\|_Q^2 + \|\mathfrak{u}(\tau)\|_R^2 \right) d\tau + \|\tilde{y}_p(t+T_p)\|_P^2 \,.$$

- The goal is to minimize, at each time t_k, the tracking error square norm $\|\tilde{y}_p(\tau)\|_Q^2 = \|r(\tau) - \hat{y}(\tau)\|_Q^2$ over a finite time interval.
- $\|\tilde{y}_p(t+T_p)\|_P^2$ gives further importance to the final tracking error.
- $\|\mathfrak{u}(\tau)\|_R^2$ allows us to menage the trade-off between performance and command activity.
- The square weighted norm of a vector $v \in \mathbb{R}^n$ is

$$\|v\|_Q^2 \doteq v^T Q v = \sum_{i=1}^{n} q_i\, v_i^2,\ Q = \mathrm{diag}(q_1, \ldots, q_n) \in \mathbb{R}^{n \times n},\ q_i \geq 0.$$

Nonlinear model predictive control

- Objective function:

$$J\left(\mathfrak{u}(t:t+T_p)\right) \doteq \int_t^{t+T_p} \left(\|\tilde{y}_p(\tau)\|_Q^2 + \|\mathfrak{u}(\tau)\|_R^2\right) d\tau + \|\tilde{y}_p(t+T_p)\|_P^2 \, .$$

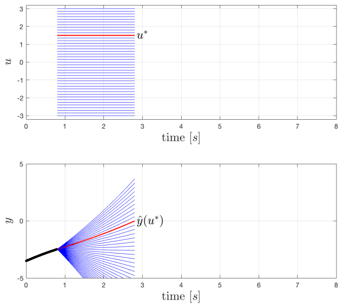
- The tracking error $\tilde{y}_p(\tau) \doteq r(\tau) - \hat{y}(\tau)$ depends on $\hat{y}(\tau)$, which is obtained by integration of (1).
- Minimization of J is thus subject to the constraints

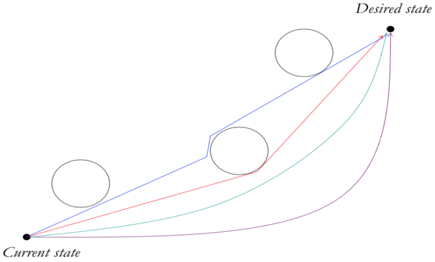
$$\begin{aligned} &\dot{\hat{x}}(\tau) = f(\hat{x}(\tau), \mathfrak{u}(\tau)),\ \hat{x}(t) = x(t),\ \tau \in [t, t+T_p] \\ &\hat{y}(\tau) = h(\hat{x}(\tau), \mathfrak{u}(\tau)). \end{aligned}$$

- Other constraints may be present on
  - the predicted state/output: $\hat{x}(\tau) \in X_c$, $\hat{y}(\tau) \in Y_c$, $\tau \in [t, t+T_p]$
    - examples: obstacles, collision avoidance;
  - the input: $\mathfrak{u}(\tau) \in U_c$, $\tau \in [t, t+T_p]$
    - examples: input saturation.

Nonlinear model predictive control

Intuitive idea





NMPC optimization problem

- At each time $t = t_k$, for $\tau \in [t, t + T_p]$, the following optimization problem is solved:

$$\begin{aligned} & u^*(t : t + T_p) = \arg\min_{\mathfrak{u}(\cdot)} J\left(\mathfrak{u}(t : t + T_p)\right) \\ & \text{subject to:} \\ & \qquad \dot{\hat{x}}(\tau) = f\left(\hat{x}(\tau), \mathfrak{u}(\tau)\right), \quad \hat{x}(t) = x(t) \\ & \qquad \hat{y}(\tau) = h\left(\hat{x}(\tau), \mathfrak{u}(\tau)\right) \\ & \qquad \hat{x}(\tau) \in \mathcal{X}_c,\ \hat{y}(\tau) \in \mathcal{Y}_c,\ \mathfrak{u}(\tau) \in \mathcal{U}_c \end{aligned} \tag{2}$$

where T_s is the sampling time, T_p is the prediction horizon, with $0 \leq T_s \leq T_p$.

NMPC optimization problem

- The optimization problem (2)
  - is in general non-convex;
  - must be solved on-line, at each time t_k.
- J is a function of the signal $u(\cdot)$. Since a signal is a function of time, J is a function of a function.
  - Such a mathematical object is often called a *functional*.
- Efficient numerical algorithm can be used for the solution of (2).
  - No guarantees to find a global minimum. They provide in general a local minimum.
  - A local minimum can be satisfactory from the point of view of control performance.

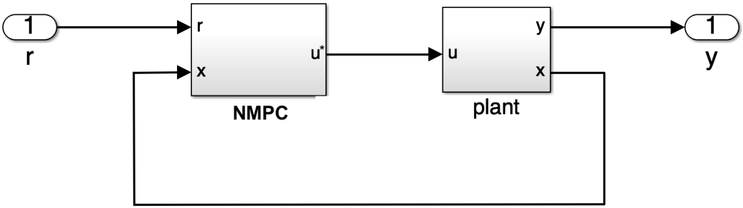
Receding horizon strategy

- Suppose that, at a time $t = t_k$, the optimal input signal $u^*(t : t + T_p)$ has been computed solving the above optimization problem.
  - $u^*(t : t + T_p)$ is an open-loop input: it depends on $x(t)$ but not on $x(\tau),\ \tau > t$.
  - $u^*(t : t + T_p)$, if applied for the entire time interval $[t, t + T_p]$, does not perform a feedback action, and thus it cannot increase the precision, reduce error and disturbance effects, or adapt to a varying scenario.
- The NMPC feedback control algorithm is obtained by means of a so-called *receding horizon strategy*:
  1. At time $t = t_k$:
     a. compute $u^*(t : t + T_p)$ by solving (2);
     b. apply only the first input value: $u(\tau) = u^*(t = t_k)$ and keep it constant for $\forall\tau \in [t_k, t_{k+1}]$.
  2. Repeat steps 1a-1b for $t = t_{k+1}, t_{k+2}, \ldots$

Nonlinear model predictive control

Closed-loop scheme

- Plant: $\dot{x} = f(x, u)$, $y = h(x, u)$
- NMPC: on-line solution of (2) and receding horizon strategy.
  - The NMPC algorithm contains a plant model, used for prediction.
  - The prediction model is of the form $\dot{\hat{x}} = \mathrm{f}(\hat{x}, \mathrm{u})$, $\hat{y} = \mathrm{h}(\hat{x}, \mathrm{u})$, where $\mathrm{f} \cong f$, $\mathrm{h} \cong h$.
    - Simplified models (f, h) are often used.
    - In the nominal case, $\mathrm{f} = f$, $\mathrm{h} = h$.



NMPC design

Choice of parameters

- T_s: In many situations, the sampling time is given and cannot be chosen. If it can be chosen, a trial and error procedure in simulation can be adopted, considering that T_s should be
  - sufficiently small to deal with the plant dynamics (Nyquist-Shannon sampling Theorem);
  - not too small, to avoid numerical problems and slow computation.
- T_p: It can be chosen through a trial and error procedure in simulation, considering that
  - a "large" T_p increases the closed-loop stability properties;
  - a "too large" T_p may reduce the short-time tracking accuracy.
- Q, R, P: Similar to LQR/LQRY. See next slide.

NMPC design

Choice of weight matrices

1. **Initial choice:** Supposing that all the variables have similar ranges of variation, Q, R and P can be chosen diagonal non-negative, with
   - $Q_{ii} = \begin{cases} > 0 & \text{in the presence of requirements on } y_i \\ = 0 & \text{otherwise} \end{cases}$
   - $P_{ii} = \begin{cases} > 0 & \text{in the presence of requirements on } y_i \\ = 0 & \text{otherwise} \end{cases}$
   - $R_{ii} = \begin{cases} > 0 & \text{in the presence of requirements on } u_i \\ \cong 0 & \text{otherwise.} \end{cases}$
2. **Trial and error (in simulation):** Change the values of Q_{ii}, R_{ii} and P_{ii}, until the requirements are satisfied.

increasing Q_{ii}, P_{ii}	$\Rightarrow$	decreasing the energy of x_i, y_i	$\Rightarrow$	reducing oscillations and convergence time
increasing R_{ii}	$\Rightarrow$	decreasing the energy of u_i	$\Rightarrow$	reducing command effort and “energy consumption”

NMPC design

Matlab/Simulink

```matlab
%% NMPC design

% The prediction model must be defined
% in a function named pred_model.m

par.nlc=0;  % no state/output constraints (default=0)
% par.nlc=1;  % presence of state/output constraints
% The constraints must be defined
% in a function named nlcon.m

% Prediction model order
par.n=...  NUMBER OF STATES
           z={x,y,...}: nz=6

% Sampling time and prediction horizon
par.Ts=...
par.Tp=...

% Weigth matrices
par.P=...
par.Q=...
par.R=...

% Command input lower and upper bounds
par.lb=...
par.ub=...

% par.Tstart=...   % Time at which the NMPC
% controller is switched on (default=0).

K=nmpc_design_st2(par);

% K: structure used by the NMPC block in Simulink.
```

```matlab
function [f,h] = pred_model(t,x,u)
% NMPC prediction model
% t: time (scalar, useful for time-varying systems).
% x: state of the system (dimension nx1).
% u: input of the system (dimension nux1).
% f,h: functions of the state and output equations:
% xdot=f(t,x,u); y=h(t,x,u).

% Initialization
f=zeros(n,1);
h=zeros(ny,1);

% State equations
f = ...;

% Output equations
h = ...;

function F = nlcon(x,y)
% NMPC constraint function
% x: state of the system; matrix of dimension n*N.
% y: output of the system; matrix of dimension ny*N.
% N is the number of samples in the time interval [t,t+Tp].
% F: constraint function; Nc*N matrix, where
% Nc is the number of constraints.
% Constraints are written in the standard form F(x,y)<=0.

% Initialization
N=size(x,2);
Nc=...;
F=zeros(Nc,N);

% Constraint functions
F(1,:) = ...;
% F(2,:) = ...;
```

Discussion

Advantages and drawbacks of NMPC

- Advantages:
  - general and flexible: complex MIMO systems;
  - intuitive formulation, based on optimality concepts;
  - constraints and input saturation accounted for;
    - constraints/saturations can be time-varying;
  - efficient management of the performance/input activity trade-off;
  - optimal trajectories (over a finite time interval);
  - unified computation of optimal trajectory and control law.
- Drawbacks:
  - high on-line computational cost;
  - possible local minima in the optimization problem;
  - problems in the case unstable zero-dynamics (like all methods).