

# Model Predictive Control

## Litterature Notes

Victor Risager

January 9, 2024

## 1 A basic formulation

We assume that state values cannot be measured, thus we need an observer. So we use state estimates  $\hat{x}(k|k)$  of  $x(k|k)$  indicating that it is a measurement that is based on all states up until time  $k$ . It is based on  $u(k-1)$  and not  $u(k)$  since that input has not been determined yet.  $\hat{u}(k+i|k)$  denotes the future values at time  $k+i$  on input  $u$  which is assumed on time  $k$ . This means that  $i$  is some horizon, and  $u(k+j|k), j=0,1,\dots,i-1$  is the inputs at each time step. we have the cost function:

$$V = \sum_{i=H_w}^{H_p} \|\hat{z}(k+i|k) - r(k+i|k)\|_{Q(i)}^2 + \sum_{i=0}^{H_u-1} \|\Delta\hat{u}(k+i|k)\|_{R(i)}^2 \quad (1)$$

where  $r(k+i|k)$  is a reference trajectory and  $\hat{z}(k+i|k)$  is the controlled outputs. The prediction horizon has length  $H_p$  but  $H_w$  indicates the predicition window, which determines when to start penalizing. If  $H_w > 1$  then we only penalize from that point forward, as there may be some delay between control inputs and effects.  $H_u$  is the control horizon, where  $H_u \leq H_p$  and future control differences between  $\Delta\hat{u}(k+i|k) = 0$  and  $\Delta\hat{u}(k+i|k)$  where  $i > H_u$ . Note that the cost function in (1) only penalizes changes in  $u$  and not  $u$  itself. The matrices  $Q(i)$  and  $R(i)$  are weights and both positive semidefinite  $(\cdot) \geq 0$ .  $Q(i)$  penalizes the change in control variables  $z$  and  $R(i)$  penalizes change in input variables.

## 2 Constraints

There are different constraints, which are assumed to hold over the entire control- and prediction horizon.

$$E \text{ vec}(\Delta\hat{u}(k|k), \dots, \Delta\hat{u}(k+H_u-1|k), 1) \leq \text{vec}(0) \quad (2)$$

$$F \text{ vec}(\hat{u}(k|k), \dots, \hat{u}(k+H_u-1|k), 1) \leq \text{vec}(0) \quad (3)$$

$$G \text{ vec}(\hat{z}(k+H_w|k), \dots, \hat{z}(k+H_p|k), 1) \leq \text{vec}(0) \quad (4)$$

where  $E, F$ , and  $G$  are matrices of suitable dimensions. (2) can be used to represent actuator slew rate (change of input of an actuator), actuator ranges (3), and control variable constraints on  $z$  based on (4)

## 3 General features of constrained predictive control

Predictive controllers often act linearly in the interior of the feasible region, but non linearly close to the constraints. Although constrained predictive controllers are nonlinear they are usually time invariant. This means that there is a control function  $u = h(x)$  such that  $u$  depends interely on the state  $x$  and not on time  $t$ .

If the number of inputs  $u$  is not the same as number of measured  $y$  and controlled  $z$  outputs, then at least one of these representations is not minimal - that is it must be either uncontrollable or unobservable since they will have different state dimensions.

## 4 Allowing computational delay

The time between measurements are assumed to be the same  $T_s$  and the plant output  $y(k)$  is measured at  $kT_s$ . Then if there is a disturbance into the system, it is assumed to be measured at the same time. This is labelled  $d_m(k)$ . There is a delay  $\tau$ , from the measurement is taken until the predictive controller has completed its computations.  $\tau$  is assumed to be the same in every timestep.

## 5 Prediction

The prediction algorithm can have great effect on the performance of the algorithm. Therefore the reader is advised to choose the prediction algorithm wisely, as a tuning parameter.

## 6 No disturbances, full state measurement

We assume that the entire state vector is measured. that is  $\hat{x}(k|k) = x(k) = y(k)$ , so  $C_y = I$ . We assume no knowledge of the disturbances or potential measurement noise. We iterate the prediction:

$$\hat{x}(k+1|k) = Ax(k) + B\hat{u}(k|k) \quad (5)$$

$$\hat{x}(k+2|k) = A\hat{x}(k+1|k) + B\hat{u}(k+1|k) \quad (6)$$

$$= A^2x(k) + AB\hat{u}(k|k) + B\hat{u}(k+1|k) \quad (7)$$

$$\vdots \quad (8)$$

$$\hat{x}(k+H_p|k) = A\hat{x}(k+H_p-1|k) + B\hat{u}(k+H_p-1|k) \quad (9)$$

$$= A^{H_p}x(k) + A^{H_p-1}B\hat{u}(k|k) + \dots + B\hat{u}(k+H_p-1|k) \quad (10)$$

## 7 Constant output disturbance

## 8 Unconstrained problems

We can rewrite the cost function (1) as

$$V(k) = \|Z(k) - T(k)\|_Q^2 + \|\Delta U(k)\|_R^2 \quad (11)$$

where

$$Z(k) = \begin{bmatrix} \hat{z}(k+H_w|k) \\ \vdots \\ \hat{z}(k+H_p|k) \end{bmatrix} \quad (12)$$

$$T(k) = \begin{bmatrix} \hat{r}(k+H_w|k) \\ \vdots \\ \hat{r}(k+H_p|k) \end{bmatrix} \quad (13)$$

$$\Delta U(k) = \begin{bmatrix} \hat{u}(k|k) \\ \vdots \\ \hat{u}(k+H_u-1|k) \end{bmatrix} \quad (14)$$

Where the  $Q$  and  $R$  matrices are weight matrices

## 9 Handling infeasibility

There are multiple ways of handling infeasibility of the quadratic programming problem

- Ad hoc measures:

- Use  $u(k-1|k)$  instead of the newly computed control law.
- Use the next  $u(k+2|k)$ , such that we still look forward
- Active constraint management
  - Relax the least important constraints.

One strategy is softening constraints. That is, constraints can be violated if absolutely necessary. **Input constraints are usually very hard.** and **Output constraints are usually better.** This makes sense as the robot may have limited maximum wheel acceleration, or max velocities etc. Introduce slack variables. remember active/inactive constraints.