Highlights

The Beggining of Control Revolution: Ofset-free Koopman MPC

Patrik Valábek

- Research highlight 1
- Research highlight 2

The Beggining of Control Revolution: Ofset-free Koopman MPC

Patrik Valábek

[a]*STU, Redlinskeho 9, Bratislava, Slovakia*

Abstract

Keywords:

1. Introduction

2. Preliminaries and Notation

2.1. Parsim-K Identification

2.2. Offset-Free MPC

2.3. Nonlinear MPC

2.4. Koopman Operator for Control

2.5. Koopman MPC

2.6. State Estimation - Observer

3. Koopman MPC with Offset-Free - Easyiest

In this section, we will present a standard offset free optimal framework consisting of Target Optimization, State Estimation - Observer and MPC. As observer is standard Kalman filter or extended Kalman filter, we will discuss mainly remaining two optimization problems. Both remaining components had to be designed in a way that they are able to work with the Koopman operator. The main goal of this section is to show that the standard formulation can be easily transformed for the offset-free MPC use.

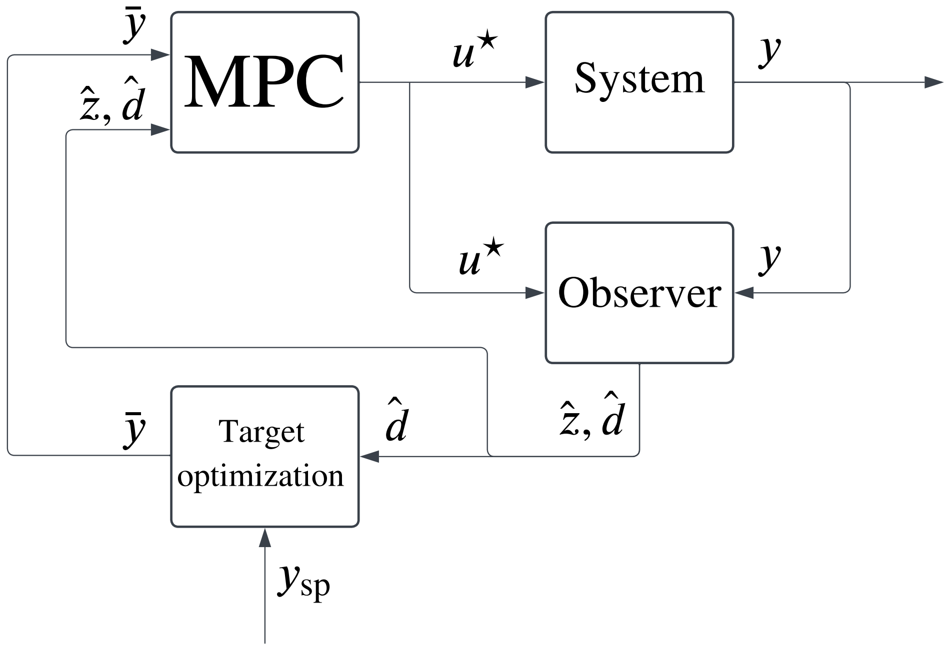

Figure 1: Closed-loop control structure.

3.1. Target Optimization

$$
\begin{aligned}
\min_{\bar{u},\bar{z},\bar{y}} \quad & (\bar{y} - y_{\text{ref}})^\intercal Q_{\text{y}} (\bar{y} - y_{\text{ref}}) && \text{(1a)} \\
\text{s.t.} \quad & \bar{z} = A\bar{z} + B\bar{u} && \text{(1b)} \\
& \bar{y} = C\bar{z} + d && \text{(1c)} \\
& u_{\min} \leq \bar{u} \leq u_{\max} && \text{(1d)} \\
& y_{\min} \leq \bar{y} \leq y_{\max} && \text{(1e)} \\
& d = \hat{d}(t), \quad y_{\text{ref}} = y_{\text{sp}} && \text{(1f)}
\end{aligned}
$$

3.2. MPC

$$
\begin{aligned}
\min_{u_0,\ldots,u_{N-1}} \quad & \sum_{k=0}^{N-1} (y_k - \bar{y})^\intercal Q_\mathrm{y} (y_k - \bar{y}) + \sum_{k=0}^{N-1} \Delta u_k^\intercal Q_\mathrm{u} \Delta u_k, && \text{(2a)}\\
\text{s.t.} \quad & z_{k+1} = A z_k + B u_k, \quad k \in \mathbb{N}_0^{N-1}, && \text{(2b)}\\
& y_k = C z_k + d, \quad k \in \mathbb{N}_0^{N-1}, && \text{(2c)}\\
& \Delta u_k = u_k - u_{k-1}, \quad k \in \mathbb{N}_0^{N-1}, && \text{(2d)}\\
& u_{\min} \leq u_k \leq u_{\max}, \quad k \in \mathbb{N}_0^{N-1}, && \text{(2e)}\\
& y_{\min} \leq y_k \leq y_{\max}, \quad k \in \mathbb{N}_0^{N-1}, && \text{(2f)}\\
& z_0 = \hat{z}(t), \quad d = \hat{d}(t), \quad u_{-1} = u(t - T_\mathrm{s}) && \text{(2g)}
\end{aligned}
$$

3.3. Closed-Loop Control Structure

4. Koopman MPC with Offset-Free - Proposed

Main drawback that we observe in a current Koopman MPC (offset-free, robust, or any other), is that the optimization is done without leveraging knowledge of the lifted space. Of course we are predicting in the lifted space, but in objective function we are using original space, that is created derived as linear transformation of the lifted space $y_k = C z_k$. This transformation is not linear by nature $y_k = h(z_k)$, but to use linear MPC, we think we had no other choice. Some papers are not using this transformation directly, but after closer look, they are usually using equivalent transformations, for example expanding lifted space with measurements, but dynamic in lifted space is linear, so transformations to achieve back this measurements is linear. We will show that we can use the lifted space directly in the objective function, showing how to handle the lifted space and transformation of an objective function. We can fully eliminate the neccessity of the back transformation used in objective function in MPC, but we still need them to compute the target for the lifted space. In this case and case of constraints, we use a first order Taylor expansion, to better approximate the transformation.

4.1. Target Optimization

The objective function of target estimation is the same as in the standard offset-free MPC. The main difference is in

The proposed target optimization problem is formulated as:

$$\min_{\bar{u},\bar{z},\bar{y}} \quad (\bar{y} - y_{\mathrm{ref}})^\intercal Q_{\mathrm{y}} (\bar{y} - y_{\mathrm{ref}}), \tag{3a}$$

$$\text{s.t.} \quad \bar{z} = A\bar{z} + B\bar{u}, \tag{3b}$$

$$\bar{y} = H(z_{\mathrm{LP}})\bar{z} + h(z_{\mathrm{LP}}) - H(z_{\mathrm{LP}})z_{\mathrm{LP}} + d, \tag{3c}$$

$$u_{\min} \le \bar{u} \le u_{\max}, \tag{3d}$$

$$y_{\min} \le \bar{y} \le y_{\max}, \tag{3e}$$

$$d = \hat{d}(t), \quad y_{\mathrm{ref}} = y_{\mathrm{sp}}, \quad z_{\mathrm{LP}} = \bar{z}_{-1}, \tag{3f}$$

where $H(z_{\mathrm{LP}})$ is the Jacobian of the transformation $h(z_{\mathrm{LP}})$. z_{LP} is a linearization point, the point where we approximate our function with first order Taylor expansion. We propose, that this point should be as close as possible to the $\bar{z}$. Therefor we are using the previously calculated target $\bar{z}_{-1}$. This allow us to better approximate the transformation of the lifted space to the original space and more precise target optimization.

4.2. MPC

The proposed MPC problem is formulated as:

$$\min_{u_0,\dots,u_{N-1}} \quad \sum_{k=0}^{N-1} (z_k - \bar{z})^\intercal Q_{\mathrm{z}} (z_k - \bar{z}) + \sum_{k=0}^{N-1} \Delta u_k^\intercal Q_{\mathrm{u}} \Delta u_k, \tag{4a}$$

$$\text{s.t.} \quad z_{k+1} = Az_k + Bu_k, \quad k \in \mathbb{N}_0^{N-1}, \tag{4b}$$

$$y_k = H(z_0)z_k + h(z_0) - H(z_0)z_0 + d, \quad k \in \mathbb{N}_0^{N-1}, \tag{4c}$$

$$\Delta u_k = u_k - u_{k-1}, \quad k \in \mathbb{N}_0^{N-1}, \tag{4d}$$

$$u_{\min} \le u_k \le u_{\max}, \quad k \in \mathbb{N}_0^{N-1}, \tag{4e}$$

$$y_{\min} \le y_k \le y_{\max}, \quad k \in \mathbb{N}_0^{N-1}, \tag{4f}$$

$$z_0 = \hat{z}(t), \quad d = \hat{d}(t), \quad u_{-1} = u(t - T_{\mathrm{s}}) \tag{4g}$$

where $H(z_0)$ is the Jacobian of the transformation $h(z_0)$ at the point z_0. This linearization point is chosen to best approximate the surroundings of current operational point. The first order Taylor expansion is done to receive y_k, which is used only in the constrains of the system. In objective funtion, we use a tuning matric Q_{z} to penalize the difference from lifted space target. We propose to derive this tuning matrix as follows:

$$Q_{\mathrm{z}} = H(\bar{z})^\intercal Q_{\mathrm{y}} H(\bar{z}). \tag{5}$$

This allows us to use and tune our problem using matrix Q_{y}, which is convenient and easily tuned. This is a significant change from the standard Koopman MPC, where the objective function is defined in the original space.

4.3. Closed-Loop Control Structure

5. Block Diagonal Deep Koopman Transformation

TODO: split this section into theory and simulation evaluation During our work, we encounter a several problems with the suboptimal or infeasible solutions using Koopman MPC using Gurobi solver. For the enhancement of the performaance, we are proposing a block diagonal structure of the matrix A. This transformation can be done after obtaining the Koopman operator, with finding the transformation matrix T that transforms the Koopman operator to the block diagonal form:

$$\bar{A} = T^{-1}AT. \tag{6}$$

This transformation is done by finding the eigenvalues and eigenvectors of the Koopman operator, and then grouping them into blocks. The transformation matrix T is constructed from the eigenvectors, and the block diagonal form $\bar{A}$ is obtained by applying the transformation to the original Koopman operator A. Using this approach, we had to modify the rest of the model accordingly, using the transformation matrix T to transform the input and output matrices B and C as well as the lifted states z:

$$\bar{B} = T^{-1}B, \quad \bar{C} = CT, \quad \bar{z}_k = Tz_k. \tag{7}$$

When computing the Jackobian, we are transforming the original Jackobian $H(z_k)$ to the block diagonal form $\bar{H}(z_k)$ using the transformation matrix T:

$$\bar{H}(z_k) = H(z_k)T = H(T^{-1}\bar{z}_k)T. \tag{8}$$

After this transformation we are able to enhace the performance in optimization as can be seen in Table 1. This table is done using the formulation of Koopman MPC which can be seen in Sec. 3. The table shows the decrease in a computational times. Also with this formulation we are able to obtain optimal solution 100% time during simulation. Contrary, the full Koopman MPC formulation is able to obtain optimal solution only 26.7% of the time with the rest of the time giving warning, that the solution is suboptimal.

Table 1: Comparison of structure of problem to solver time and control performance.

Structure	Time TE/MPC	Optimal? [%]	OBJ	ST h_1	ST h_2
Full	0.2392 / 2.3884	26.7	100.0	43	48
Block Diagonal	0.2191 / 1.6204	100.0	94.1	25	31

We can see also increase in a control performance, which is measured by the objective function value, which is in optimal case arounf 6% lower. Also the settling time for both levels is lower.

The second option how to obtain the block diagonal structure is to learn the Koopman operator directly in the block diagonal form. This can be done by limiting the number of learnable parameters in a Koopman matrix during training to those that are in the block diagonal form. This also speeds the training as less parameters are learned.

To obtain an optimal performance, which is neccessary for the real time control, we are proposing to use the block diagonal structure of the Koopman operator. There is no neccessity to use the full Koopman operator, as the block diagonal structure is able to capture the same dynamics of the system and favors the optimization performance. The way this structure is obtained, during or after the training, is not important, as long as the Koopman operator is in the block diagonal form.

6. Simulation Setup

7. Simulation Results

8. Experimental Setup

9. Experimental Results

10. Discussion

11. Conclusion

Table 2: Comparison of several Tunings

LP MPC	LP TE	Tuning	OBJ
-	-	$J(z_s)$	171.22
C	C	$J(z_s)$	147.26
$J(z_s)$ w/o y-con	$J(z_{s,k-1})$	$J(z_s)$	149.18
$J(z_k)$	$J(z_k)$	$J(z_s)$	163.69
$J(z_k)$	$J(z_s)$	$J(z_s)$	149.35
-	-	C	156.21
C	C	C	151.73
$J(z_s)$ w/o y-con	$J(z_{s,k-1})$	C	158.80
$J(z_k)$	$J(z_k)$	C	177.33
$J(z_k)$	$J(z_s)$	C	158.68

Table 3: Comparison of several Methods

ALG	MODEL	OBJ
NMPC	Tru	88.09
MPC	Pa.-K (3)	167.51
MPC	DK	156.21
TMPC	DK (C)	147.22
TMPC	DK (J(z))	149.35