EBO780

Assignment 7
Theme 9: Introduction to Constrained MPC

MATLAB implementation of constrained linear MPC with Kalman filter for disturbance and state estimation

The purpose of this assignment is to familiarize the student with the practical working of model predictive control (MPC) by implementing a simple constrained linear MPC without using the MATLAB MPC toolbox. A linear discrete Kalman filter is used to estimate the unmeasured states of the process as well as unmeasured disturbances that include model mismatch and unmodelled dynamics.

References

- Rawlings, J.B. (2000). Tutorial overview of Model Predictive Control. IEEE Control Systems Magazine, Volume 20, Issue 3, Pages 38-52.
- Qin, S. J. and T. A. Badgwell (2003). A survey of industrial model predictive control technology. Control Engineering Practice. Volume 11, Pages 733-764.
- Mayne, D. Q., J. B. Rawlings, C. V. Rao and P. O. M. Scokaert. (2000). Constrained model predictive control: Stability and optimality. Automatica. Volume 36, Pages 789-814.
- Mathworks (2020), Kalman Filtering, Control Systems Toolbox, [Online] Available: https://www.mathworks.com/help/control/ug/kalman-filtering.html, Accessed April 2020.

Assignment

For this assignment you need to code a basic linear constrained MPC in MATLAB and Simulink using only the control system toolbox and the optimisation toolbox. Any use of the MPC toolbox will result in a mark of zero.

Refer to Fig. 1 below. The process $G_P(s)$ is controlled by a linear constrained MPC. There is model-plant-mismatch between the actual plant and the controller. The prediction model for the linear MPC is $G_C(s)$. The states of the process will be estimated with the use of a Kalman Filter (KF). The KF will make use of the same model as the MPC, but will be extended to include estimation of disturbances.

The prediction model $G_C(s)$ is:

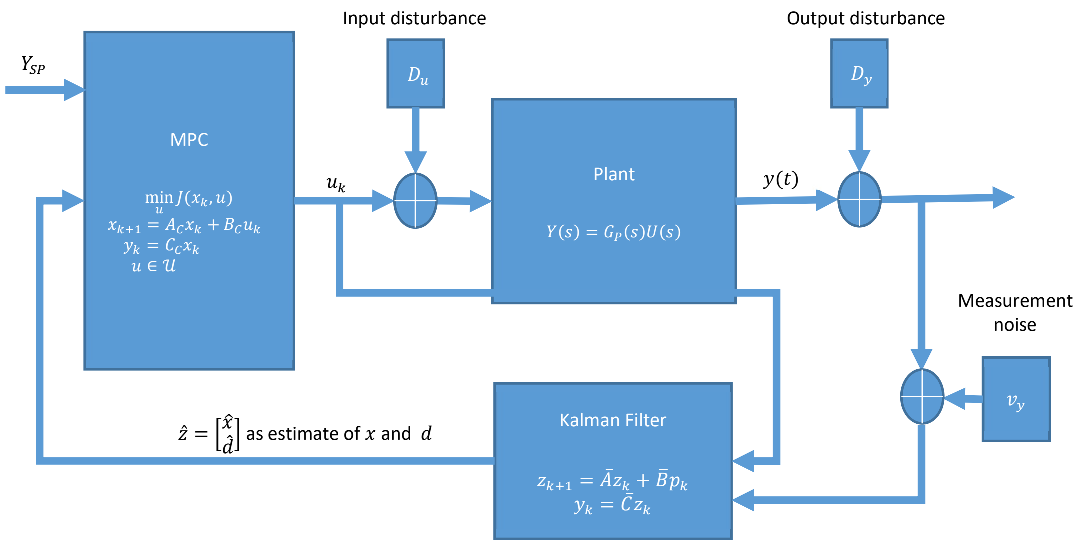
$$\frac{Y(s)}{U(s)} = G_C(s) = \begin{bmatrix} G_{C11} & G_{C12} \\ G_{C21} & G_{C22} \end{bmatrix} = \begin{bmatrix} \frac{12.8}{16.7s+1}e^{-1s} & \frac{-18.9}{21s+1}e^{-3s} \\ \frac{6.6}{10.9s+1}e^{-7s} & \frac{-19.4}{14.4s+1}e^{-3s} \end{bmatrix} \tag{1}$$

where $Y(s)$ is the controlled variable (CV) and $U(s)$ is the manipulated variable (MV)

The process model $G_P(s)$ differs from $G_C(s)$ in (1) as follows:

(a) the gain of G_{P11} will be 20% more than G_{C11},
(b) the time-constant of G_{P12} will be 20% longer than G_{C12},
(c) the gain of G_{21} will be 20% more than G_{C21}, and
(d) the time-constant of G_{22} will be 20% longer than G_{C22}.

Figure 1: Control configuration



Questions

- **Formulate and define MPC and KF:**

1. Define a discrete-time linear constrained MPC by giving the objective function and optimisation problem with constraints. Further, define all the variables/parameters involved. [10]

2. Define a discrete-time Kalman filter in state-space form which can estimate both the process states and process disturbances. [10]

- **Create continuous process model:**

3. Based on the specifications above, give the process model $G_P(s)$ in continuous LTI format. You will use this continuous LTI model in Simulink to simulate the plant. [5]

- **Create discrete MPC and EKF:**

Create the model for the MPC.

4. Create a discrete state-space prediction model called G_{Czss} based on the continuous model in (1) by discretizing it with a sampling time of 1 second. Map the time-delays to discrete-time states. (See the functions: *c2d*, *ss*, *absorbDelay,* in Matlab.) Document the discrete state-space model. [5]

Create the model for the KF. (Refer to Question 2 above.)

5. Create a discrete state-space model called $G_{Czss-kalma}$, based on your discrete-time model G_{Czs} for the MPC. The KF should also estimate the process disturbances. The process disturbances can be modelled as random walks. There should be a disturbance state for each input in G_{zss}, i.e., if there are n-inputs, there will be n-disturbance states. Ensure disturbance states are included in such a manner that you can add the disturbances to the prediction model inputs of the MPC directly. [5]

Create the KF as a state-space system.

6. Create a discrete-time linear KF using the MATLAB command "kalman" based on $G_{zss-kalman}$. The KF receives as inputs the CV values and the actual MV values from your controller. The KF produces as an output an estimate of the model states, the process disturbances and the CVs values. Document the KF system. Give the Q and R matrices that you used to design the KF. [5]

Create the MPC controller.

7. Code the MATLAB function "*constraints*" that enforce constraints on the inputs. This function will be included as part of the optimization function in the MPC. The constraints for $G_C(s)$ on MV 1 are $(-\infty, 1]$ and on MV 2 are $[-4, \infty)$. [5]

8. Code the Matlab function *ObjFunc* that calculates the objective value that will be optimized by *fmincon*. Your MPC controller must implement blocking to allow for a smooth closed-loop MV trajectory. The function "*ObjFunc*" should use the disturbance estimate from your KF as part of the predictions. Show the algorithm "*ObjFunc*" uses to calculates the predictions for the MPC. [10]

9. Add an extra input to your s-function m-file to read in your disturbance estimate from your KF. Exponentially filter the disturbance estimate to remove unwanted noise from the estimate. Add the filtered disturbance estimate to the MV values of your prediction model in such a way that the MV constraints will still be honoured by your MPC controller. [5]

Setup the simulation in Simulink

10. Ensure that the plant is simulated with $G_P(s)$. Therefore, there will be model-plant-mismatch between the plant and the model used for the MPC and KF. Create the following events in the simulation:
    - Add a step disturbance to MV 1 that will trigger at time 10 seconds with magnitude 1.
    - Add a step disturbance to MV 2 that will trigger at time 20 seconds with magnitude -1.
    - Add a step disturbance to the output of $G_P(s)$ that will add 1 to CV 1 and -1 to CV 2 at time 80 seconds.
    - Further, add band-limited white noise to the CVs of $G_P(s)$ with a noise power of $[0.1, 0.1]^T$ and a sampling time of 1 second.
    - Make a set-point step change of $[5, -5]^T$ at time 150 seconds.
    - Tune your controller to maintain the set-point if the constraints become active.

    Document your Simulink model. [10]

Run the simulation in Simulink

11. Run the Simulink model for 250 seconds. Use the fixed time step solver (ode4) with fundamental time-step of 1 second. This is set under the “Simulation->Configuration Parameters” menu. On the main screen under “Solver options” choose “Type” as “Fixed-step” and “Solver” as “ode4 (Runge-Kutta)” and set “fixed-step size (fundamental sample time)” to 1 second. Start with the initial condition of all the states at 0. Document the plots for the CVs, the setpoint changes, and the MVs. Document the plots of the disturbance estimates from the KF. Properly label the plots. [10]

12. Comment on the performance of the MPC and KF. [5]

13. Attach your MATLAB code to the end of the report.

Total: 85 marks