Constrained Linear Model Predictive Control

with Kalman filter for state and disturbance estimation

EBO780

Constrained MPC

- The optimization problem can treat constraints on:
  - Manipulated variables (MV's)
    - Eg. $u_{min} \leq u(t) \leq u_{max}$
    - If a control valve is fully open/closed, the flow-rate is constrained at its max/min value.
  - Controlled variables (CV's)
    - Eg. $y_{min} \leq y(t) \leq y_{max}$
    - In the case of cruise control, the car speed cannot be below 0 or above e.g. 120 km/h.
  - Associated variables (formula)
    - Eg. $\sqrt{u^2(t) - y^2(t)} \leq 3^2$
    - For the case above, the input and output must be within a circle with radius 3.
  - Internal states (based on estimation)
    - Eg. $x_{min} \leq x(t) \leq x_{max}$

Constrained MPC

- Constraint handling in MPC is not "magic".
  - There are many algorithms to minimize a cost function subject to constraints:
    - Interior point
    - Trust region reflective
    - Sequential Quadratic Programming
    - Active Set
    - (https://www.mathworks.com/help/optim/ug/choosing-the-algorithm.html)
  - MPC exploits the degrees of freedom available when an MV or CV is at a constraint.
  - If there are no degrees of freedom available, MPC compromises based on variable weightings in the cost function.

Problem Description (1)

Constrained optimization problem:

$$\min_{u} V(u, x_0)$$

Subject to: $x \in \mathcal{X}$

$u \in \mathcal{U}$

where:

$x_0 \triangleq$ initial state

$u \in \mathbb{R}^{N_C \times N_U}$

$\mathcal{X} \triangleq \{x \in \mathbb{R}^{N_X} | x_l \leq x \leq x_u\}$

$\mathcal{U} \triangleq \{u \in \mathbb{R}^{N_U} | u_l \leq u \leq u_u\}$

Problem Description (2)

$$V(u, x_0) = \sum_{i=1}^{N_P} \left(Y_{SP} - h(x,u)\right)^T Q \left(Y_{SP} - h(x,u)\right) + \sum_{i=1}^{N_C} \Delta u^T R \Delta u$$

where:

$\dot{x} = f(x,u)$
$y = h(x,u)$
$Y_{SP} \triangleq$ set point
$N_P =$ Prediction samples
$N_C =$ Control moves
$Q, R =$ Weights on outputs and inputs

Constraints in Matlab (1)

- Use *fmincon* for non-linear constrained optimization:

$$\min_u f(u) \text{ such that } \begin{cases} c(u) < 0 \\ c_{eq}(u) = 0 \\ Au \leq b \\ A_{eq}u = b_{eq} \\ low \leq u < high \end{cases}$$

- See Matlab for general use of *fmincon* as minimisation function: https://www.mathworks.com/help/optim/ug/fmincon.html

Constraints in Matlab (2)

- For MPC we can formulate the problem as:

$$\min_u V(x,u) \text{ such that } \begin{cases} c(x,u) < 0 \\ c_{eq}(x,u) = 0 \end{cases}$$

  - Linear and nonlinear inequality constraints captured by:

$$c(x,u) \leq 0$$

  - Linear and nonlinear equality constraints captured by:

$$c_{eq}(x,u) = 0$$

  - The *nonlcon* constraint function allows for advanced constraints. For *fmincon*, define *nonlcon* to include $c(x,u)$ and $c_{eq}(x,u)$ in optimisation problem:

```
function [c, ceq] = nonlcon(x,u)
c = […];
ceq = […];
```

Constraints in Matlab (3)

- Example:

  For MPC problem, constraints are: $|u_1| \leq 5$ and $-10 \leq u_2 \leq 3$:

  Problem formulation:

$$\min_u V(x,u) \text{ such that } c(u) < 0$$

  where

$$c(u) = \begin{bmatrix} u_1 - 5 \\ -u_1 - 5 \\ u_2 - 3 \\ -u_2 - 10 \end{bmatrix} \leq \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

  (We can easily include inequality constraints on x in this function.)

Constraints in Matlab (3)

- Example:

  A tank of 15 m^3 in an industrial process contains three entities:

  x_1, x_2, x_3 [m^3]

  The total volume of material in the tank is: $V = x_1 + x_2 + x_3$

  During operation the volume is constrained: $20\% \leq V \leq 85\%$.

  Therefore, the constraint function is:

$$c(x) = \begin{bmatrix} \frac{x_1 + x_2 + x_3}{15} - 0.85 \\ -\frac{x_1 + x_2 + x_3}{15} - 0.20 \end{bmatrix} \leq 0$$

MPC Feedback (1)

- In ideal world:
  - Models are 100% accurate.
  - All disturbances are measured.
  - Measurements are perfect.
  - There is full state-feedback, i.e. all states are measured directly:

    Consider the process described by:

$$\dot{x} = f(x, u)$$
$$y = h(x, u)$$

    where y is a measurable. The control objective is that y should follow the setpoint Y_{SP}:

$$V(u, x_0) = \sum_{i=1}^{N_P} \left(Y_{SP} - h(x, u)\right)^T Q \left(Y_{SP} - h(x, u)\right) + \sum_{i=1}^{N_C} \Delta u^T R \Delta u$$

    The measurement model $h(x, u)$ is in the objective function and not y directly.

    Full state feedback assumes we measure x directly to calculate $h(x, u)$.

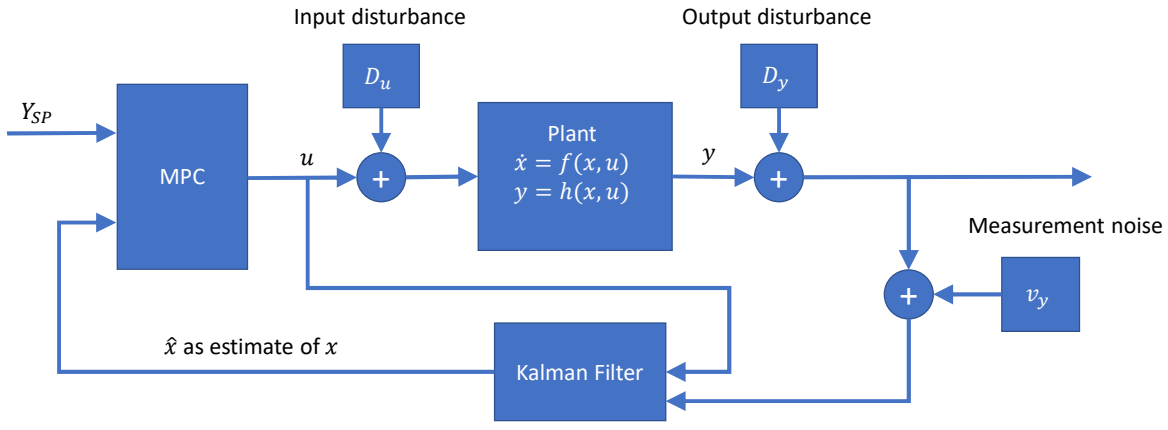
    Measurement of x ensures there is closed-loop feedback and integral action.

MPC Feedback (2)

- In real world:
  - Models are not 100% accurate.
    - There can be unmodelled dynamics (e.g. linear model of nonlinear systems)
  - Not all disturbances can be measured.
  - Measurements are noisy.
  - Not all states can be measured.
    - Since y and u can be measured, create a state estimator to estimate x.
    - Estimate of x is used as feedback to objective function to calculate $h(x,u)$.
    - The estimate of x provides closed-loop feedback to ensure integral action.

MPC Feedback: Kalman Filter (1)

- Real world for control:



MPC Feedback: Kalman Filter (2)

- Kalman Filter (KF):
  - In real world the true state x cannot be measured directly.
  - KF takes measurements of y and u to calculate $\hat{x}$.
    - $\hat{x}$ is an estimate of the true state x.
    - Measurements of y includes noise v_y.
- The process model for the KF is:

$$x_{k+1} = Ax_k + Bu_k + Gw_k$$
$$y_k = Cx_k + v_k$$

where

- w_k is the process noise with covariance: $E(w_k w_k^T) = Q$
- v_k is the measurement noise with covariance: $E(v_k v_k^T) = R$

The noise covariance matrices are important for designing the filter.

MPC Feedback: Kalman Filter (3) Algorithm

- Step 1: Measurement update

Use measurement at time k to produce estimate $\hat{x}_{k|k}$:

$$\hat{x}_{k|k} = \hat{x}_{k|k-1} + M(y_{v_k} - C\hat{x}_{k|k-1})$$

where $y_v = y + v$ is the output measurement with noise.

$\hat{x}_{k|k-1}$ is the estimate of x at time k given information until $k-1$.

$\hat{x}_{k|k}$ is the updated estimate based on last measurement y_{v_k}.

$y_{v_k} - C\hat{x}_{k|k-1}$ is the innovation function

M is the Kalman gain

- The innovation function gives the difference between the measured and predicted values of y based on the estimate $\hat{x}_{k|k-1}$.
- Kalman gain M is chosen to minimize the steady-state covariances of the estimation error based on the noise covariances Q and R.

MPC Feedback: Kalman Filter (4) Algorithm

- Step 1: Measurement update

  Use measurement at time k to produce estimate $\hat{x}_{k|k}$:

$$\hat{x}_{k|k} = \hat{x}_{k|k-1} + M\left(y_{v_k} - C\hat{x}_{k|k-1}\right)$$

- Step 2: Time update

  Predict next $\hat{x}_{k+1|k}$ with model used by KF:

$$\hat{x}_{k+1|k} = A\hat{x}_{k|k} + Bu_k$$

- Step 3: Do steps 1 and 2 at each time stamp k.

MPC Feedback: Kalman Filter (5) Algorithm Summary

- The measurement and time updates can be combined into a state-space model of the KF with matrices $A_{KF}, B_{KF}, C_{KF}, D_{KF}$:

$$\hat{x}_{k+1|k} = A_{KF}\hat{x}_{k|k-1} + B_{KF}\begin{bmatrix} u_k \\ y_{v_k} \end{bmatrix} = A(I - MC)\hat{x}_{k|k-1} + \begin{bmatrix} B & AM \end{bmatrix}\begin{bmatrix} u_k \\ y_{v_k} \end{bmatrix}$$
$$\hat{y}_{k|k} = C_{KF}\hat{x}_{k|k-1} + D_{KF}y_{v_k} = C(I - MC)\hat{x}_{k|k-1} + CMy_{v_k}$$

  where $\hat{y}_{k|k}$ is the EKF estimate of y_k.

  Note, the filter receives u_k and y_{v_k} as input information.

  Use command *kalman* in MATLAB to design the KF.

Exponential Filter

- Measurement noise can be reduced with a simple recursive exponential filter:

$$\bar{y}_k = \alpha y_{v_k} + (1 - \alpha)\bar{y}_{k-1}$$

where

$\bar{y}_k =$ filtered value at time k

α = fraction between 0 and 1 to control amount of filtering

y_{v_k} = current noisy measurement

Kalman Filter (1)
State and Disturbance Estimation

- KF can also estimate process disturbances.
- The disturbance estimate combine:
  - Unmodelled dynamics
  - Model mismatch
  - Unmeasured disturbances (both input and output)
- The MPC can use the disturbance estimate to improve disturbance rejection.

Kalman Filter (2)
State and Disturbance Estimation

- Formulation of KF for state and disturbance estimation:
  - Define a variable for the unknown process disturbances: d_k
  - The unknown process disturbance is assumed to follow a random walk:

$$d_{k+1} = d_k + w_k$$

    - For the random walk model above,
      - The disturbance variable is modelled as an integrating process.
      - The process noise influences the disturbance.

Kalman Filter (3)
State and Disturbance Estimation

- Formulation of KF for state and disturbance estimation:
  - Define the process model as:

$$x_{k+1} = Ax_k + B(u_k + d_k)$$
$$d_{k+1} = d_k + w_k$$
$$y_k = Cx_k$$

  - For the definition above, d_k is added to the input. This formulation makes it easier if an MPC needs to reject disturbances. The KF estimate of d_k can easily be added to the input in the MPC algorithm.

  - (It is possible to define the system as:

$$x_{k+1} = Ax_k + Bu_k + \bar{d}_k$$

    This is similar to the case above, such that $\bar{d}_k = Bd_k$.
    This formulation can be less amenable for use with MPC.)

Kalman Filter (4)
State and Disturbance Estimation

- Formulation of KF for state and disturbance estimation:
  - Define the process model as:

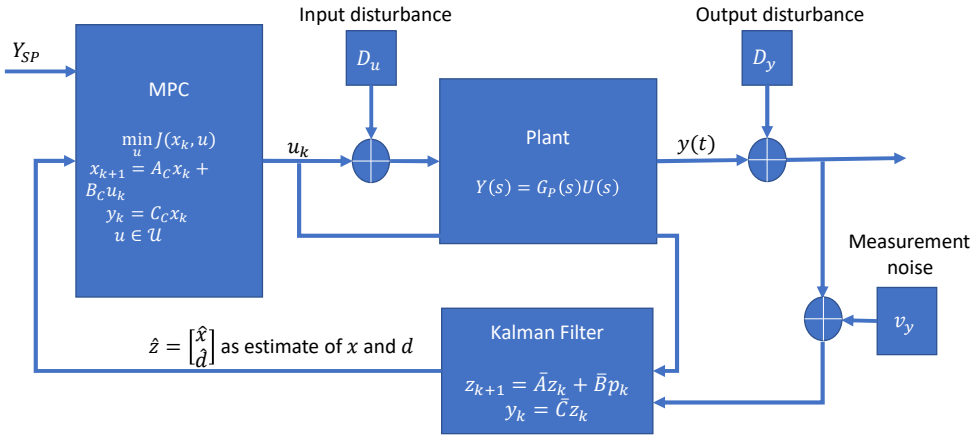
$$x_{k+1} = Ax_k + B(u_k + d_k)$$
$$d_{k+1} = d_k + w_k$$
$$y_k = Cx_k$$

  - This can be written as:

$$\begin{bmatrix} x_{k+1} \\ d_{k+1} \end{bmatrix} = \begin{bmatrix} A & B \\ 0 & I \end{bmatrix} \begin{bmatrix} x_k \\ d_k \end{bmatrix} + \begin{bmatrix} B & 0 \\ 0 & I \end{bmatrix} \begin{bmatrix} u_k \\ w_k \end{bmatrix} \quad \rightarrow \quad z_{k+1} = \bar{A}z_k + \bar{B}p_k$$
$$y_k = \begin{bmatrix} C & 0 \end{bmatrix} \begin{bmatrix} x_k \\ d_k \end{bmatrix} \quad \rightarrow \quad y_k = \bar{C}z_k$$

where $z_k = \begin{bmatrix} x_k \\ d_k \end{bmatrix}$; $p_k = \begin{bmatrix} u_k \\ w_k \end{bmatrix}$; $\bar{A} = \begin{bmatrix} A & B \\ 0 & I \end{bmatrix}$; $\bar{B} = \begin{bmatrix} B & 0 \\ 0 & I \end{bmatrix}$; $\bar{C} = \begin{bmatrix} C & 0 \end{bmatrix}$

  - Use augmented model above to create KF.



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.
- Mathworks (2020), fmincon, Control Systems Toolbox, [Online] Available: https://www.mathworks.com/help/optim/ug/fmincon.html, Accessed April 2020.
- Mathworks (2020), Choosing the Algorithm, Control Systems Toolbox, [Online] Available: https://www.mathworks.com/help/optim/ug/choosing-the-algorithm.html#bsbwxm7, Accessed April 2020.