

State Space Disturbance Model Development

1. Assume something reasonable:

- Step disturbance to output:

$$\Delta y_w(k) = e(k)$$

- Ramp disturbance to output:

$$\begin{aligned}\Delta x_w(k+1) &= \Delta x_w(k) + e(k) \\ \Delta y_w(k) &= \Delta x_w(k) + e(k)\end{aligned}$$

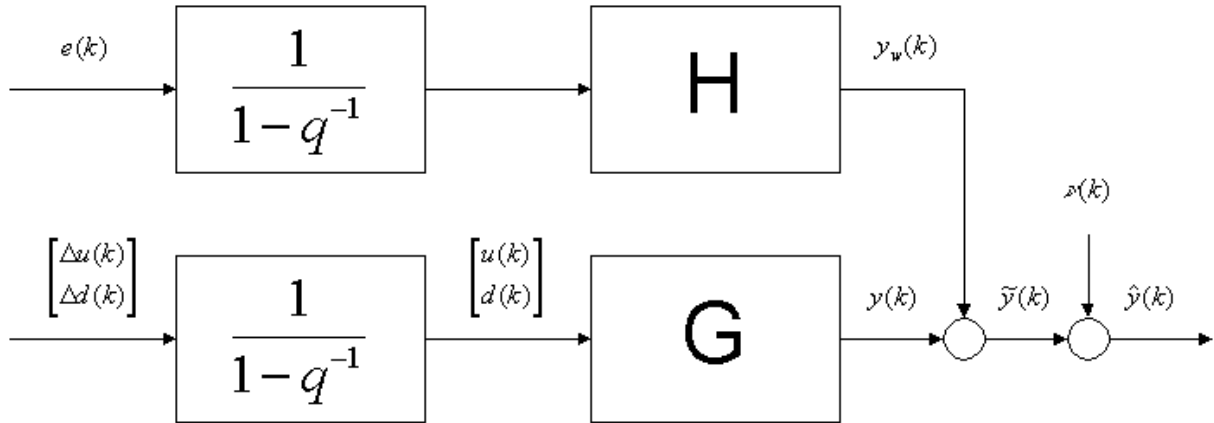
2. From fundamental ODE's: unmeasured disturbances in ODE's.

$$\begin{aligned}\Delta x_w(k+1) &= A\Delta x_w(k) + B_w e(k) \\ \Delta y_w(k) &= C\Delta x_w(k)\end{aligned}$$

3. From Historical Plant Data: Given historical plant data, the stochastic state space model of the disturbance can be obtained using various techniques like spectral factorization and subspace identification.

Overall State Space Model

Plant with noise and state space disturbance model:



Overall model:

$$\begin{bmatrix} \Delta x(k+1) \\ \Delta x_w(k+1) \\ e(k+1) \\ \tilde{y}(k+1) \end{bmatrix} = \begin{bmatrix} A & 0 & 0 & 0 \\ 0 & A_w & 0 & 0 \\ 0 & 0 & 0 & 0 \\ CA & C_w A_w & C_w B_w & I \end{bmatrix} \begin{bmatrix} \Delta x(k) \\ \Delta x_w(k) \\ e(k) \\ \tilde{y}(k) \end{bmatrix} + \begin{bmatrix} B_u \\ 0 \\ 0 \\ CB_u \end{bmatrix} \Delta u(k) + \begin{bmatrix} B_d \\ 0 \\ 0 \\ CB_d \end{bmatrix} \Delta d(k) + \begin{bmatrix} 0 \\ 0 \\ I \\ D_w \end{bmatrix} e(k+1)$$

$$\hat{y}(k) = [0 \ 0 \ 0 \ I] \begin{bmatrix} \Delta x(k) \\ \Delta x_w(k) \\ e(k) \\ \tilde{y}(k) \end{bmatrix} + \nu(k)$$

Overall State Space Model (Continued)

↓ denote the above as

$$\begin{aligned} X(k+1) &= \Phi X(k) + \Delta u(k) + \Delta d(k) + e(k+1) \\ \hat{y}(k) &= \Xi X(k) + \nu(k) \end{aligned}$$

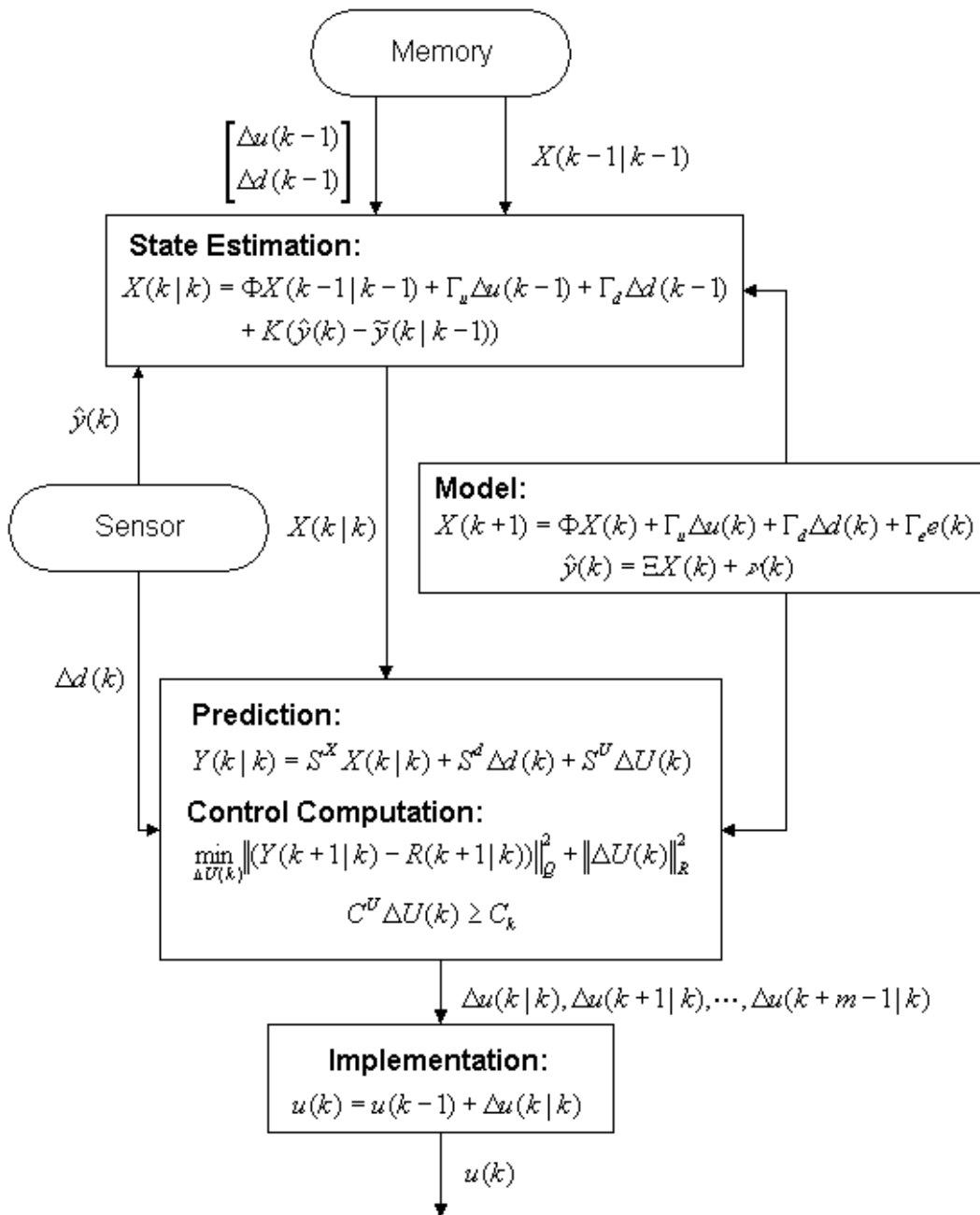
Given state space disturbance model, disturbance estimation can be done in a systematic way using well known Kalman filtering technique.

$$\begin{aligned} X(k|k-1) &= \Phi X(k-1|k-1) + \Delta u(k-1) + \Delta d(k-1) \\ X(k|k) &= X(k|k-1) + K(\hat{y}(k) - \tilde{y}(k|k-1)) \end{aligned}$$

- Recursive feedback update
- Various disturbance shape can be handled
- Cross-channel update

1.4 MPC FORMULATION USING STATE-SPACE MODEL

Overview



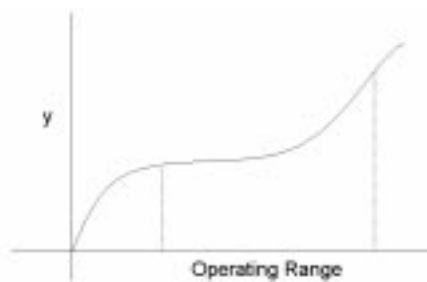
Chapter 2

NONLINEAR AND ADAPTIVE MODEL PREDICTIVE CONTROL

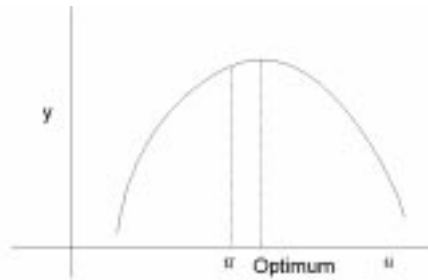
2.1 MOTIVATION

Why Nonlinear and Adaptive MPC?

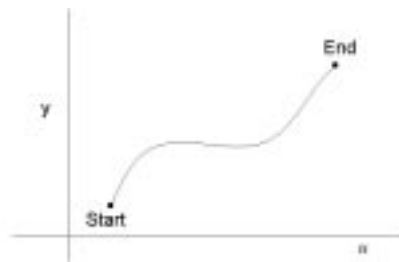
- Continuous processes with wide operating ranges



- Continuous processes with very strong nonlinearity (e.g., exothermic CSTR operated close to the optimum yield).



- Batch processes or other transition processes



These applications motivate development of

Nonlinear MPC or Adaptive MPC.

2.2 ISSUES IN NONLINEAR MPC

Issues

- **Nonlinear Models:** If first principle nonlinear ODE model is not available, do we have appropriate nonlinear system identification tools?
- **State Estimation:** At $t = k$,

$$x(k-1|k-1), u(k-1), d(k-1), y(k-1) \implies x(k|k)$$

The *open-loop* model prediction can be done through nonlinear model integration. However, the measurement correction is much more difficult. For instance, is linear gain correction

$$x(k|k) = x(k|k-1) + K(\hat{y}(k) - y(k|k-1))$$

sufficient? Also, how should we choose the gain matrix K ?

- **Control Computation:**

The prediction equation is no longer linear in the future input moves, i.e.,

$$\mathcal{Y}(k+1|k) = \tilde{\mathcal{F}}(x(k|k), d(k), \Delta\mathcal{U}(k))$$

Since we have nonlinear prediction constraints, the optimization is no longer QP and can be computationally expensive and unreliable (e.g., local minima).

Nonlinear Models

- First principle nonlinear ODE models

$$\begin{aligned}\frac{dx}{dt} &= f(x, u, d, w) \\ \hat{y} &= g(x)\end{aligned}$$

We will focus on this type of model in this lecture.

- Nonlinear difference equation model for nonlinear system identification

$$\begin{aligned}x(k+1) &= f(x(k), u(k), d(k)) \\ \hat{y}(k) &= g(x(k))\end{aligned}$$

- Artificial neural networks
- Nonlinear series expansion models such as Volterra model and NARX model.
- Rectilinear models where f and/or g are piece-wise linear.
- Linear model plus static nonlinearity
 - * Hammerstein Model: input nonlinearity
 - * Wiener Model: output nonlinearity

Generally, one should be very careful using a nonlinear model fitted to open loop data as the model can behave very differently when the loop is closed.

2.3 LINEARIZATION BASED NONLINEAR MPC

Standard Model

The standard model that we will use is of the following form:

$$\begin{aligned}\frac{dx}{dt} &= f(x, u, d, w) \\ \hat{y} &= g(x) + \nu\end{aligned}$$

We express the unmeasured disturbance w using the following stochastic equation driven by zero-mean white noise sequence $e(k)$:

$$\begin{aligned}x^e(k+1) &= A_e x^e(k) + B_e e(k) \\ w(k) &= C_e x^e(k)\end{aligned}$$

EXAMPLE: If $A_e = 1, B_e = 1, C_e = 1$, we have

$$w(k) = w(k-1) + e(k) \implies \Delta w(k) = e(k)$$

This means $w(k)$ is a random step.

We will assume the above random step model for $w(k)$ for simplicity.

Standard Model (Continued) Combining the two equations
give

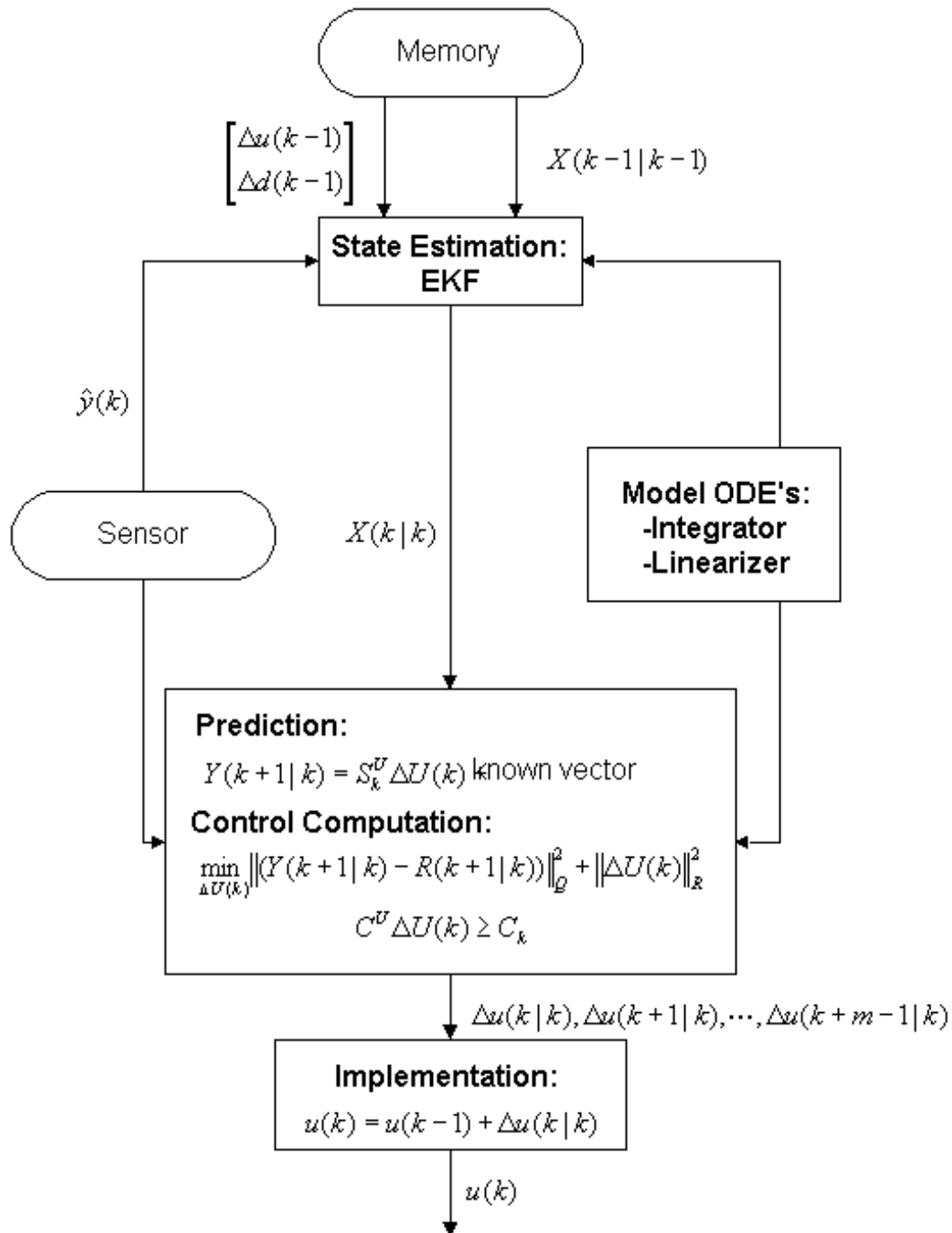
$$X(k+1) \triangleq \begin{bmatrix} x(k+1) \\ w(k+1) \end{bmatrix} = \begin{bmatrix} F_{t_s}(x(k), u(k), d(k), w(k)) \\ w(k) \end{bmatrix} + \begin{bmatrix} 0 \\ I \end{bmatrix} e(k)$$

where $F_{t_s}(x(k), u(k), d(k), w(k))$ stands for the state vector resulting from integrating the ODE for one sample interval (from $t = k$ to $t = k + 1$) with initial condition $x(k)$ and constant inputs of $u(t) = u(k)$ and $d(t) = d(k)$ and $w(t) = w(k)$.

We can also write the measurement equation as

$$\hat{y}(k) = g(x(k)) + \nu(k)$$

Overview of Linearization Based NLMPC



State Estimation

The following two steps are performed at $t = k$:

- **Model Prediction**

$$X(k|k-1) \triangleq \begin{bmatrix} x(k|k-1) \\ w(k|k-1) \end{bmatrix}$$

$$= \begin{bmatrix} F_{t_s}(x(k-1|k-1), u(k-1), d(k-1), w(k-1|k-1)) \\ w(k-1|k-1) \end{bmatrix}$$

Hence, this step involves nonlinear ODE integration for one sample interval.

- **Measurement Correction**

$$X(k|k) = X(k|k-1) + K_k \underbrace{(\hat{y}(k) - y(k|k-1))}_{\text{prediction error}}$$

where $y(k|k-1) = g(X(k|k-1))$.

K_k is the update gain (“filter gain”):

- Linear update structure is retained (suboptimal).
- The update gain needs to be varied with time due to the nonlinearity.
- The gain matrix can be computed using the model linearized with respect to the current state estimate and using linear filtering theory \implies Extended Kalman Filter (see the attached paper by Lee and Ricker for details).

Prediction

One can follow the similar argument as before and construct

$$\begin{aligned}
 \begin{bmatrix} x(k+1|k) \\ x(k+2|k) \\ \vdots \\ x(k+p|k) \end{bmatrix} &= \underbrace{\begin{bmatrix} F_{t_s}(x(k|k), u(k-1), d(k), w(k|k)) \\ F_{t_s}(x(k+1|k), u(k-1), d(k), w(k+1|k)) \\ \vdots \\ F_{t_s}(x(k+p-1|k), u(k-1), d(k), w(k+p-1|k)) \end{bmatrix}}_{\mathcal{F}: \text{ from ODE integration}} \\
 + \underbrace{\begin{bmatrix} B_k^u & 0 & \cdots & 0 \\ A_k B_k^u + B_k^u & B_k^u & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ \sum_{j=1}^p A_k^{j-1} B_k^u & \sum_{j=1}^{p-1} A_k^{j-1} B_k^u & \cdots & \sum_{j=1}^{p-m+1} A_k^{j-1} B_k^u \end{bmatrix}}_{\mathcal{S}_k^u: \text{ dynamic matrix}} \begin{bmatrix} \Delta u(k|k) \\ \Delta u(k+1|k) \\ \vdots \\ \Delta u(k+m-1|k) \end{bmatrix}
 \end{aligned}$$

where $w(k+i|k) = w(k|k)$ and A_k and B_k^u are computed through

- *Linearization*

$$\tilde{A}_k = \left(\frac{\partial f}{\partial x} \right)_{x(k|k), u(k-1), d(k), w(k|k)} ; \quad \tilde{B}_k = \left(\frac{\partial f}{\partial u} \right)_{x(k|k), u(k-1), d(k), w(k|k)}$$

- *Discretization*

$$A_k = \exp(\tilde{A}_k \cdot t_s) ; \quad B_k^u = \int_0^{t_s} (\tilde{A}_k \cdot \tau) d\tau \cdot \tilde{B}_k^u$$

Denote the above as

$$\begin{aligned}
 \mathcal{X}(k+1|k) &= \mathcal{F}(x(k|k), u(k-1), d(k), w(k|k)) \\
 &\quad + \mathcal{S}_k^u(x(k|k), u(k-1), d(k), w(k|k)) \Delta \mathcal{U}(k)
 \end{aligned}$$

Summary

At $t = k$, we are given the previous estimate $(x(k-1|k-1), w(k-1|k-1))$, previous inputs $d(k-1), u(k-1)$, and new measurements $\hat{y}(k), d(k)$. The following steps need to be performed:

1. **1-Step Model Integration:** Integrate the ODE for one time interval to obtain

$$X(k|k-1) \triangleq \begin{bmatrix} x(k|k-1) \\ w(k|k-1) \end{bmatrix} \\ = \begin{bmatrix} F_{t_s}(x(k-1|k-1), u(k-1), d(k-1), w(k-1|k-1)) \\ w(k-1|k-1) \end{bmatrix}$$

2. **Model Linearization:** Linearize the ODE and the measurement model with respect to $X(k-1|k-1)$ and $X(k|k-1)$.
3. **Filter Gain Computation:** Obtain the filter gain matrix K_k using the linearized model matrices (see the details in the attached paper by Lee and Ricker).
4. **Measurement Update of $X(k)$:** Update the estimate for $X(k)$ based on the model prediction error:

$$X(k|k) = X(k|k-1) + K_k \underbrace{(\hat{y}(k) - y(k|k-1))}_{\text{prediction error}}$$

5. **Model Linearization:** Linearize the ODE model again with respect to the updated state $X(k|k)$.
6. **Dynamic Matrix Computation:** Use the linearized model matrices to construct the dynamic matrix $\mathcal{S}_k^{\mathcal{U}}(x(k|k), u(k-1), d(k), w(k|k))$ according to the formula given earlier.
7. **p-Step Model Integration:** Integrate the ODE model for p time steps starting from $x(k|k)$ and keeping inputs constant at $u(t) = u(k-1)$, $d(t) = d(k)$ and $w(t) = w(k|k)$ for $k \leq t < k + p$.

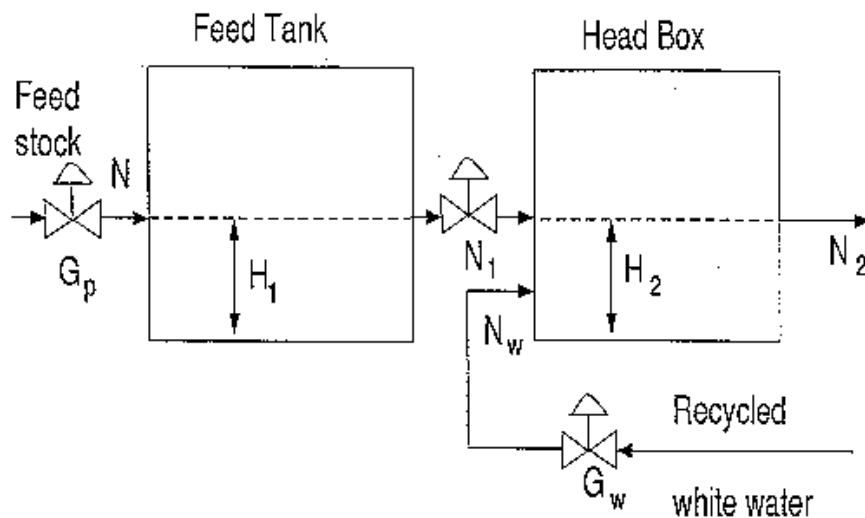
The prediction equation is

$$\begin{aligned} \mathcal{X}(k+1|k) &= \mathcal{F}(x(k|k), u(k-1), d(k), w(k|k)) \\ &\quad + \mathcal{S}_k^{\mathcal{U}}(x(k|k), u(k-1), d(k), w(k|k)) \Delta \mathcal{U}(k) \end{aligned}$$

8. **Input Computation:** Solve QP to find $\mathcal{U}(k)$.
9. **Input Implementation:** Implement $u(k) = u(k-1) + u(k|k)$.

2.4 EXAMPLE: PAPER MACHINE HEADBOX CONTROL

Problem Description



N : consistency of stock entering the feed tank.

N_w : consistency of recycled white water.

G_p : flowrate of stock entering the feed tank.

G_w : flowrate of recycled white water.

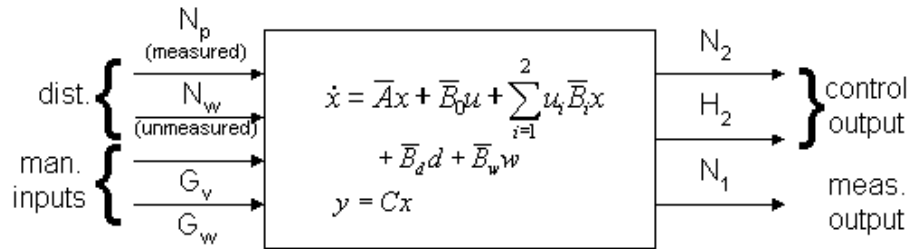
H_1 : liquid level in the feed tank.

H_2 : liquid level in the headbox.

N_1 : consistency in the feed tank.

N_2 : consistency in the headbox.

Some Specific Design Information



- ODE model for the above process is bilinear (see Lee and Ricker for model equations).
- We model N_w unmeasured disturbance as random walk, i.e.,

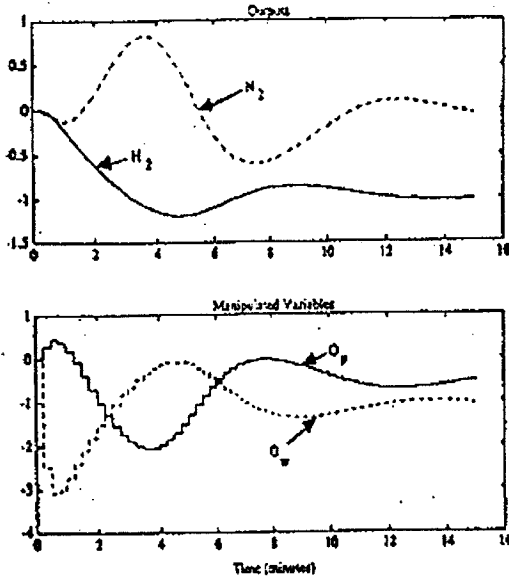
$$\begin{aligned} x^e(k+1) &= x^e(k) + e(k) \\ N^w(k) &= x^e(k) \end{aligned}$$

- We used the extended Kalman filter for state update.
- We used the following parameters for control computation:

$$p = 5, \quad m = 3, \quad y = \text{diag}\{1, 1, 0\}, \quad u = \lambda \text{diag}\{1, 1\}$$

Comparison Between Linear MPC and LB NLMPC

Linear MPC ($r = [0 \ -1]^T$, $N_w = 0$)



LBNLMPC ($r = [0 \ -1]^T$, $N_w = 0$) LBNLMPC ($r = [0 \ 0]^T$, $N_w = 10$)

