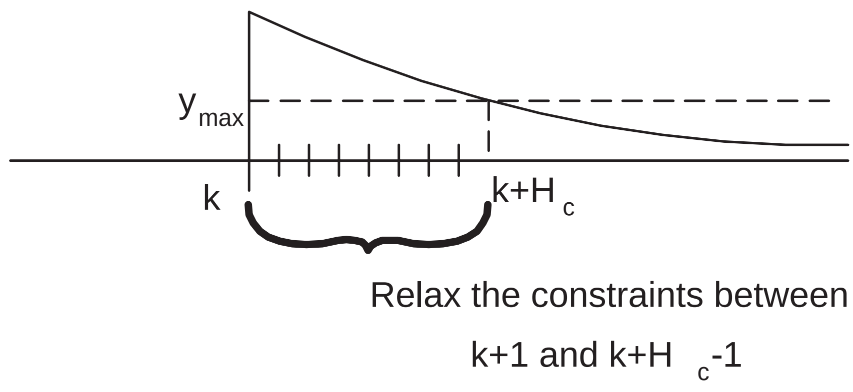


– Soften the constraint and penalize the degree of softening:

$$\min_{\epsilon,\Delta\mathcal{U}(k)}[\text{Usual Objective}] + \lambda\epsilon^2$$

$$y_{\min} - \epsilon \leq y(k+\ell|k) \leq y_{\max} + \epsilon$$
plus other constraints

2.4.2 GUIDELINES FOR CHOOSING THE HORIZON SIZE

In order to obtain good closed-loop properties and consistent tuning effect from problem to problem, it is recommended to use a very large or preferably infinite prediction horizon (Long-sighted decision making produces better results in general). ∞-horizon DMC can be implemented in the following way:

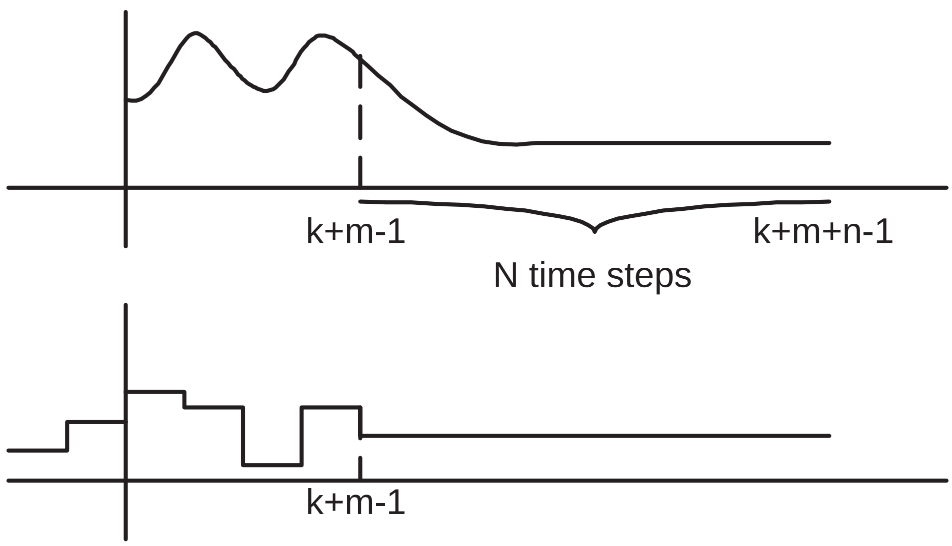
- choose m as large as possible (within the computational limit).
- choose

  $$p = m + n$$

  where n is the number of time steps for step responses to settle.
- add constraint

  $$y(k+p|k) = 0$$

Note that the above choice of p with the equality constraint amounts to choosing $p = \infty$. Stability of the closed-loop system is guaranteed under this cuhoice (regardless of choice of m). Choice of m is not critical for stability; a larger m should result in better performance at the expense of increased computational requirement.



The lesson is

- Use large enough horizon for system responses to settle.
- Try to penalize the endpoint error more (if not constrain to zero).

2.4.3 BI-LEVEL FORMULATION

In the DMC algorithm, control computation *at each sample time* is done in two steps:

- *Steady State Optimization:* Here model prediction at steady state is used to determine the optimal steady state. The steady-state model is in the form of

$$y(\infty|k) = K_s \underbrace{(u(\infty|k) - u(k-1))}_{\Delta u_s(k)} + b(k)$$

With only m moves considered,

$$\Delta u_s(k) = \Delta u(k|k) + \Delta u(k+1|k) + \ldots\ldots + \Delta u(k+m-1|k)$$

and with FIR assumption,

$$y(\infty|k) = y(k+m+n-1|k)$$

and $K_s = S_n$. Hence, the steady prediction equation can be easily extracted from the dynamic prediciton equation we had earlier.

In terms of the optimization criterion, various choices are possible.

– Most typically, some kind of linear economic criterion is used along with constraints on the inputs and outputs:

$$\min_{\Delta u_s(k)} [\ell(u(\infty|k), y(\infty|k))]$$

In this case, a linear programming (LP) results.

– Sometimes, the objective is chosen to minimize the input move size while satisfying various input / output constraints (posed by control requirements, actuator limits plus those set by the rigorous plant optimizer):

$$\min_{\Delta u_s(k)} [|\Delta u_s(k)|]$$

Again, an LP results.

– In the pure regulation problems where setpoint for the output is fixed, one may use

$$\min_{\Delta u_s(k)} [(r - y(\infty|k))^T Q (r - y(\infty|k))]$$

This combined with subsequently discussed QP results in *Infinite-Horizon MPC*.

- *Dynamic Optimization:* Once the steady-state target is fixed, the following QP is solved to drive the outputs (and sometimes also inputs) to their chosen targets quickly without violating constraints:

$$\min_{\Delta u(j|k)} \quad \Big[\sum_{i=1}^{m+n-2}(y(k+i|k)-y^*(\infty|k))^T Q (y(k+i|k)-y^*(\infty|k)) + \sum_{j=0}^{m-1}\Delta u^T(k+j|k) R \Delta u(k+j|k)\Big]$$

subject to

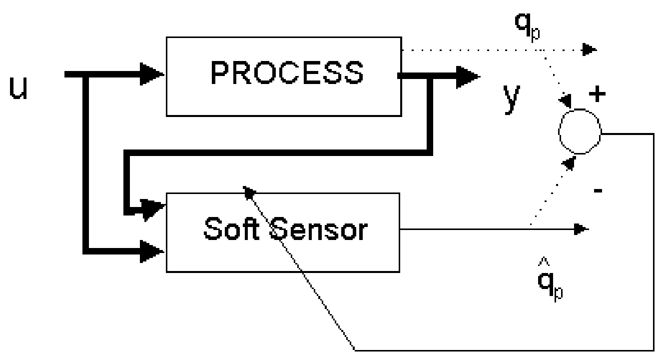
$$\Delta u(k|k)+\Delta u(k+1|k)+\ldots\ldots+\Delta u(k+m-1|k)=\Delta u_s^*(k)$$

plus various other constraints. This is a QP.

The last constraint forces $y(k+m+n-1|k)$ to be at the optimal steady-state value $y^*(k+\infty|k)$.

Note: The above steady-state optimization is to be distinguished from the rigorous plant-wide optimization. The above is performed at every sample time of MPC while the rigorous optimization is done much more infrequently.

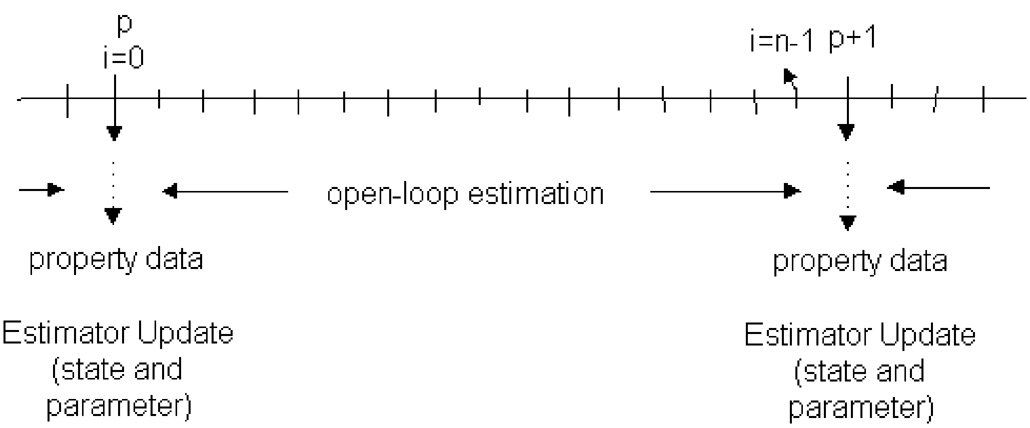
2.4.4 PROPERTY ESTIMATION



- Property data q are usually obtained through on-line analyzer or lab. analysis.
- Both have significant delays and limited sampling capabilities (more so for lab. analysis).
- On-line anayzers are highly unreliable (prone to failures).
- Using more reliable fast process measurments $\mathbf{y}$ (and possibly $\mathbf{u}$), we can estimate product properties at a higher frequency with a minimal delay.
- The property estimator (sometimes called *soft sensor*) can be constructed from a fundamental model or more commonly through data regression.
- Almost all estimators used in practice today are designed as *static* estimators.
- Since process variables exhibit different response times, *ad hoc* dynamic compensations (e.g., lead / lag elements, delays) are often added to the static estimator.

- If the number of process measurements is too large, the dimension can be reduced through PCA(principal component analysis) or orther correlation analyses.
- In some cases where nonlinearity is judged to be significant, Artificial Neural Networks are used for regression.
- Analyzer or lab results can be used to remove bias from the soft sensor. Suppose the soft sensor takes the form of $\hat{q}_s(p,i) = f(y(p,i))$. Then,

$$\hat{q}(p,i) = \hat{q}_s(p,i) + \underbrace{\lambda(q(p,0) - \hat{q}_s(p,0))}_{\text{bias correction}}, \quad 0 \leq \lambda \leq 1$$

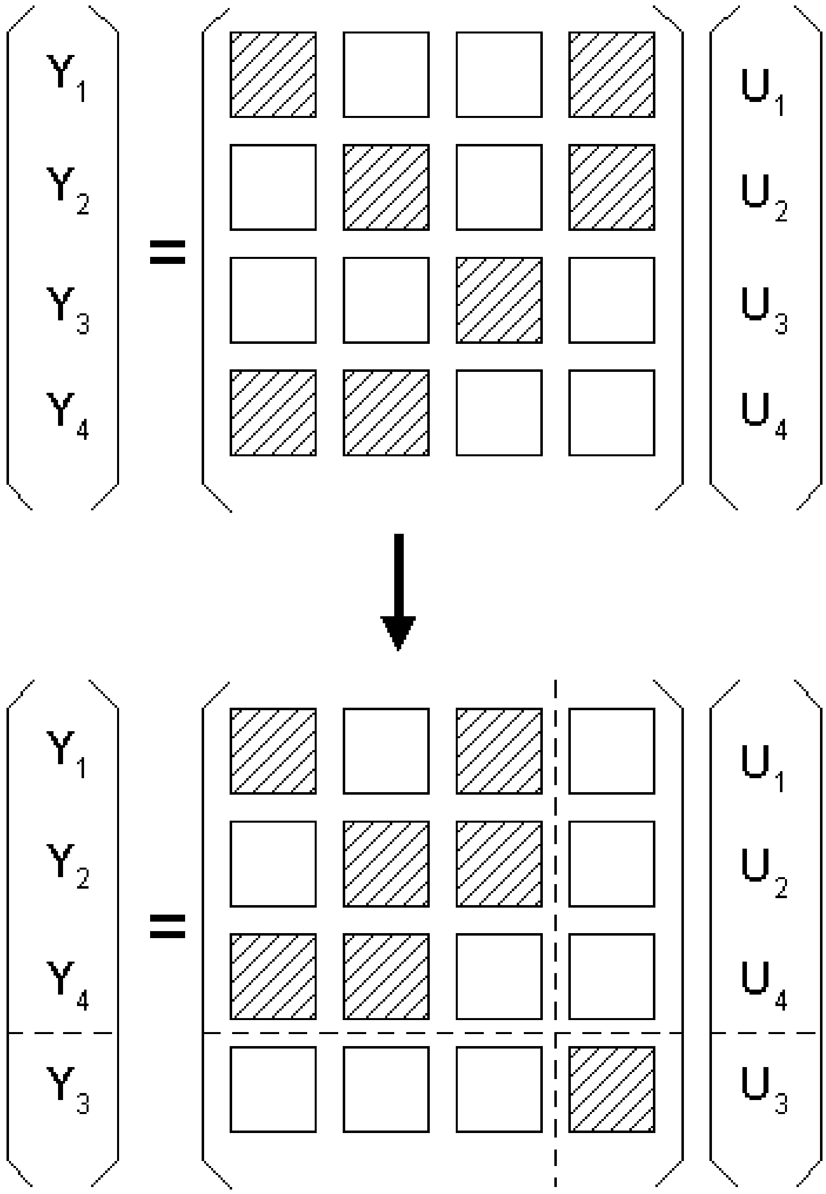


2.4.5 SYSTEM DECOMPOSITION

In MIMO processes, some input-output pairs have no or only weak coupling. Such systems can be decomposed into several subsystems and separate MPC can be designed for each subsystem.
The decentralized MPC design can reduce computational demand and improve numerical stability.

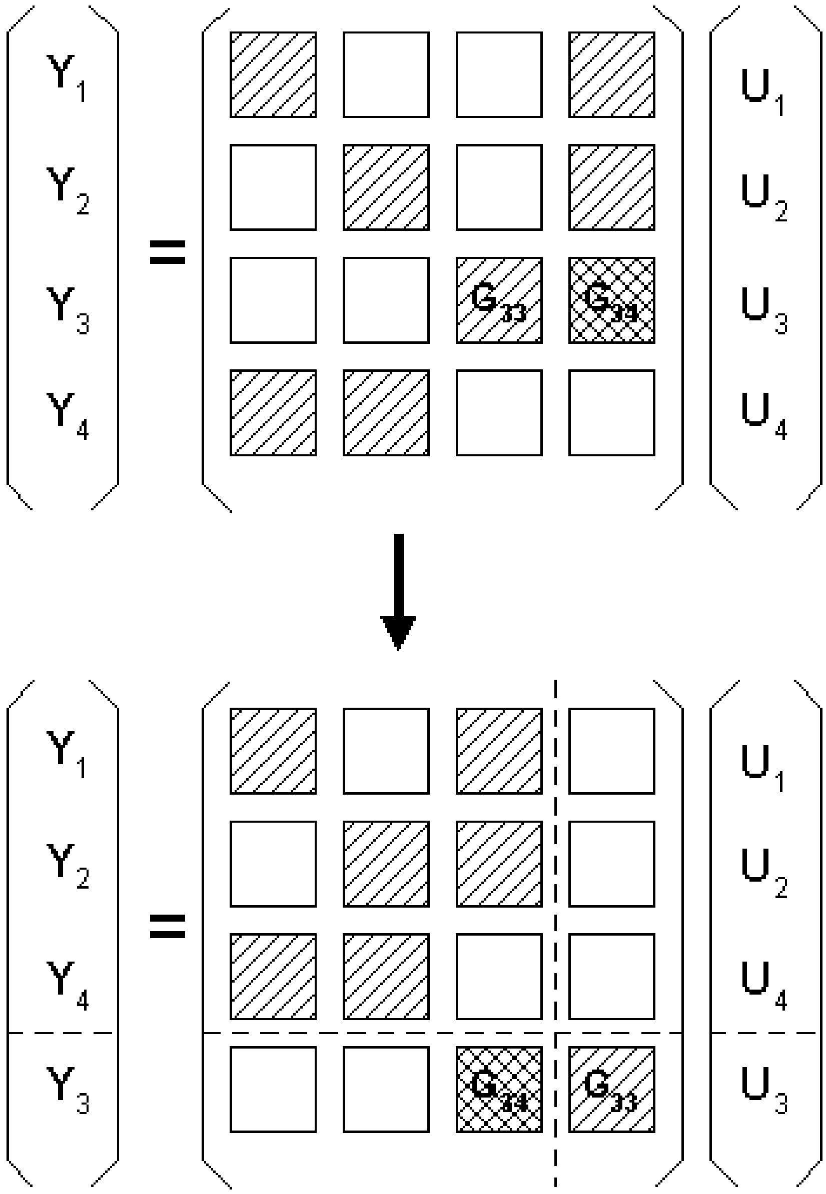
- Number of floating point computation in matrix algebra is proportional to n^2 or n^3.
- If we can decompose an n-dimensional system into two subsystems with equal size, the number of computation can be reduced from $O(n^2)$ or $O(n^3)$ to $O(n^2/4)$ or $O(n^3/8)$.
- System decomposition is not a trivial task in general. It is one of the continuing research issues studied under the title of *Control Structure Synthesis* or *Design of Decentralized Control*.
- Some of the rather obvious cases are as follows:

Case 1 : Complete Separation



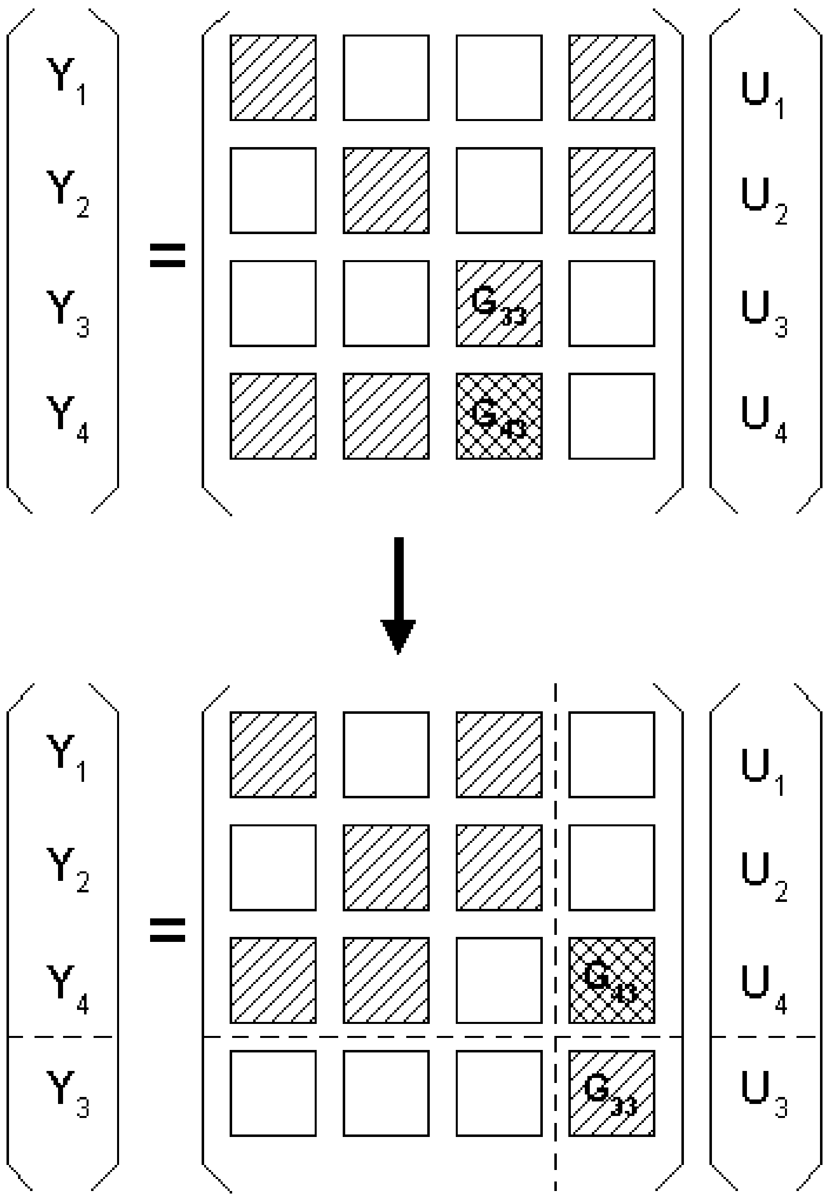
- The system can be decomposed into $(U_1\ U_2\ U_4) - (Y_1\ Y_2\ Y_4)$ and $U_3 - Y_3$ disjoint pairs.

Case 2 : Partial Separation I



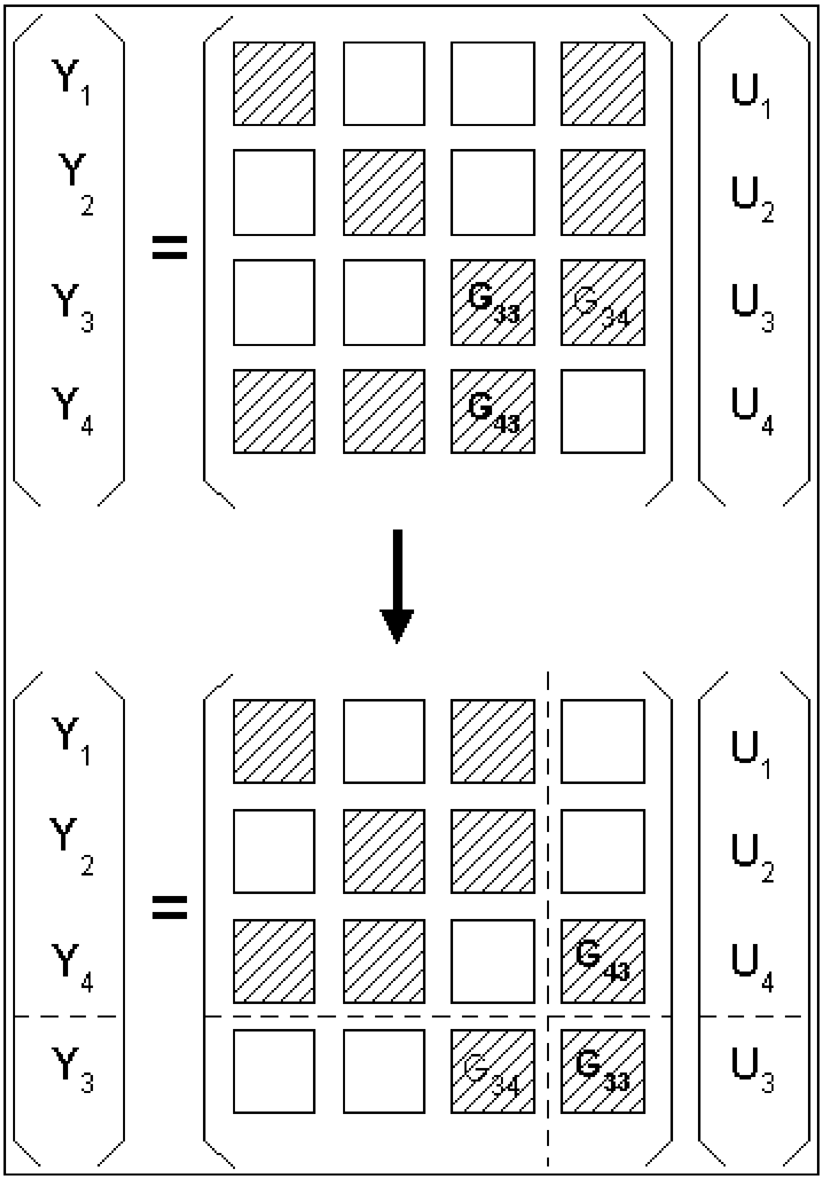
- $(Y_1\ Y_2\ Y_4)$ is not affected by U_3. But Y_3 is affected by U_4
- The system can be decomposed into two subsystems. In this case, U_4 can be treated as a measurable disturbance to the $U_3 - Y_3$ loop.

Case 3 : Partial Separation II



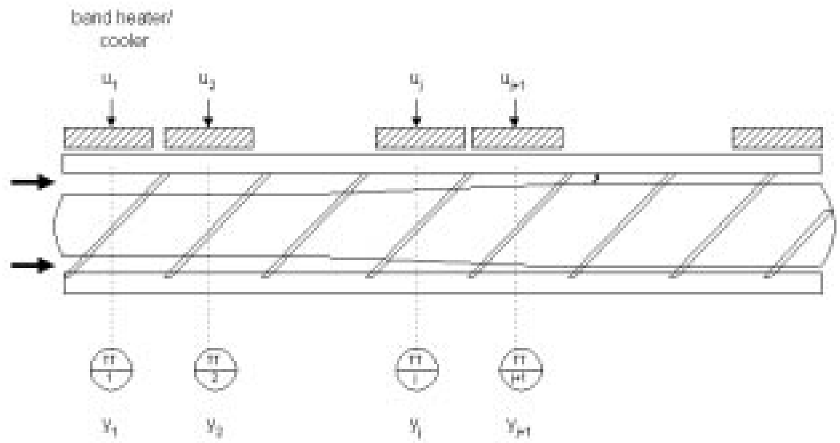
- Y_3 is not influenced by $(U_1\ U_2\ U_4)$. But, U_3 has an influence on Y_4.
- Similarly to above, the problem can be decomposed into two subproblems. U_3 acts as a measurable disturbance to the first block.

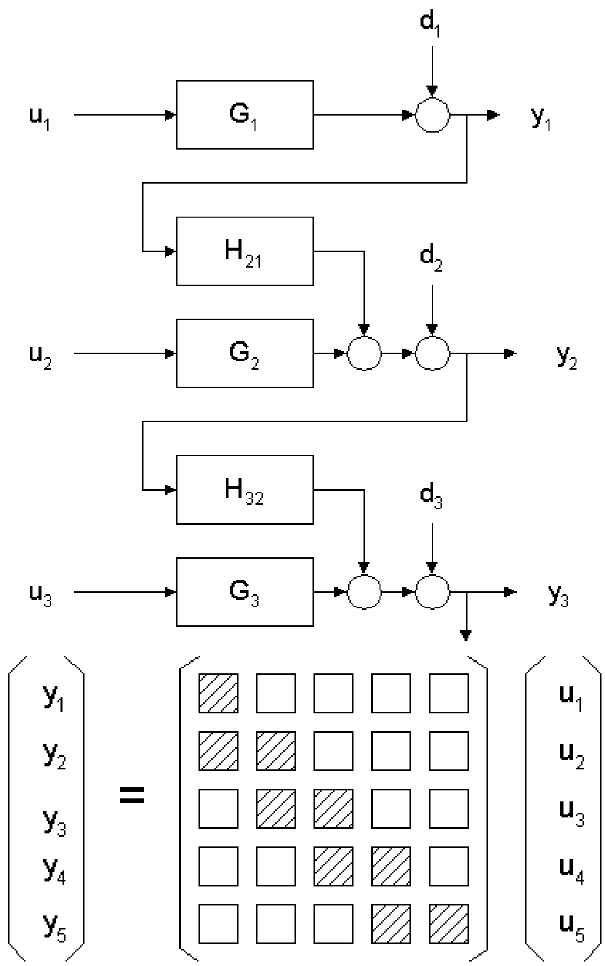
Case 4 : Partial Separation III



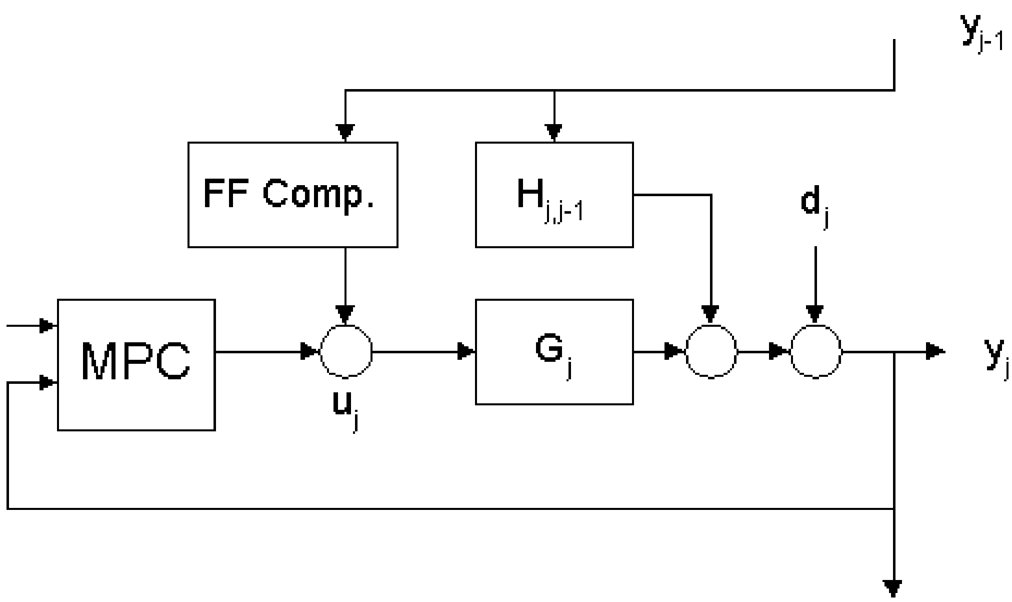
- If G_{34} and G_{43} have slower dynamics and smaller steady state gains than the other transfer functions, we may decompose the system as shown in the figure.

Example: Extrusion Process This example shows how the feedforward control can be constructed in a real process situation.





According to the input-output map, $u_j - y_j$ pair is decoupled from others while u_{j-1} plays a measurable disturbance to y_j.
Instead of treating u_{j-1} as a measured disturbance, however, it is better to take y_{j-1} as the measured disturbance and compensate its effect through the feedforward loop.



Decentralization Options

- Decentralization for both model update and optimization.
- Full model update, but decentralized optimization.
- Full model update, full steady-state optimization (LP), but decentralized dynamic optimization (QP).