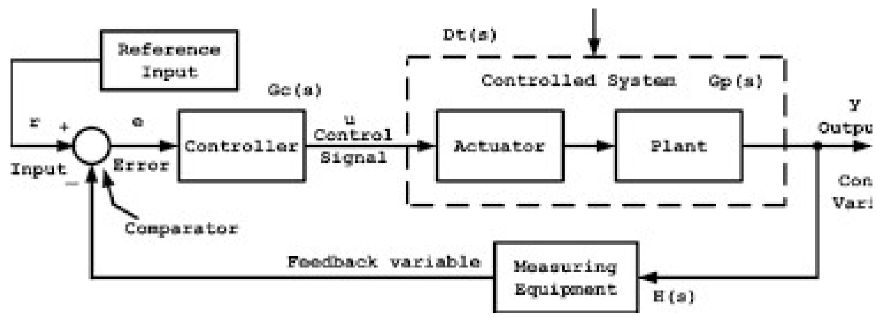
MIMO Control System

Introduction:

- At first, What is a control system?
    - A control system manages, commands, directs, or regulates the behavior of other devices or systems using control loops. It can range from a single home heating controller using a thermostat controlling a domestic boiler to large industrial control systems which are used for controlling processes or machines. The control systems are designed via control engineering process.
    - For continuously modulated control, a feedback controller is used to automatically control a process or operation. The control system compares the value or status of the process variable (PV) being controlled with the desired value or setpoint (SP), and applies the difference as a control signal to bring the process variable output of the plant to the same value as the setpoint.
- A system of inputs and outputs can be described as one of four types: SISO (single input, single output), SIMO (single input, multiple output), MISO (multiple input, single output), or MIMO (multiple input, multiple output).
    - SISO: These systems use data/input from one sensor to control one output. These are the simplest to design since they correspond one sensor to one actuator. For example, temperature (TC) is used to control the valve state of v1 through a PID controller.
    - SIMO: These systems use data/input from one sensor to control multiple outputs. For example, temperature (TC) is used to control the valve state of v1 and v2 through PID controllers.
    - MISO: These systems use data/input from multiple sensors to control one output. For example, a cascade controller can be considered MISO. Temperature (TC) is used in a PID controller (#1) to determine a flow rate set point i.e. FCset. With the FCset and FC controller, they are used to control the valve state of v1 through a PID controller (#2).

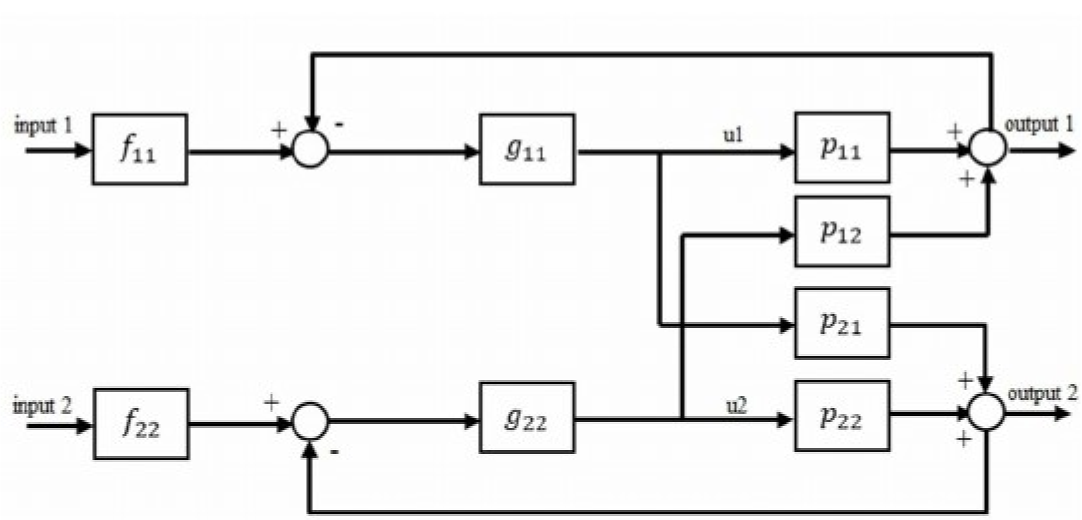
- MIMO: These systems use data/input from multiple sensors to control multiple outputs. These are usually the hardest to design since multiple sensor data is integrated to coordinate multiple actuators. For example, flow rate (FC) and temperature (TC) are used to control multiple valves (v1, v2, and v3). Often, MIMO systems are not PID controllers but rather designed for a specific situation.



MIMO controller

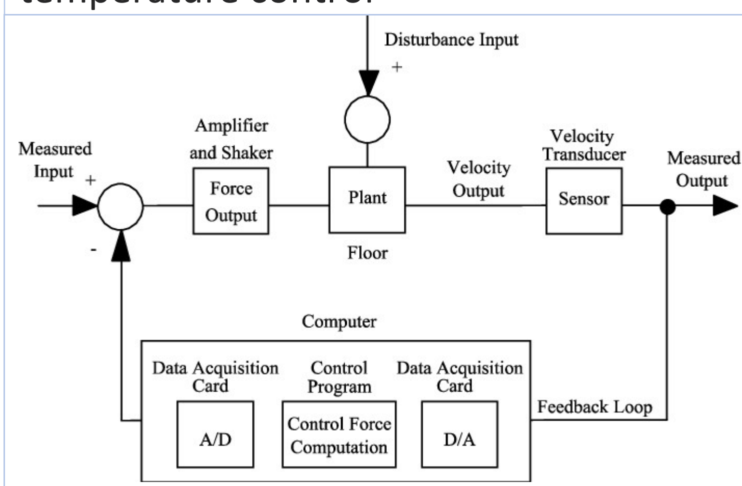
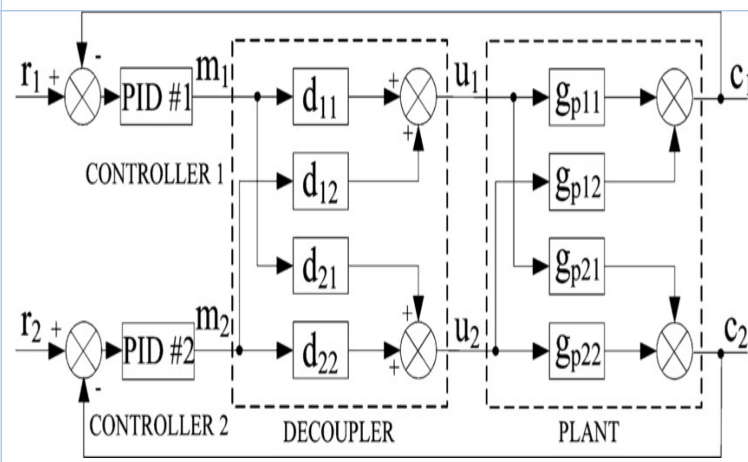
- Multiple input, multiple output (MIMO) systems describe processes with more than one input and more than one output which require multiple control loops.
- Thus such processes are difficult to control due to the presence of the interactions. The interactions between input/output variables are a common phenomenon and the main obstacle encountered in the design of multiloop controllers for interacting multivariable processes.
- Increase in complexity and interactions between inputs and outputs yield degraded process behavior. Such processes are found in process industries as they arise from the design of plants that are subject to rigid product quality specifications, are more energy efficient, have more material integration, and have better environmental performance.

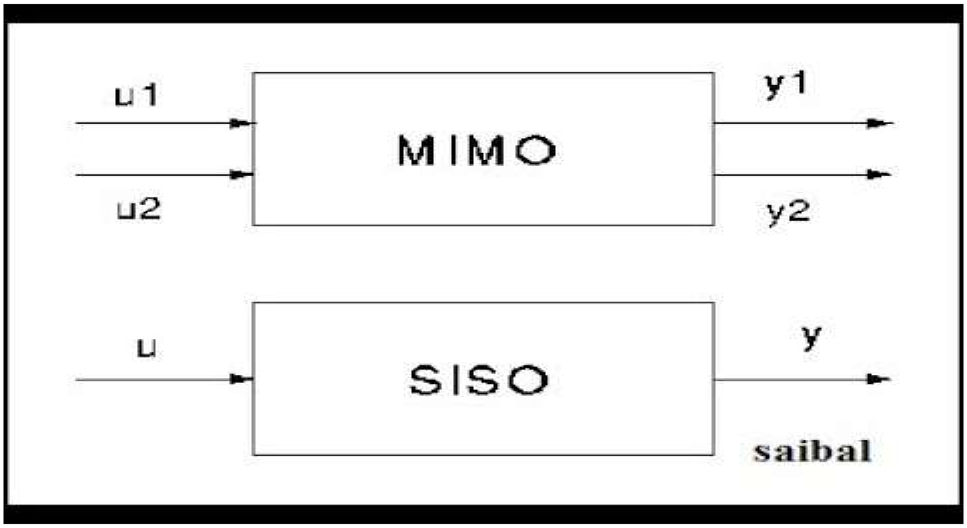
- Applying the tuning methods for a SISO (Single input- Single output) system to multi-loop systems often leads to poor performance and stability.
- Much research has been focused on how to efficiently take loop interactions into account in the multi-loop controller design. Thus, unless proper precautions are taken in terms of control system design, loop interactions can cause performance degradation and instability. This leads to the development of different controller tuning methods.
- Many methods have been proposed, including the Detuning method, Sequential Loop closing (SLC) method, Relay auto-tuning method, and independent loop method.
- Examples of MIMO systems include heat exchangers, chemical reactors, and distillation columns.
- These systems can be complicated through loop interactions that result in variables with unexpected effects. Decoupling the variables of that system will improve the control of that process.



Comparison between MIMO and SISO

SISO, as the name suggests, is a simple control system which has a single input signal and give out a single output signal. They are easy to design/implement and are used for most basic applications due to their Input/output simplicity. MIMO, as the name indicates, are complex types of control systems having multiple input signals triggering the system controller and in return generating multiple system outputs. These are larger and complex systems used in industrial applications mostly, which require a series of outputs.

SISO	MIMO
Single input single output	Multi input multi output
Less complex	More complex
Easy to design	Hard to design
Less accurate	More accurate
Ex: PID controller	Ex: model predictive control (MPC)
You can use in systems that depends on only one variable and no interaction exist	important in systems that have multiple dependencies and multiple interactions between different variables
Fan speed control, air conditioner temperature control	heat exchangers, chemical reactors, and distillation columns.
	



When to use MIMO control system?

- If your system has more than one input and output which require multiple control loops.
- in process industries as they arise from the design of plants that are subject to rigid product quality specifications, are more energy efficient, have more material integration, and have better environmental performance.

There are a lot of examples of MIMO:

- heat exchangers,
- chemical reactors
- distillation columns.

MIMO controllers

MIMO using Model Predictive Control:

Introduction:

This section describes how to control a system with multiple inputs and outputs using Model Predictive Control (MPC). MPC is a linear algebra method for predicting the result of a sequence of control variable manipulations. Once the results of specific manipulations are predicted, the controller can then proceed with the sequence that produces the desired result. One can compare this controller method to "look ahead" in chess or other board games. In look ahead, you foresee what an action might yield sometime in the future using specific knowledge of the process (or game in the case of chess), and are thereby able to optimize your actions to select for the best long term outcome. MPC methods can prevent an occurrence with conventional PID controllers in which actions taken achieve short term goals, but end up very costly in the end. This phenomenon can be described as "winning the battle but losing the war."

The open ended nature of MPC allows the process control engineer use MPC to control any system for which models can be generated.

Model Predictive Control:

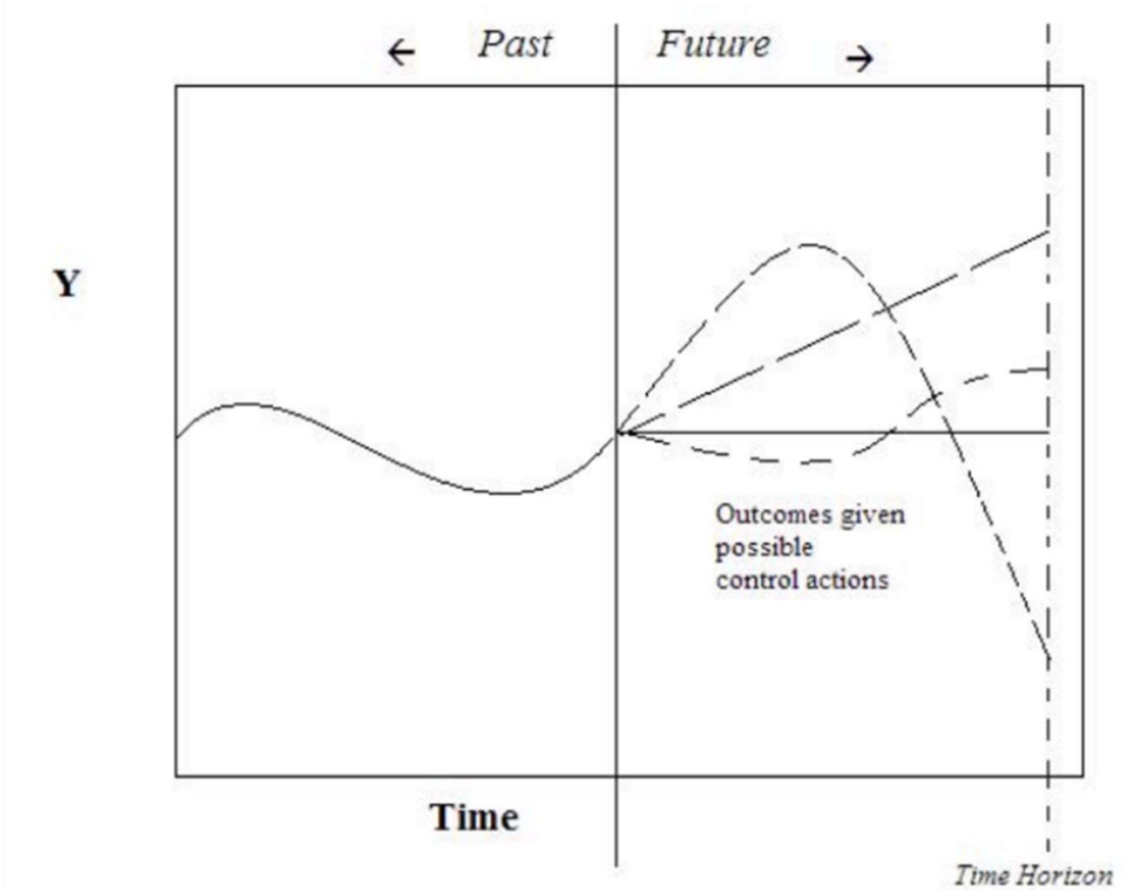
MPC is a widely used means to deal with large multivariable constrained control issues in industry. The main aim of MPC is to minimize a performance criterion in the future that would possibly be subject to constraints on the manipulated inputs and outputs, where the future behavior is computed according to a model of the plant. The model predictive controller uses the models and current plant measurements to calculate future moves in the independent variables that will result in operation that honors all independent and dependent variable constraints. The MPC then sends this set of independent variable moves to the corresponding regulatory controller set-points to be implemented in the process.

MPC uses the mathematical expressions of a process model to predict system behavior. These predictions are used to optimize the process over a defined time period. An MPC controller can operate according to the following algorithm.

1. Development of a process model by the control engineers.
2. At time t, previous process inputs and outputs are used, along with the process model, to predict future process outputs u(f)u(f) over a "prediction horizon."
3. The control signals that produce the most desired behavior are selected.
4. The control signal is implemented over a pre-defined time interval.
5. Time advances to the next interval, and the procedure is repeated from step 2.

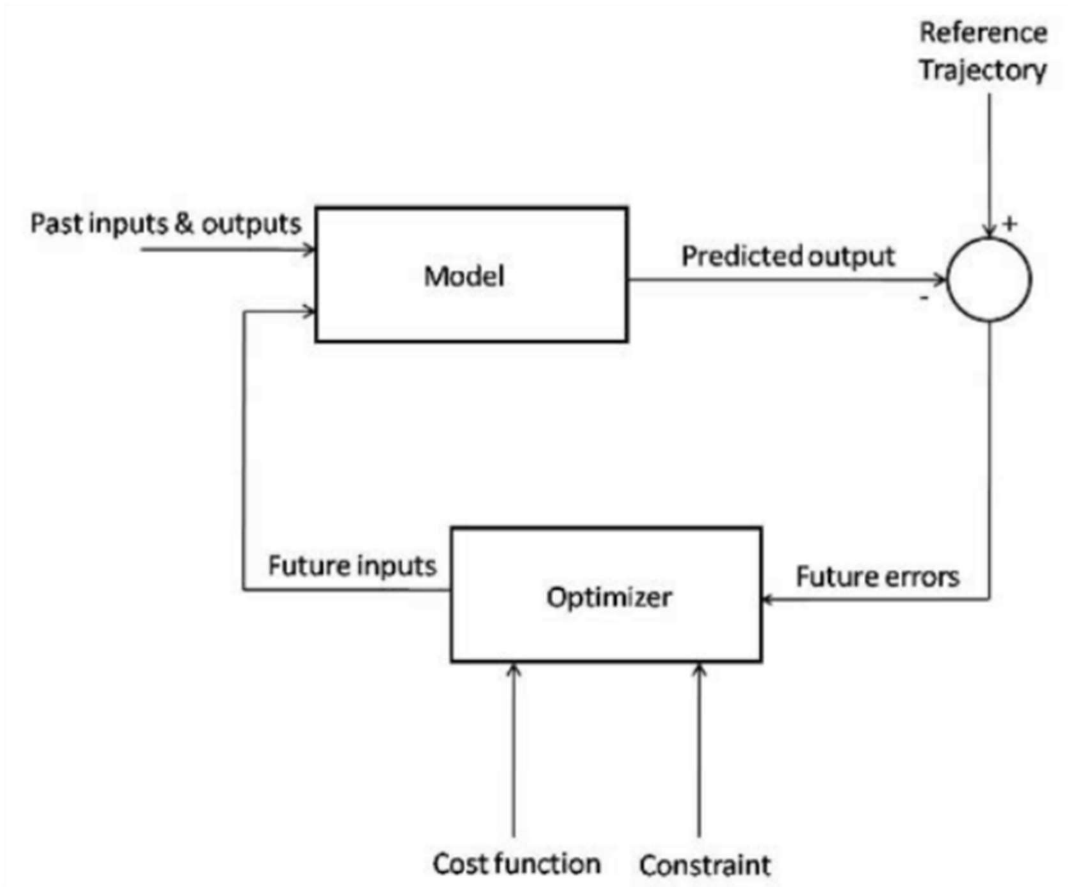
This is one of the many algorithm possibilities, which can be applied to systems with any number of inputs or outputs. The process model can be variable as well. Examples include physical models, input-output models, and state models which are all derived from the specific system being controlled.

When comparing predicted behavior to desired behavior, there are multiple techniques. A common procedure is to generate a second mathematical model that describes your desired behavior. When process behavior is predicted in step 2 of the MPC algorithm, the control signals which produce the predicted behavior that minimize deviations from your desired behavior over the [t,t+h][t,t+h] interval, are selected. The generation of models and optimization process is repeated continuously as the algorithm is repeated.



As seen in the figure above, depending on the algorithm, MPC may generate possible outcomes given possible controller action. These generations are either based on past process outputs, or the process model. After many possible outcomes are generated, the controller can pick one based on the optimization goals. This generation and optimization process is repeated at every time step.

The flow diagram below depicts the flow of information used by the controller.



The figure above shows the basic structure of a Model Predictive Controller. The model takes data from past inputs and outputs, and combines it with the predicted future inputs, and gives a predicted output for the time step. This predicted output is combined with the reference trajectory, giving the predicted future errors of the system. These errors are fed into an optimizer, which enforces the constraints of the system (for instance, ensuring that a flow rate calculation for the model is not greater than the maximum flow) on the predicted outputs and minimizes the operating cost function. This gives the predicted future inputs, which are fed back into the main model, restarting the cycle.

Motivation:

The motivation for the development and implementation of MPC controllers is compelling. MPC is very simple for sampled systems in which the control signal is constant over the interval [t,t+h]. The value of h can then be taken as the sampling interval, and the prediction horizon can become a small number of sampling intervals. This can reduce the computational power needed to implement a model predictive controller.

A highly desired feature of MPC is that constraints can be implemented in the controller. These constraints include heaters and valves that have a finite operating range, actuators with finite states (on/off or low/high), and cost or energy limits for the process. MPC can incorporate these constraints and eliminate the possibility of variables exceeding their limits. This helps the process operate efficiently, prevents damage to equipment, and prevents the system from running away by continuously increasing a variable's setting when the limit for the equipment has already been reached. Another benefit of MPC controllers incorporating system constraints is the ability of the model to dynamically react to system changes. For example, if a valve is stuck open, it can be added as a constraint, and the model will compensate accordingly. This allows the controller to continue effectively controlling the system after an equipment malfunction. In a simpler control, such as PID control, this would not be possible.

MPC is a highly specific method for controlling a process. Each controller is specific to the system it was designed for and the model equations, constraints, and set points will change for different systems. This means that a controller developed for a tank reactor will not be able to control an evaporation unit as the process model will be very different. This weakness in MPC brings out its biggest strength. The specificity and customizable nature of the controller will empower you with the freedom to design for exactly what is desired.

MPC is a flexible control technique that uses discrete time segments and is the most commonly applied advanced control technique in the chemical process industry. MPC helps to simplify or completely eliminate controller design and instead works as a system modeling controller. With MPC the designer does not have to worry about optimizing control parameters such as with PID control. The ability of MPC controllers to handle constraints in an optimal fashion is also a contributor to its success.

Model Predictive Control Example:

To demonstrate the concepts of MPC, a general example for the development of a sampled process and a first-order system will be shown.

General Model

Take Equation 12.3.112.3.1 as our process model:

$y(t)+a1y(t-h)+\ldots+any(t-nh)=b1u(t-h)+b2u(t-2h)+\ldots+bn(t-nh)(12.3.1)(12.3.1)y(t)+a1y(t-h)+\ldots+any(t-nh)=b1u(t-h)+b2u(t-2h)+\ldots+bn(t-nh)$

where uu is controller input, yy is process output, and hh is the time interval. This is a general equation that relates previous process output $y(t-h)y(t-h)$ and previous controller input $u(t-h)u(t-h)$. In some situations this equation may be created using fitted experimental data, but is most often a derivation using knowledge of your specific system and fundamentals of chemical engineering. At time t, total previous behavior y_p is shown as Equation 12.3.212.3.2.

$yp=f(y(t),y(t-h),\ldots,u(t-h),u(t-2h),\ldots)(12.3.2)(12.3.2)yp=f(y(t),y(t-h),\ldots,u(t-h),u(t-2h),\ldots)$

Future process output yfyf can be predicted using current and future control signals, $u(t)u(t)$ and $u(t+h)u(t+h)$ respectively:

$yf=f(u(t),u(t-h),\ldots,u(t+Nh))(12.3.3)(12.3.3)yf=f(u(t),u(t-h),\ldots,u(t+Nh))$

Both ypyp and yfyf could possibly be created by fits of experimental data, but are more likely to be derived from specific equations related to your system. Deviations from the desired behavior ydyd, either specified by another mathematical model or reference trajectory, produce an error function $e(t)=y(t)-yd(t)e(t)=y(t)-yd(t)$ for increments of control actions $\Delta u(t)=u(t)-u(t-h)\Delta u(t)=u(t)-u(t-h)$. The loss function JJ to be minimized is shown as Equation 12.3.412.3.4.

$J(u(t),u(t-h),\ldots,u(t+Nh))=\sum l=1t|Ne(t+kh)2+\rho(\Delta u(t+(k-1)h))2(12.3.4)(12.3.4)J(u(t),u(t-h),\ldots,u(t+Nh))=\sum l=1t|Ne(t+kh)2+\rho(\Delta u(t+(k-1)h))2$

The control inputs that minimize Equation 12.3.4 are then applied to the system by the controller over the time interval, and the process is repeated. The control input function F in Equation 12.3.5 is determined implicitly by the optimization.

$$u(t)=F(y(t),y(t-h),\ldots y(t-nh),u(t-h),y(t-2h),\ldots,u(t-nh)) \tag{12.3.5}$$

This general model is meant to be a guideline, and the equations listed representative, for the thought process required to create a model predictive controller.

First-Order System Example

Take the process model to be Equation 12.3.6:

$$\Delta y(t+h)=-a\Delta y(t)+b\Delta u(t) \tag{12.3.6}$$

Let us define $\Delta y(t)=y(t)-y(t-h)$ and $\Delta u(t)=u(t)-u(t-h)$. Let us also define our desired system behavior y_d as a function which starts at $y(t)$ and exponentially approaches a set point y_{sp} with time constant T. Our desired behavior y_d then becomes Equation 12.3.7:

$$y_d(t+h)=y(t)+\left(1-e^{-\frac{h}{T}}\right)(y_{sp}-y(t)) \tag{12.3.7}$$

Assuming that our controller can take as much action as needed to produce the desired behavior, the desired behavior can be realized in the next sampling period. This is done by setting $y(t+h)$ equal to $y_d(t+h)$, and can be seen in Equation 12.3.8:

$$y(t+h)=y(t)+\Delta y(t+h)=y(t)-\alpha\Delta y(t)+b\Delta u(t)=y(t)+\left(1-e^{-\frac{h}{T}}\right)(y_{sp}-y(t)) \tag{12.3.8}$$

Solving Equation 12.3.8 for $u(t)$ gives Equation 12.3.9:

$$\Delta u(t)=\frac{a}{b}\Delta y(t)+\frac{1-e^{-\frac{h}{T}}}{b}(y_{sp}-y(t)) \tag{12.3.9}$$

Upon examination of this result, you can see that we have produced a PI controller with gains $k=\frac{a}{b}$ and $k_i=\frac{1-e^{-\frac{h}{T}}}{b}$. It should be noted that the proportional gain *k* will only depend upon the developed process model, and the integral gain k_i depends on both the process model and the desired response rate *T*.

This process can be modified to include multiple inputs and outputs via the process model and desired behavior. We may also add system constraints to our MPC example by adding conditional statements limiting the maximum controller output discussed in the logical programs wiki. As discussed before, there are as many variations on the MPC process as you can think of. See the worked examples 1 and 2 for more MPC instances.

Differences from Other Controllers Types:

Model predictive control uses a mathematical model to simulate a process. This model then fits the inputs to predict the system behavior. In this way, MPC is a type of feed forward control. It uses system inputs as a basis of control. MPC is more complex than most other feed forward control types because of the way these predictions are used to optimize a process over a defined amount of time. Most feed forward control types do not take into account the process outputs much past a residence time. The MPC algorithm will compare predicted outputs to desired outputs and select signals that will minimize this difference over the time selected. This control type can see ahead into multiple time steps in the future in order to optimize the process. Normal PID type controllers use mathematical expressions based on error from a set point. The governing equation for MPC controllers are based on set points, system properties, and desired outcomes and optimization.

MPC is very specific to the process it is modeling. Unlike ratio or cascade control set ups, where it is simple to implement and change set points in various situations, MPC will model one specific process and optimize it. As previously mentioned this can either be an advantage or disadvantage. MPC is great for selecting one type of operation on one system and perfecting it to the desired conditions. This also has a downside in that the model equation will work for one and only one situation.

Advantages of MPC:

1. MPC can be used to handle multivariable control programs.
2. MPC can consider actuator limitations.
3. MPC can increase profits by allowing for operation close to the system constraints.
4. MPC can perform online computations quickly.
5. MPC can be used for non-minimal phase and unstable processes.
6. MPC is easy to tune.
7. MPC is able to handle structural changes.

Disadvantages of MPC:

1. Several MPC models are limited to only stable, open-loop processes.
2. MPC often requires a large number of model coefficients to describe a response.
3. Some MPC models are formulated for output disturbances, and they may not handle input disturbances well.
4. Some forms of MPC use a constant output disturbance assumption. This corrects for the fact that the output predicted by the model is not exactly equal to the actual measured output. This method assumes the correction term is constant in the future, which may not yield a good performance if there is a real disturbance at the plant input.
5. If the prediction horizon is not formulated correctly, control performance will be poor even if the model is correct.
6. Some systems have a wide range of operating conditions that change frequently. Some examples of this include exothermic reactors, batch processes, and any systems where different consumers have different product specifications. An MPC linear

model will not be able to handle the dynamic behavior of these processes. A nonlinear model must be used for better control performance.

Industrial MPC Applications:

There are many industrial applications that incorporate model predictive control in order to effectively control a multivariable system. In order to effectively do this, one needs to set up a working model by testing many different parameters in a plant. This is usually done by starting up a plant, varying many different parameters, and having the MPC program analyze the data.

In order to test the plant, one may vary parameters such as:

1. feed flow rate/composition
2. steam pressure
3. heat duty
4. recycle ratio
5. reactor temperatures

This is only a small sample of parameters that can be changed. Once these parameters have been changed, the data is analyzed, and downstream effects of these parameters are characterized as a function of these variables. This relationship can be a combination of many different relationships (linear, nonlinear, logarithmic, exponential, power, etc.).

By testing the plant thoroughly and coming up with a robust model, the engineers ensure that an MPC controller will be able to much more effectively run the plant. Barring any major process changes, this model should be accurate for normal use.

Some industrial MPC applications are:

1. Model Predictive Heuristic Control by Richard et al. 1976 (Adersa)
2. Dynamic Matrix Control (DMC) by Cutler and Ramaker 1979 (Shell Oil)
3. Quadratic-Program Dynamic Matrix (QDMC) Control by Cutler et al. 1983 (Shell Oil)
4. IDCOM-M by Setpoint, Inc (part of ASPEN Technology)
5. Generalized Predictive Control (GPC)

2-MIMO using RGA