CONTROL SYSTEM FUNCTIONS

Graphical O-O based control design

The Application Developer Software module uses graphical O-O based control design techniques to help develop complicated control schemes using very little engineering time. Several elements contribute to the speed of development using MANTRA 47.

◆Complex Pre-canned Control Algorithms

Control blocks such as auto-tracking self-tuning cascade PID, self-tuning IMC control, time proportional block, discrete device control, etc., allow rapid development of control schemes when compared to traditional engineering time spent on PLC or DCS systems.

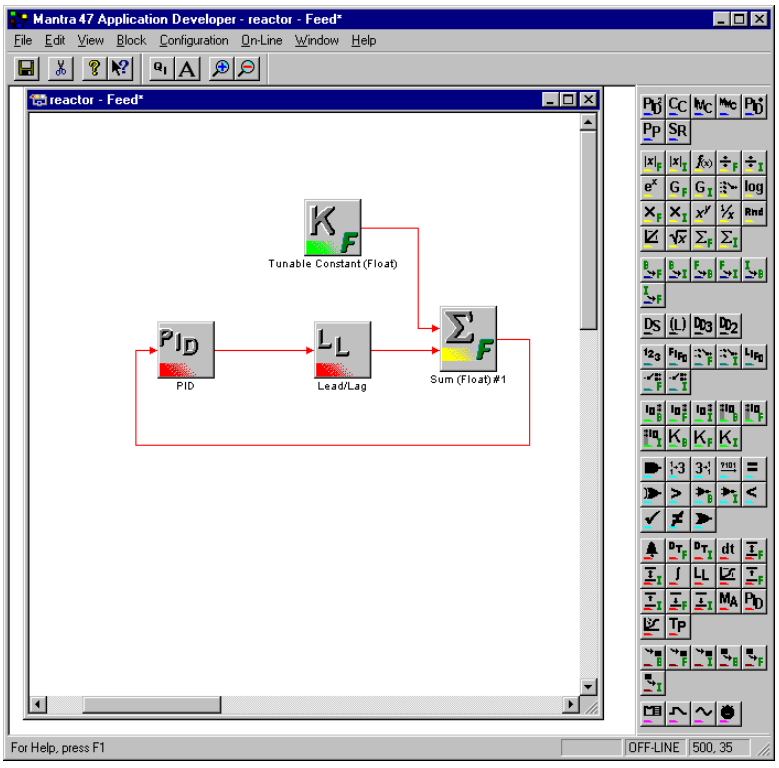
Instead of stringing together several control blocks, the user picks a single advanced control block from the MANTRA 47 library to achieve the same objective.

◆Object-oriented Drawing Tools

The MANTRA 47 Application Developer provides a function block drawing tool, which uses object-oriented methods to build the program as you draw.

Clicking and dropping a function block object on the control palette implies simultaneous development of the program. In fact, the drawing is the program.

There are no files to compile, no autocad files to debug: this unique approach minimizes the error made in translating a control function drawing to the actual program.



◆Control Function Block Objects

Since each block is an object, the control design engineer can easily manipulate the input and output dimensions and block parameter characteristics to suit his or her particular application requirement. This minimizes the need for a large library of blocks, since each block in a particular class can now serve a wide range of control functions.

Control Function Library

MANTRA 47 offers several control algorithms and I/O function blocks organized in appropriate block submenus.

Advanced
- Cascade PID
- CC 1x3
- IMC 1x1
- MMC 2x3
- Multiple Parameter PID
- Position Proportional
- Split Range

Arithmetic
- Absolute (Float)
- Absolute (Integer)
- Calculate
- Divide (Float)
- Divide (Integer)
- Exponential
- Gain (Float)
- Gain (Integer)
- General Selector
- Logarithm
- Multiply (Float)
- Multiply (Integer)
- Power
- Reciprocal
- Round
- Scaling
- Square Root
- Sum (Float)
- Sum (Integer)

Data Conversion
- Boolean to Float
- Boolean to Integer
- Float to Boolean
- Float to Integer
- Integer to Boolean
- Integer to Float

Event
- Counter
- FIFO
- Input Selector (Float)
- Input Selector (Integer)
- LIFO
- Output Selector (Float)
- Output Selector (Integer)

I/O
- Input (Boolean)
- Input (Float)
- Input (Integer)
- Output (Boolean)
- Output (Float)
- Output (Integer)
- Tunable Constant (Boolean)
- Tunable Constant (Float)
- Tunable Constant (Integer)

Logical
- And
- Bit Combine
- Bit Extract
- Bit Shift/Rotate
- Equal
- Exclusive Or
- Greater
- Invert (Boolean)
- Invert (Integer)
- Less
- Limit Check
- Not Equal
- Or

Discrete
- Drum Sequencer
- Latch
- Three State Discrete Device (DD3)
- Two State Discrete Device (DD2)

Process Control
- Alarm
- Deadtime (Float)
- Deadtime (Integer)
- Derivative
- H/L Limit (Float)
- H/L Limit (Integer)
- Integrator
- Lead/Lag
- Linearization
- Maximum (Float)
- Maximum (Integer)
- Minimum (Float)
- Minimum (Integer)
- Moving Average
- PID
- Rate Limiter
- Time Proportional

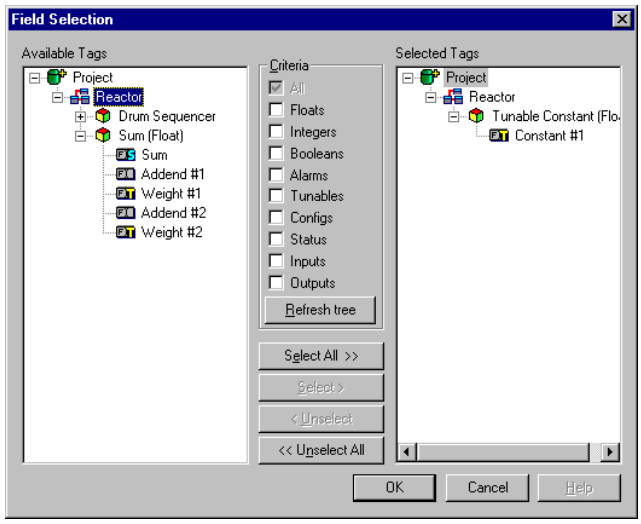
Special
- Input Connector (Boolean)
- Input Connector (Float)
- Input Connector (Integer)
- Output Connector (Boolean)
- Output Connector (Float)
- Output Connector (Integer)

Time
- Date and Time
- Profile Generator
- Sine Generator
- Timer

Watch Windows

The Watch Window in the Application Developer software allows the user to view any parameter and modify tunables in a previously designed control scheme, without having to access the database.

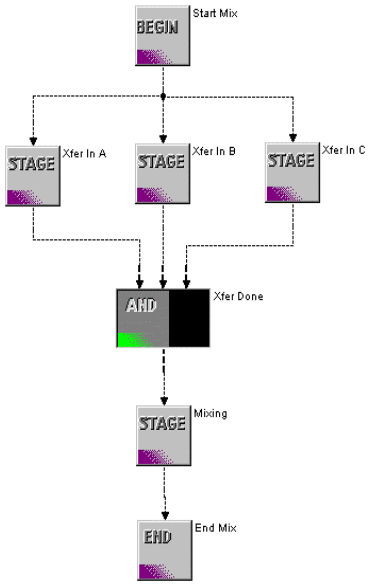
This unique approach, using a Windows Explorer type selection tool, allows the database to be transparent to the user. If the user does not know a tag name, the Browse and Add functions allow the user to highlight and select a parameter and drag it to a selected tag window.



Sequence development tools

For process control applications which involve a lot of batch control operations, traditional function block control strategies may not be very efficient. What this kind of an application needs is a programming tool that directly translates phase and state logic from the process world to a control language.

MANTRA 47 offers a very powerful tool at the control level to address batch applications. Coupled with the MANTRA 47 HMI Batching management software, the sequence development tool allows for handling of virtually any batching and recipe management operation.



The MANTRA 47 system offers batch sequencing blocks (as an option) to develop the Batch Sequencing Logic (BSL) diagrams that are used to control the execution of operations and phases in batch applications.

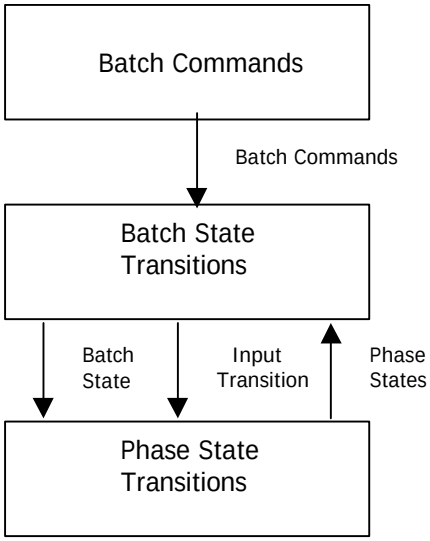
A BSL is part of the application logic that is comprised of phase logic and resides in the process controller. The BSL receives commands either from the MANTRA 47 HMI (Batching and Recipe Management package) or from application logic, and generates appropriate phase commands to activate different components in the phase logic.

Behind the BSL, it is the state transition logic that controls the transitions between the states and activates the different components in the phase logic.

A BSL has two levels of state transition logic. The top-level is the master transition logic for the batch state. The low-level is the slave transition logic for the phase state. The phase state transitions follow the batch state transitions.

Based on the current batch state and commands received, the master transition logic determines what is the next batch state to transition to, following the Batch State Transition diagram.

The slave transition logic is used to determine the transitions for the phase state. Based on the batch state and input transition, the phase state follows the phase state transition diagram to change its state from one to another. Each phase has its own state.



In the application project, you can create more than one BSL. This allows you to break a batch procedure into small, manageable operations. For each operation, you create batch sequencing logic to control the execution of phases. Then, you arrange them according to procedure hierarchy in a large operation.

Advanced model-based control

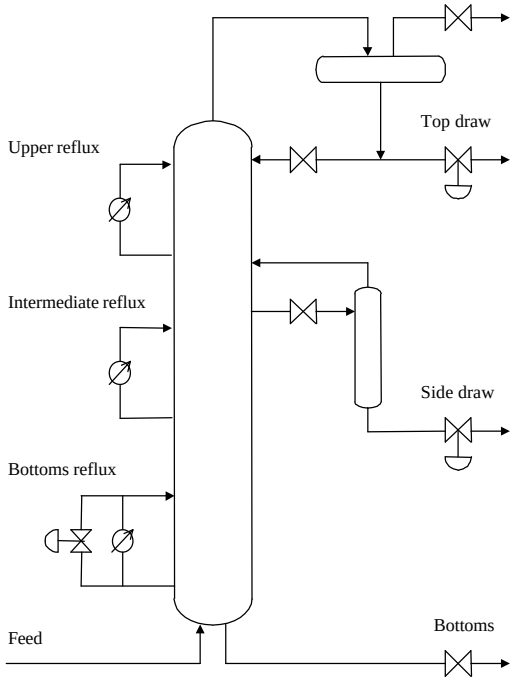
The model-based advanced control suite offers three different control blocks to address a wide range of advanced control applications.

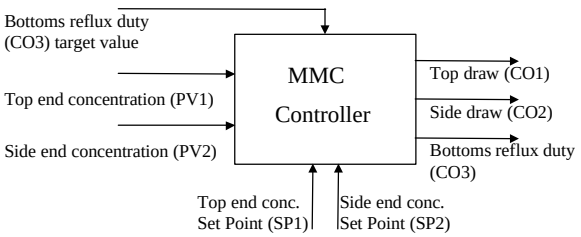
The three advanced control blocks are:

(a) IMC - internal model control
(b) CC - coordinated control
(c) MMC - modular multi-variable contrc

These control blocks provide an effective means of controlling one or more processes to their respective setpoints using two or more control variables. These advanced control algorithms take into account the interactions between processes, and allow a user to define optimization criteria and incorporate control output constraints.

MMC Control of Heavy Oil Fractionator





For applications which suffer from a large dead-time (transport delay) problem, consider using the IMC instead of PID. Available for use (as an option), the IMC control block can be connected in conventional control schemes and auto-tuned to be used just as easily as a PID.

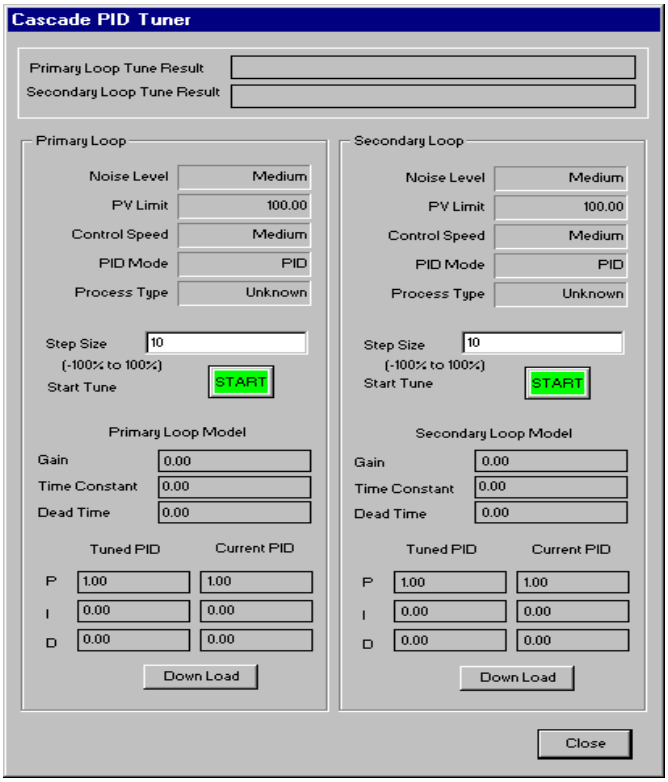
For large applications where several control values need to be manipulated to maintain a single process within a given tolerance, a CC control block may prove superior to multiple PID-based schemes. A CC block is very effective on systems such as fermentation control, split range or feedforward temperature control systems, constraint control, etc.

An MMC control block is more complex than a CC or IMC, and is typically used on multivariable process systems, where the primary goal is to decouple the process responses and provide some degree of economic optimization.

All the above control schemes are completely integrated within the MANTRA 47 system, and offer benefits like self-tuning, auto-manual control, tracking, etc. These blocks also come with their individual faceplates, which allow development of quick graphic interfaces to tune and commission such control schemes.

Integrated auto-tuning

Auto-tuning, a system feature which saves loop commissioning time, is often the missing link in many modern control systems.



MANTRA 47 offers a very powerful Auto-tuning algorithm which, over many years, has proven to be a very efficient and easy way to commission PID control loops.

MANTRA 47 also offers auto-tuning features on all other advanced control blocks, such as cascade, IMC, CC, and MMC. Availability of auto-tuning on advanced model based control algorithms is a very unique feature of the MANTRA 47 control system, which ensures that complex control schemes using such blocks can be efficiently commissioned on-site.

Function Block Faceplates

Every closed loop control and discrete control algorithm in the MANTRA 47 control system has its own associated display faceplate, which makes life easier for the user who has built display graphics for every loop that needs to be commissioned.

Such ActiveX-complaint display faceplates can be inserted within a MANTRA 47 GUI container, and connected with a few clicks to the control block of choice. This eliminates the need to understand or interpret the underlying tag database structure, needed for sophisticated HMI software development.

Faceplates allow the user to have quick run-time views of one or many control loops or discrete control blocks, which translates to a huge saving in engineering development and loop commissioning time.