MHEMPC Basics

Tutorial and Description

Buddhadeva Das

Copyright © 2019 Buddhadeva Das

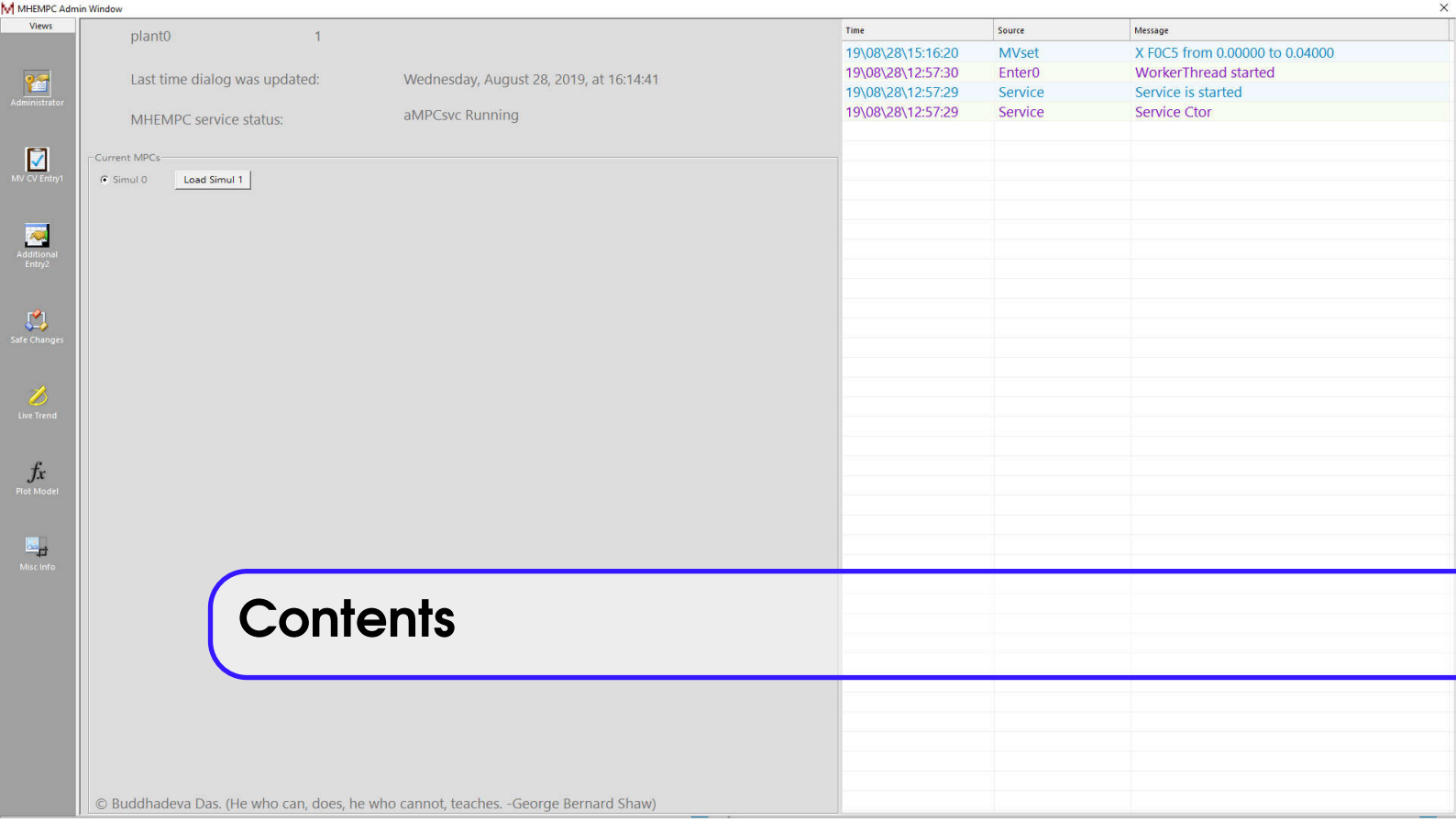
PUBLISHED BY MHEMPC, COMMITTED TO EDUCATION ON APC USING MHEMPC

MHEMPC.COM

This document completely describes MHEMPC demo product. Actual MHEMPC connected to the plant is same as the demo, the simulated plant used in demo is replaced by I/O software connected to the plant DCS.

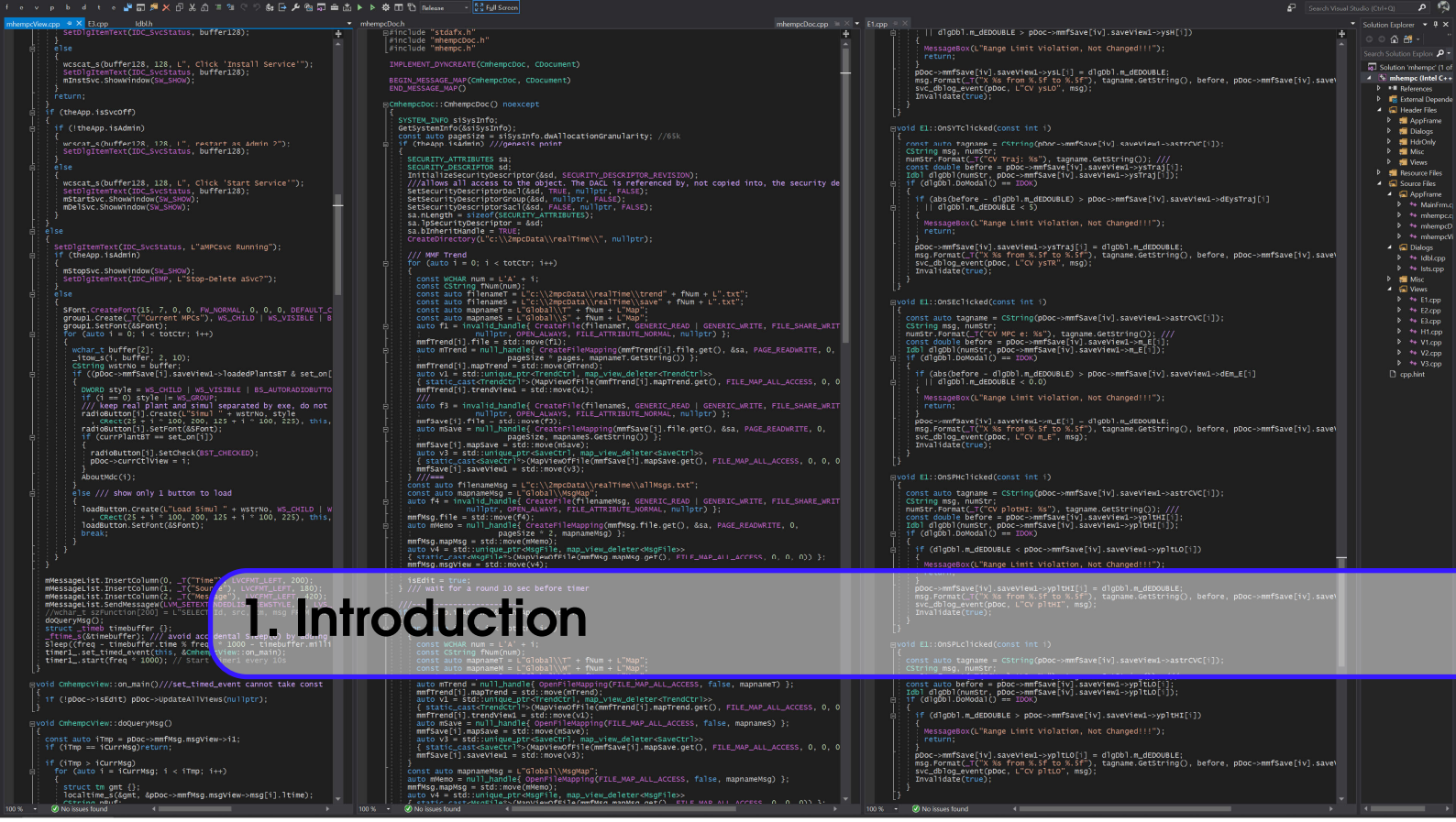
Performance in actual plant will be similar to that observed at simulation.

2nd printing, Sept 2019, replaces August 2019 version



Contents

Part One, Software Architechture



1 Introduction

This document describes basics of the product MHEMPC. A 2nd document will accompany this as a part-2 of this document and will be shipped with the actual product, when purchased. Part-2 will be titled "MHEMPC Advanced", and will contain everything else that are not included in the basic document.

At the end of this we shall be able to learn 60% of the available features. More inquisitive learners can also explore into featues that are not described in the book. Tutorial begins at chapter 2. This chapter is a discussion only.

1.1 In the beginning...

The 60s were the age of kilohertz computers. Using linear state space models and Kalman Filtering, multivariable controllers were guiding the path of space module. The most critical part was the timer to maintain a regular interval of control subroutine execution and the ruby crystals were not very accurate. And every other control system that could not afford a billion dollar computer was using pneumatic systems, which we still use for flow control valves in chemical processes. And pneumatic systems were not very ancient either, they were first deployed at war machines in WW2.

Late 80s saw megahertz computers, moderately pricey. VAX/VMS systems with virtual memory, accurate timers, became an instant hit for refinery applications. From once in 8 hours of executing an optimization program to once a minute FCCU IDCOM, the decade was busy with innovations around MicroVAX.

All that changes as we approach 2020. For 1% of the price above we have gigahertz range computers. Yet, after the flurry of activities on control algorithm developments of 80s, we stopped innovating. The best algorithms were already written and they could complete execution within a minute. Now it was time to sell, but alas, the climate suddenly started changing and all these chemical plants would disappear very soon. So Chemical Engineering became Biochemical Engineering and then Environmental Engineering.

Not anymore, Oil and Gas is back, so is Model Predictive Controller.

1.2 Redoing the MPC

Model Predictive Controllers of the 80s were focused on the problem of saving computation with approximations so that some valid and useful optimal condition could be calculated.

Not anymore, all CPUs today have very highly accurate timers and multi-core, gigahertz speeds. SSDs today are faster than RAMs of yesterday and RAMs today are faster than CPUs and CPUs are getting hundreds of cores. Since MPC workforce has moved to a salesforce, they are stuck with programs of 80s. They move to x64 CPUs but run on a simulated VMS OS on a non-VAX hardware. Some FORTRAN code are compiled for for Win32 and deployed on Windows. Programs that were taking a minute time now take a millisecond.

And yet, the code of 80s were the ultimate, could not be improved (or should not be, lest we lose all that revenue). And in this era of internet, people have forgotten that programming used to solve mathematical problems. It is all HTML, Java, C# and webpages with JavaScript. If mathematics is necessary, just use Python or MATLAB.

That is where MHEMPC comes in, because we think, we can do better. Even C++20 standard is so superior to any other programming language that it now takes less code typing to achieve the same in any other language.

1.3 Perennial Tuning

Since MPCs of the 80s were using approximations, they relied heavily on tuning. Each used one tuning paradigm, diametrically opposite to what the competition was using. Anyone remembers "move suppression vs reference trajectory"? Why could not they have both? Well, that would have again increased computation.

Not anymore, MHEMPC uses the most generic form of control, with the assumption that computational resources are infinite and therefore no approximation is required. It defines the control objective as a non-linear least-square optimization which can take calculated constraints. Process models ranging from state-space to non-linear ODEs can be used for predicting the future so that moves can be calculated and deployed with a goal of moving to the best possible future.

A Trust Region algorithm based Optimization routine finds better optimal future than the assumption that it is one of the corners of intersections of the constraints. Hence control moves are smoother. Also all combinations of tuning procedures are available together. Weights on MVs, CVs along with trajectories on CVs and MVs are available. IRVs can come from wisdom, spreadsheet run or another steady-state optimizer. Very often they are one directional, more feed or more gasoline and less bottoms.

IRVs were used in IDCOM, it stands for "Ideal Resting Value" for MVs. This is also available in MHEMPC (oops... did anyone patent that too?). When the plant is very stable, all setpoints and constraints on CVs are fully satisfied, it will tend to move MVs towards IRVs until CVs start to complain again. This is steady-state optimization.

We soon see that default values do work good enough on this demo and probably will work for for some real plants too, if all CVs are temperatures and all MVs are draw rates. But a composition is definitely not a temperature type.

1.4 MHEMPC Demo

Since non-linear systems are the defining principle of this MPC, it comes with a non-linear offset-free mechanism based on MHE (Moving Horizon Estimation). The demo uses a simulated plant based on classic first order lag and dead-time, appropriate for 95% of chemical plants.

The process model is also convolution structure based, as used in classic MPCs. This is to make transition from an existing failed MPC to MHEMPC smoother.

However, those plants that cannot have classical MPCs because they are highly non-linear, now can benefit from MHEMPC too. Modular code based on C/C++ used in MHEMPC makes it easy to interchange prediction models of wide range of complexity.

It can even take state-space based multi-variable plant models and run in a few millisecond interval. ARM processor based tiny motherboards can host Windows 10 full version too.

On the higher side, instead of moving to one large dimension MPC, multiple cooperating MPCs are recommended. Multi-core CPUs are capable of running hundreds of threads simultaneously. All that CPU power goes waste in a classic MPC. MHEMPC exploits the power of C++ threads.

Tuning is still necessary. WIth a very accurate simulator, we can simulate the plant using its appropriate model and run MHEMPC on it. We can test various target conditions, conflicts, instabilities and tune the simulated MPC. These tuning can be transferred to the plant MHEMPC.

1.5 So, let's get started...

Without further ado, let us get started with MHEMPC. Any improvement on the final product working in the plant online shall be first implemented on this demo and uploaded for everybody to download and try and give feedback. Hence the demo is the best way to learn, adopt and contribute. At an actual deployment we just remove the simulated plant from demo and attach a data I/O software to control the plant.

The goal of this demo is to generate confidence in the minds of the team members responsible for process optimization, on MHEMPC as a potential Advanced Process Control software. What works on the demo, shall work on the real plant the same way. All different authorities that are required for issuing clearance on a purchase of this product can also try this demo for reliability tests.

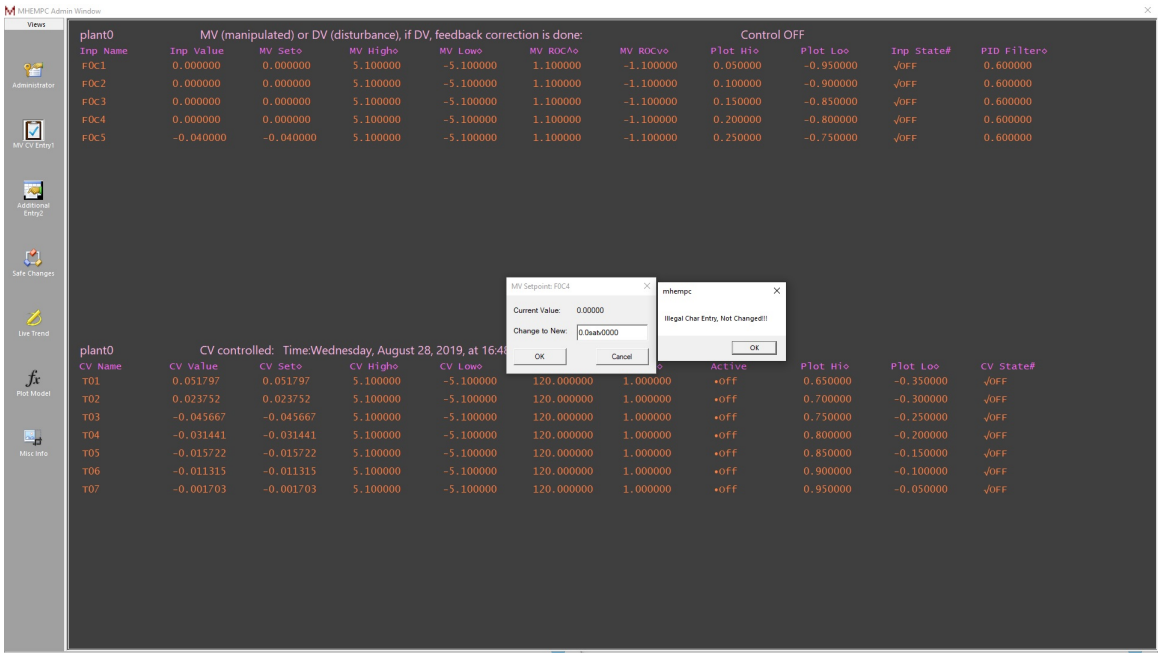


Figure 1.1: Bad character entries are protected.

1.6 Wrong entries are detected

Once a while we make wrong a entry in a Windows based software. It often interprets that entry as 0.0 and does unwanted things. Even Fire and Safety department wants to know if the on-line

software has such protection against typos by user. Figure 1.1 is an example where illegal characters are detected. Also we have an entire page devoted to safe δ entries.

1.7 Anybody can Implement APC

Since 1990, early days of APC, it had been a dream that MPCs could be implemented by anyone. However the existing vendors never made their software affordable nor user-friendly, so it never happened. MHEMPC is the first MPC that has scored in both the counts. It has a very quick learning curve, anyone operating a running chemical plant can self teach MHEMPC and quickly adopt.

That is why, a great care is taken while writing this tutorial, so that anyone can implement MHEMPC.

1.8 Data I/O Software first

Online process data input from the plant to MHEMPC and setpoint output to the plant is known as "Data I/O Software". It depends on the DCS or any other front-end electronic hardware that the regulatory PIDs use. A driver routine is a Win32 library DLL that gets linked with MHEMPC service to do the real work. This has to be supplied by the vendor of this front-end hardware supplier. Also there are OPC driver suppliers. A sample Visual C++ console program can be created to test the driver. So Visual Studio 2019 can be downloaded for evaluation (nopurchase is required).

2. Getting Started

2.1 Hardware and Software

Choose a modern x64 computer. Either Intel i7 or AMD Ryzen5 is minimum requirement. The whole system is developed and tested using an AMD Ryzen 5 1600, RAM 32GB, 4gHz, SPCC M.2 PCIe SSD. The more CPU cache and faster SSD, the better. The demo is configured to run 2 MPCs together on this hardware.

The graphics requires an 1080p monitor. Also install a fresh Windows 10 OS, with latest updates. Internet connection is not required for MHEMPC and demo does not use Intranet LAN.

> **Tip 2.1** Please avoid...
> - Do not use personal laptops.
> - Remove other unused hardware and software.
>
> ■

2.2 Installation

Copy 1mpcServer, 2mpcData, 3OffLine and 4HtmlHelp as directories under `C:\`.

2.3 Start

For one time, start the `C:\1mpcServer\mhempc.exe` as Administrator.

> **Tip 2.2** Open File Explorer.
> Right click `C:\1mpcServer\mhempc.exe` to see option for `Run as administrator`.
> The minimal screen comes up. ■

Administrator privileges are required to start and run MHEMPC. Therefore, get a dedicated x64 computer to test this software.

Try 2.3.1 Click `Install Service`. The button disappears and `Start Service` appears, which we click. Watch messages appearing above.
Close `mhempc.exe`.
Start `mhempc.exe` again normally by clicking on the file.

More Info 2.3.1 — Service Program. Windows services are background programs. Right click on the Windows icon of Taskbar and click `Search`.
Type `Services` to search and Services appears. Click `Services`.
This gives the list of services that are already running in the CPU as background programs. See " `aMPCsvc Service`" as an entry there. This is the service just installed by us. Installing and removing the service are to be done with Administrator privileges.
Do not remove the service until finish of demo.

If the service is not yet installed and `mhempc.exe` is clicked, a help HTML page appears, prompting to click to open this tutorial. Upon service installation `aSvc.exe` is loaded as this background program.

2.4 MHEMPC Admin Window

This is a multiple view GUI. On opening normally, all the buttons to browse the views appear. Go to "`MV CV Entry1`". Any value that can be changed by clicking on it is accompanied by a diamond sign. Watch "`plant0`" at the top corner. This is the name of the plant that we are working on and it is configured by default when service started by loading `C:\2mpcData\0.csv` without user doing anything.

Go to "Live Trend". Watch the plant values appearing to refresh on the screen automatically. The simulated plant values for inputs and outputs are all zero.

To see the second plant running, we may copy `C:\2mpcData\0.csv` to `C:\2mpcData\1.csv` and text-edit to change plant name, MV/CV names. MHEMPC expects the configuration files strictly with consecutive numbers, to avoid too much user interaction about loading (and sometimes misloading) configuration files.

2.5 A Word on GUI program mhempc.exe

There is a delay in starting the program because it tries to synchronize with the service.

2.6 Why Zero?

Normal values that we see in a real-time dynamic process plant are seldom zero. For control purposes we use base or nominal value be subtracted from actual. So When `F05` changes to 0.1 it may mean that actual flow may change from 127.0 to 127.1, if base as 127.0.

Try 2.6.1 Let us change F05 on page "`MV CV Entry1`". Under `MV Set` column click on 0.000000 for `F05` row and a dialog box helps to change it to 0.04.

Remark On "`Live Trend`", watch the process trend. Some outputs react, but slowly to the single move that was done. They have a process lag. Some are not immediately moving, but wait for a period of time, aka, dead-time, to show up responses.

Yes, we just did a step test on the simulated plant.

2.7 All in the Server Computer?

It may seem very inconvenient to work on the server computer on a real MPC project. This means on a real implementation, a work-group consisting of more than 1 person will have to use the same console. To mitigate this problem, an offline GUI program is designed. When using this demo, this application uses the same computer as the demo computer. But in an implementation project, a LAN intranet is a useful tool. By just configuring the `C:\2mpcData\realTime` directory of server computer as `L:\` of any client computer on the intranet, a member of the group can run `c:\3OffLine\Offline.exe` locally.

No internet connection is required. Since the software does not use any "Off-the-Self" database product or any other 3rd party product, it is impossible for any hacking to take place, even if server computer is connected to internet. Updating Windows 10 software for latest updates are not required after 1st installation because MHEMPC product will not need any of those features geared towards safe internet.

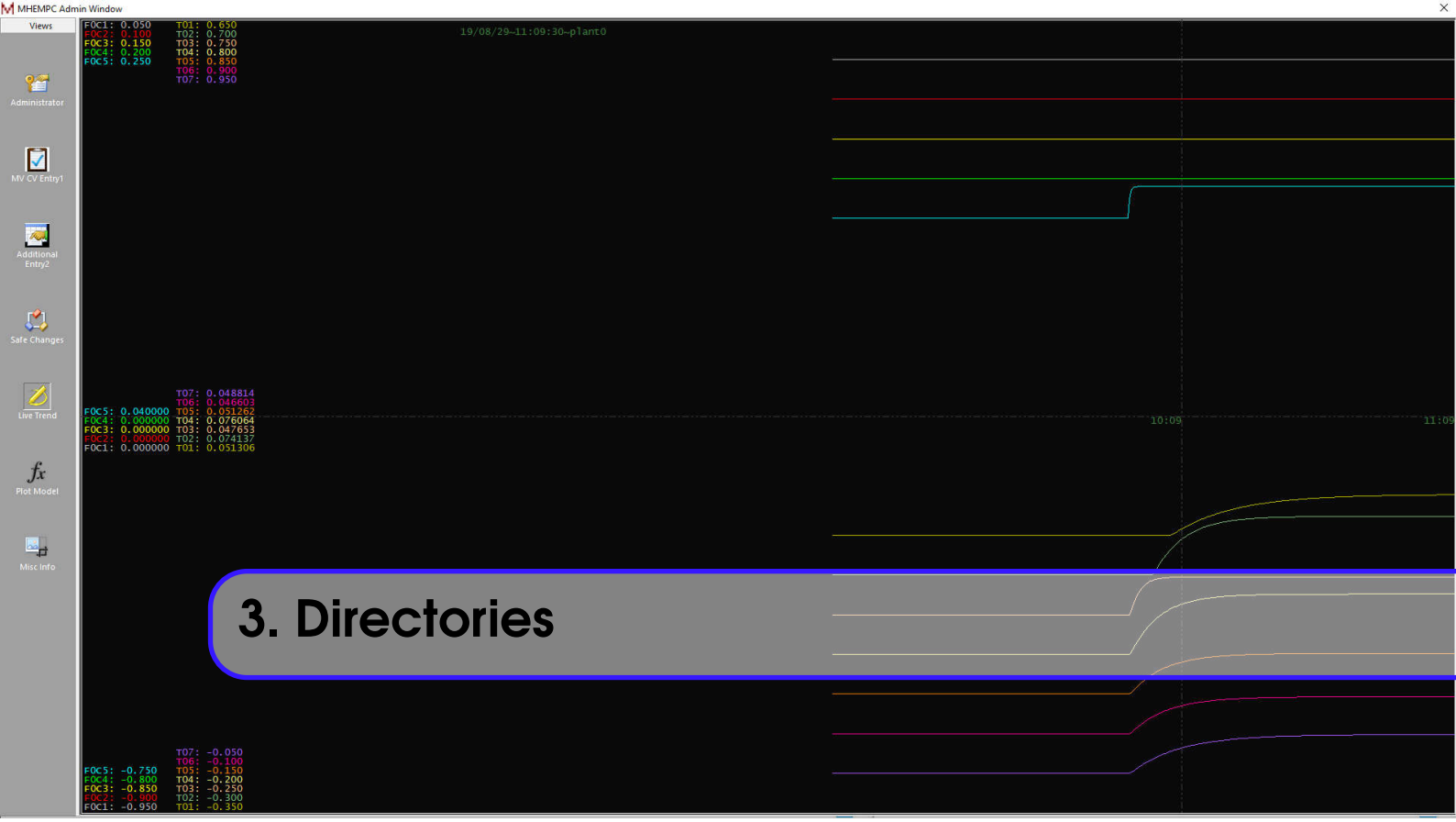
Try 2.7.1 Run `c:\3OffLine\Offline.exe` once. Close it, `L:\` is configured.
Run `c:\3OffLine\Offline.exe` again.
Click "Plot, Choose Id Segment" tab we can see the process values plotted offline. This program takes a snapshot of the realtime database and plots it without refresh, because it is not online. This is where all the actions like process modelling takes place.

2.8 Summary

Did it feel easy? This first overview can explain why MHEMPC software is just as easy as any MS Office software to operate. Mathematics constitutes only 5% of the entire codebase. A lot of care is taken as to the correct design of the whole system.

The coming chapters are to explain every step of MHEMPC. At the end of this tutorial we all shall be able to implement this on a real plant, only by replacing the simulated plant by data I/O software. Familiarity with all features will create confidence on MHEMPC. So put your Chemical Engineer instincts on hold for a while and learn all the bells and whistles of this friendly software.

Welcome to MHEMPC. The goal is to bring after-sale service calls down to zero. Distribute this demo to as many interested people as possible. However a commercial installation binary files will not work, if copied from a real licensed one.



Now, 1mpcServer, 2mpcData, 3OffLine and 4HtmlHelp as directories under `C:\` is the complete world of MHEMPC as demo. If they need to be removed, run `C:\1mpcServer\mhempc.exe` as administrator, choose `Stop`, then `Delete`. This removes the service from our computer. Then just delete the above directories.

If restarting the simulation is required, `Stop -> Delete` the service. Then delete all .txt files at `C:\2mpcData\realTime\`. These are not really text files, however, .txt extension allows them to be previewed like one.

Sorry I missed `L:\`. Let us understand that first.

3.1 The L:

On File Explorer, go to "This PC", see, there is a `L:\`.

When we ran Offline locally, it queried and found that `C:\2mpcData\realTime` directory of demo computer is not configured as `L:\` for accessing. So it runs "`subst L: c:\\2mpcData\\realTime`".

Watch your step test as in fig 3.1. It is a snapshot of `mhempc`.

> **Not Yet 3.1.1** The implementation this configuration will change on a LAN and can be an input string in a text file.

This directory is then accessed "remotely" by Offline and all *.txt files there are copied to `c:\3OffLine\SnapShot`, which are then read for further activities. The design of `Offline` enables a team member working in another floor to do a lot of engineering activities, like:

- Watch the process for last 10 days (not easy at server).
- Select data segment, do Process Identification.
- Manage models, create final model.
- Place the model in the right directory at the server.

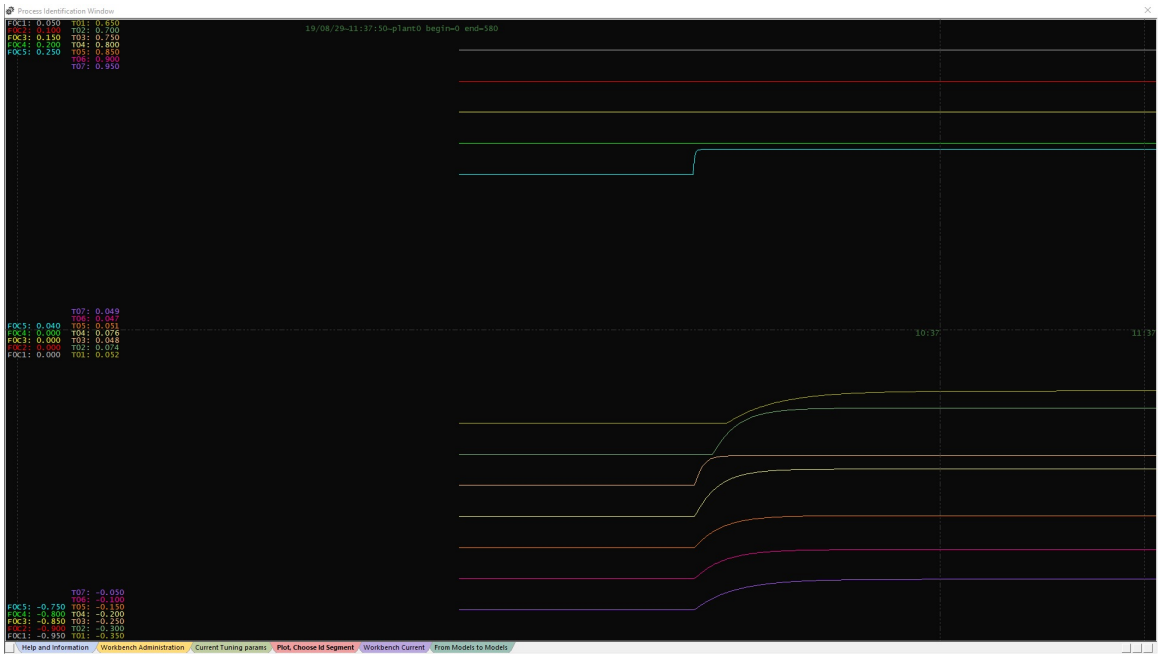


Figure 3.1: Step test view, Offline

3.2 mhempc again

Run `mhempc` again. Although it is really MHEMPC Admin Window, we shall refer it by its nickname `mhempc`. Last time we moved `F05` to 0.04. And watch the "`Live Trend`". This is how step test is done.

Remark A lot of opinions are available about doing (or not doing) step testing on a real plant. But to hone the skills,

- Do not use a dynamic simulator of the real plant as a model generator (period).
- A fairly steady state plant is good, so wait for it. Demo plant has a 1% noise.
- Do plan step test during the day, 10am to 4 pm best time (I did not mean this demo).
- Consult operating crew, explain goal and involve them in the plan. They are our costumers.
- Capturing all possible operating conditions, like night and day, is not necessary.
- Do not come on a night shift. Panel operator do not like it, Fire and Safety may not give permit.
- Start the step move at a steady state, process moves to another steady state and stay there.
- Ask the panel operator how much he/she can go up with that MV. Do that step test first.
- Next step is to ask if the same MV can be go on the negative direction by the same size. They often agree.
- Next step is to normalize the MV baack to nominal value. 3 steps are already done, 2nd one being twice as big as 1st, operator ensures safety. Plant manager is happy as thruput is not affected.
- Just one good step test is better than a dozen of bad ones. (Kiss the prince first, not all the toads.)
- If the operator says it is a good step test, it is a good step test. Now he/she feels inclusion, which later translates to easy transfer of ownership of the product.
- Raw model length is 360 sample points, minimum of 720 sample points necessary for identification.
- Identification software is completely multivariable, so test ordering is not important.

- Disturbance variables can have their models too. Demo is not ready yet for backoff modeling.

So we are currently doing a step test on our simulated plant on `F05`.

3.3 Database

Closing mhempc does not make the plant data disappear. The server maintains its database for all purposes internally. Since process database is not relational, using any SQL server is an overkill. All OS facilities that an SQL server uses is also used by our server, making it as robust as any commercial database server without relational stuffs. It maintains a daily worth of data in an "in-memory" database with MPC math accessing values as local variables. At the end of the day it spits out one full day copy for history, which `Offline` is able to read in contiguous fashion for 10 days. The data interval selected as segment can also be exported to a text file for use by MS Excel. This opens the window of using the process data by any external app.

> **Not Yet 3.3.1** The demo software only enables reading 10 days data starting from now. Final online version will have any starting day in the past, making it a true historian. Does anyone need more than 10 days of data in one sitting?

3.4 Database Files

In `L:\` one file `allMsgs.txt` captures all messages coming from server and mhempc. When it fills up with total capacity of 2047 messages, it overwrites the earliest one, operating as a circular buffer. So `Offline` comes handy. Also saveA.txt has all the live global variables used for plant0. (Anyone remembers the Global Commons used in VMS? Our database is similar to SETCON, but tightly bound and invisible to user. A fully blown SETCON?? any takers??) All process history for current day is hold in trendA.txt for plant0. Although plant-name plant0 can be changed, MPCs are enumerated by capital letters and are rigidly fixed, with a suffix "A" for the first one.

If we decide that this server is going to host 7 MPCs, all infrastructure for supporting all of those 7 MPCs are created at the starting of the server. That includes "in-memory" databases. They sit as pages in CPU cache hierarchy, RAM. OS fixes them to avoid page faults. So total number of MPCs must be fixed before the server is started, even though only 1 is currently used. Total number of MPCs are fixed at the source code level before compilation. So picking up the hardware should decide how many MPCs will be supported and if it runs, it will run as and when newer MPCs are added later on, because memory are not increased on the fly.

This code architecture ensures that at a much later time, server should not crash from memory mismanagement. STL based variable size structures are not used as a safe design option. No garbage collection is required. Reset of C++ smart pointers work wonders on fixed sized heap arrays.

Total number of MVs and CVs are both 8 in the demo. An 8⊗8 size should be the size of 95It is difficult to tune bigger size MPCs, and they also tend to be sparse. Breaking one big 16⊗16 MPC to smaller 8⊗8 cooperating MPCs are always better.

> **Not Yet 3.4.1** Demo does not have any example of cooperating MPCs, where CV/MV limits or setpoints of one MPC is set by other MPCs. User design is required for code support.

Remark The magic number 8⊗8, 16⊗16 are based on the computer hardware designs, where memory chunks are aligned 2^n multiples of 8. Just forget about database and other Computer related things. Focus on the plant. We are Chemical Engineers.

3.5 MPC Configuration File

Inside `C:\2mpcData` there is a `0.csv`, working as the configuration file for first plant. When demo server is started it looks for this first MPC configuration file, else server does not start. After all, without any MPC configured, why should the server run anyway. Subsequent plants can be loaded at "Load Simul 1" button and administrator privilege is not required, a `1.csv` is required, can be a copy of `0.csv` (change come of the name strings with a maximum of 15 characters).

This csv files can be edited by MS Excel and saved as a csv text file. Subsequent plants are enumerated as 1, 2,..etc and that is rigid. Plant-name appears as plant0 can be chaged to FCCUrxrg, if required, with maximum of 15 character long.

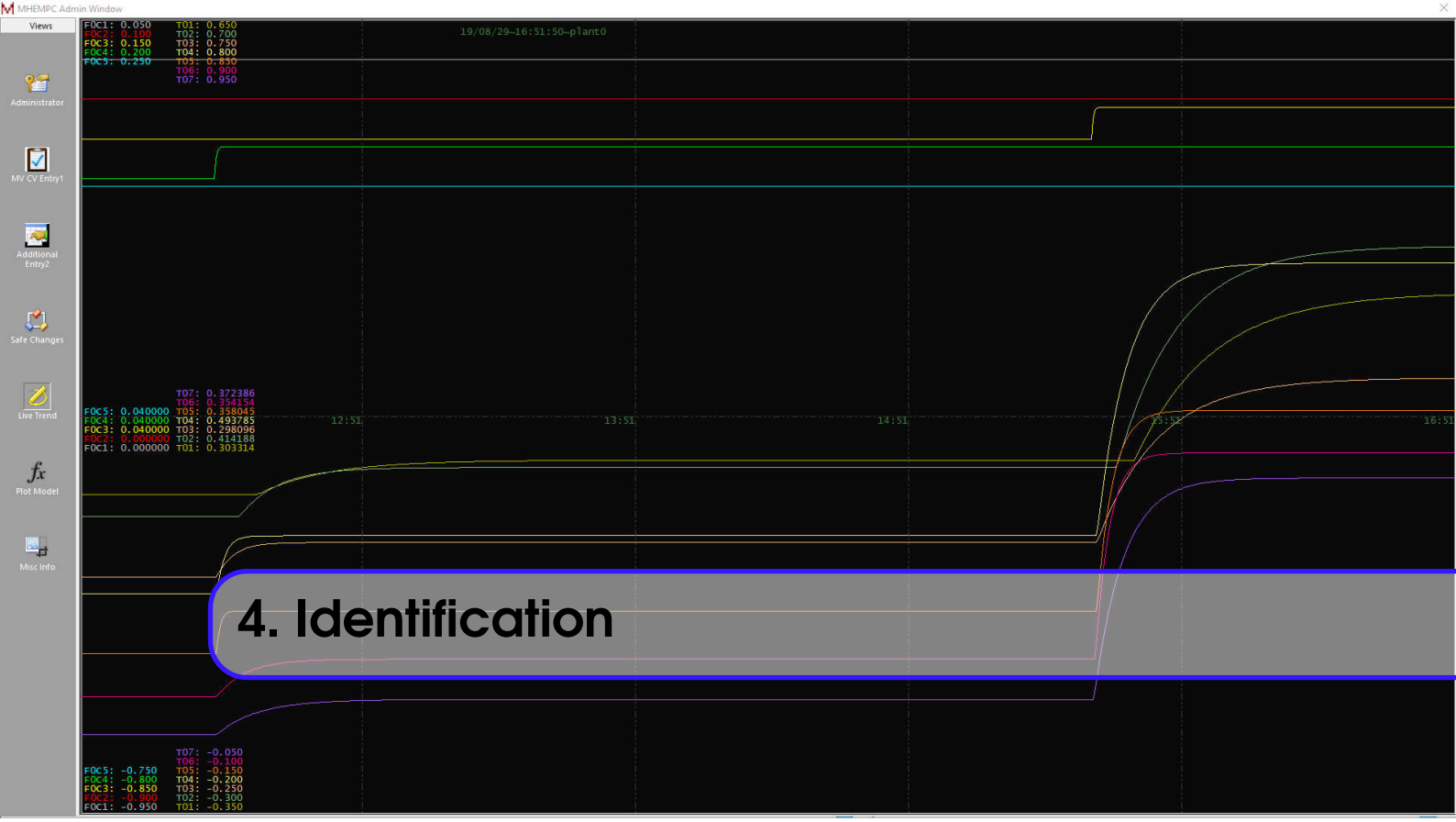
`c:\3Offline` directory should be in client computers on LAN, but exists on the demo computer for the purpose of this demo.

3.6 Automatic Directory Creation

Many directories are automatically created, which will be described later when used.

II

Part Two, Process Identification



If `mhempc` shows a "`Live Trend`" like above picture, it is time to move on to the next big thing, Model Identification. Run `Offline`, it will create relevant directories. Close it, run it again.

4.1 Contiguous 10 Days

It is not necessary to have 10 days of data for the demo. In a real plant step tests may begin after planning period and therefore may already have historian running for longer. But it does not make any difference when it comes to process identification. It also does not matter how many of the inputs are moved, at least there should be one. Also more than 720 data points be selected.

Also important that choosing the begin of data segment should be before choosing the end segment.

4.2 Browse the Trend and Select

`mhempc` shows live trend of 5 hours of 10 seconds data, useful for online tuning of MPC. For model identification browsing of 10 days of history seems enough.

Try 4.2.1 Open `Offline` and go to "`Plot, Choose Id Segment`".

- On the left end there is a vertical dashed line, same on the right end.
- Click on space, left to the dashed line. It takes us to previous window of trend.
- Click on the right of the right dashed line. It takes us to next trend.
- Try left-right a few times.

Pay attention to mouse movements, they are intuitive and very easy to use.

Try 4.2.2 Let us select a data segment.

- Go to a trend sufficiently in the past, an hour before the first step change.
- Click on the screen with left mouse button to mark beginning time.
- Go to the next trends until after the end of the last step test.

- Click on the screen with right mouse button to mark end time.
- If the screen does not blink then segment was forbidden as too small.
- Now go to "`Workbench Current`" tab.

This is how a data segment is selected for doing identification.

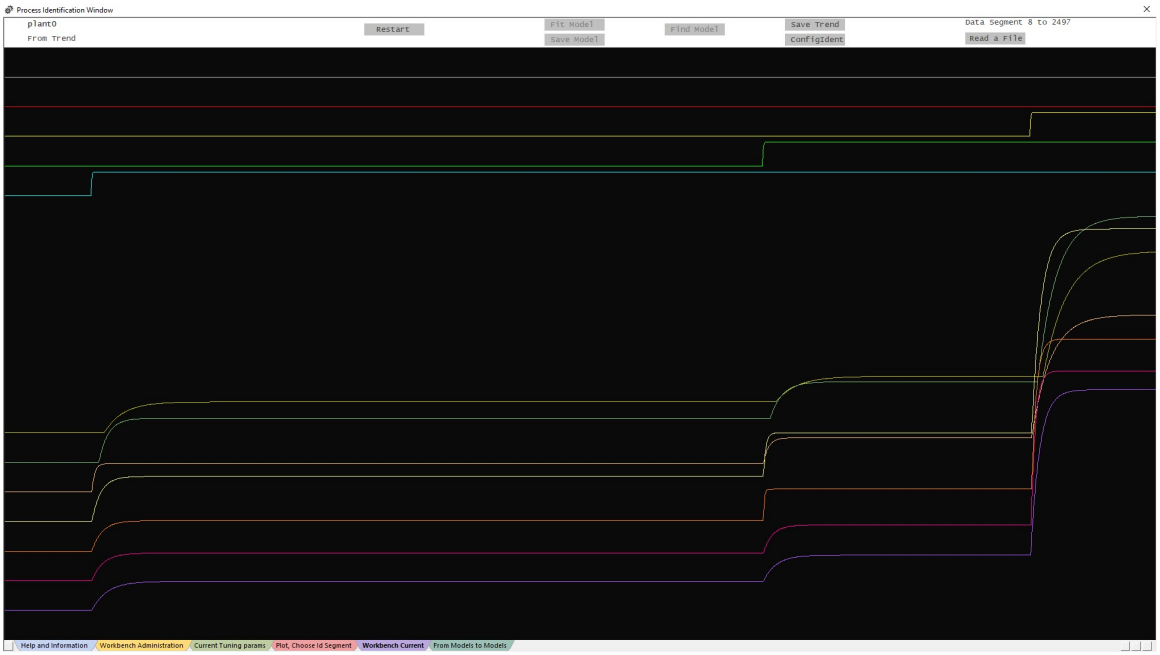


Figure 4.1: Current Workbench Data Plot.

"`Workbench Current`" tab looks like figure 4.1. On the top right, see the data segment start and end data numbers. Negative numbers just signify values from yesterday data.

4.3 Using the Identification

We may either go ahead for an identification on this data segment or export it to an MS Excel spreadsheet and reload the segment. At any time only one segment of data is taken for identification.

Remark Some experts may like to artificially change some of the data points on the step test and create a model, hence Excel export. But this is also they way to take a window of operating data and do any other calculations. Remember, if MHEMPC is running, any statistical covariance based inferences may have to consider the presence of MHEMPC.

4.4 The Simulated Plant

The plant is adopted from the book, "Fundamental Process Control, by David M Prett and Carlos E Gracia, page 21, Table 3.1", to describe DMC algorithm. It is reproduced with tag names changed in Table 4.1.

Try 4.4.1 Click "`Save Trend`".

The file save directory is default `C:\3OffLine\Modelling\A\mdrData`, but can be changed. This file can be altered (or without any alteration) and loaded to workspace.

Try 4.4.2 Click "Read a File".

The file reading default directory is C:\3OffLine\Modelling\A\mdrData. If you load this file, the workspace will be loaded with data from file, replacing the previous workspace.

This software uses a lot of memory and C/C++ was badly reputed for causing memory faults in the 90s and were used to dissuade students from a serious C/C++ programming career and choose Java or C# instead.

The real reason was C/C++ executable cannot be reverse engineered, i.e. obtain source code from executable. But modern C++ has smart memory management to call memory deletion when object goes out of scope without any explicit call to delete, thus a in-situ garbage collector.

Us	F01	F02	F03	F04	F05
T01	$\frac{4.05e^{-27s}}{50s+1}$	$\frac{1.77e^{-28s}}{60s+1}$	$\frac{5.88e^{-27s}}{50s+1}$	$\frac{1.20e^{-27s}}{45s+1}$	$\frac{1.44e^{-27s}}{40s+1}$
T02	$\frac{5.39e^{-18s}}{50s+1}$	$\frac{5.72e^{-14s}}{60s+1}$	$\frac{6.90e^{-15s}}{40s+1}$	$\frac{1.52e^{-15s}}{25s+1}$	$\frac{1.83e^{-15s}}{20s+1}$
T03	$\frac{3.66e^{-2s}}{9s+1}$	$\frac{1.65e^{-20s}}{30s+1}$	$\frac{5.53e^{-2s}}{40s+1}$	$\frac{1.16}{11s+1}$	$\frac{1.27}{6s+1}$
T04	$\frac{5.92e^{-11s}}{12s+1}$	$\frac{2.54e^{-12s}}{27s+1}$	$\frac{8.10e^{-2s}}{20s+1}$	$\frac{1.73}{5s+1}$	$\frac{1.79}{19s+1}$
T05	$\frac{4.13e^{-5s}}{8s+1}$	$\frac{2.38e^{-7s}}{19s+1}$	$\frac{6.23e^{-2s}}{10s+1}$	$\frac{1.31}{2s+1}$	$\frac{1.26}{22s+1}$
T06	$\frac{4.06e^{-8s}}{13s+1}$	$\frac{4.18e^{-4s}}{33s+1}$	$\frac{6.53e^{-1s}}{9s+1}$	$\frac{1.19}{19s+1}$	$\frac{1.17}{24s+1}$
T07	$\frac{4.38e^{-20s}}{33s+1}$	$\frac{4.42e^{-22s}}{44s+1}$	$\frac{7.20}{19s+1}$	$\frac{1.14}{27s+1}$	$\frac{1.26}{32s+1}$

Table 4.1: Continuous Time Heavy Oil Fractionator Nominal model. Gains have dimension(Temp/Flow) and time units are 10 seconds.

4.5 Identification

On the workspace the entire data segment on which identification will be done, is plotted. If not liked, choose another range from "Plot, Choose Segment" or by reading another file. If fresh plant data is desired, close Offline and open again.

Try 4.5.1 Click ConfigIdent.

This detects which MVs are moved and chooses all CVs.

Try 4.5.2 Click Find Model.

For small data segments, less MVs, model finding is quick. The models of 2 MVs and all CVs will be found and displayed as on figure 4.2. It contains the convolution model structure normalized for space, gain and dead-time. If the model is OK, we save it as anyname.mdr in default directory. A process data fit can be done.

Anytime, choosing the data segment process can be restarted by clicking `Restart`.

4.6 Data Fit

Here we take those MVs participating in the current identification workspace, use the models found to predict all the process CVs back from the MVs. They may or may not match, depending on the noise level, plant drift, plant-model mismatch and/or bad quality model. On a demo setting everything looks very good and the same windows changes to as in figure 4.3. Predicted values are plotted white, here sits on top of raw plant values. On a real plant plant values may not get eclipsed.

4.7 Congratulations

We just did some modeling, but not all the models are done yet. This intermediate and incomplete models or submodels, can be saved in a file. But we must have a discipline about them.

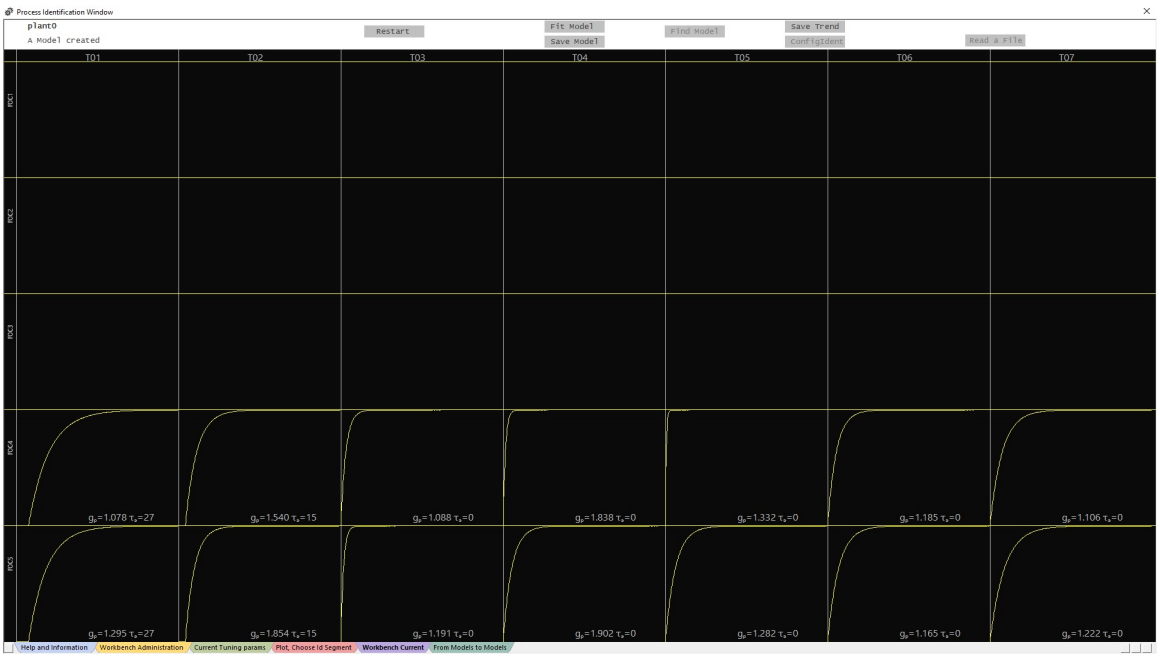


Figure 4.2: Plotting unsaved model.

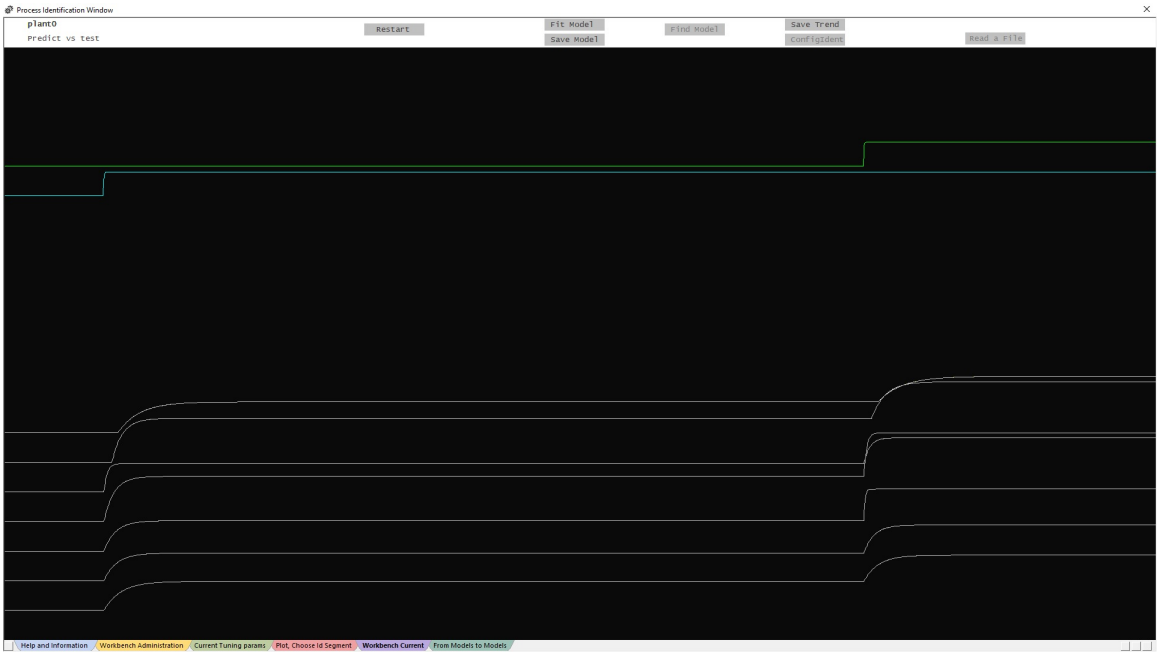
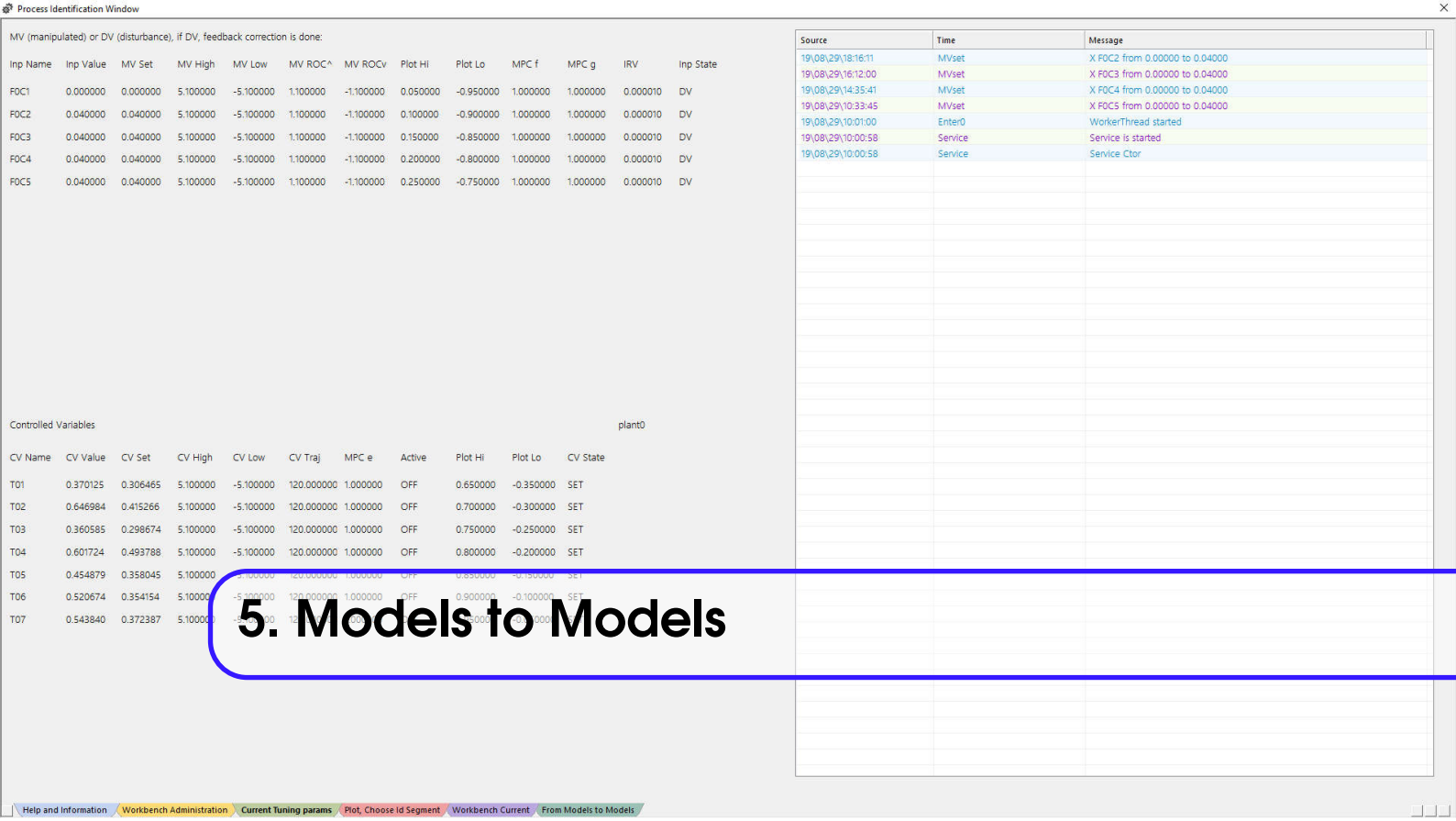


Figure 4.3: Fit the unsaved model.



Good models give good MPC performance.

But in the middle of so many inputs, so many outputs, so many step tests and so many models, confusion easily sets in. We select a data segment, find a submodel then forget which data gave which submodel. So we save both the data and its submodel on the same name, but create our own directory and file name handling system and it gets messy very quickly. Have a look at fig 5.1 for my own example. Here we cannot figure out which mdr file belongs to which data set.

5.1 Managing Directories

Let us create and maintain a protocol among our teams. It is a good practice to adhere to this. It is also important for record keeping and may help meet organization requirements. There is already a bit of discipline there.

`Offline` creates this directory structure shown in fig 5.1 when user runs it. With 2 plants configured as the total number of plants on server, they are enumerated as 'A' and 'B'. So `SnapShot` directory keeps current copy of global, trend and messages in `saveA.txt`, `trendA.txt` for 1st plant. All messages from all plants are in one file `allMsgs.txt`. They are current copy of server `L:\`.

Sub-directories A and B are for historical data. If user clicks for previous data and it is available at the server, they will be copied on demand. Note that the file sizes are small and copy is fast, so user does not feel any delay for 10 days history access. We do not open SnapShot directory for any purpose.

On "`Current Tuning Params`" tab of `Offline`, current copy of messages and current tuning parameters are displayed. They may not be useful during the step test, but come handy when on-line live MPC needs to be tuned. We look at the current trend, tuning and messages and can take decisions MPC tuning using offline client computers. Sometime these are very contentious among team members, so decisions are taken offline first, implemented later after a consensus. Tuning is not described in this document.

For now, we are using `Offline` for process model identification.

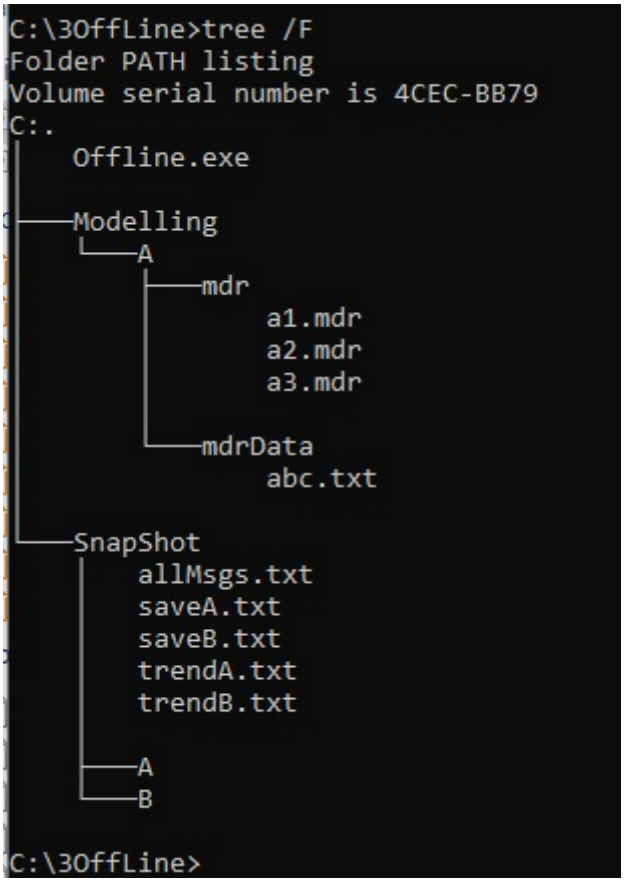


Figure 5.1: Directory structure for modeling with disorganized files

Remark MPC tuning parameters have to be entered only using `mhempc`, which is available at the server. Only after a final process model is loaded on the server and MPC is started, that tuning is meaningful.

5.2 Systematic Model building

After selecting a segment of identification data, save in a datafile, at least for record. This automatically opens `A\mdrData` directory when "`Save Trend`" is clicked. Later "`Read a File`" can be used to load this data on the workspace and displayed. When a submodel is found we use "`Save Model`", which opens `A\mdr` for saving as and `.mdr` file. Although a different directory is allowed by Windows, it is safe to use these default directories.

Remark We keep the name of submodel file and data segment file same, like `543step1`, we made step test on MV5, MV4, MV3 and this is step test enumerated as 1 as in fig 5.2.

5.3 Submodels

Submodels are just intermediate models obtained after an identification process. They serve many purposes.

- We pick final model from submodels.
- Submodels can provide "backoff models" for identification under a disturbance input. (not included in demo).

Submodels are stored in files with `.mdr` extensions, while the final model used by MPC has `.mdc` extension. A submodel found in one plant is not readable by another plant workspace (Each model is protected by a unique prime number.). Plantname is displayed at the top of the page and switching between plants is done at "`Workbench Administration`" using radio buttons. Currently only one plant on demo, not much switching.

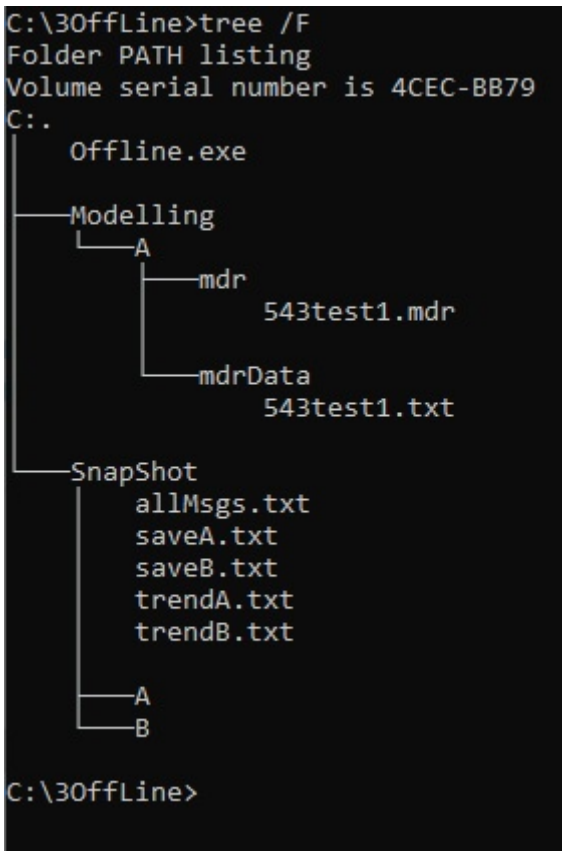


Figure 5.2: Directory structure for modeling organized files

There may be several step test sessions and each give rise to some MVs vs all CVs model being found. Some models may be duplication, spread over multiple files. That is where tab "`From Models to Models`" comes handy.

5.4 From Models to Models

This tab "`From Models to Models`" has few tricky mouse operations.

Try 5.4.1 Choosing from submodels:

- Right click on the middle of the screen. Directory `mdr` opens for selecting any existing `.mdr` file.
- Load one by clicking.

Total of 8 such files can be loaded simultaneously here. Select one or more. Since demo submodels are always the same, models will be be indistinguishable from one another and will overlap. But in a real plant different step tests will yield similar but not the same submodels.

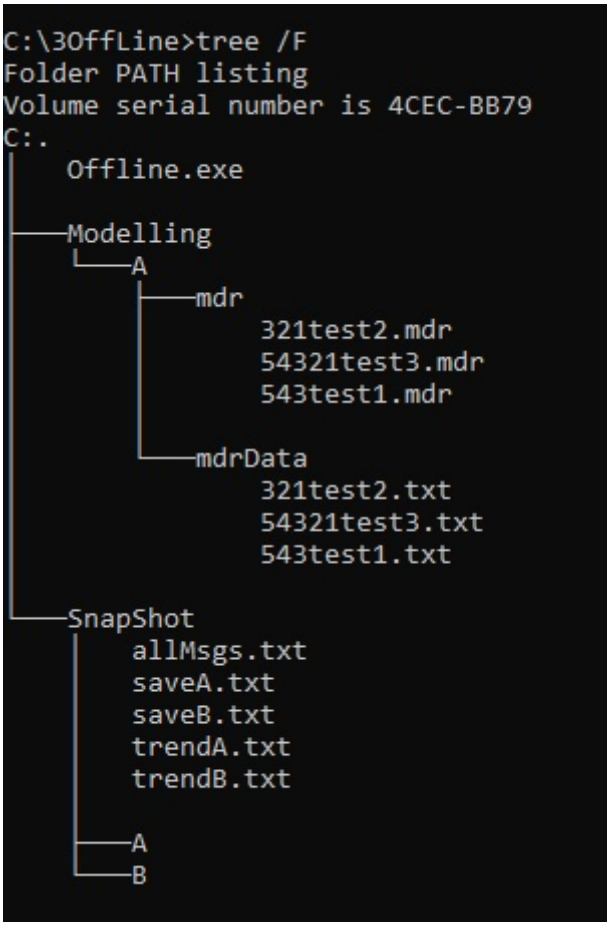


Figure 5.3: Directory structure after 3 modelings

5.5 A Complete Example

- Do step test in the plant in sequence of F05, F04, F03, F02 and F01.
- Select first 3 tests.
- Save the trend as `543test1.txt`.
- Do "Find Model" of first zone F05, F04, F03.
- Save the submodel as `543test1.mdr`.
- Select last 3 tests
- Save the trend as `321test2.txt`.
- Do "Find Model" of last zone F03, F02, F01.
- Save the submodel as `321test2.mdr`.
- Select all 5 tests.
- Save the trend as `54321test3.txt`.
- Do "Find Model".
- Save the submodel as `54321test3.mdr`.

Now the Modelling directory looks like fig. 5.3. In From Models to Models right click and load all .mdr files in that order. The screen looks as fig. 5.4. Get a feel of time taken for identification.

5.6 Choose or Discard

Each model curve from all 3 files are displayed if they exist, with a solid box before G_p and t^d. Let us discard some of them by click-toggle on the solid box to hollow box. To get rid of any `.mdr` file, right-click on the box representing that file on the narrow band at the top containing `plant0`.

Try 5.6.1 After finalizing which submodels will participate in final control model `0.mdc`, left-click on the narrow band at the top containing `plant0`. `L:\A\mdc\` is the directory to save as `0.mdc`.

More Info 5.6.1 If saved in any other name, MPC server will not pick it. For fist MPC, `L:\A\mdc\0.mdc`, second MPC, `L:\B\mdc\1.mdc`, so on. This enumeration allow the server load Control Model at minimal interaction.

5.7 Missing Models, Bad Models, Disturbance Variable Models

This is a big topic of discussion among APC implementors about model quality, here is a guideline:

- If a chemical steady state calculation points at the existence of a steady-state relationship, there is a dynamic model.Check instruments.
- Dynamics are marred by noisy measurement, send the instrument for maintenance before step test.
- During the run of MPC, instruments can go for maintenance and zero-checked, same models will work.
- If a real relationship exists and control model blanks it out, MPC tuning is very difficult.
- Do not enter into pedagogy of "feedback vs feedforward", it does not work well.
- Control models should be smooth and rising, not twisted snake-like figures.
- Last 20% of the model must at least capture final steady-state.
- If one model has end zone still rising, MPC tuning is difficult.
- If an input has good measurement, but bad/unavailable PID, configure as DV.

Not Yet 5.7.1
- Filtering step test trends before modeling is possible, not in demo.
- Backing off a disturbance variable model from identification is possible, not in demo.

Next chapter, we assume all models are available for this demo. Configure and run the MPC on `plant0`.

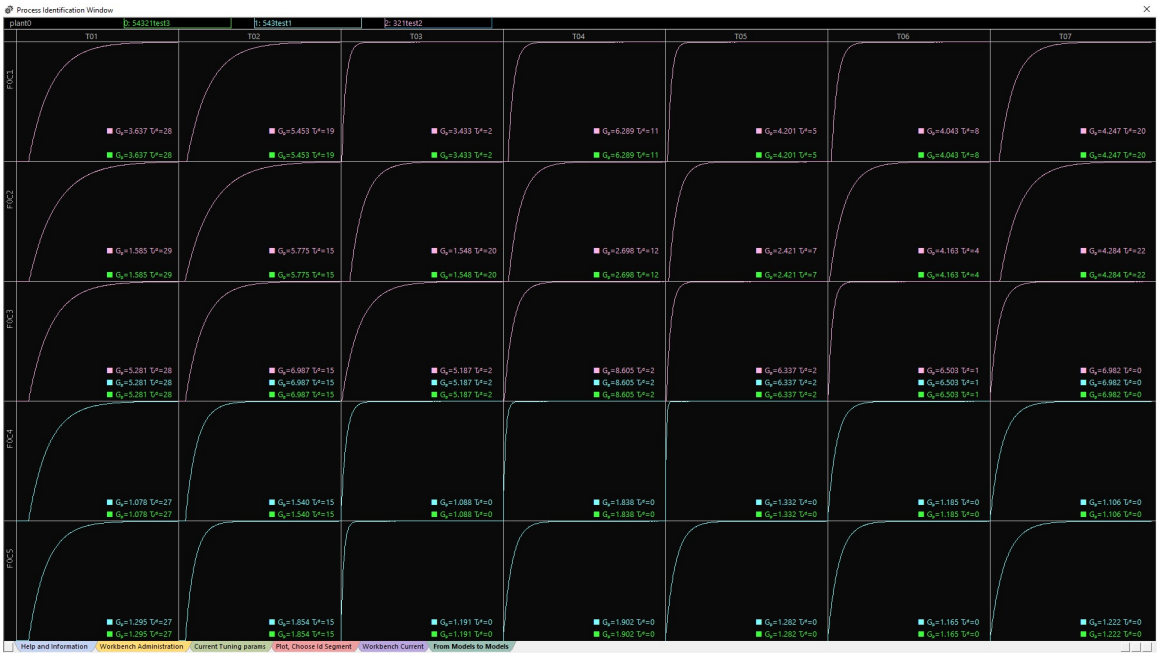


Figure 5.4: Submodels loaded from 3 modelings

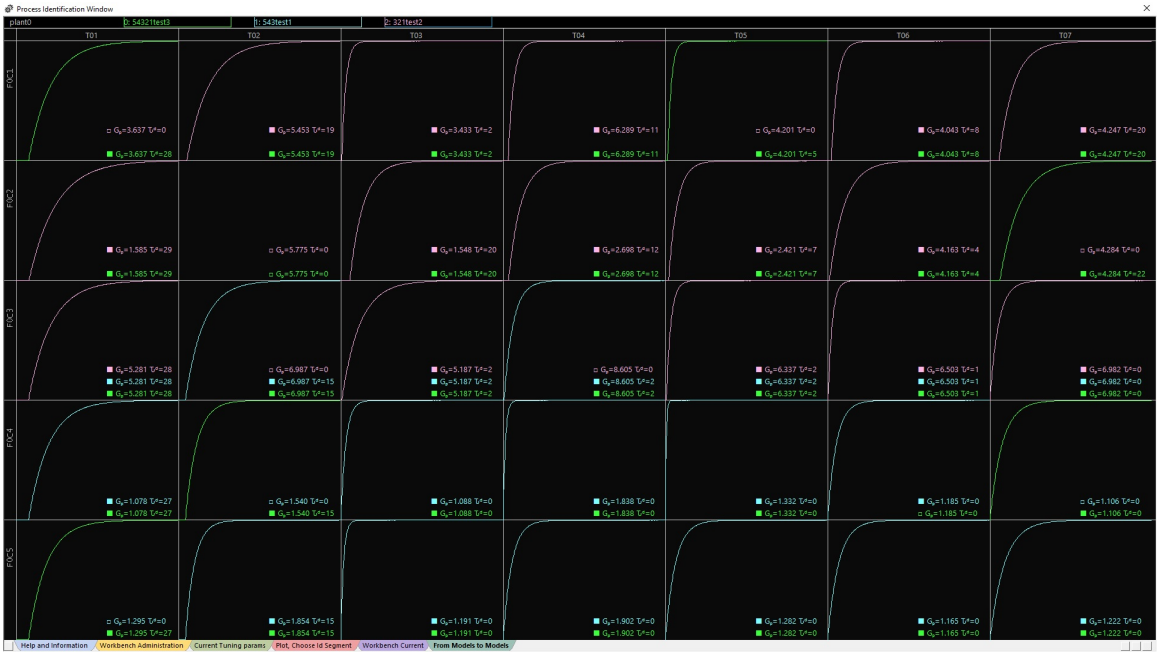


Figure 5.5: Some submodels unselected, all for FC2-T02

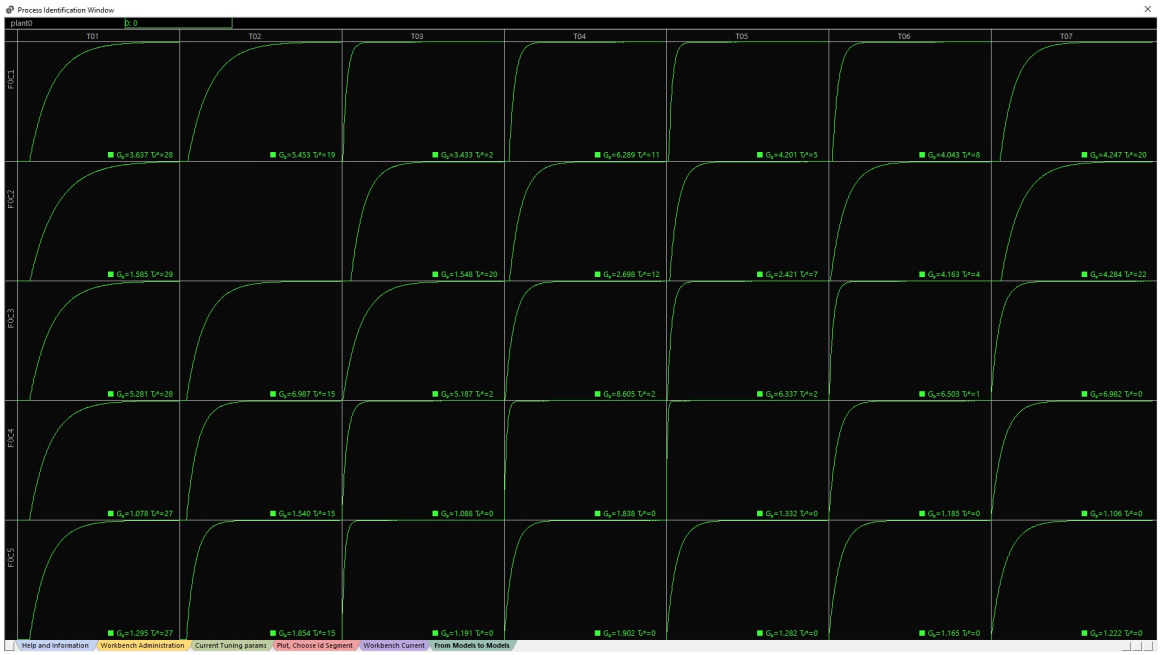
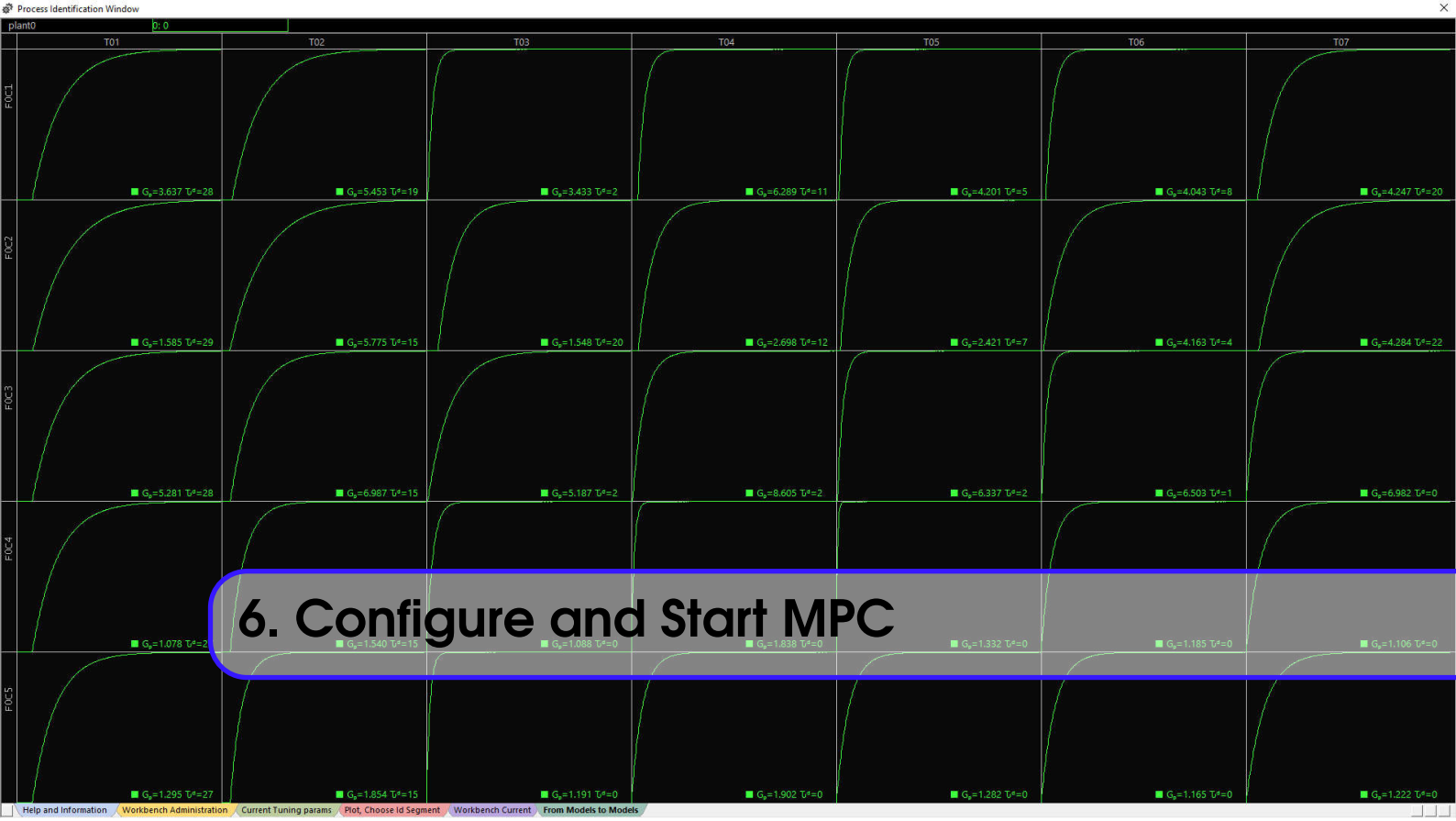


Figure 5.6: Final control model, FC2-T02 missing

Part Three, Plant0 MPC



6. Configure and Start MPC

For simplicity we will enter all the relationships as shown above and configure our MPC. Already the MPC server is running the `plant0`.

6.1 Auto-Detect a new Control Model

Open `mhempc`. Feel the delay in loading because of synchronization between MPC server and `mhempc`. Every cycle of both is 10 sec, to make sure plant calculated/simulated values are available for control, `mhempc` has a 1 second delay and synchronized at each start. But real control does not happen at `mhempc`, it is to send user entry data to the MPC server.

If a new `0.mdc` is detected at `L:\A\mdc\`, a button "`Load Model 0`" appears below "`Simul 0`" as in fig 6.1. Click the button, close `mhempc`, open again. The button is gone as control model file is now loaded to the server for `plant0`.

Remark Sometime, even when a new `0.mdc` file is kept and `mhempc` is started, "`Load Model 0`" does not show up. It takes one cycle-time for MPC server to detect a new file. So if you are expecting that button, just restart `mhempc`.

To make sure that `0.mdc` is correctly loaded:

1. Go to "`Misc Info`" of `mhempc`. Check the process gain table at right top.
2. Go to "`Plot Model`" and see the model look as expected. Two curves are side by side. Right one is from `Offline`, truncated to total length, left one is the model modified for controller frequency.

6.2 Start a minimal MPC

We now go to "`MV CV Entry1`".

Try 6.2.1 Starting a minimal MPC,

- Under "`CV State#`" column, Click on ✓`OFF` for `T07`.

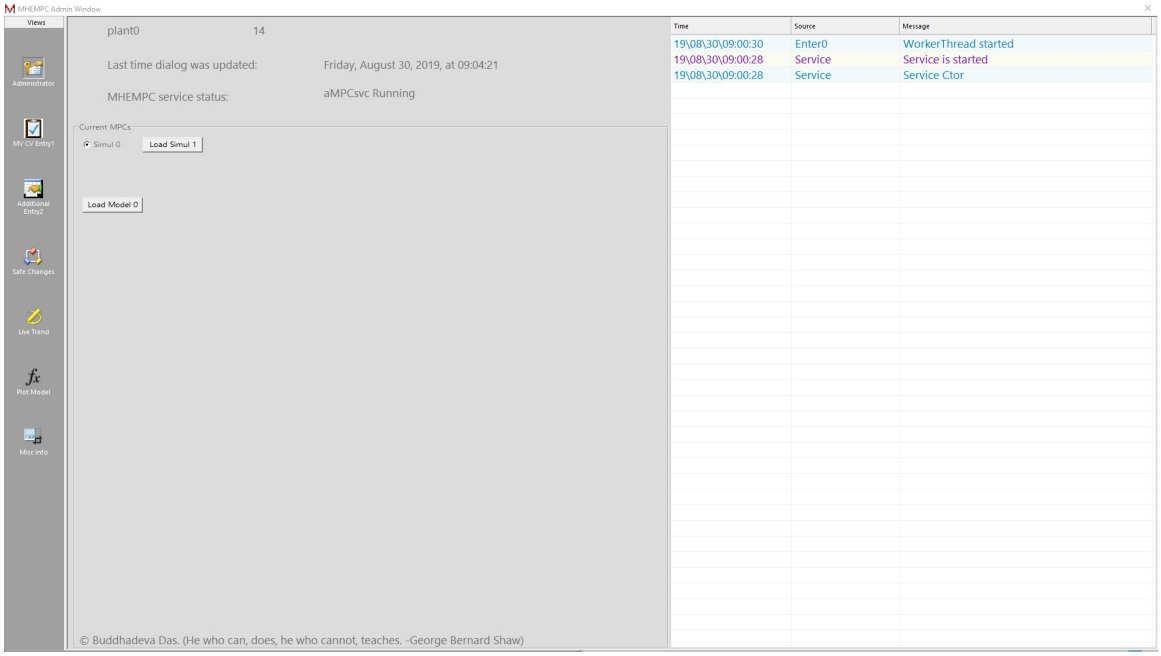


Figure 6.1: New model, click to load.

- Select `SET` at the dialog box.
- Under "`Inp State#`" column, Click on ✓`OFF` for `FC5`.
- Select `MV` at the dialog box.

And at the top row, "`Control OFF`" changes to "`Control ON`" automatically. See fig 6.2, there is a ← beside the setpoint of T07 after the first execution of control. Change the setpoint by 0.01.

If at least one MV is available for one CV, MPC is switched ON. In due course of time other MVs and CVs can be active. However if the whole MPC is to go OFF at once, left-click on "`Control ON`. The entire MPC is switched OFF at once.

Try 6.2.2 Switch the MPC OFF.

6.3 Issues

Uptil now we have not thought about performance and safety issues, after all, we are still playing with a demo version. But very soon play-time will be up.

6.3.1 Demo vs Actual MPC

In the actual plant user interfaces for using MHEMPC will be different. The operator interacts with the DCS window created for the MPC. The functionalities are similar, but entry page "`MV CV Entery1`" is now moved to the DCS. Few entries like "`MV ROC`" (Rate of Change), "`CV Traj`" (trajectory), "MPC el" (weight on CV) will stay at `mhempc` and rest are available for operator interaction.

6.3.2 Changing MVs

If "`Inp State` is OFF, operator can manually change MV, otherwise not. It is possible to demonstrate DVs in this demo this way. At a much later time we will learn IRVs.

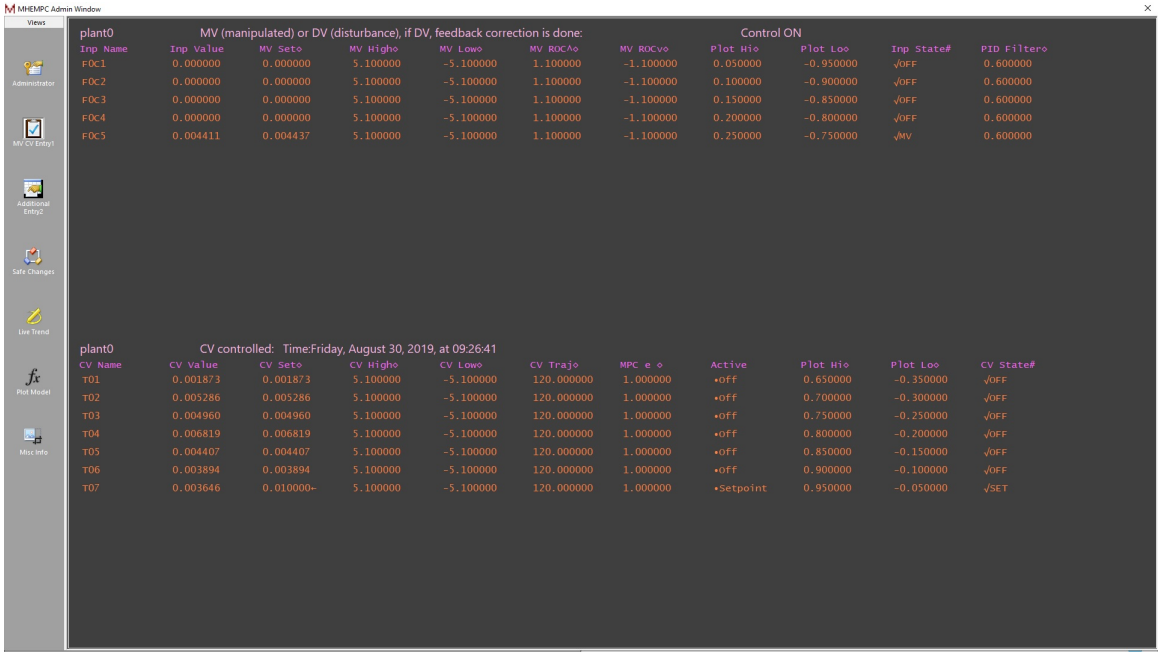


Figure 6.2: One CV and one MV.

6.3.3 Changing CVs

If "`CV State#` is ✓`SET`, then CV Setpoint can be changed, on every other occasion, not allowed. Just change it for T07 by 0.01 and see the F05 setpoint move. It will take 50 seconds for `Plant 0` to calculate an MV. Go to "`Misc Info`" for statistics.

CV Tracking

IF "`CV State`" is not Set, then "`CV Set`" tracks current value. In case the CV is changed to "`SET`", where CV Set is now meaningful, it should begin with current value at the time it changed to "`CV Set`". This is called bump-less transfer. Operator can now change it.

Limit Based CVs

MHEMPC lets us configure CVs as one of,

- SET, strict setpoint
- HIGH, only high limit
- LOW, only low limit
- BOTH, high and low limits

Mathematics governing degrees of freedom says that with 5 MVs available to be chosen at any combination, we get a deterministic result on only 5 of the CVs. Other 2 CVs go to their respective places and cannot be chosen. So MPC implementors tend to think they can choose 5 CVs with strict setpoints with other 2 at zone limits.

Wrong design. Here is why:

1. Let us fix 5 CVs, final steady state value on all MVs are fixed in a deterministic way.
2. We do not like these MV positions, they are not profitable, especially top flow and top temperature.
3. So we put limits on MVs, now we have to change 2 of the CVs to zone limits, only 3 left.
4. Just to keep 3 temperatures on setpoint, MVs move a lot, a perennial dynamic changes.
5. Again put 2 more CVs on Zone limit and we are left with only 1 CV on setpoint.

To Set or not to Set

The lone CV on Setpoint works as the heartbeat of the MPC. We should not choose 2 CVs on Setpoint, but a judgment can be made with a lot of observations (monogamy works better). In FCCU reactor-regenerator MPC, choose "Regenerator Delta Temperature" as the heartbeat variable. Let other CVs depend on the strict choice of this heartbeat. Their limits are relaxed to be close to their steady values, but not close enough to be picked by MPC very frequently and intermittently. This injects a ringing in MPC, a perennial oscillation.

Remark Chemical plant operators do not agree to much with Lyapunov stability. A little oscillatory control is still within a finite ball and good for a control mathematician. What the operator wants is least hassle. If they can think of their taxes or kids while sitting on the panel, they have confidence on MPC and that is the pure definition of stability in a real plant. (Do not tell this to the operators, they should not think of their taxes or kids while at work.)

The most difficult CV to control is already known to operators. It is the one with longer deadtime or steeper lag or attracts maximum and very frequent MV movements or all of them.

The naughty, the nice and the unnoticed

On the demo system there is an easy way, because all CVs are same type of things, just draw temperatures, all MVs are same type too, just draw rates. They can be pitched among themselves without any bias. We go to "Plot Model" page. We watch them, take a print screen, send to a printer, then we contemplate, which CV among you 7 is the naughtiest.

I choose T05. here is how:

1. Deadtime first; Base raw model has a length of 300, truncated from 360, mentioned at `0.csv`. The highest deadtime is 29, less than 10% of the model length. Control model has a dead-time 5, with frequeny of 50 seconds and total model coefficients are 60. Dead-time is not so important.
2. Lag next; FC4 vs T05 is the quickest, so are FC4 vs T04 and FC05 vs T03.
3. Watch their gains, FC4 vs T04 drops out, between FC5 vs T03 and FC4 vs T05, later is steeper.
4. Switch on this pair FC4 -T05 first and alone. They are the naughtiest pair.

Do not let the Grinch steal your Christmas

The Grinch is the naughtiest on the otherside. Take your pick. My pick is FC2 vs T01. Here is why.

1. Worst dead-time
2. low gain, shall cause excessive MV movement.
3. At the end of the model process response is still rising.

Any friend of the Grinch is a Grinch too

This is the naughtiest runners-up. I pick FC1 vs T02. Here is why.

1. Bad deadtime.
2. At the end of the model process response is still rising.

And there are angels everywhere

As you switch on FC4 on MV and T05 on Setpoint, other CVs that are still OFF, start moving too. While tuning watch the Grinch first, angels later.

Why 2 model lines

At Plot Model we see 2 models closely following each other. The one to the left is what controller uses and the other is from process identification.

Original identification is always done for 360 time intervals representing convolution structure. When we click to load `0.mdc`, first we check what length from this is considered. At 0.csv we say it is 300, so last 60 are just discarded. Then we check for control cycle as 5. So the first 5 coefficients are added to create the first element. Second 5 elements are added and when plotted, it goes to the spot for the 6th value representing 6th to 10th. So the curve is finally plotted shifting to the left.

Total number of points on the convolution structure used by final control is 60. We can consider the entire 360 points found by identification and and run the control at 60 seconds a cycle. This will also give control total number of points as 60. MHEMPC is strictly based on 60 control points. The second option of running at 6 intervals can be configured as the 2nd plant and compare how the two variations compare. Demo plant parameters will remain the same. Another option is to run the plant every 20 seconds and get a better steady-state part in the models. All these experiments can be done later. Let us finish what we are still working on.

6.4 Switch MPC ON again

Try 6.4.1 Click "`CV State`" of the CV T05 and change it to "`SET`". For FC4, click "`Inp State`" and change it to MV.

Watch,

1. the top of the page show "`Control ON`".
2. at the first execution of MPC a ← appears beside the number showing the setpoint for T05.
3. On "`Active`" column, Off changes to Setpoint.

Try 6.4.2 Change the setpoint a little.

Watch the MPC calculate and change a new value for FC4. And CV T05 will change towards the setpoint. All other CVs change too. Watch the trend.

6.5 Watch the Trend

When MV changes by small amounts and CV responds slowly, the trend resolution may be changed.On the top left corner of Live Trend upper limits of MVs and CVs are shown. The can be changed by clicking on them. Same goes for the lower limit shown at the bottom.

6.6 Misc Info

Congratulations, the controller is runnning. Most often both MHE and MPC give a lot of informations about their health. Some are listed at Misc Info. Time taken for MHE and MPC in milliseconds are important. Together they should not exceed too much beyond 10000 milliseconds. Our controller cycle is 50000 milliseconds. Beginning residuals r1 and end residuals r2 are worth watching.

MPC runs in 2 stages. First stage calculates a nominal MPC using current CVs selection. This MV set can bring a new CV under constraint violation. So the second stage runs with the new variable in picture. Even if a CV is selected as a limit control, it may be dropped and chosen by MPC.

6.7 Switch ON others

Let other CVs be on `BOTH` and othe MVs on MV all together. Let the zones for the CVs wide so that they are not selected. Watch all MVs moving for T05 only, now much different.

Change T01 High limit down

Try 6.7.1 Try changing "CV High" close to current CV value of T01. A message comes up that the value is not changed. Go to page "Safe Changes".

Here see that under "δCV High" we have 0.5. This forces CV High to be changed not more than that amount at one go. This safety mechanism can be useful later. Now Change it to 5.0.

Now Change T01 "CV High" again. When T05 will come to setpoint, T01 will also move close to violating upper limit and create a conflict for the MVs to react in a multivariable way. We see the MHEMPC in real action. We get a real feel on the MPC. When a constraint is selected for control, a appears on that limit. Look at T01 closely in fig. 6.2, the high limit is already selected although not violated yet. Prediction mechanism tells to bring it down, as a CV violation will occur in near future.

Change T02 Low limit up

Create some more conflicting situations. Spend some time.

6.8 Test a DV

The naughtiest of the MV was seen as F04. Switch it OFF. Change its current value a little.

MPC does not know that this value is changing. As it changes all the CVs, MPC reacts to correct the CVs. this is feedback.

Repeat the same by putting F04 as a DV. Now any change in F04 is intercepted by MPC as potentially distubing the future. So a prior correction will be taken. In the simulation enter a new value for DV, but in real implementation, a DV changes on its own.

Remark If you feel tired, it is OK. Take a break, but we have already seen a lot of MHEMPC.

6.9 Practice makes us Perfect

We are a white belt in MHEMPC now, roughly 60% of MHEMPC ideas are covered. Long before we become a black belt, we have to go through a lot of other colors, a lot of try, a lot of practice. But the demo software has all other functionalities that can be tested. This will build a level of confidence to place an order on MHEMPC.

And if you spend some time on MHEMPC, you may observe that it is selecting limits for control just before it will be violated.

Try 6.9.1 At this point we may be able to do any of the following, and perhalps some more:

1. Put all the CVs ON, one on setpoint, others on limits.
2. Put all MVs on, one on IRV
3. Create almost conflicting environments.

and see how MHEMPC performs.