# Part A

Firstly, we analyse and define the control problem to be solved. The task is to implement an MP controller for a gantry crane, capable of tracking a given square in XY (horizontal) plane. The square is to be tracked by the mass suspended from the crane and not necessarily the crane itself.

This problem can be deconstructed into 4 separate problems of moving the mass from a given point A to a given point B. The four movements are then executed in sequence, such that the mass tracks the required square in the end. This definition of the problem allows for the use of controller implemented for previous assignments with only minor changes.

Having decided to use code from previous assignments, a choice has to be made between using controller with only hard constraints and controller with soft constraints implemented. To that end, both controllers were tested in moving the weight from a given point to another given point. Both controllers were set up with the same cost matrices and constraints, apart from the angle constraint, where the soft constraints were used. This testing showed superior performance of the controller with hard constraints only. The controller with soft constraints took significantly longer to compute, since the slack variables increase the size of the system to be optimised cit. At the beginning of the simulation, the controller would not have computed the solution in time (solutions took more than 1 second with sampling time of 1/30 seconds). Moreover, the settling times of controlled variables of hard constrained controller were shorter.

We can now begin to adjust the hard-constrained controller to the square tracking task. The first step is to define the square to be tracked. Since the problem is posed as moving the crane from a given point A to a given point B, the square is defined by 4 points in its corners. These points should be far apart, to avoid problems connected with the stickiness of the crane (larger force required to begin moving it). If the controller exerts large force to overcome the stickiness, it would risk overshooting the corner of a small square. Equally, if solution is set up to heavily penalise overshooting, the crane might not reach its end state, or even start moving. The square is oriented so that its sides are parallel with the X and Y axes. This simplifies the definition of the constraints (as explained later). Moreover, this way the movement along each side is controlled by one actuator only, there is no coupling of the two actuators’ action in the movement, and consequently any oscillations are controlled more easily.

In the laboratory sessions, it was also noted that the crane experiences larger stickiness in the X-axis than in the Y-axis. This is due to the fact that in the X-axis larger mass (crane car and rail it moves on) has to be moved than in the Y-axis (crane car only). To ensure a square is tracked instead or a rectangle, a stickiness correction constant is introduced to the code. This constant is added to the higher X coordinate of the corners and subtracted from the lower X coordinate. By doing this, the controller is set to track a rectangle, but due to stickiness it will track a square. Value of this constant is best set experimentally in real hardware. In the simulations, the increased stickiness is not well modelled, and hence the value of the constant is set to zero.

After the target states (i.e. the corners of the square) are defined, the constraints are defined too. The constraints are imposed on the same states as in the previous assignments – on X and Y position and on X and Y angle. The constraints set on angle are constant. The position constraints are different for the four different target states. For each state, the position constraint is a thin rectangle with shorter sides lying on the previous and current target state and longer sides parallel to the side of the square. The constraint is defined by variable margin, which describes the distance of the longer side from side of the square.

Using the notation from lectures and previous assignments, the only matrices that are affected by the existence of 4 different constraints are the b constraint matrices. Therefore, there must be 4 sets of calculations performed in the initialisation stage of the program. The constraint matrices bb and target states are then cycled through based on the time of simulation/run of the hardware. Time of the simulation is taken, divided by distance between two adjacent states (i.e. length of the side of the square) to ensure longer time is allowed for larger squares tracked. This result is then multiplied by a constant, by which the action of the controller can be sped up or slowed down. Remainder after division by 4 is then calculated from this time control signal to pick the appropriate state and constraint matrix from the set of 4.

To evaluate the controller, a metric of “squarness” was developed. The trace of the pendulum is estimated from the simulation data (position and angle) by assuming length of the string. Even though this is not accurate calculation, it is still informative, as it involves the angular deviation of the string (i.e. oscillations). Smallest possible rectangle is then circumscribed and largest possible rectangle inscribed to the trace. The difference in the areas of these two rectangles gives then the rectangular area the pendulum sweeps. The smaller this area, the less oscillation the pendulum experiences and the more perfect is the square tracked. Note that rectangles were used in this method, since any deviation from perfect square shape can be readily solved by adjusting the stickiness correction variable.

Using this metric, the parameters of the controller were tuned using trial and error. The final set of the parameters is given in the table below.

*Use of constraints – define constraints around desired path towards the current target point. Can define 4 sets of constraints corresponding to 4 target states. Active constraint is always the path from previous target to current target. Matrices to be changed are bb (matrix of state constraints). 4 are calculated and then cycled through by target calculator.*

*Evaluate square and tune – definition of squarness and calculation. Inscribe and circumscribe a rectangle to the path tracked. Not drawing squares, since squarness problems (one side being longer than the other) can be solved by increasing the stickiness constant*

*Table of final parameters after the tuning*

*Constraint margin had to be loosened for the linear simulation to work. In lin model, if no solution is found inf/NaN is produced for input. Lin sim cannot be run with these. In non-lin sim, these values are limited*

*Discussion of tuning results as per notes*

*Linear simulation worse than non-linear – mismatch between the linear and non-linaer model. If optimised for the non-lin model, lin model will suffer in performance*

**Part B**

Discuss the practical challenges that arise with MPC in a continuous process industry plant, both (i) during the implementation phase and (ii) during the entire life cycle.

(i)

Parametrisation – running long tests

Converting the plant into the model – maths (complex dynamics of the plant) continuous processes are inherently complicated (sometimes not even properly understood [1]) as they involve many loops, consist of interconnected units – BUT techniques exist where cost is independent of the state size [2]

Model is non-linear – has to linearize it – non-linear parts computationally expensive

Lack of experts and time to do this

Significant initial investment – although costs retrieved soon

(ii)

Change in plant requires change of model – difficult

In the following section, practical challenges connected with use of MPC in continuous process industry are discussed. It is split into 2 subsections: first discussing issues during implementation phase and second challenges in maintenance of MPC controller.

The first issue can arise before the MPC design phase is commenced. Even though MPC is turning into the standard technique of industrial control and is widely accepted as an efficient and economical technique [10], management might often require a solid business case to be presented, before an investment into MPC is approved. However, engineers often only have indirect methods of estimating benefits (e.g. methods based on variance reduction), which are simplistic and based on experience, and as such do not offer a well-grounded case [9]. However, there are techniques to create a simple simulation of the MPC controlled plant and base the business case on it, which offers much more solid case [12].

When the investment into MPC is approved, the design phase can commence. Usually the first, and most difficult step is to devise a mathematical model of the plant to be controlled. This can account for up to 80% of the design work and investment [11]. The main reason for this difficulty is the inherent complexity of the plants that are to be controlled. They often include many loops and interdependent subsystems with complex dynamics, which are difficult to analyse [2]. The system size (number of controlled and manipulated variables) can easily reach into hundreds of tightly coupled variables [4], further complicating the analysis. Moreover, in the continuous process industry, the processes to be modelled are sometimes poorly understood [1].

To obtain the necessary data for modelling and later verification of the model, step tests are usually used. These excite the system by inputting series of step inputs and observe the reaction of the outputs. This way a black-box model (not derived from first principles) of the system can be obtained. However, there are challenges related to step tests, too. Since time constants in continuous process plants can reach hours, tests might take week or months to complete [2] [12]. The choice of tests is also a problem, as the inputs that would yield the most useful information are likely to take the system out of its operating range [7]. On the other hand, software packages are available, which can automatically use data from tests to aid model building

All these factors make the analysis of the plant and derivation of the model extremely challenging task. The situation is further aggravated by that lack of expert control engineers with the necessary experience and skill to perform these analyses [7] [12]. This makes the modelling of the plant the single largest challenge in implementing MPC in industry. On the upside, this motivates research into the area, and some techniques to simplify the design task have been proposed [5].

After the model is complete, the next challenge is to tune the system parameters (e.g. cost matrices) to achieve optimal

Literature

[2] "multivariable dynamics of a plant composed of several interconnected units with recycle, bypass, and heat streams, which can significantly increase the complexity of the plant-wide dynamics"

system with large dimension (many state variables) difficult to control online (computation power) and difficult to estimate/measure state variables (many long tests)

this paper however, suggests a control scheme independent of the system size

[1] processes to be transformed into the model are sometimes not even understood properly - inherent difficulty in creating models

[3] variable quality of feed – need to change the model continually

Changing inputs require additional layer of control to estimate/calculate its effect on the system overall

[4] “However, as the plant size increases, the complexity of the process model increases rapidly, making process modeling and control system design very difficult and expensive”

[5] schemes proposed to reduce time and effort it takes to fit an MPC to a plant (model fitting might not be that difficult)

[6] first MPC formulations used one norm cost functions solved by linear programming LP – then it moved to quadratic costs necessitating QP (more computationally expensive) but without some problems of LP

[7] MPC depends on expert resources, which are limited – because of this MPC should be designed to that future interventions are minimised

“Difficult MPC set-up may cause less robust tuning and model mismatch due to software use errors. Chances of sustainable performance are immediately reduced.”

Choice of one MPC vs. several small, loosely coupled controllers – one can optimise whole process, several are easier to monitor

“A common problem with this approach is that the type of plant experiments that yield the best data are also likely to perturb the process beyond current operating limit”

Model re-designs often done by non-experts – higher chance of error

Trade off: simple, adjustable model vs accurate and complicated model

Controller tuning difficult – ranges of variables can be huge + size of the system, no automated way to do this

To use lin MPC in non-lin model – define several operating points and respective tuning (gain scheduling technique)

There are expectations to be managed – if operators expect miracles from MPC and don’t get them, they might lose trust

Need to have performance review framework to ensure MPC is bringing value and not deteriorating (due to sensor/actuator wearing, model changes, etc)

If model needs to be adapted, there are packages which can do it automatically, but the changed model still needs to be checked by a person

[8] operator might have problems working with MPC even after training, if the user interface is poorly designed

[9] To achieve desirable performance, it is necessary to build an accurate model and to tune control parameters appropriately. However, both are difficult in practice due to process nonlinearity and changes in process characteristics

Most APC suppliers and users are required to report the benefit to management. Benefit estimation methods based on variance reduction are still carried out, but they are sometimes rudimentary and based on experience

[10] MPC regulators need maintenance to keep their performance on point

„To set the optimization problems, constraints and weighting matrixes are added to the control and prediction horizon. All these parameters can be considered tuning parameters”

Presents method to evaluate performance of MPC

[11] 80% of MPC design spent on model development

[12] MPC limited by expert time

Takes long tests to parametrize all variables in the system

Complicated maths to create a model, some processes need to be modelled from first principles

Fit additional sensors and actuators = cost

Continuous plants usually require only linear model

[13] MPC deteriorates over time as the plant changes by virtue of its dynamics

[14] “Processes degrade over time due to wear and tear on actuators, design changes and replaced parts, to name a few factors. Eventually the difference between the current process and initial model becomes too large and the control performance suffers, often leading to distrust in the MPC and the controller being turned off.”

---

[1] T. Glaser *et al.*, “Model predictive control of continuous drum granulation,” *J. Process Control*, vol. 19, no. 4, pp. 615–622, Apr. 2009.

[2] A. Mesbah, J. A. Paulson, R. Lakerveld, and R. D. Braatz, “Plant-wide model predictive control for a continuous pharmaceutical process,” in *2015 American Control Conference (ACC)*, 2015, pp. 4301–4307.

[3] C. F. W. Sanders, M. J. Hounslow, and F. J. Doyle, “Identification of models for control of wet granulation,” *Powder Technol.*, vol. 188, no. 3, pp. 255–263, Jan. 2009.

[4] S. Xu and J. Bao, “Distributed control of plantwide chemical processes,” *J. Process Control*, vol. 19, no. 10, pp. 1671–1687, Dec. 2009.

[5] V. Tzovla and A. Mehta, “Simplified Intergrated approach Model Predictive Control Implementation.” Instrument Society of America, 2000.

[6] M. Short, “A Simplified Approach to Multivariable Model Predictive Control,” *Int. J. Eng. Technol. Innov.*, vol. 5, no. 1, pp. 19–32, 2015.

[7] M. G. Forbes, R. S. Patwardhan, H. Hamadah, and R. B. Gopaluni, “Model Predictive Control in Industry: Challenges and Opportunities.”

[8] S. Guerlain, G. A. Jamieson, P. Bullemer, and R. Blair, “The MPC elucidator: a case study in the design for human-automation interaction,” *IEEE Trans. Syst. Man, Cybern. - Part A Syst. Humans*, vol. 32, no. 1, pp. 25–40, 2002.

[9] M. Kano and M. Ogawa, “The state of the art in chemical process control in Japan: Good practice and questionnaire survey,” *J. Process Control*, vol. 20, no. 9, pp. 969–982, Oct. 2010.

[10] V. Botelho, J. O. Trierweiler, M. Farenzena, and R. Duraiski, “Perspectives and challenges in performance assessment of model predictive control,” *Can. J. Chem. Eng.*, vol. 94, no. 7, pp. 1225–1241, Jul. 2016.

[11] Z. Sun, S. J. Qin, A. Singhal, and L. Megan, “Performance monitoring of model-predictive controllers via model residual assessment,” *J. Process Control*, vol. 23, no. 4, pp. 473–482, Apr. 2013.

[12] J. Birk, “Model Based Control from an Application Point of View in Process Industry.” Imperial College London, 2017.

[13] G. Ji *et al.*, “Design and implementation of an MPC system with model error detection,” in *2013 Nirma University International Conference on Engineering (NUiCONE)*, 2013, pp. 1–5.

[14] C. A. Larsson, C. R. Rojas, X. Bombois, and H. Hjalmarsson, “Experimental evaluation of model predictive control with excitation (MPC-X) on an industrial depropanizer,” *J. Process Control*, vol. 31, pp. 1–16, Jul. 2015.