Martin Opatovsky (CID 00862881)

# Part A

## Introduction problems

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. This definition of the problem allows for the use of controller implemented for previous assignments with only minor changes.

A choice was made to use controller with only hard constraints instead of controller with soft constraints implemented. The controller with soft constraints takes significantly longer to compute, since the slack variables increase the size of the system to be optimised [1]. In testing, the controller did not compute 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.

## Algorithm

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. 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 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 because 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 is the distance of the longer side from side of the square. This way, the crane car must follow the side of the defined square very tightly.

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 (positions) are then cycled through based on the time of simulation/run of the hardware. Time of the simulation is 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 *warp*, 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.

## Results and discussion

To evaluate the controller, a metric of “squarness” was developed. The trace of the pendulum was 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 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 Table I below. Note that the tuning was done considering the results of the non-linear simulation.

The trace of the best controller (i.e. the one with smallest area traced) is shown in Fig. 1 on the nest page. A case can be made for the controller which trace is shown in Fig. 2, since the trace is smoother (without oscillation in the corners) and may be more appealing to a human observer. This controller has the same parametrisation as the best one, apart

1. parameters of the best controller

|  |  |  |  |
| --- | --- | --- | --- |
| **Q(1,1), Q(3,3)** | 10 | **TS** | 1/30 |
| **Q(2,2), Q(4,4)** | 1 | **Angle constraint** | 5 deg. |
| **Q(5,5), Q(7,7)** | 50 | **N** | 20 |
| **Q(6,6), Q(8,8)** | 2 | **Margin** | 0.001 |
| **R(1,1), R(2,2)** | 0.01 | **Warp** | 0.1 |
| **P(1,1), P(3,3)** | 5 |  |  |
| **P(5,5), P(7,7)** | 30 |  |  |

from *warp* = 0.18. The solutions of both controllers calculated in time (i.e. quicker than the sampling time) and performed better than the non-constrained controller, which suffered significant oscillations in the corners.

There are few points to be noted. First, by optimising the parameters for non-linear simulation, the performance of the linear simulation suffered. This can be attributed to the mismatch between the two models. Linear simulation was also more sensitive to tight constraints margin, and could only run with *margin* = 0.03. Lower margin, however, improved the performance of the non-linear simulation. This can be explained by the internal differences between the linear and non-linear simulation, which it was not possible to explore in more detail.

Next, in the best simulation, the crane is doing small stabilising movements around the corners to minimise the oscillations. This, however, might not be possible in reality due to the stickiness of the crane. Interesting behaviour was also observed with the angle constraints – loosening of these lead to improved results. This is probably due to the crane not performing so many stabilising moves to decrease the string angle, resulting in less erratic movement hence improved trace.

There was also a trade-off between the angle and position costs. If the former was increased, the angle deviation

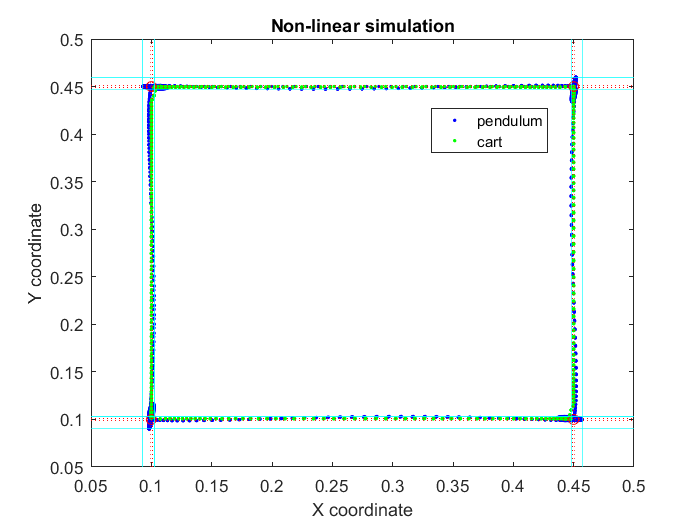


Figure 1: trace of the best controller, red dashed lines are constraints, cyan lines are inscribed and circumscribed rectangles.

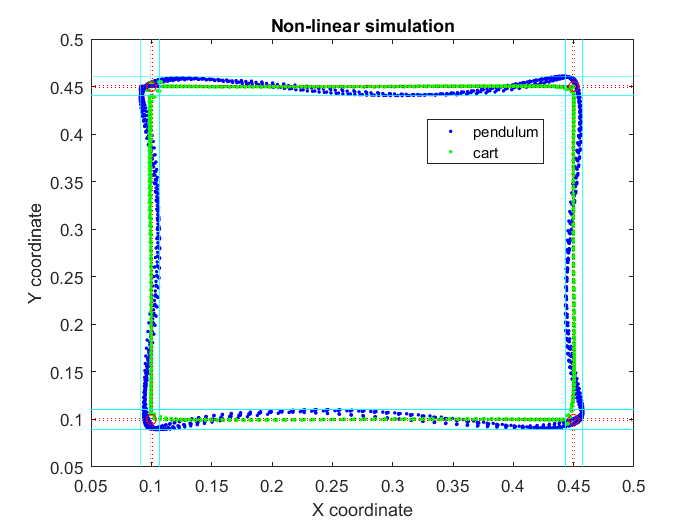


Figure 2: trace of the smoother controller, constraints in red dashed lines, inscribed and circumscribed rectangles in cyan

suffered, resulting in poorer trace. If the angle cost was increased, the crane was susceptible to not reaching its final state, even if final state cost was increased. The latter situation, however, is more easily solved with re-definition of the desired square. Hence the angle cost in the final controller is higher than the position cost.

# Part B

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.

## Design phase

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 [2], management might 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 [3]. On the other hand, there are techniques to create a simple simulation of the MPC controlled plant and base the business case on it, which offers much more persuasive case [4].

When the investment into MPC is approved, the design phase can begin. 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 effort and investment [5]. The modelling is especially problematic, if detailed models derived from first principles are needed. The plants to be modelled often include many loops and interdependent subsystems with complex dynamics, which are difficult to analyse [6]. The system size (number of controlled and manipulated variables) can easily reach into hundreds of tightly coupled variables [7], further complicating the analysis. Moreover, in the continuous process industry, the processes to be modelled are sometimes poorly understood [8].

If only a black-box model is required, the analysis is somewhat simplified. For this, software packages capable of model identification can be used [9]. The model is constructed from data obtained in plant tests. However, there are challenges connected with the tests as well. Since time constants in continuous process plants can reach hours, tests might take week or months to complete [4], [6]. 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 [10].

After the model is complete and the controller designed, the next challenge is to tune the controller parameters to achieve optimal operation. Cost matrices, soft and hard constraints can all be considered tuning parameters [2]. Since the size of the system is usually large and there can be tight coupling between the parameters, the task is inherently complicated [7]. Moreover, the tuning is mostly manual job and no effective automatization procedure exists [10]. After the tuning is complete, validation of the model can be performed, for which plant test data are again necessary.

All these factors make the design procedure 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 [4], [10]. This makes the modelling of the plant and design of optimal controller the single largest challenge in implementing MPC in industry. On the upside, this motivates research into the area, and techniques to simplify and shorten the design task have been proposed [11].

Aside from the technical challenges, there are also challenges connected with the human factor. The MPC control operators need to be knowledgeable and skilled in operating the new control system. For this they need extensive training, which imposes additional financial and time cost [4]. Moreover, there are also psychological factors at play. If at the beginning of the MPC integration unrealistic expectations about its performance were formed, the operators might be later dissatisfied with the new controller. In such situation, they are more likely to not use it properly, or even turn it off completely [10]. If the user interface of the controller is complicated and not user-friendly, it can lead to similar negative consequences [12].

## MPC lifecycle

The challenges connected with the continuous use of MPC controller are tightly coupled with the ones solved during the design phase. This is because design decisions naturally affect the operation of the controller and its ability to cope with any emergent situations. Discussions of the following issues will therefore often reach back to the design stage.

The first issue that can be identified is variance of the inputs to the controlled plant. Naturally, in the continuous process industry the ingredients of the product will be of varying quality and characteristics (e.g. grain size, moisture, density). Due to legal requirements, the product must keep its quality irrespective of the inputs. The controller must therefore adapt to changing circumstances. This can be done for instance by allowing for a range of input qualities in the design process, or by defining several operation points of the plant based on the quality categories of the inputs [13].

Possibly the largest issue that has to be solved during the lifecycle of the MPC controller is changing characteristics of the plant. This can be attributed to slow sensor and actuator wear, process or recipe changes and replaced parts [14], [15]. For this reason, there should exists a business and engineering framework to continually review the performance of the MPC controller to assess, whether a change to the model is necessary [2], [10]. However, the MPC is usually designed by an outside specialist contractor company, which is typically not available to support the system during its entire lifecycle. The adjustments to be made are then responsibility of the in-house engineers and operators, who might lack the necessary skills [10]. This creates another consideration in the design phase, where simpler controller will be easier to maintain and adjust, however, likely more imprecise in its operation. Moreover, a model redesign will usually cause disruption of the production process, since testing is required similarly as in the initial design process [15].

An alternative approach is to design the MPC so that it is capable of adjusting its model automatically. There are solutions available which can support this. However, the automatic model adjustment relies of continuous excitation of the system to gather data based on which the model can be re-adjusted. This necessarily means some performance of the controller is sacrificed [15]. Moreover, the controller operators need to be comfortable with supporting such adaptive system, which might be problematic due to aforementioned lack of skilled engineers and costs associated with their training [2].

If the business employing MPC in its processes wants to maintain its competitive edge, it should also be following the latest technologies available and deploying them into their operations. These can be for instance integrated real-time optimisation or economic MPC solutions [4]. To be able to do this, the business naturally has to employ additional personnel involved with research and development, imposing further costs and hiring issues (due to lack of skilled labour).

References

[1] G. Frison and J. B. Jørgensen, “Efficient solvers for soft-constrained MPC,” in *Proceedings of the 19th Nordic Process Control Workshop*, 2015.

[2] 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.

[3] 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.

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

[5] 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.

[6] A. Mesbah, J. A. Paulson, R. Lakerveld, and R. D. Braatz, “Plant-wide model predictive control for a continuous pharmaceutical process,” in *Proceedings of the American Control Conference*, 2015, vol. 2015–July, pp. 4301–4307.

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

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

[9] Y. Zhu, “Progress in MPC Identification : A Case Study on Totally Closed-Loop Plant Test 2 Key Issues in MPC Identification,” in *Procedings of the ERTC Computing Conference*, 2003, pp. 1–11.

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

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

[12] 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.

[13] 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.

[14] 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.

[15] 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.