TTK4210 Advanced Control of Industrial Processes
Department of Engineering Cybernetics
Norwegian University of Science and Technology
Spring 2018 - Solution to assignment 7

The MPC controller is designed based on a state-space model, with the *change* of the manipulated variable as the free variable in the optimization problem, as described in the MPC chapter of the lecture notes. You can run the controller in the following way:

1. Define the model by running `modell.m`
2. Construct input matrices by running `MPCinput.m`
3. Simulate MPC control using `mpcsim.m`

Depending on which parts of the problem you want to run, you will have to do simple changes (add/remove commenting) in `mpcsim.m`. The routine `mpcsim.m` runs both `stateref.m`, which calculates reference values for the state vector, and `mpccalc.m`, which constructs and solves the MPC problem.

State references The state reference is not necessarily uniquely defined from the reference for the measurement. Here we have used the state vector that satisfies the measurement reference in stationary conditions. Note that the “input” in the augmented model will then always be zero, since we can not have any changes in the real input in stationary conditions. This is solved by using a pseudo inverse.

Stability To be able to guarantee stability you need to

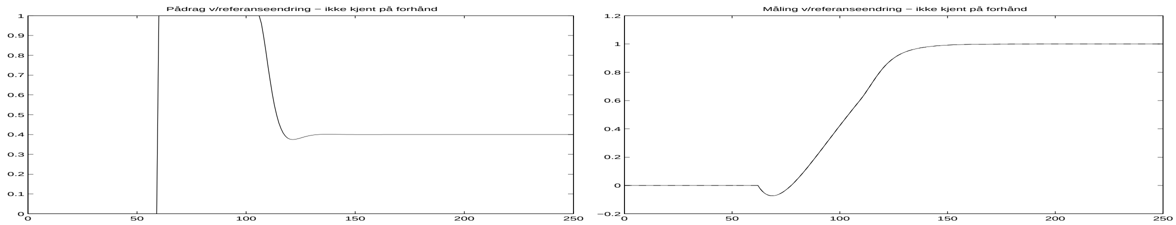
- have sufficient weighting for the last state in the control horizon, and
- ensure that the constraints can be satisfied also for a window after the control horizon.

In this formulation we have assumed that we after the control horizon use

$$u = -K\left(x - x_{ref}\right)$$

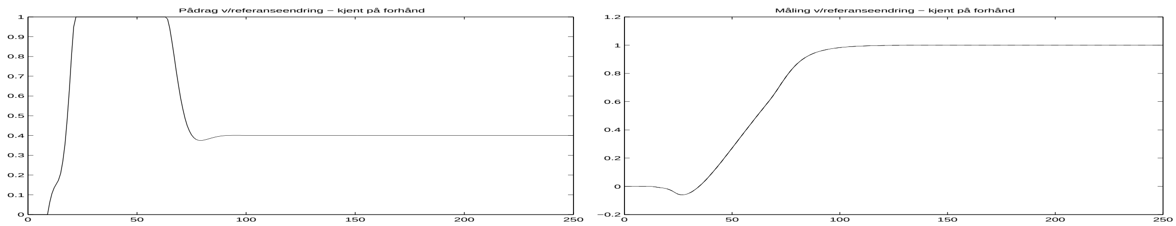
where K is an LQ optimal controller (in relation to the weighting for the last state vector).

It turns out that you can easily achieve stability with a moderate control horizon, even without consideration of the constraints after the control horizon (as long as reasonable weights are chosen). When you then consider these constraints, they will turn inconsistent, unless you increase the control horizon significantly! This is related to the inverse response for the model. In the simulations, a sample time of 1 and a control horizon of 50 is used, and the constraints are considered for a window of 20 time steps after the control horizon.



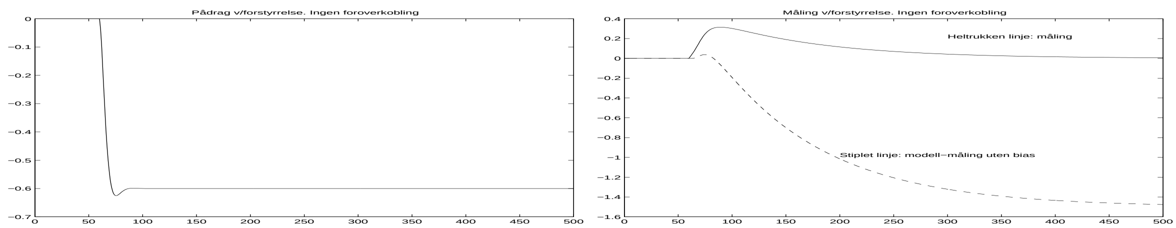
(a) Input u (b) Output y

Figure 1: Results for exercise **a**



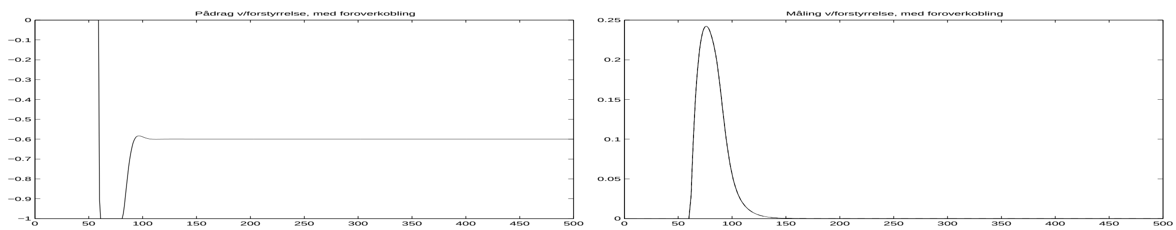
(a) Input u (b) Output y

Figure 2: Results for exercise **b**



(a) Input u (b) Output y

Figure 3: Results for exercise **c**



(a) Input u (b) Output y

Figure 4: Results for exercise **d**

More thorough analysis, along the lines of what is presented in the course notes, may show that this extra 'constraint window' of 20 samples may be reduced. In step response based MPC formulations it is common to have a control horizon of three times the dominating time constant. In this case it means 300 time steps for a sample time of 1. Normally you would choose a sample time of $5-10$ to decrease the size of the problem. Thus, if you wish rapid control, the sample time might be limited.

Model update The model is updated by a simple bias on predicted measurements, i.e. we assume that the existing difference between model and process will stay there in the future. This bias is used to adjust the references for the model states.

Results The results (Figures 1–4) show that knowing the reference step changes in advance does not improve the performance significantly, while a forward-connection from the disturbance measurement is significant. A stepwise reference change is dynamically similar to a stepwise disturbance on the measurement, and the model update assumes disturbances with this shape. However the real disturbance has slow dynamics, and thus the simple bias update works quite bad. We could avoid this by using a more advanced model update, for example with an enhanced Kalman filter using a disturbance model.