Comparison of Linear Controllers for Nonlinear, Open-loop Unstable Reactor

Mohammad Khalili and Bernt Lie

University of South-Eastern Norway, Bernt.Lie@usn.no

Abstract

Many industrially important reactors are operated over a regime where they exhibit nonlinear behavior. Exothermal reactors are often open-loop unstable. For such reactors, safe operation with good performance requires good control. We have considered linear controllers and tested these on a simple chemical engineering non-linear, open loop unstable model and an extension of the model in order to do a basic test of model uncertainty. Specifically, a PI controller has been tuned and tested, the operation of an Extended Kalman Filter (EKF, KF) has been tuned and tested. Based on feedback from estimated states in the EKF, a linear quadratic controller with integral action (LQG+I) has been tuned and tested; the study has been carried out using MATLAB for KF and LQG+I tuning; the remaining study has been carried out in a Jupyter Notebook using Python in tandem with Modelica. The PI controller lead to negative cooling water temperature upon a step change in temperature reference. When constraining the input to liquid water with anti-windup, PI control gives considerable undesirable overshoot in the reactor temperature. The LQG+I controller performs much better wrt. temperature overshoot. Overall, the reported work has demonstrated how a modern simulation set-up (OpenModelica + Python) can be used for model based control analysis and design.

Keywords: nonlinear reactor, model uncertainty, control design, PI control, LQG control

1 Introduction

1.1 Background

Many industrially important reactors are operated over a regime where they exhibit nonlinear behavior. Exothermal reactors are often open-loop unstable. For such reactors, safe operation with good performance requires good control.

It is of interest to compare how how modern tools can be used to efficiently and accurately assess the performance of both standard PI controllers and more advanced multivariable controllers. In the simplest case, classical output feedback linear quadratic reactors with state estimation (LQG) could be considered, possibly extended with constraint handling as in linear model predictive control (MPC). A further extension could involve nonlinear MPC and nonlinear state estimation, but also nonlinear feedback controllers such as passivity based controllers and nonlinear observers.

To make the comparison concrete, it is useful to introduce a representative model which is nonlinear and open loop unstable, with constraints in inputs and outputs. It is also of interest to consider systems which are multivariable input with multivariable outputs (MIMO), and also systems with stable or unstable zero dynamics.

1.2 Previous Work

Classical control is well developed (Åstrøm and Murray, 2008), (Seborg et al., 2011); the same is true for linear quadratic optimal control (Anderson and Moore, 1989). Model predictive control is newer (Maciejowski, 2002), (Rawlings et al., 2017). State estimation is described, e.g., in (Simon, 2006). Basic nonlinear control is described in (Henson and Seborg, 1997).

A nonlinear, open loop unstable single input - single output (SISO) reactor model is described in (Seborg et al., 2011); an extension of the model is described in (Sund et al., 2018). The Modelica modeling language is useful for encoding dynamic models (Fritzson, 2015), and extended analysis can be achieved by combining Modelica with Python (Lie et al., 2016).

1.3 Organization of Paper

Section 2 describes the chosen illustrative process model, together with the control problem. In Section 3, a PI controller for the open-loop unstable process model is developed, including a description of an anti-windup algorithm that is easy to implement in Modelica. Further, continuous LQ control with integral action is described, and a controller is tuned. Next, a discrete time constant gain state estimator is discussed. In Section 4, the state estimator is tested for both the original model and an extended model, then the PI controller is tested for both models. Finally, the LQG+I controller is tested. In Section 5, the results are discussed, and some conclusions are drawn. In three appendices, the extended model is described, a nominal linear model is given, and the extended LQ+I system is given.

https://doi.org/10.3384/ecp18153185

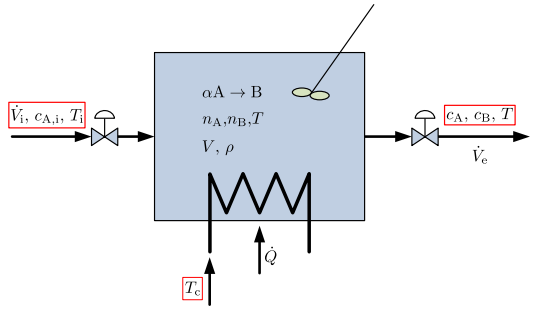

Figure 1. Cooled liquid reactor with reaction $a\mathrm{A} \to \mathrm{B}$.

2 Problem Description

2.1 Process Overview

We consider a liquid reactor of constant volume V, with influent volumetric flow rate $\dot{V}_\mathrm{i}$, influent concentration $c_\mathrm{A,i}$ of reactant A carried via an inert solvent S,[1] and influent temperature T_i, Fig. 1.

It is of interest to convert species A into species B through an exothermic reaction

$$a\mathrm{A} \to \mathrm{B}; \tag{1}$$

the products are carried out of the reactor via solvent S. To keep control of reactor temperature T, heat rate $\dot{Q}$ is added by flowing a liquid at temperature T_c through the tube side of a coil/heat exchanger. With a high flow rate of the cooling liquid, T_c is constant through the heat exchanger, and the heat rate can be described as

$$\dot{Q} = \mathscr{U} A \left(T_\mathrm{c} - T\right). \tag{2}$$

where $\mathscr{U} A$ is a parameter. If $T_\mathrm{c} < T$, $\dot{Q} < 0$ and the reactor is cooled.

The rate of reaction r is given as

$$r = k c_\mathrm{A}^a \tag{3}$$

where a is the reaction order and k is given by the Arrhenius expression

$$k = k_0 \exp\left(-\frac{E/R}{T}\right). \tag{4}$$

The operation of the reactor is influenced by inputs $\dot{V}_\mathrm{i}$, $c_\mathrm{A,i}$, T_i and T_c, and it is of interest to study how these influence the outputs c_A, c_B and T.

Assuming that the reaction mixture constitutes an ideal solution, a DAE formulation of the system model is discussed in (Sund et al., 2018). Appendix A suggests how the DAE model can be manipulated into an implicit ODE, and furthermore how assuming first order reaction ($a = 1$), constant reactor density, and constant thermal parameters leads to the model in (Seborg et al., 2011). A Linear Time Invariant (LTI) approximate model in form

$$\frac{dx^\delta}{dt} = Ax^\delta + Bu^\delta \tag{5}$$

$$y^\delta = Cx^\delta + Du^\delta \tag{6}$$

is given in Appendix B where superscript δ indicates deviation from a nominal solution, based on the conditions suggested in (Sund et al., 2018) and computed using a Python API for OpenModelica (Lie et al., 2016).

2.2 Control Problem

Although the case study has inputs $u = \left(\dot{V}_\mathrm{i}, c_\mathrm{A,i}, T_\mathrm{i}, T_\mathrm{c}\right)$ and outputs $y = \left(c_\mathrm{A}, c_\mathrm{B}, T\right)$, in a control problem one may choose to control the temperature $y = T$ by manipulating the input $u = T_\mathrm{c}$. In that case, the additional inputs $\left(\dot{V}_\mathrm{i}, c_\mathrm{A,i}, T_\mathrm{i}\right)$ may be considered disturbances.

Realistic controllers must utilize output feedback; internal states are not known in practice, thus state feedback is infeasible. We compare the operation of linear controllers such as proportional (P) and linear quadratic (LQ) and linear quadratic with state estimation (LQG), as well as their versions with integral action (PI, LQ+I, LQG+I). Some key differences are that P+PI and LQG, LQG+I are output feedback controllers, while LQ, LQ+I are unrealistic state feedback controllers. The main difference between P+PI and LQ(G), LQ(G)+I controllers are that P+PI controllers do not use much information about the system dynamics, while LQ(G), LQ(G)+I controllers include a dynamic model[2].

With digital controllers, measurements are available at given sample times, and it is common to use zero-order hold of the control input between each sample. In modern control systems, also P+PI controllers are usually implemented as digital controllers.

3 Controller Design

3.1 PI Controller

Using the linearized model in Appendix B with T_c as control input, the closed loop matrix A_cl with a proportional controller (P controller) is

$$A_\mathrm{cl} = A - K_\mathrm{p} B_{:,1} C \tag{7}$$

where $B_{:,1}$ is the first column of B. Looping through $K_\mathrm{p} \in [-1, 8]$ leads to the closed loop eigenvalues as depicted in Figure 2.

The P-controller stabilizes the system for $K_\mathrm{p} \gtrapprox 1.2$; $K_\mathrm{p} = 5.7$ gives two real, closed loop eigenvalues/poles at approximately $\lambda \approx -5$, which implies closed loop time constants $\tau_j \approx \frac{1}{5} = 0.2$.

For a proportional + integral controller, it is reasonable to let the reset time (= integral time) be, say, 10 times larger than the closed loop time constants of a P controller. Thus, the PI controller

$$T_\mathrm{c}(s) = T_\mathrm{c}^* + K_\mathrm{p} \frac{1 + T_\mathrm{int} s}{T_\mathrm{int} s} \cdot e(s) \tag{8}$$

$$e(s) = T_\mathrm{ref}(s) - T(s) \tag{9}$$

[1] *Inert* implies non-reacting.

[2] Essentially, LQ controllers give "intelligent" derivative action.

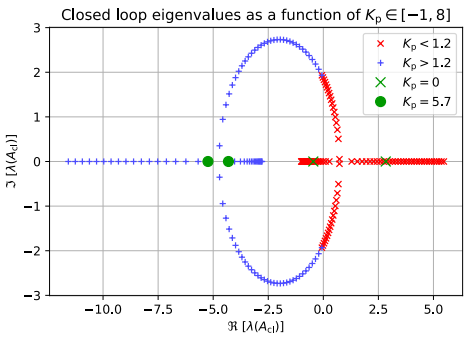

Figure 2. Root locus plot $\lambda\left(A_{\mathrm{cl}}; K_{\mathrm{p}}\right)$ for $K_{\mathrm{p}} \in [-1, 8]$.

with $K_{\mathrm{p}} = 5.7$ and $T_{\mathrm{int}} = 2$ may be an acceptable choice.[3] Nominal input T_{c}^* is not needed with integral action, but is useful to avoid an initial "kick" in the control action. T_{ref} is the reference temperature. If we let $T_{\mathrm{int}} \to \infty$, the controller becomes a P controller.

In the time domain, we can express the PI controller as

$$T_{\mathrm{c}} - T_{\mathrm{c}}^* = K_{\mathrm{p}} e + \tilde{T}_{\mathrm{c}} \tag{10}$$

$$\frac{d\tilde{T}_{\mathrm{c}}}{dt} = \frac{K_{\mathrm{p}}}{T_{\mathrm{int}}} e. \tag{11}$$

To handle constraints for $T_{\mathrm{c}} \in [4, 96]\,^{\circ}\mathrm{C}$, if $T_{\mathrm{c}} = K_{\mathrm{p}} e + T_{\mathrm{c}}^* + \tilde{T}_{\mathrm{c}}$ breaks this constraint, we set T_{c} equal to the constraint and $\frac{d\tilde{T}_{\mathrm{c}}}{dt} = 0$ to avoid controller wind-up.

3.2 LQ+I Controller

We consider an infinite horizon optimal linear quadratic (LQ) controller with cost function

$$\mathscr{J} = \frac{1}{2} \int_0^{\infty} \left(e_y^T Q_y e_y + e_u^T R e_u \right) dt \tag{12}$$

where matrix $Q_y > 0$ puts weight on *quality* deviation $e_y = y - y_{\mathrm{ref}}$ and matrix $R > 0$ puts weight on the *regulating* input $e_u = u - u^*$. Postulating a reference state x_{ref} such that $y_{\mathrm{ref}} = C x_{\mathrm{ref}}$, this gives the standard LQ cost function

$$\mathscr{J} = \frac{1}{2} \int_0^{\infty} \left(e_x^T Q e_x + e_u^T R e_u \right) dt \tag{13}$$

where $e_x = x - x_{\mathrm{ref}}$ with $x_{\mathrm{ref}} = \begin{pmatrix} T_{\mathrm{ref}} & c_{\mathrm{A,ref}} \end{pmatrix}^T$ and $c_{\mathrm{A,ref}}$ is found by solving Eq. 36 in steady state:

$$c_{\mathrm{A,ref}} = \frac{\dot{V}_{\mathrm{i}}/V}{k_0 \exp\left(-\frac{E/R}{T_{\mathrm{ref}}}\right) + \dot{V}_{\mathrm{i}}/V} c_{\mathrm{A,i}}, \tag{14}$$

while $Q = C^T Q_y C \geq 0$. The solution is

$$e_u = -K_{\mathrm{c}} e_x \Rightarrow u = u^* - K_{\mathrm{c}} e_x \tag{15}$$

where K_{c} can be found, e.g., using the MATLAB control toolbox as `>> Kc = lqr(A,B,Q,R)`. Here, it is required that (A, B) is stabilizable and (A, C) detectable. The closed loop eigenvalues are given by

$$A_{\mathrm{cl}} = A - B K_{\mathrm{c}}. \tag{16}$$

See, e.g., (Anderson and Moore, 1989) for a classical treatment.

A relatively common rule-of-thumb for choosing weight matrices is to let them be diagonal with elements

$$Q_{y,j,j} = 1/\varepsilon_{y,j}^2 \tag{17}$$

$$R_{j,j} = \rho/\varepsilon_{u,j}^2 \tag{18}$$

where $\varepsilon_{y,j}$ and $\varepsilon_{u,j}$ are some acceptable deviations in $e_{y,j}$ and $e_{u,j}$, respectively, while ρ is some scalar tuning parameter for R which is manipulated to get acceptable closed loop eigenvalues. We could choose, say, approximately 1% of nominal values:

$$Q_y = \frac{1}{3^2}, \qquad R = \rho \frac{1}{3^2}. \tag{19}$$

If $\rho \to 0$, this should give a fast closed loop system since this means that the control input is "cheap". Figure 3 shows the closed loop eigenvalues $\lambda\left(A_{\mathrm{cl}}\right)$ as ρ varies logarithmically over $\left[10^1, 10^{-1}\right]$, and illustrates the well known result that one eigenvalue approaches the ("stable") zero at $\lambda = -2$ when $\rho \to 0$ (Kwakernaak and Sivan, 1972). From an input-output perspective, the closed loop pole at $\lambda \approx -2$ will approximately cancel out the zero; see transfer function in Appendix B.

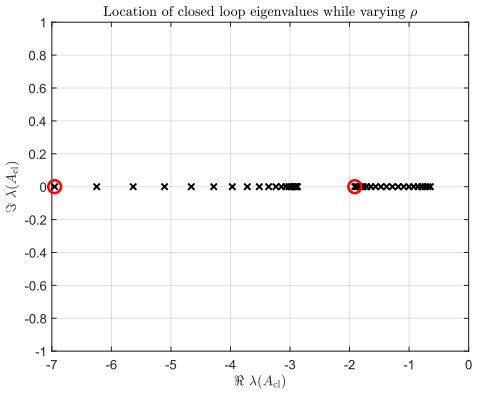

Figure 3. Variation of closed loop eigenvalues $\lambda\left(A_{\mathrm{cl}}\right)$ with ρ varying logarithmically over $\left[10^1, 10^{-2}\right]$; red circles indicate "cheap control".

A decent compromise would be $\rho = 0.2$, leading to eigenvalues at $\lambda\left(A_{\mathrm{cl}}\right) = (-5.1761, -1.8246)$ where the fast eigenvalue would be similar to that achieved via P control. The resulting state feedback gain is

$$K_{\mathrm{c}} = \begin{pmatrix} 4.4838, & 107.4579 \end{pmatrix}. \tag{20}$$

This standard LQ controller gives steady deviation in e_y. One way to achieve zero steady deviation in e_y is to augment the system with state

$$\frac{dz}{dt} = e_y \tag{21}$$

[3] The integral time is denoted T_{int} in order to make a distinction between integral time and influent temperature, T_{i}.

and extend the cost function to

$$\tilde{\mathscr{J}} = \mathscr{J} + \frac{1}{2}\int_0^\infty z^T Q_z z dt : \tag{22}$$

unless $e_y \to 0$, $\tilde{\mathscr{J}}$ will be unbound and can not be considered optimal.[4] The extended control problem with integrator is solved as above, with augmenting the system into state $\tilde{x} = (x, z)$, and likewise for $\tilde{Q}$, see Appendix C.[5]

A possible rule-of-thumb for Q_z is $Q_z = Q_y/\rho_z$ where first Q_y and R are chosen to give decent closed loop poles of the original system, and then ρ_z is tuned to give reasonably fast integral action. Setting $\rho_z = 4$ leads to

$$\tilde{K}_{\mathrm{cl}} = \begin{pmatrix} 4.7073, & 106.0020, & 1.1180 \end{pmatrix} \tag{23}$$

and closed loop eigenvalues $\lambda\left(\tilde{A}_{\mathrm{cl}}\right) = (-5.1568, -0.5011, -1.8105)$.

Using a digital controller with fixed sample time, a discrete version of the LQ controller can easily be found by (i) discretizing the system $(A,B,C,D) \overset{t_s}{\to} (A_d, B_d, C, D)$ with sample time t_s using, e.g., MATLAB function `c2d(sys,ts)`, and related optimal control function `>> Kc = dlqr(Ad,Bd,Q,R)`.

3.3 State Estimator

Because measurements normally are available in discrete time with sample time t_s, it is more convenient to operate with a discrete time model. Thus, we assume a linearized model

$$x_{t+1} = A_d x_t + B_d u_t + G_d w_t \tag{24}$$

$$y_t = C x_t + D u_t + v_t \tag{25}$$

where B_d is discretized input matrix $B_{:,1}$, G_d is discretized input matrix $B_{:,2:4}$, w_t is process disturbance and v_t is measurement noise. We will assume that w and v are uncorrelated, zero mean white noise with a Gaussian distribution, and co-variance matrices W and V, respectively. Thus

$$\mathrm{E}\left\{w_t w_{t+\tau}^T\right\} = W \cdot \delta_\tau \tag{26}$$

$$\mathrm{E}\left\{v_t v_{t+\tau}^T\right\} = V \cdot \delta_\tau \tag{27}$$

$$\mathrm{E}\left\{w_t v_{t+\tau}^T\right\} = 0 \tag{28}$$

where δ_τ is the (single argument) Kronecker delta. With input T_c used as control variable, inputs $\left(T_i, \dot{V}_i, c_{A,i}\right)$would be natural disturbances. With *a priori* estimates $\hat{x}_{t|t-1}$ given by solving the non-disturbed, nonlinear model over sample time t_s with initial value $\hat{x}_{t-1|t-1}$, and *a priori* measurement $\hat{y}_{t|t-1} = g\left(\hat{x}_{t|t-1}, u_{t-1}\right)$, we can find the *a posteriori* estimate $\hat{x}_{t|t}$ from Kalman's expression as

$$\hat{x}_{t|t} = \hat{x}_{t|t-1} + K_e\left(y_t - \hat{y}_{t|t-1}\right). \tag{29}$$

[4] Even if $e_y \to 0$, most likely $z \neq 0$ for infinite time leading to infinite value for $\tilde{\mathscr{J}}$; still, some optimum may be found. However, if e_y is different from zero at infinite time, no optimum can be found.

[5] Seemingly, integral action can be achieved in a simpler way by replacing weight on u in $\mathscr{J}$ with weight on $\dot{u}$. However, this requires an unbiased model to work, with added complexity of estimating the bias in a state estimator.

Here, a constant Kalman gain K_e can be found, e.g., by using MATLAB's control toolbox, `>> Ke = dlqe(Ad, Gd, C, W, V)`. It is required that (A_d, C) is detectable and (A_d, G_d) is stabilizable in order for the estimator to work.

In practice, W and V are not known accurately. A possible rule-of-thumb for tuning the state estimator could be as follows. Let $\hat{\varsigma}_{w,j}$ be assumed variation in w_j over time horizon τ_j. Then choose diagonal elements of W as $W_{j,j} = \left(\hat{\varsigma}_{w,j}\frac{t_s}{\tau_j}\right)^2$.[6] For the measurement noise, assume instantaneous standard deviation $\hat{\sigma}_{v,j}$, but we set diagonal elements to $V_{j,j} = \mu\hat{\sigma}_{v,j}$ where μ is a tuning parameter to give suitable closed loop performance of the estimator: if $\mu \to 0$, we assume little measurement noise and the estimator should become fast. Ideally, μ should be unity.

We could choose, say, approximately 1% of nominal values for $\hat{\varsigma}_{w,j}$ in the course of $\tau_{w,j} = \tau$, and assume a 1% measurement standard deviation:

$$W = \left(\frac{t_s}{\tau}\right)^2 \begin{pmatrix} \left(\frac{300}{100}\right)^2 & 0 & 0 \\ 0 & \left(\frac{100}{100}\right)^2 & 0 \\ 0 & 0 & \left(\frac{1}{100}\right)^2 \end{pmatrix}, \quad V = \mu \cdot \left(\frac{300}{100}\right)^2. \tag{30}$$

To simplify matters, we will assume direct disturbance to each state, $w_t' = G_d w_t$, where

$$W' = G_d W G_d^T. \tag{31}$$

Choosing $t_s = \frac{1}{10}$, $\tau = t_s$, and $\mu = 1$ leads to Kalman gain

$$K_e = \begin{pmatrix} 0.45942 \\ -0.0030453 \end{pmatrix} \tag{32}$$

and estimator closed loop eigenvalues at $\lambda = (0.74733, 0.91768)$ which are well within the unit circle, hence stable.

The fast mode in the *controller* has a time constant of approximately $\tau_j \approx \frac{1}{5} = 0.2$, which implies that a step change is damped to $\exp(-1) \approx 0.37$ in 0.2 min, or to approximately 0.74 in $t_s = 0.1$ min. This means that the fast mode of the controller in a *discretized* system is approximately $\lambda_d \approx 0.74$. In other words: the Kalman Filter gain in Eq. 32 makes the estimator *slower* than the controller. If we tune μ to $\mu = 10^{-2}$ instead, we find

$$K_e = \begin{pmatrix} 0.75947 \\ -0.0018525 \end{pmatrix} \tag{33}$$

with eigenvalues at $\lambda = (0.37741, 0.80854)$. Reducing μ further, the second eigenvalue is stuck at a value around 0.8; the first eigenvalue, is, however, twice as fast as the closed loop controller eigenvalue.

3.4 LQG+I Controller

For linear systems, the controller and the estimator (with Gaussian noise, LQG) can be tuned independently. We

[6] The assumed discrete time standard deviation would be $\hat{\sigma}_{w,j} = \hat{\varsigma}\frac{t_s}{\tau_j}$.

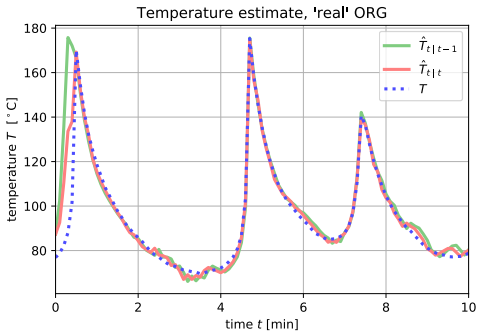

Figure 4. Real temperature T, a priori estimate $\hat{T}_{t|t-1}$, and *a posteriori* estimate $\hat{T}_{t|t}$ when estimator model is identical to the real system — except initial states.

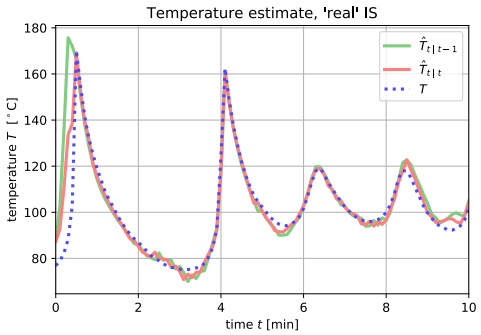

Figure 5. Real temperature T, a priori estimate $\hat{T}_{t|t-1}$, and *a posteriori* estimate $\hat{T}_{t|t}$ when estimator model differs from the real system, including different initial states.

will assume that this is true here, too. Because it doesn't make sense to assume "disturbance" on the integral state, we will base the estimator on the original model with state x, and not on state $\tilde{x}$.

4 Simulation Results

4.1 Operation

The model as given in Appendix A is used for all designs (P+PI controller tuning, LQ(G)+I design); this original model is referred to as the `org` model. A slightly more complex model (ideal solution assumption, `is`) is used. White noise disturbances, additive to the nominal values of $\left(T_\mathrm{i}, \dot{V}_\mathrm{i}, c_{\mathrm{A,i}}\right)$ in (Sund et al., 2018), are used with values as discussed in Section 3.3 for all simulations. For testing the state estimator, the measurement noise indicated in Section 3.3 is used with parameter $\mu = 1$ in the Kalman Filter tuning, together with a 3% deviation in the initial states of the estimator. Because a continuous time PI controller is implemented in Modelica, it is difficult to have continuous measurement noise: for the controller comparisons, the measurement noise is set to zero, and the tuning parameter of the Kalman Filter is set to $\mu = 10^{-2}$.

4.2 State Estimation

Figure 4 shows the estimate of T when the "real" system equals the `org` model.

Figure 5 shows the resulting estimate of T when the "real" system equals the `is` model.

Figures 6 and 7 show the similar estimates of c_A, respectively.

4.3 Proportional + Integral Control

Figure 8 shows the use of PI controller to keep reactor temperature T close to a reference T_ref. The PI controller tuned for the `org` model, is also applied to the `is` model.

The result indicates that the controller easily handles the model difference between the two models. Figure 9 shows the applied control input T_c as well as the integral state $\bar{T}_\mathrm{c}$ in the controller for the two model cases.

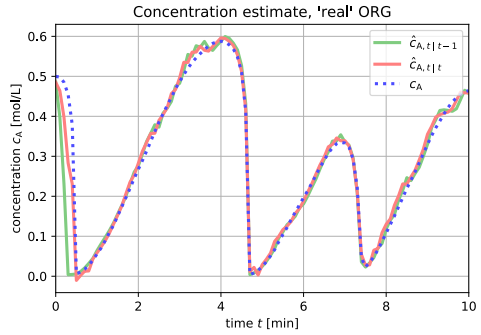

Figure 6. Real temperature c_A, a priori estimate $\hat{c}_{\mathrm{A},t|t-1}$, and *a posteriori* estimate $\hat{c}_{\mathrm{A},t|t}$ when estimator model is identical to the real system — except initial states.

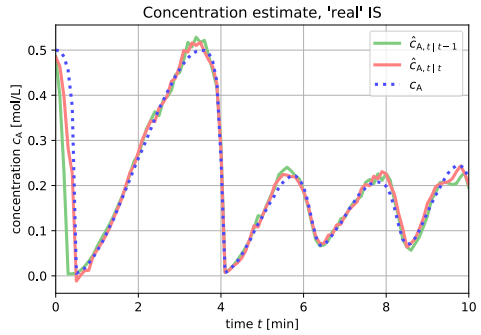

Figure 7. Real temperature c_A, a priori estimate $\hat{c}_{\mathrm{A},t|t-1}$, and *a posteriori* estimate $\hat{c}_{\mathrm{A},t|t}$ when estimator model differs from the real system, including different initial states.

https://doi.org/10.3384/ecp18153185

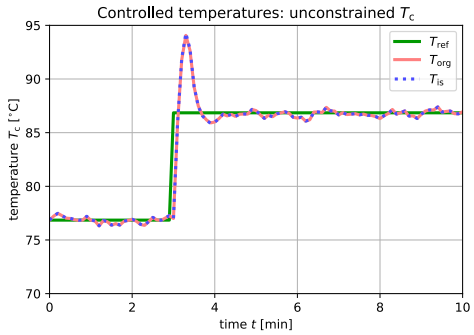

Figure 8. Output T as controlled with PI controller tuned for `org` model, and applied to `org` and `is` model.

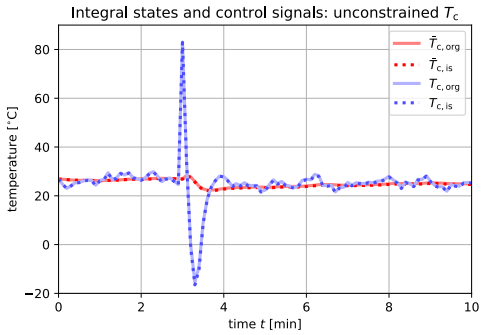

Figure 9. PI control signal T_c and integrator state $\bar{T}_c$ for `org` and `is` models.

Figure 9 clearly shows a problem for the controller: the cooling water can not take on negative temperatures T_c. We therefore add the constraint that $T_c \in [4, 96]$ °C, which together with anti-windup leads to the results in Figures 10 and 11 for output T and controller T_c, respectively.

4.4 Linear Quadratic Control

Figure 12 shows the use of LQG+I controller to keep reactor temperature T close to a reference T_{ref}.

Figure 13 shows the applied control input T_c.

Figures 12 and 13 should be compared with Figures 8

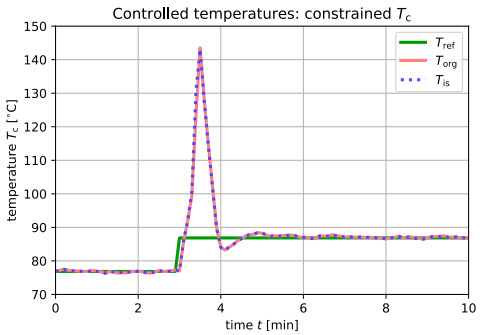

Figure 10. Output T as controlled with PI controller tuned for `org` model, and applied to `org` and `is` model: control input T_c is constrained to $[4, 96]$ °C and anti-windup is applied.

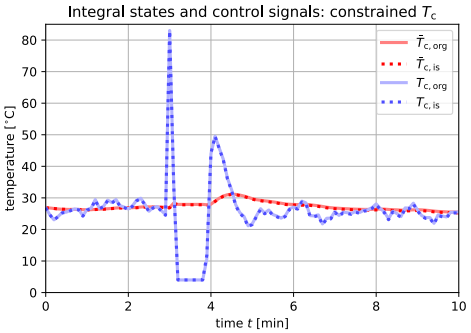

Figure 11. PI control signal T_c and integrator state $\bar{T}_c$ for `org` and `is` models: control input T_c is constrained to $[4, 96]$ °C and anti-windup is applied.

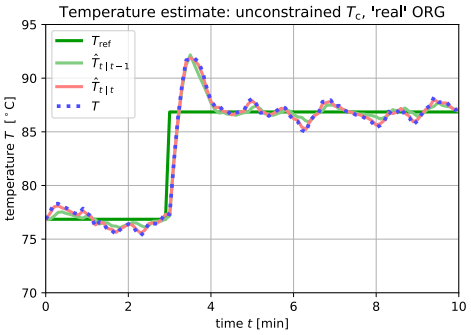

Figure 12. Output T as controlled with LQG controller tuned for `org` model, and applied to `org` model.

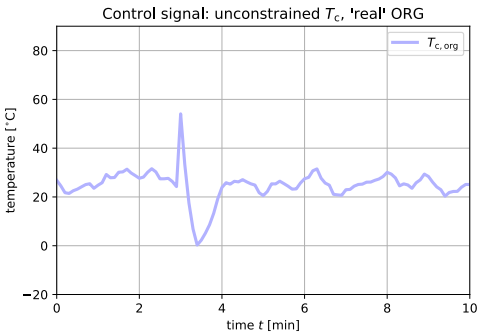

Figure 13. Control signal T_c from LQG controller.

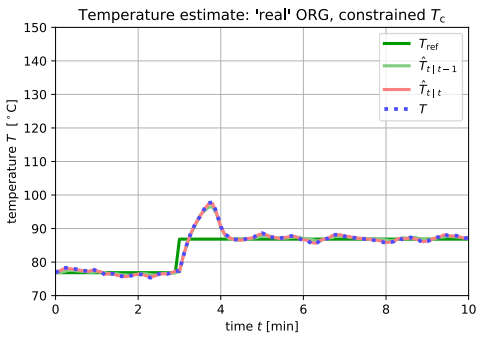

Figure 14. Output T as controlled with LQG controller tuned for `org` model, and applied to `org` model: control input T_c is constrained to $[4,96]$ °C and anti-windup is applied.

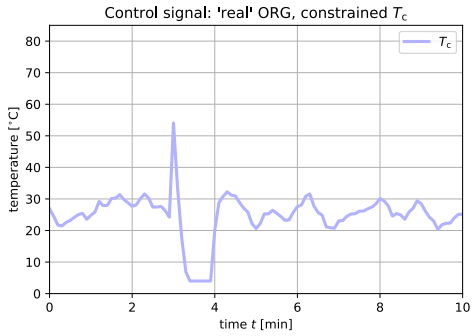

Figure 15. Control signal T_c from LQG controller: control input T_c is constrained to $[4,96]$ °C and anti-windup is applied.

and 9, respectively.

Adding the same constraint and anti-windup in the LQG+I controller leads to the results in Figures 14 and 15 for output T and controller T_c, respectively. These should be compared to Figures 10 and 11, respectively.

5 Discussion and Conclusions

Practical procedures for tuning a PI controller for an open loop unstable system is described, together with some continuous time implementation details with anti-windup action suitable for implementation in Modelica. A disadvantage of a continuous time implementation is that the chosen combination (Python + Modelica) does not allow for continuous time measurement noise. Details of tuning a constant gain Extended Kalman Filter is discussed, with an algorithm suitable for implementation in a combination of Python and Modelica. Finally, design of a linear LQG controller is described, and a controller is tuned. The method is extended to include integral action.

The developed control algorithms are applied to the nonlinear models, and tested through simulation. The EKF estimator works well in a highly non-linear regime, even for a more complex model. The developed PI controller is tuned; the example shows the need to take control input constraint into account. This leads to an unfortunate overshoot in the controlled variable (temperature). The LQG controller gives superior control with a considerable reduction in temperature overshoot, but gives slightly less disturbance attenuation; this problem would have been solved by setting KF parameter μ even smaller. To further reduce the temperature overshoot, utilizing the nonlinearity of the system in the controller design should be attempted.

The design of the LQG controller was carried out using the MATLAB Control Toolbox[7]. All other parts of the design and testing was carried out in a Jupyter Notebook[8] where Python code was combined with Modelica code.

The main message of this work is not so much the design of a controller, but rather to demonstrate how modern modeling and simulation tools such as OpenModelica in tandem with a script based language such as Python can ease and streamline control design, analysis, and testing through simulation.

A Model details

The ideal solution extension of the model in (Seborg et al., 2011) is presented with some details in (Sund et al., 2018): model parameters and nominal operating conditions are given, together with a manipulation of the model into an implicit ODE model of form $M \cdot \frac{dx}{dt} = f(x,u;\theta)$. The sequel discusses how this implicit ODE model can be simplified to the model in (Seborg et al., 2011).

When assuming that solvent S totally dominates in the mixture wrt. density, "mass matrix" M in Appendix A of (Sund et al., 2018) simplifies to the identity matrix, $M \to I$. Furthermore, the total heat capacities simplify to

$$C_p = m_S \hat{c}^{\bullet}_{p,S} = C_{p,i}. \tag{34}$$

With this assumption, information about species B becomes irrelevant — unless we are particularly interested in c_B. Thus, when c_B is of no particular interest, the model can be simplified to

$$\frac{dc_A}{dt} = \frac{\dot{V}_i}{V}(c_{A,i} - c_A) - a \cdot r \tag{35}$$

$$\frac{dT}{dt} = \frac{\dot{V}_i}{V}(T_i - T) + \frac{\left(-\Delta_r \tilde{H}\right) r}{\rho^{\bullet}_S \hat{c}^{\bullet}_{p,S}} + \frac{\dot{Q}}{\rho^{\bullet}_S \hat{c}^{\bullet}_{p,S} V}, \tag{36}$$

where r and $\dot{Q}$ are given by Eqs. 3 and 2, respectively, while $\Delta_r \tilde{H}$ is given by

$$\Delta_r \tilde{H} = \tilde{H}^{\circ}_B - a\tilde{H}^{\circ}_A + \left(\tilde{c}^{\bullet}_{p,B} - a\tilde{c}^{\bullet}_{p,A}\right)(T - T^{\circ}). \tag{37}$$

Consider now the Continuous Stirred Tank Reactor (CSTR) in Example 2.5 of (Seborg et al., 2011): this model is identical to the simplified model in Eqs. 35–36[9] *provided* that $a \equiv 1$ and $\Delta_r \tilde{H}$ is constant wrt. temperature. Assuming that the standard state specific enthalpies are

[7] www.mathworks.com

[8] www.jupyter.org

[9] The model notation is changed from that of (Seborg et al., 2011).

temperature independent, i.e., $\hat{H}_j^\circ$ is temperature independent, this requires that $\tilde{c}^\bullet_{p,\mathrm{B}} - a\tilde{c}^\bullet_{p,\mathrm{A}} \equiv 0$ — or $\tilde{c}^\bullet_{p,\mathrm{B}} \equiv \tilde{c}^\bullet_{p,\mathrm{A}}$. The parameters and operating conditions in Tables 3-4 of (Sund et al., 2018) are judiciously chosen to ensure that the solvent-dominating model is identical to the model in (Seborg et al., 2011).

B Model linearization

For the model, with $x = (T, c_A)$, $u = (T_\mathrm{c}, T_\mathrm{i}, \dot{V}_\mathrm{i}, c_{\mathrm{A,i}})$, and $y = T$, the linearized model is:

$$\frac{dx^\delta}{dt} = Ax^\delta + Bu^\delta \tag{38}$$

$$y^\delta = Cx^\delta + Du^\delta \tag{39}$$

where for any z, $z^\delta \triangleq z - z^*$ and asterisk $*$ indicates nominal value. Using the Python API described in (Lie et al., 2016) together with OpenModelica, let `sr_org` be a Python object of the original reactor in (Seborg et al., 2011) (but with n_A replacing c_A as state). We can then linearize the model in Python with the statement:

```
>>> A, B, C, D = sr_org.linearize()
```

The following system matrices are found:

$$A = \begin{pmatrix} 4.3796 & 209.205 \\ -0.035714 & -2 \end{pmatrix} \tag{40}$$

$$B = \begin{pmatrix} 2.09205 & 1 & 0 & 0 \\ 0 & 0 & 0.005 & 1 \end{pmatrix} \tag{41}$$

$$C = \begin{pmatrix} 1 & 0 \end{pmatrix} \tag{42}$$

$$D = \begin{pmatrix} 0 & 0 & 0 & 0 \end{pmatrix}. \tag{43}$$

The open loop eigenvalues of A are $\lambda = (2.83388381, -0.45432613)$, hence the system is open loop unstable. The open loop transfer function from T_c^δ to T^δ is

$$T^\delta(s) = 2.092\frac{s+2}{s^2 - 2.38s - 1.288} \cdot T_\mathrm{c}^\delta(s) \tag{44}$$

which implies that the system has a "stable" zero at $s = -0.5$ at the nominal operating point.

C Extended LQ+I system

The extended LQ problem with integral action is given as follows. Thee extended system is[10]

$$\frac{d}{dt}\underbrace{\begin{pmatrix} x \\ z \end{pmatrix}}_{\tilde{x}} = \underbrace{\begin{pmatrix} A & 0 \\ C & 0 \end{pmatrix}}_{\tilde{A}}\tilde{x} + \underbrace{\begin{pmatrix} B \\ 0 \end{pmatrix}}_{\tilde{B}}u \tag{45}$$

while the extended cost function $\tilde{\mathscr{J}}$ given as

$$\tilde{\mathscr{J}} = \frac{1}{2}\int_0^\infty \left(e_{\tilde{x}}^T \tilde{Q} e_{\tilde{x}} + e_u^T R e_u\right) dt \tag{46}$$

where

$$\tilde{Q} = \begin{pmatrix} C^T Q_y C & 0 \\ 0 & Q_z \end{pmatrix}. \tag{47}$$

[10] T_ref acts as a disturbance

References

Brian D. O Anderson and John B. Moore. *Optimal Control: Linear Quadratic Methods*. Prentice-Hall, Inc., 1989.

Karl Johan Åstrøm and Richard M. Murray. *Feedback Systems. An Introduction for Scientists and Engineers*. Princeton University Press, Princeton, NJ, 2008. ISBN 978-0-691-13576-2.

Peter Fritzson. *Principles of Object-Oriented Modeling and Simulation with Modelica 3.3: A Cyber-Physical Approach*. Wiley-IEEE Press, Piscataway, NJ, second edition, 2015. ISBN 978-1-118-85912-4.

Michael A. Henson and Dale E. Seborg. *Nonlinear Process Control*. Prentice Hall, Upper Saddle River, New Jersey, 1997.

Huibert Kwakernaak and Raphael Sivan. *Linear Optimal Control Systems*. Wiley-Interscience, New York, 1972.

Bernt Lie, Sudeep Bajracharya, Alachew Mengist, Lena Buffoni, Arun Kumar, Martin Sjölund, Adeel Asghar, Adrian Pop, and Peter Fritzson. Api for accessing openmodelica models from python. In *Proceedings of EuroSim 2016*, Oulu, Finland, 2016, September 2016.

Jan M. Maciejowski. *Predictive Control with Constraints*. Prentice Hall, Harlow, England, 2002.

James B. Rawlings, David Q. Mayne, and Moritz M. Diehl. *Model Predictive Control: Theory, Computation, and Design*. Nob Hill Publishing, LLC, second edition, 2017.

Dale E. Seborg, Thomas F. Edgar, Duncan A. Mellichamp, and III Doyle, Frank J. *Process Dynamics and Control*. John Wiley & Sons, Hoboken, NJ, third edition edition, 2011. ISBN 978-0-470-12867-1. ISBN 978-0-470-12867-1.

Dan Simon. *Optimal State Estimation: Kalman, H Infinity, and Nonlinear Approaches*. Wiley-Interscience, Hoboken, New Jersey, 2006.

Sveinung M. Sund, Marianne Plouvier, and Bernt Lie. Comparison of simulation tools for dynamic models. In *Proceedings, SIMS 2018*, Oslo Metropolitan University, September 2018. SIMS, Linköping University Press.