STUDIO 4. THE TANK REACTOR: STABILITY.

In this exercise we first consider the problem how to choose the flux through the tank reactor and the temperature of the cooler, to obtain a stationary point at given (wanted!) values of the concentration and the temperature in the tank reactor. From numerical simulation we then observe that this operating point is not stable. To be able to "adjust" the tank reactor so that it will operate in a stable way at our desired stationary point, we need to understand the reason for the instability. We therefore perform a linear stability analysis based on eigenvalues.

Important: write a readable report of your work in each studio session. You will need this at the end of the course when you prepare for the written exam; several exam questions will be based directly on the studio work which is only documented in your own notes and the instructions.

Some exercises are called homework in order to save time in the studio classroom.

Theory: S. Larsson, "Dynamiskt system. Stabilitet." (K)

1. Introduction

(K 1.1) Recall from *Studio 2. The Tank Reactor: Heat Balance* the system of differential equations,

$$
\begin{aligned}
&\frac{dX_1}{ds} = U_1(1-X_1) - X_1 f(X_2) = F_1(X,U),\ s>0,\\
&\frac{dX_2}{ds} = U_1(1-X_2) + \alpha X_1 f(X_2) - \beta(X_2-U_2) = F_2(X,U),\ s>0,\\
&X_1(0) = X_{1,0},\ X_2(0) = X_{2,0},
\end{aligned}
\tag{1}
$$

which is our mathematical model for the dynamics of the tank reactor. Recall, also, the *state variables* $X_1 = X_1(s)$ (dimensionless concentration) and $X_2 = X_2(s)$ (dimensionless reactor temperature), depending on s (dimensionless time). These two variables, that describe the *state* of the tank reactor, are the ones that we compute by solving (1), i.e., X_1 and X_2 are *output data*.

There are two kinds of *input data*. First we have the *initial data*, $X_0 = \begin{bmatrix} X_{1,0} \\ X_{2,0} \end{bmatrix}$, then the *control variables* $U_1 = U_1(s)$ (dimensionless flux) and $U_2 = U_2(s)$ (dimensionless cooler temperature).

Recall, finally, that $k\tau = f(X_2) = \delta e^{\gamma(1-1/X_2)}$ is the dimensionless rate coefficient given by the Arrhenius law. In *Studio 3. The Tank Reactor: Arrhenius' law* you determined the dimensionless numbers γ and δ, which can be thought of as non-dimensional counterparts to the activation energy and the rate constant of the reaction, by fitting the rate law to data. As for the other two dimensionless numbers appearing in (1), α is also reaction dependent since it is proportional to the heat of reaction, whereas β is proportional to the area and the heat transfer coefficient of the cooler.

Our final goal is to *design* the tank reactor in such a way that it runs in a stable manner at a specified, desired, operating point $\bar{X} = \begin{bmatrix} \bar{X}_1 \\ \bar{X}_2 \end{bmatrix}$. In order to achieve this goal, we first determine corresponding values of the control variables $\bar{U} = \begin{bmatrix} \bar{U}_1 \\ \bar{U}_2 \end{bmatrix}$, for which $\bar{X}$ *is* a stationary point. Then, we analyze the stability of $\bar{X}$ with respect to perturbations of input data and, if necessary, change the value of one or more *design parameters*. This could, for instance, physically mean varying the area of the cooler, i.e., varying the value of β in our mathematical model (1).

Date: April 29, 2002, Niklas Ericsson, Stig Larsson, Mathematics, Chalmers University of Technology.

2. Stationary points

(K 1.2) Our first task is, *given* a desired operating (stationary) point $\bar{X} = \begin{bmatrix}\bar{X}_1\\ \bar{X}_2\end{bmatrix}$, to *determine* corresponding values of the control variables $\bar{U} = \begin{bmatrix}\bar{U}_1\\ \bar{U}_2\end{bmatrix}$ by solving the system of equations,

$$\begin{aligned} 0 &= \bar{U}_1(1-\bar{X}_1) - \bar{X}_1 f(\bar{X}_2), \\ 0 &= \bar{U}_1(1-\bar{X}_2) + \alpha \bar{X}_1 f(\bar{X}_2) - \beta(\bar{X}_2 - \bar{U}_2). \end{aligned} \tag{2}$$

Exercise 1. Show that the solution to (2) is given by,

$$\begin{aligned} \bar{U}_1 &= \frac{\bar{X}_1}{1-\bar{X}_1} f(\bar{X}_2), \\ \bar{U}_2 &= \bar{X}_2 - \frac{1}{\beta}\Big(\frac{\bar{X}_1}{1-\bar{X}_1}(1-\bar{X}_2) f(\bar{X}_2) + \alpha \bar{X}_1 f(\bar{X}_2)\Big). \end{aligned} \tag{3}$$

□

We here stress a subtle point. *Given* $\bar{X}$, clearly, $\bar{U}$ is *uniquely* defined by (3). On the other hand, if we instead consider $\bar{U}$ as fixed, we know that $\bar{X}$ is *one* solution to (2), however, it might not be the *only* solution, i.e., there may exist more than one stationary point corresponding to $\bar{U}$. We will return to this later.

To concretize, let us now specify the values of the state variables at the desired operating point: our objective is to design the tank reactor to operate in a stable manner at $(c_f - \bar{c})/c_f = 0.5$ ("50% omsättningsgrad") and at reactor temperature $\bar{T} = 99°C$, i.e., at $\bar{X} = \begin{bmatrix}\bar{X}_1\\ \bar{X}_2\end{bmatrix} = \begin{bmatrix}0.5\\ (99+273.15)/T_f\end{bmatrix}$.

Exercise 2. Modify the file `data.m` from *Studio 2. The Tank Reactor: Heat Balance*, so that, given $\bar{X}$ (as above), $\bar{U}$ is computed from (3). At the same time, check that you have changed the old values of γ ($= 30$) and δ ($= 0.1$) that we used in *Studio 2* to the new ones that you determined in *Studio 3. The Tank Reactor: Arrhenius' law.* Also, make sure that you have set the area of the cooler $A_K = 1$ m^2. Hint: You need to change the line

```
Ubar = [1; 0.97];
```

into

```
Xbar = [0.5; (99 + 273.15)/Tf]; % Tf = 70 + 273.15
Ubar = zeros(2,1); % initialize (column vector) Ubar
Ubar(1) = ...; % insert the expression for Ubar(1)
Ubar(2) = ...; % insert the expression for Ubar(2)
```

□

Exercise 3. Check `data.m` by calling `tank2.m` that you wrote in *Studio 2. The Tank Reactor: Heat Balance*:

```
>> data
>> Xprime = tank2(0, Xbar)
```

What should the result be? (Note that the value of the first argument may be given arbitrarily, since there is no explicit time dependence in the right-hand side of (1).) □

3. Instability of the operating point

We now perform a first stability check of the operating point $\bar{X}$. We do this by introducing small initial perturbations, i.e., small initial deviations from $\bar{X}$, in X. In this test, we do not consider perturbations in the control variables, i.e., we set $U = \bar{U}$ in `tank2.m`.

Exercise 4. Assuming that `solve2.m` is the name of your script file from *Studio 2. The Tank Reactor: Heat Balance*, from which the call to `ode45` is made and the solution is plotted, solve (1) by giving the following Matlab commands:

```
>> data
>> S = 20;
>> X0 = Xbar + [0; 0.05];
>> solve2
>> X0 = Xbar - [0; 0.05];
>> solve2
```

Is $\bar{X}$ stable with respect to these perturbations? Also try some other initial perturbations. □

As you have just seen, a small deviation from $X = \bar{X}$ causes the tank reactor to depart from the desired operating point. Since these kinds of perturbations are inevitable in practice, the reactor will not remain in the desired state, which is therefore *not* stable. Rather, it will (depending on the initial perturbation) reach one of two *other* equilibrium points, which seem to be stable ones. These two are also stationary points, corresponding to $\bar{U}$, i.e., they are also solutions to (2). This is the non-uniqueness mentioned in Section 2.

4. Linear stability analysis

(K 1.3) In order to learn how to "adjust" the tank reactor so that it will operate in a stable way at $\bar{X}$, we need to systematically study the stability of solutions to (1). We will perform a *linear stability analysis* based on the assumption of *small perturbations*.

Let $X(s)$ with input data X_0, $U(s)$ be a solution to (1) that is close to $\bar{X}$. With

$$X(s) = \bar{X} + \Delta X(s), \quad X_0 = \bar{X} + \Delta X_0, \quad U(s) = \bar{U} + \Delta U(s), \tag{4}$$

we may consider $\Delta X(s)$ as a *perturbation* in $X(s)$ caused by the perturbations ΔX_0 and $\Delta U(s)$ in input data.

If $\Delta X(s)$ and $\Delta U(s)$ are *small*, we obtain the *linear system*

$$\begin{aligned} x'(s) &= Ax(s) + Bu(s), \quad s > 0, \\ x(0) &= x_0, \end{aligned} \tag{5}$$

for the *approximate perturbation* $x(s) \approx \Delta X(s)$ caused by the perturbations in input data $x_0 = \Delta X_0$ and $u(s) = \Delta U(s)$. In (5),

$$A = \begin{bmatrix} \dfrac{\partial F_1}{\partial X_1}(\bar{X}, \bar{U}) & \dfrac{\partial F_1}{\partial X_2}(\bar{X}, \bar{U}) \\[2ex] \dfrac{\partial F_2}{\partial X_1}(\bar{X}, \bar{U}) & \dfrac{\partial F_2}{\partial X_2}(\bar{X}, \bar{U}) \end{bmatrix} = \begin{bmatrix} -\bar{U}_1 - f(\bar{X}_2) & -\bar{X}_1 f'(\bar{X}_2) \\[2ex] \alpha f(\bar{X}_2) & -\bar{U}_1 + \alpha \bar{X}_1 f'(\bar{X}_2) - \beta \end{bmatrix}, \tag{6}$$

where $f'(\bar{X}_2) = \dfrac{\gamma}{\bar{X}_2^2} f(\bar{X}_2)$, and

$$B = \begin{bmatrix} \dfrac{\partial F_1}{\partial U_1}(\bar{X}, \bar{U}) & \dfrac{\partial F_1}{\partial U_2}(\bar{X}, \bar{U}) \\[2ex] \dfrac{\partial F_2}{\partial U_1}(\bar{X}, \bar{U}) & \dfrac{\partial F_2}{\partial U_2}(\bar{X}, \bar{U}) \end{bmatrix} = \begin{bmatrix} 1 - \bar{X}_1 & 0 \\[2ex] 1 - \bar{X}_2 & \beta \end{bmatrix}, \tag{7}$$

are called *Jacobi matrices* of $F(X, U) = \begin{bmatrix} F_1(X, U) \\ F_2(X, U) \end{bmatrix}$ at $\bar{X}$, $\bar{U}$.

Homework 1. Verify (6) and (7). □

5. Stability with respect to perturbations of initial data

(K 1.4) In this section we consider the case $u(s) = 0$, i.e., we only consider perturbations in initial data. In this case, (5) simplifies to

$$\begin{aligned} x'(s) &= Ax(s), \quad s > 0, \\ x(0) &= x_0, \end{aligned} \tag{8}$$

with solution (we assume that A is diagonalizable)

$$x(s) = c_1 e^{\lambda_1 s} g_1 + c_2 e^{\lambda_2 s} g_2, \tag{9}$$

where λ_i, g_i are eigenvalues and eigenvectors of A, and the c_i are constants depending on x_0. Clearly, the growth of $x(s)$ (and accordingly the stability of $\bar{X}$) depends on the eigenvalues of A.

Exercise 5. Compute the eigenvalues of A using Matlab. Hint: First write the function file `jacobianA.m` that computes A:

```
function A = jacobianA(Xbar)
global alpha beta gamma delta Ubar
A = zeros(2,2); % initialize (2x2 matrix) A
A(1,1) = ...; % insert the expression for A(1,1)
A(1,2) = ...; % insert the expression for A(1,2)
A(2,1) = ...; % insert the expression for A(2,1)
A(2,2) = ...; % insert the expression for A(2,2)
```

Then you can compute the eigenvalues of A by typing:

```
>> data
>> global A % we declare A as global because we will need this later
>> A = jacobianA(Xbar)
>> eig(A)
```

□

As you (hopefully!) noticed, A has two real eigenvalues, one positive and one negative. Because of the positive eigenvalue one of the terms in (9) will grow exponentially with time, and this explains the instability of $\bar{X}$.

Exercise 6. Solve (8) with the same initial perturbations as in Exercise 4. Hint: First write the function file `lineartank.m` that computes the right-hand side of (8):

```
function y = lineartank(s,x)
global A % this is the reason we declared A as global
y = A*x;
```

Then modify `solve2.m` into `linearsolve.m`. (Just replace `tank2` by `lineartank` in the call to `ode45`, and `X` by `x` everywhere.) Now you can solve (8) by typing:

```
>> figure % opens a new figure
>> data
>> S = 1;
>> x0 = [0; 0.05];
>> linearsolve
>> x0 = [0; -0.05];
>> linearsolve
```

□

One clearly sees the perturbation growth. Note that (8) was derived on the basis of the assumption of *small* perturbations and that it is not valid if $x(s)$ becomes "too" large. So there is no point in computing much further than to $S = 1$.

It is instructive to compare $X(s)$, computed as in Exercise 4, to $\bar{X} + x(s)$, with $x(s)$ computed as in Exercise 6:

Exercise 7. First solve (1), as in Exercise 4, with `S = 1` and `X0 = Xbar + [0; 0.05]`. Then solve (8), as in Exercise 6, with `S = 1` and `x0 = [0; 0.05]`. Now compare the first solution, `X`, to `Xbar + x`, where `x` is the second solution. You can do the comparison by writing and running the following script:

```
clf % clear current figure
plot(X(:,1), X(:,2)) % plots second versus first component of X
hold on
```

```
plot(Xbar(1) + x(:,1), Xbar(2) + x(:,2), '--')
title('Phase portraits:   Solid: X   Dashed: Xbar + x')
xlabel('X_1, Xbar_1 + x_1')
ylabel('X_2, Xbar_2 + x_2')
hold off
```

□

Note how the two curves successively diverge, and how the linear approximation fails to find the stable equilibrium point.

Next studio session, *Studio 5. The Tank Reactor: Design for Stability*, we will conclude the exercise on the tank reactor by "adjusting" it in such a way that the operating point $\bar{X}$ becomes stable. The idea is to try to "move" the eigenvalues of A so that their real parts get the right (negative!) sign. We will also briefly consider stability with respect to perturbations in the control variables.