Homework 1 - Equilibria, Eigen Analysis, Simulation

Jose Eduardo Laruta Espejo
Facultad de Ingeniería - Universidad Mayor de San Andrés

May 20, 2019

1 Finding Equilibria

1.1 Simulating the model

Our plant model is given by the following ODE system:

$$\dot{x_1} = \frac{u_1}{V}(u_3 - x_1) - R \tag{1}$$
$$\dot{x_2} = \frac{u_1}{V}(u_2 - x_2) + \frac{\gamma}{\sigma}R \tag{2}$$

where $R = \alpha x_1 e^{-\frac{\beta}{x_2+46}}$, and the constant are: $\alpha = 3.0 \times 10^{11} min^{-1}, \beta = 2.8 \times 10^3 Rankin/10, \gamma = 8BTU/klb - mole, \sigma = 5BTU/(gal - degF), V = 1000gallons/10$ and constant inputs: $u_1 = 50.5, u_2 = 33.794, u_3 = 19.5037$.

The resulting code for simulating the model is as follows:

```
function xdot = exo(t, x)
% Constants and known values
alpha = 3.0e+11;
beta = 2.8e+03;
gamma = 9;
sigma = 5;
V = 1000;
% constant inputs
u1 = 50.5;
u2 = 33.794;
```

```
u3 = 19.5037;
% non linear term
R = alpha * x(1) * exp(-beta/(x(2) + 46));
% space state ODEs
xdot(1) = (u1 / V) * (u3 - x(1)) - R;
xdot(2) = (u1 / V) * (u2 - x(2)) + (gamma / sigma) * R;
xdot = xdot(:); % column vector
```

Now, we can find the two stable equilibria by simulation, with the following initial condition and function calls:

```
tspan = [0 250]; %reasonable time for steady state
% from slide 37 we choose some initial conditions
xeq1 = [5; 67];      % close to one equilibrium
xeq2 = [18; 30];     % close to another
% first equilibrium point
[t1, x1] = ode45('exo', tspan, xeq1);
% steady state value
xsim_eq1 = x1(length(t1), :)
% second equilibrium point
[t2, x2] = ode45('exo', tspan, xeq2);
% steady state value
xsim_eq2 = x2(length(t2), :)
```

the results of simulation:

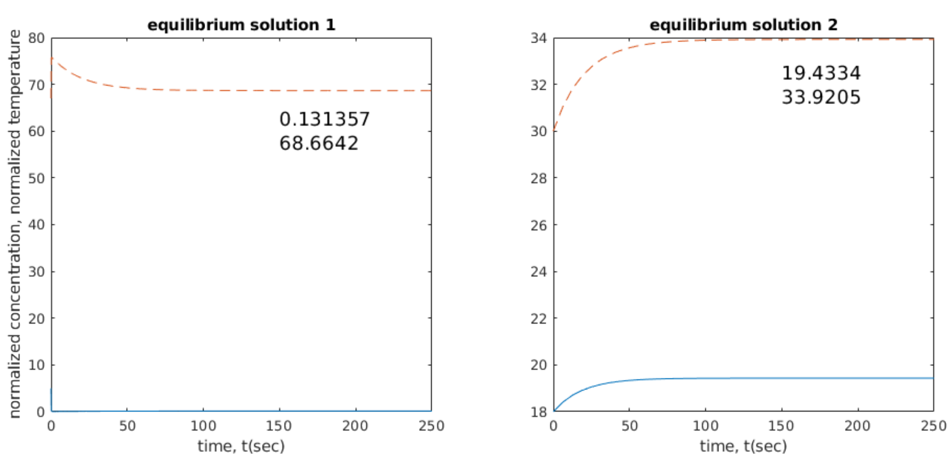

Figure 1: Simulation of the model

The equilibria are: [**0.131357, 68.6642**] and [**19.4334, 33.9205**].

1.2 fminsearch function

After having found two equilibria by simulation, we can use a "shell" function in order to use *fminsearch* function:

```
% Shell function for fminsearch
function objective = mag_sq_xdot(x)
xdot = exo(0, x);
objective = xdot' * xdot;
```

```
% equilibrium by fminsearch
% from slide 37 we choose some initial conditions
xeq1 = [1 70];
xmin_eq1 = fminsearch('mag_sq_xdot' , xeq1)

xeq2 = [20 30];
xmin_eq2 = fminsearch('mag_sq_xdot' , xeq2)

xeq3 = [10 49];
xmin_eq3 = fminsearch('mag_sq_xdot' , xeq3)
```

with results:

$$x_{eq1} = \begin{bmatrix} 0.1314 \\ 68.6642 \end{bmatrix} \tag{3}$$

$$x_{eq2} = \begin{bmatrix} 19.4334 \\ 33.9206 \end{bmatrix} \tag{4}$$

$$x_{eq3} = \begin{bmatrix} 11.6442 \\ 47.9411 \end{bmatrix} \tag{5}$$

As we can see, the first two equilibrium points are very similar to the ones we found through simulation. However, there is a third equilibrium that we need to analyze because it corresponds to the unstable equilibrium.

2 Small Signal Linearization

In order to linearize our model around our operating points we use the following:

$$A = \left.\frac{\partial f}{\partial x}\right|_{x_0,u_0} \tag{6}$$

$$A = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} \end{bmatrix}\Bigg|_{x_0,u_0} \tag{7}$$

we can define our f_i functions based on our model defined in equations (1) and (2):

$$f_1 = \frac{u_1 u_3}{V} - \frac{u_1 x_1}{V} - \alpha x_1 e^{-\frac{\beta}{x_2+46}} \tag{8}$$

$$f_2 = \frac{u_1 u_2}{V} - \frac{u_1 x_2}{V} - \frac{\alpha \gamma x_1}{\sigma} e^{-\frac{\beta}{x_2+46}} \tag{9}$$

then, compute the partial derivatives:

$$\frac{\partial f_1}{\partial x_1} = -\frac{u_1}{V} - \alpha e^{-\frac{\beta}{x_2+46}} \tag{10}$$

$$\frac{\partial f_1}{\partial x_2} = -\frac{\alpha \beta x_1}{(x_2+46)^2} e^{-\frac{\beta}{x_2+46}} \tag{11}$$

$$\frac{\partial f_2}{\partial x_1} = \frac{\alpha \gamma}{\sigma} e^{-\frac{\beta}{x_2+46}} \tag{12}$$

$$\frac{\partial f_2}{\partial x_2} = -\frac{u_1}{V} + \frac{\alpha \beta \gamma x_1}{\sigma (x_2+46)^2} e^{-\frac{\beta}{x_2+46}} \tag{13}$$

and write a matlab function for the linearized model:

```
% small signal linearization of the system
function A = linear_exo(x)
% Constants and known values
alpha = 3.0e+11;
beta = 2.8e+03;
gamma = 9;
sigma = 5;
V = 1000;
% constant inputs
u1 = 50.5;
u2 = 33.794;
u3 = 19.5037;
% small signal linearization for operating point x
f11 = -(u1/V) - alpha * exp((-beta)/(x(2) + 46));
f12 = -(alpha * beta * x(1) * exp((-beta)/(x(2) + 46))) / ((x
    (2) + 46)^2);
f21 = (gamma * alpha * exp((-beta)/(x(2) + 46))) / sigma;
```

```
f22 = -(u1/V) + (alpha * beta * gamma * x(1) * exp((-beta)/(x
    (2) + 46))) / (sigma * (x(2) + 46)^2);
% A matrix for our system
A = [f11 f12; f21 f22];
```

2.1 Eigen Analysis

Once we have our linearized model around the equilibria, it is easy to find its eigenvalues and eigenvectors to analyze the behaviour of the system using the matlab function *eig*:

```
% Eigenvalues and eigenvectors analysis
% first equilibrium
A1 = linear_exo(xmin_eq1)
[v1 d1] = eig(A1)
% second equilibrium
A2 = linear_exo(xmin_eq2)
[v2 d2] = eig(A2)
% third equilibrium
A3 = linear_exo(xmin_eq3)
[v3 d3] = eig(A3)
```

2.1.1 First Equilibrium

In this case for the equilibrium at [**0.1314, 68.6642**], our A matrix, eigenvalues and eigenvectors are:

$$A = \begin{bmatrix} -7.4983 & -0.2083 \\ 13.4060 & 0.3245 \end{bmatrix} \tag{14}$$

$$\lambda_1 = -7.1233 \qquad v_1 = \begin{bmatrix} -0.4856 \\ 0.8742 \end{bmatrix} \tag{15}$$

$$\lambda_2 = -0.0505 \qquad v_1 = \begin{bmatrix} 0.0280 \\ -0.9996 \end{bmatrix} \tag{16}$$

from this results the following conclusions can be made:

- both eigenvalues, $\lambda_1 = -7.1233$ and $\lambda_2 = -0.0505$ have **negative real part** which means that both are stable, hence, this equilibrium is **stable**.

- λ_1 is significantly bigger than λ_2, therefore we can say that this eigenvalue is the fast one and λ_2 the slow one.
- the trajectories near the first equilibrium follow the directions given by the eigenvectors v_1 and v_2. The state will follow v_1 faster than v_2 since λ_1 is the fastest eigenvalue.

2.1.2 Second Equilibrium

In this case for the equilibrium at [**19.4334, 33.9206**], our A matrix, eigenvalues and eigenvectors are:

$$A = \begin{bmatrix} -0.0507 & -0.0016 \\ 0.0003 & -0.0477 \end{bmatrix} \tag{17}$$

$$\lambda_1 = -0.0505 \qquad v_1 = \begin{bmatrix} -0.9932 \\ 0.1166 \end{bmatrix} \tag{18}$$

$$\lambda_2 = -0.0479 \qquad v_1 = \begin{bmatrix} 0.4856 \\ -0.8742 \end{bmatrix} \tag{19}$$

from this results the following conclusions can be made:

- both eigenvalues, $\lambda_1 = -0.0505$ and $\lambda_2 = -0.0479$ have **negative real part** which means that both are stable, hence, this equilibrium is **stable**.
- λ_1 and λ_2 are very close in magnitude, therefore both eigenvalues are very similar in behaviour.
- the trajectories near the equilibrium follow the directions given by the eigenvectors v_1 and v_2. The state will follow v_1 and v_2 at very similar rate so there is no dominant direction.

2.1.3 Third Equilibrium

For the last equilibrium found at [**11.6442, 47.9411**], our A matrix, eigenvalues and eigenvectors are:

$$A = \begin{bmatrix} -0.0846 & -0.1259 \\ 0.0614 & 0.1762 \end{bmatrix} \tag{20}$$

$$\lambda_1 = -0.0505 \qquad v_1 = \begin{bmatrix} -0.9653 \\ 0.2613 \end{bmatrix} \tag{21}$$

$$\lambda_2 = 0.1421 \qquad v_1 = \begin{bmatrix} 0.4856 \\ -0.8742 \end{bmatrix} \tag{22}$$

from this results the following conclusions can be made:

- in this case λ_1 has negative real part, however, λ_2 has **positive real part**, therefore this equilibrium is **unstable**.
- due to the nature of this equilibrium, the trajectories nearby follow directions that get farther from it.

2.2 Visualization of eigenvectors

In order to get a better understanding of the nature of the equilibria of our system, we can plot the eigenvectos located at each operating point, the results are shown in Figure(2).

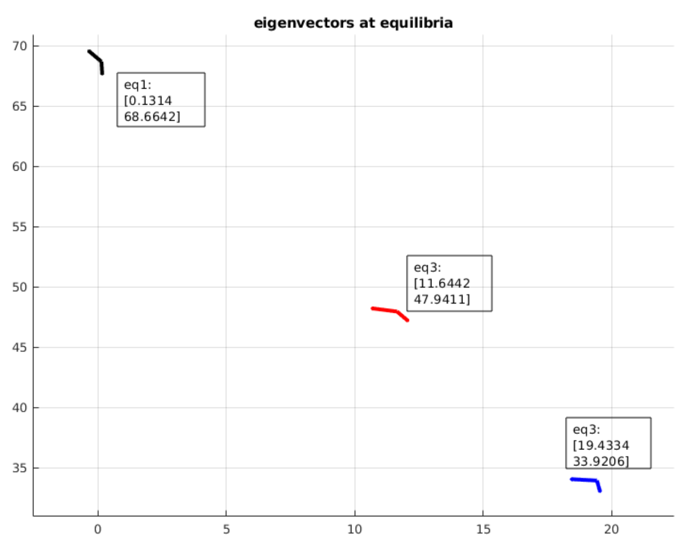

Figure 2: Graphic Visualization of the system eigenvectors at equilibria

2.3 Simulation of trajectories

Once we analyzed the equilibria, and the behaviour of the system near them through eigen analysis, we can run several simulations for visualizing the nature of the trajectories and validate the previous results. In this case, a little routine for run several simulation in a loop with random initial condition has been developed, the results are shown in Figure(3).

For this simulation we ran 200 trajectories with random initial conditions. It is clear that the trajectories behave as we expected when we analyzed equilibria and eigenvalues and eigenvectors of SSL system at equilibria.

Note: The code for both the simulations and this report can be found in this github repository.

```
% limits for initial conditions
tspan = [0 30];
x1_min = -10;
x1_max = 45;
x2_min = 20;
x2_max = 80;
t_span = [0 10];
% plot eigen vectors
hold on;
grid('on')
xlabel('x1');
ylabel('x2');
title('trajectories simulation')
xlim([-5 25])
ylim([30 75])
% random initial conditions
for i=1:200
    random_x = [((x1_max - x1_min).*rand(1,1) + x1_min); ((x2_max - x2_min).*rand(1,1) + x2_min)];
    % plot simulation
    [t, x] = ode45('exo', tspan, random_x);
    plot(x(:,1), x(:,2), 'g');
end
% draw eigenvectors at equilibria
quiver(xmin_eq1(1), xmin_eq1(2), v1(1,1),v1(2,1),6*d1(1,1), 'k', 'LineWidth',3);
quiver(xmin_eq1(1), xmin_eq1(2), v1(1,2),v1(2,2),6*d1(2,2), 'k', 'LineWidth',3);

quiver(xmin_eq2(1), xmin_eq2(2), v2(1,1),v2(2,1),6*d2(1,1), 'b', 'LineWidth',3);
```

```
quiver(xmin_eq2(1), xmin_eq2(2), v2(2,1),v2(2,2),6*d2(2,2), 'b', 'LineWidth',3);

quiver(xmin_eq3(1), xmin_eq3(2), v3(1,1),v3(2,1),6*d3(1,1), 'r', 'LineWidth',3);
quiver(xmin_eq3(1), xmin_eq3(2), v3(1,2),v3(2,2),6*d3(2,2), 'r', 'LineWidth',3);

hold off;
```

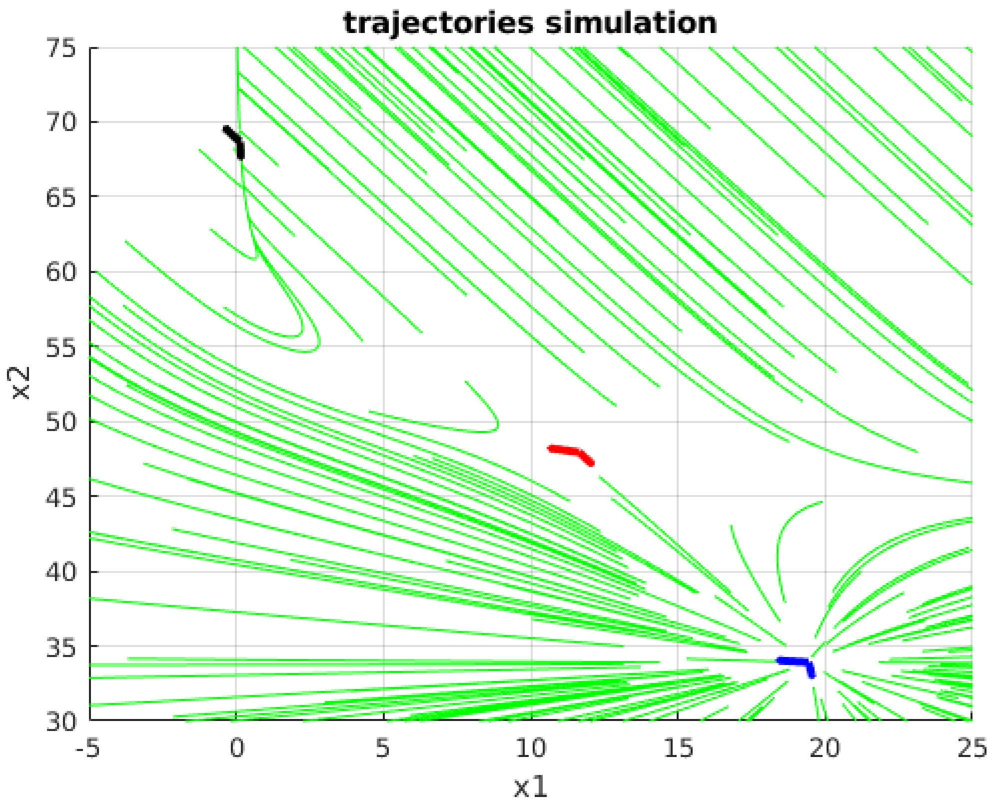

Figure 3: Graphic Visualization of the system trajectories