NTNU – Trondheim
Norwegian University of
Science and Technology

For questions during the exam:
Lars Imsland, tel. 47 23 19 49.

Exam in TTK4130 Modeling and Simulation

Wednesday, May 20 2015
09:00 – 13:00

Permitted aids (code A): All written and handwritten examination support materials are permitted.

Answers in English, Norwegian, or a mixture of the two accepted.

Grades available: As specified by regulations.

Problem 1 (12 %)

(6 %) (a) Find the stability function for the Runge-Kutta method given by the Butcher array below. The answer should contain calculations.

$$\begin{array}{c|ccc} 0 & & & \\ 1/3 & 1/3 & & \\ 2/3 & 0 & 2/3 & \\ \hline & 1/4 & 0 & 3/4 \end{array}$$

> **Solution:** This can be calculated either by applying the R-K method to the test function, or by using the formula for stability functions:
>
> $$R(s) = 1 + s + \frac{s^2}{2} + \frac{s^3}{6}.$$
>
> It is a 3rd order Runge-Kutta method (Heun's method).

(4 %) (b) A dynamic system has an oscillatory mode with eigenvalues $\lambda = \pm 1j$ (that is, eigenvalues on the imaginary axis). Is the method in (a), with steplength $h = 1$, stable for this mode? Calculations required for full score.

> **Solution:** The stability function for this eigenvalue (I check only $h\lambda = +j$, as $-j$ will be the same) evaluates:
>
> $$\begin{aligned} R(1j) &= 1 + j + \frac{j^2}{2} + \frac{j^3}{6} \\ &= 1 + j - \frac{1}{2} - j\frac{1}{6} \\ &= \frac{1}{2} + j\frac{5}{6} \end{aligned}$$
>
> from which we see that
>
> $$|R(1j)| = |\frac{1}{2} + j\frac{5}{6}| = \sqrt{\left(\frac{1}{2}\right)^2 + \left(\frac{5}{6}\right)^2} = \sqrt{\frac{34}{36}} < 1.$$
>
> That is, the method is stable for this mode.

(2 %) (c) Is the method A-stable? L-stable?

Solution: No explicit methods are A-stable, nor L-stable.

Problem 2 (28 %)

(8 %) (a) Consider the second-order system $\ddot{y} = -\omega_0^2 y$. For what combination of ω_0 and steplength h will the simulation be stable, if you use i) explicit Euler, ii) modified Euler, iii) Heun's method (ERK3, third order explicit Runge-Kutta) and iv) implicit Euler?

Figure 1 might be of use.

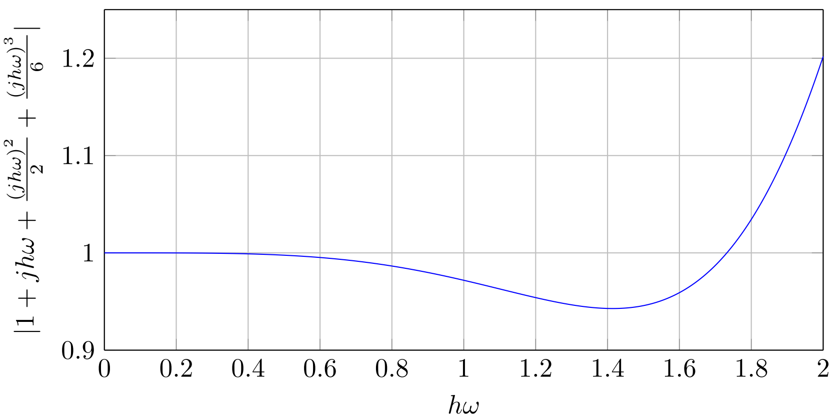

Figure 1: Helpful figure?

Solution: The system (the harmonic oscillator) has eigenvalues $\lambda_{1,2} = \pm j\omega_0$, that is, at the imaginary axis. Explicit and Modified Euler does not include part of the imaginary axis (except the origin) in their stability domain, therefore these methods will never be stable for this system, except when $\omega_0 = 0$.

(To show this, for Modified Euler, calculate $|R(j\omega_0)| = |1+jh\omega_0+{}^1\!/_2(jh\omega_0)^2| = \sqrt{1+{}^1\!/_4(h\omega_0)^4}$ which is stable (smaller than or equal to 1) only for $h\omega_0 = 0$. Calculations were not required for Explicit Euler and Modified Euler.)

However, Heun's method does include parts of the imaginary axis in the stability domain. From Figure 1 we see that the stability function for a 3rd order Runge-Kutta method is smaller than or equal to 1, and therefore stable, for $h\omega \leq 1.73$ (approximately).

Implicit Euler will be stable for all combinations of h and ω_0.

For some second-order systems (e.g. mechanical systems) a method called "semi-implicit Euler" is popular. This method is not mentioned in the book. For a second-order system on the form

$$\dot{y}_1 = y_2 \tag{1a}$$
$$\dot{y}_2 = f(y_1, y_2, t) \tag{1b}$$

this method can can be written

$$\begin{aligned} y_{1,n+1} &= y_{1,n} + hy_{2,n+1} \\ y_{2,n+1} &= y_{2,n} + hf(y_{1,n}, y_{2,n}, t_n) \end{aligned}$$

Like explicit and implicit Euler, one can show that this method is accurate of first order (you are not asked to show this).

(4 %) (b) Why do you think this method is called "semi-implicit"? How complex is it to implement this method (for systems like (1)), compared with explicit Euler and implicit Euler?

> **Solution:** Comparing with explicit and implicit Euler, we see that this method is a mix by utilizing the calculated $y_{2,n+1}$ to calculate $y_{1,n+1}$. Due to the structure of the system, this can be done without solving any implicit equation, by calculating the second line first. So the method is "halfway" implicit, but is as easy/complex as explicit Euler to implement, and therefore much less complex than implicit Euler (no nonlinear equations to solve).

(10 %) (c) If we apply this method to the system in (a), for what combination of h and ω_0 will it be stable? Hint: You cannot apply the normal test-system $\dot{y} = \lambda y$ to this method. Instead, use the system in (a) as test system and analyze eigenvalues of the resulting discrete-time system. The following might become useful:

$$\begin{pmatrix} 1 & -h \\ 0 & 1 \end{pmatrix}^{-1} = \begin{pmatrix} 1 & h \\ 0 & 1 \end{pmatrix}$$

$$\text{for } g_{1,2}(s) = 1 - \frac{s^2}{2} \pm \frac{\sqrt{s^2\left(s^2-4\right)}}{2} \text{ we have } \begin{cases} |g_{1,2}(s)| = 1 & |s| \leq 2 \\ |g_1(s)| > 1 \text{ or } |g_2(s)| > 1 & |s| > 2 \end{cases}$$

> **Solution:** The method applied to the system in (a) becomes
>
> $$\begin{aligned} y_{1,n+1} &= y_{1,n} + h y_{2,n+1} \\ y_{2,n+1} &= y_{2,n} + h\left(-\omega_0^2 y_{1,n}\right) \end{aligned}$$
>
> Gathering the variables, this can be written
>
> $$\begin{pmatrix} 1 & -h \\ 0 & 1 \end{pmatrix} \begin{pmatrix} y_{1,n+1} \\ y_{2,n+1} \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ -h\omega_0^2 & 1 \end{pmatrix} \begin{pmatrix} y_{1,n} \\ y_{2,n} \end{pmatrix}$$
>
> which can be solved as
>
> $$\begin{aligned} \begin{pmatrix} y_{1,n+1} \\ y_{2,n+1} \end{pmatrix} &= \begin{pmatrix} 1 & -h \\ 0 & 1 \end{pmatrix}^{-1} \begin{pmatrix} 1 & 0 \\ -h\omega_0^2 & 1 \end{pmatrix} \begin{pmatrix} y_{1,n} \\ y_{2,n} \end{pmatrix} \\ &= \begin{pmatrix} 1 & h \\ 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 \\ -h\omega_0^2 & 1 \end{pmatrix} \begin{pmatrix} y_{1,n} \\ y_{2,n} \end{pmatrix} \\ &= \begin{pmatrix} 1 - h\omega_0^2 & h \\ -h\omega_0^2 & 1 \end{pmatrix} \begin{pmatrix} y_{1,n} \\ y_{2,n} \end{pmatrix} \end{aligned}$$
>
> The "system matrix"
>
> $$A = \begin{pmatrix} 1 - h^2\omega_0^2 & h \\ -h\omega_0^2 & 1 \end{pmatrix}$$
>
> has eigenvalues
>
> $$\lambda_{1,2} = 1 - \frac{h^2\omega_0^2}{2} \pm \frac{\sqrt{h^2\omega_0^2\left(h^2\omega_0^2 - 4\right)}}{2}$$
>
> From the hints in the problem, we see that both eigenvalues are inside (or actually on) the unit circle, and therefore stable, for $-2 \leq h\omega_0 \leq 2$ ($h\omega_0 \leq 2$ gives also full score).

(6 %) (d) Based on the results in (c), why do you think this method is popular for these systems, compared to explicit Euler? (If you did not manage to solve (c), assume that there exist reasonable combinations of h and ω_0 that makes the method stable.)

Solution: The method is as easy as explicit Euler to implement, but in contrast to explicit Euler it can be stable for eigenvalues on the imaginary axis – it even has better stability margins than Heun's method (ERK3), which requires much more computation (but on the other hand, Heun's method will be far more accurate for a given steplength h).

Remark: This method is also sometimes called "symplectic Euler", and one can show that it (almost) conserves energy. For instance, we have seen in this problem that for the harmonic oscillator, the eigenvalues on the imaginary axis are mapped to discrete-time eigenvalues on the unit circle (when $h\omega_0 \leq 2$), such that the simulations will have "standing oscillations" just as the continuous-time system.

Problem 3 (40 %)

In this problem, we will study how to develop a model of the Quanser "3-DOF Helicopter", see Figure 2, using Lagrangian mechanics. This helicopter is popular as a lab-setup to illustrate advanced control concepts to engineering students. We will assume that the helicopter consists of three point masses, one at each rotor and one counterweight, and that the rods connecting these points are massless. The helicopter can move in three degrees of freedom about a base point; these degrees of freedom are usually parametrized with a travel angle, an elevation angle, and a pitch angle. These angles are illustrated in Figure 3, along with lengths. We disregard friction, and we ignore the angle of the counterweight we can see in Figure 2 (not present in Figure 3).

The two rotors (front rotor and back rotor) give a lift force of magnitude F_f and F_b, respectively. If we assume a coordinate system fixed in each rotor with z-axis pointing in the direction of these forces, these forces can be written

$$\mathbf{f}_f^f = \begin{pmatrix} 0 \\ 0 \\ F_f \end{pmatrix} \quad \text{and} \quad \mathbf{f}_b^b = \begin{pmatrix} 0 \\ 0 \\ F_b \end{pmatrix},$$

where superscript f and b denote the coordinate system fixed in the front and back rotor, respectively.

Figure 2: Quanser 3-DOF Helicopter

(4 %) (a) Why do we prefer Lagrangian mechanics over using the Newton-Euler equations of motion for this system? What should we choose as generalized coordinates?

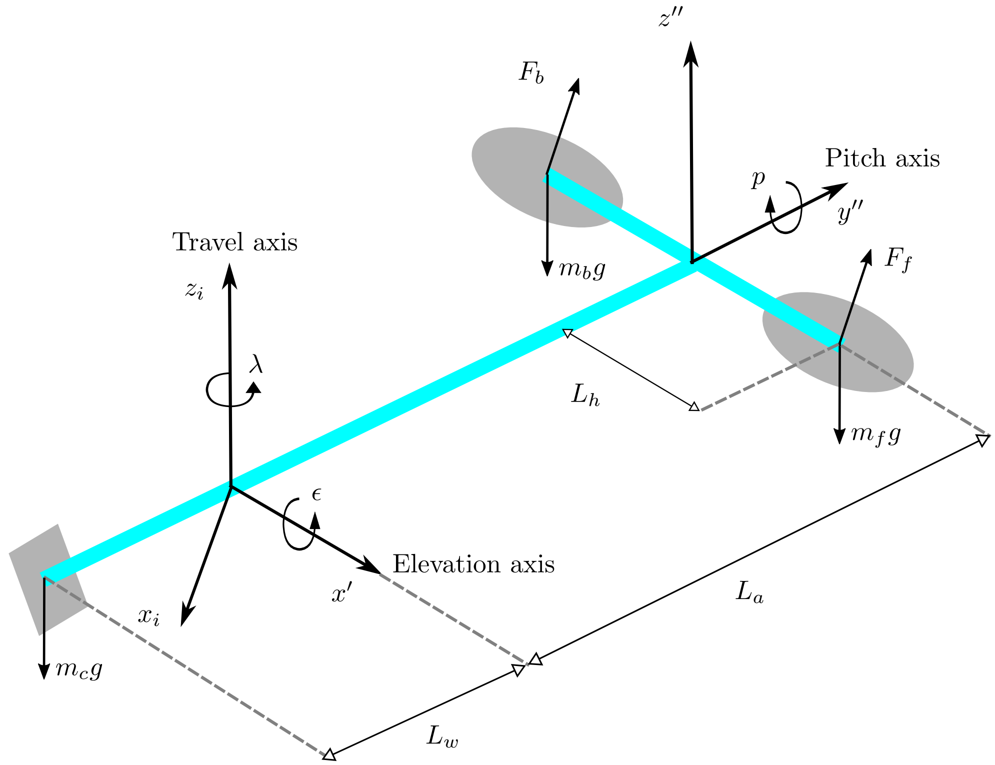

Figure 3: "Free body diagram" of 3-DOF helicopter. Note that angles are defined with positive direction according to right-hand rule about axis shown. The lift forces from the front and back rotors are denoted F_f and F_b, respectively.

> **Solution:** The motion is constrained, and can be parametrized by the three angles (travel, elevation, pitch), which are the natural generalized coordinates.

(10 %) (b) The positions of the front motor, back motor and counterweight in the inertial system, as functions of the generalized coordinates, are given by the following equations:

$$\text{Front motor :} \quad \mathbf{r}_f^i = \begin{pmatrix} x_f \\ y_f \\ z_f \end{pmatrix} = \begin{pmatrix} L_h \cos\lambda \cos p - L_h \sin\lambda \sin\epsilon \sin p - L_a \sin\lambda \cos\epsilon \\ L_h \sin\lambda \cos p + L_h \cos\lambda \sin\epsilon \sin p + L_a \cos\lambda \cos\epsilon \\ -L_h \cos\epsilon \sin p + L_a \sin\epsilon \end{pmatrix}$$

$$\text{Back motor :} \quad \mathbf{r}_b^i = \begin{pmatrix} x_b \\ y_b \\ z_b \end{pmatrix} = \begin{pmatrix} -L_h \cos\lambda \cos p + L_h \sin\lambda \sin\epsilon \sin p - L_a \sin\lambda \cos\epsilon \\ L_h \sin\lambda \cos p - L_h \cos\lambda \sin\epsilon \sin p + L_a \cos\lambda \cos\epsilon \\ L_h \cos\epsilon \sin p + L_a \sin\epsilon \end{pmatrix}$$

$$\text{Counterweight :} \quad \mathbf{r}_c^i = \begin{pmatrix} x_c \\ y_c \\ z_c \end{pmatrix} = \begin{pmatrix} L_w \sin\lambda \cos\epsilon \\ -L_w \cos\lambda \cos\epsilon \\ -L_w \sin\epsilon \end{pmatrix}$$

Explain how the equations for positions for the front rotor can be derived by defining five (or fewer) homogenous transformation matrices (for example rotation, rotation, translation, rotation, translation). You do not have to do the matrix multiplications for full score (if you want to check your answer, you may want to anyway), but you should write down each of the homogenous transformation matrices.

Solution: To come from the inertial system to a system fixed in the front rotor, we must first rotate an angle λ about the z_i-axis, then rotate an angle ϵ about the rotated x-axis, then do a translation L_a along the resulting y-axis, rotate an angle p around this y-axis, and do a translation L_h along the newly rotated x-axis. Writing up homogenous transformation matrices for these transformations, denoting each intermediate coordinate system with a, b, etc., the coordinates of the position of the origin of the coordinate system fixed in the front rotor (f) are the elements of the r-vector in the overall transformation matrix:

$$\mathbf{T}_f^i = \mathbf{T}_a^i \mathbf{T}_b^a \mathbf{T}_c^b \mathbf{T}_d^c \mathbf{T}_f^d$$

where

$$\mathbf{T}_a^i = \begin{pmatrix} \cos\lambda & -\sin\lambda & 0 & 0 \\ \sin\lambda & \cos\lambda & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}, \qquad \mathbf{T}_b^a = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\epsilon & -\sin\epsilon & 0 \\ 0 & \sin\epsilon & \cos\epsilon & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}, \qquad \mathbf{T}_c^b = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & L_a \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$\mathbf{T}_d^c = \begin{pmatrix} \cos p & 0 & \sin p & 0 \\ 0 & 1 & 0 & 0 \\ -\sin p & 0 & \cos p & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}, \qquad \mathbf{T}_f^d = \begin{pmatrix} 1 & 0 & 0 & L_h \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Multiplying these, we get a matrix on the form

$$\mathbf{T}_f^i = \begin{pmatrix} \mathbf{R}_f^i & \mathbf{r}_f^i \\ 0 & 1 \end{pmatrix}$$

where elements of $\mathbf{r}_f^i = (x_f, y_f, z_f)^\mathsf{T}$.

It is possible to use fewer transformation matrices.

Remark: It seems Quanser has defined the angles λ and p positive in the opposite direction compared to our solution (at least in the manual I have).

For the interested; Matlab code, using the symbolic toolbox, for doing the transformations (used to calculate the positions given in the problem):

```
clear all; close all
syms l e p Lh La Lw F_f F_b real

% Position/orientation of front motor
T1 = [cos(l) -sin(l) 0 0; sin(l) cos(l) 0 0; 0 0 1 0; 0 0 0 1];
T2 = [1 0 0 0; 0 cos(e) -sin(e) 0; 0 sin(e) cos(e) 0; 0 0 0 1];
T3 = [1 0 0 0; 0 1 0 La; 0 0 1 0; 0 0 0 1];
T4 = [cos(p) 0 sin(p) 0; 0 1 0 0; -sin(p) 0 cos(p) 0; 0 0 0 1];
T5 = [1 0 0 Lh; 0 1 0 0; 0 0 1 0; 0 0 0 1];
T = T1*T2*T3*T4*T5;
r_f = T(1:3,4)
F_fi = T(1:3,1:3)*[0;0;F_f]

% Position/orientation of back motor
T5 = [1 0 0 -Lh; 0 1 0 0; 0 0 1 0; 0 0 0 1];
T = T1*T2*T3*T4*T5;
r_b = T(1:3,4)
F_bi = T(1:3,1:3)*[0;0;F_b]
```

```
% Position/orientation of counterweight
T3 = [1 0 0 0; 0 1 0 -Lw; 0 0 1 0; 0 0 0 1];
T = T1*T2*T3;
r_c = T(1:3,4)

% generalized forces
tau_p = simplify(diff(r_f,p)'*F_fi + diff(r_b,p)'*F_bi)
tau_l = simplify(diff(r_f,l)'*F_fi + diff(r_b,l)'*F_bi)
tau_e = simplify(diff(r_f,e)'*F_fi + diff(r_b,e)'*F_bi)
```

(6 %) (c) Find an expression for the kinetic energy for the counterweight (mass m_c) as a function of the generalized coordinates (and their derivatives), and explain how you would find the total kinetic energy (you do not have to find the expression for the total kinetic energy, only explain the procedure). Assume the mass for front and back rotor is m_f and m_b, respectively.

Solution: The kinetic energy for the counterweight is

$$T_c = \frac{1}{2} m_c (\mathbf{v}_c^i)^\mathsf{T} (\mathbf{v}_c^i)$$

where

$$\mathbf{v}_c^i = \frac{\mathrm{d}}{\mathrm{d}t}\mathbf{r}_c^i = \begin{pmatrix} L_w \cos\lambda\cos\epsilon\dot{\lambda} - L_w \sin\lambda\sin\epsilon\dot{\epsilon} \\ L_w \sin\lambda\cos\epsilon\dot{\lambda} + L_w\cos\lambda\sin\epsilon\dot{\epsilon} \\ -L_w\cos\epsilon\dot{\epsilon} \end{pmatrix} = \begin{pmatrix} L_w\cos\lambda\cos\epsilon & -L_w\sin\lambda\sin\epsilon & 0 \\ L_w\sin\lambda\cos\epsilon & L_w\cos\lambda\sin\epsilon & 0 \\ 0 & -L_w\cos\epsilon & 0 \end{pmatrix} \begin{pmatrix} \dot{\lambda} \\ \dot{\epsilon} \\ \dot{p} \end{pmatrix}$$

Defining $\mathbf{q} = (\lambda, \epsilon, p)^\mathsf{T}$, this can be written

$$\mathbf{v}_c^i = \mathbf{J}_c(\mathbf{q})\dot{\mathbf{q}}$$

where

$$\mathbf{J}_c(\mathbf{q}) = \begin{pmatrix} L_w\cos\lambda\cos\epsilon & -L_w\sin\lambda\sin\epsilon & 0 \\ L_w\sin\lambda\cos\epsilon & L_w\cos\lambda\sin\epsilon & 0 \\ 0 & -L_w\cos\epsilon & 0 \end{pmatrix}$$

such that

$$T_c(\mathbf{q}, \dot{\mathbf{q}}) = \frac{1}{2} m_c \dot{\mathbf{q}}^\mathsf{T} \mathbf{J}_c(\mathbf{q})^\mathsf{T} \mathbf{J}_c(\mathbf{q})\dot{\mathbf{q}}$$

Defining/finding $\mathbf{v}_f^i = \mathbf{J}_f(\mathbf{q})\dot{\mathbf{q}}$ and $\mathbf{v}_b^i = \mathbf{J}_b(\mathbf{q})\dot{\mathbf{q}}$ in a similar manner, we can find the total kinetic energy

$$\begin{aligned} T(\mathbf{q}, \dot{\mathbf{q}}) &= T_c(\mathbf{q}, \dot{\mathbf{q}}) + T_f(\mathbf{q}, \dot{\mathbf{q}}) + T_b(\mathbf{q}, \dot{\mathbf{q}}) \\ &= \frac{1}{2}\dot{\mathbf{q}}^\mathsf{T} \left(m_c \mathbf{J}_c(\mathbf{q})^\mathsf{T}\mathbf{J}_c(\mathbf{q}) + m_f \mathbf{J}_f(\mathbf{q})^\mathsf{T}\mathbf{J}_f(\mathbf{q}) + m_b\mathbf{J}_b(\mathbf{q})^\mathsf{T}\mathbf{J}_b(\mathbf{q}) \right)\dot{\mathbf{q}} \end{aligned}$$

There are several ways to answer this question, all answers explaining how to find the correct kinetic energy give full score.

(4 %) (d) Find the potential energy for the system, as a function of the generalized coordinates.

Solution: The potential energy of each mass is given by the height (z-coordinate) in the inertial system, defining zero-level when this height is zero:

$$\begin{aligned} U(\mathbf{q}) &= m_f g z_f + m_b g z_b + m_c g z_c \\ &= -m_f g \left(L_h \cos\epsilon \sin p - L_a \sin\epsilon\right) + m_b g \left(L_h \cos\epsilon \sin p + L_a \sin\epsilon\right) - m_c g L_w \sin\epsilon \end{aligned}$$

(8 %) (e) The generalized forces for travel λ and elevation ϵ are

$$\begin{aligned} \tau_\lambda &= \left(F_b - F_f\right)\left(L_h \sin\epsilon - L_a \cos\epsilon \sin p\right) \\ \tau_\epsilon &= (F_b + F_f) L_a \cos p \end{aligned}$$

Set up equations on coordinate form, involving rotation matrices, to calculate these from $\mathbf{f}_f^f$ and $\mathbf{f}_b^b$. You do not have to (you should not) do the actual calculations/differentiations/matrix multiplications.

Solution: The generalized forces for particle systems, for generalized coordinate q_i, are given by (from eq. (8.7) in the book):

$$\tau_i = \sum_{k=1}^{N} \frac{\partial \vec{r}_k}{\partial q_i} \cdot \vec{F}_k$$

where $\vec{r}_k$ is the position of particle k, and $\vec{F}_k$ is the force on particle k. In this case, these are the rotor forces – since the gravity force is already taken care of in the potential energy, we do not include that in $\vec{F}_k$ here.

To calculate this product, we must choose a coordinate basis for the vectors. The most natural choice is the inertial system, I think, but other choices are possible. We already know the positions in the inertial system (see above). The forces in question are the rotor forces, in the coordinate systems fixed in the rotors (the systems we end up with after the rotations used in the homogenous transformation matrices above) these are

$$\mathbf{f}_f^f = \begin{pmatrix} 0 \\ 0 \\ F_f \end{pmatrix}, \quad \mathbf{f}_b^b = \begin{pmatrix} 0 \\ 0 \\ F_b \end{pmatrix}.$$

To transform these to the inertial system, we must use the rotation matrices $\mathbf{R}_f^i$ and $\mathbf{R}_b^i$ from the homogenous transformation matrices,

$$\mathbf{f}_f^i = \mathbf{R}_f^i \mathbf{f}_f^f, \quad \mathbf{f}_b^i = \mathbf{R}_b^i \mathbf{f}_b^b.$$

The expression for the generalized force belonging the the generalized coordinate λ then becomes

$$\tau_\lambda = \frac{\partial \mathbf{r}_f^i}{\partial \lambda} \mathbf{R}_f^i \mathbf{f}_f^f + \frac{\partial \mathbf{r}_b^i}{\partial \lambda} \mathbf{R}_b^i \mathbf{f}_b^b,$$

and likewise for ϵ,

$$\tau_\epsilon = \frac{\partial \mathbf{r}_f^i}{\partial \epsilon} \mathbf{R}_f^i \mathbf{f}_f^f + \frac{\partial \mathbf{r}_b^i}{\partial \epsilon} \mathbf{R}_b^i \mathbf{f}_b^b,$$

where the expressions for $\mathbf{r}_f^i$ and $\mathbf{r}_b^i$ are given in (a).

Actually doing these calculations involves a lot of tedious and error-prone algebra, and is best done using symbolic computer software. This is illustrated in the end of the Matlab script above.

(The same procedure can of course also be used for the third generalized force, but this force can also easily be found from "looking", see next problem.)

Remark: In the lab manual, the generalized forces used to develop models for this system has apparantly been simplified by assuming $\sin\epsilon = 0$, $\cos\epsilon = 1$ and $\cos p = 1$.

(4 %) (f) What is τ_p, the generalized force for p? (You can rely on "physical insight" for finding τ_p, you do not have to use the procedure in the above question.)

Solution: The forces on the rotors give cause to a moment around the pitch-axis (the p-axis of rotation):

$$\tau_p = (F_b - F_f)L_h.$$

(4 %) (g) Explain how you can find the equations of motion based on the results above. You should not find them, but explain the procedure.

Solution: Forming the Lagrangian,

$$\mathcal{L}(\mathbf{q}, \dot{\mathbf{q}}) = T(\mathbf{q}, \dot{\mathbf{q}}) - U(\mathbf{q})$$

the equation of motion is given by calculating

$$\begin{aligned}
\frac{\mathrm{d}}{\mathrm{d}t}\frac{\partial \mathcal{L}}{\partial \dot{\lambda}} - \frac{\partial \mathcal{L}}{\partial \lambda} &= \tau_\lambda \\
\frac{\mathrm{d}}{\mathrm{d}t}\frac{\partial \mathcal{L}}{\partial \dot{\epsilon}} - \frac{\partial \mathcal{L}}{\partial \epsilon} &= \tau_\epsilon \\
\frac{\mathrm{d}}{\mathrm{d}t}\frac{\partial \mathcal{L}}{\partial \dot{p}} - \frac{\partial \mathcal{L}}{\partial p} &= \tau_p
\end{aligned}$$

Problem 4 (20 %)

In this problem, we consider a stirred tank that cools an inlet stream, see Figure 4. The tank is cooled by a "jacket" that contains a fluid of (presumably) lower temperature than the tank. The inlet stream to the tank has density ρ, temperature T_1, and massflowrate w_1. The outflow from the tank is

$$w_2 = Cu\sqrt{h},$$

where C is a constant and u is the valve opening. The liquid level is h. You can assume that the outflow is controlled such that the level does not exceed the height of the jacket.

The inlet and outlet massflowrates for the jacket is matched such that the jacket is always filled with fluid ($w_3 = w_4$). The cooling fluid has density ρ_c, and the inlet stream to the jacket has temperature T_3. Since the tank is stirred, we assume homogenous conditions, that is, the temperature T is the same everywhere in the tank. Similarly, we assume that the temperature T_c is the same everywhere in the jacket.

The cross-sectional area of the tank is A. The volume of the jacket is V_c.

The heat transfer from the tank to the jacket is

$$Q = Gh(T - T_c),$$

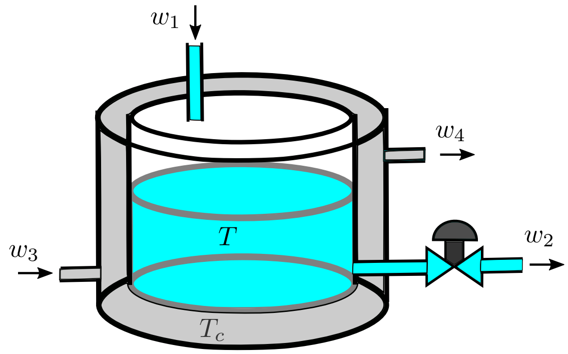

Figure 4: Tank with cooling jacket.

where h is the height of the liquid in the tank, and G a (constant) heat transfer coefficient. We assume that the jacket (and tank) is well insulated from the surroundings, meaning there are no other heat losses.

We assume both fluids incompressible, meaning that specific internal energy and enthalpy both can be assumed equal and proportional to temperature, with constant of proportionality being c_p and c_{pc} for the two fluids, respectively.

(20 %) (a) Set up differential equations for the temperatures T in the tank and T_c in the jacket, and the level h in the tank.

Solution: We must first set up mass balances. For the tank, the mass balance

$$\frac{\mathrm{d}}{\mathrm{d}t}(\rho A h) = w_1 - w_2$$

gives

$$\boxed{\dot{h} = \frac{1}{\rho A}\left(w_1 - Cu\sqrt{h}\right)}$$

For later use, the mass balance for the jacket:

$$\frac{\mathrm{d}}{\mathrm{d}t} m = w_3 - w_4 = 0.$$

Then, the energy balance for the tank gives

$$\begin{aligned}
\frac{\mathrm{d}}{\mathrm{d}t}\left(\rho c_p T A h\right) &= w_1 c_p T_1 - w_2 c_p T - Gh(T - T_c)\\
\rho c_p A h \frac{\mathrm{d}}{\mathrm{d}t} T + \rho c_p A T \frac{\mathrm{d}}{\mathrm{d}t} h &= w_1 c_p T_1 - w_2 c_p T - Gh(T - T_c)\\
\rho c_p A h \frac{\mathrm{d}}{\mathrm{d}t} T + \rho c_p A T \frac{1}{\rho A}\left(w_1 - w_2\right) &= w_1 c_p T_1 - w_2 c_p T - Gh(T - T_c)\\
\rho c_p A h \frac{\mathrm{d}}{\mathrm{d}t} T &= w_1 c_p (T_1 - T) - Gh(T - T_c)
\end{aligned}$$

$$\boxed{\frac{\mathrm{d}}{\mathrm{d}t} T = \frac{w_1}{\rho A h}(T_1 - T) - \frac{G}{\rho c_p A}(T - T_c)}$$

For the jacket:

$$\frac{\mathrm{d}}{\mathrm{d}t}\left(\rho_c c_{p,c} T_c V_c\right) = w_3 c_{p,c} T_3 - w_4 c_{p,c} T_c + Gh(T - T_c)$$

$$\rho_c c_{p,c} V_c \frac{\mathrm{d}}{\mathrm{d}t} T_c = w_3 c_{p,c} T_3 - w_4 c_{p,c} T_c + Gh(T - T_c)$$

Since $w_3 = w_4$:

$$\boxed{\frac{\mathrm{d}}{\mathrm{d}t} T_c = \frac{w_3}{\rho_c V_c}(T_3 - T_c) + \frac{Gh}{\rho_c c_{p,c} V_c}(T - T_c)}$$