From double pendulum with massless rods to simple robotic arms: mathematical modeling using Lagrange approach

Zdenek Hurak

December 5, 2012

1 Double pendulum with massless rods

Consider the system consisting of two beads of identical mass connected by massless rods of identical length as depicted in Fig.1. Note that our definition of the two generalized coordinates (angles) is different from most treatments in the books and online resources. We did this on purpose. First, to obtain a new assignment :-) but mainly because this fits nicely into our ultimate goal of deriving models of complex mechanic multibody systems such as robotic arms (and the rotary inverted pendulum too, of course...:-). For these systems, defining the generalized coordinates as the "relative" position variables makes much physical sense. These are then the variables that can be sensed with displacement or angular sensors.

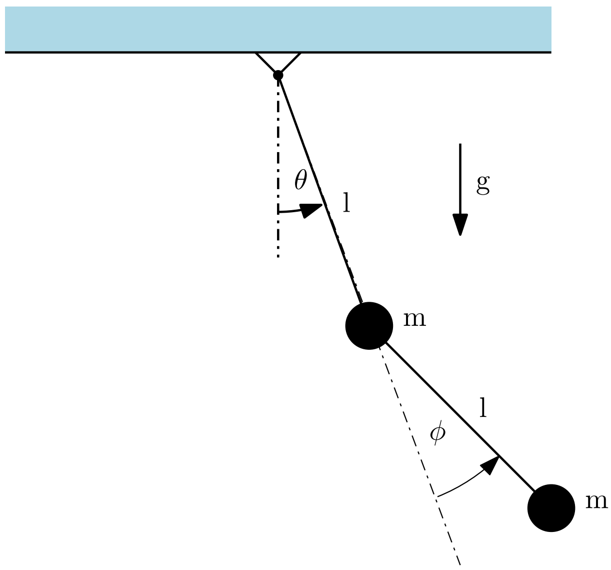


Figure 1: Double pendulum with massless rods

In order to evaluate the Lagrangian, we have to find the total kinetic and total potential energies. The nice fact is that it is possible to find the total kinetic energy as a sum of contributions by the individual parts. That is

$$T = T_1 + T_2 \tag{1}$$

and

$$V = V_1 + V_2. \tag{2}$$

Let's start with the individual contributions

$$T_1 = \frac{1}{2} m \mathbf{v}_1^2(t) \tag{3}$$

$$T_1 = \frac{1}{2} m \mathbf{v}_2^2(t), \tag{4}$$

where the square of the vector stands for the inner product $\mathbf{v}_1^{\mathrm{T}} \cdot \mathbf{v}_1$. Apparently we have to evaluate the velocities of the two bobs

$$\mathbf{v}_1^2(t) = \left(\frac{\mathrm{d}x_1(t)}{\mathrm{d}t}\right)^2 + \left(\frac{\mathrm{d}y_1(t)}{\mathrm{d}t}\right)^2, \tag{5}$$

$$\mathbf{v}_2^2(t) = \left(\frac{\mathrm{d}x_2(t)}{\mathrm{d}t}\right)^2 + \left(\frac{\mathrm{d}y_2(t)}{\mathrm{d}t}\right)^2. \tag{6}$$

In order to evaluate the x and y-components of the velocities above, the coordinates are

$$x_1(t) = \sin\theta(t) l, \tag{7}$$

$$y_1(t) = -\cos\theta(t) l \tag{8}$$

and

$$x_2(t) = x_1(t) + \sin\left(\theta(t) + \phi(t)\right) l = \sin\theta(t) l + \sin\left(\theta(t) + \phi(t)\right) l, \tag{9}$$

$$y_2(t) = y_1(t) - \cos\left(\theta(t) + \phi(t)\right) l = -\cos\theta(t) l - \cos\left(\theta(t) + \phi(t)\right) l. \tag{10}$$

Their derivatives with respect to time are

$$\frac{\mathrm{d}x_1(t)}{\mathrm{d}t} = \cos\theta(t)\dot{\theta}(t) l, \tag{11}$$

$$\frac{\mathrm{d}y_1(t)}{\mathrm{d}t} = \sin\theta(t)\dot{\theta}(t) l \tag{12}$$

and

$$\frac{\mathrm{d}x_2(t)}{\mathrm{d}t} = \cos\theta(t)\dot{\theta}(t) l + \cos(\theta(t) + \phi(t)) \left(\dot{\theta}(t) + \dot{\theta}(t)\right) l, \tag{13}$$

$$\frac{\mathrm{d}y_2(t)}{\mathrm{d}t} = \sin\theta(t)\dot{\theta}(t) l + \sin(\theta(t) + \phi(t)) \left(\dot{\theta}(t) + \dot{\phi}(t)\right) l. \tag{14}$$

The squared velocities of the first bead is (with the dependence on time omitted in favor of simplicity)

$$\mathbf{v}_1^2 = (\cos^2\theta + \sin^2\theta)\dot{\theta}^2 l^2, \tag{15}$$

which yields the kinetic energy

$$T_1 = \frac{1}{2} m l^2 \dot{\theta}^2. \tag{16}$$

The squared velocity of the second bead is a bit more involved (hence the derivation is more susceptible to errors, check the derivation)

$$
\begin{aligned}
\mathbf{v}_2^2 &= \left[\cos(\theta)\dot{\theta}l + \cos(\theta+\phi)\left(\dot{\theta}+\dot{\phi}\right)l\right]^2 + \left[\sin(\theta)\dot{\theta}l + \sin(\theta+\phi)\left(\dot{\theta}+\dot{\phi}\right)l\right]^2 \\
&= \cos^2(\theta)\dot{\theta}l + 2\cos(\theta)\dot{\theta}l\cos(\theta+\phi)\left(\dot{\theta}+\dot{\phi}\right)l + \cos^2(\theta+\phi)\left(\dot{\theta}+\dot{\phi}\right)^2 l^2 \\
&\quad + \sin^2(\theta)\dot{\theta}l + 2\sin(\theta)\dot{\theta}l\sin(\theta+\phi)\left(\dot{\theta}+\dot{\phi}\right)l + \sin^2(\theta+\phi)\left(\dot{\theta}+\dot{\phi}\right)^2 l^2 \\
&= (\cos^2\theta + \sin^2\theta)\dot{\theta}^2 l^2 + \left(\cos^2(\theta+\phi) + \sin^2(\theta+\phi)\right)\left(\dot{\theta}+\dot{\phi}\right)^2 l^2 \\
&\quad + 2\dot{\theta}\left(\dot{\theta}+\dot{\phi}\right)l^2 \underbrace{\left(\cos\theta \underbrace{\cos(\theta+\phi)}_{\cos\theta\cos\phi-\sin\theta\sin\phi} + \sin\theta \underbrace{\sin(\theta+\phi)}_{\sin\theta\cos\phi+\cos\theta\sin\phi}\right)}_{\cos\phi} \\
&= \dot{\theta}^2 l^2 + \left(\dot{\theta}+\dot{\phi}\right)^2 l^2 + 2\dot{\theta}\left(\dot{\theta}+\dot{\phi}\right)l^2\cos\phi \\
&= l^2\dot{\theta}^2 + l^2\left(\dot{\theta}^2 + 2\dot{\theta}\dot{\phi} + \dot{\phi}^2\right) + 2\left(\dot{\theta}^2 + \dot{\theta}\dot{\phi}\right)l^2\cos\phi \\
&= \left(2l^2 + 2l^2\cos\phi\right)\dot{\theta}^2 + \left(2l^2 + 2l^2\cos\phi\right)\dot{\theta}\dot{\phi} + \left(l^2\right)\dot{\phi}^2. \qquad (17)
\end{aligned}
$$

The contribution of the second bead to the kinetic energy is

$$
\begin{aligned}
T_2 &= \frac{1}{2}m\left(2l^2 + 2l^2\cos\phi\right)\dot{\theta}^2 + \frac{1}{2}m\left(2l^2 + 2l^2\cos\phi\right)\dot{\theta}\dot{\phi} + \frac{1}{2}m\left(l^2\right)\dot{\phi}^2 \\
&= ml^2\left(1+\cos\phi\right)\dot{\theta}^2 + ml^2\left(1+\cos\phi\right)\dot{\theta}\dot{\phi} + \frac{1}{2}ml^2\dot{\phi}^2. \qquad (18)
\end{aligned}
$$

The total kinetic energy is

$$
\begin{aligned}
T &= \frac{1}{2}ml^2\dot{\theta}^2 + ml^2\left(1+\cos\phi\right)\dot{\theta}^2 + ml^2\left(1+\cos\phi\right)\dot{\theta}\dot{\phi} + \frac{1}{2}ml^2\dot{\phi}^2 \\
&= ml^2\left(\frac{3}{2}+\cos\phi\right)\dot{\theta}^2 + ml^2\left(1+\cos\phi\right)\dot{\theta}\dot{\phi} + \frac{1}{2}ml^2\dot{\phi}^2. \qquad (19)
\end{aligned}
$$

Now comes the time for deriving the potential energy. This should be a little bit more straightforward

$$
\begin{aligned}
V_1 &= mgy_1 = -mgl\cos\theta, \qquad (20) \\
V_2 &= mgy_2 = -mgl\cos\theta - mgl\cos\left(\theta+\phi\right), \qquad (21)
\end{aligned}
$$

and

$$
V = -2mgl\cos\theta - mgl\cos\left(\theta+\phi\right). \qquad (22)
$$

We could easily add some constant to the potential energy to achieve zero value for $\theta = \phi = 0$. But since we are only interested in the derivative with respect to the two angles, we do not care about the particular value of the potential energy.

Now let's form the Lagrangian

$$
L = ml^2\left(\frac{3}{2}+\cos\phi\right)\dot{\theta}^2 + ml^2\left(1+\cos\phi\right)\dot{\theta}\dot{\phi} + \frac{1}{2}ml^2\dot{\phi}^2 + 2mgl\cos\theta + mgl\cos\left(\theta+\phi\right). \qquad (23)
$$

And the final effort (not quite insignificant) is to find all the various derivatives

$$\frac{\partial L}{\partial \dot\theta} = 2ml^2\left(\frac{3}{2}+\cos\phi\right)\dot\theta + ml^2\left(1+\cos\phi\right)\dot\phi, \tag{24}$$
$$\frac{\partial L}{\partial \dot\phi} = ml^2(1+\cos\phi)\dot\theta + ml^2\dot\phi, \tag{25}$$
$$\frac{\mathrm{d}}{\mathrm{d}t}\frac{\partial L}{\partial \dot\theta} = 2ml^2\frac{3}{2}\ddot\theta - 2ml^2\sin(\phi)\dot\phi\dot\theta + 2ml^2\cos(\phi)\ddot\theta + ml^2\ddot\phi - ml^2\sin(\phi)\dot\phi^2 + ml^2\cos\phi\ddot\phi, \tag{26}$$
$$\frac{\mathrm{d}}{\mathrm{d}t}\frac{\partial L}{\partial \dot\phi} = ml^2\ddot\theta - ml^2\sin(\phi)\dot\phi\dot\theta + ml^2\cos(\phi)\ddot\theta + ml^2\ddot\phi, \tag{27}$$
$$\frac{\partial L}{\partial \theta} = -2mgl\sin\theta - mgl\sin(\theta+\phi), \tag{28}$$
$$\frac{\partial L}{\partial \phi} = -ml^2\dot\theta^2\sin\phi - ml^2\dot\theta\dot\phi\sin\phi - mgl\sin(\theta+\phi). \tag{29}$$

Now we substitute into the notoriously known Lagrange equations. In this case, they are just two, one corresponding to derivatives with respect to θ and the other with respect to ϕ

$$\frac{\mathrm{d}}{\mathrm{d}t}\frac{\partial}{\partial \dot\theta}L - \frac{\partial}{\partial \theta}L = 0, \tag{30}$$
$$\frac{\mathrm{d}}{\mathrm{d}t}\frac{\partial}{\partial \dot\phi}L - \frac{\partial}{\partial \phi}L = 0. \tag{31}$$
$$\tag{32}$$

This results in two second-order ordinary differential equations in two variables. These are (after putting together the corresponding terms and dividing both equations by ml^2):

$$(3+2\cos\phi)\ddot\theta + (1+\cos\phi)\ddot\phi - 2\sin(\phi)\dot\phi\dot\theta - \sin(\phi)\dot\phi^2 + 2\frac{g}{l}\sin\theta + \frac{g}{l}\sin(\theta+\phi) = 0, \tag{33}$$
$$(1+\cos\phi)\ddot\theta + \ddot\phi + \dot\theta^2\sin\phi + \frac{g}{l}\sin(\theta+\phi) = 0. \tag{34}$$

These are our desired differential equations — implicit second order ordinary differential equations. However, in order to simulate these equations, we either first convert the two equations into four first-order explicit ODEs (aka state-space model) and use standard solvers, or we can also invoke one of the available solvers for implicit equations. Proceeding in Modelica, we only need to convert the model to four first-order implicit equations.

```
model DoublePendulum "Double pendulum with masless rods"
  parameter Real m = 1;
  parameter Real l = 1;
  parameter Real g = 9.81;
  Real theta(start = 0, fixed = true);
  Real phi(start = 1, fixed = true);
  Real Dtheta(start = 0, fixed = true);
  Real Dphi(start = 0, fixed = true);
  annotation(experiment(StartTime = 0.0, StopTime = 10.0, Tolerance = 0.000001));
equation
  (3+2*cos(phi))*der(Dtheta) + (1+cos(phi))*der(Dphi) - 2* sin(phi)*Dphi*Dtheta
     - sin(phi)*Dphi^2 + 2*g/l*sin(theta) + g/l*sin(theta+phi) = 0;
  (1+cos(phi))*der(Dtheta) + der(Dphi) + sin(phi)*Dtheta^2 + g/l*sin(theta+phi) = 0;
  Dtheta = der(theta);
  Dphi = der(phi);
end DoublePendulum;
```

Compare the simulation results with the the alternative model where the angle characterizing the second bob is defined as a deviation from the vertical direction. The equations are given, for example, at `http://scienceworld.wolfram.com/physics/DoublePendulum.html`. The model in Modelica is:

```
model DoublePendulumEricWeisstein "Double pendulum with masless rods"
  parameter Real m = 1;
  parameter Real m1 = m,m2 = m;
```

```
  parameter Real l = 1;
  parameter Real l1 = l,l2 = l;
  parameter Real g = 9.81;
  Real theta1(start = 0, fixed = true);
  Real theta2(start = 3, fixed = true);
  Real theta1dot(start = 0, fixed = true);
  Real theta2dot(start = 0, fixed = true);
  Real phi(start = 3, fixed = true);
  annotation(experiment(StartTime = 0.0, StopTime = 10.0, Tolerance = 0.00000000001));
equation
  (m1+m2)*l1*der(theta1dot) + m2*l2*der(theta2dot)*cos(theta1-theta2)
     + m2*l2*theta2dot^2*sin(theta1-theta2) + g*(m1+m2)*sin(theta1) = 0;
  m2*l2*der(theta2dot) + m2*l1*der(theta1dot)*cos(theta1-theta2)
     - m2*l1*theta1dot^2*sin(theta1-theta2) + m2*g*sin(theta2) = 0;
  theta1dot = der(theta1);
  theta2dot = der(theta2);
  phi = theta2 - theta1;
end DoublePendulumEricWeisstein;
```

If a more complicated processing of the simulation output is needed, for instance if visualization of the pendulum is desired, the Modelica simulation outputs can be imported in an environment of choice, say, Matlab. For that purpose, Modelica stores the simulation data in mat-files which Matlab can easily import. Check the setting for your OMEdit environment in which you specify the working directory for OpenModelica. In my case, I set the working directory to the subdirectory /Dokumenty/openmodelica of my home directory (on a Linux Machine; correct accordingly on you MS Windows machine). The Matlab code for importing and visualizing follows

```
%% Importing the data produced by OpenModelica
load('~/Dokumenty/openmodelica/DoublePendulum_res.mat')

name'   % check the names of the variables and possibly correct below

t = data_2(1,:);
theta = data_2(5,:);
phi = data_2(4,:);

l = 1;

%% Computing the xy coordinates of the bobs for my own solution
x1 = sin(theta)*l;
y1 = -cos(theta)*l;

x2 = x1 + sin(theta+phi)*l;
y2 = y1 - cos(theta+phi)*l;

%% Plotting the response in xy
F(length(theta)) = struct('cdata',[],'colormap',[]);
figure(1)
for k = 1:length(theta)
    plot([0 x1(k) x2(k)],[0 y1(k) y2(k)],'-o')
    xlabel('x_[m]')
    ylabel('y_[m]')
    axis([-2 2 -3 1])
    F(k) = getframe;
end

%% Playing the animation
movie(F,1,10)
```

Note the two useful commands *getframe* and *movie*, which record and play the movie, respectively. Running the code produces the animation as in Fig. 2.

If numerical solution of the ODE is preferably done in Matlab, first the two second-order ODEs (33) and (34) need to be converted to explicit ones (apart from converting them to first-order equations as before). We proceed by expressing $\ddot{\phi}$ from the latter and substitute it into the former. Then $\ddot{\theta}$ is evaluated at (33) and substituted into (34). This straightforward procedure yields

$$\ddot{\theta} = \frac{1}{2-\cos^2(\phi)}\left[\sin(\phi)(2+\cos\phi)\dot{\phi}^2 + \frac{g}{l}\cos\phi\sin(\theta+\phi) + 2\sin(\phi)\dot{\theta}\dot{\phi} - \frac{2g}{l}\sin\theta\right], \tag{35}$$

$$\ddot{\phi} = -\frac{1+\cos(\phi)}{2-\cos^2(\phi)}\left[\sin(\phi)(2+\cos\phi)\dot{\phi}^2 + \frac{g}{l}\cos\phi\sin(\theta+\phi) + 2\sin(\phi)\dot{\theta}\dot{\phi} - \frac{2g}{l}\sin\theta\right] - \sin(\phi)\dot{\theta}^2 - \frac{g}{l}\sin(\theta+\phi). \tag{36}$$

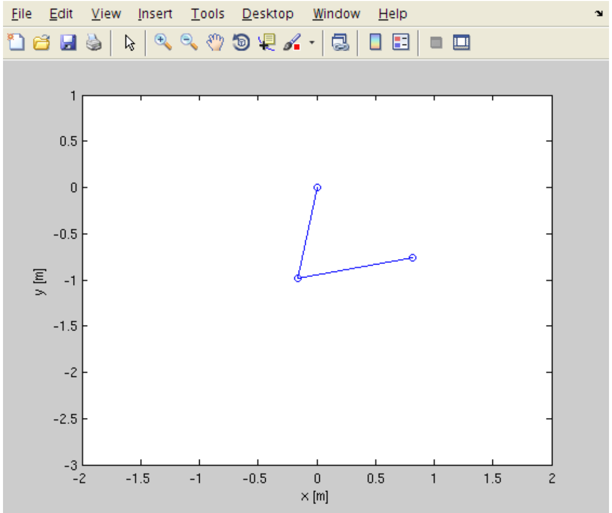


Figure 2: Screenshot from the animation of a double pendulum with massless rods in Matlab

In order to solve (simulate) these two second-order explicit equations in Matlab, an m-function needs to be written

```
function dydt = double_bob_pendulum(t,y)
theta = y(1);
phi = y(2);
dthetadt = y(3);
dphidt = y(4);

g = 9.81;
l = 1;
m = 1;

dydt = zeros(4,1);
dydt(1) = dthetadt;
dydt(2) = dphidt;
dydt(3) = 1/(2+(cos(phi))^2)*((2+cos(phi))*sin(phi)*dphidt^2 + ...
   cos(phi)*g/l*sin(theta+phi) + 2*sin(phi)*dphidt*dthetadt - 2*g/l*sin(theta));
dydt(4) = -(1+cos(phi))/(2+(cos(phi))^2)*((2+cos(phi))*sin(phi)*dphidt^2 + ...
   cos(phi)*g/l*sin(theta+phi) + 2*sin(phi)*dphidt*dthetadt - 2*g/l*sin(theta)) - ...
   sin(phi)*dthetadt^2 - g/l*sin(phi+theta);
```

Then this function is invoked using the mechanism of a *function handle* in combination with the general-purpose ODE solver ode45

```
[t,y] = ode45(@double_bob_pendulum,[0,10],[0,3,0,0]);
```

As a side note, it is well-known that some models like the pendulum exhibit so-called *chaotic* behavior, which means that they are very sensitive to the initial conditions. Let's check it, just out of curiosity. Let's consider the initial value for *phi* perturbed a bit. Hence it will start not at $\phi(0) = 3$ but at $\phi(0) = 3.01$:

```
[t2,y2] = ode45(@double_bob_pendulum,[0,10],[0,3.01,0,0]);
```

The two responses are at Fig. 3. Apparently, the two responses stop resembling each other after something like 7 s.

Can this be helped somehow? Partially yes. The problems can only be postponed to later simulation times by decreasing the tolerance of numerical errors in the algorithm — there is a parameter of the numerical solver called *relative tolerance*. We will talk about it more later in the course, but right now you can use it as a magic knob to improve the accuracy of your solution (of course at the cost of an increased computational effort). You set it through the *odeset* command

```
options = odeset('RelTol',1e-6);
[t,y] = ode45(@double_bob_pendulum,[0,10],[0,3,0,0],options);
[t2,y2] = ode45(@double_bob_pendulum,[0,10],[0,3+0.01,0,0],options);
```

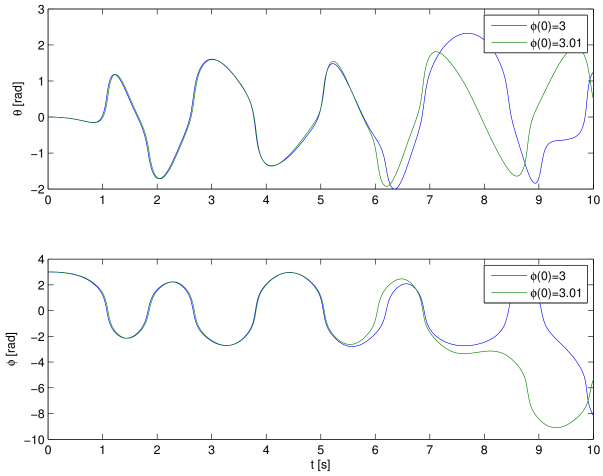


Figure 3: Responses for two nearby initial angles of ϕ

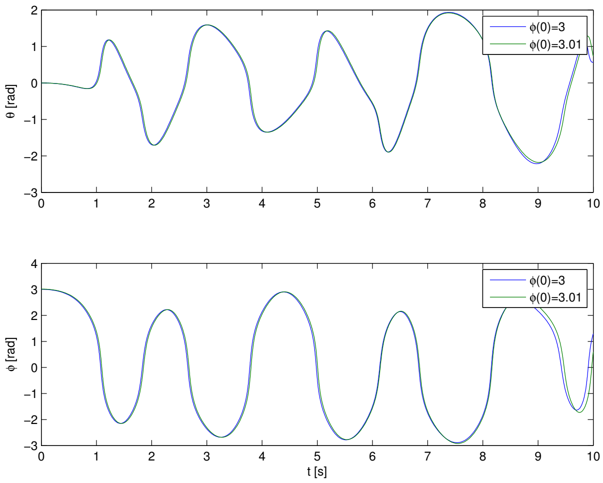


Figure 4: Responses for two nearby initial angles of ϕ

The new simulation results are at Fig. 4.

Now, understanding how challenging this deceptively simple model of a double pendulum is, let's compare the simulation outputs from Matlab and OpenModelica. The responses are at Fig. 5.

The highest time to abandon the textbook-scale double pendulum with two bobs and massless rods in favor of a bit more realistic setup.

2 Double pendulum formed by two rigid links and free joints approached directly

Instead of the mass concentrated at single points (bobs), consider it distributed uniformly along rigid links. The length l of each of the two links is identical with the length of the massless rods in the previous case. Also, the mass m of each rigid bars agrees with the mass of the bobs. The sketch is at Fig. 6.

Additional information encoded in the above figure are three coordinate systems — $o_0x_0y_0$, $o_1x_1y_1$ and $o_2x_2y_2$, colored red, blue and green, respectively. Their origins (o_0, o_1 and o_2) are conveniently located at the joints but other choices are possible in principle. The only requirement is that they must be firmly attached to the respective moving bodies. As a matter of fact, this requirement enforces that

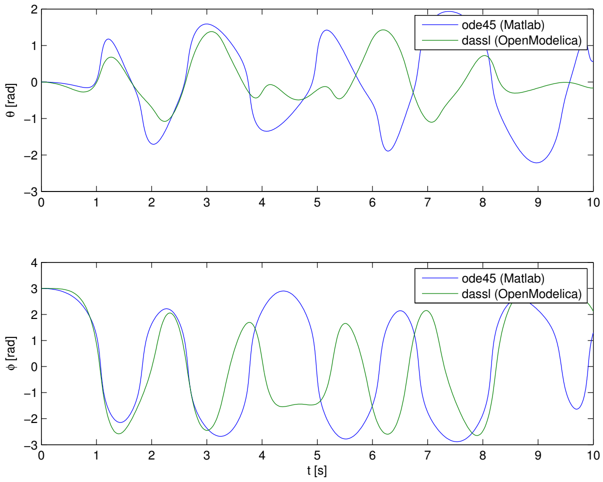


Figure 5: Comparing the simulation outputs from Matlab and OpenModelica

then the generalized coordinates are "relative" angles. The role of these coordinate frames is to "merely" assist us in composing vectors of absolute velocities by adding vectors of relative velocities. This will only become useful once we switch to full 3D space. In the planar example that we are considering, we could easily live without (as we previously did in the massless rod example).

Our particular choice in the figure is quite convenient in that for zero angular deviations, all the three coordinate systems are oriented identically (although their origins are displaced). Other choices are acceptable as well. In the robotics literature, there are some conventions which simplify things a bit on the notation side. For instance, Denavit-Hartenberg (DH) convention is quite popular. But we skip this in our introductory study.

From the modeling perspective, the situation is similar to the massless rod case with one key difference: now the kinetic energy is not only accumulated in the translation (change of position) but also in rotation (change of orientation).

For the analysis of translation, it is natural to focus on the motion of the *center of mass* of the bar. The rotation is then naturally analyzed around the axis going through the center of mass. This is sketched in some detail for a single-bar pendulum in the left part of Fig. 7 — the translation velocity of the center of mass is v_{c1} and the angular velocity is $\omega_{01} = \dot{\theta}$ (the lower index 01 reminds us that this is rotation of the frame 1 with respect to the frame 0, hence absolute velocity).

The kinetic energy can be composed of two components

$$T_{1\text{ trans}} + T_{1\text{ rot}}. \tag{37}$$

The translation motion is along a circular path, hence the velocity is given by a vector $\mathbf{v}_{c1}$ which is orthogonal to the bar (that is, in the direction of y_1)) and its size is

$$v_{c1} = \frac{l}{2}\omega_{01} = \frac{l}{2}\dot{\theta}. \tag{38}$$

The contribution of translation motion to the total kinetic energy is

$$T_{1\text{translation}} = \frac{1}{2} m v_{c1}^2 = \frac{1}{2} m \left(\frac{l}{2}\right)^2 \dot{\theta}^2. \tag{39}$$

The other component of the motion — the rotation — is directly characterized by the angular velocity $\dot{\theta}$ and contributes to the kinetic energy by the term

$$T_{1\text{rotation}} = \frac{1}{2} I_c \dot{\theta}^2, \tag{40}$$

where I_c is the moment of inertia around the center. Hence the total kinetic energy is

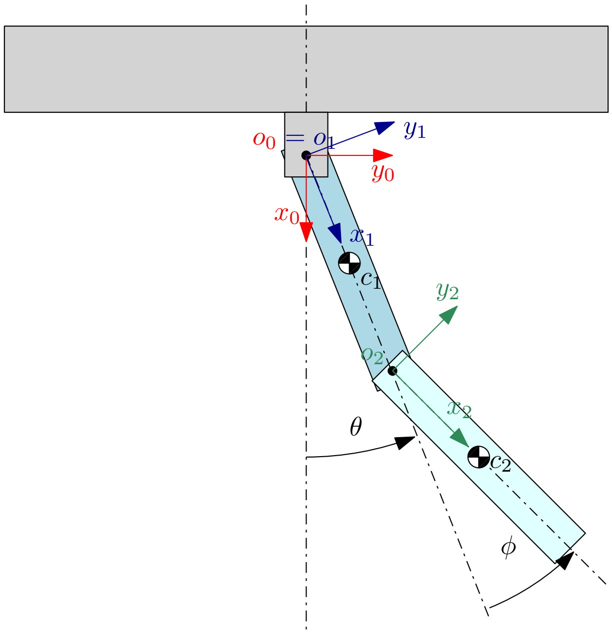


Figure 6: Double pendulum with rigid links

$$T_1 = \frac{1}{2}\left(I_c + m\left(\frac{l}{2}\right)^2\right)\dot{\theta}^2. \tag{41}$$

However, it may be convenient to analyze the contribution of the motion of the pendulum to the kinetic energy while considering the motion with respect to the joint — the origin o_1 of the coordinate system $o_1x_1y_1$. This is sketched in the right part of Fig. 7. Focusing on this new reference point, the pendulum now experiences only a rotation with respect to the (inertial) frame 0, no translation. This rotation is characterized by the same angular rate $\dot{\theta}$. Nonetheless, we have to realize that the rotation motion considered in this case needs to be considered with respect to another axis. Invoking the *Steiner's theorem*, the moment of inertia changes to

$$I = I_c + m\left(\frac{l}{2}\right)^2. \tag{42}$$

Clearly, the kinetic energy accumulated in the rotational motion agrees with (41) and we can conclude that, the choice of the reference point, that is, the point with respect to which rotation is considered and whose translation is considered, has no impact on the resulting kinetic energies.

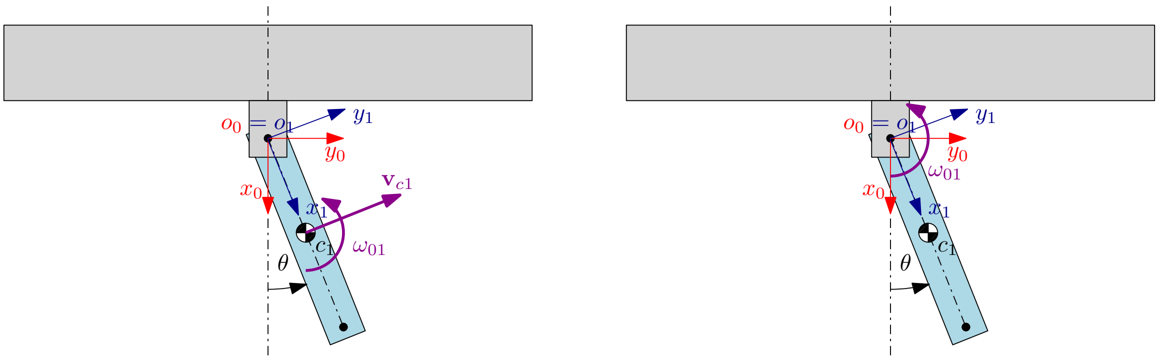
It is tempting to extend this trick to the second bar in the double pendulum, that is, instead of translation of the center of mass c_2 and rotation around the same point we may want to shift the point of reference to o_2. The important point is that this time it is no longer possible to perform this shift in the same way as for the first bar. Let's have a closer look at the situation in Fig. 8.

The coordinate frame $o_2x_2y_2$ is attached to the second rigid bar of a double pendulum and the origin o_2 of the frame is attached to the joint and as such moves with the absolute velocity $\mathbf{v}_{02} = l\dot{\theta}$. The center c_2 of mass moves with the relative velocity $\mathbf{v}_{o_2c_2}$ with respect to the joint. It is orthogonal to the x_2 axis and its magnitude is

$$v_{o_2c_2} = (\dot{\theta} + \dot{\phi})\frac{l}{2}. \tag{43}$$

The absolute velocity of c_2, that is, the velocity $\mathbf{v}_{0c_2}$ is given by the sum of the above two velocities

$$\mathbf{v}_{0c_2} = \mathbf{v}_{02} + \mathbf{v}_{o_2c_2}. \tag{44}$$



(a) Translation of the center of mass and rotation around the center of mass are considered

(b) Translation of the left end of the bar (the one at the joint) is considered — it is zero. Rotation about the same point is considered

Figure 7: Two choices of a reference point for analysis of translation and rotation motion for a single rigid bar pendulum.

This is the velocity that should be used to evaluate kinetic energy. Kinetic energy is then given by

$$
\begin{aligned}
T_2 &= \underbrace{\frac{1}{2}m\left(\mathbf{v}_{02} + \mathbf{v}_{o_2c_2}\right)^2}_{T_{2trans}} + \underbrace{\frac{1}{2}I_c(\underbrace{\omega_{01} + \omega_{12}}_{\omega_{02}})^2}_{T_{2\ \mathrm{rot}}}, \\
&= \frac{1}{2}m(\mathbf{v}_{02})^2 + \frac{1}{2}m(\mathbf{v}_{o_2c_2})^2 + {\color{red}m\mathbf{v}_{02}^{\mathrm{T}} \cdot \mathbf{v}_{o_2c_2}} + \frac{1}{2}I_c(\dot{\theta} + \dot{\phi})^2, \\
&= \frac{1}{2}m(\mathbf{v}_{02})^2 + \frac{1}{2}\left(\underbrace{I_c + m\left(\frac{l}{2}\right)^2}_{I}\right)(\dot{\theta} + \dot{\phi})^2 + {\color{red}m\mathbf{v}_{02}^{\mathrm{T}} \cdot \mathbf{v}_{o_2c_2}}.
\end{aligned} \tag{45}
$$

Note the surprising third term (in red)! It includes an inner product of two vector, hence it can be evaluated as

$$
m\mathbf{v}_{02}^{\mathrm{T}} \cdot \mathbf{v}_{o_2c_2} = mv_{02}v_{o_2c_2}\cos\phi = ml\dot{\theta}\frac{l}{2}(\dot{\theta} + \dot{\phi})\cos\phi. \tag{46}
$$

To conclude, one must not forget that when the reference point is changed from the center of mass to something else, typically a rotation joint, the contribution of the given bar to the total kinetic energy is not only composed of the component corresponding to translation of the reference point and the component corresponding to rotation around that point but there is also some new term!

The whole story is explained in every other textbook on analytical mechanics, for example [2] and [1], which were used in preparation of this course.

Rather then finishing the derivation here, let us move on to introduction of a bit more systematic tools — rotation transformations — which will lead us towards finishing the model.

3 Double pendulum formed by two rigid links and free joints approached through rotational transformations

The above procedure is completely straightforward and transparent. However, it is rather tedious and error-prone. Here we introduce a more systematic procedure based on rotational transformations. Moreover, this would allow us to solve problems in 3D.

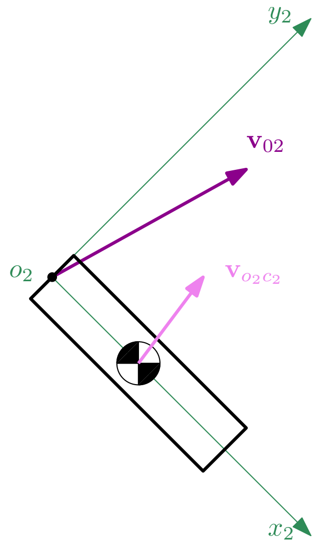
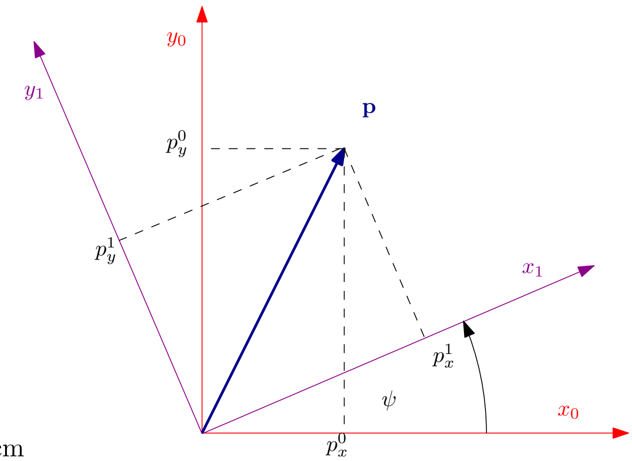


Figure 8: The coordinate frame attached to the second rigid bar of a double pendulum. The origin o_2 of the frame (attached to the joint) moves with the absolute velocity v_2. The center c_2 of mass moves with the relative velocity $v_{o_2c_2}$ with respect to the joint.

In order to introduce these tools, it is vital to recognize the difference between a vector and the n-tuple of its coordinates. These coordinates are intimately tied to the underlying coordinate system. The coordinates are different for different coordinate frames. Consider the 2D situation in Fig. 9.



cmcm

Figure 9: A vector expressed in two coordinate frames

The vector $\mathbf{p}$ can be expressed in the coordinate frame given by the origin 0 and the two directions x_0 and y_0 and its coordinates are given by the column $\mathbf{p}^0$

$$\mathbf{p}^0 = \begin{bmatrix} p_x^0 \\ p_y^0 \end{bmatrix}. \tag{47}$$

Alternatively, the same vector can be expressed in the coordinate system given by (again) the origin 0 and the directions x_1 and y_1 and its coordinates are then

$$\mathbf{p}^1 = \begin{bmatrix} p_x^1 \\ p_y^1 \end{bmatrix}. \tag{48}$$

What we are heading for is a relation between $\mathbf{p}^0$ and $\mathbf{p}^1$. It is easy to check that this relationship is expressed as

$$\begin{bmatrix} p_{x0} \\ p_{y0} \end{bmatrix} = \underbrace{\begin{bmatrix} \cos\psi & -\sin\psi \\ \sin\psi & \cos\psi \end{bmatrix}}_{R_1^0} \begin{bmatrix} p_{x1} \\ p_{y1} \end{bmatrix} \tag{49}$$

The matrix relating the two columns is denoted R_1^0 as it transforms the coordinates given in the system 1 into the coordinates in the system 0. An important property of this *rotation matrix* is that its inverse can be computed by mere transposition and, moreover, it can be interpreted as a transformation from the system 0 back to the system 1

$$(R_1^0)^{-1} = (R_1^0)^{\mathrm{T}}. \tag{50}$$

Now we can embed the 2D scenario into the full 3D by considering a third direction — the z axes — pointing out of the plane (in order to form an orthogonal right-hand coordinate system). Our rotation matrix can be modified formally to

$$R_1^0 = \begin{bmatrix} \cos\psi & -\sin\psi & 0 \\ \sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{51}$$

An alternative notation can be useful here which suggests that the original coordinate system is rotated around its own coordinate axis z by the angle ψ

$$R_1^0 = R_{z,\psi}. \tag{52}$$

Of course, it is also possible to rotate the system around other key axis, namely x and y. The rotation matrices can be shown to be

$$R_{x,\phi} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & -\sin\phi \\ 0 & \sin\phi & \cos\phi \end{bmatrix} \tag{53}$$

and (mind the minus sign here)

$$R_{y,\theta} = \begin{bmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{bmatrix}. \tag{54}$$

It is naturally possible that one coordinate systems is actually rotated with respect to another system in yet another way — about an arbitrary axis. It is a fundamental result that such rotation can be described as if the two coordinate system were sequentially rotated using the three basic rotations introduced above. The key point is, however, that the order of the rotation is of fundamental importance. Once it is specified, it must be kept. There are many conventions in the literature and they range a community from community. For example, in the aerospace community it is common to relate two coordinate systems by this sequence of rotations

$$R_{x^2,\phi} R_{y^1,\theta} R_{z^0,\psi}. \tag{55}$$

Two points are worth emphasizing. First, the subsequent rotations are considered around an axis in the new (rotated) coordinate system. Second, in order to interpret the above sequence of rotations, one must start reading from the right. Hence, we first rotate the object (robot arm, aircraft) around its z axis (typically a vertical axis, the positive direction of rotation is determined by the fact whether the axis is directed downwards or upwards, in aerospace it directs downwards and the positive rotation is to the right). Then we rotate the object around its new horizontal y axis (directing to the left in the aerospace, hence the positive rotation corresponds to the motion "nose up" in aerospace). Finally, the object is rotated around its new x axis. Indeed, the symbol R stands both for the rotation transformation and for the matrix which expresses it, hence the composed rotation can be characterized by a product of the three matrices.

Fine, now we know how to express a vector for which we have its components in one coordinate frame in a new, rotated coordinate frame. The key use of this skill is when we want to combined the angular velocities of the individual rigid bodies.

In particular, in the above example of a two-rigid-link pendulum, the total rotation velocity of the second coordinate frame (that is, angular velocity of the second coordinate frame with respect to the zero-th coordinate frame, denoted ω_{02}) was obtained as $\dot{\phi} + \dot{\theta}$. However, this was so simple only thanks to the fact that the whole problem is only two-dimensional (planar). Therefore we perhaps even did not realize that we are actually combining two vectors. True, the two vectors are aligned and their direction was out of plane, hence we did not emphasize their vector character and just summed their sizes. But in general, we must keep the vector character in mind and when summing the vectors, they must be first transformed to a common coordinate system. In our case we can choose to express all the vectors in the zero-th frame

$$\begin{aligned}\boldsymbol{\omega}_{02}^0 &= \boldsymbol{\omega}_{01}^0 + \boldsymbol{\omega}_{12}^0, \\ &= R_1^0 \boldsymbol{\omega}_{01}^1 + \underbrace{R_1^0 R_2^1}_{R_2^0} \boldsymbol{\omega}_{12}^2. \end{aligned} \tag{56}$$

We can also express the angular velocity in the second frame

$$\boldsymbol{\omega}_{02}^1 = \boldsymbol{\omega}_{01}^1 + \boldsymbol{\omega}_{12}^1. \tag{57}$$

We will discuss in a while which one of the common coordinate frames will be more useful when calculating the rotational contribution to the kinetic energy. For the time being, let's state that both can lead us to a correct result.

Similarly we can evaluate the translation velocities. With the placement of the the coordinate systems at the tops of the individual links we can use the full vector version of the well-known relationship $v = \omega r$

$$\underbrace{\dot{\mathbf{r}}_{01}^0}_{\mathbf{v}_{01}^0} = \boldsymbol{\omega}_{01}^0 \times \mathbf{r}_{01}^0, \tag{58}$$

where the symbol $\times$ stands for the vector (outer) product and the vector $\mathbf{r}$ describes displacement of the origin of the coordinate system 1 with respect to 0.

Similarly,

$$\mathbf{v}_{02}^0 = \mathbf{v}_{01}^0 + \mathbf{v}_{12}^0, \tag{59}$$

$$= \mathbf{v}_{01}^0 + \boldsymbol{\omega}_{12}^0 \times \mathbf{r}_{12}^0, \tag{60}$$

$$= \mathbf{v}_{01}^0 + R_1^0 (\boldsymbol{\omega}_{12}^1 \times \mathbf{r}_{12}^1). \tag{61}$$

Once again, the coordinate frame within which the above velocities are expresses, is completely our choice. We only have to be careful about expressing the vectors in a common coordinate frame.

Let's check the validity of the above equation in our case of a two-bar pendulum

$$\mathbf{v}_{01}^0 = \begin{bmatrix} 0 \\ 0 \\ \dot{\theta} \end{bmatrix} \times \begin{bmatrix} l\cos\theta \\ l\sin\theta \\ 0 \end{bmatrix} = \begin{bmatrix} -\dot{\theta} l \sin\theta \\ \dot{\theta} l \cos\theta \\ 0 \end{bmatrix} \tag{62}$$

and

$$\mathbf{v}_{02}^{0} = \begin{bmatrix} -\dot{\theta} l \sin\theta \\ \dot{\theta} l \cos\theta \\ 0 \end{bmatrix} + \begin{bmatrix} \cos\phi & -\sin\phi & 0 \\ \sin\phi & \cos\phi & 0 \\ 0 & 0 & 1 \end{bmatrix} \left(\begin{bmatrix} 0 \\ 0 \\ \dot{\phi} \end{bmatrix} \times \begin{bmatrix} l\cos\phi \\ l\sin\phi \\ 0 \end{bmatrix} \right) \tag{63}$$

$$= \begin{bmatrix} -\dot{\theta} l \sin\theta \\ \dot{\theta} l \cos\theta \\ 0 \end{bmatrix} + \begin{bmatrix} \cos\phi & -\sin\phi & 0 \\ \sin\phi & \cos\phi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} -\dot{\phi} l \sin\phi \\ \dot{\phi} l \cos\phi \\ 0 \end{bmatrix}, \tag{64}$$

$$= \begin{bmatrix} -\dot{\theta} l \sin\theta - 2l\sin\phi\cos\phi\dot{\phi} \\ \dot{\theta} l \cos\theta - l\sin^2\phi\dot{\phi} + l\cos^2\dot{\phi} \\ 0 \end{bmatrix}. \tag{65}$$

Knowing how to express both the rotation and translation velocities of the individual frames (attached to the moving parts), we proceed to evaluating the kinetic energies.

The translation contributions are straightforward. The velocities only need to be expressed in the inertial coordinate frame and then the general expression for the kinetic energy for the i-th frame is

$$T_{i\ \text{trans}} = \frac{1}{2} m_i \left(\mathbf{v}_{0i}^{0}\right)^{\mathrm{T}} \mathbf{v}_{0i}^{0}. \tag{66}$$

Some discussion is needed for the contribution of the rotation motion. Whereas in the planar (2D) case the contribution of a general i-th frame was $\frac{1}{2}J_i(\omega_{0i})^2$, in the full 3D the concept of the moment of inertia needs to be redefined (or extended). It can no longer be just a scalar number because the rotation can be realized around an arbitrary axis. The moment of inertia with respect to a given axis is

$$I = \iiint \rho(x,y,z)\mathbf{r}^2(x,y,z)\mathrm{d}x\mathrm{d}y\mathrm{d}z \tag{67}$$

For example, the rotation about the x axis is labeled as I_{xx} and is given as

$$I_{xx} = \rho \iiint (\mathbf{y}^2 + \mathbf{z}^2)\mathrm{d}x\mathrm{d}y\mathrm{d}z \tag{68}$$

Considering that the rotation can really be just about any axis, the moment of inertial grows to a matrix, typically denoted I and given as a symmetric matrix

$$I = \begin{bmatrix} I_{xx} & I_{xy} & I_{xz} \\ I_{yx} & I_{yy} & I_{yx} \\ I_{zx} & I_{zy} & I_{zz} \end{bmatrix}, \tag{69}$$

where the off-diagonal terms are given by

$$I_{xy} = \rho \iiint (\mathbf{x}^{\mathrm{T}} \cdot \mathbf{y})\mathrm{d}x\mathrm{d}y\mathrm{d}z \tag{70}$$

and the kinetic energy is then written as

$$T_{i\ \text{rot}} = \frac{1}{2}(\boldsymbol{\omega}_{0i}^{0})^{\mathrm{T}} I^{0} \boldsymbol{\omega}_{0i}^{0}, \tag{71}$$

where the upper index with the matrix of moment of inertia reminds us that even the moment of inertia is always specified with respect to a certain coordinate system (the coordinate system in which we did the integration). It is clear that when a general object is rotated, its moment of inertia can change because the integration is always with respect to the 0-th (global) coordinate system. This make the moment of inertia dependent on the orientation, which is quite inconvenient. As a solution, one can define the moment of inertia with respect to the coordinate system fixed to the body. No matter how the body rotates, the integration in (68) always yields the same result — a constant matrix. Hence we have another way how to specify the kinetic energy

$$T_{i\ \mathrm{rot}} = \frac{1}{2}(\boldsymbol{\omega}_{0i}^{i})^{\mathrm{T}} I^{i} \underbrace{\boldsymbol{\omega}_{0i}^{i}}_{R_0^1\boldsymbol{\omega}_{0i}^{0}}, \tag{72}$$

from which the relationship between the two moments of inertia follows

$$I^0 = (R_0^1)^{\mathrm{T}} I^1 R_0^1. \tag{73}$$

Indeed the matrix I^0 depends on the orientation (parameterized possibly by ϕ, θ and ϕ). To conclude, which common coordinate frame is chosen for evaluation of the rotation contribution to kinetic energy is not important; it is, however, important to use the appropriate moment of inertia. It appears more comfortable to evaluate the rotational velocities in the body coordinate frame since the moment of inertia is then constant.

We can now formally state the kinetic energy of the whole system.

$$T = \frac{1}{2}\sum_i \left(m_i \left(\mathbf{v}_{0i}^{i}\right)^{\mathrm{T}} \mathbf{v}_{0i}^{i} + \frac{1}{2}(\boldsymbol{\omega}_{0i}^{i})^{\mathrm{T}} I_i^i \boldsymbol{\omega}_{0i}^{i} \right). \tag{74}$$

For the translation velocities the chosen coordinate frame in which they are expressed plays no role. Mass is the same in all coordinate frames. Hence, again the most convenient frame is typically the body frame.

Hence we have finally a general expression for the kinetic energy. Let's state here also the general expressions for the potential energy. Here the task is simpler

$$V_i = -m_i \left(\mathbf{g}^0\right)^{\mathrm{T}} \cdot \mathbf{r}_{ci}^0, \tag{75}$$

where $\mathbf{g}^0$ is the vector of gravitational acceleration expressed in the inertial frame and $\mathbf{r}_{ci}^0$ is the radius vector for the i-the body' center of mass expressed in the inertial frame.

And this is our final stop. The kinetic and potential energies computed above yield Lagrangian whose appropriate derivatives give the desired second-order nonlinear differential equations of motion. The full code in Matlab is listed below

```
% Determining equation motion of a two-bar pendulum using rotation matrices
syms m l g t theta(t) phi(t);

dthetadt(t) = diff(theta(t),t);
dphidt(t) = diff(phi(t),t);

%% Rotation matrices, moments of inertia, angular rates
% the notation R01 = rotation from 1 to 0
R01 = [cos(theta(t)) -sin(theta(t)) 0; sin(theta(t)) cos(theta(t)) 0; 0 0 1];
R12 = [cos(phi(t)) -sin(phi(t)) 0; sin(phi(t)) cos(phi(t)) 0; 0 0 1];
R02 = R01*R12;

syms Ixx Iyy Izz
I = [Ixx 0     0;
     0   Iyy   0;
     0   0     Izz];          %Both bars have the same moment of inertia
                              %(with respect to the center of mass).

omega011 = [0; 0; dthetadt(t)];   %Notation: the last index is the frame
                              %in which the vector is expressed.
omega122 = [0; 0; dphidt(t)];     %Note that the rotation with respect to the
                              %previous bar is around the z axis in both
                              %cases.
omega010 = R01*omega011;
omega012 = R12.'*omega011;

omega020 = omega010 + R02*omega122;
omega022 = omega012 + omega122;

T1rot = 1/2*omega011.'*I*omega011;
T2rot = 1/2*omega022.'*I*omega022;
Trot = T1rot + T2rot;

%% Translation velocities of the centers of mass
r011 = [l;0;0];        %Note that the x axes are aligned with the long axes
                       %of the two bars.
r122 = [l;0;0];
```

```
rc011 = [l/2;0;0];  %Relative locations of centers of mass for
                    %determination of the potential energy.
rc122 = [l/2;0;0];
rc010 = R01*rc011;
rc022 = R12.'*r011+rc122;
rc020 = R02*rc022;

v011 = cross(omega011,r011);
v022 = R12.'*v011 + cross(omega022,r122);
vc011 = cross(omega011,rc011);
vc022 = R12.'*v011 + cross(omega022,rc122);

T1trans = 1/2*m*vc011.'*vc011;
T2trans = 1/2*m*vc022.'*vc022;
Ttrans = T1trans + T2trans;

T = Ttrans+Trot;

%% Potential velocity and Lagrangian
V = -m*[g,0,0]*rc010 -m*[g,0,0]*rc020;
L = T - V;

%% Calculating all the needed derivatives
%first convert the symbolic function into symbolic expression
L = subs(L,{dthetadt,dphidt,theta,phi},{'dthetadt','dphidt','theta','phi'});

dLdthetadt = diff(L,'dthetadt');
dLdphidt = diff(L,'dphidt');
dLdtheta = diff(L,'theta');
dLdphi = diff(L,'phi');

d2Ldthetadt2 = diff(subs(dLdthetadt,{'dthetadt','dphidt','theta','phi'},...
              {dthetadt,dphidt,theta,phi}),t);
d2Ldphidt2 = diff(subs(dLdphidt,{'dthetadt','dphidt','theta','phi'},...
              {dthetadt,dphidt,theta,phi}),t);

dLdtheta = subs(dLdtheta,{'dthetadt','dphidt','theta','phi'},{dthetadt,dphidt,theta,phi});
dLdphi = subs(dLdphi,{'dthetadt','dphidt','theta','phi'},{dthetadt,dphidt,theta,phi});

lhs1 = d2Ldthetadt2 - dLdtheta;
lhs2 = d2Ldphidt2 - dLdphi;

pretty(simplify(lhs1))
pretty(simplify(lhs2))
```

The two nonlinear differential equations of motion are

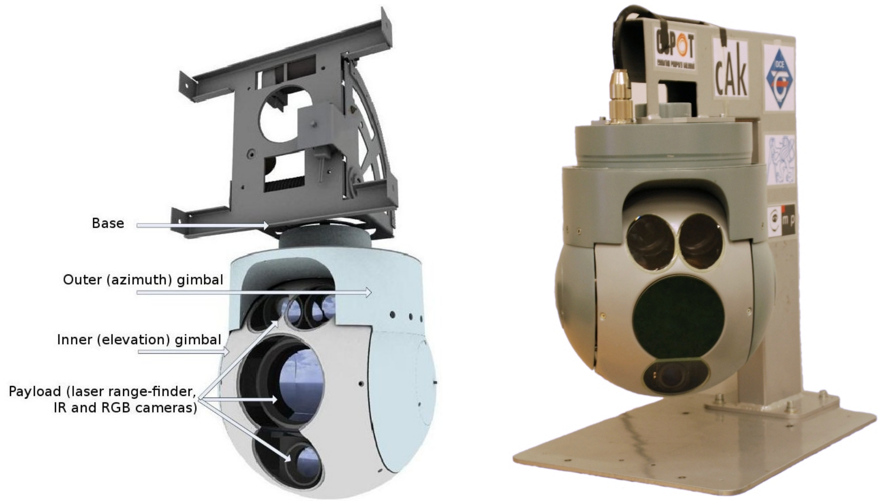
$$\left(I_{zz}+\frac{l^2m}{4}+\frac{l^2m}{2}\cos(\phi)\right)\ddot{\phi}+\left(2I_{zz}+\frac{3l^2m}{2}+l^2m\cos(\phi)\right)\ddot{\theta}-\frac{l^2m}{2}\sin(\phi)\dot{\phi}^2 \\ -l^2m\sin(\phi)\dot{\phi}\dot{\theta}+\frac{glm}{2}\sin(\phi+\theta)+\frac{3glm}{2}\sin(\theta)=0 \quad (76)$$

and

$$\left(I_{zz}+\frac{l^2m}{4}\right)\ddot{\phi}+\left(I_{zz}+\frac{l^2m}{4}+\frac{l^2m}{2}\cos(\phi)\right)\ddot{\theta}+\frac{l^2m}{2}\sin(\phi)\dot{\theta}^2+\frac{glm}{2}\sin(\phi+\theta)=0. \quad (77)$$

Modelica code is below

```
model DoublePendulumFromRotations
  parameter Real m = 1;
  parameter Real l = 1;
  parameter Real g = 9.81;
  parameter Real Izz = (m * l ^ 2) / 12;
  Real theta(start = 0, fixed = true);
  Real phi(start = 3, fixed = true);
  Real Dtheta(start = 0, fixed = true);
  Real Dphi(start = 0, fixed = true);
  annotation(experiment(StartTime = 0.0, StopTime = 10.0, Tolerance = 0.0000000001));
equation
  (Izz+l^2*m/4+l^2*m/2*cos(phi))*der(Dphi)+(2*Izz+3/2*l^2*m+l^2*m*cos(phi))*der(Dtheta)-
    (l^2*m)/2*sin(phi)*Dphi^2-l^2*m*sin(phi)*Dphi*Dtheta+g*l*m/2*sin(phi+theta) +
    3*g*l*m/2*sin(theta) = 0;
  (Izz+l^2/4*m)*der(Dphi) + (Izz+l^2/4*m+l^2/2*m*cos(phi))*der(Dtheta) +
    l^2/2*m*sin(phi)*Dtheta^2 + g*l*m/2*sin(phi+theta) = 0;
  Dtheta = der(theta);
```



(a) 3D visualization of the platform: both the azimuth and the elevation angles can rotate $n \times 360°$.

(b) Photo of the benchmark system. The supporting structure is only used in a lab.

Figure 10: Platform developed by Czech Air Force and Air Defense Technological Institute in collaboration with Czech Technical University in Prague and Essa company.

```
  Dphi = der(phi);
end DoublePendulumFromRotations;
```

4 Double-gimbal line-of-sight inertially stabilized aerial camera platform

Another example is a double-gimbal line-of-sight inertially stabilized aerial camera platform. A picture of a prototype of such system (co)developed by teams at CTU in Prague is at Fig. 10. The system consists of two motorized gimbals which rotate the optoelectronic payload (day-vision camera, night-vision camera, laser range-finder) around two axes — outer gimbal which rotates the inner gimbal about the vertical (azimuth) axis, and the inner gimbal which rotates the payload about the horizontal (elevation) axis (with respect to the outer gimbal). Some more background information can be found in a recent publication [3]. The key information from the modeling perspective is that ideally the system is expected to be perfectly mass-balanced. That means that the two axis go directly through the center of mass. Well, this is never the case, but perfect balancing can be regarded as a reasonable initial simplifying assumption.

Exploiting the assumption of perfect mass balance, the three needed coordinate systems are all centered at one point. The orientation of the coordinate system of the base is agreed to have the x-axis oriented in the direction of the nose of the carrier (aircraft), the y-axis points towards the starboard and the z-axis points down to the ground. See Fig. 11.

The sequence of the two key rotations expressing the pose of the inner gimbal (fixed to camera) with respect to the base (carrier) is visualized in Fig. 11 and for completeness it is given by

$$R_1^0 = \begin{bmatrix} \cos\psi & -\sin\psi & 0 \\ \sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{78}$$

and

$$R_2^1 = \begin{bmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{bmatrix}, \tag{79}$$

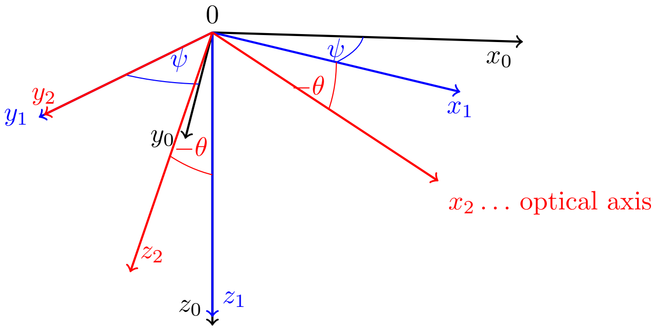


Figure 11: Composition of rotation of coordinate frames attached to the carrier (base), outer (azimuth) gimbal and inner (elevation) gimbal.

where the lower and upper indices are used here as "rotation matrix expressing the coordinate triade of the 1-frame within the 0-frame". Applying the right-hand rule, the (outer) azimuth gimbal rotates to right for the positive angle ψ and the (inner) elevation gimbal rotates up for a positive increment in the θ angle. Using the common shorthand notation like $c_\psi = \cos\psi$, the composition of the two rotations is given by the matrix product

$$R_2^0 = \begin{bmatrix} c_\psi c_\theta & -s_\psi & -c_\psi s_\theta \\ s_\psi c_\theta & c_\psi & -s_\psi s_\theta \\ -s_\theta & 0 & -c_\theta \end{bmatrix}. \tag{80}$$

As a consequence of perfect static mass balance, both the translation contribution to kinetic energy and the potential energy are zero in this case. Hence the only phenomenon in which the energy is accumulated is the rotation motion of the two gimbals:

$$L = T - V = T_{\text{rot }1} + T_{\text{rot }1} = \frac{1}{2}(\omega_{01}^1)^{\mathrm{T}} I_1^1 \omega_{01}^1 + \frac{1}{2}(\omega_{02}^2)^{\mathrm{T}} I_2^2 \omega_{02}^2. \tag{81}$$

In order to evaluate the Lagrangian successfully, the angular velocities need to be determined. The inertial angular velocity of the first (the outer) frame is

$$\omega_{01}^1 = \begin{bmatrix} 0 \\ 0 \\ \dot\psi(t) \end{bmatrix} \tag{82}$$

and it can be also expressed in the 2-coordinate frame

$$\omega_{01}^2 = (R_2^1)^{\mathrm{T}} \begin{bmatrix} 0 \\ 0 \\ \dot\psi(t) \end{bmatrix}, \tag{83}$$

which we will need shortly.

The inertial angular velocity of the second frame is

$$\omega_{02}^2 = (R_2^1)^{\mathrm{T}} \begin{bmatrix} 0 \\ 0 \\ \dot\psi(t) \end{bmatrix} + \begin{bmatrix} 0 \\ \dot\theta(t) \\ 0 \end{bmatrix}. \tag{84}$$

Now the velocities only need to be substituted into the expression for the Lagrangian and then the Lagrangian must be differentiated with respect to the needed variables in order to arrive at the two equations of motion. All the computations are implemented in the following Matlab code

```
syms t theta(t) psi(t) I1xx I1yy I1zz I2xx I2yy I2zz;

I1 = [I1xx 0    0;
      0    I1yy 0;
      0    0    I1zz];
```

```
I2 = [I2xx 0     0;
      0    I2yy  0;
      0    0     I2zz];

dthetadt(t) = diff(theta(t),t);
dpsidt(t) = diff(psi(t),t);

d2thetadt2(t) = diff(dthetadt(t),t);
d2psidt2(t) = diff(dpsidt(t),t);

R01 = [cos(psi(t))     -sin(psi(t))     0;        % notation R01 = rotation from 1 to 0
       sin(psi(t))      cos(psi(t))     0;
       0                0               1 ];

R12 = [cos(theta(t))    0               sin(theta(t));
       0                1               0;
       -sin(theta(t))   0               cos(theta(t))];

R02 = R01*R12;

omega011 = [0; 0; dpsidt(t)];             % notation: the last index is the frame
omega122 = [0; dthetadt(t); 0];           %           in which the vector is expressed

omega010 = R01*omega011;
omega012 = R12.'*omega011;

omega020 = omega010 + R02*omega122;
omega022 = omega012 + omega122;

T1rot = 1/2*omega011.'*I1*omega011;
T2rot = 1/2*omega022.'*I2*omega022;
T = T1rot + T2rot;

V = 0;

L = T-V;

%% Calculating the derivatives, first convert the symbolic function into symbolic expression
L = subs(L,{dthetadt,dpsidt,theta,psi},{'dthetadt','dpsidt','theta','psi'});

dLdthetadt = diff(L,'dthetadt');
dLdpsidt = diff(L,'dpsidt');
dLdtheta = diff(L,'theta');
dLdpsi = diff(L,'psi');

d2Ldthetadt2 = diff(subs(dLdthetadt,{'dthetadt','dpsidt','theta','psi'},...
                {dthetadt,dpsidt,theta,psi}),t);
d2Ldpsidt2 = diff(subs(dLdpsidt,{'dthetadt','dpsidt','theta','psi'},...
                {dthetadt,dpsidt,theta,psi}),t);

dLdtheta = subs(dLdtheta,{'dthetadt','dpsidt','theta','psi'},{dthetadt,dpsidt,theta,psi});
dLdpsi = subs(dLdpsi,{'dthetadt','dpsidt','theta','psi'},{dthetadt,dpsidt,theta,psi});

lhs1 = d2Ldthetadt2 - dLdtheta;
lhs2 = d2Ldpsidt2 - dLdpsi;

pretty(simplify(lhs1))
pretty(simplify(lhs2))
```

The above code produces the left-hand sides of the motion equations. Considering the right hand sides formed by the frictional and the motor torques, the full equations of motion are

$$\left(I_{1zz} + I_{2zz}\cos^2(\theta) + I_{2xx}\sin^2(\theta)\right)\ddot{\psi} + 2(I_{2xx} - I_{2zz})\cos(\theta)\sin(\theta)\dot{\theta}\dot{\psi} = -T_{\text{az. frict.}}(\dot{\psi}) + T_{\text{az. motor}}, \tag{85}$$

$$I_{2yy}\ddot{\theta} - (I_{2xx} - I_{2zz})\sin(\theta)\cos(\theta)(\dot{\psi})^2 = -T_{\text{el. frict.}}(\dot{\theta}) + T_{\text{el. motor}}. \tag{86}$$

The state-space equations are

$$\ddot{\psi} = -\frac{2(I_{2xx} - I_{2zz})}{I_{1zz} + I_{2zz}\cos^2(\theta) + I_{2xx}\sin^2(\theta)}\cos(\theta)\sin(\theta)\dot{\theta}\dot{\psi} - T_{\text{az. frict.}}(\dot{\psi}) + T_{\text{az. motor}}, \tag{87}$$

$$\ddot{\theta} = \frac{I_{2xx} - I_{2zz}}{I_{2yy}}\sin(\theta)\cos(\theta)(\dot{\psi})^2 - T_{\text{el. frict.}}(\dot{\theta}) + T_{\text{el. motor}}. \tag{88}$$

For convenience the Matlab file implementing these equations for the purpose of numerical simulation is here. The parameters of the moments of inertia are *roughly corresponding* to the real values which were obtained from 3D CAD modeling. Friction is modeled here just as a viscous friction. This turns out insufficient in this particular project but let's skip this very specialized modeling issue in favor of simplicity.

```
function dydt = double_gimbal_state_space_model(t,y)
psi = y(1);
theta = y(2);
dpsidt = y(3);
dthetadt = y(4);

I1zz = 1e-3;             % [kg m2]
I2xx = 10e-3;
I2yy = 9e-3;
I2zz = 5e-3;

kpsi = 0.01;             % viscous friction coefficient [Nms/rad]
ktheta = 0.01;

dydt = zeros(4,1);
dydt(1) = dpsidt;
dydt(2) = dthetadt;
dydt(3) = -2*(I2xx-I2yy)/(I1zz+I2zz*(cos(theta))^2 +...
          I2xx*(sin(theta))^2)*cos(theta)*sin(theta)*dpsidt*dthetadt - kpsi*dpsidt;
dydt(4) = (I2xx-I2zz)/I2yy*sin(theta)*cos(theta)*dpsidt^2 - ktheta*dthetadt;
```

The simulation response at Fig. 12 is obtained using the code

```
[t,y] = ode45(@double_gimbal_state_space_model,[0,10],[0,-0.9*pi/2,2*pi,0]);
```

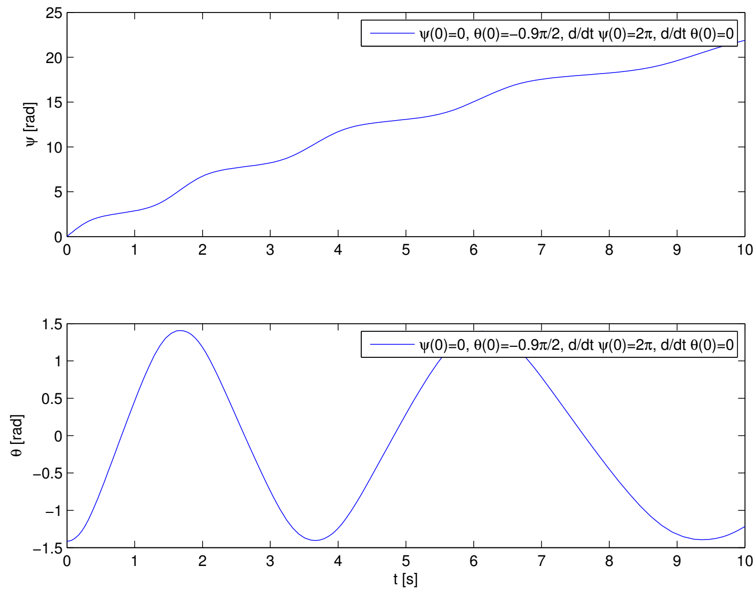


Figure 12: Simulation response of the double-gimbal model to some nonzero initial conditions: the camera pointing nearly towards the ground and the azimuth axis rotating; the centrifugal effect visible.

References

[1] Donald T. Greenwood. *Classical Dynamics*. Dover Publications, July 1997.

[2] Donald T. Greenwood. *Advanced Dynamics*. Cambridge University Press, November 2006.

[3] Z. Hurak and M. Rezac. Image-based pointing and tracking for inertially stabilized airborne camera platform. *IEEE Transactions on Control Systems Technology*, 20(5):1146 –1159, September 2012.