This handout will assist you in implementing the numerical part of Lab 11. We are going to analyze a rotating system whose restoring force is generated via two springs.
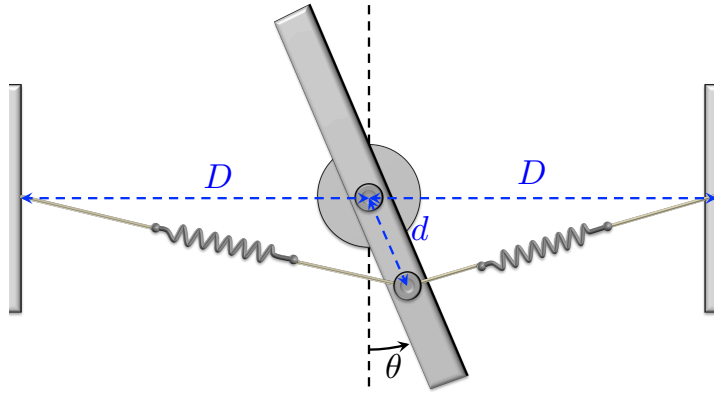

Figure 1: For Lab 11

The following steps are a recommendation for how to program the MATLAB function `alpha.m` that will be the input to the `evolve()` function. You will then be able to evolve the function and make predictions. In homework this week, we will use this function to make predictions that we will verify in Lab 12 next week.

1. Define the header of your function.

```
function[ alpha_z ] = alpha(t, theta, theta_dot)
% angular acceleration for non-linear pendulum in Lab 11, PS12a
% RSK and RMH, 3/15/13
```

2. Defined the physical parameters that you measured.

```
g = 9.8; % m/s^2

% physical parameters
D =          % m, distance between piviot and outside atachment point of springs
l0=          % m, unstretched length of spring
d =          % m, distance from pivot to the inside attachment point for springs.
L =          % m, length of metal bar
l =          % m, distance from pivot to attached mass
M =          % kg, mass of bar
m =          % kg, mass of metal disk
r =          % m, radius of metal disk

% rotational inertia
I_bar =
I_disk =
I_tot = I_bar + I_disk;

% damping coefficient from logger pro-fit
beta = 0.078;     % 1/s
b = 2*I_tot*beta;  % kg m^2/s =
```

```
% calculating k,
w =            % small angle oscillation frequency
k =            % spring constant
```

3. There are two spring forces, and computing the force is a bit tricky. Let's start with the left one. Consider the vectors shown in figure 2.
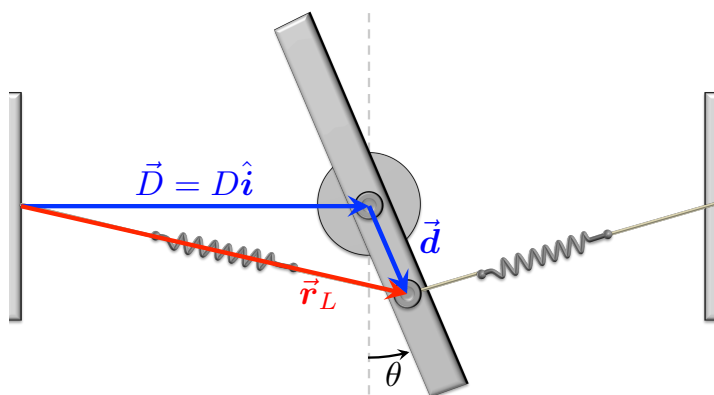

Figure 2: Find the tension in the left spring. Its length is given by $|\vec{r}_L|$. When not stretched, the spring and the attacked strings have length ℓ_0.

- Find an expression for $\vec{D}$ and $\vec{d}$ in terms of D, d, and θ. In MATLAB use 3-dim column vectors when inputing these into the function. You might structure your code as follows.

```
%% Left side spring

% left spring vector
D_vec = [D; 0; 0];
d_vec =
rL = D_vec + d_vec;
```

- The vector sum $\vec{r}_L = \vec{D} + \vec{d}$ points along the spring. Find expressions for the force the left spring exerts on the rod $\vec{F}_L$ in terms of $\vec{r}_L$, the spring constant k, and the equilibrium distance for the spring ℓ_0.

```
% compute magnitude of and unit vector of spring vector
rL_mag =
rL_hat = rL/rL_mag;

% compute the force of the left spring
F_sL =

% compute the torque
Torque_sL = cross(d_vec, F_sL);
```

- Instead of computing the cross product by hand, use the `cross()` function to compute the torque.

$$\vec{\tau}_L = \vec{d} \times \vec{F}_L \tag{1}$$

```
% compute the torque
Torque_sL = cross(d_vec, F_sL);
```

4. Repeat this analysis of part (c) for right spring.

5. Compute the torque due to the drag force using the drag coefficient b you determined by fitting the small angle, damped, data.

$$\tau_z = -b\dot{\theta} = -b\frac{d\theta}{dt}$$

```
%% air drag term
Torque_drag = [ 0; 0; -b*theta_dot];
```

6. After finding each torque, and after finding the moment of inertia, you can then use the 2nd law of rotation to find the angular acceleration α_z.

$$(\boldsymbol{\tau}_{\text{net}})_z = \sum \tau_z = I\alpha_z \tag{2}$$

We will only need the z-component of $\vec{\alpha}$, so you might use:

```
%% compute alpha

Torque_tot = Torque_sL + Torque_sR + Torque_drag;
alpha_z = Torque_tot(3)/I_tot;
```

7. Finally, for each of the following values for θ and $\dot{\theta}$, compute α_z and verify that you have reasonable numbers. You can check against the numbers the teaching staff generated shown below. You numbers should be close, but not exactly equal to these.

$$
\begin{aligned}
(\theta, \dot{\theta}) &= (0.2\,,\text{rad}, 0.4\,,\text{rad/s}) & \alpha_z &= -4.8765 \text{ rad/s}^2 \\
(\theta, \dot{\theta}) &= (0\,,\text{rad}, -0.5\,,\text{rad/s}) & \alpha_z &= 0.0780 \text{ rad/s}^2 \\
(\theta, \dot{\theta}) &= (-0.3\,,\text{rad}, 0\,,\text{rad/s}) & \alpha_z &= 6.9886 \text{ rad/s}^2
\end{aligned}
\tag{3}
$$