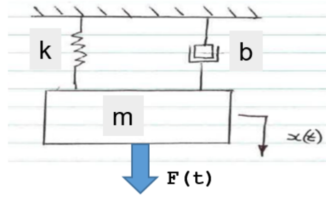
Explore Spring Mass Damper equations of motion:



From our year 1 class in physics and mechanics, we derived using Newton's 2nd law, the equation of motion for the dynamics of a Spring Mass damper system. Recall that it had the following form:

$$m.\ddot{x} + b.\dot{x} + k.x = F(t)$$

Today we'll use the Lagrange approach to derive the same equations of motion for our spring mass damper. Recall our earlier class where we derived and summarised the Lagrangian equations:

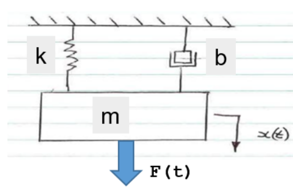
$$\frac{d}{dt}\frac{\partial L}{\partial \dot{q}_k} - \frac{\partial L}{\partial q_k} = Q_k \text{ where } Q_k = \sum_{i=1}^{Nf_{nc}}\left(\vec{F}_i \cdot \frac{\partial \vec{v}_i}{\partial \dot{q}_k}\right) + \sum_{j=1}^{N\tau_{nc}}\left(\vec{\tau}_j \cdot \frac{\partial \vec{\omega}_i}{\partial \dot{q}_k}\right)$$

where:

- L : is the system Lagrangian, ie: L = KE - PE
- q_k : is the k^{th} generalised co-ordinate
- Q_k : is the generalised force associated with the k^{th} generalised co-ordinate q_k
- Nf_{nc} : is the number of active NON conservative forces
- $N\tau_{nc}$: is the number of active NON conservative TORQUES
- $\vec{v}_i$: is the velocity vector of the point associated with the applied force.
- $\vec{\omega}_i$: is the angular velocity about the point associated with the applied torque.

Bradley Horton : 13-Sep-2016, bradley.horton@mathworks.com.au

STEP_1: Define Model parameters



Define some Symbolic variables that parameterise our model:

```
syms  m  k  b  F
```

And here are some variables associatd with our $x(t)$, $\dot{x}(t)$ and $\ddot{x}(t)$

```
syms          t        x(t)
```

```
syms                THE_X     THE_XD       THE_XDD
HOLDER_list = [    THE_X,    THE_XD,      THE_XDD];
actual_list = [        x,    diff(x,t),   diff(x,t,2)];
```

STEP_2: Understanding of governing physics Interesting Part

```
v  = diff(x,t);   % velocity
KE = 0.5*m*v^2;   % KINETIC energy
PE = 0.5*k*x^2;   % POTENTIAL energy
L  = KE - PE      % our Lagrangian
```

L(t) =

$$\frac{m\left(\frac{\partial}{\partial t}x(t)\right)^2}{2}-\frac{k\,x(t)^2}{2}$$

STEP_3a: Apply Lagrange's equation - PART 1 of 3 Could be Automated

Now let's start applying Lagranges equation $\frac{d}{dt}\frac{\partial L}{\partial \dot{x}}-\frac{\partial L}{\partial x}$:

```
                         % OLD_LIST        NEW_LIST
L_new   = subs(L, actual_list,   HOLDER_list);
```

Our 1st piece is: $\frac{\partial L}{\partial x}$

```
dLdx    = diff(L_new, THE_X);
```

Our 2nd piece is: $\frac{\partial L}{\partial \dot{x}}$

```
dLdxdot =  diff(L_new, THE_XD);
```

Our 3rd piece is: $\frac{d}{dt}\frac{\partial L}{\partial \dot{x}}$

```
                                     % OLD_LIST      NEW_LIST
dLdxdot       =  subs(dLdxdot, HOLDER_list,  actual_list );
dt_of_dLdxdot = diff(dLdxdot, t);
```

Now put it all together: $\frac{d}{dt}\frac{\partial L}{\partial \dot{x}}-\frac{\partial L}{\partial x}$

```
our_EOM_LHS = dt_of_dLdxdot - dLdx;
our_EOM_LHS =  subs(our_EOM_LHS, HOLDER_list, actual_list )
```

our_EOM_LHS(t) =

$$m \frac{\partial^2}{\partial t^2} x(t) + k\, x(t)$$

STEP_3b: Apply Lagrange's equation - PART 2 of 3 Could be Automated

Now calculate the generalised force Q :

$$Q_k = \sum_{i=1}^{Nf_{nc}} \left(\vec{F_i} \cdot \frac{\partial \vec{v}_i}{\partial \dot{q}_k} \right) + \sum_{j=1}^{N\tau_{nc}} \left(\vec{\tau_j} \cdot \frac{\partial \vec{\omega}_j}{\partial \dot{q}_k} \right)$$

Define Forces and velocities:

```
Fv_mat = [  F,   (-b*THE_XD),   THE_XD,   THE_XD;
            0,             0,        0,        0;
            0,             0,        0,        0;
        ];
F_mat = Fv_mat(:,1:2);
v_mat = Fv_mat(:,3:4);
```

Calculate the GENERALISED forces Q_k:

```
Q     = 0;
for zz=1:2
    F_vec = F_mat(:,zz);
    v_vec = v_mat(:,zz);

    dvdq  = diff(v_vec, THE_XD);
    Q     = Q + sum( F_vec .* dvdq);
end

our_EOM_RHS = Q;
```

STEP_3c: Apply Lagrange's equation - PART 3 of 3

Now put it all together: $\frac{d}{dt}\frac{\partial L}{\partial \dot{x}} - \frac{\partial L}{\partial x} = Q$

```
our_EOM     = (our_EOM_LHS == our_EOM_RHS);
our_EOM     =  subs(our_EOM, HOLDER_list, actual_list )
```

our_EOM(t) =

$$m \frac{\partial^2}{\partial t^2} x(t) + k\, x(t) = F - b \frac{\partial}{\partial t} x(t)$$

STEP_4: Isolate the term of interest $\ddot{x}$

In addition to solving for $\ddot{x}$, we'll show the resulting expression using the "alternate" symbol list:

```
                                          % OLD_LIST      NEW_LIST
our_EOM                 =  subs(our_EOM, actual_list,  HOLDER_list);
```

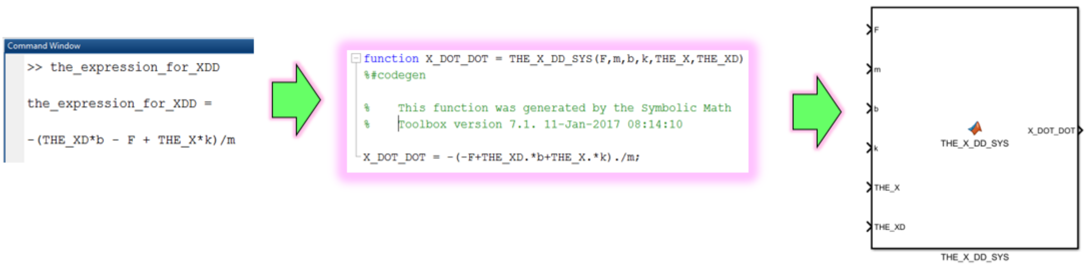
Come on ... what's $\ddot{x}$?

```
the_expression_for_XDD = solve(our_EOM, THE_XDD)
```

the_expression_for_XDD =

$$-\frac{\mathrm{THE}_{\mathrm{XD}}\, b - F + \mathrm{THE}_X\, k}{m}$$

STEP_5: Convert symbolic expression into a block diagram model




```
MODEL_NAME        = 'SIM_SMD_WILL_BE_DELETED';
close_system(MODEL_NAME,0);   new_system(MODEL_NAME);
 open_system(MODEL_NAME)
```

Automatically convert our $\ddot{x}$ expression into s Simulink block:

```
matlabFunctionBlock( [MODEL_NAME,'/THE_X_DD_SYS'], the_expression_for_XDD, ...
                         'Vars',      {F, m,b,k,THE_X,THE_XD}, ...
                         'Outputs',  {'X_DOT_DOT'}    );
```

STEP_6: Simulate model

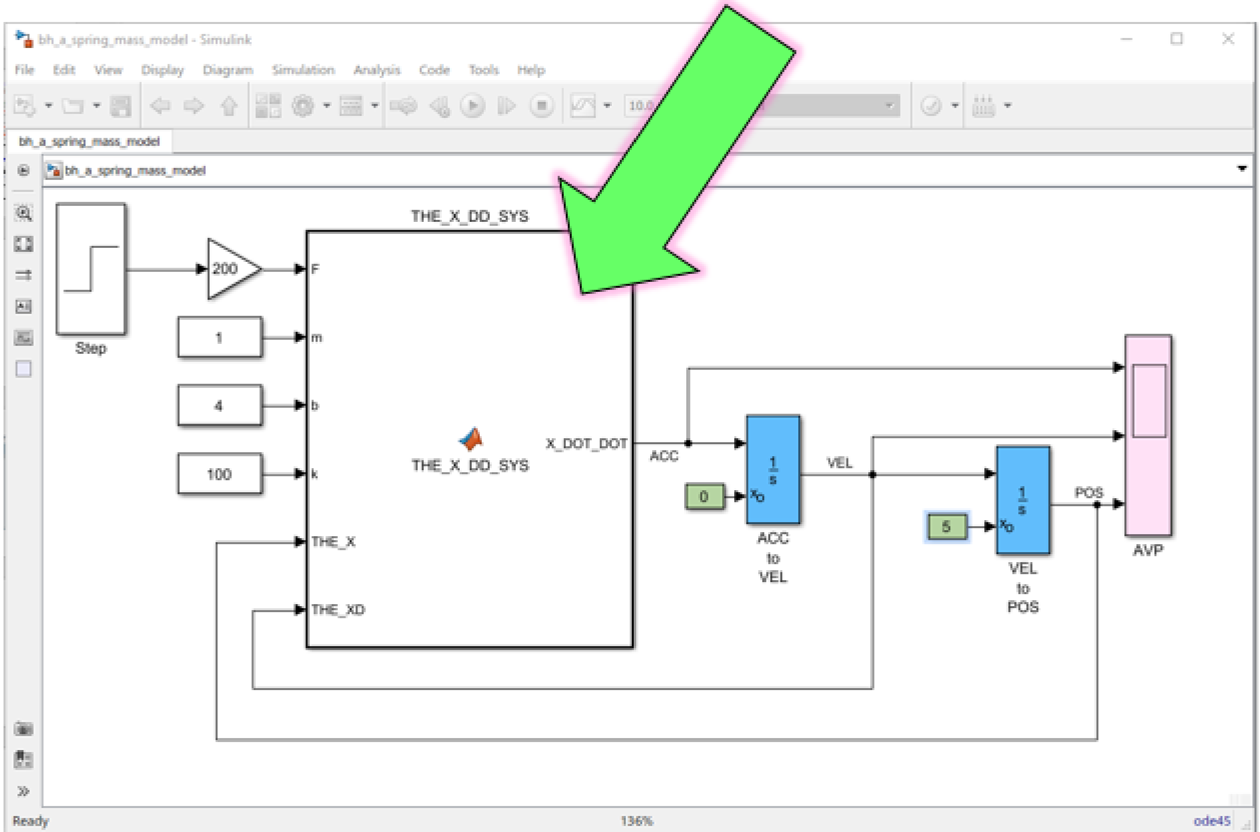
Let's use the model that we just derived, and implement it in Simulink - where we'll numerically solve it. The parameters that we'll use for this Numerical simulation are:

$$\ddot{x}(t) + 4.\dot{x}(t) + 100.x(t) = 200.u(t-5)$$

with
- $x(0) = 5$
- $\dot{x}(0) = 0$

Have a look at our Simulink model .... and NOTE how we use the integrator blocks to integrate: $\ddot{x} \rightarrow \frac{1}{s} \rightarrow \dot{x} \rightarrow \frac{1}{s} \rightarrow x$

```
open_system('bh_a_spring_mass_model')
```



How does this help me make a Robot write ***Hello*** ?

So *IFFFF* we understand the system physics we can scale this Computational thinking approach to bigger and more interesting systems .... like 4-LINK robotic manipulators. Capabilities that allow us to scale, include:

- **`diff()`**
- **`matlabFunctionBlock()`**

And these partner with the capabilities that allow us to explore and design:

- Simulink
- Apps for Control system design