1.5 Basic simulation in the simscape framework

A basic matlab script to run the simulation is presented below in listing 1. In this script:

1. The user sets up some simulation parameters (like sampling rate and controller gains),
2. creates the controller and simulation handler objects,
3. activates the custom-made simulations (`s.customSimulationsSwitch(true)`),
4. sets up the task they want to simulate, creates the corresponding events, and
5. runs the simscape simulation

```
%% Simulation paramters: ------------------------
Tsampling = 1e-3;

%set K for class implementation:
Kp = [10;10;10];
Kd = [1;1;1];
Ki = 0*[1;1;1];
K = [Kp,Kd,Ki];

%%  Set up simulation: --------------------------
r= legRobot(tmax=1.007344*[7;10;10]);
qinitial=[0;0;0]; %set initial conditions for simscape

PC = PosControl(r,Tsampling,K); %Position Controller
TC = TrajControl(r,Tsampling,K); %Trajectory Controller
EC = TrajControl(r,Tsampling); %Trajectory Controller

% simulation_preparation
s = simulation_preparation(r,PC,TC,EC);
s.setInitialConditions([0;0;0;0;0;0],0);
s.customSimulationsSwitch(true); % false -> do not run custom sim

%%  set up tasks: -------------------------------
PC.setTarget([0.1667;0.3;0.5]);
s.setEvent(0,1,1);

%script to set up a 3xN vector x, and a 1xN vector tw. N = #
    waypoints
testXpos
TC.generateQ(x,tw,1);
s.setEvent(1,2,tw(end));

PC.setTarget([0.1667;0.3;0.5]);
s.setEvent(tw(end)+1,1,1);

%% Run simscape simulation:
[Select_Controller,Qpos,TSqd,TSqtd,TSh] = s.SimScapeSimulation();
out = sim('Leg_Simulation_Framework_test_2.slx',[0 s.t_total]);
```

Listing 1: Basic script to set up and run the matlab/simscape simulations

1.5.1 Framework available commands

For each task the user wants to simulate, the user uses the controller methods to set the task parameters and then they create an event using the command [7]:

```
s.setEvent(t_start,Controller,dt);
```

where `t_start` is the start time of the event, `Controller` specifies the controller that is activated during the task and `dt` is the duration of the task. Regarding the controllers, the following correspondence is true:

1. → Position Controller
2. → Trajectory Controller
3. → Effort Controller

Important: To run the custom-made simulations, the `customSimulationsSwitch` method must be called, passing `true`. This method must be called before setting any event!

The user can currently simulate the following tasks:

- Set a joint-angle target for the position controller:

```
PC.setAngleTarget(Qdesired)
```

  where `Qdesired` is a 3×1 column vector with the desired joint angles.

- Set a Cartesian target for the position controller:

```
PC.setTarget(Xdesired)
```

  where `Xdesired` is a 3×1 column vector with the desired position in the world frame.

- Set a custom trajectory with N waypoints for the trajectory controller:

```
TC.generateQ(Xwp,Twp,Tstart)
```

  where `Xwp` is a $3 \times N$ column vector with the desired positions of the waypoints in the world frame, `Twp` is the timestamps for each of the waypoints and `Tstart` is the starting time of the trajectory.

- Set an ellipse in the YZ plane for the trajectory controller:

```
TC.generateEllipse(a,b,DX,Dth,T,Npoints,Tstart)
```

  where `a`, `b`, `DX`, `dth` are ellipse parameters (`DX` is a 2×1 vector specifying the center of the ellipse in the YZ plane $DX = [DY, DZ]^T$), `T` is the period of the trajectory and `Tstart` is the starting time of the trajectory. The ellipse parameterization is the following:

$$r(\theta) = \frac{ab}{\sqrt{a^2 sin^2(\theta - d\theta) + b^2 cos^2(\theta - d\theta)}}$$

$${}^W P = r(\theta) \begin{bmatrix} 0 \\ cos(\theta) \\ sin(\theta) \end{bmatrix} + \begin{bmatrix} LB0 + L12 \\ DY \\ DZ \end{bmatrix}$$

- Set a effort target for the effort controller:

```
EC.setWrench(Hdesired)
```

  where `Hdesired` is a 3×1 column vector with the desired force vector in the world frame.

The user has access to the robot states and actuator inputs from the custom simulation using this command:

```
[t,y,u] = s.getCustomSimData();
```

[7] In this example `s` is the `simulation_preparation` object. In the following examples `PC`, `TC` are the position and trajectory controller objects respectively.