5.2 Design the controller using the discrete LQR technique

1. Start from a sampling frequency of 200Hz, and compute (Ad, Bd, Cd, Dd) as in Section 5.1
2. compute the LQR gain Kd minimizing the cost (24) by using the Matlab function `dlqr`;
3. LabB_ControllerOverSimulator_Discrete_Paramaters.m, insert Kd and simulate the corresponding simulator.
4. If the controller still stabilizes the simulator, reduce the sampling frequency and start over, up to the moment for which the simulated robot falls

```
fSamplingPeriod = 0.005;
[Ad,Bd,Cd,Dd] = getDiscreteStateSpace(fSamplingPeriod)
```

```
Ad = 4x4
   1.000000000000000   0.002169375804939  -0.000035615030253   0.000059371727425
                   0   0.169257508881752  -0.009548994253862   0.017409977283237
                   0   0.012386212728786   1.000597546217141   0.004740938014341
                   0   3.635660710145991   0.218485145882911   0.924248671303915

Bd = 4x2
   0.000133880874091   0.000014993384476
   0.039291874579947   0.004687647461305
  -0.000585834385821   0.000074871411322
  -0.171956925479762   0.035681147521597

Cd = 2x4
     1     0     0     0
     0     0     1     0

Dd = 2x2
     0     0
     0     0
```

```
%Paramaters for dlqr
R = 1;N=0;
Cweight = [20 5 10 1];   %Our previous weighting vector
rho = 20;                %And previous rho
```

```
W = rho*Cweight'*Cweight %Corresponding weight matrix
```

```
W = 4x4
1.0e+02 *

   8.500000000000000   0.850000000000000   8.500000000000000   1.700000000000000
   0.850000000000000   0.085000000000000   0.850000000000000   0.170000000000000
   8.500000000000000   0.850000000000000   8.500000000000000   1.700000000000000
   1.700000000000000   0.170000000000000   1.700000000000000   0.340000000000000
```

```
[Kd,~,CLP] = dlqr(Ad,Bd(:,1),W,R);
Kd                      %Discrete controller gain matrix, %We get basically the same Kd
```

```
Kd = 1x4
 -21.198476619388558 -42.156663040993337 -60.653271802144033  -9.978116709025945
```

```
CLP                     %Closed loop poles
```

```
CLP = 4x1 complex
  0.987713294823011 + 0.001877731796920i
  0.987713294823011 - 0.001877731796920i
  0.974888347586292 + 0.000000000000000i
  0.051702133068815 + 0.000000000000000i
```

```
```