**Inverted pendulum**

# System model

Consider the system depicted in Figure 1 composed by a pendulum fixed on a cart and controlled through a DC motor.



The model parameters are given as *M = 10 kg, m = 2kg, L = 1m, Rr = 0.1m, Ra = 10Ω,   
Kc = 2V s/rad, Ke = 2Nm/A*. The goal of this work is to stabilize the pendulum in its vertical position (around *ϑ = 0*) acting on the motor voltage *v*.

To derive the dynamical model the system can be decomposed as in Figure2.



* Cart:
* Pendulum:

Rearranging the terms to remove the force H we obtain

and substituting the terms and in the first equation we have the following model

To use a state-space representation define the variables

# Equilibrium and linearization

We are now interested in linearizing the system model around the vertical pendulum position with zero input, i.e. and the resulting linearized model is the following

Substituting the parameters values the linearized system model can be written as

In matlab there are several ways of obtaining the linearized model without making all the computations by hand. **One possible solution is to construct a Simulink model of the system, put it at the equilibrium and use the ‘time-based linearization’ block.** A second way is based on the symbolic toolbox and allows one to compute the analytical derivatives and then to substitute the value of the parameters. To do that, firstly define all the variables and parameters as symbolic using the command ‘syms’ and then use the command ‘diff’ to compute the derivatives with respect to all the state and input variables. At this point substitute the values at the equilibrium and the parameters using the command ‘eval’.

# Controller design

Given the nonlinear model and its linearization the goal is to compute a controller that stabilizes the pendulum in the vertical position without incurring in large oscillations of the cart. Try to do the following:

* Run the initialization file “pendulum\_sys\_init”. Then open the Simulink file “pendulum\_openloop\_anim” and linearize the system around the vertical position equilibrium.
  + *How? You must set the initial conditions and the input at the equilibrium point. Then, linearize the system using time-based linearization block on Simulink. Remember that you must set which are the inputs and the outputs using the ports “In” and “Out” blocks respectively.*
* Assume firstly that the state is fully accessible. After checking is the system is reachable, design a proportional controller using pole placement.
  + *How? To compute the reachability matrix use the matlab function “ctrb(A,B)”. To define the pole placement gain you can use the matlab function “place”.   
    Defining as P the position of the desired poles, it becomes K\_pp = place(A, B, P).  
    (e.g. P=[ -2, -2.5, -3, -4]).*
* Enlarge the system with an integrator on the cart position and design a proportional controller with pole placement (e.g. P=[ -2, -2.5, -3, -4 -10]). Test its performances with respect to the previous case.
* Design a LQ controller on the enlarged system with the integrator.  
  (There is not a perfect choice for Q and R, a possible choice could be  
  *Q = diag([0.1, 10, 0.1, 10, 10]), R=0.001* )*.*
  + *How? To define the LQ gain you can use the matlab function “lqr”. Defining the weights Q and R, it becomes K\_LQ = lqr(A,B,Q,R).*
* Assume now that only the measurements of the position of the cart, , and the angular position of the pendulum, , are available. Design a **Kalman observer** and apply it to the previously computed controllers.
  + *How? To define the Kalman gain you can use the matlab function “lqr”. Defining the noise variance Q\_var and R\_var, it becomes L\_Klm = ( lqr(AT, CT, Q\_var ,R\_var ) )T.*