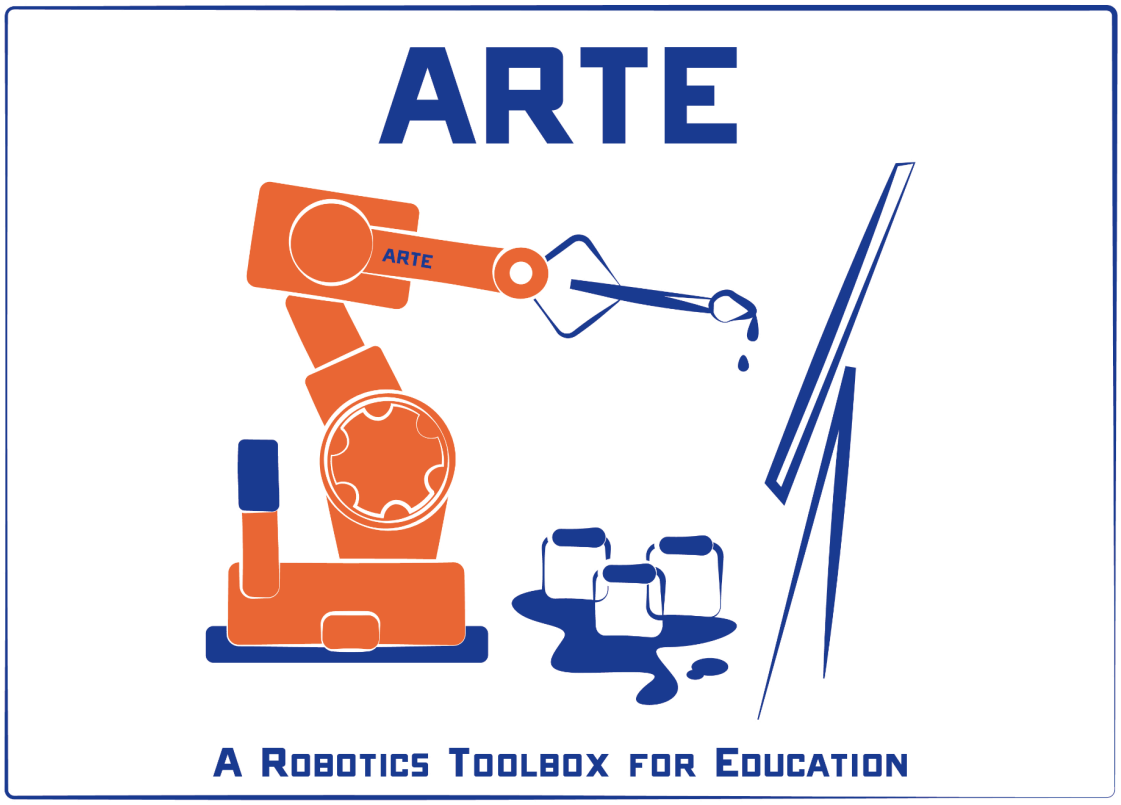


PRACTICAL SESSION 3: INVERSE DYNAMICS

Arturo Gil Aparicio

arturo.gil@umh.es

OBJECTIVES

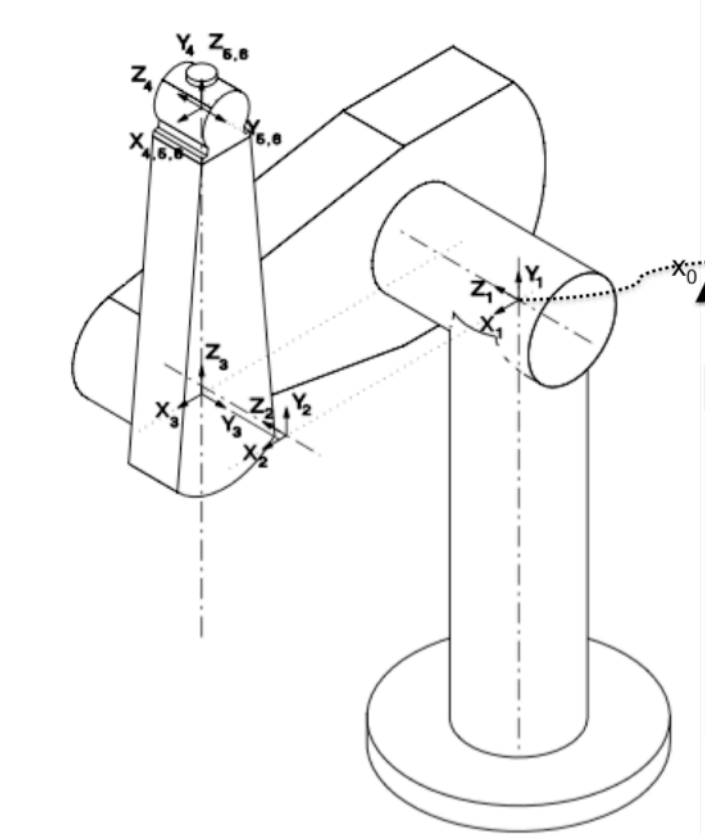
This practical session deals with the following concepts:

- Computation of the inverse dynamics of a serial manipulator by means of the recursive Newton-Euler formulation.
- Computation of the torques required by the robot at different situations. A simulation of the worst case is carried out.
- Motor selection based on trapezoidal speed profiles.

Index

1 First steps

The dynamic analysis of a robot manipulator is considered and advanced concept. In most of the cases the dynamic parameters of the robotic arm are not provided by the manufacturer. In consequence, during this session we will restrict our analysis to a very well known robotic manipulator: the UNIMATE PUMA 560, whose parameters have been previously identified. Figure 1 presents the robot and its D-H parameters table.



	Theta	D	A	Alfa
1	q_1	0	0	π/2
2	q_2	0	0.4318	0
3	q_3	0.15005	0.0203	-π/2
4	q_4	0.4318	0	π/2
5	q_5	0	0	-π/2
6	q_6	0	0	0

Figure 1

All the parameters belonging to this robot can be found under arte/robots/unimate/puma560/parameters.m. The student should now take a look at these parameters.

```
function robot = parameters()
%KYNEMATICS
robot.name= 'puma_560';

robot.DH.theta = '[q(1) q(2) q(3) q(4) q(5) q(6)]';
robot.DH.d='[0  0   0.15005  0.4318     0    0.04]';
robot.DH.a='[0 0.4318  0.0203      0       0    0]';
robot.DH.alpha= '[pi/2  0   -pi/2  pi/2   -pi/2 0]';
robot.J=[];
robot.name='Puma 560 robotic arm';

robot.inversekinematic_fn = 'inversekinematic_puma560(robot,
T)';


%R: rotational, T: translational
robot.kind=['R' 'R' 'R' 'R' 'R' 'R'];

%number of degrees of freedom
robot.DOF = 6;

%minimum and maximum rotation angle in rad
robot.maxangle =[deg2rad(-160) deg2rad(160); %Axis 1, minimum,
maximum
                 deg2rad(-110) deg2rad(110); %Axis 2, minimum,
```

```
maximum
                deg2rad(-135) deg2rad(135); %Axis 3
                deg2rad(-266) deg2rad(266); %Axis 4: Unlimited
(400ÔøΩ default)
                deg2rad(-100) deg2rad(100); %Axis 5
                deg2rad(-266) deg2rad(266)]; %Axis 6: Unlimited
(800ÔøΩ default)

%maximum absolute speed of each joint rad/s or m/s
robot.velmax = [1
                1
                1
                1
                1
                1];%not available
% end effectors maximum velocity
robot.linear_velmax = 0.5; %m/s, from datasheet
robot.accelmax=robot.velmax/0.1; % 0.1 is here an acceleration
time

%base reference system
robot.T0 = eye(4);

%INITIALIZATION OF VARIABLES REQUIRED FOR THE SIMULATION
%position, velocity and acceleration
robot=init_sim_variables(robot);
robot.path = pwd;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%
% GRAPHICS
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%
%read graphics files
robot.graphical.has_graphics=1;
robot.graphical.color = [150 180 130]./255;
%for transparency
robot.graphical.draw_transparent=0;
%draw DH systems
robot.graphical.draw_axes=1;
%DH system length and Font size, standard is 1/10. Select 2/20,
3/30 for
%bigger robots
robot.graphical.axes_scale=1;
%adjust for a default view of the robot
robot.axis=[-1 1 -1 1 -0.66 2];
robot = read_graphics(robot);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%
%DYNAMIC PARAMETERS
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%
robot.has_dynamics=1;

%consider friction in the computations
robot.dynamics.friction=0;
```

```
%link masses (kg)
robot.dynamics.masses=[0 17.4 4.8 0.82 0.34 0.09];

%COM of each link with respect to own reference system
robot.dynamics.r_com=[0       0          0; %(rx, ry, rz) link 1
    -0.3638  0.006   0.2275; %(rx, ry, rz) link 2
    -0.0203 -0.0141  0.070;  %(rx, ry, rz) link 3
    0       0.019      0;%(rx, ry, rz) link 4
    0       0          0;%(rx, ry, rz) link 5
   0         0        -0.008];%(rx, ry, rz) link 6

%Inertia matrices of each link with respect to its D-H reference
system.
% Ixx   Iyy Izz Ixy Iyz Ixz, for each row
robot.dynamics.Inertia=[0      0.35 0       0   0   0;
    0.13    0.524   0.539   0   0   0;
    0.066   0.086   0.0125  0   0   0;
    1.8e-3  1.3e-3  1.8e-3  0   0   0;
    0.3e-3  0.4e-3  0.3e-3  0   0   0;
    0.15e-3 0.15e-3 0.04e-3 0   0   0];


%Please note that we are simulating the motors as presented in
MAXON
%catalog
robot.motors=load_motors([5 5 5 5 5 5]);


%Actuator rotor inertia
%robot.motors.Inertia=[200e-6 200e-6 200e-6 33e-6 33e-6 33e-6];
%Speed reductor at each joint
%robot.motors.G=[-62.6111 107.815 -53.7063 76.0364 71.923
76.686];
%Please note that, for simplicity in control, we consider that
the gear
%ratios are all positive
robot.motors.G=[62.6111 107.815 53.7063 76.0364 71.923 76.686];
```

By calling `load_robot`, we load all these parameters into a workspace variable. The variable `robot.has_dynamics` tells us that the robot dynamics are available. You can load the PUMA 560 robot and draw it in 3D.

```
>> init_lib
>> robot=load_robot('unimate','puma560')
>> T=directkinematic(robot, [0 0 0 0 0 0])

T =

    1.0000         0         0    0.8000
         0   -1.0000   -0.0000   -0.0000
         0    0.0000   -1.0000    0.4000
         0         0         0    1.0000
>> drawrobot3d(robot, [0 0 0 0 0 0])
```

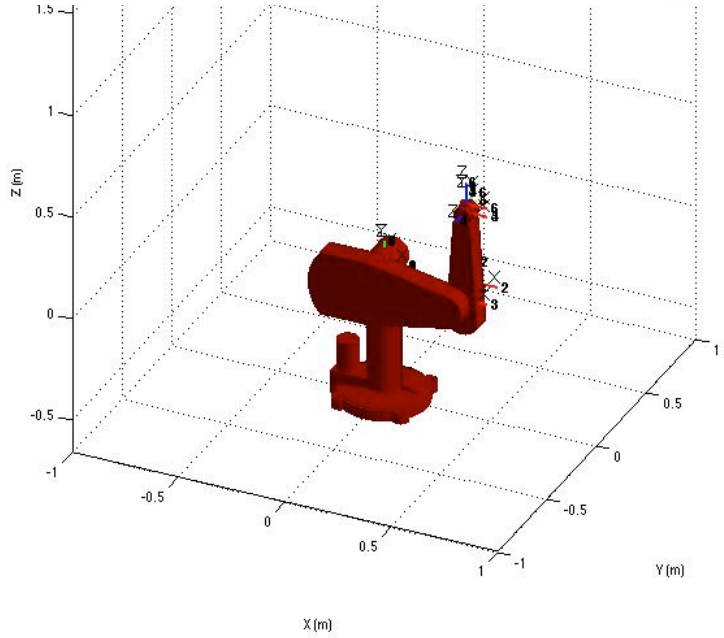


Figure 2

2 Inverse dynamics

Given the joint positions, velocities, accelerations and arm load, we can compute the torques/forces at each link to instantaneously bring the robot to this state. We can now analyse the robot under different dynamic situations.

```
>> init_lib
>> robot=load_robot('unimate','puma560')
>> q1 = [0 0 0 0 0 0];
>> T=directkinematic(robot, q1)
T =

    1.0000         0         0    0.4521
         0    1.0000         0   -0.1500
         0         0    1.0000    0.4718
         0         0         0    1.0000

>> drawrobot3d(robot, q1)
>> tau = inversedynamic(robot, q1, [0 0 0 0 0 0], [0 0 0 0 0 0], [0  0
9.81]', [0 0 0 0 0 0]')
```

The above code computes the torques `tau` when the robot is placed at `q1`, with zero speeds and accelerations, the gravity acts at its Z_0 axis and no load are applied at its end effector.

At the demos directory you should find three files:

a) `inversedynamics_2DOFplanar.m`: Performs a simulation of a 2DOF arm. The dynamic parameters in this case are simplified for the sake of an easy-to-understand simulation.

b) `inversedynamics_3DOFplanar.m`: The simulation is extended to the case of a 3 DOF planar arm.

c) `inversedynamics_puma560.m`: The file simulates the inverse dynamic model of the Unimate Puma 560 robotic arm. The arm is simulated under different conditions.

Exercise 1:

Compute the torques at each joint under the following conditions.

```
q =  [0 0 -pi/2 0 0 0]
qd =  [0 0 0 0 0 0] %zero speeds
qdd =[1 1 1 0 0 0]
```

Justify your results. Now repeat the experiments but consider that the robot carries a point centered mass with 5 Kg. Describe the differences found.

Exercise 2:

Adapt the dynamic parameters for your chosen robot. The link masses and actuator's weights are generally not specified by the manufacturers. In consequence, the link masses should be imagined considering the total weight of the robot. Simulate your robot under different conditions.

3 Motor selection

Open the file named motor_selection.m under `arte/demos`. This file simulates the inverse dynamics of the robot under a set of practical situations. The approach considers that the robot has a joint configuration that makes the torques at each actuator maximum. In this sense, the inverse dynamic function computes the torques at each joint that would bring the robot to the specifed motion state (that is, joint positions, joint speeds and accelerations). The student should consider the arm and place it at a joint position where the torque requirements are higher. For the case of the Puma 560, this configuration is presented in Figure 3.

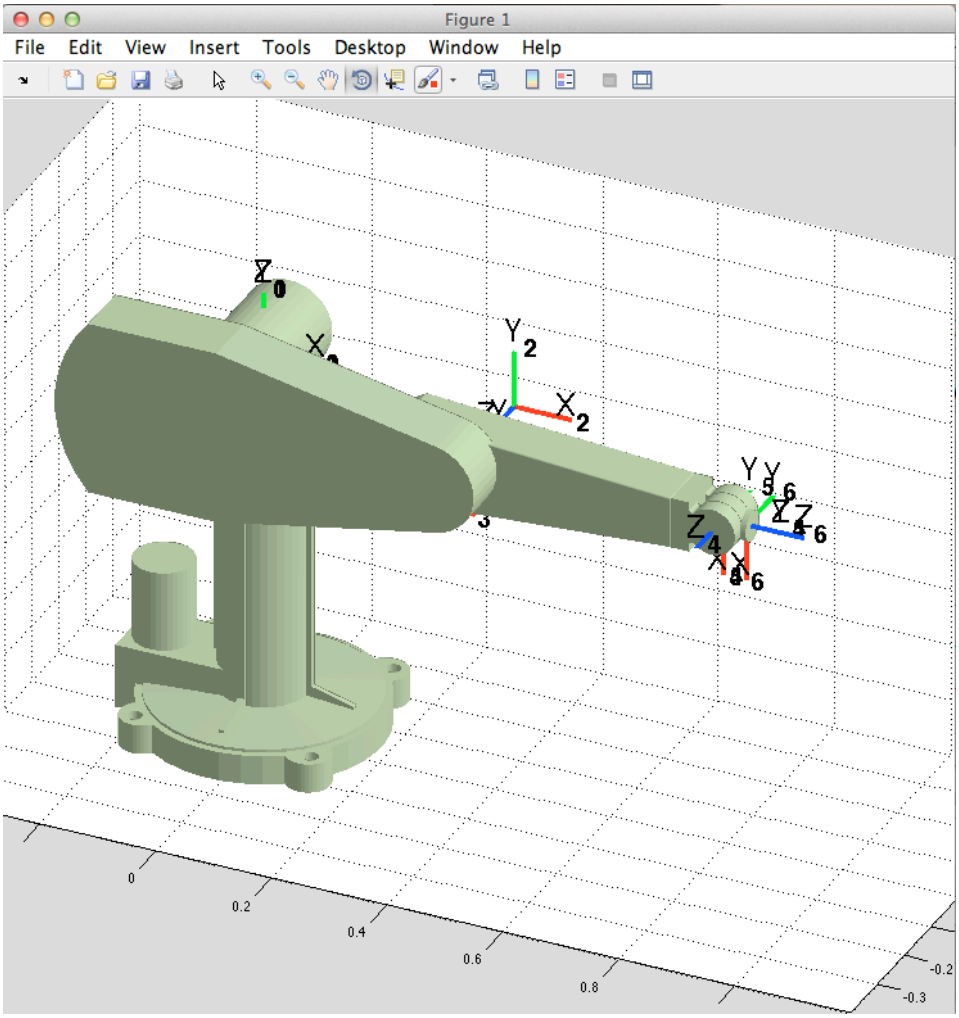


Figure 3

The approach considers that, when a trajectory is planned by the robot's controller, each actuator must follow a trapezoidal speed profile shown in Figure 4. This speed profile is often used in the selection of electric motors. This speed profile considers an acceleration phase (at a maximum angular acceleration), a phase at constant speed (maximum speed) and a deceleration phase. Each robot manufacturer specifies these parameters for their robots (you may have a look at any of the datasheets included in the library). It is common that the joints corresponding to the first, second and third joints move at lower speeds and accelerations, compared to the rest of the joints. As you will observe, these joints require higher torques and power. The shape of the trapezoidal profile for each axis can be changed at the parameters section in the file `motor_selection.m`, by changing the maximum speed and acceleration for each joint.

```
% robot pose: experiment by changing the pose while observing
the different
%              torques at each joint
q=[0 0 -pi/2 0 0 0]; %rad

%maximum speeds for joint 1, 2, 3, 4, 5 and 6
maximum_speeds=[3 3 4 5 5 5]; %rad/s
```

```
%maximum acceleration/deceleration for each joint
maximum_accels=[5 5 6 7 8 9]; %rad/second^2

% time of the trapezoidal profile that the joint moves at
maximum speed
time_at_constant_speed=0.4; %seconds
```

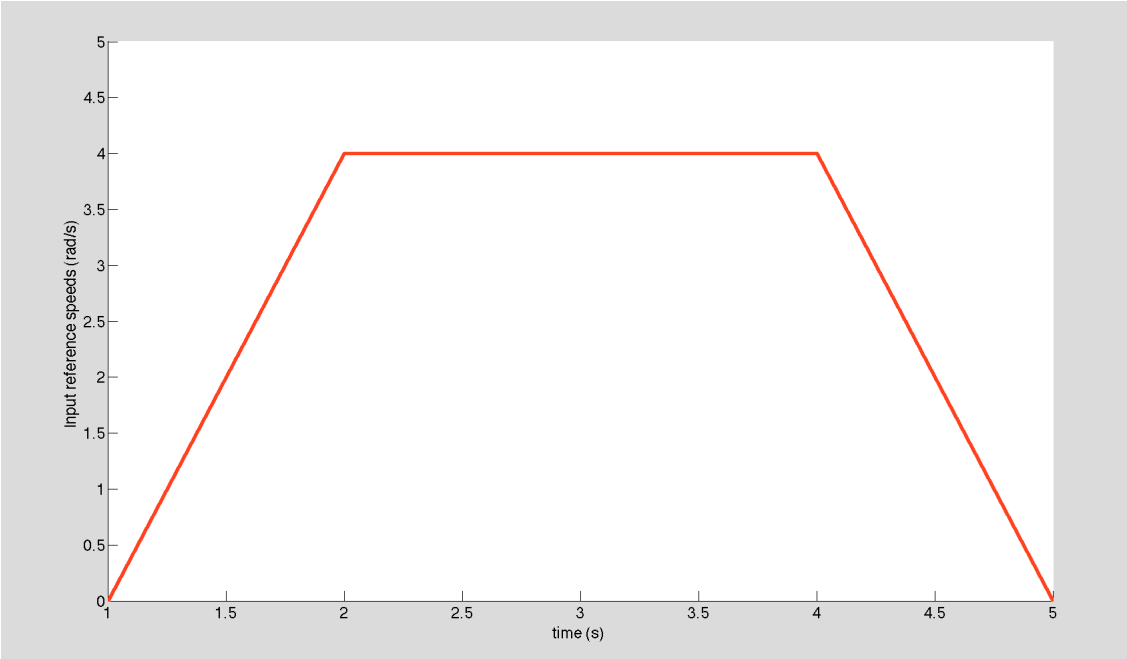


Figure 4

Figure 5 presents a typical result obtained after simulating motor_selection.m. If we have a look at this figure, we can observe that the peak torque is around 6 N·m, whereas the nominal torque is approximately 1.8 N·m. Note that the matlab script takes the gear reduction ratio in consideration. Thus, the plot presents the torque on the motor axis that differs from the torque at the joint. Selecting a suitable gear is, in addition, complex. We should take several parameters in consideration, such as:

- Gear reduction ration.
- Maximum allowed input torque.
- Maximum allowed output torque.
- Maximum allowed output or input speeds.
- Maximum axial load.
- Average backslash.

Motor manufacturers offer sometimes a selection programme that allows to select the actuator, gearhead and electric drive as a single configuration, considering at the same time most of the parameters above. It is nontheless a complex task that requires experiencies. In order to simplify this selection, the

gearhead in our case is only modelled by its reduction ratio, meaning that we are only interested in the conversion of torque and speed. This means that the torque seen by the electric motor will be reduced by the gear ratio whereas the rotor will turn at a speed increased by the same ratio.

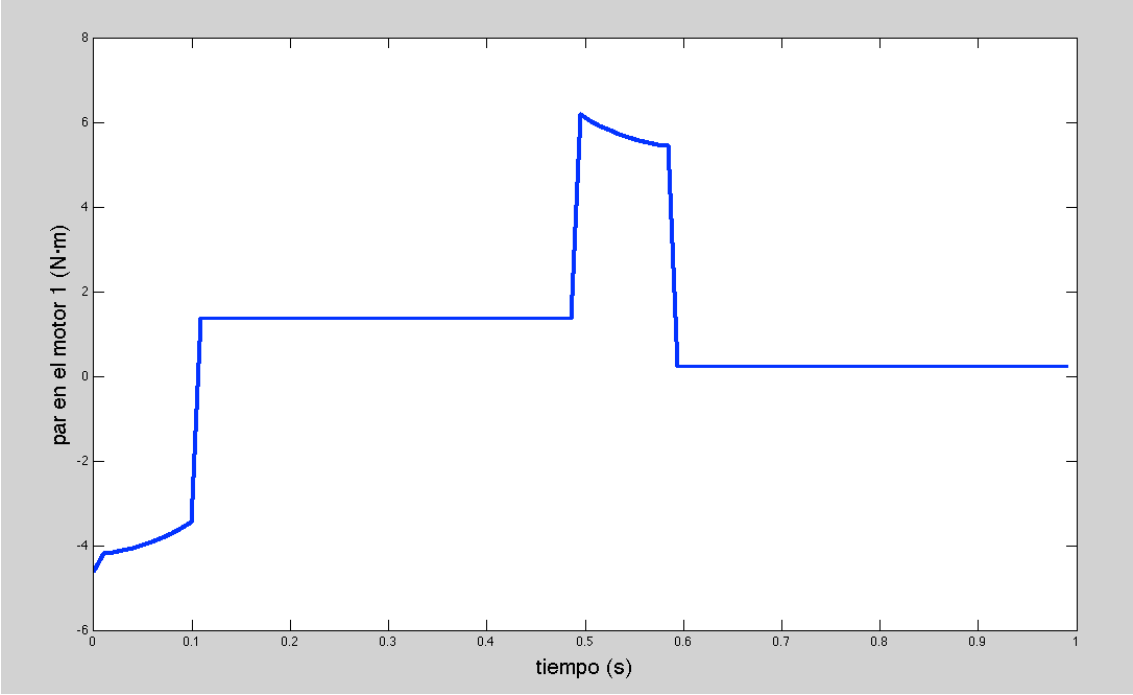


Figure 5

After executing the `motor_selection.m` script you will find the following figures:

- Figure 6 presents the input joint speeds. That is, the speed at which each joint is moving in rad/s.
- Figure 7 presents the actuator speeds. That is, the speed at which each actuator is moving in rev/min. Please note that the speed in the actuators' rotor depend on the gear ratio that is parameter that should be specified. The máximum speed in rev/min of the actuator should not surpass the value specified by the motor manufacturer.
- Figure 8 presents the torque at each joint at every time step. It is important to note that the inverse dynamics function simulates the robot at different motion states, that are specified by the speed profiles. In this sense, it is important to understand that, during the simulation:
    - The joint positions remain constant.
    - The speed varies according to the specified profile.
    - The acceleration varies accordint to the same specified profile.

It is also impor

- Figure 9 presents the torque at each joint at every time step reduced by the gear ratio, which is also a parameter that must be chosen.

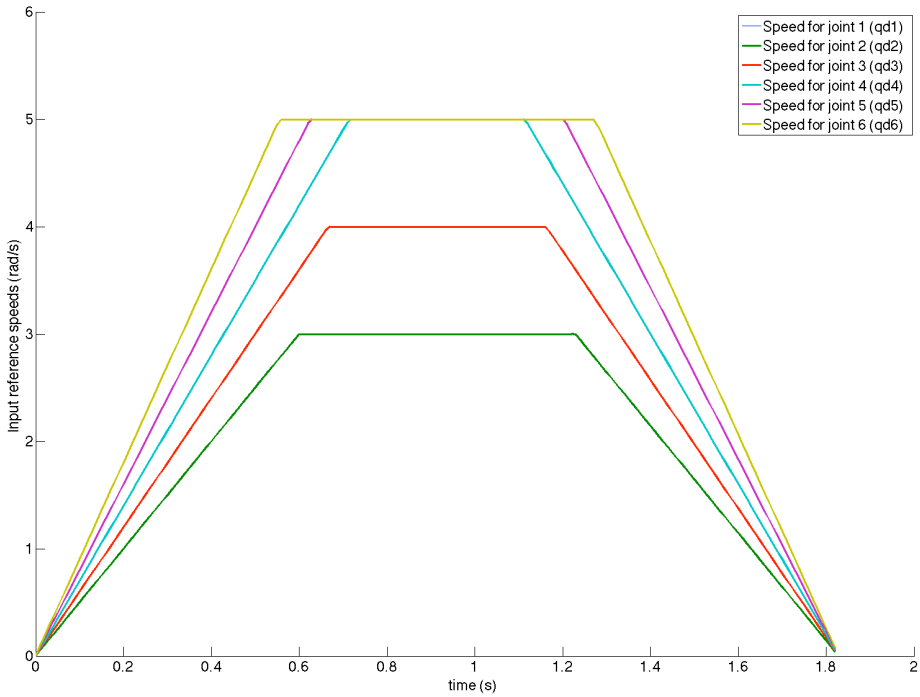


Figure 6

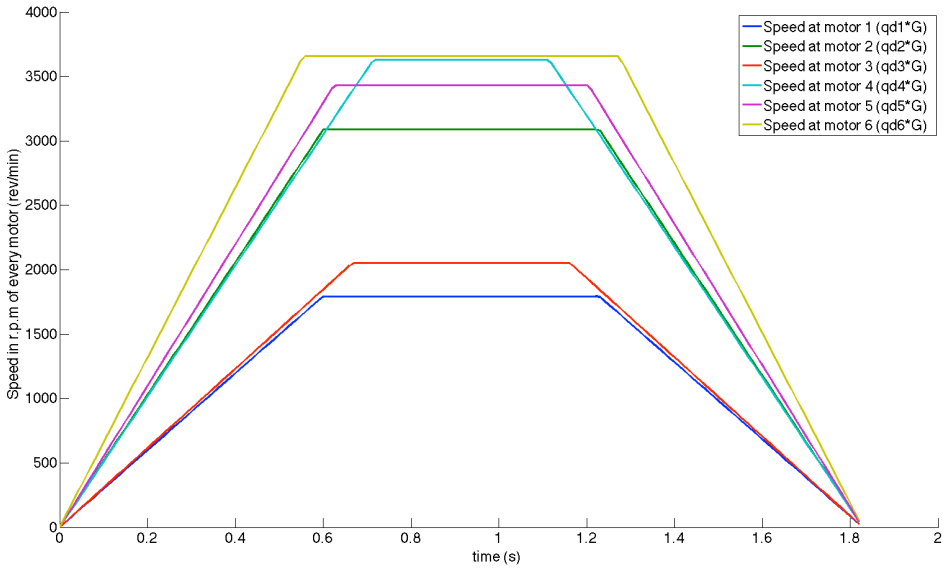


Figure 7

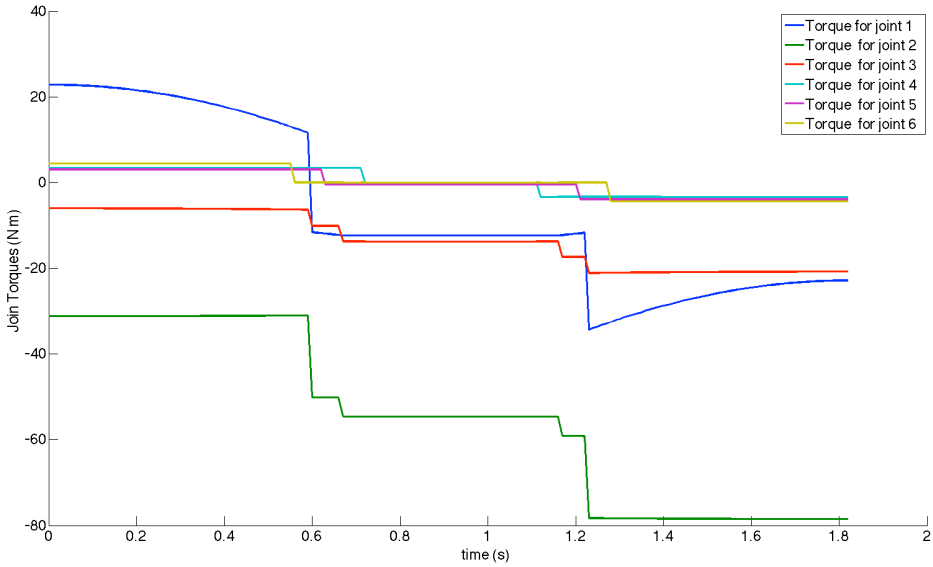


Figure 8

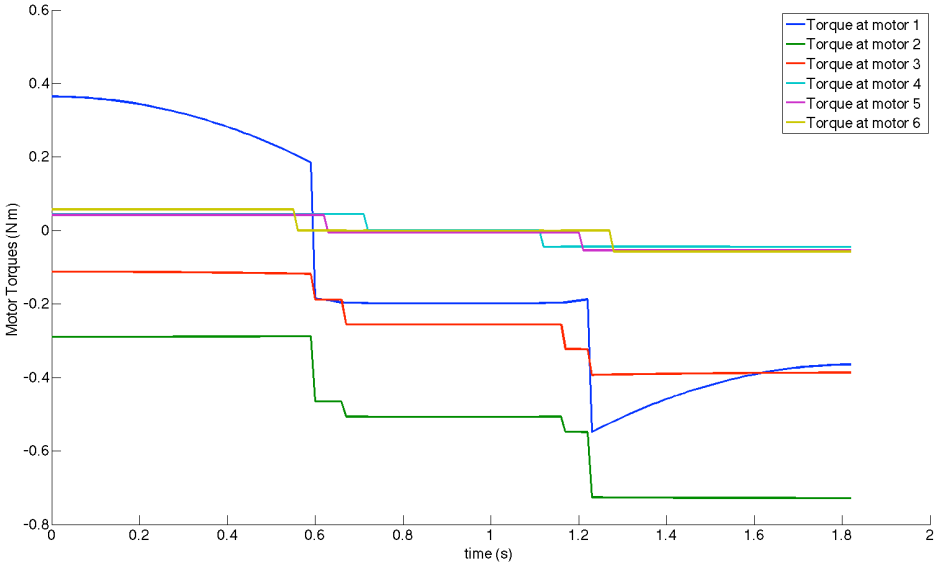


Figure 9

Please note that if we select a higher reduction ratio we will obtain lower torques at each actuator, at the prize of increasing the speed that each actuator should move. In consequence, the process of selecting an actuator and a gearhead can be very complex. For educational purposes, we have simplified the process of selecting an actuator, by considering only its maximum speed and maximum torque. It is important to note that motor manufacturers sometimes give a chart similar to the one presented in Figure 10, where the torque and speed are related to each other. In addition, in this chart we find an area of nominal speed and speed, meaning that the motor should normally work within this range of speeds and torques. Eventually (approximately a 10% of the total time), the motor may go into the peak torque and speed area, where the requirements for the motor are higher. The student should also consider

that, if we are not able to find a combination of motor and gearhed, perhaps it may be necessary to select a different motor model. Several motor manufacturers exist. In the motor_catalog directory you can find datasheet from different motor companies. If higher torques are required, perhaps we should select a bigger actuator that will bear higher torques and perhaps speeds. However, this normally increases the weight that should carry the robot, thus it is necessary to recompute the whole robot model and validate the actuator selection in a cyclic scheme.

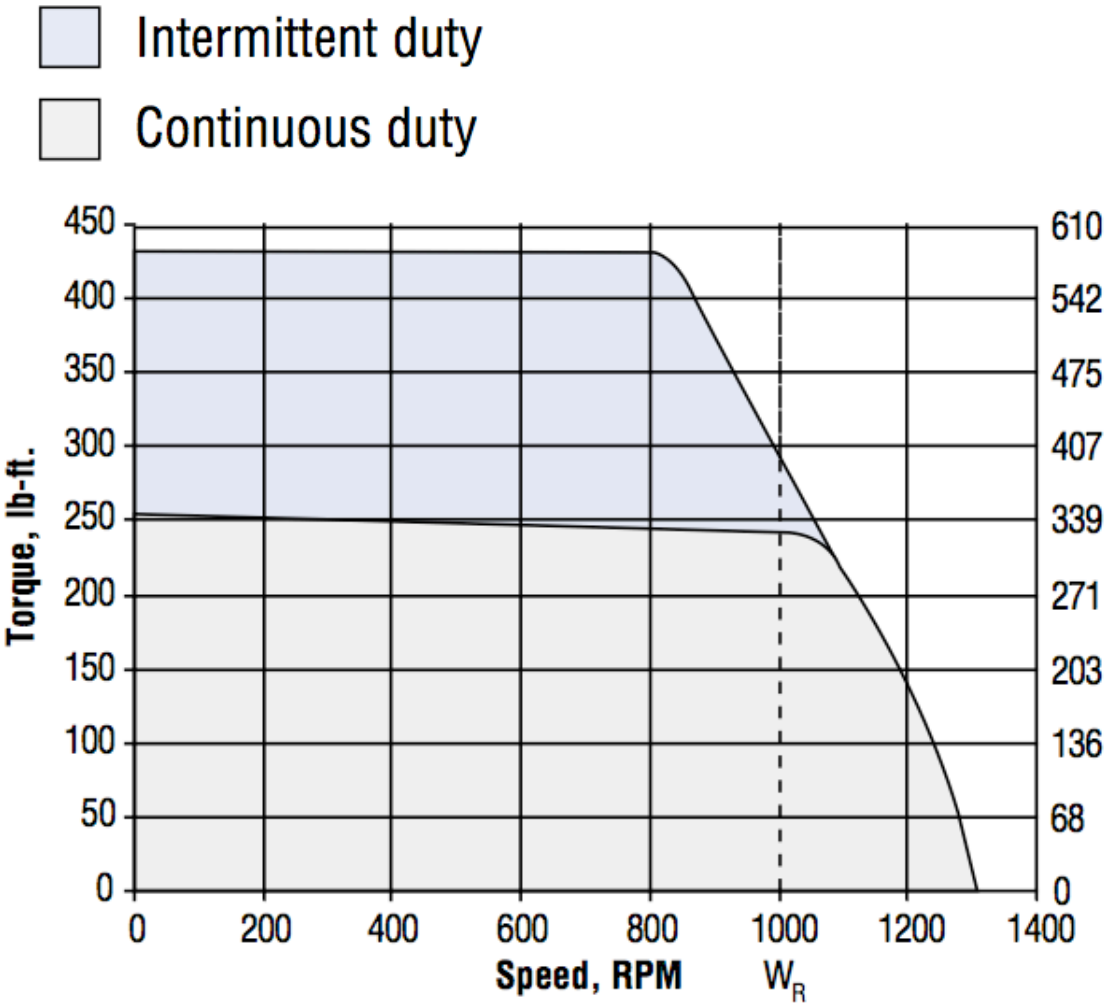


Figure 10

In order to understand how the `motor_selection.m` script works, perform the following steps:

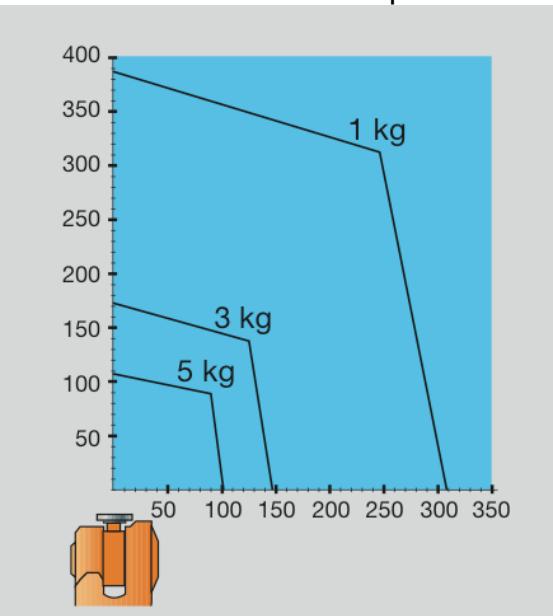
Exercise 3:

- Open the file `motor_selection.m`. Execute it.
- Visualize the trapezoidal speed profile at each joint.
- Change the máximum speeds and accelerations for each joint and observe the results in terms of joint and actuator torques.
- Find the call to the function inversedynamic(). Consider changing the load carried by the robot on its end effector.

- Plot the torque at each joint and the required motor torque. Identify the peak torque and the nominal torque.
- These torques should be increased by a 1.5 factor for safety when selecting a rotor.
- Now, you should search in the catalog (motor_catalog_maxon.pdf, ABB_catalog.pdf, PowerTec_catalog.pdf) a motor that suits the torque and speed requirements for the robot.
- Finally, note down the maximum torques at each joint for your robot.

Exercise 4:

You can now simulate the PUMA 560 robot under different configurations and loads. For example, the torques should be lower if the robot is placed at a different joint position. In consequence, the load that the robot is able to carry may be greater. As an exercise, try to draw a load diagram as the one shown that specifies the load as a function of the robot position.



Finally, the student should note that the forces implied in the movement of the arm are coupled. That means that if one of the links accelerates at a high rate, forces and torques will be induced to different joints. In this sense, in some cases it may be interesting to simulate the movement and accelerations with different sign, since it is “a priori” difficult to foresee the results.

4 Summary

Everything is summarized in the following videos:

- Inverse dynamic of a 6 DOF robot.

http://arvc.umh.es/arte/videos/pr3_video1_din_inversa.mp4 - Motor selection (simulink).
http://arvc.umh.es/arte/videos/pr3_video1_din_inversa.mp4