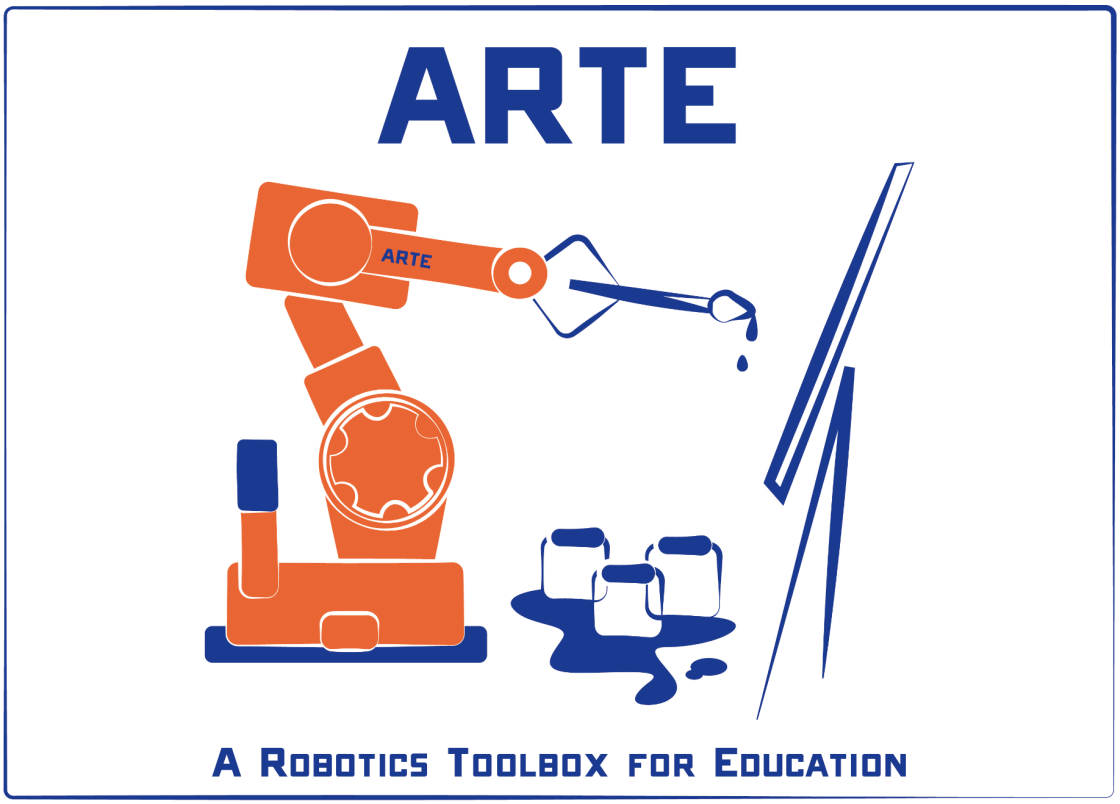
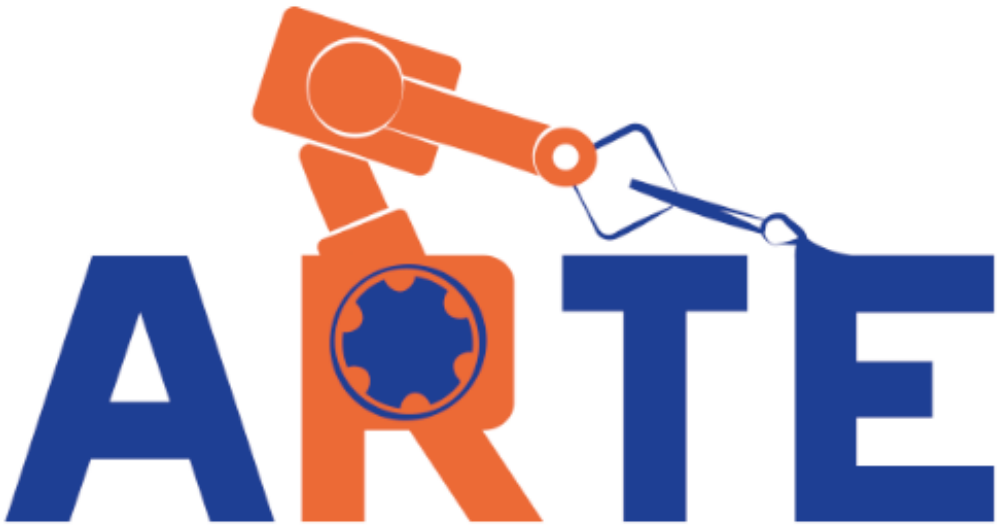


PRACTICAL SESSION 4: FORWARD DYNAMICS

Arturo Gil Aparicio

arturo.gil@umh.es



OBJECTIVES

After this practical session, the student should be able to:

- Simulate the movement of a simple mechanism using the library.
- Simulate the movement a robotic arm when a set of torques are applied to its joints.
- Modify the parameters of the arm and observe the results.

1 First steps

The dynamic simulation of a robotic arm is based upon a set of equations and assumes that a great number of parameters are known. These parameters include the inertia matrices that model the robot links, the center of mass of each link with respect to its D-H reference system and more... such as viscous friction factors.

On our last practical session we investigated the inverse dynamic problem. In this case, we consider that each robot joint is at a particular position, and each joint moves at a known speed and acceleration. The inverse dynamic equations allow us to compute the torque joints that would bring the robot instantaneously to this movement state.

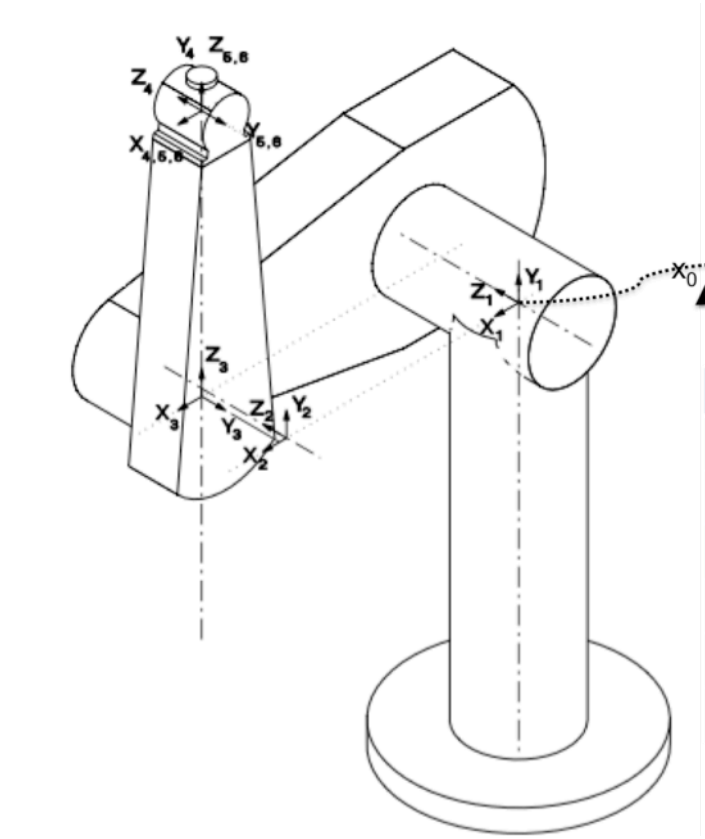
The forward dynamics model allows us to compute the movement of the robot when a set of torques are applied at its joints. This model is usually expressed in the following manner:

In this case, given that the acceleration of each joint can be computed, the movement of the robot can be simulated by integrating this acceleration over time starting from any initial condition.

During this session we will focus on the simulation of the Unimate Puma 560 robot,

since this robot has been studied for years and its dynamic parameters are known with some precision. Figure presents the robotic arm and its D-H parameter table.

The parameters that define the dynamics of this arm can be found under arte/robots/unimate/puma560. You should now edit this file and observe these parameters.



	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

As usual, we can the robot and observe it:

```
>> pwd
>> init_lib
>> robot = load_robot('unimate','puma560')
>> q=[0 0 0 0 0 0]
>> drawrobot3d(robot, q)
```

2 Simulating the robot

You can find different examples to simulate the forward dynamics under the demos directory:

- `forward_dynamics_demo_1dof.m`: simulate a 1DOF robot link. The dynamic parameters for this robot are easy to understand.
- `forward_dynamics_demo_2dof.m`: simulate a 1DOF robot link. The dynamic parameters for this robot are easy to understand.

- `forward_dynamics_demo_3dof.m`: simulates the case of a 3 DOF planar arm. The dynamic parameters for this robot are easy to understand. In addition, the student may easily predict and understand the simulated movement of this robot.

- `forward_dynamics_demo.m`: simulates the case of a Unimate Puma 560 robotic manipulator. This is a well known classic robot and its dynamic parameters have been estimated by several researchers.

Students should start by editing and executing the `forward_dynamics_demo_3dof.m` file. This demo will surely help them understand the forward dynamic solutions included in the library. Inside the file the simulation parameters are commented and you may change them and observe the results in the animation.

Next, run the demo file arte/demos/forward_dynamic.m. The demo simulates the movement of the PUMA 560 robot when no torques are applied at its joints. The robot is placed at its initial position at the beginning of the motion. When no torques are applied at its joints, we should observe that the links start a free motion under the action of gravity. You should also observe the position and velocity plots.

Exercise 1:

Start by executing the forwarddynamics_1DOF.m demo. Simulate the following cases:

- 1.a) Robot with no friction on joint 1 (q1).
  (Explain the results).
- 1.b) Robot with different friction coefficients on joint 1 (q1).
  Edit parameters.m in arte/robots/example/1dofplanar
  Edit `robot.motors.Viscous = [1.2];`
  Edit with the following values--> 0.1, 1.5, 3, 5
  (Explain the results).

- 1.c) Simulate different gravities:

  - g= [0 -9.81 0] m/s^2
  - g= [0 9.81 0] m/s^2
  - g= [9.81 0 0] m/s^2
  - g= [0 0 9.81] m/s^2

(Explain the results).

- Now apply torques to the joint.

tau = [0] means that no torques are applied... what if a constant torque is applied and

tau = [10]

tau = [-10]

Try to change tau so that the system is balanced at the starting joint position.

Exercise 2:

Start by executing the forward_dynamics_3dof.m demo and justify the results. Repeat the activities from exercise 1.

The following code allows us to compute the torques using the inverse dynamic model when the robot stands at its initial position with zero speed and accelerations and no forces are applied externally at its end effector. The last line computes the acceleration when the previosly computed torques are applied.

```
>> q = [0 0 0 0 0 0] %position
>> qd = [0 0 0 0 0 0] %velocity
>> qdd = [0 0 0 0 0 0] %acceleration
>> g = [0 0 9.81]' %gravity
>> fm=[0 0 0 0 0 0] % forces and moments
>> tau = inversedynamic(robot, q, qd, qdd, g, fm)
>> qdd = accel(robot, q, qd, tau, g)
```

Exercise 3:

Execute the following code:

```
>> robot=load_robot('example', '3dofplanar');
>> q = [0 0 0 0 0 0] %position
>> qd = [0 0 0 0 0 0] %velocity
>> qdd = [0 0 0 0 0 0] %acceleration
>> g = [0 0 9.81]' %gravity
>> fm=[0 0 0 0 0 0] % forces and moments
>> tau = inversedynamic(robot, q, qd, qdd, g, fm)
>> qdd = accel(robot, q, qd, g, tau)
```

The last line makes reference to the forward dynamics model of the robot. Observe the result saved at qdd. How can you justify it?

```
qdd = accel(robot, q(:), qd(:), tau(:), g)
```

- Explain the results.
- Propose an application for that technique!

You should now simulate the `forward_dynamics_3dofdemo.m` demo using the torques tau computed before. What results should you expect?

Exercise 4:

Apply the same concept to the 3 DOF planar demo. Obtain the torques needed to match any acceleration qdd.

You should now simulate the `forward_dynamics_demo.m` demo using the torques tau computed before. What results should you expect?

3 Simulating your robot

In this section you should add the required dynamic parameters to your robot. These data are not always provided by the robot manufacturers. In consequence, you should give a guess to most of the parameters. For example, the manufacturers publish the total mass of the robot, but not the mass of each link. You can try to distribute the total mass along each link with some logic.

Exercise 7:

7.1 Simulate the equations of a DC motor.
Look under exercises/simulation/simulate_motor.mdl
Change R, L, Kt and Ki and observe the results. Max voltage in the motor is 48 V.
- Visualize in a scope:
- motor position, motor speed, motor acceleration.
- motor current.
- motor power: Total power, mechanical power, Joule losses and friction losses.
- Limit the maximum current (A) using a saturation block. Limit it to 84 A.
- Find the maximum speed of the motor to withstand 3 and 5 Nm.

7.2 Simulate the case in which the motors actuate over the mechanism.
The output of the motor block as simulated before is tau.
The input on the forward dynamics function are also torques!
please analyze the system under
exercises/simulation/simulate_robot_and_motors.mdl

- change robot.motors.G and observe the results.
- modify the torque constant to match the constant of your selected motor, what can you see?