# Report of Assignment 2

## Report of the second assignment  of Robot Dynamics and Control

*Marco Macchia, mat.* 4687564

## 0. Introduction

In order to solve the inverse dynamic problem in the second assignment, I had to use the recursive Newton-Euler algorithm.

Manipulator inverse dynamics, or simply inverse dynamics, consists of the computation of the torques required from all the joints in order to produce a given motion trajectory consisting of a set of joint positions, velocities and accelerations.

External torques, the robot's current configuration, and the velocity and acceleration of each joint were used as inputs, and the final torques were output.

## 1. Newton-Euler algorithm

### 1.1. Recursive Newton-Euler algorithm

The recursive Newton-Euler algorithm is based on a precise computational path that is divided into two phases: *Forward recursion* and *Backward recursion.*

To begin, the *forward equations* compute the link velocities and accelerations, as well as the dynamic wrench on each link, from link 1 to link n.

The reaction wrenches on the links and, as a result, the joint torques are calculated using the *backward equations* from link n to the base.

As previously stated, this approach computes joint torques based on the joint configuration.

### 1.2. Forward recursion

For each computational step, going from link 1 to n, first I need to compute the position of $P_i$ with respect to $P_{i-1}$:

$$(1.2.1) \qquad r_{i/i-1} = Q_{i-1}^{+} - P_{i-1} = \begin{cases} Q_{i-1} - P_{i-1} & i \in RJ \\ \\ Q_{i-1} - P_{i-1} + k_i q_i & i \in TJ \end{cases}$$

where $k_i q_i$ is the displacement of the translational joint.

After that I compute the angular and linear velocities, using the formulas below:

$$(1.2.2) \qquad w_{i/0} = \begin{cases} w_{i-1/0} + k_i \dot{q}_i & i \in RJ \\ w_{i-1/0} & i \in TJ \end{cases}$$

$$(1.2.3) \qquad v_{i/0} = \begin{cases} v_{i-1/0} + w_{i-1/0} \times r_{i/i-1} & i \in RJ \\ v_{i-1/0} + w_{i-1/0} \times r_{i/i-1} + k_i \dot{q}_i & i \in TJ \end{cases}$$

where $k_i \dot{q}_i$ is the velocity of displacement of the translational joint, and $r_{i/i-1}$ is constant just in case of a rotational joint.

The linear and angular accelerations must also be determined to complete the forward part of the algorithm, using this formulas:

$$(1.2.4) \qquad \dot{w}_{i/0} = \begin{cases} \dot{w}_{i-1/0} + (w_{i-1/0} \times k_i)\dot{q}_i + k_i \ddot{q}_i & i \in RJ \\ \dot{w}_{i-1/0} & i \in TJ \end{cases}$$

$$(1.2.5) \qquad \dot{v}_{i/0} =$$
$$\begin{cases} \dot{v}_{i-1/0} + (\dot{w}_{i-1/0} \times r_{i/i-1}) + w_{i-1/0} \times (w_{i-1/0} \times r_{i/i-1}) & i \in RJ \\ \dot{v}_{i-1/0} + (\dot{w}_{i-1/0} \times r_{i/i-1}) + w_{i-1/0} \times (w_{i-1/0} \times r_{i/i-1}) + 2(w_{i-1/0} \times k_i)\dot{q}_i + k_i \ddot{q}_i & i \in TJ \end{cases}$$

which takes into account both the ***Coriolis effect*** and the ***centripetal effect***.

### 1.3. Backward recursion

The *backward equations* from link n to the base supply the reaction wrenches on the links and, as a result, the joint torques.

First, I compute the dynamic force and dynamic moment in the reverse phase:

$$(1.3.1) \qquad \begin{aligned} D_i &= m_i \dot{v}_{Ci/0} \\ \Delta_i &= I_{Ci} \dot{w}_{i/0} + (w_{i/0} \times I_{Ci} w_{i/0}) \end{aligned}$$

Then I compute the forces and moments using the following formulas, which takes into account both the Coriolis and the centripetal effect:

$$(1.3.2) \qquad F_{i/i-1} = F_{i+1/i} - m g_i + F_i^{ext} + D_i$$
$$(1.3.3) \qquad M_{i/i-1} = M_{i+1/i} - M_{Ci}^{ext} - (r_{i/Ci} \times F_{i/i-1}) + (r_{i+1/Ci} \times F_{i+1/i}) + \Delta_i$$

Then $\tau_i$ can be simply found just by using this formula:

$$(1.3.4) \qquad \tau_i = \begin{cases} M_{i/i-1} k_i & i \in RJ \\ F_{i/i-1} k_i & i \in TJ \end{cases}$$

## 2. Exercise 1

In the first exercise I had to implement the recursive Newton-Euler algorithm for inverse dynamics, while keeping some specific requirements in mind.

Of course, since the other three exercise are meant to evaluate the implemented method, this exercise is essential for the others. The MATLAB script in my folder includes a fully commented implementation of the algorithm, which is a function called `NewtonEuler`.

## 2.1. Robot Data

For each of the other exercises I choose to develop a proper data struct (because there are, of course, different data for each exercise related to that specific robot).

Here's an example of the robot struct structure:

```
RobData2_1 = struct();
RobData2_1.jnum = 2; % number of joints
RobData2_1.jtypes = [0 0]; % type of joints
RobData2_1.rot_axis = [3 3]; %rotation axis of the joint
RobData2_1.gaxis = 2; % gravity axis on which acts gravity
RobData2_1.mass = [22 19]; % mass of links
RobData2_1.len = [1 0.8]; % length of links
RobData2_1.I = zeros(3,3,RobData2_1.jnum);% inertia matrix
RobData2_1.I(:,:,1) = diag([0.4 0.4 0.4]);
RobData2_1.I(:,:,2) = diag([0.3 0.3 0.3]);
RobData2_1.q = [deg2rad(20) deg2rad(40)]; % configuration of the joints
RobData2_1.qdot = [0.2 0.15]; % velocities of joints
RobData2_1.q2dot = [0.1 0.085]; % acceleration of joints
RobData2_1.Mext = [0; 0; 0]; % external moments acting on the robot
RobData2_1.Fext = [0; 0; 0]; % external forces acting on the robot
RobData2_1.Rabs = zeros(3,3,RobData2_1.jnum); %axis orient. of frame i wrt frame 0
RobData2_1.Rabs(:,:,1) = eye(3);
RobData2_1.Rabs(:,:,2) = eye(3);
```

# 3. Exercise 2

This is the configuration of the robot manipulator for the second exercise of the assignment.

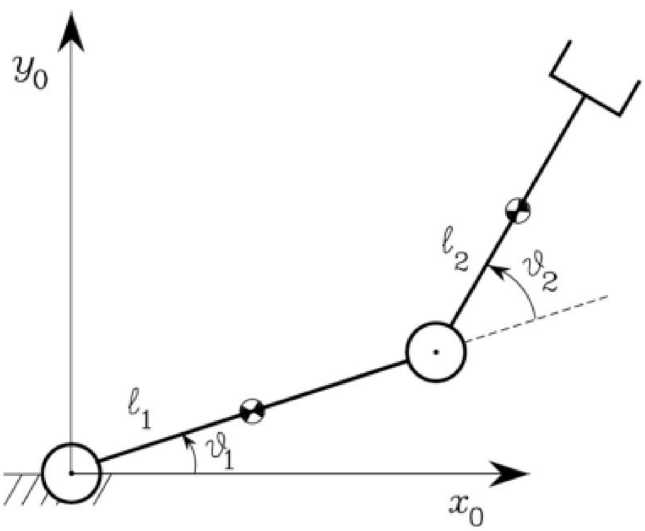


Figure 1 : 2R Planar Manipulator

**General data :**

- $m_1 = 22Kg, m_2 = 19Kg$
- $l_1 = 1m$ , $l_2 = 0.8m$
- $I_{zz1} = 0.4Kgm^2, I_{zz2} = 0.3Kgm^2$

Assumed that the links are prisms with a uniform mass distribution, the $CoM$ of each one is located at the geometric center. I have to find the inverse dynamics joint torques $\tau_1, \tau_2$ for the following two configurations, first without gravity, then with gravity acting along the $y_0$ axes.

## 3.1. Instance 1 with no gravity

**Data :**

- $\theta_1 = 20^\circ, \theta_2 = 40^\circ$
- $\dot{\theta}_1 = 0.2 rad/s, \dot{\theta}_2 = 0.15 rad/s$
- $\ddot{\theta}_1 = 0.2 rad/s^2, \ddot{\theta}_2 = 0.085 rad/s^2$

With this particular configuration I expect low result because gravity is absent. Moreover, since $\theta_1, \theta_2$ are both positive, and seen that the joint velocities and accelerations are all positive, the  the results should have a positive value.

**Results :**

| | **Result** |
|---|---|
| $\tau_1$(Nm) | 4.3641 |
| $\tau_2$ (Nm) | 1.3955 |

## 3.2. Instance 1 with gravity

**Data :**

- $\theta_1 = 20^\circ, \theta_2 = 40^\circ$
- $\dot{\theta}_1 = 0.2 rad/s, \dot{\theta}_2 = 0.15 rad/s$
- $\ddot{\theta}_1 = 0.2 rad/s^2, \ddot{\theta}_2 = 0.085 rad/s^2$

Here the values should be higher because of the gravity. The other considerations also applies in this case, so I expect positive values of tau

**Results :**

| | **Result** |
|---|---|
| $\tau_1$(Nm) | 318.1937 |
| $\tau_2$ (Nm) | 38.6735 |

## 3.3. Instance 2 with no gravity

**Data :**

- $\theta_1 = 90^\circ, \theta_2 = 45^\circ$
- $\dot{\theta}_1 = -0.8 rad/s \; \dot{\theta}_2 = 0.35 rad/s$
- $\ddot{\theta}_1 = -0.4 rad/s^2, \ddot{\theta}_2 = 0.1 rad/s^2$

Using this configuration I expect low result because gravity is absent. Here, since the configuration tells that the first joint is rotating clockwise and the second one is rotating counterclockwise, I expect $\tau_1$ to be negative and $\tau_2$ to be positive.

**Results :**

| | **Result** |
|---|---|
| $\tau_1$(Nm) | -12.3727 |
| $\tau_2$ (Nm) | 0.2878 |

### 3.4. Instance 2 with gravity

**Data :**

- $\theta_1 = 90^\circ$, $\theta_2$ = $45^\circ$
- $\dot{\theta}_1 = -0.8 rad/s$ $\dot{\theta}_2 = 0.35 rad/s$
- $\ddot{\theta}_1 = -0.4 rad/s^2$, $\ddot{\theta}_2 = 0.1 rad/s^2$

With the gravity the results gets higher. Moreover, I expect $\tau_2$ to become negative, because the second joint has a total angle of $\theta = 135^\circ$, and therefore in order to counteract the gravity effect the application torque should follow the clockwise direction.

**Results :**

| | **Result** |
|---|---|
| $\tau_1$(Nm) | -65.0917 |
| $\tau_2$ (Nm) | -52.4313 |

## 4. Exercise 3

This is the configuration of the robot manipulator for the third exercise of the assignment.

**IMPORTANT REMARK:** Here the exercise data hasn't a clear, unique interpretation, because in the given text $d2 < l_1$ for both the configurations, which is geometrically impossible. So I interpreted this picture as if $l_1$ is the total length of the joint 1 (meaning that with $\theta_1 = 0 \quad CoM_1 = [0.5, 0]$) and the joint 2 has the $CoM$in the same position as the end of the link 1, with displacement $d_2$.

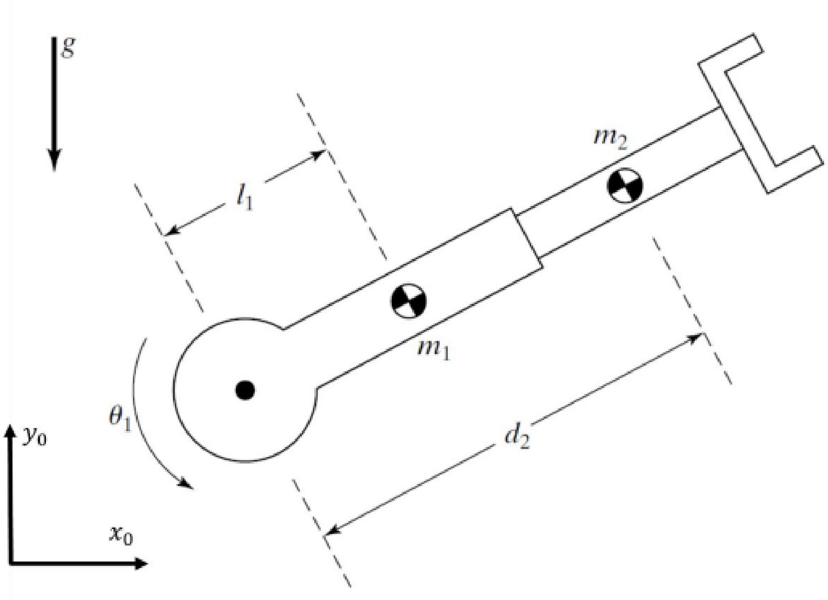

Figure 2 : RP Planar Manipulator

**General data :**

- $m_1 = 10Kg, m_2 = 6Kg$
- $l_1 = 1m$
- $I_{zz1} = 0.4Kgm^2, I_{zz2} = 0.3Kgm^2$

Assumed that the links are prisms with a uniform mass distribution, the $CoM$ of each one is located at the geometric center. I have to find the inverse dynamics joint torques $\tau_1, \tau_2$ for the following two configurations, first without gravity, then with gravity acting along the $y_0$ axes.

## 4.1. Instance 1 with no gravity

**Data :**

- $\theta_1 = 20^\circ, d_2 = 0.2m$
- $\dot{\theta}_1 = 0.08rad/s, v_2 = 0.03m/s$
- $\ddot{\theta}_1 = 0.1rad/s^2, a_2 = 0.01m/s^2$

In this particular configuration gravity is not considered, so the results should be low. The configuration of the the manipulator highlights that the first joint is rotating counterclockwise and the second joint is *expanding*, so I expect two positive values.

**Results :**

| | Result |
|---|---|
| $\tau_1$(Nm) | 1.2186 |
| $\tau_2$ (N) | 0.0139 |

## 4.2. Instance 1 with gravity

**Data :**

- $\theta_1 = 20^\circ, d_2 = 0.2m$
- $\dot{\theta}_1 = 0.08 rad/s, v_2 = 0.03 m/s$
- $\ddot{\theta}_1 = 0.1 rad/s^2, a_2 = 0.01 m/s^2$

Here gravity is consideres, so I only expect higher values of $\tau_1, \tau_2$.

**Results :**

| | **Result** |
|---|---|
| $\tau_1$(Nm) | 113.6829 |
| $\tau_2$ (N) | 20.1452 |

### 4.3. Instance 2 with no gravity

**Data :**

- $\theta_1 = 120^\circ, d_2 = 0.6m$
- $\dot{\theta}_1 = -0.4 rad/s, v_2 = -0.08 m/s$
- $\ddot{\theta}_1 = -0.1 rad/s, a_2 = -0.01 m/s^2$

Here gravity is set to 0, so results shouldn't be high. Moreover the first joint is rotating counteclockwise, while the second joint is *retracting*, so I expect two low negative values for the two *tau*.

**Results :**

| | **Result** |
|---|---|
| $\tau_1$(Nm) | -1.2416 |
| $\tau_2$ (N) | -1.5960 |

### 4.4. Instance 2 with gravity

**Data :**

- $\theta_1 = 120^\circ, d_2 = 0.6m$
- $\dot{\theta}_1 = -0.4 rad/s, v_2 = -0.08 m/s$
- $\ddot{\theta}_1 = -0.1 rad/s, a_2 = -0.01 m/s^2$

Here gravity is present again. The first joint should have an even lower value of $\tau_1$ because the gravity tries to rotate the joint counterclockwise. In this particular configuration also $\tau_2$ should have an high positive value in order to counterbalance the gravity effect that tries to *push down* too much the prismatic joint.

**Results :**

| | **Result** |
|---|---|
| $\tau_1$(Nm) | -72.8546 |
| $\tau_2$ (N) | 49.3783 |

## 5. Exercise 4

This is the configuration of the robot manipulator for the fourth exercise of the assignment.

**IMPORTANT REMARK:** The given base frame is a left-handed frame, and therefore it is not acceptable. In order to have a right-handed frame it is sufficient to invert the sense of the $y_0$ axis.

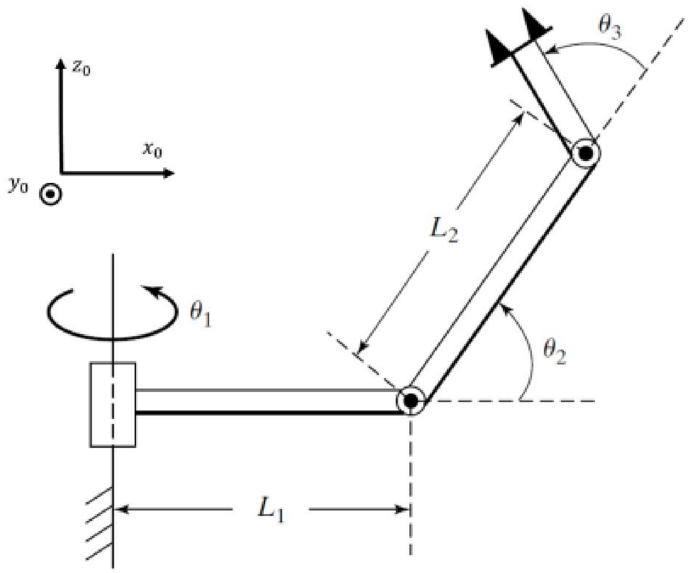


Figure 3 : 3R Non-Planar Manipulator

**General data :**

- $m_1 = 20Kg, m_2 = 20Kg, m_3 = 6Kg$
- $l_1 = 1m, l_2 = 0.8m, l_3 = 0.35m$
- $I_1 = diag(0.2Kgm^2, 0.2Kgm^2, 0.8Kgm^2)$
- $I_2 = diag(0.2Kgm^2, 0.2Kgm^2, 0.8Kgm^2)$
- $I_3 = diag(0.08Kgm^2, 0.08Kgm^2, 0.1Kgm^2)$

Assumed that the links are prisms with a uniform mass distribution, the $CoM$ of each one is located at the geometric center. I have to find the inverse dynamics joint torques $\tau_1, \tau_2, \tau_3$ for the following configuration, first without gravity, then with gravity acting along the $z_0$ axes.

## 5.1. Instance with no gravity

**Data :**

- $\theta_1 = 20^\circ, \theta_2 = 40^\circ, \theta_3 = 10^\circ$
- $\dot{\theta}_1 = 0.2rad/s, \dot{\theta}_2 = 0.15rad/s, \dot{\theta}_3 = -0.2rad/s$
- $\ddot{\theta}_1 = 0.1rad/s^2, \ddot{\theta}_2 = 0.085rad/s^2, \ddot{\theta}_3 = 0.0rad/s^2$

In this unique configuration for exercise four, without the effect of the gravity, I expect low results for the three *tau*. The first joint is rotating counterclockwise along the $z_0$ axis, so I expect a positive value for $\tau_1$. The same thing applies for $\tau_2$, where the joint is rotating in the same direction along the $y_0$ axis. At first glance, since the joint 3 is rotating clockwise I can expect a negative value of $\tau_3$, but seen that the joint velocity is constant (because $\ddot{\theta}_3 =$

$0.0rad/s^2$), a small amount of torque should be applied in the counterclockwise direction in order to counteract the **inertia** generated by the movement.

**Results :**

| | **Result** |
|---|---|
| $\tau_1$(Nm) | 5.1128 |
| $\tau_2$ (Nm) | 1.3712 |
| $\tau_3$ (Nm) | 0.1532 |

### 5.2. Instance with gravity

**Data :**

- $\theta_1 = 20^\circ, \theta_2 = 40^\circ, \theta_3 = 10^\circ$
- $\dot{\theta_1} = 0.2rad/s, \dot{\theta_2} = 0.15rad/s, \dot{\theta_3} = -0.2rad/s$
- $\ddot{\theta_1} = 0.1rad/s^2, \ddot{\theta_2} = 0.085rad/s^2, \ddot{\theta_3} = 0.0rad/s^2$

Here the only difference is the presence of the gravity. $\tau_1$ has the same value as before because gravity acts in the same axis as the rotation axis of the joint 1. The other two values are higher due to the gravity.

**Results :**

| | **Result** |
|---|---|
| $\tau_1$(Nm) | 5.1128 |
| $\tau_2$ (Nm) | 104.1829 |
| $\tau_3$ (Nm) | 6.7743 |

## 6. Conclusions

It was a bit challenging to implement the algorithm because I want to make it the most general possible, and so I had to pay attention to many aspects that could require different treatment in the code (for example the computation of the $CoM$ given $Q$, which also requires to know the axis of rotation of $Q$ and also the joint type, so that the function should know if that particular configuration has to be intended as an angle or as a displacement.)

I'm happy because everything is computed in real-time and not hard-coded in the robot struct (such as the $CoM$ and the joint positions), in order to make the function even more generic.

I consider the results obtained compatible with the given configurations. I don't have any kind of solution so it is difficult to tell if the values are very precise, but in every exercise the values computed are consistent, meaning that it is clearly visible the effect of the gravity, and the direction of each $\tau$ is fully consistent with the given configuration.