Università
degli Studi
di Padova

UNIVERSITÀ DEGLI STUDI DI PADOVA

DEPARTMENT OF INDUSTRIAL ENGINEERING

Dipartimento Di Ingegneria Industriale - DII

MASTER DEGREE IN MECHANICAL ENGINEERING

DYNAMIC BALANCING METHODS FOR AERIAL MANIPULATORS

Supervisor:
Prof. COCUZZA SILVIO

Author:
GIUNCHI SILVIO
Student ID:
2018688

ACADEMIC YEAR 2022/2023

To my Mother, my Father and Laura who believed in me
and pushed me to overcome my difficulties.
To Lucia, who made me believe in myself.

Summary

Aerial manipulation using small and medium-scale drones is a rapidly growing field of research, which does, however, presents certain criticalities. The main ones are related to the instability of the drone during manipulation in hovering conditions, due to the movement of the manipulator mounted to it. Therefore, the objective of this thesis is to devise and propose various manipulator configurations to reduce this instability. In particular, trajectories of the manipulator's end-effector are treated exclusively in the vertical plane, thus attempting to reduce the drone rotation identified as roll, and its horizontal and vertical translation. The disturbances on the drone can be classified as gravitational, typically of greater intensity, and inertial. The configurations presented consider both.

The inverse kinematics of the manipulator to follow the trajectory imposed on the end-effector was performed using the generalised Jacobian, and in the case of redundancies the extended Jacobian or the Pseudo-inverse. The best methods in terms of results and feasibility, considering position and velocity limits of the UAM's joints, were obtained by using the drone's battery as a translating counterweight moving simultaneously with the manipulator, and with the manipulator mounted on it. Simulations were also carried out for a load picking task of $0.5\,kg$. The methods presented were developed on MatLab® and validated with Adams View®.

Sommario

La manipolazione aerea mediante droni di piccole e medie dimensioni è é una campo di ricerca in grande espansione, che presenta però alcune criticità. Le principali sono legate all'instabilità del drone durante la manipolazione in condizione di volo stazionario, dovuta al movimento del manipolatore ad esso connesso. Pertanto, l'obiettivo di questa tesi é ideare e proporre varie configurazioni di manipolatori che permettano di ridurre questa instabilità. In particolare, vengono trattate traiettorie del end-effector del manipolatore esclusivamente nel piano verticale, cercando perciò di diminuire la rotazione del drone identificata come rollio, la sua traslazione orizzontale e verticale. I disturbi sul drone sono classificabili di natura gravitazionale, tipicamente di maggiore intensità, ed inerziale. Le configurazioni presentate considerano entrambi.

La cinematica inversa del manipolatore per seguire la traiettoria imposta all'end-effector é stata effettuata mediante l'utilizzo dello Jacobiano generalizzato, e nel caso di ridondanze lo Jacobiano esteso o la Pseudo-inversa. I metodi migliori in quanto a risultati ed a fattibilità, considerando limiti di posizione e di velocità dei giunti del UAM, sono stati ottenuti sfruttando la batteria del drone come contrappeso traslante muovendosi in simultanea col manipolatore e, con il manipolatore montato su di essa. Sono state eseguite anche delle simulazioni con presa di un carico di 0,5 kg. I metodi presentati sono stati sviluppati su MatLab® e validati con Adams View®.

Contents

1. Introduction

1.1 State of the Art

In recent years, the use of Unmanned Aerial Vehicle (abbreviated as UAV), small-scale rotorcraft robotic systems (defined as lighter than $25\,kg$ or smaller than $10\,m$ in any direction) in particular, has been increasingly extended in both scientific and commercial fields [1]. In the commercial field are used in applications such as: photography; remote sensoring of agricultural products; disaster monitoring as forest fire monitoring; search and rescue; transportation such as urgent delivery of medicines [2]; border monitoring and environmental surveillance; transmission line inspection; and plant assets inspection (Figure 1.1). Among the advantages of using UAVs are: the reduction of risk tasks for human operators replacing them in the direct task through tele-operated UAV [4], automated task or through the implementation of AI (Artifi-

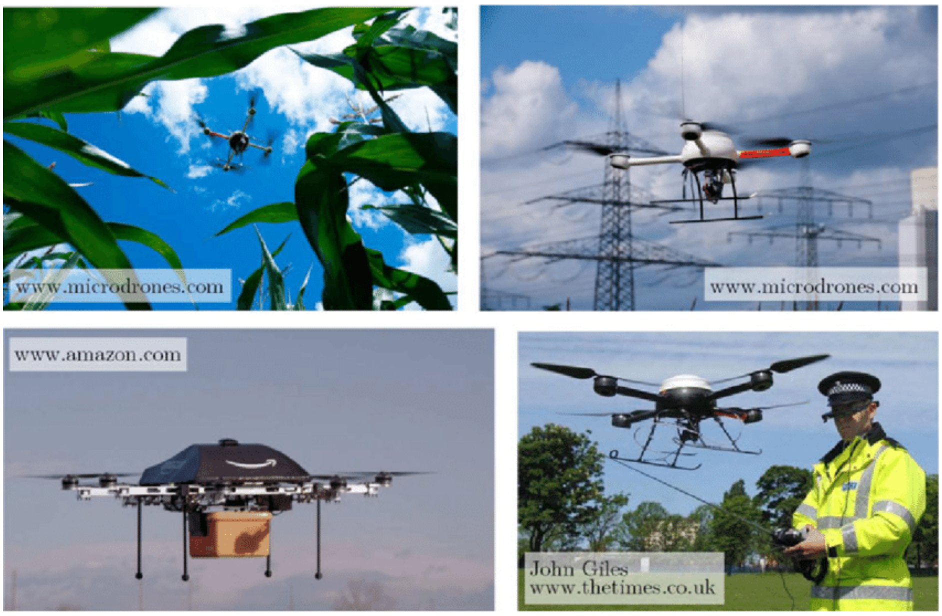


Figure 1.1: some UAV applications: monitoring of crops, inspection of power lines, transport of packages or police air reconnaissance are just a few examples [3]

cial Intelligence); ease of operation in tight and hard-to-reach environments; speed in reaching the required places; and less expensive reducing the need for specialised operators also from the point of view of safety for hazardous tasks. For instance, UAVs for pipeline inspection has become the standard practice for onshore and offshore platforms [5]. In the above-mentioned applications physical interaction with the surrounding environment is strictly avoided. To cross this limit a new area of scientific research, known as aerial manipulation, is born. Aerial manipulation is creatively used in construction, valve turning, tool operation, unknown environment sensing, bulb screwing, aerial writing, and object assembling. The physical interaction of the UAV with its surrounding environment falls within a well-studied broad research category known as mobile manipulation. However, most of the research carried out in mobile manipulation focuses on ground robots. The main distinct challenges in the aerial manipulation problem are [6]:

- UAVs do not have a stable base, therefore the movement of the manipulation mechanism and/or the payload generates forces and torques affecting the vehicle's position, attitude and even its stability;
- UAVs' propulsion system vary in close vicinity of the ground and/or walls;
- UAVs are often under-actuated platforms with highly nonlinear coupled dynamics, introducing further complications into their control design;
- UAVs stringent payload weight constraints avoid the accommodation of industrial dextrous robotic manipulators.

The above challenges have encouraged the development of a new research theme for the aerial manipulation problem, and articles like [7], [6], [8], and [9] summarize the state of the art of the aerial manipulation. Different configuration of manipulating devices have been studied and developed to optimize specific aerial manipulation tasks, they are divided in four main categories [6]:

1. Gripper for perching and load transportation;
2. Robotic manipulator for force/torque exertion and pick and place tasks other than perching and load transportation;

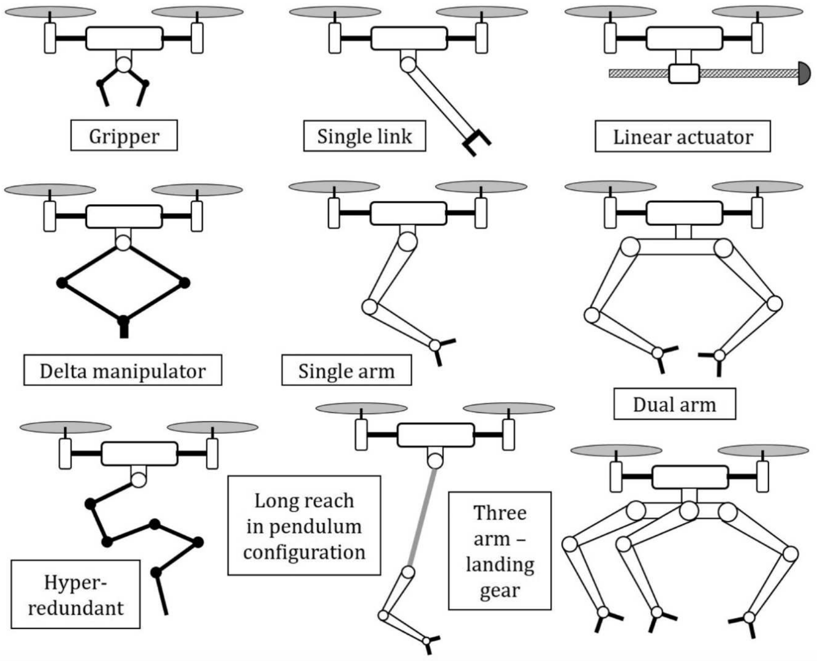


Figure 1.2: different morphologies of aerial manipulation robots [7]

3. UAV body or rigid tool for force/torque exertion;

4. Cable and tether for load transportation.

In particular, the manipulation with robotic manipulators is the more promising one which use different configurations (Figure 1.2) and redundancy, due to the possible high number of degrees of freedom (DoF), to task like drone stabilizing by compensating forces/torques transferred on it (with compensation for the center of gravity displacements [10], [11] or by minimizes the reaction torques from the manipulator to the drone [12]).

The use of additional sensors and control systems such as the PID (Proportional Integral Derivative), or the more sophisticated LQR (Linear Quadratic Regulator), VPIB (Variable Parameter Integral Backstepping), TLC (Trajectory Linearization Control), for the attitude control of the UAV do not ensure high accuracy and repeatability. Therefore, with this thesis will be analyzed static and dynamic methods for the reduction of UAV disturbances generated by the manipulator expanding the work done in [13] and [14]. This thesis is based on the free-floating base manipulator simulator for its kinematics and dynamics developed in [15], adapted to aerial manipulation in [12].

1.2 Unmanned Aerial Vehicles

Typically, an aerial manipulator system consist of two main physical subsystems, a UAV platform and a manipulator mechanism. In this section are described the most common UAV platforms, why has been chosen a multicopter for this thesis, the multicopter dynamic equations and its control system.

1.2.1 Classifications and Uses

Due to the great utilization and development of UAVs have been created many different types, and they are classified according to weight, ranging from *Micro* (under $2\,kg$) to *Large* (over $150\,kg$), and or according to the structure categorizable in [16]:

1. Fixed-wing drones: uses a wing like normal airplanes and have high fuel efficiency because they do not use energy to stay afloat so also high average flying time of about couple of hours or even more than 10 hours if they are powered by liquid fluid or gas. They move forward and are ideal for long-distance operations such as surveillance or mapping. Downsides of this drones are that they intrinsically require to maintain a minimum forward velocity to stay airborne, require higher skill for flying, need a runway or catapult for the launch, and need a runway, parachute or net for landing.

2. Multirotor drones: are the most heavily used ones and are classified by the number of rotors in a platform (*Bicopters* with two-rotors, *Tricopters* with three-rotors, *Quadcopters* with four-rotors, *Hexacopters* with five-rotors, and *Octocopters* with eight-rotors). They allow vertical take-off and the Quadcopter is the most widely-used one due to its mechanical design simplicity, hovering capabilities, low-cost, agility and precise control schemes which makes it ideal for the aerial manipulation. Their downsides are the relatively low speed, limited endurance and flying time (typically about $20 - 30\,min$)

3. Single-rotor helicopter: has the same structure of an helicopter, that is a big-sized rotor with a smaller one on the tail. It is more efficient than multi-rotor drones and usually has higher flying times. Its downsides are risk of injuries from the large-size blades (of high cost), and require more skill to fly than that required to

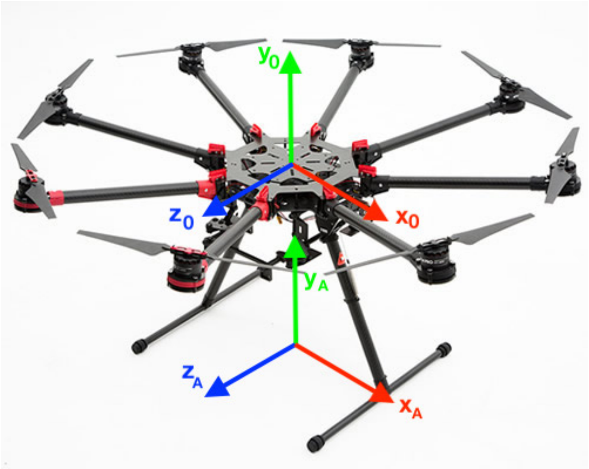


Figure 1.3: Reference frames for UAV model (DJI S1000). The inertial frame is denoted with $\sum_A$ (Absolute) and UAV base-fixed frame with $\sum_0$

fly a multirotor.

4. Fixed-wing hybrid VTOL (Vertical Take Off & Landing): has the best of both features from rotor-based and fixed-wing design, but they are still in the experimental or trial stage and are less commercially available than the other dornes.

Also Lighter-Than-Air (LTA) vehicles and their hybrid UAV (quadcopter + airship) have been used in aerial manipulation [17] although are not frequently used due to their low payload-to-volume ratio, high air resistance and sensitivity to aerodynamic disturbances [6].

1.2.2 UAV Used and its Mathematical Model

For the reasons described in subsection 1.2.1 the drone utilized in this thesis is the DJI S1000 [18]. The DJI S1000 is an Octocopter thus with a structure which consists of eight propellers (and each of these rotates in opposite direction respect the previous one to zero the gyroscopic effects) mounted at the end of eight equally spaced arms of equal length. These arms are linked to the central body of the drone, and below it there is the battery which powers the drone. The drone allows a stable support on the ground thanks to the retractable landing gear. The image of the drone with its reference frames is .

The Octocopter is considered an under-actuated system because the complete control of the six DoF in the space during the flight is obtained using only four control channels as can be seen in Figure 1.4. Each of them does [13]:

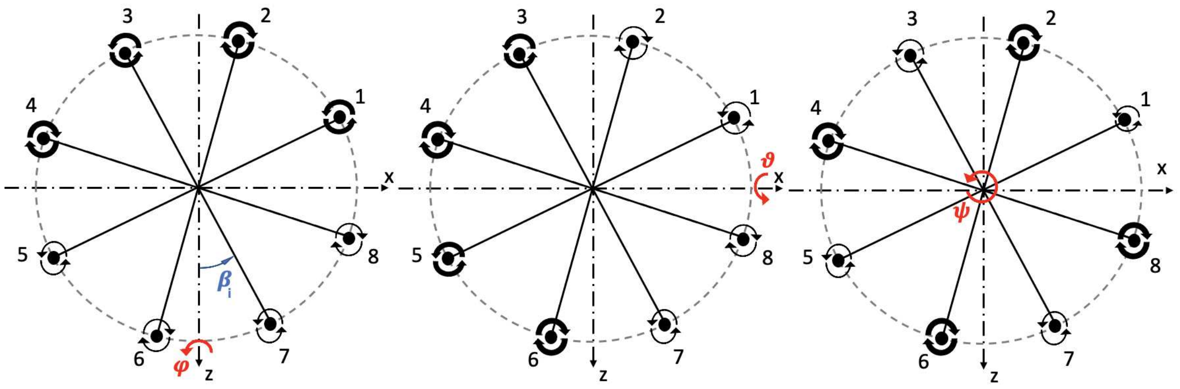


Figure 1.4: drone motors controls to perform roll, pitch, yaw, respectively. The red arrows represent the directions of rotation of the drone, while the black arrows represent the direction of rotation of the propellers, the thicker ones indicate a higher rotation speed

1. Total thrust, increase and decrease equally the power of all the motors to vary the altitude;

2. Roll φ, increasing or decreasing the speed of motors 1, 2, 3, 4 respect those of motors 5, 6, 7, 8 to create a torque along the UAV's local z-axis (reference $\sum_0$);

3. Pitch ϑ, increasing or decreasing the speed of motors 3, 4, 5, 6 respect those of motors 7, 8, 1, 2 to create a torque along the UAV's local x-axis (reference $\sum_0$);

4. Yaw ψ, increasing or decreasing the speed of motors rotating clockwise respect those rotating counter clockwise to create a torque along the UAV's local y-axis (reference $\sum_0$);

The translation of the drone can not be performed without one or more of these rotations.

For simplicity in this thesis the drone and the manipulator move in the absolute xy plane (reference $\sum_A$) defined in subsection 1.2.2. Therefore the vertical thrust explained above will be along the local y-axis (reference $\sum_0$) and will be named U_1, and the only rotation of the drone will be the roll φ (about the UAV's CoG) controlled by the roll torque named U_2.

The dynamic model of the UAV in the xy plane is described by the following equations [19]:

$$m_0\ \ddot{x} = U_1\ sin(\varphi) \tag{1.2.1}$$

$$m_0\ \ddot{y} = U_1\ cos(\varphi) - m_0\ g \tag{1.2.2}$$

$$I_0\ \ddot{\varphi} = U_2 \tag{1.2.3}$$

where m_0 and I_0 are the UAV mass and the rotational inertia about local z-axis (reference $\sum_0$) of the UAV, respectively. With U_1 and U_2 described by:

$$U_1 = b(\Omega_1^2 - \Omega_2^2 + \Omega_3^2 - \Omega_4^2 + \Omega_5^2 - \Omega_6^2 + \Omega_7^2 - \Omega_8^2 \tag{1.2.4}$$

$$\begin{aligned} U_2 = b[&sin(\beta_1)\Omega_1^2 + sin(\beta_2)\Omega_2^2 + sin(\beta_3)\Omega_3^2 + sin(\beta_4)\Omega_4^2 \\ &+sin(\beta_5)\Omega_5^2 + sin(\beta_6)\Omega_6^2 + sin(\beta_7)\Omega_7^2 + sin(\beta_8)\Omega_8^2] \end{aligned} \tag{1.2.5}$$

where Ω_i with $i = 1, ..., 8$ is the rotational speed of each motor, β_i with $i = 1, ..., 8$ is the angle between the local x-axis (reference $\sum_0$) and the i-th arm, and b is the lift coefficient of the propellers. The UAV is considered as a rigid body in these equations and the aerodynamic effects are simplified with the lift coefficient b. The U_1 is always considered parallel to the local y-axis (reference $\sum_0$).

The inertial parameters utilized in this thesis comes from the structure of the DJI S1000 and are:

parameter	value	unit of measure
m_0	4.2	$[kg]$
I_0	0.4097	$[kg\ m^2]$

Table 1.1: DJI S1000 inertial parameters

1.2.3 Control Methods

In the research literature there is a vast amount of control algorithms applicable to UAVs, e.g. Proportional-Derivative-Integral (PID) which will be used in this thesis, Linear Quadratic Regulator (LQR), H∞, sliding mode variable structure, backstepping and adaptive control. For simplicity, the UAV controls in the xy plane here studied use a PID control, which is set according to the following equations:

$$U_1 = -k_{P_y}\ y - k_{D_y}\ \dot{y} - k_{I_y}\ \int_0^t y\ dt \tag{1.2.6}$$

$$U_2 = -k_{P_\varphi}\ \varphi - k_{D_\varphi}\ \dot{\varphi} - k_{I_\varphi}\ \int_0^t \varphi\ dt \tag{1.2.7}$$

where $k_{P\_}$, $k_{D\_}$, $k_{I\_}$ are the proportional, derivative and integral coefficients for the controller for subscript coordinates y and φ. These coefficients values are showed in and come from [14] after load picking simulations with MatLab and Adams.

	k_P	k_D	k_I
altitude	37 $[N/m]$	18 $[N/(m\ s)]$	8 $[N\ s/m]$
roll	40 $[N/rad]$	3 $[N/(rad\ s)]$	35 $[N\ s/rad]$

Table 1.2: PID parameters for altitude and roll

1.3 UAM control

The manipulation mechanisms are mainly: robotic manipulators, grippers, rigid tools, and tethers. The UAVs' under-actuation anticipated in the beginning of section 1.1 in the case of quadcopters can be complemented with a robotic arm with additional DoF to obtain a 6 DoF control of the aerial manipulator end-effector. Furthermore, with redundant and hyper-redundant manipulators can be achieved higher reliability, optimization of a secondary given task, and access to hard-to-reach locations. The secondary task primarily addressed in this thesis is the minimization of the base movement given by the link movements developing what was set out in [19, 12].

The primary control objective of the UAM is to drive the manipulator end-effector a desired position/trajectory to accomplish a task. The position control of such non-linear system can be categorised in *decoupled* and *coupled* based on the dynamic model [6]:

- in the decoupled control design the UAV and the manipulator are treated as separate subsystems. The effects of the manipulator on the UAV are treated as disturbances, thus not taken into account in the controller design. This approach performs well in simple scenarios, but taking account of the UAM's CoG movement from its geometric center in the control design the performances improve [20];

- in the coupled approach the UAV and manipulator are controlled as a unified system. The UAM dynamic model take into account the non-linear coupled dynamics of the UAV and manipulator which results a more accurate control. This type of controller design is theoretically difficult and its implementation depends on the onboard computational resources of the system.

1.3.1 Direct and Inverse Kinematics

A manipulator consists of rigid bodies called links connected to each other with kinematic pairs called joints (which the most common ones are *revolute* and *prismatic*, also used in this thesis).

The aim of the direct kinematics is that of retrieve the pose of the end-effector $\vec{x}_e = [\vec{p_e}^T \quad \vec{\phi_e}^T]^T$ $(m \times 1)$(where $\vec{p}_e$ and $\vec{\phi}_e$ are the vector of the end-effector position and orientation, respectively, $m = 6$ in the space) starting from the joint variables $\vec{q} = [q_1, q_2, \ldots, q_n]^T$ $(n \times 1)$(vector of the joint variables for n number of joints). The direct kinematics equation can be written as:

$$\vec{x}_e = \vec{k}(\vec{q}) \tag{1.3.1}$$

where $\vec{k}(\cdot)$ $(m \times 1)$ is a vector function, typically non-linear.

The inverse kinematics problem concerns the determination of joint variables to obtain a given end-effector position and orientation. In order to transform the motion assigned to the end-effector from the operational space into the corresponding joint space. This problem is more complex than the that of the direct kinematics because:

- The equation to solve typically are non-linear, so may not be a closed-form solution;
- May have multiple solutions;
- May have infinite solutions, as for a kinematically redundant manipulator;
- Might be solutions not acceptable for the kinematic and mechanic structure of the manipulator

2. Static Balancing Systems

The two types of forces and torques exerted by the movement of the manipulator on the drone, which act as disturbances on the flight of the drone, are the *gravitational* one and the *dynamic* one. Slower is the movement of the arm smaller will be the dynamic effects respect the gravitational effects. Therefore, typically the dynamic effects are negligible. In this section will be discussed the types of mechanisms apt to compensate only this gravitational effect. For simplicity, all the thesis will treat movements and forces in the x-y plane (where the y axis is directed as the gravitational acceleration $\vec{g}$ but in opposite verse, and the x axis is perpendicular to it by definition) and so the focus will be only on the torque of the drone's roll axis.

These mechanisms are called *static balancing mechanisms* because they will compensate only the torque on the drone generated by the gravity acting on the manipulator. This torque will remain the same whether the arm is stationary, moving at constant speed or accelerating. This is why it is called static balancing.

To counteract this disturbance given by the manipulator on the drone will be used torques generated by the force of gravity on a counterweight element. In practice the position of this counterweight element is calculated before the joint positions of the manipulator, which is based on the configuration of the manipulator at the previous calculation step. The general rule is to calculate the center of gravity (it will henceforth be abbreviated as CoG) of the manipulator at the previous calculation step and to give the position of the counterweight apt to generate a torque equal and opposite to that of the manipulator. In the positioning of the counterweight the orientation of the UAV, that is only defined by its roll angle in the two dimensional simulations, will be considered.

2.1 Initial UAM model

The trajectory of the manipulator is imposed in the Cartesian plane, as the trajectory of the end-effector position, and it will remain always the same to allow a simple comparison between the following methods. The two trajectories which will be imposed to the end-effector are shown in Figure 2.1a and Figure 2.1b (the starting position in the x-y absolute plane will be kept the same as possible for all the simulations in order to not change too match the manipulator configuration to compare the results under the same conditions). In order to compare the effectiveness of the methods that will be covered forward is reported results with the simulation of the simple 2 DoF (Degrees of Freedom) manipulator mounted on the CoG of the drone (to minimize the disturbances on the drone by the manipulator). In Figure 2.2 and Figure 2.3 are showed meaningful results about the simulations for an imposed end-effector trajectory of 19 s and 10 s, respectively, with the same settings. In particular, in Figure 2.2a and Figure 2.3a are showed the UAM in the last configuration of the two simulations, then are showed the UAV roll angle, UAV translation, and the controls of the UAV during the two simulations (where U_1 is the force and U_2 is the torque exerted on the UAV by its propellers) in Figure 2.2b and Figure 2.3b, Figure 2.2c and Figure 2.3c, Figure 2.2d and Figure 2.3d, respectively. All these parameters will be used in the subsequent simulations to comparison and analysis and will have the same structure:

1. the first figure will represent the final configuration of the UAM in the x-y plane (reference $\sum_A$) in meters where typically the green line symbolises the outline of the UAV; black straight lines represent the single links; black circles represent the joint positions; red circles represent the CoGs of joints; magenta circle represent the end-effector position; red asterisk represents the UAM's CoG; magenta curve represents the actual path taken by the end-effector during the simulation; the blue curve represents the actual path taken bt the UAV's CoG; and the vertical black dash-dot line facilitate the identification of the UAV's CoG and its alignment with the UAM's CoG;

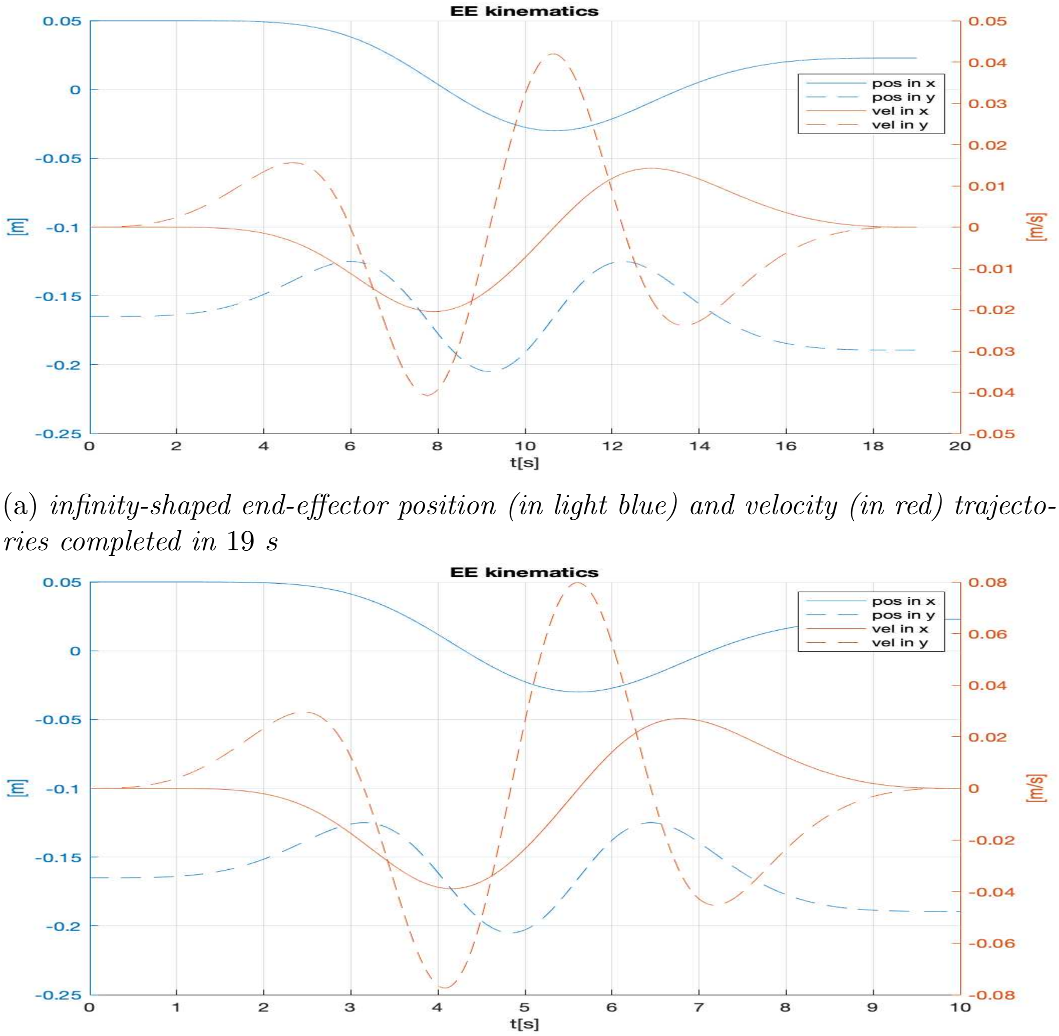
2. the second figure will show the displacements of the UAV's CoG along the there absolute axes (reference $\sum_A$) in meters, obviously the displacement in the z axis will be always zero;

3. the third figure will show the orientation of the UAV, in particular only the roll angle in degrees;

4. the fourth figure will show the force U_1 and torque U_2 generated by the controls of the UAV in Newton and Newton · meters, respectively;

5. the last two figures will show the positions (in radians for revolute joints and in meters for translational joints) and velocities (in radians over second for revulte joints and in meters over second for translational joints) of all the joints in the system over time (in seconds) during the simulation.

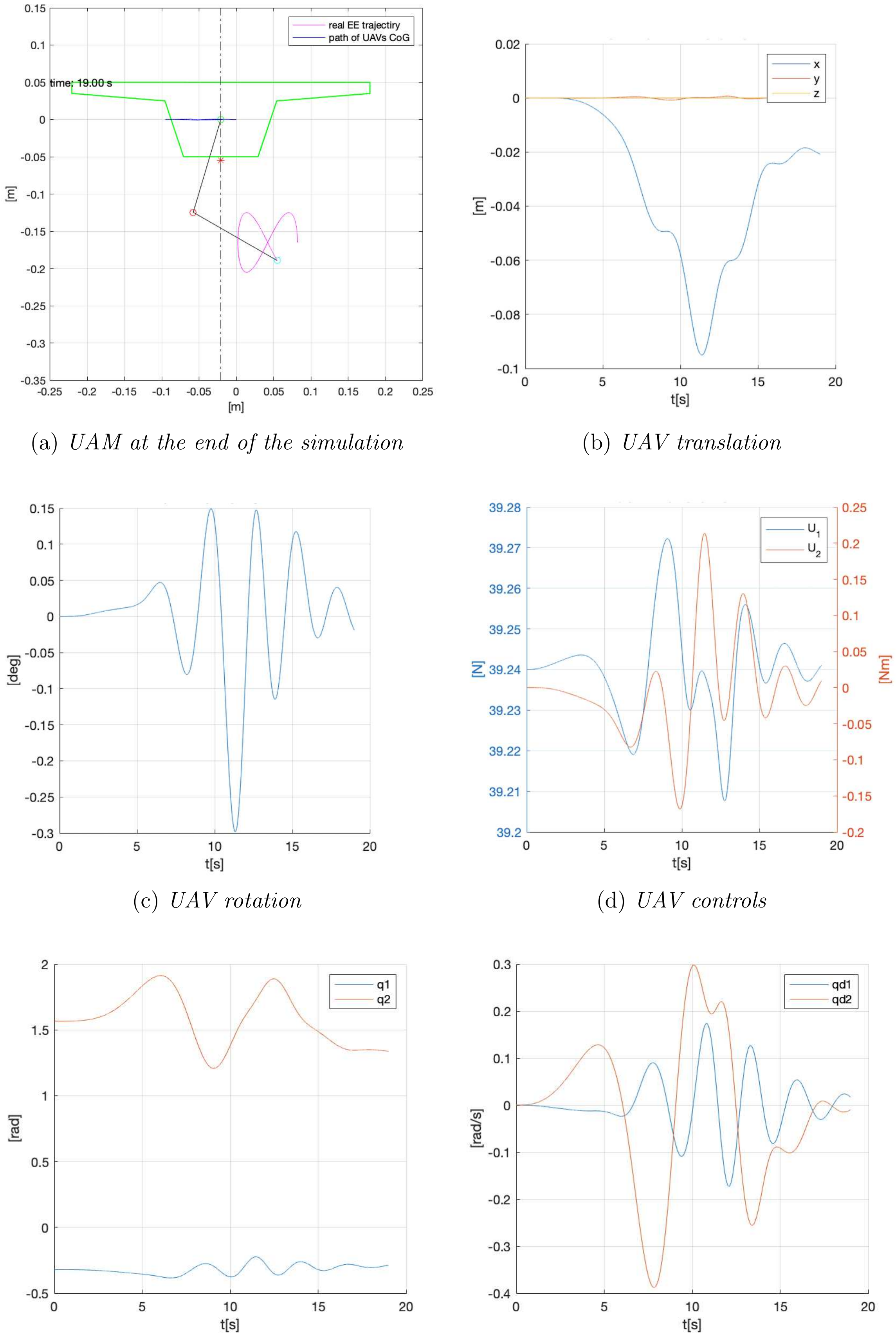
The simulations in the Adams View® environment are showed in Figure 3.6 and confirm the results of simulations performed in MatLab®.



(a) *infinity-shaped end-effector position (in light blue) and velocity (in red) trajectories completed in* 19 *s*

(b) *infinity-shaped end-effector position (in light blue) and velocity (in red) trajectory completed in* 10 *s*

Figure 2.1: chosen end-effector trajectories for comparison of simulations



(a) *UAM at the end of the simulation*

(b) *UAV translation*

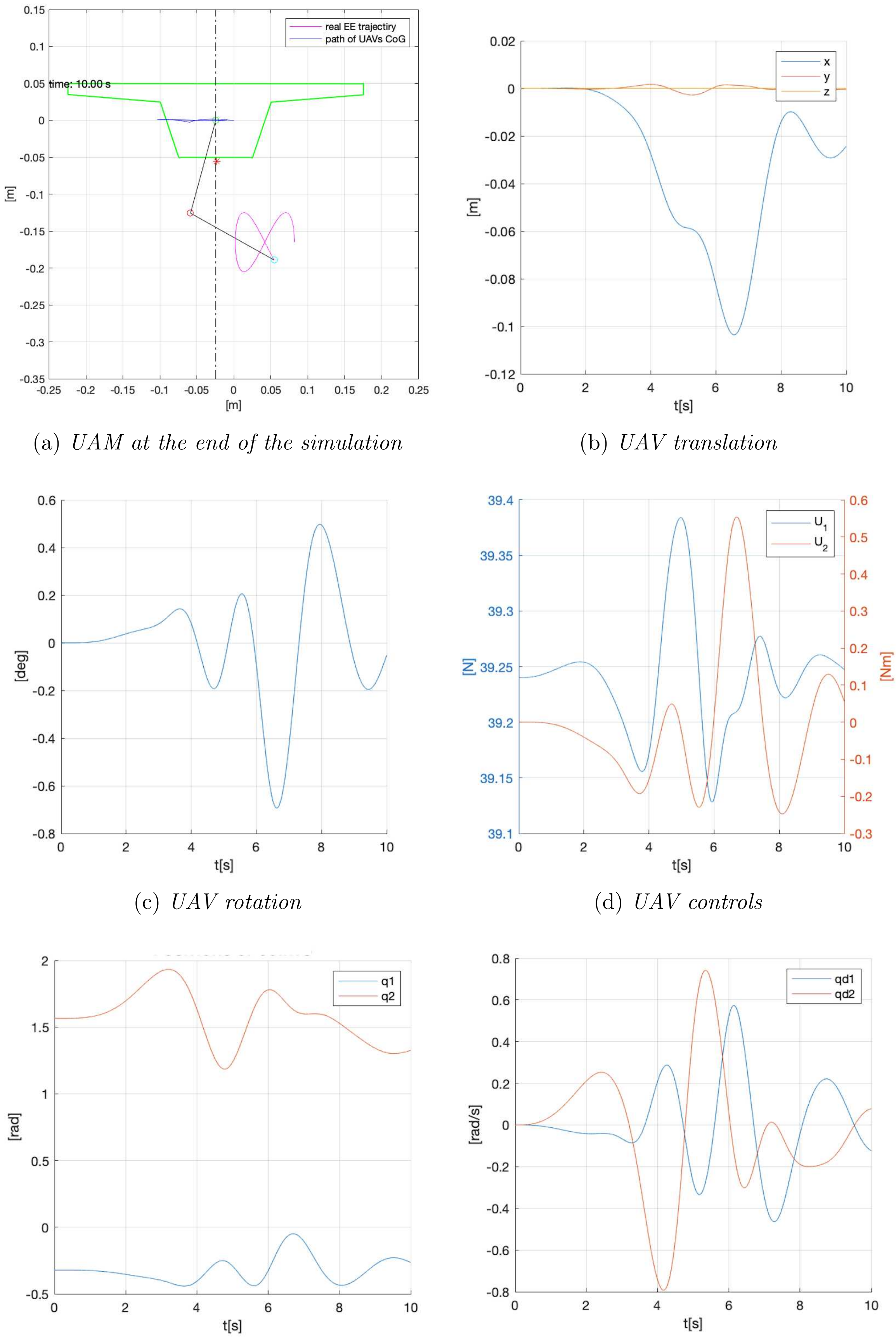
(c) *UAV rotation*

(d) *UAV controls*

(e) *manipulator's joint positions*

(f) *manipulator's joint velocities*

Figure 2.2: results of UAM without extra balancing methods for an infinity-shaped end-effector trajectory in 19 s



(a) *UAM at the end of the simulation*

(b) *UAV translation*

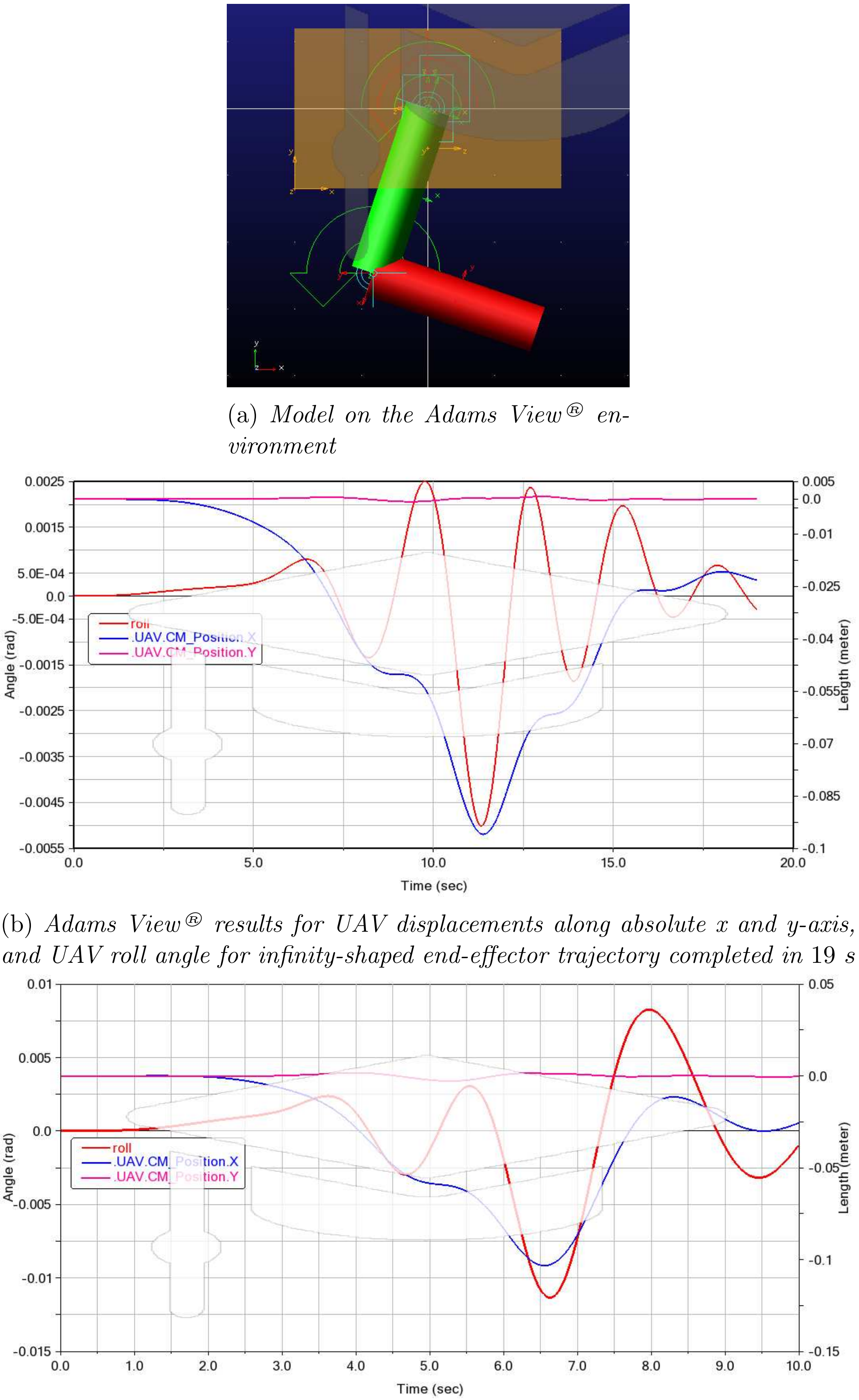
(c) *UAV rotation*

(d) *UAV controls*

(e) *manipulator's joint positions*

(f) *manipulator's joint velocities*

Figure 2.3: results of UAM without extra balancing methods for an infinity-shaped end-effector trajectory in 10 s



(a) *Model on the Adams View® environment*

(b) *Adams View® results for UAV displacements along absolute x and y-axis, and UAV roll angle for infinity-shaped end-effector trajectory completed in* 19 *s*

(c) *Adams View® results for UAV displacements along absolute x and y-axis, and UAV roll angle for infinity-shaped end-effector trajectory completed in* 10 *s*

Figure 2.4: Simulations on Adams View ® of UAM without extra balancing methods

2.2 Sliding Battery UAM model

The simplest mechanism to compensate the torque generated by gravity on the manipulator on the UAV is by translate the battery of the UAV as did in [21] and in [14]. The battery powers the UAV and is responsible for the flight autonomy which is typically a restrictive constrain for its operation. The choice of the battery is a trade off between its capacity (greater it is and longer the flight autonomy is) and its weight (greater it is and lower the flight autonomy is and the reactivity due to the increase of the inertia). Loading the electric powered UAV with larger capacity batteries does not necessarily mean a longer endurance, because heavier weight will result in higher power demand that may offset the additional energy brought by larger capacity [22]. Therefore, take advantage of the mass of the battery, which is a necessary and heavy component (in particular, the mass of the battery is equal to $m_c = 2,2\ m$), to use it as counterweight will save an additional load. For simplicity the mechanism to move the battery will be prismatic joint horizontally to the UAV's local frame $\sum_0$.

To compensate the position of the manipulator's CoG is necessary to locate it with reference to the UAV's reference frame $\sum_0$, as:

$$
{}^{0}\vec{r}_G = \frac{\sum\limits_{i=1}^{n} m_i\ {}^{A}\vec{r}_i}{\sum\limits_{i=1}^{n} m_i} -\ {}^{A}\vec{r}_0 \tag{2.2.1}
$$

where m_i is the mass of the link i, ${}^{A}\vec{r}_i$ is the vector from the origin of the reference frame $\sum_A$ to the CoG of link i, and i (with $n = 2$) defines the number of manipulator's links.

Now is possible to retrieve the position of the battery in order to balance the manipulator's CoG, or to align vertically (along the y absolute axis $\sum_A$) the CoG of the manipulator plus the counterweight (the battery's CoG) with that of the UAV. The battery position (in its generalised coordinate q_c) will be:

$$
q_c = -\frac{\sum\limits_{i=1}^{n} m_i}{m_c}\ {}^{0}r_{G,x} \tag{2.2.2}
$$

where m_c is the counterweight mass (battery mass), and ${}^{0}r_{G,x}$ is the component in the x axis (first row) of the vector ${}^0\vec{r}_G$.

During the simulation the UAV will rotate varying its roll angle because the lack of a perfect control for the UAV. Thus, the generalised coordinate of the counterweight should be increased or decreased considering its new projection on the absolute x axis (reference $\sum_A$, in which the torque for the gravity is higher). The correct sliding battery generalised coordinate will be:

$$q_c = \frac{x_s}{cos(\phi)} + r_s \; tan(\varphi)$$

$$q_c = \frac{\left(-\sum_{i=1}^{n} m_i/m_c\right) {}^0r_{G,x}}{cos(\phi)} + r_s \; tan(\varphi) \tag{2.2.3}$$

where r_s is the fixed position of the prismatic joint of the sliding battery from the UAV's CoG along its relative y axis (reference $\sum_0$), x_s is the previous calculated q_c which considering the zero roll angle in Equation 2.2.2, and φ is the UAV roll angle.

Now can be determined the sliding battery generalised velocity to determine afterwards the generalised velocities of the manipulator's joints with the inversion of the *generalised Jacobian.* The generalised velocity of sliding battery is calculated with finite difference approximations:

$$\dot{q}_c = \frac{q_{c,k} - q_{c,k-1}}{\Delta t} \tag{2.2.4}$$

where k refers to the current numerical calculation step and $k-1$ to the previous one, and Δt is the chosen time step to the simulation. All simulations was ran with a $\Delta t = 10^{-3} s$

Now the velocities of the manipulator's joints can be determined with the generalised Jacobian, long standing in research literature presented in the article [23]. In this Thesis will be used a MatLab library able to simulate every type of manipulator with free-floating base in absence of external forces and torques. The procedure is similar to that showed in [24, 25]. Actually, here the base is subject to the external force of gravity and UAV's controls, hence will be used the approach described in [12].

One starts from Jacobian matrix $J(\vec{q}) \in \mathbb{R}^{2\times n}$ (with n as the DoF of the robotic arm) needed for the forward (and inverse) manipulator differential kinematics:

$$\vec{v}_e = J(\vec{q})\ \dot{\vec{q}} \tag{2.2.5}$$

where $\vec{v}_e$ is the velocity vector of the end-effector in Cartesian space coordinates (known quantity in the inverse kinematics), $\dot{\vec{q}}$ is the robot arm's joint velocities (unknown quantity in the inverse kinematics), and $J(\vec{q}) = \frac{\partial \vec{x}_e(\vec{q})}{\partial \vec{q}}$ is the Jacobian matrix of the forward manipulator kinematic equations $\vec{x}_e(\vec{q}) = \vec{p}_e \in \mathbb{R}^2$ in the Cartesian two-dimentional space (where $\vec{p}_e$ is the position of the end-effector with reference to the base of the manipulator $\sum_0$).

To find the Jacobian matrix of the system the differential kinematics has to consider not only the robotic arm (with fixed base), but the entire system handling the UAV platform as a 6 DoF robotic arm (with fixed base) with 3 prismatic joints and 3 rotational joints. So it is possible decoupled the component of the Jacobian matrix to write the forward manipulator differential kinematics equation as:

$$\begin{aligned} \vec{v}_e &= \begin{bmatrix} J_b & J_m \end{bmatrix} \begin{Bmatrix} \dot{\vec{q}_b} \\ \dot{\vec{q}_m} \end{Bmatrix} + \vec{v}_0 \\ \vec{v}_e &= J_b\ \dot{\vec{q}_b} + J_m\ \dot{\vec{q}_m} + \vec{v}_0 \end{aligned} \tag{2.2.6}$$

where $\dot{\vec{q}_b}$ and $\dot{\vec{q}_m}$ are the generalised velocities of the base (UAV platform sees as the 6 DoF robotic arm) and generalised velocities of the manipulator, respectively, J_b and J_m are the corresponding Jacobians, ans v_0 is the initial translational velocity of the end-effector. The kinematics equations of the system for velocity-space are defined in the CoG's reference frame of the entire system $\sum_S$, to which the inertias of the base and links are also referred [12]. Now is considered the momentum conservation for the system in the CoG's refrence frame $\sum_S$ and is decoupled:

$$\vec{L}_0 = H_b\ \dot{\vec{q}_b} + H_m\ \dot{\vec{q}_m} \tag{2.2.7}$$

where $\vec{L}_0$ is the momentum of the system at the initial time, H_b and H_m are the mass/inertia matrices of the base and manipulator in the CoG's frame , respectively.

Combining the Equation 2.2.6 with Equation 2.2.7 is possible to express the end-effector velocity in the Cartesian space v_e only as a function of manipulator generalised velocities $\dot{\vec{q}}_m$:

$$\begin{aligned} \vec{v}_e &= \underbrace{(J_m - J_b\ H_b^{-1}\ H_m)}\ \dot{\vec{q}}_m + \underbrace{J_b\ H_b^{-1}\ \vec{L}_0 + \vec{v}_0} \\ \vec{v}_e &= \qquad J^* \qquad\quad \dot{\vec{q}}_m + \qquad \vec{p}_0 \end{aligned} \tag{2.2.8}$$

where the element $J_b\ H_b^{-1}\ \vec{L}_0 + \vec{v}_0 = \vec{p}_0$ and J^* is the *generalised Jacobian.* Therefore, the inverse kinematic to find the generalised velocities of the robotic arm to follow the imposed end-effector trajectory that take account of the base movements become:

$$\dot{\vec{q}}_m = J^{*-1}\ (\vec{v}_e - \vec{p}_0) \tag{2.2.9}$$

If the UAM is working in absence of forces and torques its linear and angular momenta will preserve $\vec{L}_0 = cost$, but if it is working under gravity, forces and torques generated by its propellers its linear and angular momenta will change continuously:

$$\vec{L}(t) = \begin{Bmatrix} \vec{P}(t) \\ \vec{K}(t) \end{Bmatrix} \tag{2.2.10}$$

$$\begin{aligned} \vec{P}(t) &= \int_{t_0}^{t} \vec{F}(t)\ dt + \vec{P}(t_0) \\ \vec{K}(t) &= \int_{t_0}^{t} \vec{\tau}(t)\ dt + \vec{K}(t_0) \end{aligned} \tag{2.2.11}$$

where $\vec{P}(t) = \begin{bmatrix} P_x(t) & P_y(t) & P_z(t) \end{bmatrix}^T$ is the vector of the linear momenta which changes continuously because of the forces applied $\vec{F}(t) = \begin{bmatrix} F_x(t) & F_y(t) & F_z(t) \end{bmatrix}^T$ on the absolute reference frame $\sum_A$ with $\vec{P}(t_0)$ as the initial linear momentum, $\vec{K}(t) = \begin{bmatrix} K_x(t) & K_y(t) & K_z(t) \end{bmatrix}^T$ is the vector of the angular momenta which changes continuously because of the torques applied $\vec{\tau}(t) = \begin{bmatrix} \tau_x(t) & \tau_y(t) & \tau_z(t) \end{bmatrix}^T$ on the absolute axes with $\vec{K}(t_0)$ as the initial angular momentum. In the simulation this update of the linear and angular momenta happens in a discretised manner every timestep Δt

(sampling time interval) so Equation 2.2.11 become:

$$\begin{aligned} \vec{P}_{k+1} &= \vec{F}_k \; \Delta t + \vec{P}_k \\ \vec{K}_{k+1} &= \vec{\tau}_k \; \Delta t + \vec{K}_k \end{aligned} \tag{2.2.12}$$

where k is the current calculation step. To find the linear momentum of the system $\vec{P}_{k+1}$ (for each calculation step k) under the external forces as stated in Equation 2.2.12 will be used the equation:

$$\vec{P}_{k+1} = \vec{P}_k + \left(A_0 \; \vec{U}_1 + \begin{Bmatrix} 0 \\ m_T \; g \\ 0 \end{Bmatrix} \right) \Delta t \tag{2.2.13}$$

where A_0 is the rotation matrix of the UAV, $\vec{U}_1$ is the thrust generated by the propellers of the drone in y axis relative to the UAV (reference frame of the drone $\sum_0$) that change orientation with the rotation of the UAV, and m_T is the total mass of the UAM. Whereas to find the angular momentum of the system $\vec{K}_{k+1}$ (for each calculation step k) under the external forces and torques as stated in Equation 2.2.12 will be used the equation:

$$\vec{K}_{k+1} = \vec{K}_k + \left(-\vec{v}_{b,k-1} \times \vec{P}_k + \begin{Bmatrix} 0 \\ 0 \\ U_2 \end{Bmatrix} + (\vec{r}_G - \vec{r}_B) \times \begin{Bmatrix} 0 \\ m_T \; g \\ 0 \end{Bmatrix} \right) \Delta t \tag{2.2.14}$$

where $\vec{v}_{b,k-1}$ is the vector of linear velocities of the base (first 3 rows of $\dot{\vec{q}_b}$) at calculation step $k-1$, assuming the velocities do not change significantly between two timesteps.

The momentum equation in Equation 2.2.10 can be expressed in terms of the mass/inertia matrix of the system $H \in \mathbb{R}^{6\times(6+n)}$, base generalised coordinates $\dot{\vec{q}_b} \in \mathbb{R}^6$, and manipulator generalised coordinates $\dot{\vec{q}_m} \in \mathbb{R}^n$ as has been done also in Equation 2.2.7:

$$H \begin{Bmatrix} \dot{\vec{q}_b}(t) \\ \dot{\vec{q}_m}(t) \end{Bmatrix} = \begin{Bmatrix} \vec{L}(t) \\ \vec{K}(t) \end{Bmatrix} \tag{2.2.15}$$

the matrix H will be partitioned into base component $H_b \in \mathbb{R}^{6\times 6}$ and manipulator component $H_m \in \mathbb{R}^{6\times n}$ as $H = \begin{bmatrix} H_b & H_m \end{bmatrix}$. So the end-effector velocity retrieved in

Equation 2.2.8 now can take into account the variation of angular momenta replacing $\vec{L}_0$ into $\vec{L}$.

Now can be calculated the generalised velocities of the manipulator, and since the generalised velocity of the counterweight $\dot{q}_c$ has been previously determined in Equation 2.2.4 and the Jacobian matrix has to be a square matrix to be inverted, the following *extended Jacobian* will be used:

$$J_{ext} = \begin{bmatrix} \begin{bmatrix} J^* \end{bmatrix} \\ 0 \quad 0 \quad 1 \end{bmatrix} \tag{2.2.16}$$

where J^* is the generalised Jacobian of the UAM with the battery static balancing method. And the related output vector of the inverse kinematic, the *extended velocity vector*, will become:

$$V_{ext} = \begin{Bmatrix} v_{e,x} - p_{0,x} \\ v_{e,y} - p_{0,y} \\ \dot{q}_c \end{Bmatrix} \tag{2.2.17}$$

In Figure 2.6b, Figure 2.7c, Figure 2.6c and Figure 2.7d are noticeable the improvement of the static balancing method with the sliding battery. The rotation of the UAV was reduced at least 40 times in the slow trajectory and at least 27 times in the fast trajectory and so the control of the UAV (the U_2 torque) intervenes less. A further advantage of this method is that it allows the position of the CoG of the system manipulator + battery to be kept along the absolute x-axis (reference $\sum_A$), thus also keeping the CoG of the UAV in the same initial position as can be seen comparing Figure 2.2b (and Figure 2.3b) with Figure 2.6a (and Figure 2.7b). Other simulation were carried out to understand the effect of the parameter r_s on the UAM behaviour. In particular, configurations with r_s equal to: 0.06 m; 0.03 m; 0.01 m; 0 m; -0.01 m; -0.03 m and -0.06 m were tested, in which the positive sign points towards the sky and the negative sign towards the ground. The results showed significant improvements in reducing the UAV roll angle (while keeping all other parameters constant) by reducing r_s. This phenomenon might be due to dynamic effects. In particular, the lower values of the UAV roll angle might be due to the fact that if the CoG of the manipulator moves away from the CoG of the UAV, typically the first generalised coordinate of the manipulator q_1 will rotate clockwise, generating a change in the angular momentum on the CoG of the UAV and thus a torque in the opposite direction (counter clockwise). Therefore, in this scenario, with the sliding battery, moving the battery away from the CoG of the UAV to balance the gravitational effect (so in the opposite direction of the of manipulator's CoG), will be generated a torque on the CoG of the UAV due to the inertia of the battery in the opposite direction, counter clockwise if r_s is positive, and clockwise if r_s is negative. The more negative r_s is, the greater the torque will be as the lever arm increases.

For this reason a negative r_s was chosen to perform the following simulations, with a value of 0.03 m in order to find a compromise between good results and encumbrance in the movement of the manipulator.

The simulations in the Adams View$^{®}$ environment are showed in Figure 2.8 and confirm the results of simulations performed in MatLab$^{®}$.

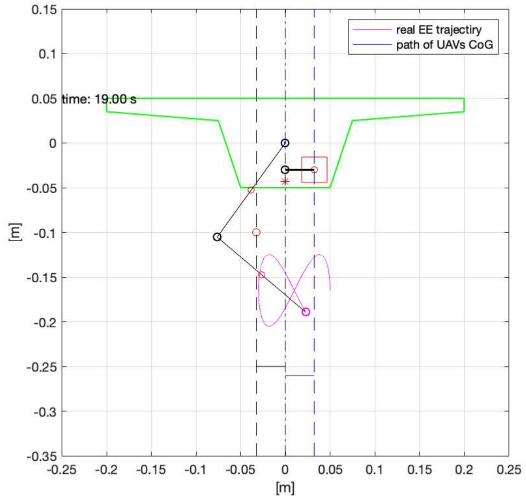
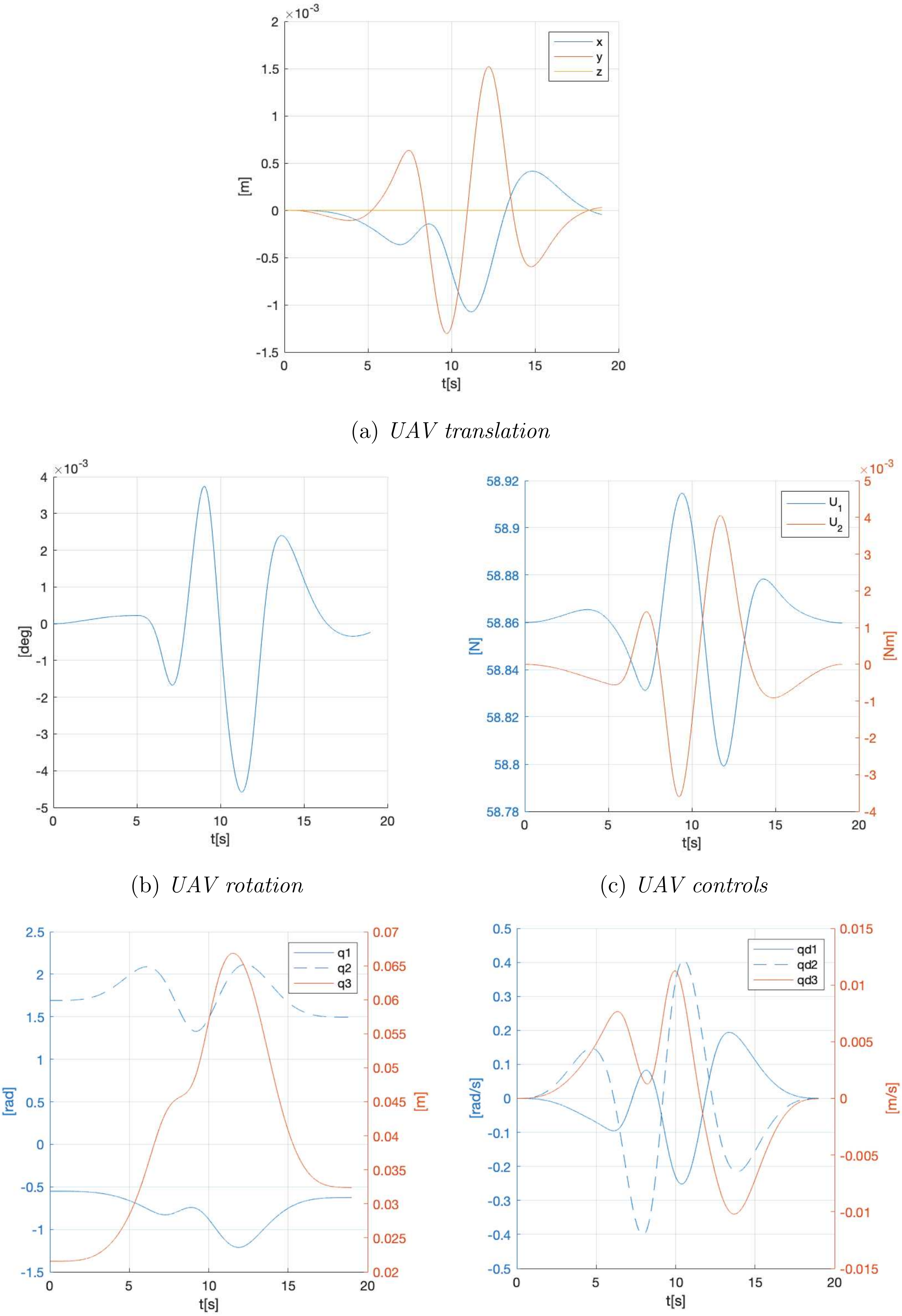


Figure 2.5: UAM at the end of the simulation for an infinity-shaped end-effector trajectory in 19 s, where the blue and the black vertical dashed lines represent the projection of the counter-weight and manipulator CoGs with the relative horizontal lines which connect them to the UAV's CoG, respectively, and red square represents the sliding battery with its prismatic joint connected to the UAV



(a) *UAV translation*

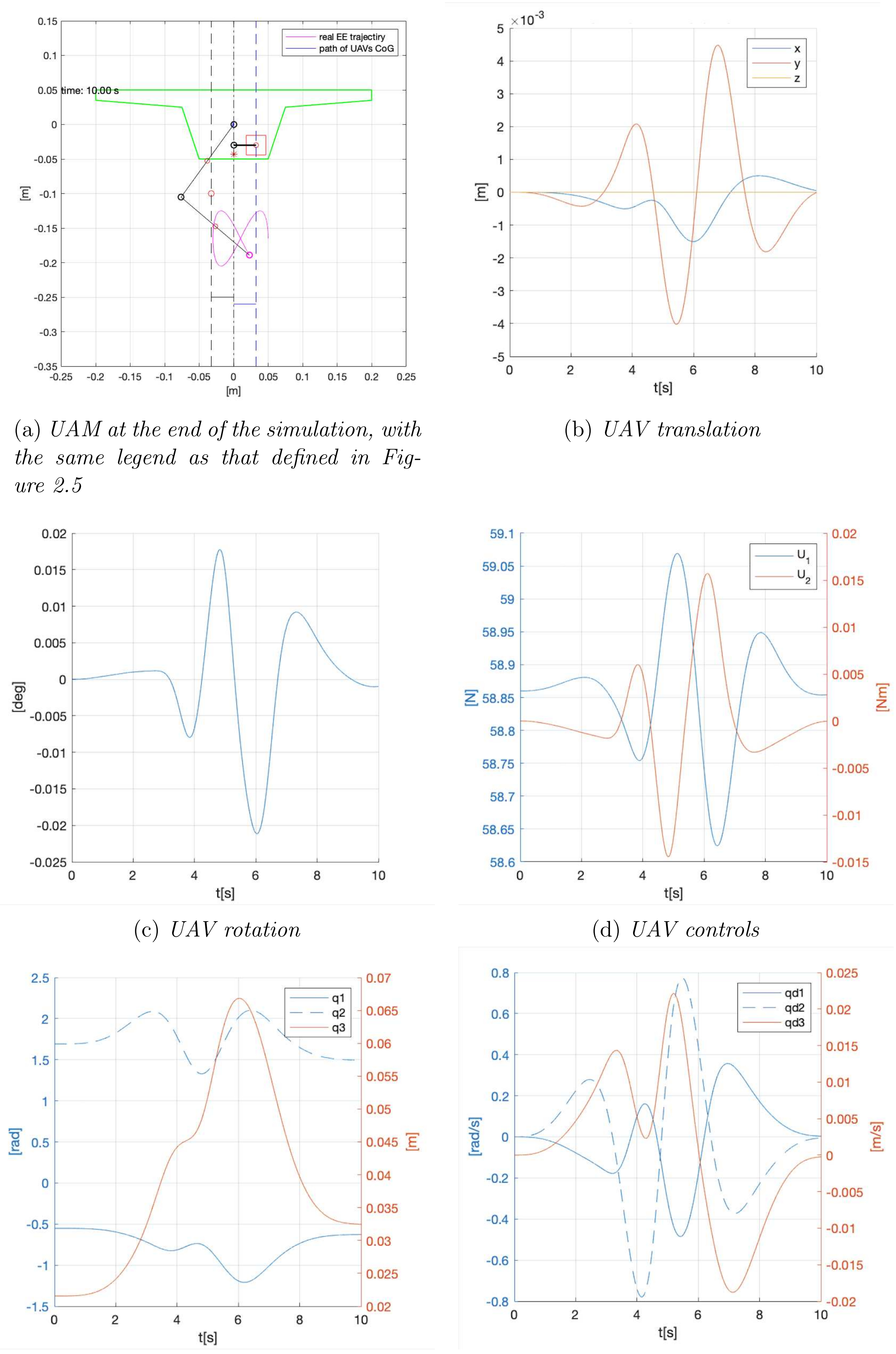
(b) *UAV rotation*

(c) *UAV controls*

(d) *manipulator's joint positions*

(e) *manipulator's joint velocities*

Figure 2.6: results of UAM with static balancing sliding battery for an infinity-shaped end-effector trajectory in 19 s



(a) *UAM at the end of the simulation, with the same legend as that defined in Figure 2.5*

(b) *UAV translation*

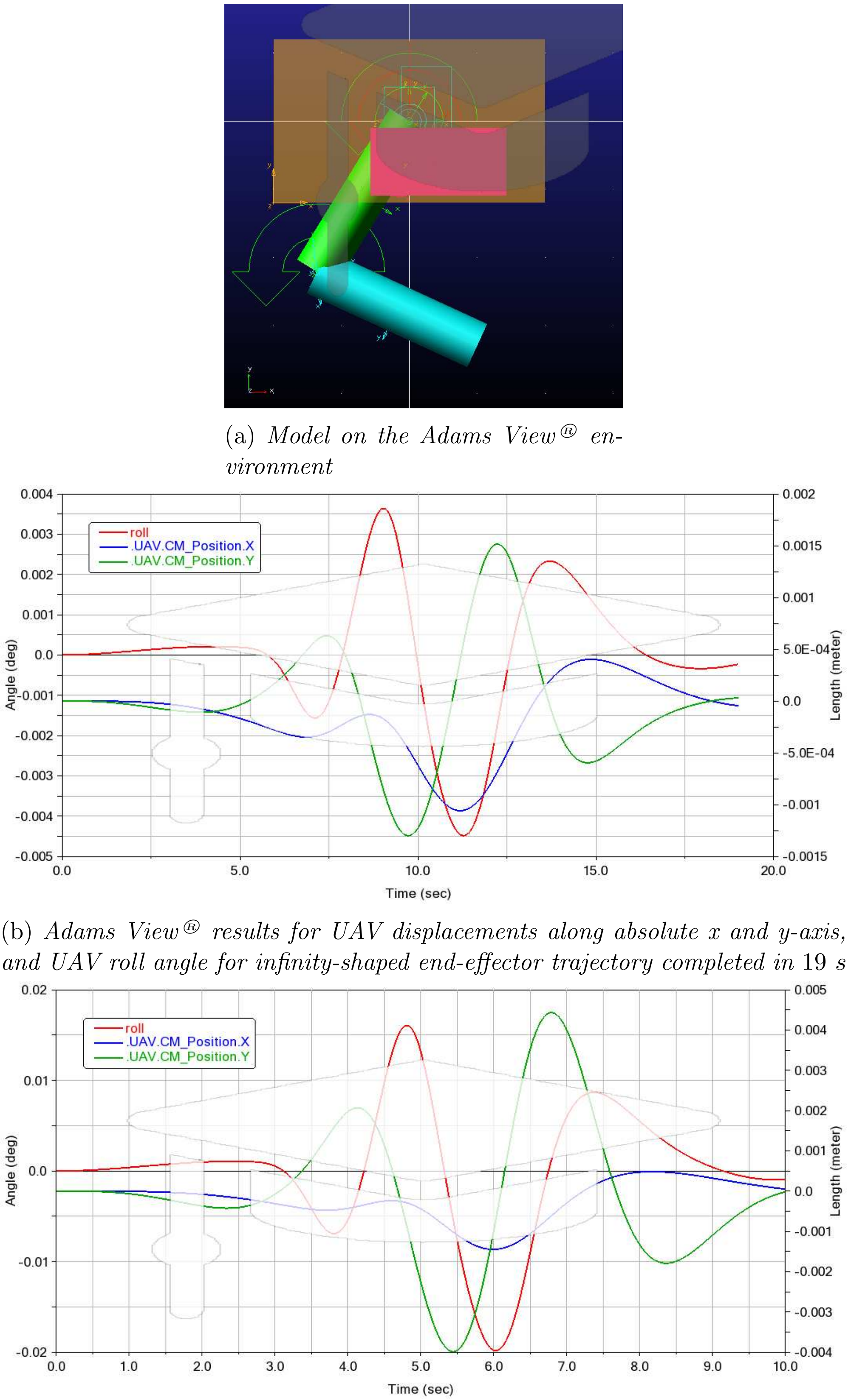
(c) *UAV rotation*

(d) *UAV controls*

(e) *manipulator's joint positions*

(f) *manipulator's joint velocities*

Figure 2.7: results of UAM with static balancing sliding battery for an infinity-shaped end-effector trajectory in 10 s



(a) *Model on the Adams View® environment*

(b) *Adams View® results for UAV displacements along absolute x and y-axis, and UAV roll angle for infinity-shaped end-effector trajectory completed in* 19 *s*

(c) *Adams View® results for UAV displacements along absolute x and y-axis, and UAV roll angle for infinity-shaped end-effector trajectory completed in* 10 *s*

Figure 2.8: Simulations on Adams View ® of static balancing sliding battery UAM model

2.3 Pendulum UAM model

Another simple static balancing method is to use a pendulum as a counterweight. It will be used a link of the same length of the links of the manipulator with its CoG positioned at the end of it, to simplify the concept of a real pendulum in which most of the mass is concentrated to one end. The mass of this pendulum is the same of the mass used for the sliding battery in section 2.2, that is $m_c = 2\,kg$.

The procedure is similar to that explained in section 2.2, the only addition is that the generalised coordinate of the sliding battery q_c calculated in Equation 2.2.2 is not the generalised coordinate of the pendulum but its projection on the x-axis, thus the generalised coordinate of the pendulum q_c will be:

$$q_c = \arcsin\left(\frac{a_c}{x_s}\right)$$

where a_c is the length of the pendulum and x_s is the position where the CoG of the pendulum along the absolute x-axis (orientation reference $\sum_A$) relative to the CoG of the UAV (origin reference $\sum_0$) as to be, namely the q_c calculated in Equation 2.2.2.

To consider the orientation of the UAV, the roll angle φ, is simply added this term to q_c already determined. Therefore the correct q_c will be:

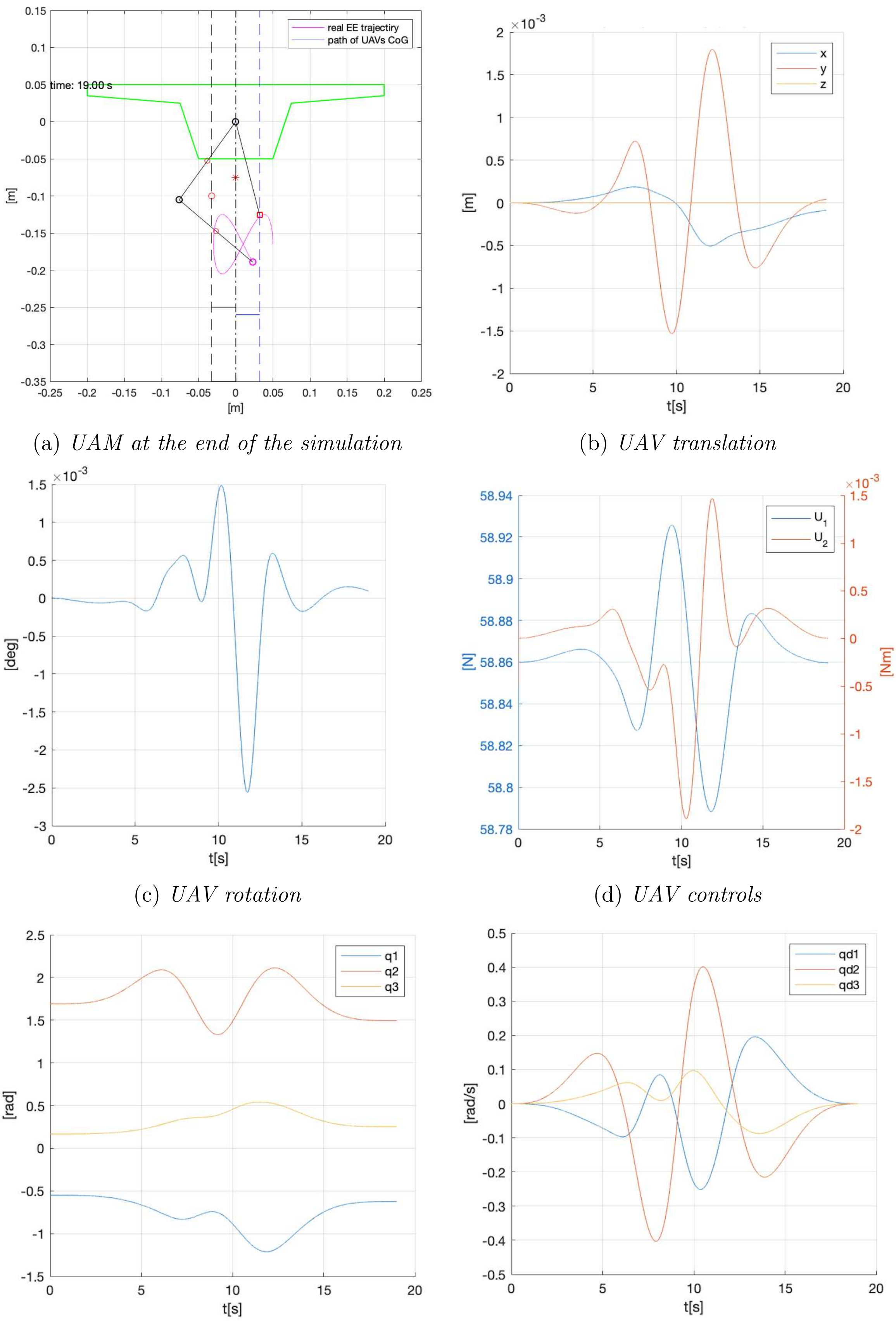
$$q_c = \arcsin\left(\frac{a_c}{x_s}\right) + \varphi \tag{2.3.1}$$

Then, is calculated the generalised velocity of the pendulum to reach the position of the generalised coordinate q_c, calculated in Equation 2.3.1, at the next calculation step as it has been done in Equation 2.2.4. Also, the extended Jacobian and the related output vector of the inverse kinematic will be the same used in Equation 2.2.16 and Equation 2.2.17, respectively.

In Figure 2.9 and Figure 2.10 are showed meaningful results of of the simulation of $19\,s$ and $10\,s$, respectively, to compare with results of previous method. Indeed, for the simulation with the slower trajectory (and for the faster trajectory in brackets) in Figure 2.9c (Figure 2.6b) the roll angle of the UAV results at least 1.8 (1.9) times lower than that in the simulation with the sliding battery in Figure 2.6b (Figure 2.7c). The projection of the counterweight's CoG along the absolute x-axis relative to the

UAV's CoG x_s is the same of that calculated in section 2.2 and so is the torque on the UAV's generated by the gravity effect. Therefore, the lower UAV roll angle values might be due to the lower angular momentum of the system with UAV's CoG as pole (a consequence of the different configuration of the UAM), because the angular momentum is linked with the total torque on the UAV's CoG There has also been an improvement concerning the UAV's CoG displacement along the absolute x-axis (reference $\sum_A$) reducing it about 2 times observable in Figure 2.9b and in Figure 2.10b for the simulations with an end-effector trajectory of $19\,s$ and $10\,s$, respectively.

The simulations in the Adams View® environment are showed in Figure 2.11 and confirm the results of simulations performed in MatLab®.



(a) *UAM at the end of the simulation*

(b) *UAV translation*

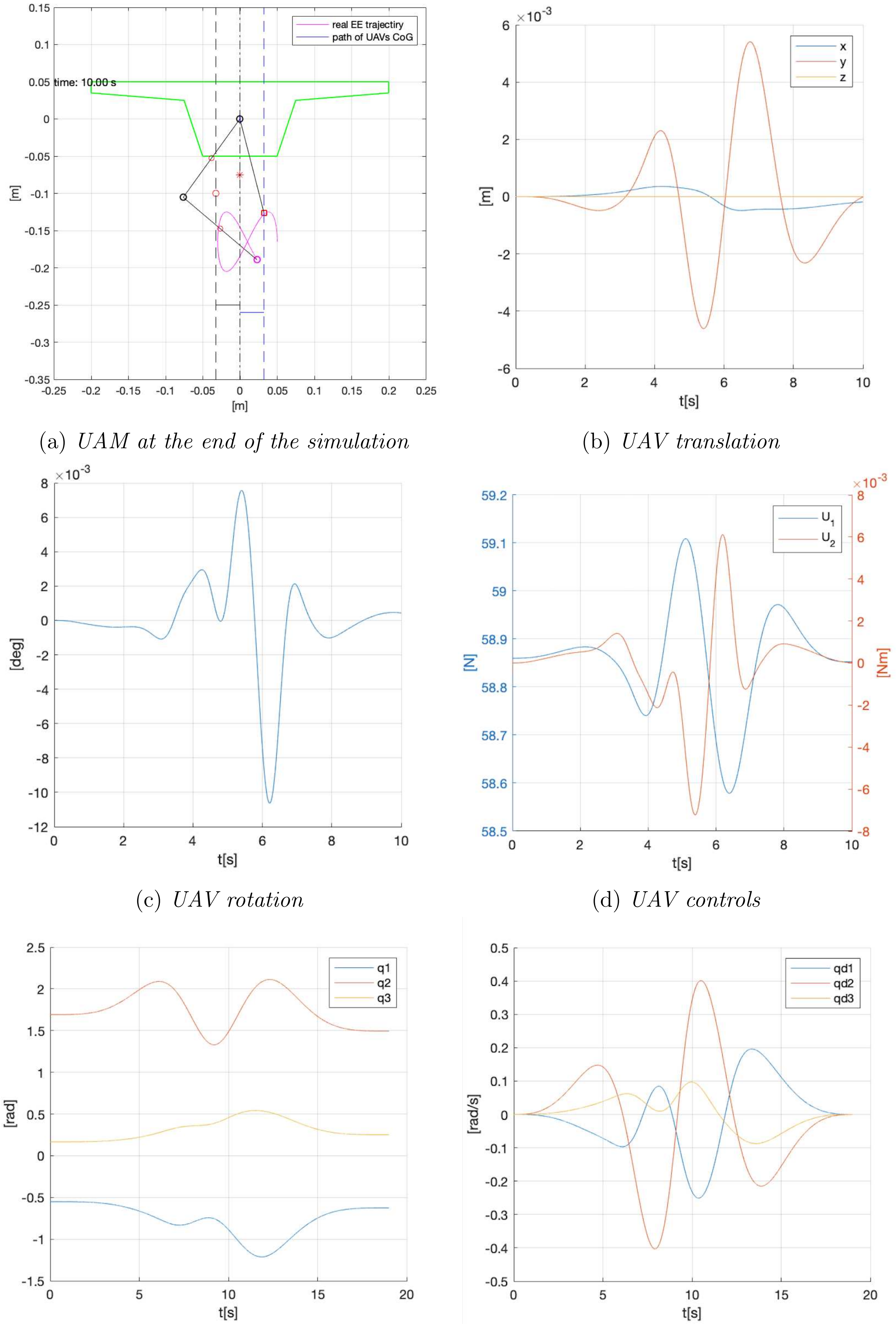
(c) *UAV rotation*

(d) *UAV controls*

(e) *manipulator's joint positions*

(f) *manipulator's joint velocities*

Figure 2.9: results of UAM with static balancing pendulum for an infinity-shaped end-effector trajectory in 19 s



(a) *UAM at the end of the simulation*

(b) *UAV translation*

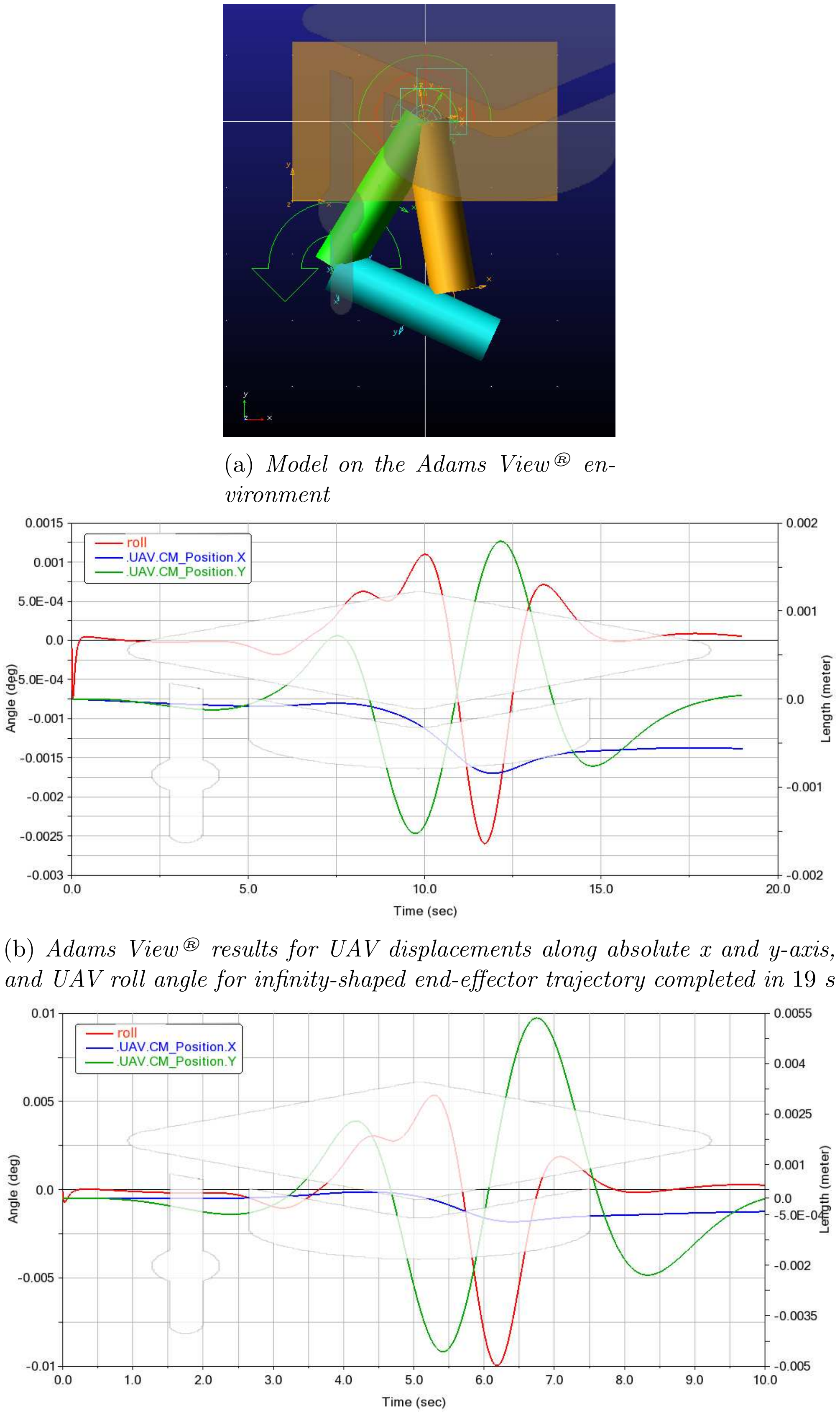
(c) *UAV rotation*

(d) *UAV controls*

(e) *manipulator's joint positions*

(f) *manipulator's joint velocities*

Figure 2.10: results of UAM with static balancing pendulum for an infinity-shaped end-effector trajectory in 10 s



(a) *Model on the Adams View® environment*

(b) *Adams View® results for UAV displacements along absolute x and y-axis, and UAV roll angle for infinity-shaped end-effector trajectory completed in* 19 *s*

(c) *Adams View® results for UAV displacements along absolute x and y-axis, and UAV roll angle for infinity-shaped end-effector trajectory completed in* 10 *s*

Figure 2.11: Simulations on Adams View ® of static balancing pendulum UAM

2.4 Dual Arm UAM model

The last static balancing mechanism analysed will be another 2 DoF robotic arm mounted on the same place as joint 1 of the manipulator. It is an extension of the pendulum model in section 2.3 in which the mass to balance the CoG of the manipulator will be the sum of the masses of the counterweight robotic arm concentrated in the CoG of this robotic arm. The advantage over the pendulum model will be the presence of an additional task to be exploited during manipulator balancing. It could be used to attempt to eliminate the linear momentum of the manipulator's CoG along absolute y-axis (reference $\sum_A$) as will be seen later on.

Obviously, can be imposed to the counterweight robotic arm the opposite joint velocities trajectories (in joint space) of those of the manipulator, in a starting configuration of equilibrium. This will cancel perfectly (theoretically) the influences of the manipulator on the UAV concerning torques and force along absolute x-axis, but will double the force along absolute y-axis.

The procedure to perform a static balancing with the counterweight robotic arm is, first of all, to localize the position of the manipulator's CoG with reference to the UAV's coordinate system $\sum_0$, as has been done in Equation 2.2.1. Secondly, to balance it with the counterweight robotic arm's CoG. Thus, the motion of the counterweight robotic will be imposed tracking a trajectory with its CoG. This trajectory is the opposite of that done by manipulator's CoG. The tracking of this trajectory will be executed with an inverse kinematic of the counterweight robotic arm. This time the robotic arm will be managed as if it had a fixed base because because the trajectory will be always relative to the UAV's CoG position. Only the roll angle of the UAV will be taken into account because the orientation of the axes on which the trajectory is defined is that of the absolute reference frame $\sum_A$ adding the roll angle φ to the first generalised coordinate of the counterweight robotic arm. Therefore, the geometric Jacobian of the counterweight arm is needed and it is:

$$J_{c,CoG} = \begin{bmatrix} \dfrac{a_3\,\mathrm{c}(q_3+\varphi)(m_3+2\,m_4)+m_4\,a_4\,\mathrm{c}(q_3+q_4)}{m_3+m_4} & \dfrac{m_4\,a_4\,\mathrm{c}(q_3+q_4)}{m_3+m_4} \\ \dfrac{a_3\,\mathrm{s}(q_3+\varphi)(m_3+2\,m_4)+m_4\,a_4\,s(q_3+q_4)}{m_3+m_4} & \dfrac{m_4\,a_4\,\mathrm{s}(q_3+q_4)}{m_3+m_4} \end{bmatrix} \tag{2.4.1}$$

where a_3 and a_4 are the semi-length of link 3 and 4 (first and second link of the counterweight robotic arm) which localize their CoG form their joint, respectively; m_3 and m_4 are their respective masses; q_3 and q_4 their respective generalised coordinates; c and s are abbreviations to mean cos and sin. There is another part of the kinematics equations which considers the derivative of the roll angle φ and it will appear in the vector of velocities $\vec{v}_{CoG}$ Equation 2.4.7.

Now, to perform the kinematic inversion, it is necessary to define the trajectory to be followed by the counterweight robotic arm's CoG. This is the opposite of the manipulator's CoG velocity. So it is necessary to calculate the manipulator's CoG velocity ${}^{0}\dot{\vec{r}}_G$:

$$ {}^{0}\dot{\vec{r}}_{G,k} = \frac{{}^{0}\vec{r}_{G,k} - {}^{0}\vec{r}_{G,k-1}}{\Delta t} \tag{2.4.2} $$

which has the same structure of Equation 2.2.4.

To achieve static balancing, CoG compensation must first be imposed along the absolute x-axis (reference $\sum_0$) and then could be chosen an additional task for the second DoF of the counterweight robotic arm. It could consist of keeping the CoG of the manipulator + counterweight robotic arm system in the same position with reference to the UAV's CoG position (axes orientation as that of the reference $\sum_A$). It is important to emphasise that for x-axis compensation to work, the initial configuration of manipulator and counterweight robotic arm must be in equilibrium because trajectories are imposed at velocities. To obtain what has just been described, the vector of velocities for the inverse kinematic of the counterweight robotic arm will be:

$$ \vec{v}_{CoG} = \begin{Bmatrix} -\ {}^{0}\dot{r}_{G,x} \\ 0 \end{Bmatrix} - \frac{m_3\, a_3\, \omega_{0,z}}{m_3 + m_4} \begin{Bmatrix} \cos(q_3 + \varphi) \\ \sin(q_4 + \varphi) \end{Bmatrix} \tag{2.4.3} $$

where ${}^{0}\dot{r}_{G,x}$ is the absolute x-axis (reference $\sum_A$) component of vector ${}^{0}\dot{\vec{r}}_G$ and $\omega_{0,z}$ is the time derivative of roll angle φ. The second element of the equation comes from the process to obtain the geometric Jacobian $J_{c,CoG}$ showed in Equation 2.4.1.

Now is possible to determine generalised velocities of the counterweight robotic arm q_3 and q_4 with kinematic inversion:

$$ \begin{Bmatrix} q_3 \\ q_4 \end{Bmatrix} = J_{c,CoG}^{-1}\ \vec{v}_{CoG} \tag{2.4.4} $$

Next, the generalised coordinates of the manipulator will be calculated using the generalised Jacobian for the entire system (manipulator + counterweight robotic arm) obtaining a 2 × 4 matrix J^* which require to be extended to be square as did in Equation 2.2.16, but with two generalised velocities now:

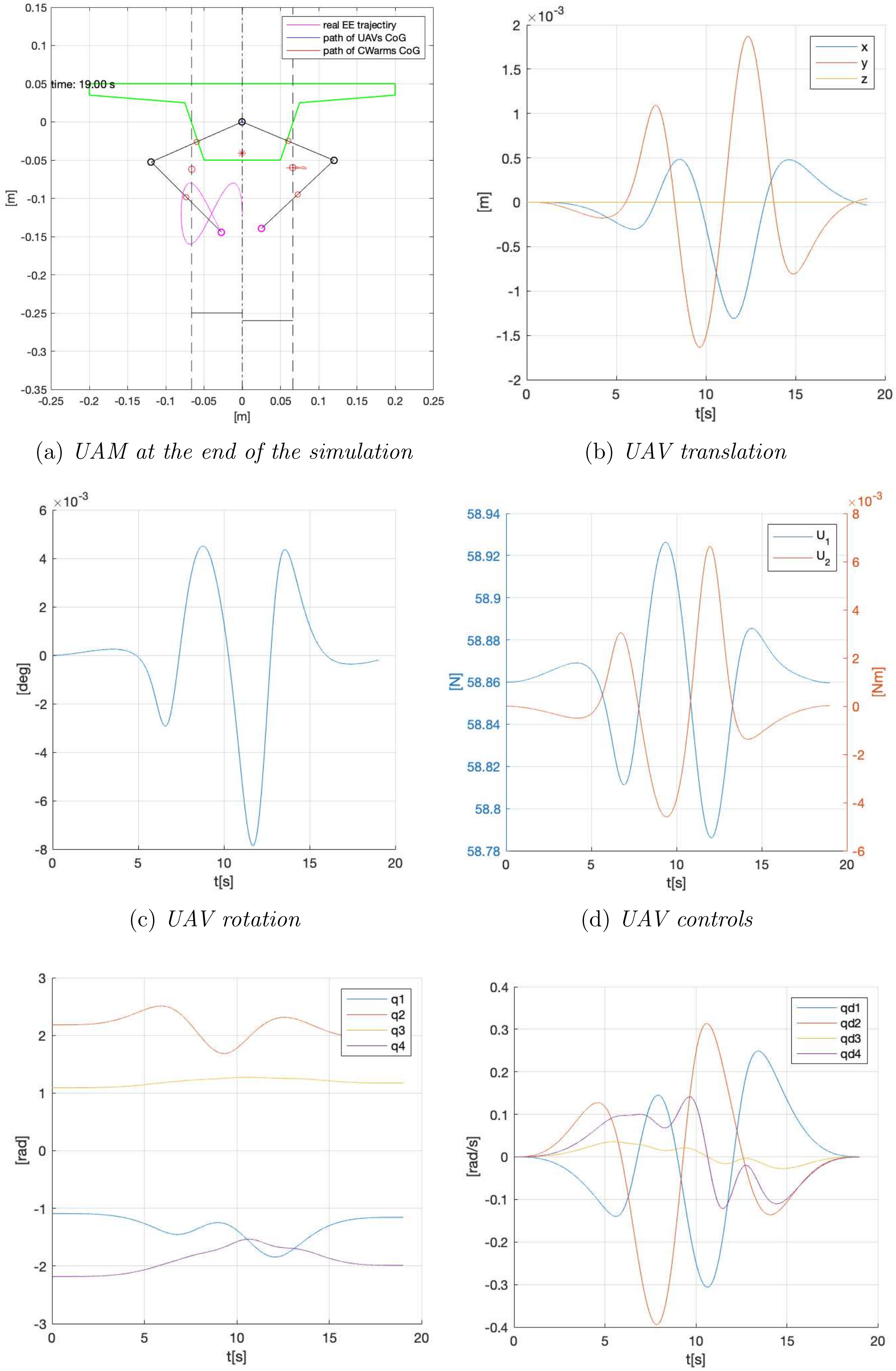
$$J_{ext} = \begin{bmatrix} & \begin{bmatrix} J^* \end{bmatrix} & & \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.4.5}$$

where J^* is the generalised Jacobian of the UAM with the dual arm static balancing method. And the related output vector of the inverse kinematic will become:

$$V_{ext} = \begin{Bmatrix} v_{e,x} - p_{0,x} \\ v_{e,y} - p_{0,y} \\ \dot{q}_3 \\ \dot{q}_4 \end{Bmatrix} \tag{2.4.6}$$

In Figure 2.12 and Figure 2.13 are showed meaningful results of of the simulation of 19 s and 10 s, respectively, to compare with results of previous methods. For both simulations with fast and slow trajectories, results are less effective than those of the previous methods as may be seen, in particular compared to the static pendulum balancing method.

The simulations in the Adams View® environment are showed in Figure 2.14 and confirm the results of simulations performed in MatLab®.



(a) *UAM at the end of the simulation*

(b) *UAV translation*

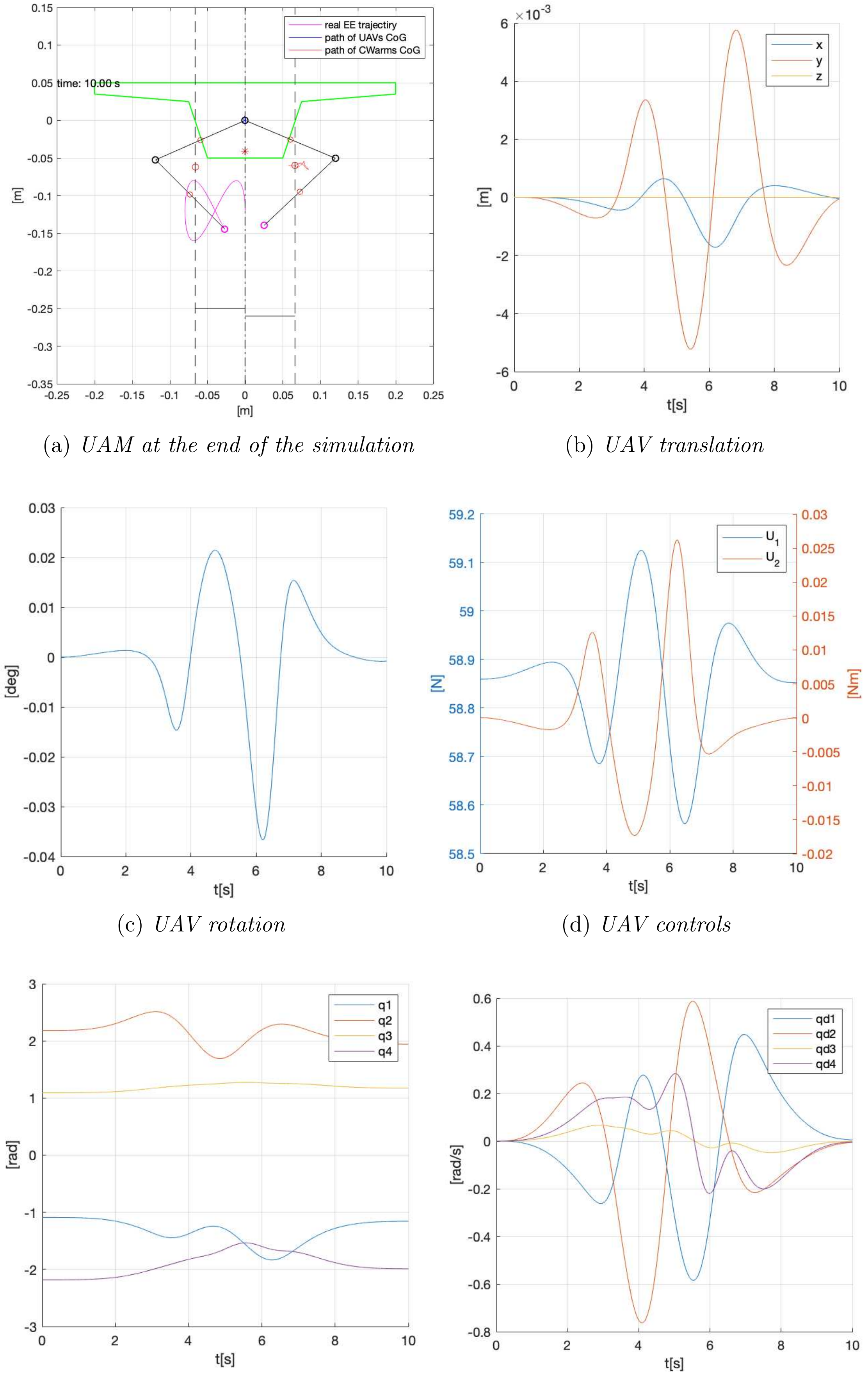
(c) *UAV rotation*

(d) *UAV controls*

(e) *manipulator's joint positions*

(f) *manipulator's joint velocities*

Figure 2.12: results of UAM with static balancing dual arm with $v_{y,CoG} = 0$ for an infinity-shaped end-effector trajectory in $19\,s$



(a) *UAM at the end of the simulation*

(b) *UAV translation*

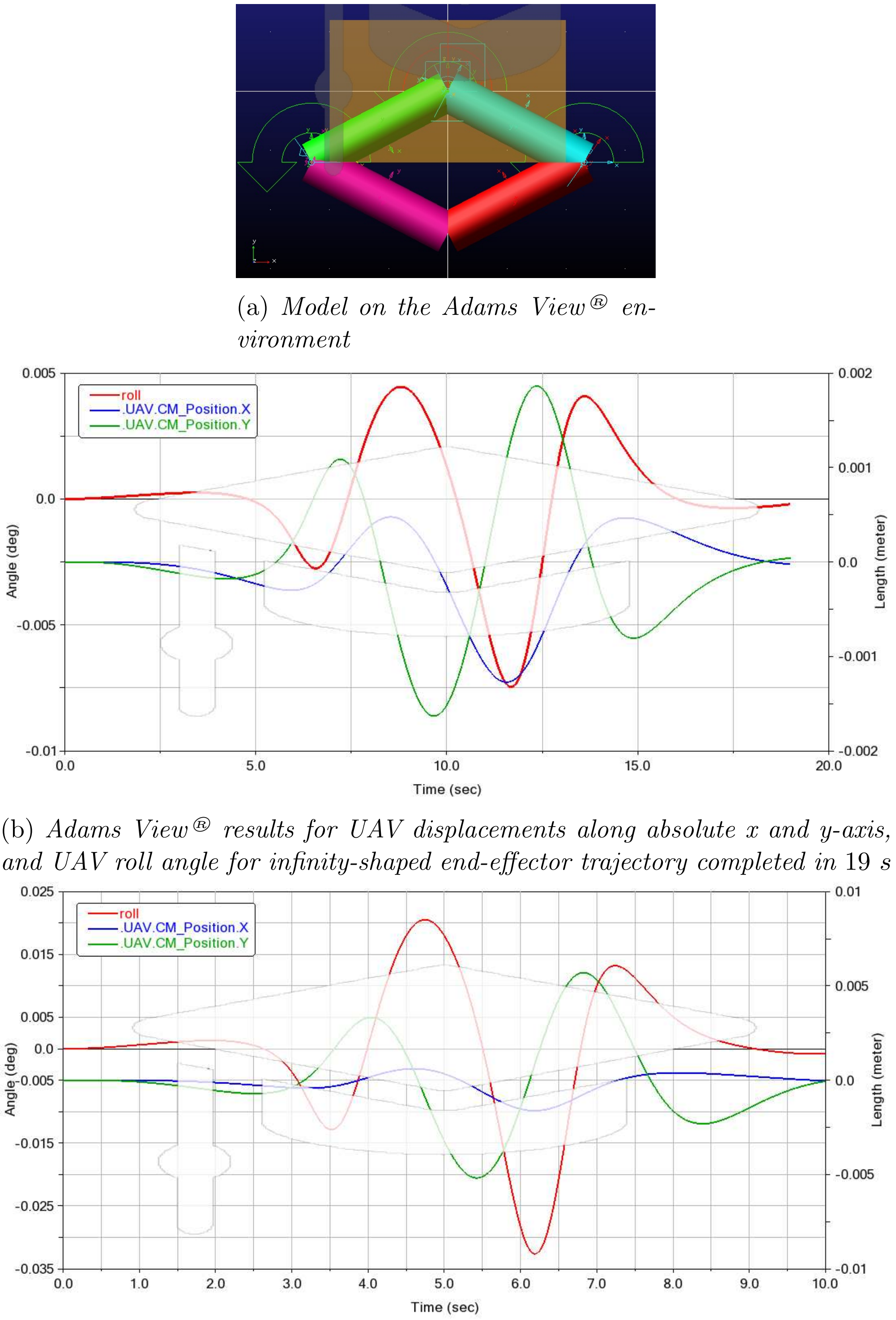
(c) *UAV rotation*

(d) *UAV controls*

(e) *manipulator's joint positions*

(f) *manipulator's joint velocities*

Figure 2.13: results of UAM with static balancing dual arm with $v_{y,CoG} = 0$ for an infinity-shaped end-effector trajectory in $10\,s$



(a) *Model on the Adams View® environment*

(b) *Adams View® results for UAV displacements along absolute x and y-axis, and UAV roll angle for infinity-shaped end-effector trajectory completed in* 19 *s*

(c) *Adams View® results for UAV displacements along absolute x and y-axis, and UAV roll angle for infinity-shaped end-effector trajectory completed in* 10 *s*

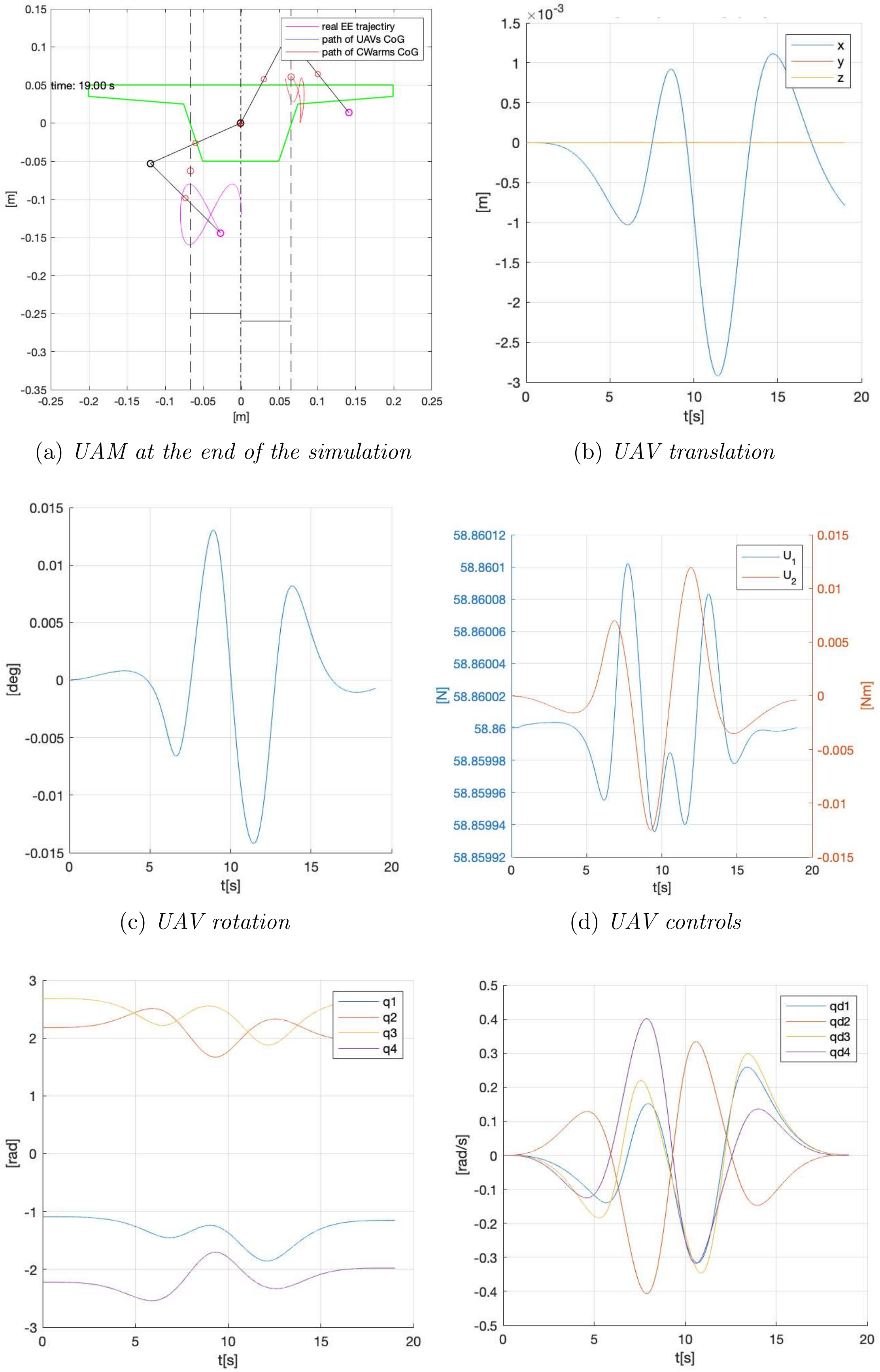
Figure 2.14: Simulations on Adams View ® of static balancing dual arm UAM with $v_{y,CoG} = 0$

Now a different additional task will be tried, the one to cancel the linear momentum of manipulator's CoG along absolute y-axis (reference $\sum_A$). It will be performed imposing to the counterweight robotic arm's CoG to follow the opposite velocity trajectory of the manipulator's CoG along absolute y-axis (reference $\sum_A$), so $\vec{v}_{CoG}$ will become:

$$\begin{aligned}
\vec{v}_{CoG} &= \begin{Bmatrix} -\ {}^0\dot{r}_{G,x} \\ -\ {}^0\dot{r}_{G,y} \end{Bmatrix} - \frac{m_3\, a_3\, \omega_{0,z}}{m_3 + m_4} \begin{Bmatrix} \cos(q_3 + \varphi) \\ \sin(q_4 + \varphi) \end{Bmatrix} \\
\vec{v}_{CoG} &= -\ {}^0\dot{\vec{r}}_G - \frac{m_3\, a_3\, \omega_{0,z}}{m_3 + m_4} \begin{Bmatrix} \cos(q_3 + \varphi) \\ \sin(q_4 + \varphi) \end{Bmatrix}
\end{aligned} \tag{2.4.7}$$

In Figure 2.15 and Figure 2.16 are showed meaningful results of the simulation of 19 s and 10 s, respectively. In Figure 2.15b for the slow trajectory and in Figure 2.16b for the fast trajectory may be noticed that the translation of the UAV's CoG along absolute y-axis (reference $\sum_A$) has been decreased even almost cancelled. Indeed in Figure 2.15d and Figure 2.15d can be seen that the vertical (respect the UAV's reference frame $\sum_0$) control $U1$ intervenes very little. In contrast the roll angle reaches higher values as can be seen in Figure 2.15d and Figure 2.16d probably due to dynamic effects because the angular momentum of the counterweight robotic arm is not negligible in particular when both of the first joints of manipulator and counterweight rotate in the same direction adding up to that of the manipulator (as happens here in these last simulations). The different starting configuration here (still in balance) seen in Figure 2.15a and Figure 2.15a has been chosen so that the end-effector trajectory could be executed without the counterweight robotic arm reaching singular configurations.

Also for this variant, the simulations in the Adams View® environment are showed in Figure 2.17 and confirm the results of simulations performed in MatLab®.



(a) *UAM at the end of the simulation*

(b) *UAV translation*

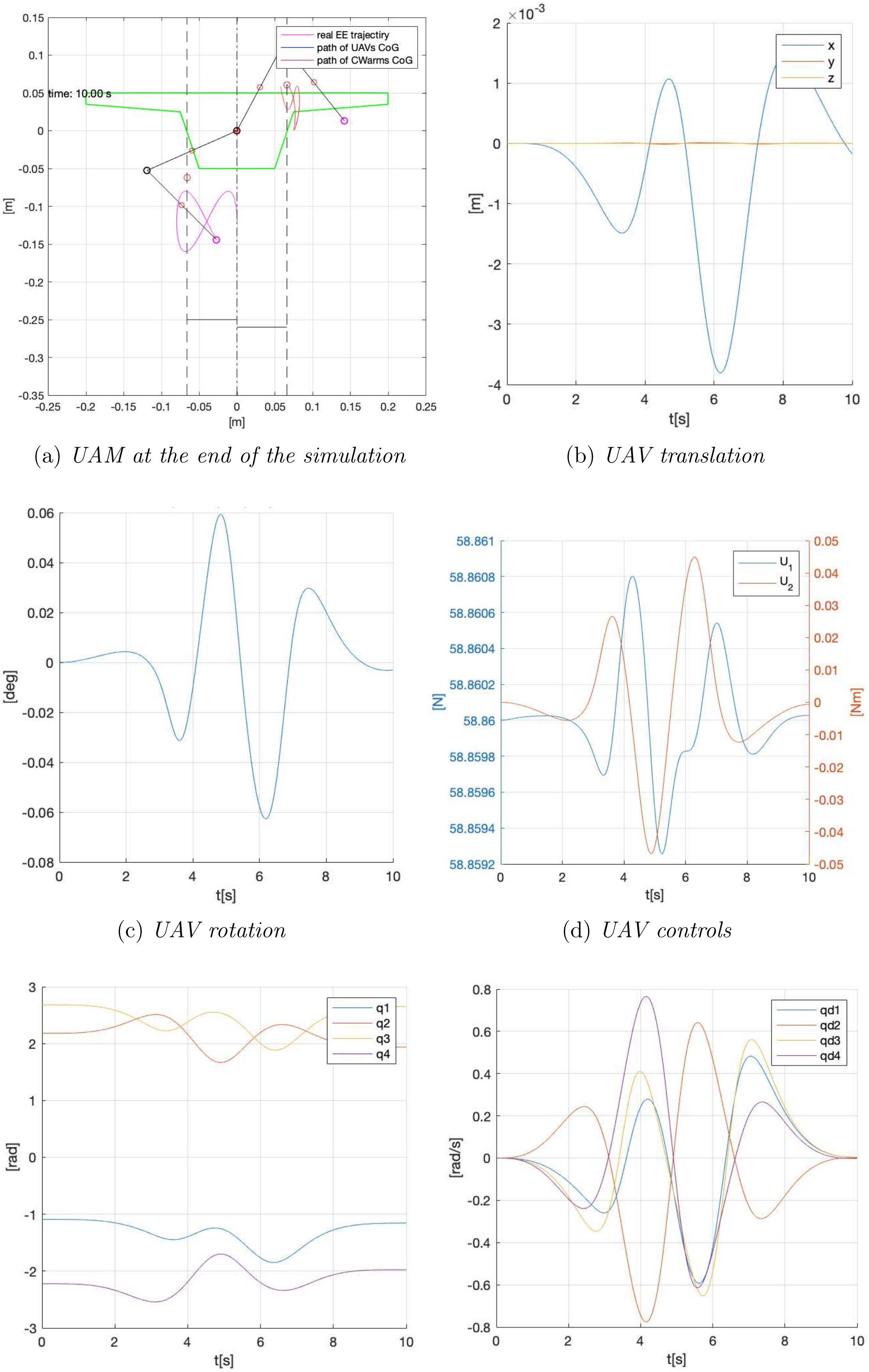
(c) *UAV rotation*

(d) *UAV controls*

(e) *manipulator's joint positions*

(f) *manipulator's joint velocities*

Figure 2.15: results of UAM with static balancing dual arm with manipulator's CoG linear momentum in y-axis compensation for an infinity-shaped end-effector trajectory in 19 s



(a) *UAM at the end of the simulation*

(b) *UAV translation*

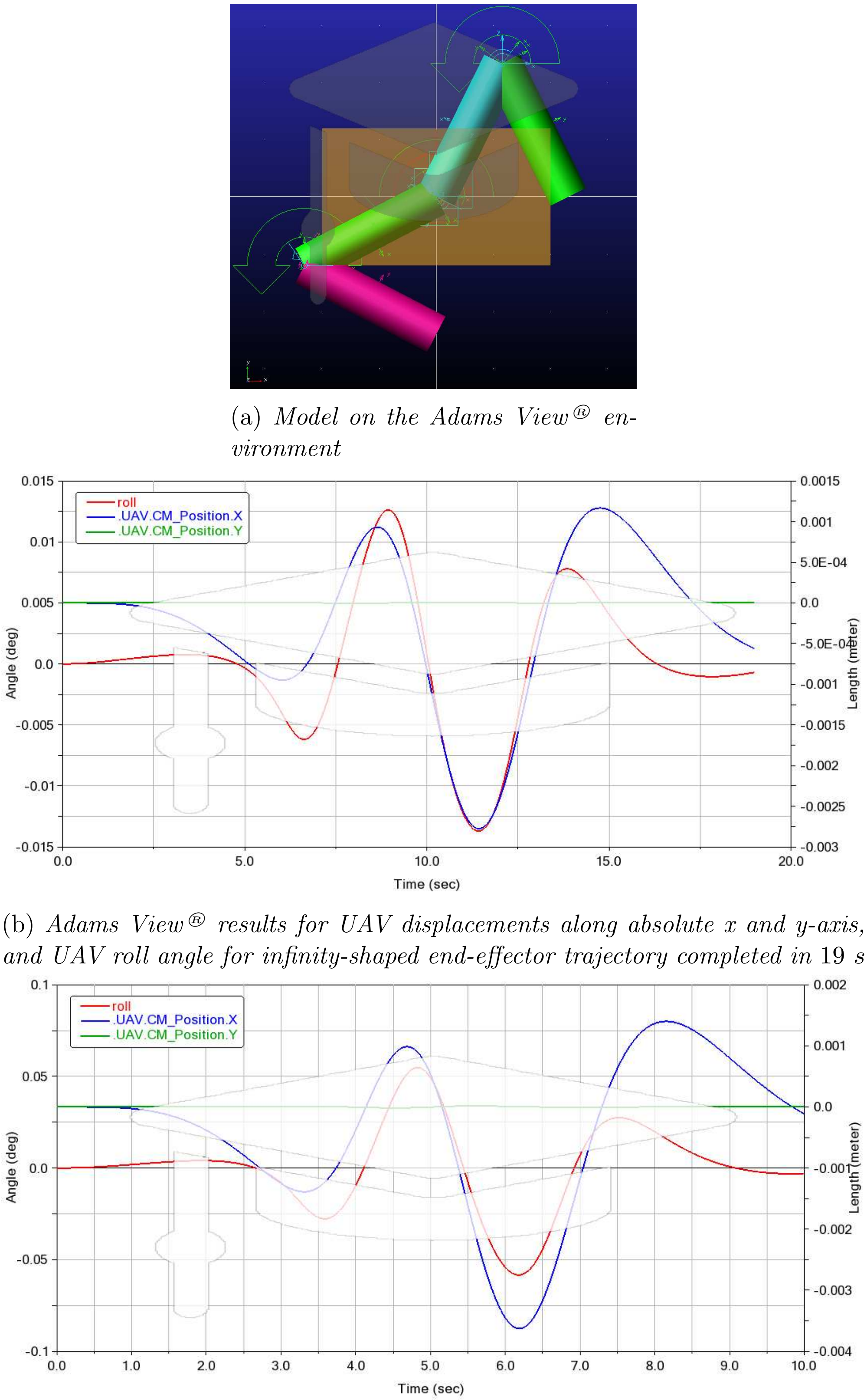
(c) *UAV rotation*

(d) *UAV controls*

(e) *manipulator's joint positions*

(f) *manipulator's joint velocities*

Figure 2.16: results of UAM with static balancing dual arm with manipulator's CoG linear momentum in y-axis compensation for an infinity-shaped end-effector trajectory in $10\,s$



(a) *Model on the Adams View® environment*

(b) *Adams View® results for UAV displacements along absolute x and y-axis, and UAV roll angle for infinity-shaped end-effector trajectory completed in* 19 *s*

(c) *Adams View® results for UAV displacements along absolute x and y-axis, and UAV roll angle for infinity-shaped end-effector trajectory completed in* 10 *s*

Figure 2.17: Simulations on Adams View ® of static balancing dual arm UAM with manipulator's CoG linear momentum in y-axis compensation

3. Dynamic Balancing Systems

These types of balancing systems do not consider only the torque on the UAV given by the manipulator generated by the gravity force, but take into account the total torque transferred to the UAV from the manipulator, to counteract it.

3.1 Reaction Wheel

Reaction wheel is a method utilized to correct the attitude of satellites in aerospace engineering. It consist of a rotational mass (a wheel) with high moment of inertia on its axis of rotation, so here will be presented as a wheel consisting of a rim, where most of the mass is located, connected to its hub by spokes. Its aim is to generate an angular momentum that cancels out the angular momentum of the UAV, or can also be understood as generating the required torque simply by varying its angular momentum, thus generating torque in the direction of rotation opposite to its variation. The method used here will be to consider it as a torque generator.

The torque generated by the manipulator (mounted on the CoG of the UAV) on the joint 0 is determined with the *Inverse Dynamic computation by the Recursive Newton-Euler method.* This torque could be reduced or cancelled by a countertorque on the UAV CoG generated by the angular acceleration of the reaction wheel:

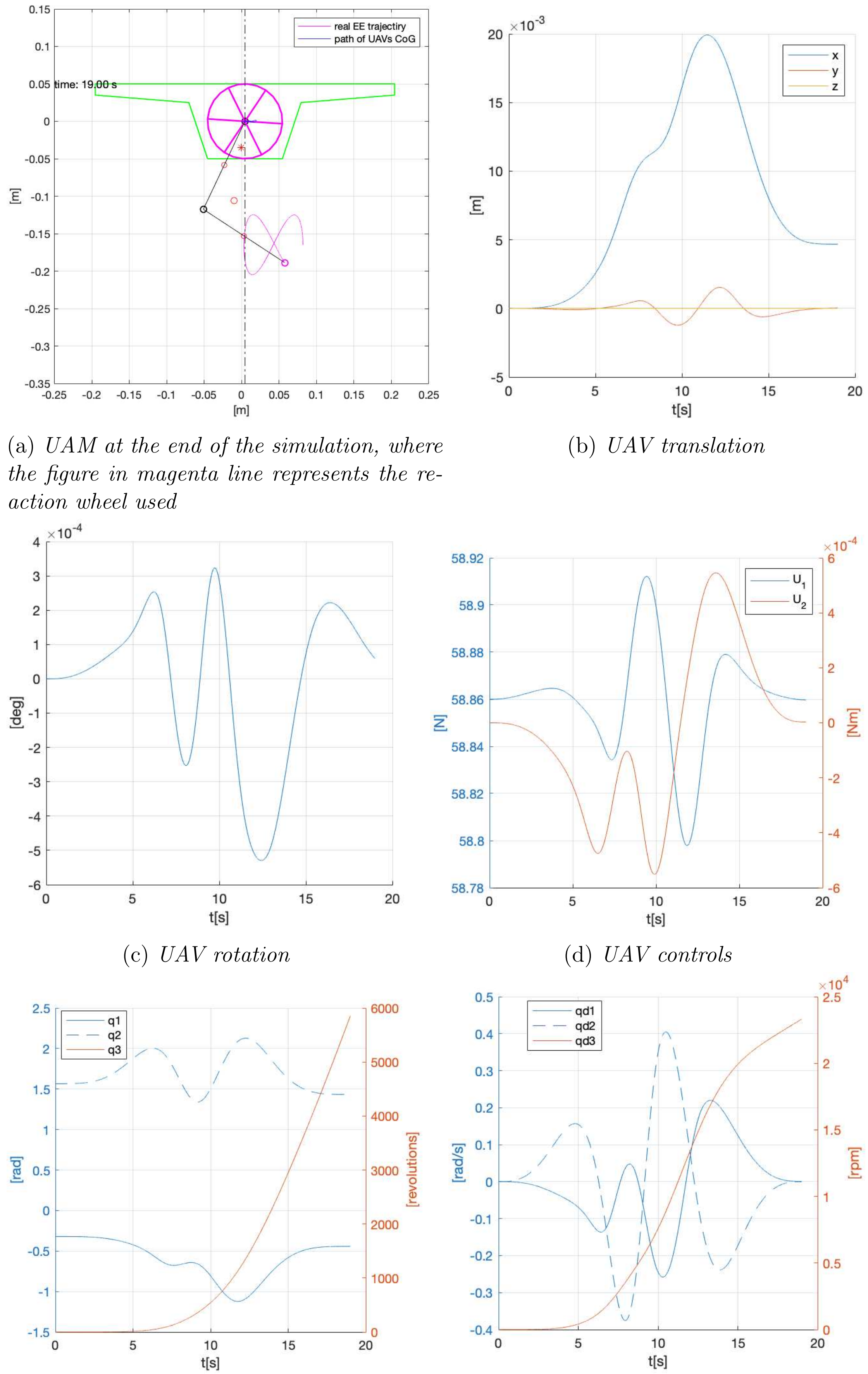
$$\begin{aligned} \tau_{1,k-1} &= \ddot{q}_{RW}\, I_{RW} \\ \ddot{q}_{RW,k} &= \frac{\tau_{1,k-1}}{I_{RW}} \end{aligned} \tag{3.1.1}$$

where k is the index of calculation step; $\tau_{1,k-1}$ is the torque generated by the manipulator on the UAV; $\ddot{q}_{RW}$ is the generalised acceleration of the reaction wheel and I_{RW} is the reaction wheel's moment of inertia along the absolute z-axis (reference $\sum_A$). The generalised velocity of the reaction wheel is retrieved as:

$$\dot{q}_{RW,k} = \ddot{q}_{RW,k}\, dt + \dot{q}_{RW,k-1} \tag{3.1.2}$$

In Figure 3.1 and Figure 3.2 are showed meaningful results of the simulation of 19 s and 10 s, respectively, with the addition of the torques behaviour on the UAV's CoG. In Figure 3.1c and Figure 3.2c can be seen as the UAV roll angle as achieved the lowest maximum and minimum untill now. Comparison of roll angle performance in the static balance pendulum method (Figure 2.9c and Figure 2.10c) with the reaction wheel method show at least 5 times lower values of the roll angle with the reaction wheel for the slow end-effector trajectory and at least 6.6 times lower values for the fast one. This because dynamic balancing methods give best results than the static ones in particular when dynamic effects of the system are substantial, so when the trajectory of the end-effector is faster with all other variables being equal. Unfortunately, required reaction wheel speeds are too high from a technological point of view considering also that the reaction wheel implementation mechanism must be supported by the UAV and can't be too heavy and cumbersome. This is a well known problem with reaction wheel in satellites: if an attitude control system that transfers any unwanted angular momentum from spacecraft to the reaction wheels is always running, spinning them up or down, they will approach their speed limits (saturating the system). So will be determined the direction of torque needed to slow them down and then, using some external source of torque, like attitude control thrusters, magnetotorquers, or other systems apply that torque to the spacecraft letting the attitude control system automatically transfer it to the wheels. One reason to look at it this way is that, usually, whatever external attitude control mechanism used won't be as precise or as quickly reactive as reaction wheels. This procedure is called desaturation of the reaction wheels. These desaturation routines were not tested here because the accelerations of the reaction wheel's joint would have been too high anyway.

The simulations in the Adams View® environment are showed in Figure 3.3 and confirm the results of simulations performed in MatLab®.



(a) *UAM at the end of the simulation, where the figure in magenta line represents the reaction wheel used*

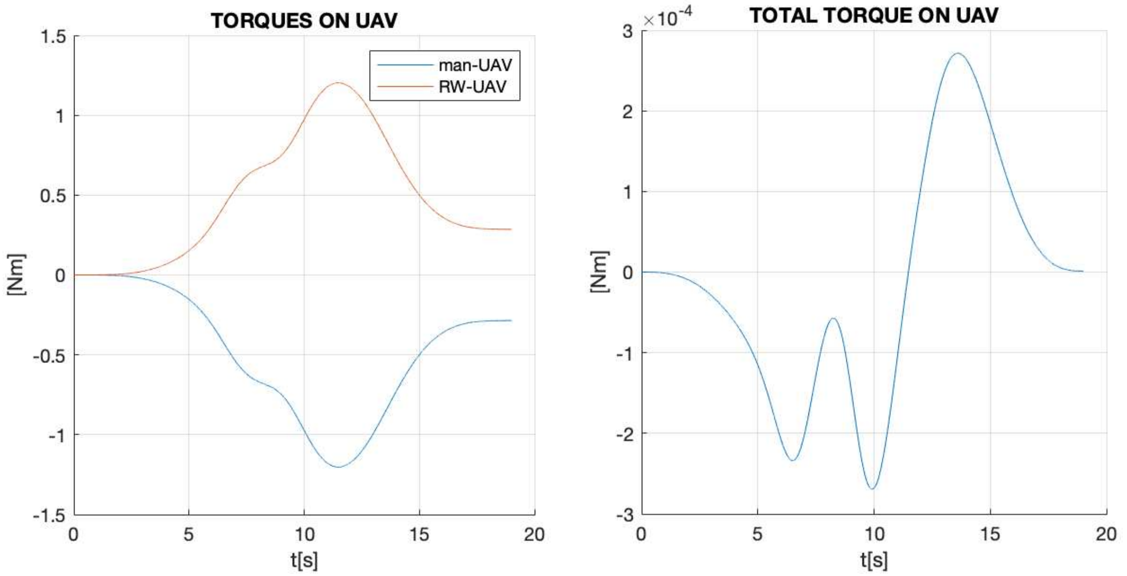
(b) *UAV translation*

(c) *UAV rotation*

(d) *UAV controls*

(e) *manipulator's joint positions*

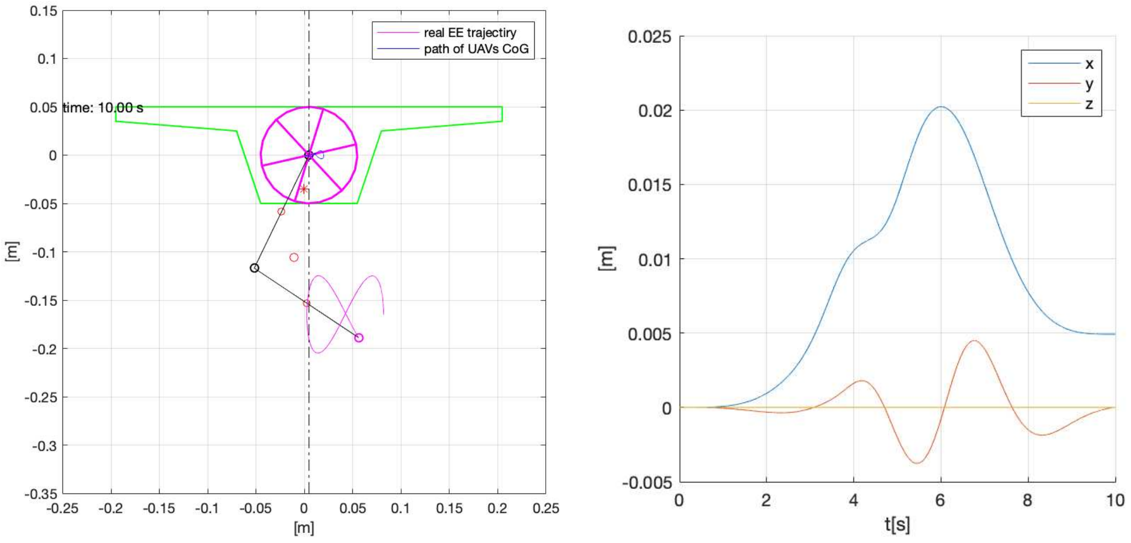
(f) *manipulator's joint velocities*



(g) *manipulator's and reaction wheel's torques acting on UAV*

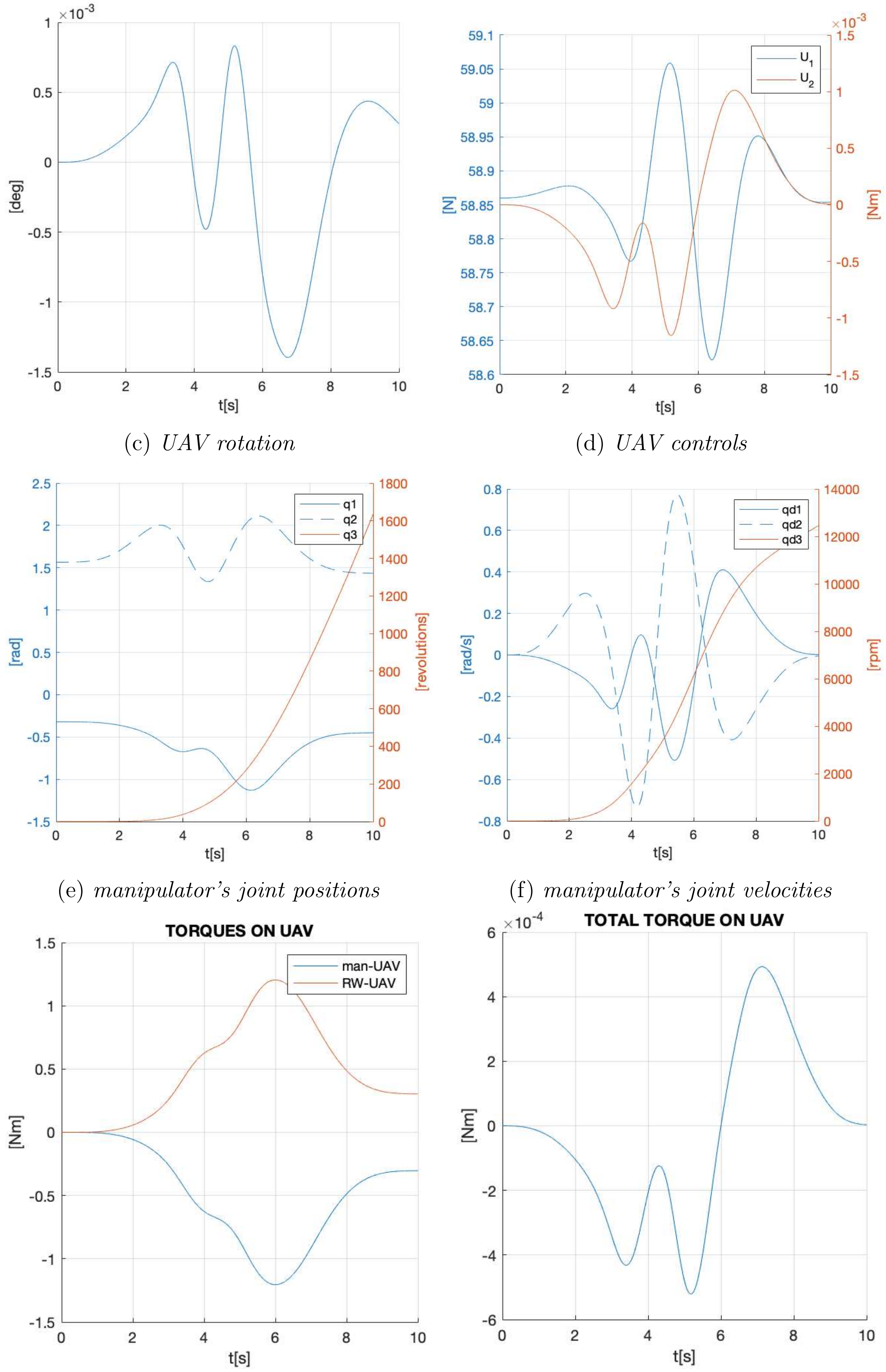
(h) *total torque acting on UAV (non considering* $U2$ *torque)*

Figure 3.1: results of UAM with dynamic balancing reaction wheel method for an infinity-shaped end-effector trajectory in $19\,s$



(a) *UAM at the end of the simulation*

(b) *UAV translation*



(c) *UAV rotation*

(d) *UAV controls*

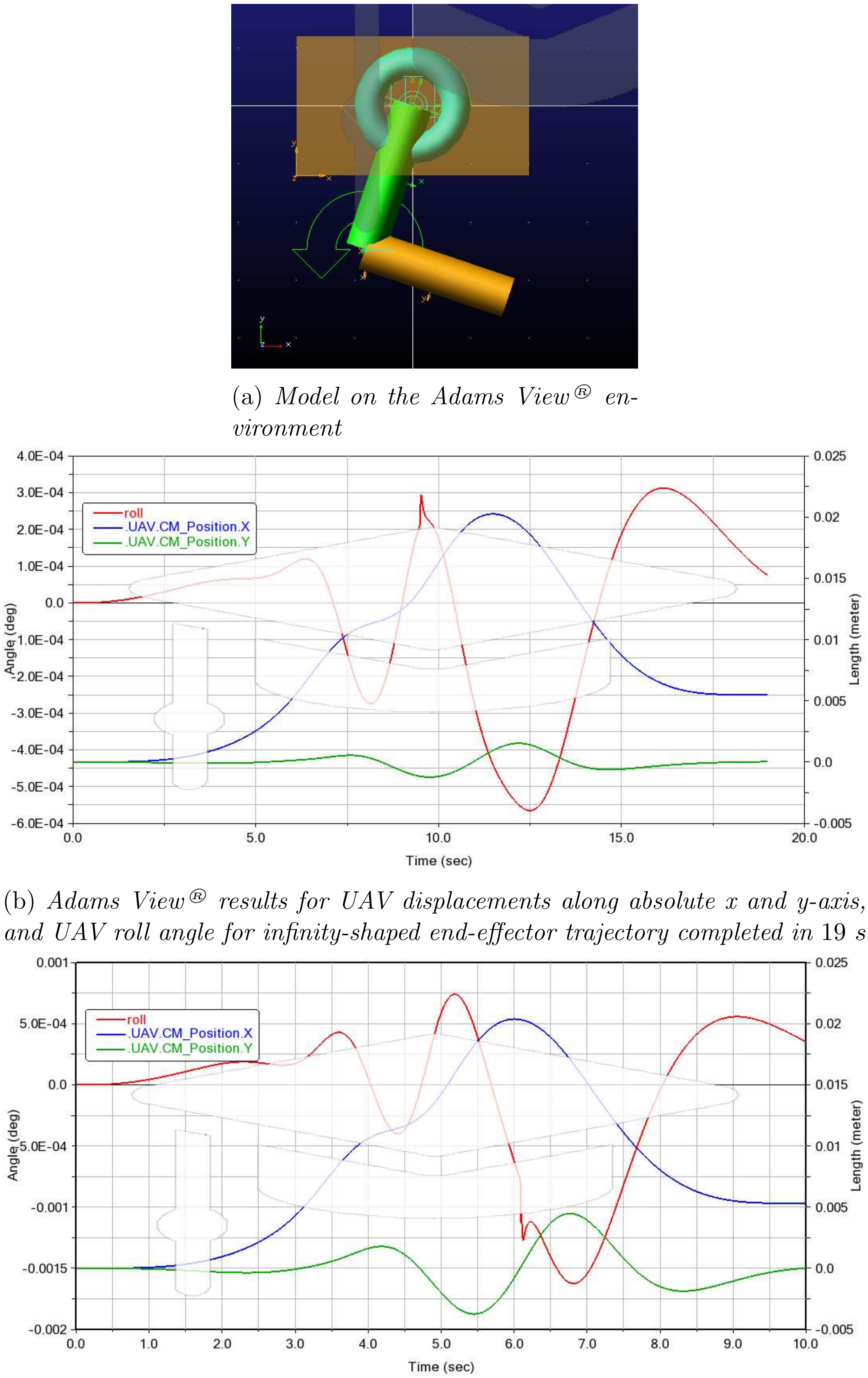
(e) *manipulator's joint positions*

(f) *manipulator's joint velocities*

(g) *manipulator's and reaction wheel's torques acting on UAV*

(h) *total torque acting on UAV (non considering $U2$ torque)*

Figure 3.2: results of UAM with dynamic balancing reaction wheel method for an infinity-shaped end-effector trajectory in 10 s



(a) *Model on the Adams View® environment*

(b) *Adams View® results for UAV displacements along absolute x and y-axis, and UAV roll angle for infinity-shaped end-effector trajectory completed in* 19 *s*

(c) *Adams View® results for UAV displacements along absolute x and y-axis, and UAV roll angle for infinity-shaped end-effector trajectory completed in* 10 *s*

Figure 3.3: Simulations on Adams View ® of UAM with the reaction wheel

3.2 Manipulator Mounted on the Sliding Battery

This method comes as a modification of the static balancing sliding battery in section 2.2. It consist of attaching the 2 Dof manipulator to the CoG of the sliding battery. In this way the workspace and dextrous space of the manipulating increase because of an additional DoF, so the manipulator become redundant. Therefore, the inverse kinematic for this new configuration requires a different approach or an extended Jacobian matrix. Here will be utilized an extended Jacobian, so a new task other than following the imposed end-effector trajectory has to be defined. The chosen task will be that used also in [12] to minimize the torque distuurbance from the manipulator to the base. The general form of the additional constrain for a 3 DoF manipulator is $f_1\,\dot{q}_1 + f_2\,\dot{q}_2 + f_3\,\dot{q}_3 = b$. Adding this constrain to the generalised Jacobian for a 3 DoF system allow to find a unique solution:

$$\begin{Bmatrix} v_x \\ v_y \\ b \end{Bmatrix} = \begin{bmatrix} & \begin{bmatrix} J^* \end{bmatrix} & \\ f_1 & f_2 & f_3 \end{bmatrix} \begin{Bmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_2 \end{Bmatrix} \tag{3.2.1}$$

The force/torques acting on the manipulator change its angular momentum about the base CoG, so its derivative results as:

$$\frac{\mathrm{d}\vec{K}_{man}}{\mathrm{d}t} = -\vec{v}_B \times \vec{L}_{man} + \vec{\tau}_B + \left(\vec{\tau}_m + \vec{r}_m \times \begin{Bmatrix} R_x \\ R_y \\ R_z \end{Bmatrix} \right) \tag{3.2.2}$$

where $\vec{v}_B$ is the velocity vector of the drone's CoG (pole of the angular momentum), $\vec{L}_{man}$ is the vector of the linear momentum of the manipulator, r_m is the vector linking UAV's CoG with joint 1 of the manipulator, $\vec{\tau}_B$ is the torque vector about the drone's CoG generated by the gravity force $\vec{F}_{man} = m_{man}\,\vec{g}$ (m_{man} as the total mass of the manipulator) and by external forces on end-effector. The term in the brackets is the change in angular momentum of the manipulator caused by the reaction torques exerted by the drone on the manipulator (is equal and opposite to the torque disturbance transmitted form the manipulator to the UAV, it will be imposed null. So, the

Equation 3.2.2 will become:

$$\frac{\mathrm{d}\vec{K}_{man}}{\mathrm{d}t} = -\vec{v}_B \times \vec{L}_{man} + \vec{\tau}_B \tag{3.2.3}$$

Now will be expressed the manipulator angular momentum in terms of the generalised coordinates of the base $\dot{\vec{q}}_b$ and manipulator $\dot{\vec{q}}_m$:

$$\vec{K}_{man} = H_{man} \begin{Bmatrix} \dot{\vec{q}}_b \\ \dot{\vec{q}}_m \end{Bmatrix} = \begin{bmatrix} H_{b,man} & H_{m,man} \end{bmatrix} \begin{Bmatrix} \dot{\vec{q}}_b \\ \dot{\vec{q}}_m \end{Bmatrix} \tag{3.2.4}$$

where H_{man} has been partitioned into $H_{b,man} \in \mathbb{R}^{3\times3}$ and $H_{m,man} \in \mathbb{R}^{3\times n_e}$.

Now Equation 3.2.2 will be numerically integrated (in a discretised manner) and putted together with Equation 3.2.4 also discretized:

$$\begin{aligned} \frac{\mathrm{d}\vec{K}_{man}}{\mathrm{d}t} &= \frac{\vec{K}_{man,k} - \vec{K}_{man,k-1}}{\Delta t} \\ \vec{K}_{man,k} &= \begin{bmatrix} H_{b,man}(\vec{q}_{k-1}) & H_{m,man}(\vec{q}_{k-1}) \end{bmatrix} \begin{Bmatrix} \dot{\vec{q}}_{b,k} \\ \dot{\vec{q}}_{m,k} \end{Bmatrix} \\ &= \vec{K}_{man,k-1} + (\vec{\tau}_{B,k-1} - \vec{v}_{B,k-1} \times \vec{L}_{man,k-1})\Delta t \end{aligned} \tag{3.2.5}$$

The generalised velocities of UAV $\dot{\delta q}_{b,k}$ can be expressed in terms of generalised velocities of manipulator $\dot{\vec{q}}_{m,k}$ and of the angular momenta of the UAM to obtain the constraint equation and so the inverse kinematics equation for the extended Jacobian, dropping the discretised notation for joint angles $(\vec{q}_{k-1})$ for brevity:

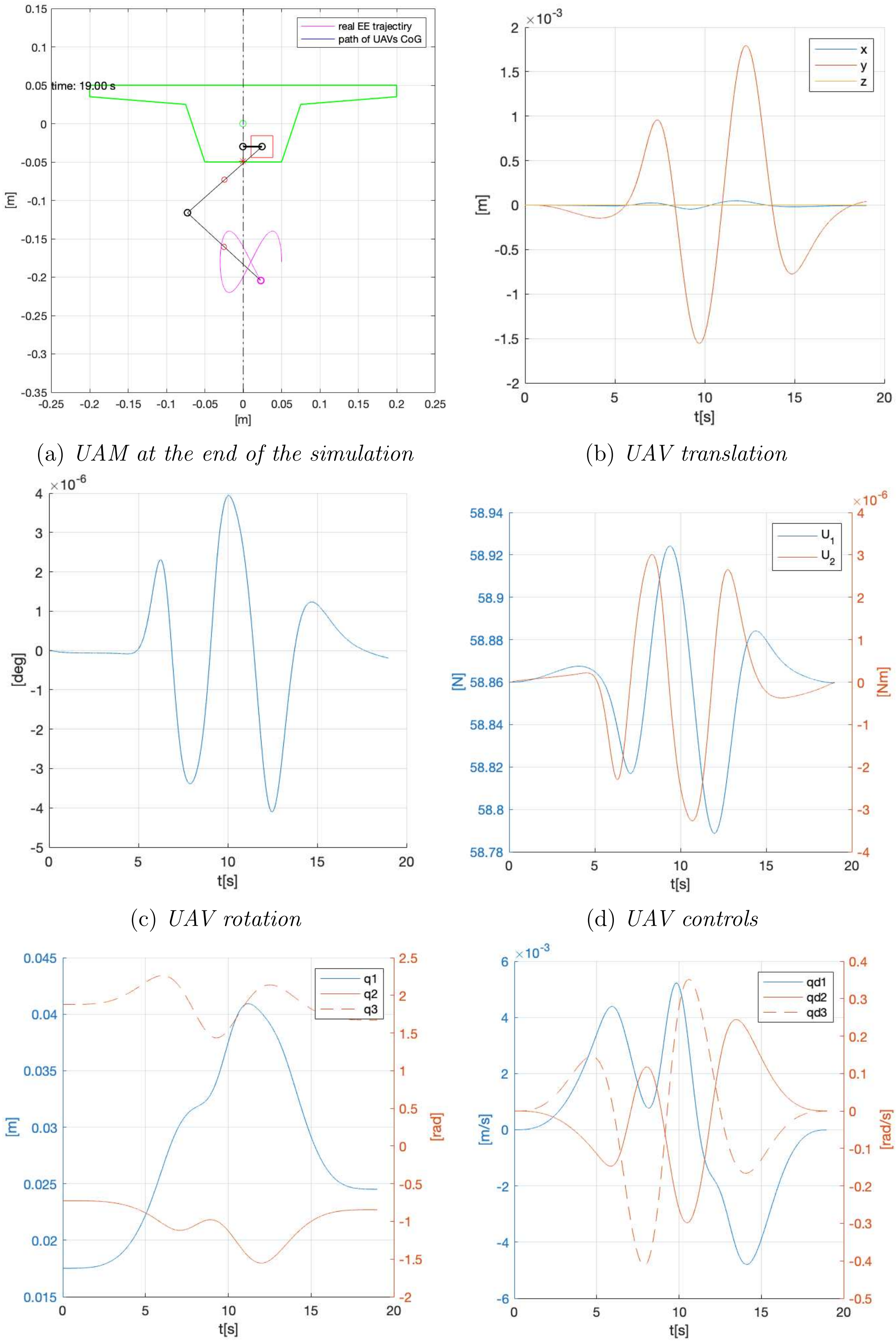
$$\begin{bmatrix} f_1 & f_2 & f_3 \end{bmatrix} = \begin{bmatrix} H_{b,man}\, H_b^{-1}\, H_m + H_{m,man} \end{bmatrix} \dot{\vec{q}}_{m,k} \tag{3.2.6}$$

$$b = \left(\vec{K}_{man,k-1} + \left(\vec{\tau}_{B,k-1} - \vec{v}_{B,k-1} \times \vec{L}_{man,k-1} \right) \Delta t - H_m^T\, H_b^{-1} \begin{Bmatrix} \vec{P}_k \\ \vec{K}_k \end{Bmatrix} \right) \tag{3.2.7}$$

where Equation 3.2.6 and Equation 3.2.7 form the added constraint equation needed to make the Jacobian square introduced in Equation 3.2.1.

In Figure 3.4 and Figure 3.5 are showed meaningful results of of the simulation of $19\,s$ and $10\,s$, respectively. Figure 3.4c and Figure 3.5c highlight the lowest UAV roll angle achieved so far, while the displacements of the UAV's CoG showed in Figure 3.4b and Figure 3.5b present good results and are comparable with the other dynamic and static balancing method previously analysed. The generalised velocities (reported in Figure 3.4f, Figure 3.5f) and coordinates (reported in Figure 3.4e, Figure 3.5e) are feasible and with acceptable values, so it proved to be a good configuration for balancing the UAM. The only drawback of this configuration may be the reduced workspace instead of that with the sliding battery decoupled from the manipulator, but this will be described in section 3.4.

The simulations in the Adams View® environment are showed in **??** and confirm the results of simulations performed in MatLab®.



(a) *UAM at the end of the simulation*

(b) *UAV translation*

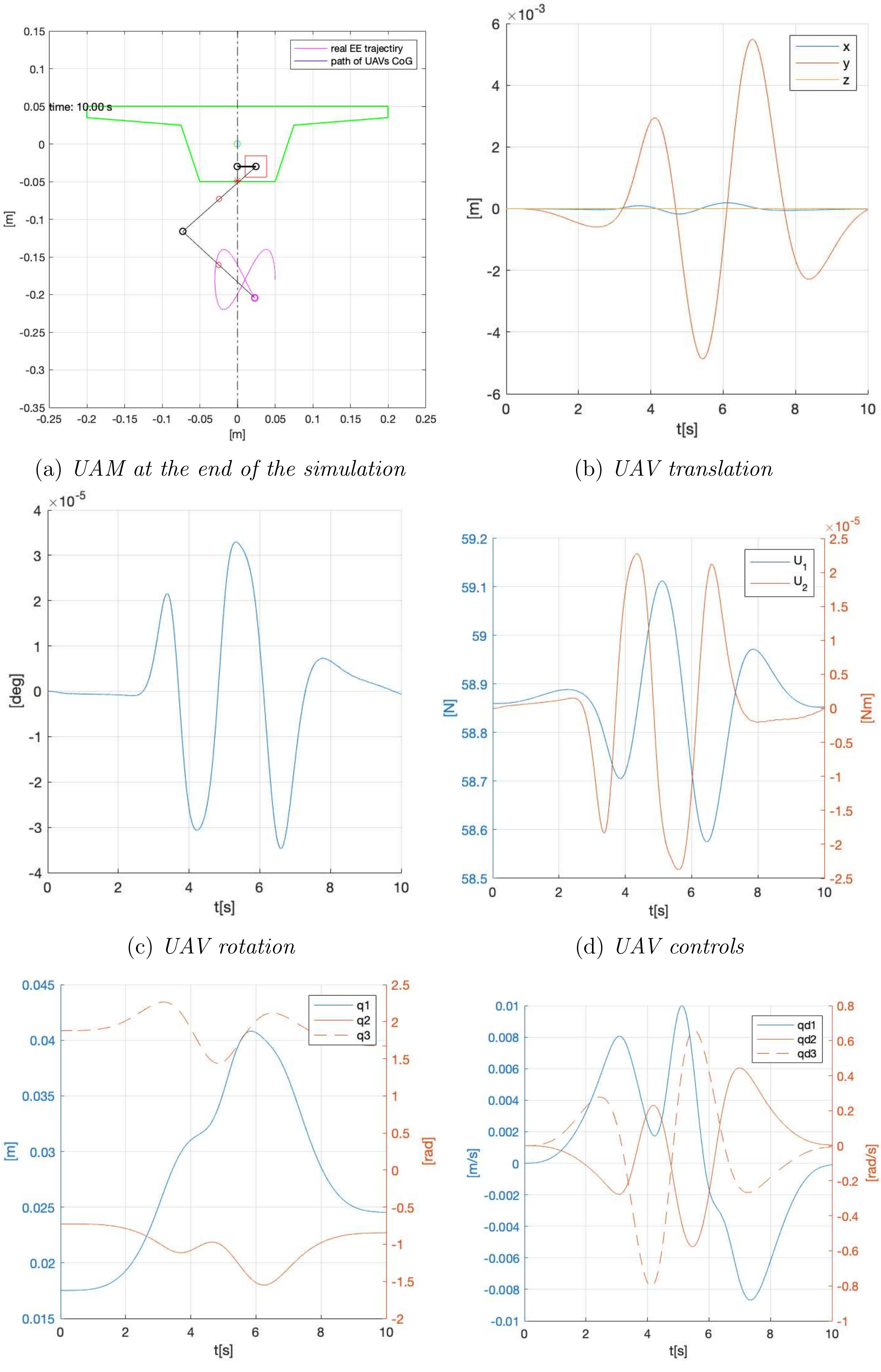
(c) *UAV rotation*

(d) *UAV controls*

(e) *manipulator's joint positions*

(f) *manipulator's joint velocities*

Figure 3.4: results of UAM with arm on slider for an infinity-shaped end-effector trajectory in 19 s



(a) *UAM at the end of the simulation*

(b) *UAV translation*

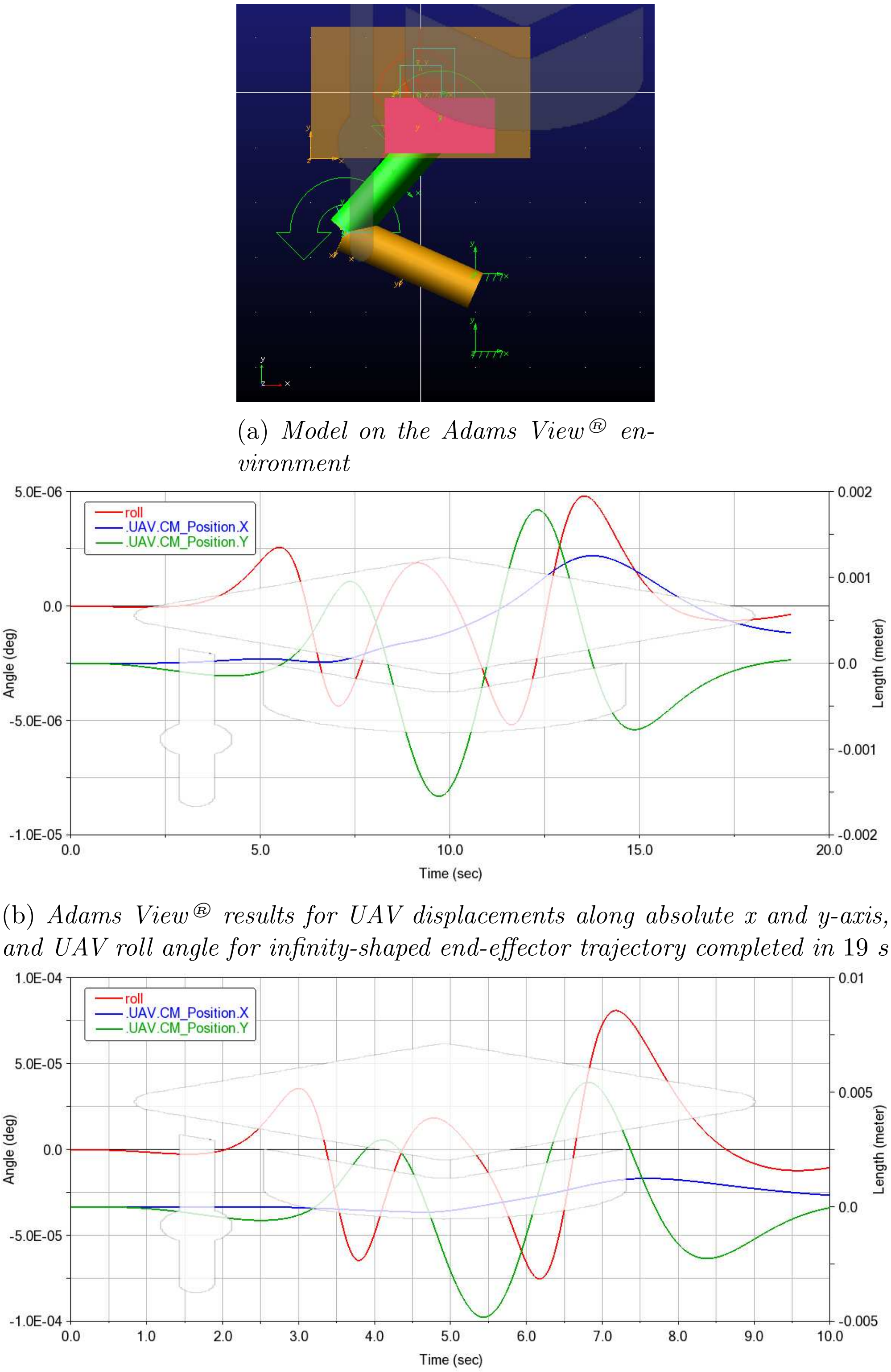
(c) *UAV rotation*

(d) *UAV controls*

(e) *manipulator's joint positions*

(f) *manipulator's joint velocities*

Figure 3.5: results of UAM with arm on slider for an infinity-shaped end-effector trajectory in 10 s



(a) *Model on the Adams View*® *environment*

(b) *Adams View*® *results for UAV displacements along absolute x and y-axis, and UAV roll angle for infinity-shaped end-effector trajectory completed in* 19 *s*

(c) *Adams View*® *results for UAV displacements along absolute x and y-axis, and UAV roll angle for infinity-shaped end-effector trajectory completed in* 10 *s*

Figure 3.6: Simulations on Adams View ® of UAM with the manipulator mounted on the sliding battery

3.3 Sliding Battery

In this section will be used the same configuration of the UAM used with the manipulator plus the sliding battery in section 2.2, but now considering also the dynamic effects. Focusing on torques this model will take into account the manipulator's torques about the first joint of the manipulator (here coincident with the UAV's CoG) generated by the gravity force $\vec{F}_{man} = m_{man}\,\vec{g}$ (m_{man} as the total mass of the manipulator), and torques on the same point generated by manipulator's angular inertia. T is the total torque generated by the arm on the joint 1 (here UAV's CoG). The sliding battery can generates a torque on the UAV's CoG equal and opposite to that just described exerted by the manipulator on the UAV's CoG in two ways: exploiting the battery's force of gravity $\vec{F}_{c,g} = m_c\,\vec{g}$ (m_{man} as the mass of the battery) with its generalised coordinate projected on the horizontal plane (absolute x-axis in $\sum_A$) as lever arm; exploiting the battery force of inertia $\vec{F}_{c,d} = m_c\,\ddot{\vec{q}_c}$ with its distance from the UAV's CoG r_s (fixed position of the prismatic joint of the sliding battery from the UAV's CoG to the joint of the battery along its relative y axis). The total torque exerted from the manipulator on the UAV's CoG to compensate $\vec{T}$ is retrieved with the Inverse Dynamic computation by the Recursive Newton-Euler method on the joint 1 as did in section 3.1. The dynamic balancing equation in the plane will be:

$$T = q_s\ m_c\ g + \ddot{q}_s\ m_s\ r_s \tag{3.3.1}$$

To take into account the UAV roll angle φ, the real position of the sliding battery's generalized coordinate will be:

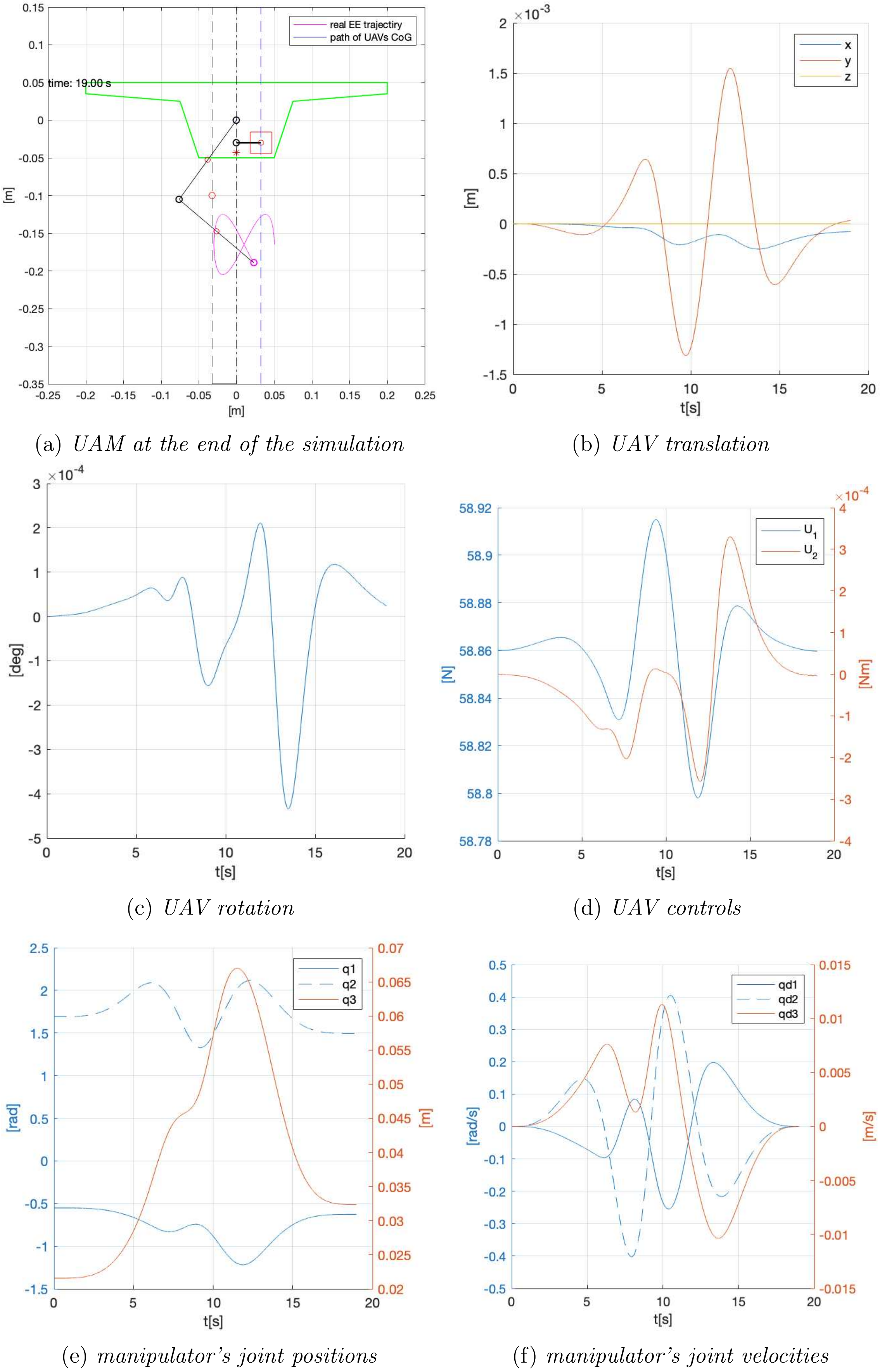
$$q_s = \frac{x_s}{cos(\varphi)} + r_s\ tan(\varphi) \tag{3.3.2}$$

So the acceleration of the sliding battery will be:

$$\ddot{q}_s = \frac{T - (q_s - r_s\ tan(\varphi))m_s\ g}{m_s\ r_s} \tag{3.3.3}$$

In Figure 3.7 and Figure 3.8 are showed meaningful results of of the simulation of $19\,s$ and $10\,s$, respectively. The maximum and minimum values for the UAV roll angle showed in Figure 3.7c and Figure 3.8c are two orders higher than those with the arm on sliding battery configuration in Figure 3.4c and Figure 3.5c, while the displacements of the UAV's CoG are slightly smaller than those of the arm on sliding battery configuration. The generalised velocities (reported in Figure 3.7f, Figure 3.8f) and coordinates (reported in Figure 3.7e, Figure 3.8e) are similar to that of the arm on sliding battery configuration with regard to the rotational joints, and more than double with regard to the translation joints, but still feasible and with acceptable values.

So the method presented in section 3.2 presents the best results with the simulations performed.



(a) *UAM at the end of the simulation*

(b) *UAV translation*

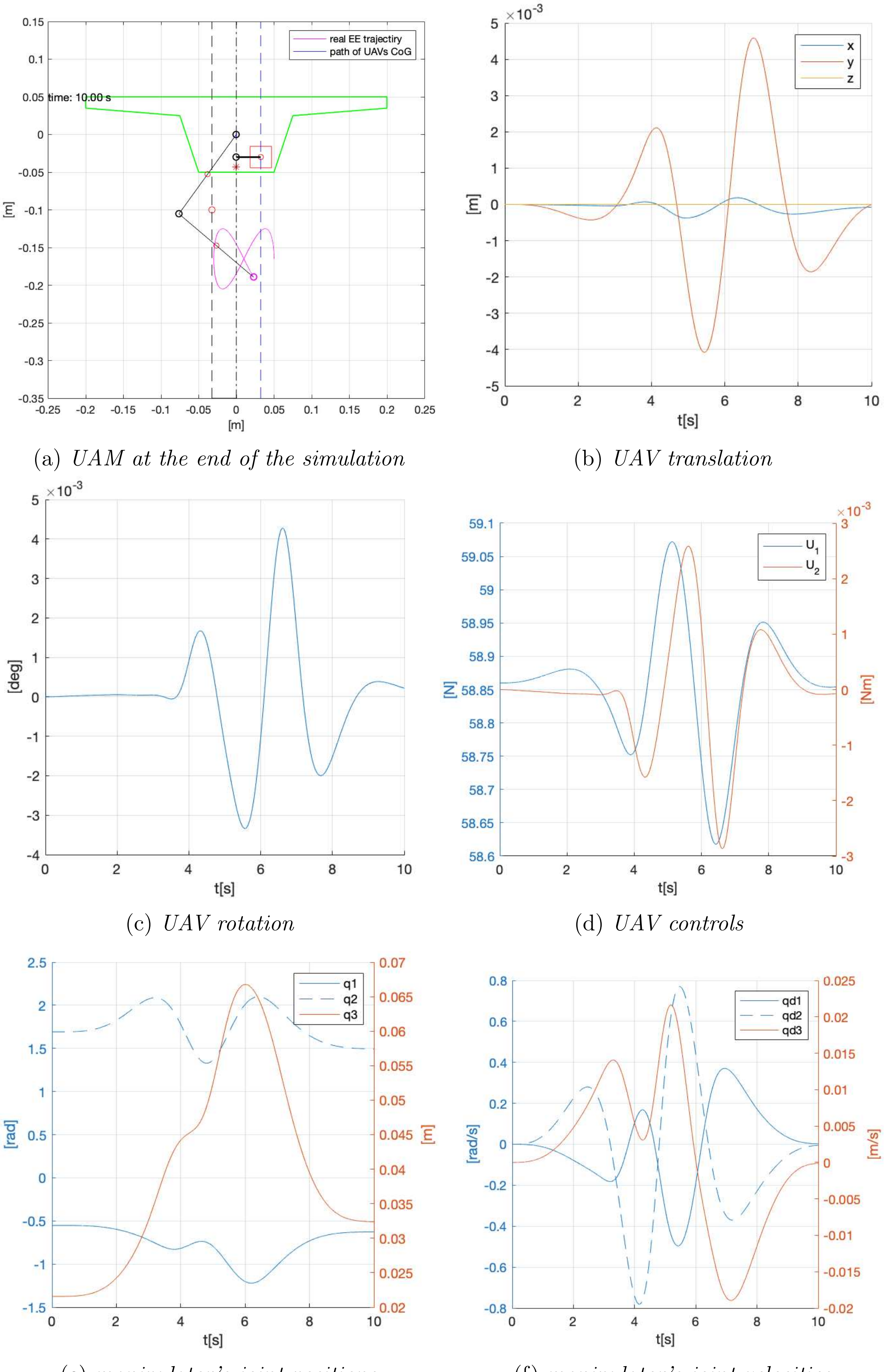
(c) *UAV rotation*

(d) *UAV controls*

(e) *manipulator's joint positions*

(f) *manipulator's joint velocities*

Figure 3.7: results of UAM with arm on slider for an infinity-shaped end-effector trajectory in 19 s



(a) *UAM at the end of the simulation*

(b) *UAV translation*

(c) *UAV rotation*

(d) *UAV controls*

(e) *manipulator's joint positions*

(f) *manipulator's joint velocities*

Figure 3.8: results of UAM with arm on slider for an infinity-shaped end-effector trajectory in 19 s

3.4 Workspace Comparison

As anticipated in section 3.2 the method of the 2dof manipulator mounted on the sliding battery there presented creates a 3dof manipulator with the first joint of translation, leading to better results than the method of the 2 dof manipulator with the decoupled sliding battery in section 3.3. The downside of the first method is that has a smaller working space than that of the sliding battery decoupled. To show that will be compared the maximums workspace for these two methods with a procedure similar to that used in [26] explained hereafter. Since the in aerial manipulation the manipulator has not a fixed base on which transfer forces and torques, its workspace does not depend only on the joint limits, robot structure and initial configuration. The floating base can change its position in the space due to the inertial effects of the manipulator when it is moving and to the UAV controls, thus varying the the distance between the desired manipulator's end-effector position and its base (the UAV). Therefore the conventional workspace analysis used for a fixed base manipulator, considering joints range of motion and velocities, will be augmented with a limit for the maximum acceptable UAV translation.

A straight trajectory for the end-effector will be defined in order to scan the workspace. The trajectory is divided into the initial part which consist of a parabolic acceleration (that means a smooth velocity trajectory) and the final part which consist of a constant velocity of indefinite duration. Will be simulated two different trajectories for the end-effector in order to see how these affect the workspaces, one will have to reach double of the speed in half of the time of the other. In Figure 3.9 are reported the two end-effector trajectories explained above stopped at the time when the end-effector will have travelled $0.5\,m$.

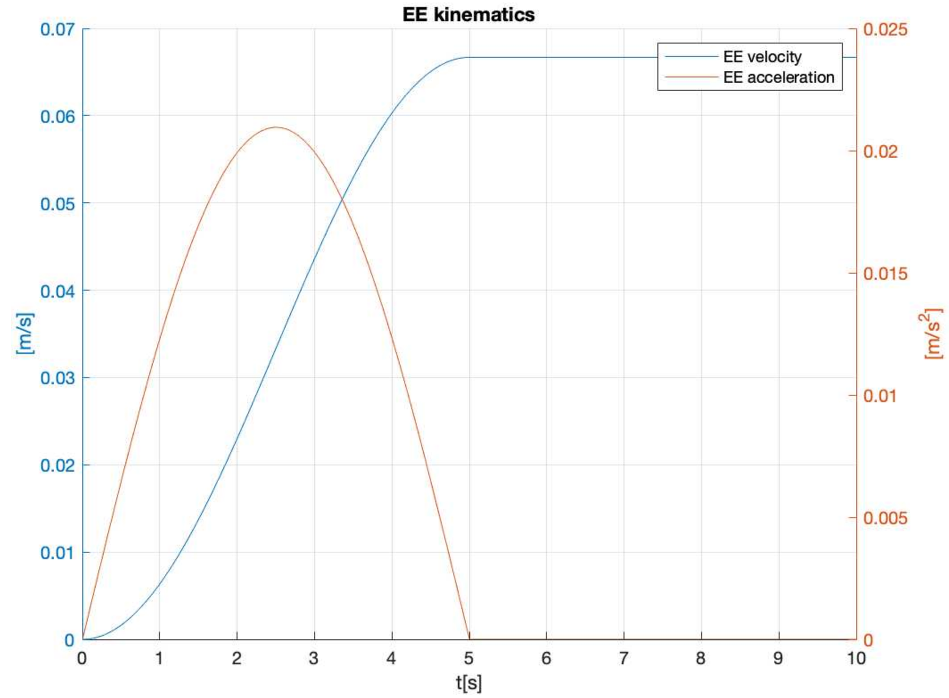
Starting from the initial equilibrium configuration of the UAM, this trajectory will be executed in all directions by rotating it around the end-effector initial point. 60 equally-spaced lines will be used to define the two workspaces. The algorithms described for the arm on sliding battery in section 3.2 and for the dynamic balancing sliding battery in section 3.3 are applied to simulate each of these trajectories. Since the desired trajectories have infinite lengths the simulation will be stopped in three events that are:

- joint limits exceed a threshold value which will be ±90° for the first revolute joint of the manipulator, ±140° for the second revolute joint of the manipulator, and ±0.125 m for the translational joint;

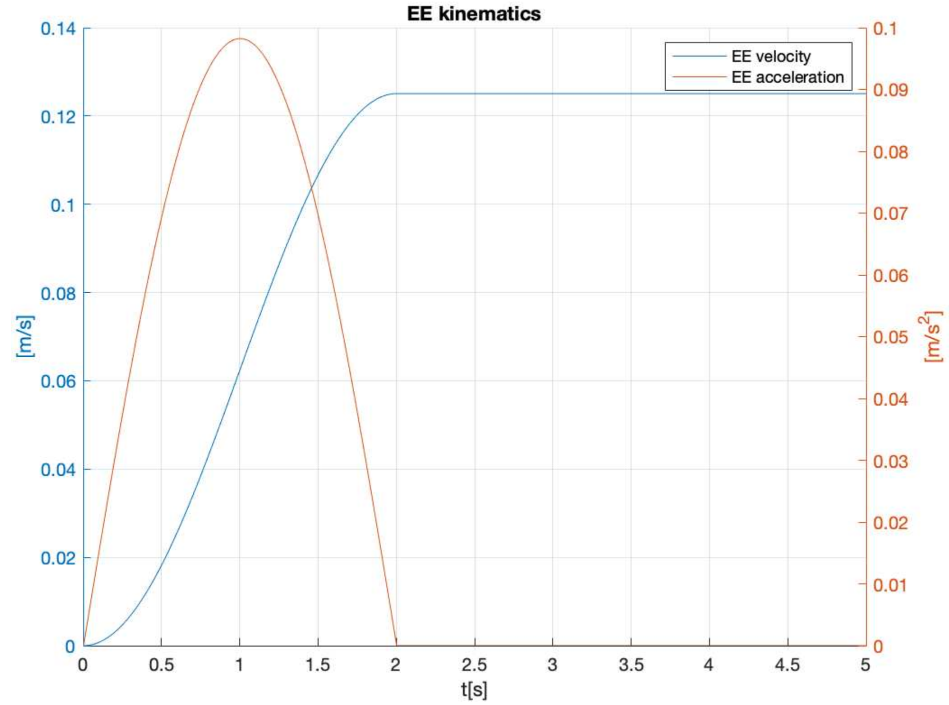
- joint velocities exceed a threshold value, which consider the case in which the manipulator is approaching a singular configuration. These threshold values will be $\pm 2\ rad/s$ for the revolute joints of the manipulator, and $\pm 0.3\ m/s$ for the translational joint;

- UAV motions, in particular when the horizontal translation of the UAV exceeds the threshold value of 0.02 m;

The workspace limits computed are reported in Figure 3.10 with the stopping criteria type for each trajectory indicated with different colored markers. For both the manipulator configurations, can be seen that the workspace is not symmetric, this is due to the imposed initial conditions of the manipulators (the right arm configuration in these cases) that do not change in the simulation limiting the maximum workspace. An inverted initial configuration would invert the workspace limits. Only the joint position limits and joint velocity limits are responsible for the workspaces here defined for these end-effector trajectories (magenta and blue circles in Figure 3.10), without the joints position limits the workspaces would be similar to those showed in Figure 3.10c and Figure 3.10d but with the part of the workspace in the positive x-axis mirroring the y-axis. Thus, presenting a larger workspace for the configuration with the decoupled sliding battery and a slighty larger one for the configuration with the manipulator on the sliding battery. Looking at the decoupled dynamic sliding battery configuration, the workspace is more extended for trajectories in the upper direction (y positive semi-axis) and wider than the that of the other configuration. The lower part of the workspaces is very similar.

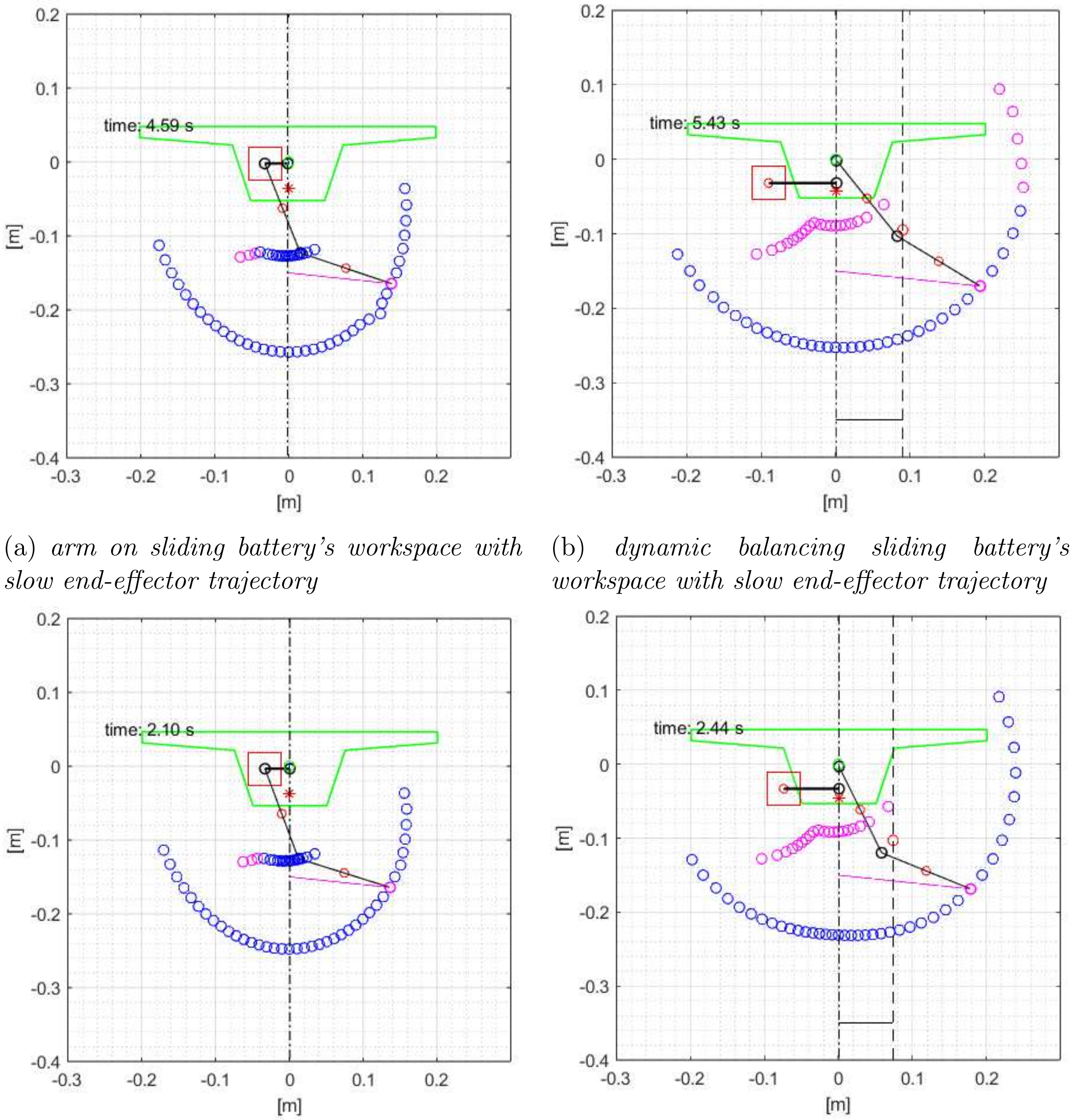


(a) *velocity and acceleration of the slow end-effector trajectory*



(b) *velocity and acceleration of the fast end-effector trajectory*

Figure 3.9: imposed end-effector trajectory for the workspaces comparison between arm on sliding battery model and decoupled dynamic sliding battery model



(a) *arm on sliding battery's workspace with slow end-effector trajectory*

(b) *dynamic balancing sliding battery's workspace with slow end-effector trajectory*

(c) *arm on sliding battery's workspace with fast end-effector trajectory*

(d) *dynamic balancing sliding battery's workspace with fast end-effector trajectory*

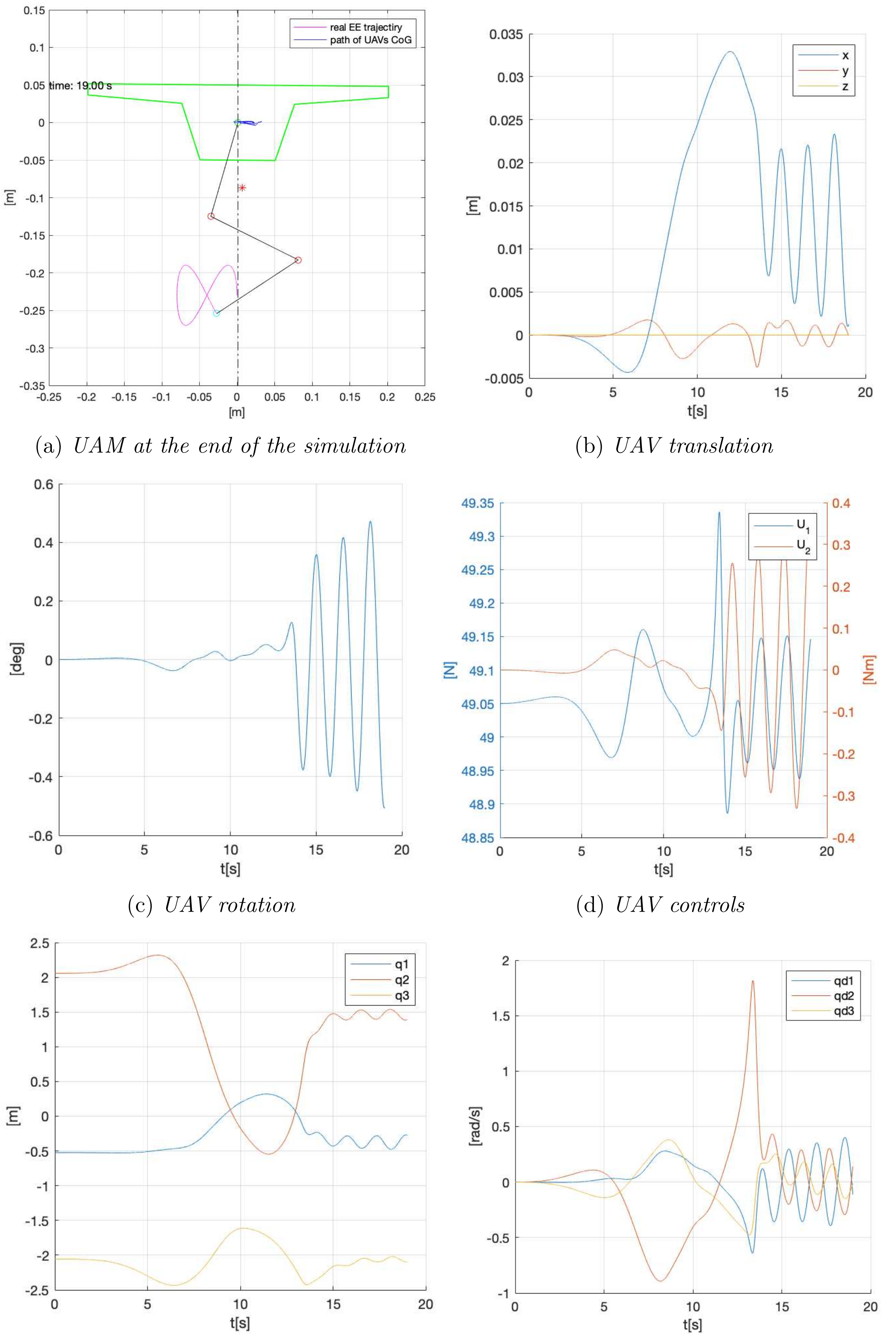
Figure 3.10: workspaces comparison between arm on sliding battery model and decoupled dynamic sliding battery model with the slow and fast end-effector trajectory, where markers indicate the stopping position while color refers to the stopping condition: magenta for excessive joint positions, and blue for excessive joint velocities. The UAM final configuration is shown (green drone outline), manipulator's/links' CoG (red asterisk), joints (black circles), links (black lines), link CoGs (red circles), end-effector position (magenta circle), last end-effector path (magenta line).

4. Kinematically Redundant Robotic Arm

In this chapter the degrees of freedom of some of the balancing methods previously explained will be extended. As the results in [14] have proved, the use of the extended Jacobian for the resolution of the inverse kinematics (as has been explained and done in section 3.2) seems an effective method to reduce the rotation of the UAV, generated by the manipulator disturbances. Therefore this approach will be used for the extension: of the UAM model with no compensation mechanisms showed in section 2.1; of the static balancing sliding battery UAM model in section 2.2; and of the manipulator mounted on the sliding battery in section 3.2 with an additional constraint. These new methods will not be simulated in Adams View® considering them as extensions of previously validated methods.

4.1 Initial UAM model

For the 3 DoF manipulator without compensation mechanisms the extended Jacobian used will be the same already seen in Equation 3.2.1 with the same constraint function Equation 3.2.6 and Equation 3.2.7. The results from the simulations are showed in Figure 4.1 and in Figure 4.2. The result of the UAV roll angle for the slow end-effector trajectory in Figure 4.1c show higher values than those obtained from the 2 DoF manipulator without compensation mechanisms in Figure 2.2a, instead the results for the fast end-effector trajectory in Figure 4.2c are slightly better. This show an interesting property of this UAM configuration which is a strong roll stability of the UAV within a certain range. As regards the UAV translation this method show better results for both trajectories (showed in Figure 4.1b and Figure 4.2b) respect the 2 DoF corresponding mechanisms. Anyway, in the fast trajectory the second joint reaches 3 rad/s which is the highest value so far and may be a little to high for a manipulator's joint. The Damped Least Square technique has been used to reduce the joint velocities (and also to avoid the singularity problem) for the same simulations. It consists in balancing the cost of a large solution ($\dot{\vec{q}}$ joint velocities) against the cost of a large residual error ($\vec{V}_{ext} - J_{ext}\ \dot{\vec{q}}$) using a damping factor to achieve higher joint velocities at the expense of the residual error and vice versa [27]. However, this technique gave higher UAV roll angle results than the one satisfying $\vec{V}_{ext} = J_{ext}\ \dot{\vec{q}}$.



(a) *UAM at the end of the simulation*

(b) *UAV translation*

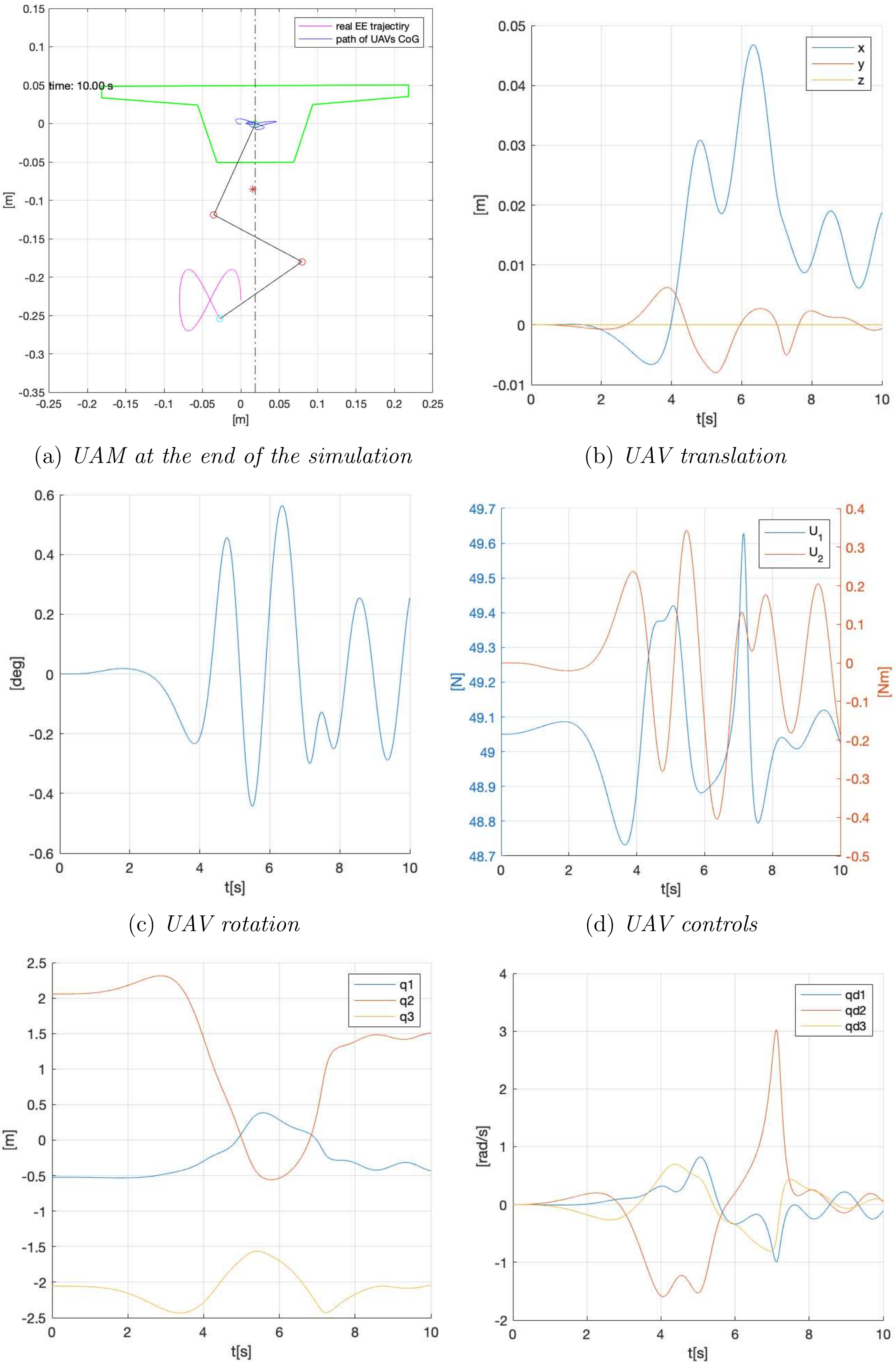
(c) *UAV rotation*

(d) *UAV controls*

(e) *manipulator's joint positions*

(f) *manipulator's joint velocities*

Figure 4.1: results of UAM with 3 DoF manipulator without compensation mechanisms for an infinity-shaped end-effector trajectory in 19 s



(a) *UAM at the end of the simulation*

(b) *UAV translation*

(c) *UAV rotation*

(d) *UAV controls*

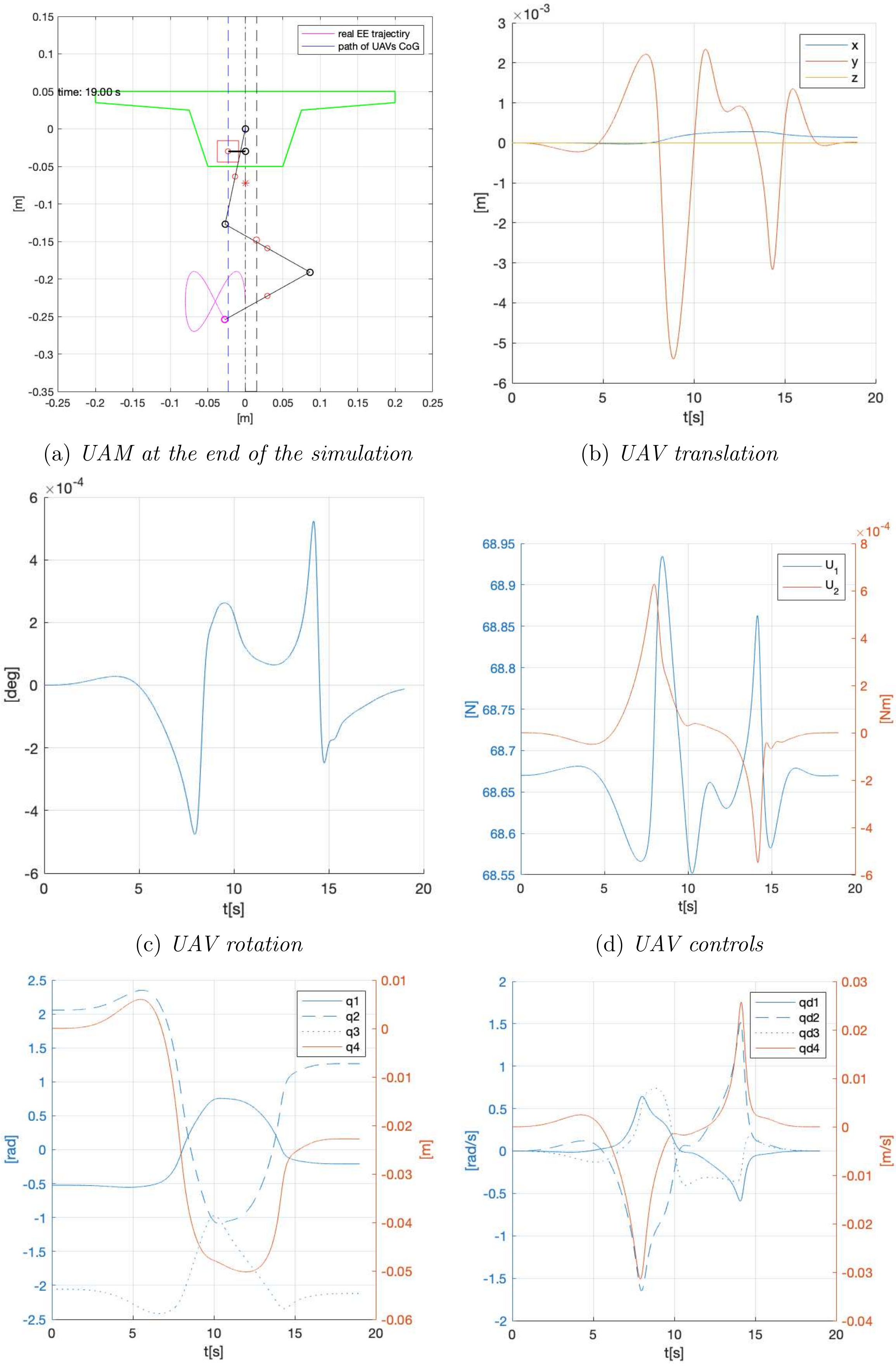
(e) *manipulator's joint positions*

(f) *manipulator's joint velocities*

Figure 4.2: results of UAM with 3 DoF manipulator without compensation mechanisms for an infinity-shaped end-effector trajectory in 10 s

4.2 Static Balancing with Sliding Battery

For the 3 DoF manipulator with static balancing sliding battery the extended Jacobian used will be the same used in the method just described augmented in the same way did in section 2.2, with Equation 2.2.16 and Equation 2.2.17 where in this case the J^* become the J_{ext} utilized in the 3 DoF manipulator without compensation mechanisms, and the generalised velocity of the sliding battery $\dot{q}_c = \dot{q}_4$ will be the same described in Equation 2.2.4 but with q_c described in Equation 2.2.3 which considers also the additional DoF of the manipulator. The results from the simulations are showed in Figure 4.3 and in Figure 4.4. Compared to the corresponding 2 DoF manipulator configuration this one has almost an order of magnitude lower with regard to the UAV roll angle as can be seen in Figure 4.3c and Figure 4.4c. Instead, the translations of the UAV are similar. As for the 3 DoF manipulator without compensation mechanisms here previously analysed the generalised velocity of the manipulator's 2^nd^ joint reaches almost 3 rad/s.



(a) *UAM at the end of the simulation*

(b) *UAV translation*

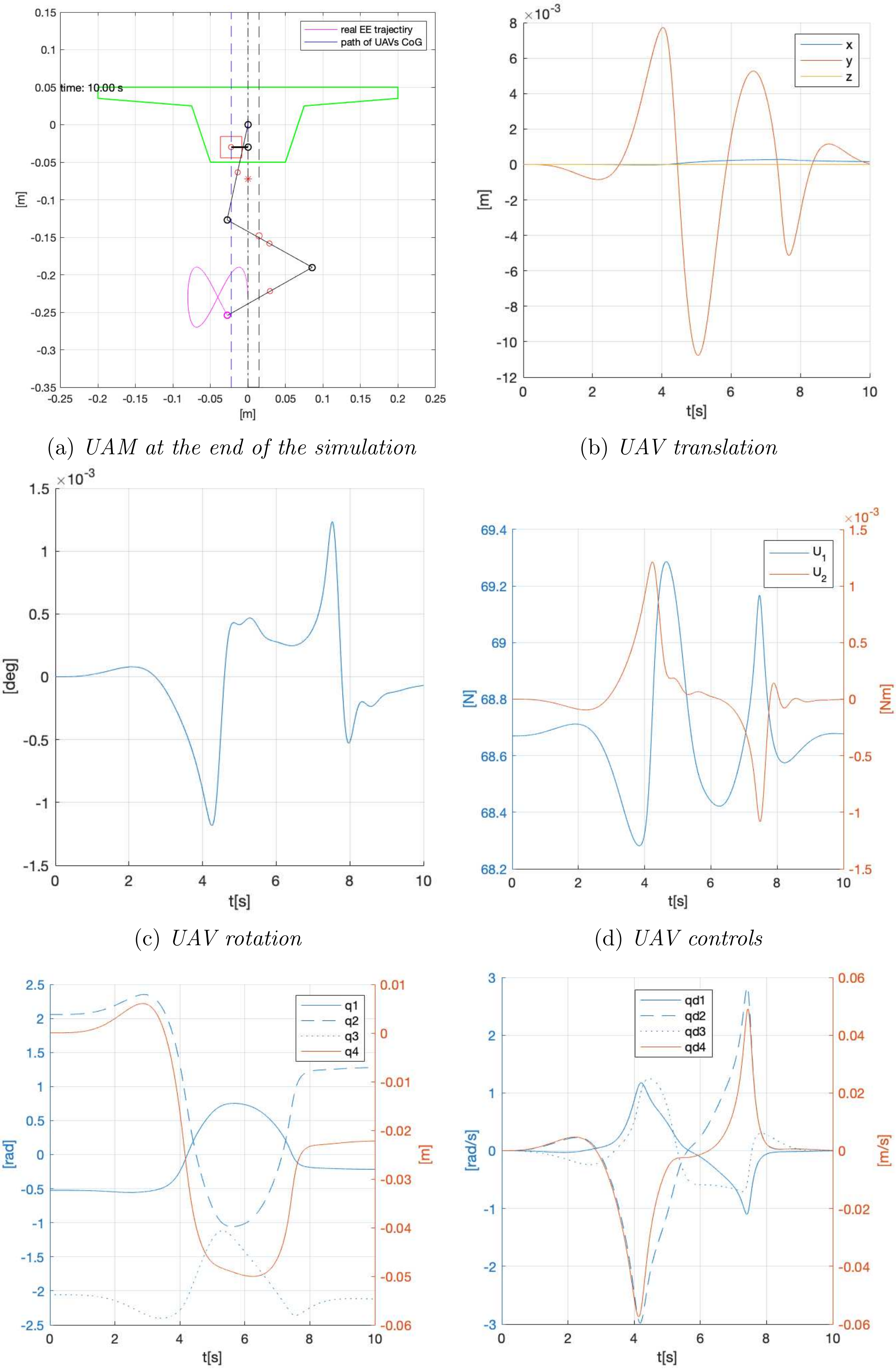
(c) *UAV rotation*

(d) *UAV controls*

(e) *manipulator's joint positions*

(f) *manipulator's joint velocities*

Figure 4.3: results of UAM with 2 DoF manipulator with static balancing sliding battery for an infinity-shaped end-effector trajectory in 19 *s*



(a) *UAM at the end of the simulation*

(b) *UAV translation*

(c) *UAV rotation*

(d) *UAV controls*

(e) *manipulator's joint positions*

(f) *manipulator's joint velocities*

Figure 4.4: results of UAM with 2 DoF manipulator with static balancing sliding battery for an infinity-shaped end-effector trajectory in 10 s

4.3 Manipulator Mounted on the Sliding Battery

Now a DoF will be added to the configuration of the UAM with 2 DoF manipulator mounted on the sliding battery as seen in section 3.2. As the previous two methods analysed in this chapter had high values of generalised velocities, the excessive DoF (the 4th) will be utilized to reduce joint velocities with the Moore-Penrose jeneralised inverse as explained in [28]. Here 3 DoF are used for the end-effector trajectory tracking (2 DoF for the 2D Cartesian space) and for the minimization of torque on the UAV (1 DoF), and 1 DoF remains free, so is a redundant manipulator ($r < n$ with r is the number of tasks and n is the number of manipulator's DoF). So the solutions $\dot{\vec{q}}$ have to satisfy the linear equations $\vec{V}_{ext} = J_{ext}\ \dot{\vec{q}}$ described in Equation 3.2.1 and minimize the quadratic cost function of joint velocities:

$$g(\dot{\vec{q}}) = \frac{1}{2}\dot{\vec{q}}^{\,T}\ W\ \dot{\vec{q}}$$

where W $(n \times n)$ is a symmetric positive definite weighting matrix. In order to solve this problem the method of Lagrange multipliers can be involved, whereby the cost function becomes:

$$g(\dot{\vec{q}}, \vec{\lambda}) = \frac{1}{2}\dot{\vec{q}}^{\,T}\ W\ \dot{\vec{q}} + \vec{\lambda}^{\,T}\ (\vec{V}_{ext} - J_{ext}\ \dot{\vec{q}})$$

where $\vec{\lambda}$ $(r \times 1)$ is a vector of unknown multipliers that allows the constraint in Equation 3.2.1 to be incorporated into the function to minimize. The solution sought has to satisfy the necessary conditions:

$$\left(\frac{\partial g}{\partial \dot{\vec{q}}}\right)^T = \vec{0} \quad , \quad \left(\frac{\partial g}{\partial \vec{\lambda}}\right)^T = \vec{0} \tag{4.3.1}$$

from which, the first condition leads to:

$$\begin{aligned} W\ \dot{\vec{q}} - J_{ext}^T\ \vec{\lambda} &= \vec{0} \\ \dot{\vec{q}} &= W^{-1}\ J_{ext}^T\ \vec{\lambda} \end{aligned} \tag{4.3.2}$$

where the inverse of W exists, and the solution Equation 4.3.2 is a minimum since $\partial^2\ g/\partial\ \dot{\vec{q}}^{\,2} = W$ is a positive definite; and the second condition in Equation 4.3.1 leads to the constraint:

$$\vec{V}_{ext} = J_{ext}\ \dot{\vec{q}} \tag{4.3.3}$$

Combining the two conditions Equation 4.3.2 and Equation 4.3.3 one obtains:

$$\vec{V}_{ext} = J_{ext}\ W^{-1}\ J_{ext}^{T}\ \vec{\lambda}$$

assuming that J_{ext} has full rank, the matrix $J_{ext}\ [W]^{-1}\ J_{ext}^{T}\ (r \times r)$ is a squared materix of rank r and is therefore invertible. Solving for $\vec{\lambda}$:

$$\vec{\lambda} = \left(J_{ext}\ W^{-1}\ J_{ext}^{T}\right)^{-1}\ \vec{V}_{ext} \tag{4.3.4}$$

which replacing it into Equation 4.3.2 the optimal solution can be obtained:

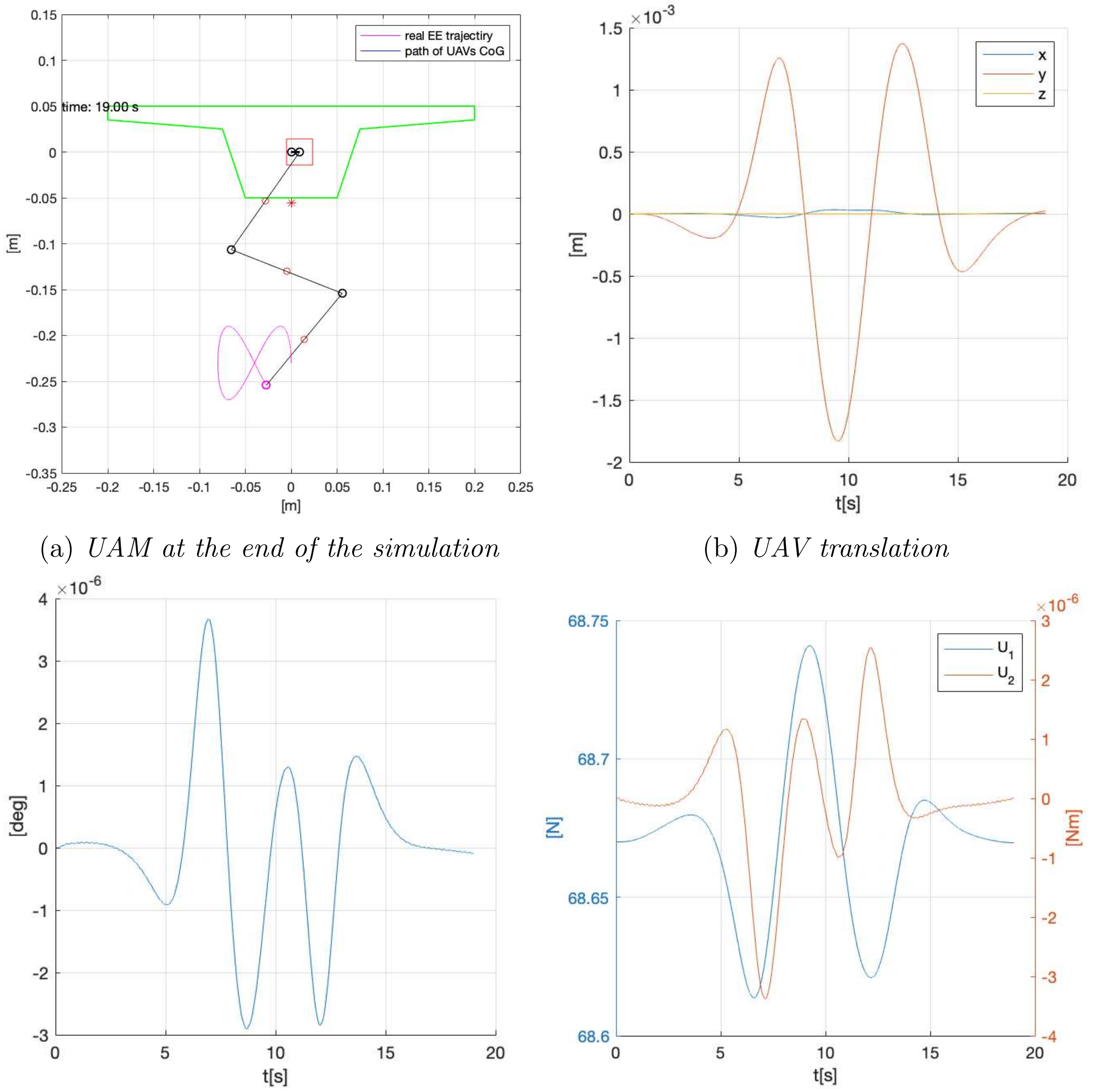
$$\dot{\vec{q}} = W^{-1}\ J_{ext}^{T}\ \left(J_{ext}\ W^{-1}\ J_{ext}^{T}\right)^{-1}\ \vec{V}_{ext} \tag{4.3.5}$$

Now will be introduced the Moore-Penrose inverse, also called pseudo-inverse. A particular case of the procedure just treated is that in which the weighting matrix W is the identity matrix I and the solution in Equation 4.3.5 simplifies into:

$$\dot{\vec{q}} = J_{ext}^{\dagger}\ \vec{V}_{ext} \tag{4.3.6}$$
$$J_{ext}^{\dagger} = J_{ext}^{T}\ \left(J_{ext}\ J_{ext}^{T}\right)^{-1}$$

where $J_{ext}^{\dagger}$ is the right pseudo-inverse of J_{ext}, and the Equation 4.3.6 locally minimizes the norm of joint velocities.

The simulations were performed using the inverse kinematics expressed in Equation 4.3.6 and their results are showed in Figure 4.1 and in Figure 4.1. Compared to the results for the 2 DoF manipulator mounted on the sliding battery in Figure 3.4 and Figure 3.5 the UAV translation and roll angle are very similar. Only the velocity of the sliding battery $\dot{q}_1$ in Figure 4.5f and Figure 4.6f is a little bit higher while the other joint velocities $\dot{q}_2$ and $\dot{q}_3$ are a little bit smaller. As regards the joint positions in Figure 4.5e and Figure 4.5e the extended method has almost halved the excursion of the sliding battery q_1 and further reduced the first revolute joint of the manipulator q_2. Comparing this last method with the previous ones presented in this chapter the UAV translation has been further reduced and the UAV roll angle has been greatly reduced also having much lower joint velocities by keeping away from the limits.



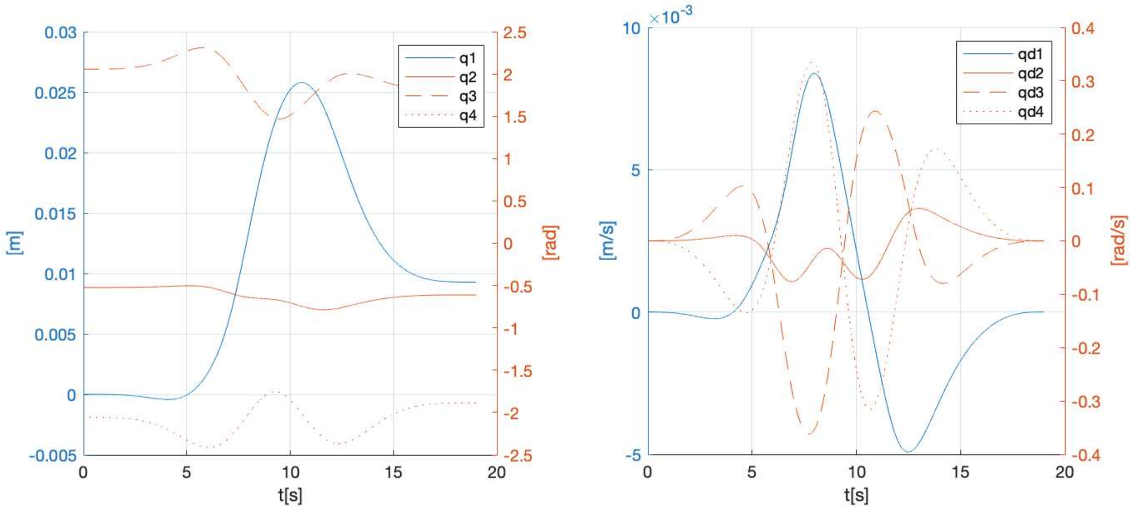
(a) *UAM at the end of the simulation*

(b) *UAV translation*

(c) *UAV rotation*

(d) *UAV controls*

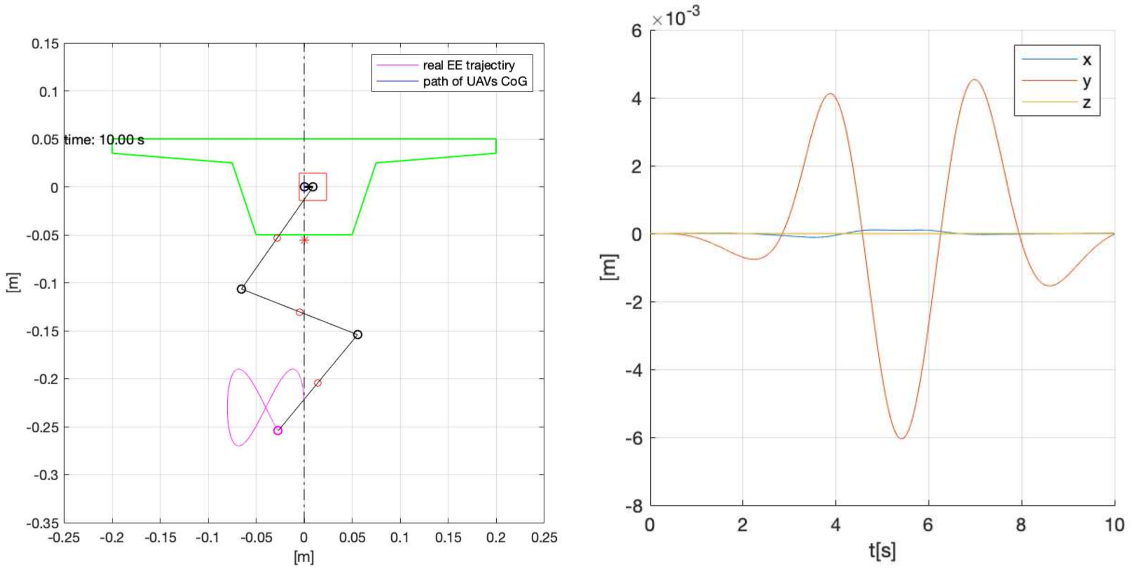
Figure 4.5: results of UAM with 3dof manipulator on sliding battery for an infinity-shaped end-effector trajectory in 19 s



(e) *manipulator's joint positions*

(f) *manipulator's joint velocities*

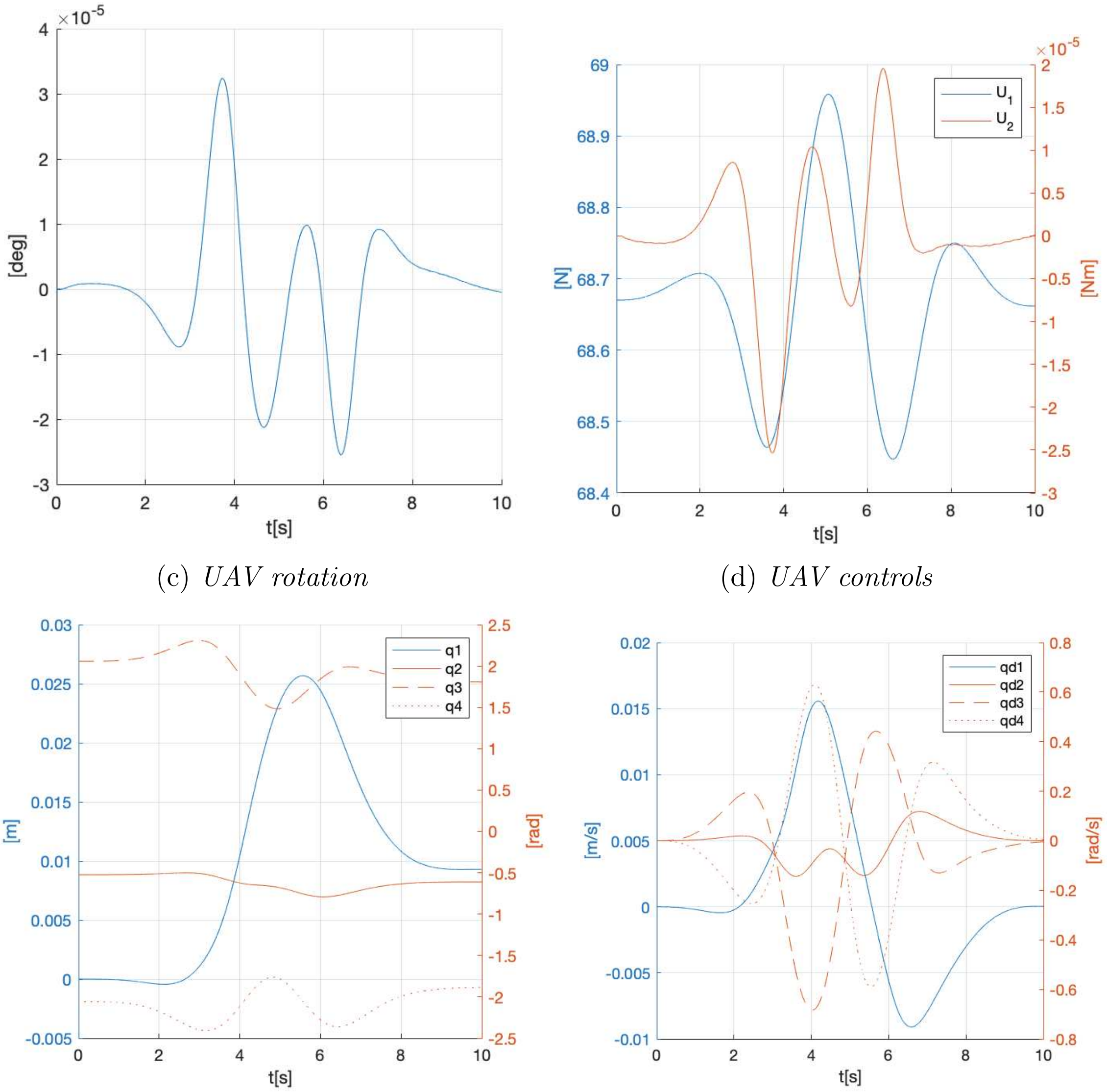
Figure 4.5: results of UAM with 3dof manipulator on sliding battery for an infinity-shaped end-effector trajectory in 19 s (cont.)



(a) *UAM at the end of the simulation*

(b) *UAV translation*

Figure 4.6: results of UAM with 3dof manipulator on sliding battery for an infinity-shaped end-effector trajectory in 10 s



(c) *UAV rotation*

(d) *UAV controls*

(e) *manipulator's joint positions*

(f) *manipulator's joint velocities*

Figure 4.6: results of UAM with 3dof manipulator on sliding battery for an infinity-shaped end-effector trajectory in 10 s (cont.)

5. Load Picking

An essential task in the aerial manipulation is the carrying and manipulation of loads as picking and placing tasks. Therefore, in this chapter, the approaching of the load with the free manipulator and the picking/lifting of it will be addressed.

5.1 Model Definition

Here the gripping mechanism of the manipulator (the gripper) will be disregarded, assuming a fixed connection between the load and the manipulator at the chosen time. The approach that will be used will be that described in [12]. Since the methods that have been seen so far use the generalized Jacobian for inverse kinematic control, a concentrated mass payload attached to the end-effector can be considered as additional external forces and torques generated by it. These new contributions will be accounted for in the terms $\vec{F}_k$ and $\vec{\tau}_k$ that varied the UAM's linear and angular momenta $\vec{L}_0 = cost$ in Equation 2.2.12. In the extended generalized Jacobian, the additional torque is also considered in the term $\vec{\tau}_B$ of Equation 3.2.3. The effect of the payload on the end-effector, the force load $\vec{F}_{laod,EE}$, consists of three terms: weight force, inertial force, and contact force. As in a real case scenario the load will be initially lay on surface and picked from there. Here will be used a model which replicates a virtual elastic spring without damping on top of a rigid surface and between the load and this virtual spring there will be zero friction. This will be made in order to eliminate discontinuities in force application, which reduces the inaccuracy of taking the force at the previous timestep how it has been done in all the simulations. After the payload is grasped by the manipulator, so the payload is added on the position of the end-effector, the load force applied on the end-effector is defined by the following equations and can thus be calculated at each timestep:

$$\vec{F}_{load,EE} = \begin{Bmatrix} 0 \\ m_l\ g \\ 0 \end{Bmatrix} - m_l \begin{Bmatrix} \ddot{\vec{x}}_l \\ \ddot{\vec{y}}_l \\ \ddot{\vec{z}}_l \end{Bmatrix} + \vec{F}_{con} \tag{5.1.1}$$

$$\vec{F}_{con} = \begin{Bmatrix} 0 \\ max(0, k_s(y_s - y_l)) \\ 0 \end{Bmatrix} \tag{5.1.2}$$

where m_l is the mass of the load, x_l, y_l, z_l are the Cartesian coordinates of the load, k_s is the contact spring stiffness, and y_s is the unloaded height of the contact spring.

5.2 Picking Trajectory

The best load picking configuration found consist in the end-effector move of almost $4\,cm$ on the absolute y-axis (reference $\sum_A$) and of almost $6\,cm$ on the absolute x-axis (reference $\sum_A$) through a linear trajectory which lasts $20\,s$ in the following stages (all the trajectory is imposed with smooth accelerations):

1. the approach of the manipulator to the payload laying on the soil is accomplished from $0\,s \leq t \leq 8.5\,s$. This part lasts so long to avoid fast accelerations and decelerations which affect the dynamic methods;

2. at $t = 8.5\,s$ acceleration and velocity of the end-effector will be null and the grasp starts adding the payload to the end-effector;

3. the backward motion begins immediately with a very slow and smooth acceleration to avoid also excessive lose in UAV's altitude due to the relative slowness of the UAV's PID control;

4. at $t \cong 14.95\,s$ there will be the complete detachment of the load form the soil where it was laying (as can be seen in Figure 5.1), so $\vec{F}_{con} = 0$, and the load is entirely sustained by the manipulator so will be increased the end-effector backward acceleration and thus its velocity;

For the simulation hereafter will be considered a load mass $m_l = 0.5\,kg$, a contact spring stiffness $k_s = 1000\,N\ m/s$, and y_s will be chosen such that the load is in

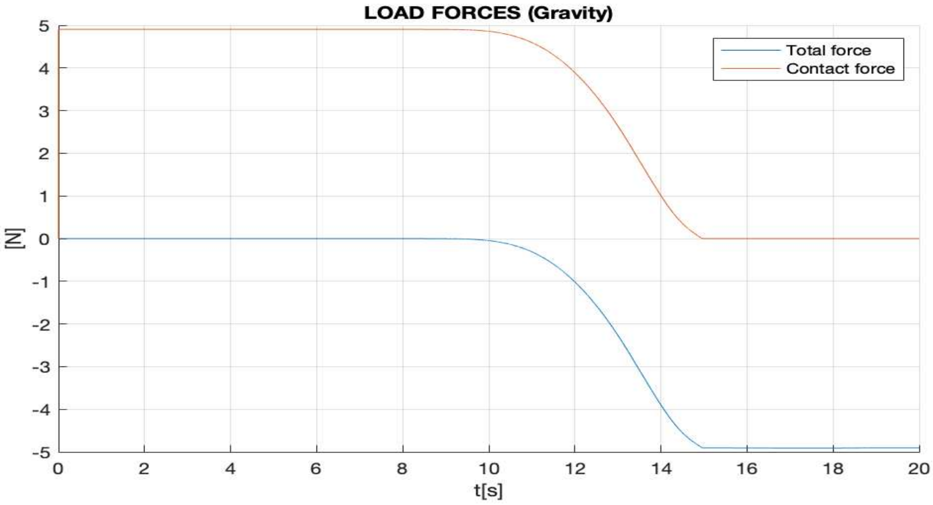
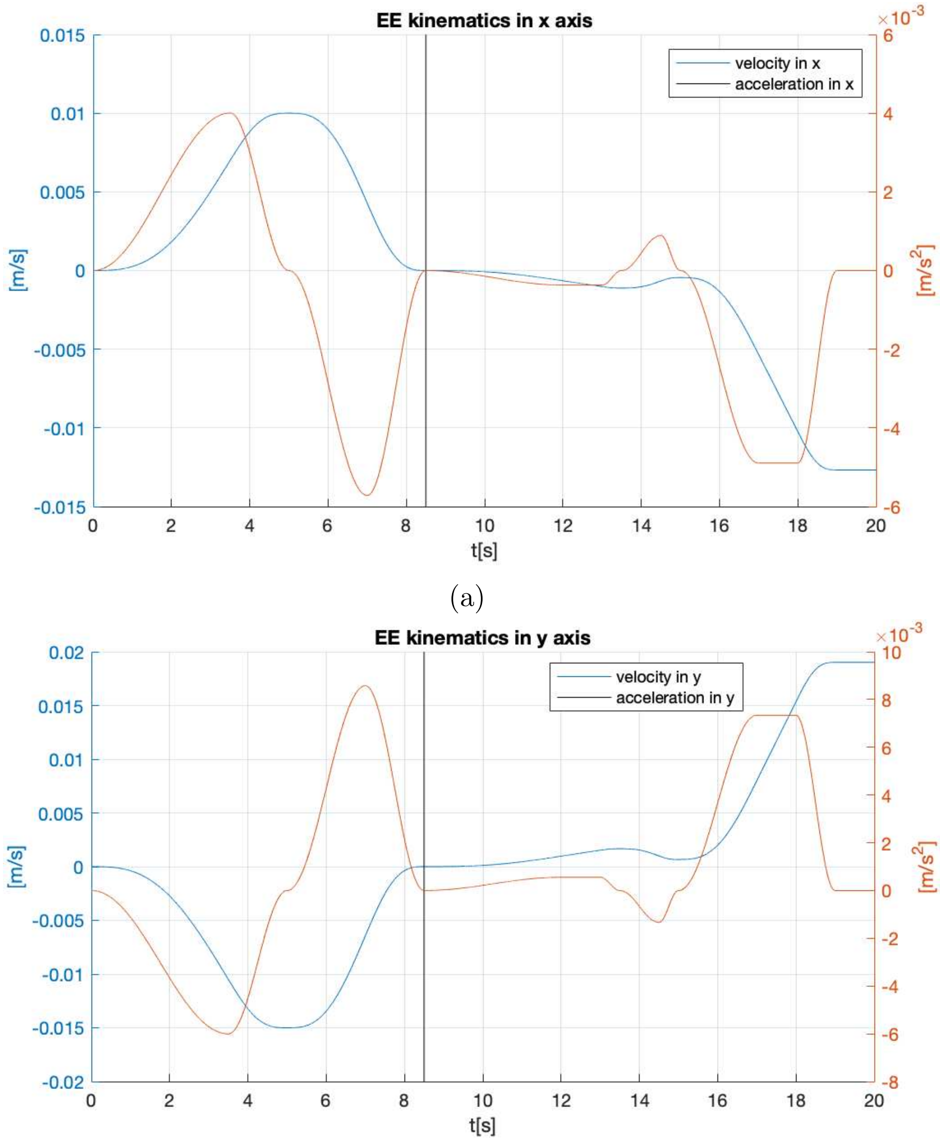


Figure 5.1: in red line is reported the force generated by the load on the soil and in blue line the force of the load on the end-effector over time, along y absolute axis for a load of $0.5\,kg$

equilibrium at its initial position (equal to the position of the end-effector at $t = 8.5\,s$). The end-effector acceleration and velocity profile for the load picking task is shown in Figure 5.2.



(a)

(b)

Figure 5.2: desired end-effector acceleration and velocity for the absolute x-axis, above, and for the y absolute axis, below. The vertical black line indicates the start time of contact at 8.5 s

5.3 Simulation Results

The UAM configurations here analysed will be: static balancing sliding battery with 2 and 3 serial manipulator's DoF; dynamic balancing sliding battery with 2 serial manipulator's DoF; 3 DoF manipulator with no compensation method for comparison; and 2 and 3 DoF serial manipulator mounted on the sliding battery. The UAV roll angle result for the simulation with the 2 DoF manipulator without compensation mechanisms is not reported because of failure to the instability with that load and PID control. Also the simulation with the dynamic balancing sliding battery with 3 serial manipulator's DoF is not mentioned because of the instability in the configuration used.

The comparison between simulations will consider only the roll angle of the UAV taking account that: the altitude of the UAV is almost identical for every simulation, the maximum or minimum horizontal displacement of the UAV is similar for every simulation except for the simulation with the 3 DoF serial manipulator mounted on the sliding battery and decoupled sliding battery, because them present less than half of the horizontal displacement of the other methods. With regard to the manipulators control limits all the simulations respects the joint velocity limits. The results defined above are shown in Figure 5.3, here one may notice the increasing improvement as regard the UAV roll angle:

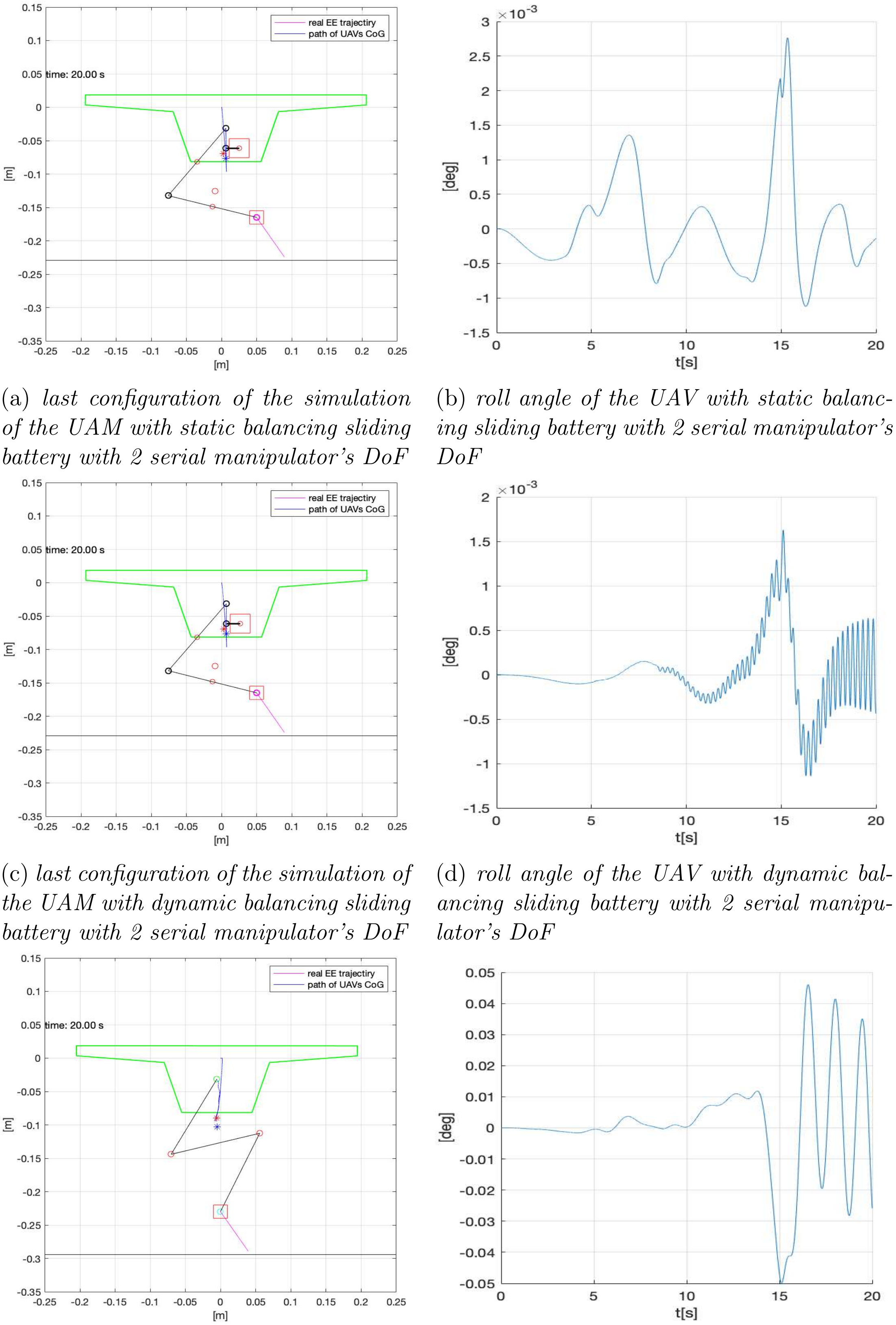
- the maximum value for the UAV roll angle from the 2 DoF manipulator with static balancing sliding battery (Figure 5.3b) to the 2 DoF manipulator with dynamic balancing sliding battery (Figure 5.3d) was almost halved, meanwhile the minimum value remained the same. The general trend in Figure 5.3d has less waves but increasing vibrations which became visible in the instant of load grasping because the battery has to moves faster to compensate the increasing mass (so force) of the end-effector + load.

- the simulation with the 3 DoF manipulator without compensation mechanisms is used to compare the following methods with 3 DoF. Anyway, respect to the 2 previous methods this one leads to maximum and minimum values at least 10 times higher, as showed in Figure 5.3f.

- the simulation with the 2 DoF manipulator mounted on the sliding battery gives better results as regards UAV roll angle than all the previous methods, going under $10^{-4\circ}$. As can be seen in Figure 5.3h there are vibrations which start form the instant of load grasping as in Figure 5.3d, but these vibrations are much smaller and increase in amplitude much less than those in the simulation with the dynamic balancing sliding battery.

- the simulation with the 3 DoF manipulator with a static balancing sliding battery shows a maximum and minimum UAV roll angle almost 5 times bigger than those of the 2 DoF manipulator mounted on the sliding battery, as can be seen in Figure 5.3j, but however much smaller than those of the previous method, and there are not vibrations.

- the last simulation is that with the 3 DoF manipulator mounted on the sliding battery and has the best result as regards the UAV roll angle (reported in Figure 5.3l) with maximum and minimum values at least 5 times lower than the second best method here compared.



(a) *last configuration of the simulation of the UAM with static balancing sliding battery with 2 serial manipulator's DoF*

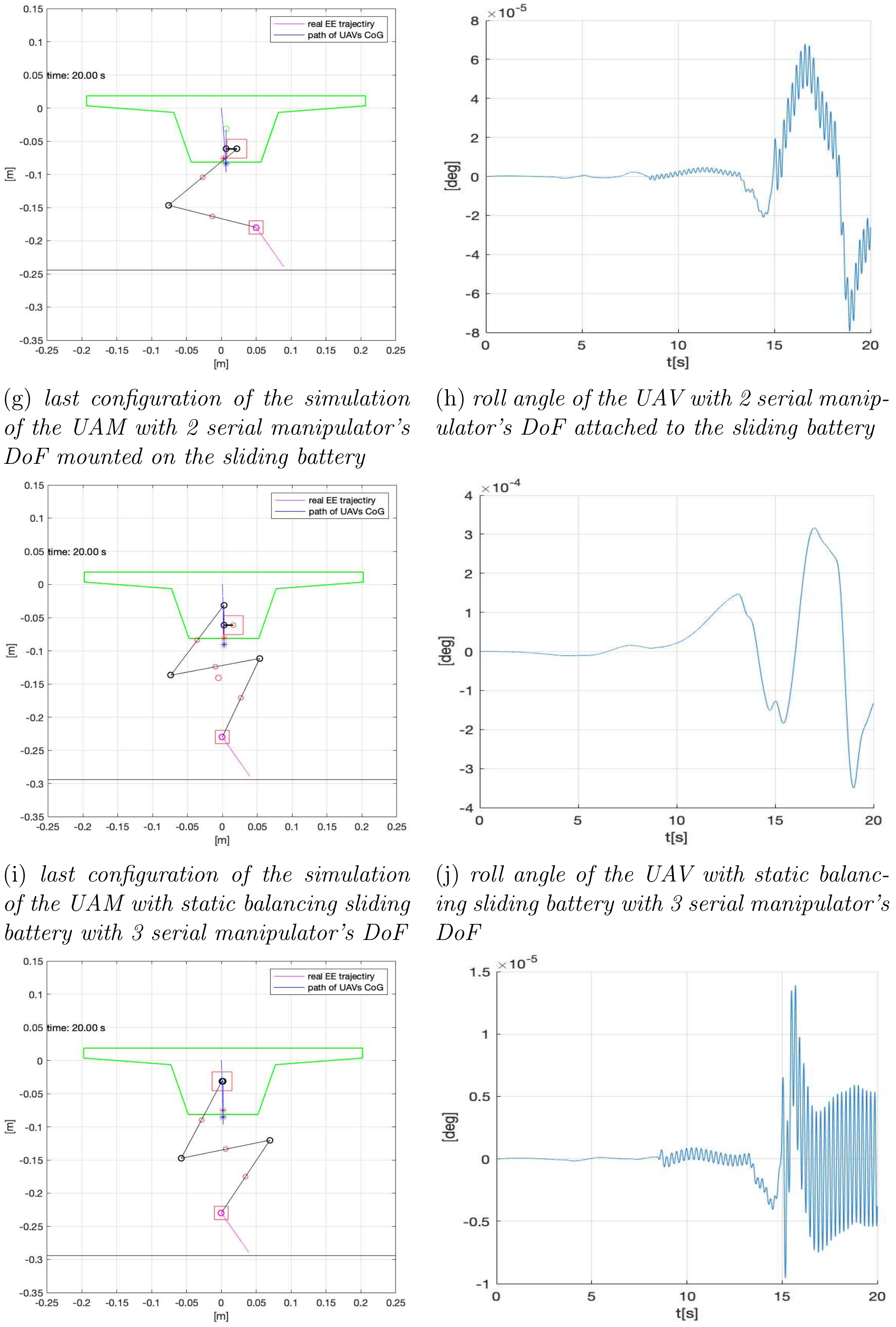
(b) *roll angle of the UAV with static balancing sliding battery with 2 serial manipulator's DoF*

(c) *last configuration of the simulation of the UAM with dynamic balancing sliding battery with 2 serial manipulator's DoF*

(d) *roll angle of the UAV with dynamic balancing sliding battery with 2 serial manipulator's DoF*

(e) *last configuration of the simulation of the UAM with 3 serial manipulator's DoF without compensation mechanisms*

(f) *roll angle of the UAV with 3 serial manipulator's DoF without compensation mechanisms*



(g) *last configuration of the simulation of the UAM with 2 serial manipulator's DoF mounted on the sliding battery*

(h) *roll angle of the UAV with 2 serial manipulator's DoF attached to the sliding battery*

(i) *last configuration of the simulation of the UAM with static balancing sliding battery with 3 serial manipulator's DoF*

(j) *roll angle of the UAV with static balancing sliding battery with 3 serial manipulator's DoF*

(k) *last configuration of the simulation of the UAM with 3 serial manipulator's DoF mounted on the sliding battery*

(l) *roll angle of the UAV with 3 serial manipulator's DoF attached to the sliding battery*

Figure 5.3: roll angle results from the load picking task simulations for the different balancing methods

6. Conclusions and Future Developments

The main objective of this thesis was design and analyse different types of configurations for the UAM in order to improve its stability during a task in hovering conditions. In particular, the focus was on the vertical plane, thus reducing the UAV roll angle, which would otherwise lead to an horizontal translation of the UAV.

The theory used to perform the inverse kinematics of the UAM's manipulator, taking into account the base (UAV) rotation and displacements due to the inertial effects, is based on the generalised Jacobian, and extended Jacobian for redundant manipulators. It was already implemented in a MatLab® script in which the new proposed designs were modeled and then validated as multibody models in Adams View®. The quantities and parameters used in this work came from the UAV DJI S1000 (held by the University of Padua), and from previous works and papers cited in the beginning of this thesis. The UAV was modelled in a perfect stationary flight without considering aerodynamic disturbances which are very significant in the reality;

In Chapter 2 were proposed and analysed simple UAM configurations to counterbalance the torque on the UAV generated by the gravity on the manipulator with an equal and opposite torque by means of a counterweight. These methods gave good results for the chosen end-effecotr trajectory. In particular the one with the sliding battery mechanism allow to exploit the weight of the battery that is usually a disadvantage, and seams more suitable than that with the pendulum because of its encumbrance. Finally, the UAM with dual arm could be used to improve the UAM stability in payloads handling.

In Chapter 3 were proposed and analysed two new UAM configurations and augmented the one with the sliding battery to take count also of the inertial effects of the manipulator and counterweights themselves. The value of the torque generated between the manipulator and the UAV, needed to counteract it with dynamic methods, was obtained through the inverse dynamics of the manipulator operated by the recursive Newton-Euler method in the MatLab® library. The best configuration resulted the manipulator mounted on the sliding battery, but has a narrower workspace than that of the UAM with the sliding battery detached from the manipulator.

The previous configurations had a maximum of 3 DoF, so were presented two new configurations augmented with an additional degree of freedom. The redundancy of the manipulators were handled with the extended Jacobian with a constraint to minimize torque between manipulator and UAV, and then transformed in a right pseudo-inverse to minimize joint velocities. The best method here remained the one with the manipulator mounted on the sliding battery. The workspace analysis was not performed with these new configurations.

In the final Chapter was performed a simple load picking task of a payload weighting $0.5\,kg$ from a frictionless elastic platform for almost all the proposed methods. The augmented methods performed better than the original methods, but the ones that use the dynamics of the manipulator show small vibrations that increase in amplitude at every abrupt changes in acceleration. Thus, the task could not be performed faster to increase the dynamic effects.

Future developments of this work could be:

- Design the mechanisms to test the UAM with the sliding battery and with the manipulator on the sliding battery to see their real behaviours;

- Improve the dual arm method considering the manipulator dynamics in the stabilisation of the UAV. Then, could be augmented with more DoF to stabilise the UAV reducing disturbances of the main manipulator while helping with the picking and manipulation of a payload of different shapes and weights;

- Develop kinematic resolution at the acceleration level for the proposed methods maybe lead to more accurate solutions;

- Test the methods here developed in a simulation with external disturbances as simulated random gusts of wind or the aerodynamic wall effect on the UAV;

6. Bibliography

[1] Vijay Kumar and Nathan Michael. Opportunities and challenges with autonomous micro aerial vehicles. *The International Journal of Robotics Research*, 31(11):1279–1291, 2012.

[2] Evan Ackerman and Eliza Strickland. Medical delivery drones take flight in east africa. *IEEE Spectrum*, 55(1):34–35, 2018.

[3] Matěj Karásek. *Robotic hummingbird: Design of a control mechanism for a hovering flapping wing micro air vehicle.* PhD thesis, 11 2014.

[4] Guido Gioioso, Mostafa Mohammadi, Antonio Franchi, and Domenico Prattichizzo. A force-based bilateral teleoperation framework for aerial robots in contact with the environment. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 318–324. IEEE, 2015.

[5] Malcolm Thomas Connolly. The use of multi rotor remotely operated aerial vehicles roavs as a method of close visually inspecting cvi live and difficult to access assets on offshore platforms. In *Abu Dhabi International Petroleum Exhibition and Conference*. OnePetro, 2014.

[6] Hossein Bonyan Khamseh, Farrokh Janabi-Sharifi, and Abdelkader Abdessameud. Aerial manipulation—a literature survey. *Robotics and Autonomous Systems*, 107:221–235, 2018.

[7] Anibal Ollero, Marco Tognon, Alejandro Suarez, Dongjun Lee, and Antonio Franchi. Past, present, and future of aerial robotic manipulators. *IEEE Transactions on Robotics*, 38(1):626–645, 2021.

[8] DING Xilun, GUO Pin, XU Kun, and YU Yushu. A review of aerial manipulation of small-scale rotorcraft unmanned robotic systems. *Chinese Journal of Aeronautics*, 32(1):200–214, 2019.

[9] Fabio Ruggiero, Vincenzo Lippiello, and Anibal Ollero. Aerial manipulation: A literature review. *IEEE Robotics and Automation Letters*, 3(3):1957–1964, 2018.

[10] Alejandro Suarez, Manuel Perez, Guillermo Heredia, and Anibal Ollero. Cartesian aerial manipulator with compliant arm. *Applied Sciences*, 11(3):1001, 2021.

[11] Nursultan Imanberdiyev, Sumil Sood, Dogan Kircali, and Erdal Kayacan. Design, development and experimental validation of a lightweight dual-arm aerial manipulator with a cog balancing mechanism. *Mechatronics*, 82:102719, 2022.

[12] Alberto Pasetto, Yash Vyas, and Silvio Cocuzza. Zero reaction torque trajectory tracking of an aerial manipulator through extended generalized jacobian. *Applied Sciences*, 12(23):12254, 2022.

[13] E Rossetto. Interazione dinamica tra moto del manipolatore e dell'uav nella manipolazione aerea, a.a. 2019-2020.

[14] Alberto Pasetto. Minimizzazione dei disturbi dinamica trasmessi ad un uav durante operazioni di manipolazione robotica aerea, a.a. 2021-2022.

[15] Silvio Cocuzza, Isacco Pretto, and Stefano Debei. Least-squares-based reaction control of space manipulators. *Journal of Guidance, Control, and Dynamics*, 35(3):976–986, 2012.

[16] Rohit Chaurasia and Vandana Mohindru. Unmanned aerial vehicle (uav): A comprehensive survey. *Unmanned Aerial Vehicles for Internet of Things (IoT) Concepts, Techniques, and Applications*, pages 1–27, 2021.

[17] Christopher M Korpela, Todd W Danko, and Paul Y Oh. Designing a system for mobile manipulation from an unmanned aerial vehicle. In *2011 IEEE Conference on Technologies for Practical Robot Applications*, pages 109–114. IEEE, 2011.

[18] DJI S1000 drone website. https://www-v1.dji.com/spreading-wings-s1000.html. Accessed: 05-04-2023.

[19] Silvio Cocuzza, Edoardo Rossetto, and Alberto Doria. Dynamic interaction between robot and uav in aerial manipulation. In *2020 19th International Conference on Mechatronics-Mechatronika (ME)*, pages 1–6. IEEE, 2020.

[20] Daniel Mellinger, Quentin Lindsey, Michael Shomin, and Vijay Kumar. Design, modeling, estimation and control for aerial grasping and manipulation. In *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2668–2673. IEEE, 2011.

[21] Fabio Ruggiero, Miguel Angel Trujillo, Raul Cano, H Ascorbe, Antidio Viguria, C Peréz, Vincenzo Lippiello, Aníbal Ollero, and Bruno Siciliano. A multilayer control for multirotor uavs equipped with a servo robot arm. In *2015 IEEE international conference on robotics and automation (ICRA)*, pages 4014–4020. IEEE, 2015.

[22] Tan Chang and Hu Yu. Improving electric powered uavs' endurance by incorporating battery dumping concept. *Procedia Engineering*, 99, 12 2015.

[23] Yoji Umetani, Kazuya Yoshida, et al. Resolved motion rate control of space manipulators with generalized jacobian matrix. *IEEE Transactions on robotics and automation*, 5(3):303–314, 1989.

[24] C Menon, A Aboudan, S Cocuzza, A Bulgarelli, and F Angrilli. Free-flying robot tested on parabolic flights: kinematic control. *Journal of Guidance, Control, and Dynamics*, 28(4):623–630, 2005.

[25] Bruno Siciliano. Kinematic control of redundant robot manipulators: A tutorial. *Journal of intelligent and robotic systems*, 3:201–212, 1990.

[26] Yash Vyas, Alberto Pasetto, Victor Ayala-Alfaro, Nicola Massella, and Silvio Cocuzza. Null-space minimization of center of gravity displacement of a redundant aerial manipulator. *Robotics*, 12(2):31, 2023.

[27] Dragomir N Nenchev. Redundancy resolution through local optimization: A review. *Journal of robotic systems*, 6(6):769–798, 1989.

[28] Pedro Sanz. Robotics: Modeling, planning, and control (siciliano, b. et al; 2009)[on the shelf]. *IEEE Robotics & Automation Magazine*, 16(4):101–101, 2009.