



UNIVERSITY OF THE
WITWATERSRAND,
JOHANNESBURG

Control of Aerial Manipulators for Contact Inspection

James Matosse

(Student number: 1770322)

School of Mechanical, Industrial and Aeronautical Engineering

University of the Witwatersrand

Johannesburg, South Africa.

supervised by

Chioniso Kuchwa-Dube

A research **dissertation** submitted to the Faculty of Engineering and the Built Environment,
University of the Witwatersrand, in fulfilment of the requirements for the Master of Science in
Engineering.

25 October 2023

Declaration

I declare that this dissertation is my own, unaided work, except where otherwise acknowledged. It is being submitted for the degree of Degree of Master of Science in Engineering in the University of the Witwatersrand, Johannesburg. It has not been submitted before for any degree or examination at any other university.

Date: 25 October 2023

Signature

A handwritten signature in black ink, appearing to read 'James Matosse', written over a horizontal line.

Name and Surname (Student no.): James Matosse (1770322)

Abstract

Aerial manipulators are flying robotic systems that consist of a multi-rotor vehicle tailored with one or more manipulators. These systems are promising for contact inspection of chemical plants, boilers, tunnels and chimneys without human intervention. They require sophisticated control algorithms for control and stabilisation of complex varying dynamics due to the flying, manipulating and interacting with the environment. The dynamics of an aerial manipulator are treated as two subsystems, the quadrotor and the manipulator. The quadrotor dynamics are extracted through the Newton-Euler algorithm using quaternion-based orientation instead of Euler angles. The manipulator dynamics are extracted using the recursive Newton-Euler equations for purposes of handling the floating-base effect. A classic impedance control is a widely used effective control for contact force tracking but it requires the location of the environment relative to the end-effector and the environmental stiffness to be known precisely, and mostly the desired force as a constant. In contact inspection without human intervention, the environmental properties (geometrical and mechanical) are uncertain, causing the classic impedance control method to be less effective. Presented in this study are: an adaptive variable impedance control (AVIC) for force tracking combined with the integral linear quadratic regulator (LQR) for the quadrotor and proportional-derivative (PD) for the manipulator's navigation control. AVIC has the capability to counteract for environmental uncertainties by varying the impedance parameter (damping coefficient) on-line corresponding to the force tracking error. This control has been utilised effectively in industrial robot manipulators for a stable force tracking performance. So far in the literature, AVIC together with integral LQR and PD controllers have not been applied to aerial manipulators as a control strategy. Since the aerial manipulator is a coupled system, a linearisation process was performed on the quadrotor dynamics coupled with the manipulator to achieve a fair approximation of the non-linear model. This study has improved force tracking abilities through the implementation of AVIC that allows the aerial manipulator to achieve a desired contact force on the inspected contour while the quadrotor remains in a stable position. Finally, the developed system was simulated using MATLAB/SIMULINK program and the attained results proved the effectiveness of the strategy in comparison to the classic impedance control.

Contents

1	Introduction	1
1.1	Motivation	1
1.2	Literature Review	4
1.2.1	Control algorithms for contact inspection	4
1.2.2	Control algorithms for path tracking	5
1.2.3	Manipulator controls	8
1.2.4	Aerial Manipulator	9
1.3	Problem statement	12
1.4	Aim and objectives of dissertation	12
1.5	Organisation of Dissertation	13
2	System Kinematics	14
2.1	Forward Kinematics	14
2.2	Chapter summary	17
3	Dynamic Model	18
3.1	Quadrotor Dynamics	18
3.2	Manipulator Dynamics	25
3.3	Dynamics of the Aerial Manipulator	27
3.4	Chapter summary	29
4	Controller Design	30
4.1	Quadrotor Control	31
4.1.1	Controller implementation	38
4.2	Position control	42
4.3	The Environment model and Contact Forces	44
4.4	Impedance control of the Manipulator	45
4.4.1	The contact force tracking based on AVIC strategy	47
4.5	Chapter summary	49
5	Simulation Results and Discussion	50
5.1	Simulation Studies	51
5.2	Manipulator Results	55

5.3	Quadrotor results	66
5.4	Chapter summary	79
6	Conclusion and Future Work	80
6.1	Conclusion	80
6.2	Future Work	81
A	Detailed derivation of the Manipulator dynamics	87
A.1	Forward Recursion: Link 1	89
A.2	Forward Recursion: Link 2	89
A.3	Backward Recursion: Link 2	90
A.4	Backward Recursion: Link 1	91
B	Stability and convergence of the AVIC strategy	94

List of Figures

1.1	Sample of aerial manipulators	2
1.2	Block diagram for behavioural motion controller	6
2.1	Aerial manipulator with a two-link manipulator	16
3.1	Quadrotor sketch with labelled axes, rotational motions and motors	20
3.2	Quadrotor validation: Thrust against Zero Inputs	23
3.3	Quadrotor validation: Thrust and non-zero input torques	24
3.4	Aerial manipulator validation: Motion for zero control inputs	29
4.1	The overall control architecture for aerial manipulator	30
4.2	Quadrotor validation: Testing the quadrotor linear model	35
4.3	Aerial manipulator validation: Zero control inputs to the quadrotor	36
4.4	Block diagram of the LQR controller	39
4.5	Quadrotor's block diagram for the integral LQR controller	41
4.6	Simulink block diagram for the manipulator PD control	43
4.7	Orientation of contact normal force	45
4.8	The adaptive variable impedance control architecture	46
5.1	Quadrotor desired position - Scenario 3	54
5.2	Manipulator's end-effector desired position - Scenario 3	54
5.3	End-effector position tracking without physical interaction - Scenario 1	55
5.4	Manipulator joint angle tracking errors - Scenario 1	56
5.5	Manipulator joint inputs from computed torques - Scenario 1	56
5.6	End-effector position tracking during contact inspection - Scenario 2	57
5.7	Manipulator joint angle tracking errors - Scenario 2	58
5.8	Contact force tracking between AVIC and classic strategy - Scenario 2	58
5.9	Contact force tracking errors - Scenario 2	59
5.10	Manipulator joint inputs and contact torques - Scenario 2	60
5.11	End-effector position tracking - Scenario 3	60
5.12	Manipulator joint angle tracking errors - Scenario 3	61
5.13	Contact force tracking between AVIC and classic strategy - Scenario 3	61
5.14	Contact force tracking errors - Scenario 3	62
5.15	Manipulator joint inputs and contact torques - Scenario 3	63

5.16	Contact force tracking between AVIC and classic strategy - Scenario 4	64
5.17	Contact force tracking errors - Scenario 4	64
5.18	Manipulator joint angle tracking errors - Scenario 4	65
5.19	Manipulator joint inputs and contact torques - Scenario 4	65
5.20	Quadrotor trajectory tracking without physical interaction - Scenario 1	67
5.21	Quadrotor position tracking errors - Scenario 1	67
5.22	Quadrotor's angular velocities - Scenario 1	68
5.23	Quadrotor's control inputs from the integral LQR - Scenario 1	69
5.24	Quadrotor trajectory tracking during physical interaction - Scenario 2	70
5.25	Quadrotor position tracking errors - Scenario 2	70
5.26	Quadrotor's angular velocities - Scenario 2	71
5.27	Quadrotor's control inputs from the integral LQR - Scenario 2	71
5.28	Comparing the quadrotor's resultant Euler angles between AVIC and classic strategy - Scenario 2	72
5.29	Quadrotor trajectory tracking during physical interaction - Scenario 3	73
5.30	Quadrotor position tracking errors - Scenario 3	73
5.31	Quadrotor's translational velocities - Scenario 3	74
5.32	Quadrotor's angular velocities - Scenario 3	75
5.33	Quadrotor's control inputs from the integral LQR - Scenario 3	75
5.34	Comparing the quadrotor's resultant Euler angles between AVIC and classic strategy - Scenario 3	76
5.35	Quadrotor's angular velocities - Scenario 4	77
5.36	Quadrotor's control inputs from the integral LQR - Scenario 4	77
5.37	Comparing the quadrotor's resultant Euler angles between AVIC and classic strategy - Scenario 4	78
A.1	Two-link revolute joint manipulator	88

List of Tables

5.1	System parameters	51
5.2	Way points used for quadrotor’s desired trajectory generation using cubic polynomials	52
5.3	Manipulator Control Parameters	55
5.4	Quadrotor Control Parameters	66

Nomenclature

α_i	The manipulator's angular acceleration of link i .
$\bar{\omega}$	Angular velocity of the quadrotor expressed in the world-fixed frame $\{W\}$.
$\bar{q}$	The quaternion representing the quadrotor's orientation in the world-fixed frame $\{W\}$.
$\ddot{\vec{r}}$	The acceleration vector of the quadrotor expressed in the world-fixed frame $\{W\}$.
$\dot{\vec{r}}$	Velocity of body frame $\{B\}$ with respect to the world-fixed frame $\{W\}$.
$\dot{\vec{r}}_{EB}^B$	Velocity vector for frame $\{E\}$ with respect to body frame $\{B\}$ expressed in body frame $\{B\}$.
$\dot{p}_e$	Linear velocity of frame $\{E\}$ in the world-fixed frame $\{W\}$.
$\dot{q}$	The (2×1) Vector of the manipulator joint velocities.
$\dot{x}, \dot{y}, \dot{z}$	The quadrotor's velocity along X_W , Y_W , and Z_W , respectively.
η_B	The torque vector exerted on the quadrotor due to interconnection with the manipulator.
$\{0\}$	The manipulator base-fixed frame.
$\{B\}$	Quadrotor body reference frame.
$\{E\}$	Frame attached to the end-effector.
$\{i\}$	The frame rigidly attached to link i of the manipulator.
$\{W\}$	World-fixed reference frame.
ω_e	Angular velocity of frame $\{E\}$.
Ω_i	The angular velocity of each propeller.
ω_i	The manipulator's angular velocity of link i .
ω_x	The angular velocity around the X_B axis.

ω_y	The angular velocity around the Y_B axis.
ω_z	The angular velocity around the Z_B axis.
ω_{EB}^B	Angular velocity of frame $\{E\}$ relative to body frame $\{B\}$ and expressed in $\{B\}$.
Φ_b	The triple XYZ roll-pitch-yaw angles.
τ	The joint torque vector for the manipulator.
τ_i	The torque acting on link i .
τ_{ext}	The torque vector due to environment interaction.
$\vec{\Gamma}$	The torque input vector representing roll, pitch, and yaw torque.
$\vec{\omega}$	The angular velocity vector of the quadrotor expressed in the body frame $\{B\}$.
$\vec{r}$	Position of body frame with respect to $\{W\}$.
$\vec{r}_{EB}^B$	Position of $\{E\}$ with respect to $\{B\}$ expressed in $\{B\}$.
$\vec{U}$	The vector inputs to the quadrotor dynamics.
$\vec{X}$	The vector states for the quadrotor.
$a_{c,i}$	Acceleration of the center of mass (COM) for link i .
$a_{e,i}$	Acceleration of the end of link i .
b_k	The propeller's drag coefficient.
b_t	The propeller's thrust coefficient.
$C(q, \dot{q})$	The Coriolis matrix of the manipulator.
$D(q)$	The inertia matrix of the manipulator.
f_i	The force applied by link $i - 1$ to link i .
f_W	The force vector exerted on the quadrotor due to interconnection with the manipulator.
g	The gravitational acceleration.
$g(q)$	The gravity vector of the manipulator dynamics.
I_i	The manipulator's moment of inertia for link i .
J	The quadrotor's rotational inertia matrix (assumed diagonal) with respect to the body frame $\{B\}$.

$J(q)$	The manipulator Jacobian.
J_{xx}	The quadrotor's rotational inertia about the X_B axis.
J_{yy}	The quadrotor's rotational inertia about the Y_B axis.
J_{zz}	The quadrotor's rotational inertia about the Z_B axis.
l	The length of the quadrotor's arm.
l_i	The length of link i .
M	The mass of the quadrotor.
m_i	The mass of link i .
M_t	Total mass of the aerial manipulator.
M_u	The conversion matrix between squared motor speeds and resulting outputs of thrust and torques.
p_e	The position of $\{E\}$ with respect to $\{W\}$.
q	A (2×1) Vector of joint angles of the manipulator.
q_0, q_i, q_j, q_k	The quaternion elements representing the quadrotor's orientation in the world-fixed frame $\{W\}$.
R_B^E	Rotation matrix from end-effector frame $\{E\}$ to body frame $\{B\}$.
R_E	Rotation matrix from end-effector frame $\{E\}$ to world-fixed frame $\{W\}$.
R_i^{i+1}	The rotation matrix for rotating an object from frame $\{i+1\}$ to frame $\{i\}$.
R_W^B	Rotation matrix for rotating an object from body frame $\{B\}$ to world-fixed frame $\{W\}$.
$r_{i+1,ci}$	The displacement from COM of link i to frame $\{i\}$.
$r_{i,ci}$	The displacement from frame $\{i-1\}$ to COM of link i .
$r_{i,i+1}$	The displacement from frame $\{i-1\}$ to frame $\{i\}$.
T_B	Rotation matrix that transforms Euler rates to angular velocities expressed in the body frame $\{B\}$.
U_1	The net thrust generated by the propellers.
U_2	The net torque generated by the propellers around X_B .
U_3	The net torque generated by the propellers around Y_B .

U_4	The net torque generated by the propellers around Z_B .
v_b	The generalised velocity of the quadrotor.
v_e	The generalised end-effector velocity.
v_{EB}^B	A (6×1) Vector of the generalised velocity of the end-effector with respect to body frame $\{B\}$.
x, y, z	The quadrotor's position along X_W , Y_W , and Z_W , respectively.

Chapter 1

Introduction

In the recent years, researchers from the field of aerial robotics have grown interest for aerial manipulation, investigating control algorithms for vertical take-off and landing of unmanned aerial vehicles (UAVs) that are equipped with a gripper or a robotic arm. A UAV with a manipulator or multiple robotic arms is an aerial manipulator with similar features as shown in Figure 1.1. The manipulator (serial or delta type) allows the vehicle to pick-up and transport objects to a desired location, perform maintenance and/or repair infrastructure, or even for water quality sampling [24]. These tasks are can be performed in places where access is restricted for crew personnel and/or ground robots.

1.1 Motivation

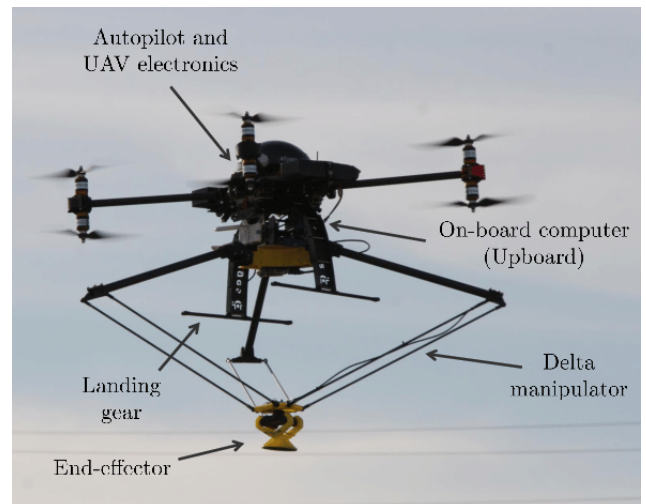
Traditional UAVs are used for surveillance and remote monitoring in rescue missions and dangerous environments. Inspection and maintenance of dangerous infrastructures such as tunnels, bridges, industrial plants, power lines, including communication towers can be expensive and dangerous for qualified inspectors or any crew member, however, it can instead be performed by robots. The use of aerial manipulators enables the performance of inspection, maintenance, or assembly operations at any point in space, which presents an advantage over ground robot in areas hard-to-reach [8]. The goal is to develop a simulated prototype of an aerial manipulator that can perform a wide variety of autonomous tasks, mimicking and extending the capabilities of mobile ground robots in the manufacturing sector.



(a) Quadrotor with bottom-attached serial manipulator [1]



(b) UAV with top-attached serial manipulator [3]



(c) UAV with bottom-attached delta manipulator [18]

Figure 1.1: Sample of aerial manipulators

In general, multi-rotor vehicles are under actuated systems as they only have four independent control inputs available over six degrees-of-freedom (DOF) which are three translational, three rotational. This under-actuation can cause unexpected variations in the end-effector orientation but can be addressed by considering the system's dynamics in the trajectory planning phase [4]. Every vehicle has an attitude, altitude and position control, and a collision avoidance technique. Attitude control is for keeping the orientation of the vehicle at a preferred value while altitude control keeps its distance to the ground at a desired value. Rolling and/or pitching the vehicle performs the position control to keep it on the desired point (x, y) .

Aerial manipulators are promising for contact inspection of railway infrastructure, chemical plants, boilers and chimneys without human intervention. Contact inspections can be classified into three types as: periodic, special and damage inspections especially for railway infrastructures like bridges, tunnels, etc. The first two types of inspections are used to monitor the infrastructure's quality following international and/or national standard requirements. Whilst the damage inspections are typically performed after a disastrous event has occurred [4]. During contact inspection the aerial manipulator will make physical contact with the environment to detect construction faults and defects. A sensor tool can be mounted as an end-effector for data collection such as strain gauges, tilt meters, eddy currents, ultrasonic sensors and other various sensors for non-destructive testing (NDT) methods [4, 33].

Therefore, the aerial manipulator requires a sophisticated algorithm for control and stabilization of the complex varying dynamics due to the flying, manipulating and interacting with the environment. When the end-effector makes contact with a rigid surface, contact forces are experienced by the manipulator and the quadrotor due to coupling effects. These contact forces increase the complexity of the system dynamics and can be measured by mounting a force sensor on the quadrotor to track precise contact force [4]. A number of control philosophies have been proposed and investigated to address the complexity of aerial manipulators in contact inspection. The aerial manipulator can be modelled by considering the manipulator as an external perturbation exerted on the quadrotor or by developing entire kinematics of the whole system to compute the dynamics [8]. The contact forces can be modelled for simulations to consider their effects on the quadrotor stability during contact inspection.

1.2 Literature Review

Most common controls for aerial manipulators are back-stepping and Cartesian impedance control. The impedance control theory is often used in vehicle-environment iterations. The model based controller uses the exact dynamic modelling and the actual values of the system parameters [34]. A back-stepping approach is developed to perform trajectory tracking control, for validity of the tracking error, its storage function should satisfy the standard Lyapunov condition [31]. Other innovative controls referred to in this section were developed for aerial manipulators for purposes of automating manipulation systems.

1.2.1 Control algorithms for contact inspection

Jimenez-Cano et al., [3] obtained dynamic models of a multi-rotor (with eight rotors) and robotic arm of 5-DOF for contact inspection from the Euler-Lagrange formulation. This derivation of the Euler-Lagrange equations starts as the difference between kinetic and potential energies of the system with the robotic arm at the top of the multi-rotor as seen in Figure 1.1b. A control technique known as variable parameter integral back-stepping (VPIB) was used for attitude and global position stabilization of the vehicle considering the arm in operation and contact forces with the environment [3]. This is because the controller provides good dynamic response and its parameters vary accordingly with the manipulator's dynamics. The dynamic effects due to the physical interaction between the end-effector and the environment were described using a Hunt-Crossley interaction model [36]. The vehicle was simulated for forward flying and moving the arm comparing the VPIB controller to a standard proportional-integral-derivative (PID) controller. The prototype was tested with an ultrasound sensor head as the end-effector inspecting a structure (bridge). It was found that adding an extra force to maintain contact long enough decreases the vehicle pitch and roll oscillations instead of the aerial manipulator being unstable [3].

Lippiello and Ruggiero, [44] revised the Cartesian impedance control to realize a desired dynamical relationship between the platform motion and the exerted external forces on a 3-DOF robotic arm. The dynamic model of the platform was derived in a symbolic matrix form through the Euler-Lagrangian formalism. The quadrotor with a 3-DOF robotic arm was subjected to the contact forces and external disturbances and then simulated through the MATLAB/SIMULINK environment. A non-linear robust adaptive Cartesian impedance control was also proposed for a similar platform by Sharifi and Sayyaadi, [34] and the control proved to adjust the end-effector impedance and guaranteed stability during the environmental (wall) interaction. The proposed controller utilises the impedance control theory, adaptive control theory for robustness to parametric uncertainties, sliding-mode control, and adaptation law in joint space for reducing calculation complexities [34]. The simulated scenario models a 'peg-in-hole' task environmental contact and the force sensor measurement becomes non-zero when the end-effector position exceeds a certain value in one dimension, meaning the environmental (wall) contact

begins. But the controller was not developed for tracking a desired force, rather, it regulates the measured force and maintains the resulted position and orientation thereafter.

Nava et al., [13] proposed a quadratic programming (QP) optimization as force control technique for aerial manipulators in physical interaction with the environment. The strategy can perform different manoeuvres without violating physical limitation like contact maintenance, and minimum propeller speeds. The experiments used an aerial vehicle that had six coplanar-center propellers that guarantees full local actuation of the vehicle's dynamics. The effectiveness was demonstrated through three different tasks, one of them was exploiting contacts for high acceleration motions. The control strategy employs the force-torque (FT) sensor to attain direct force feedback .

A control architecture for an aerial manipulator which integrated the delta manipulator and a quadcopter was proposed by Chermprayong et al., [37] for application of high precision tasks in aerial repair experiments outdoor. The system was demonstrated with a repair mission of holes and cracks patching on a the oil pipe using automated depositing technique for two-components of LD40 polyurethane foam on the damaged section. This aerial system has the actuators of the manipulator directly mounted on the quadcopter frame which reduces the disturbance the quadcopter in comparison with the serial manipulator aerial systems. Another control is a hybrid approach system which allows the vehicle to physically interact with the environment that was presented by Alexis et al., [20]. The UAV's dynamics changed drastically during physical contact, the hybrid automata was used to capture. The designed hybrid model had a predictive control technique that considered the system dynamics and self-imposed constraints, which resulted in a stable system during transitions and with optimal responses.

1.2.2 Control algorithms for path tracking

Kobilarov [31] proposed a non-linear trajectory control of an aerial manipulator by employing free flying dynamics incorporating the back-stepping control. The approach provides an alternative coordinate-free geometric algorithm that eliminate discontinuities. It could serves as a guideline for utilizing tasks that must track the system's center of mass and simultaneous track the and the end-effector position, without considering the uncertainties and control input limitations. Chee and Zhong, [23] proposed an implementation of a non-linear complementary filter and (PI) rate controller for attitude estimation and stabilization. Collision avoidance technique and altitude strategy required four infra-red sensors and an ultrasonic sensor, respectively. It was recommended to add sensors like a laser range finder to map terrains effectively improve the control's capability of the aerial manipulator.

Behavioural control was also proposed to guarantee that the aerial manipulator for coordination and management of complex tasks with different behaviours encompassed in a direct and beneficial way [21]. This control makes use of elementary behaviours, compound behaviours

(combination of elementary behaviours with a priority sequence) and the supervisor mode (responsible for changing between behaviours) based on the aerial manipulator and the state of the surrounding area (environment). Figure 1.2 shows four main modules that the motion controller had. It uses a PID controller and a mechanism to counterweight the arm movement [21]. The different modules handled different tasks; sensors information was devised and executed within the estimator module.

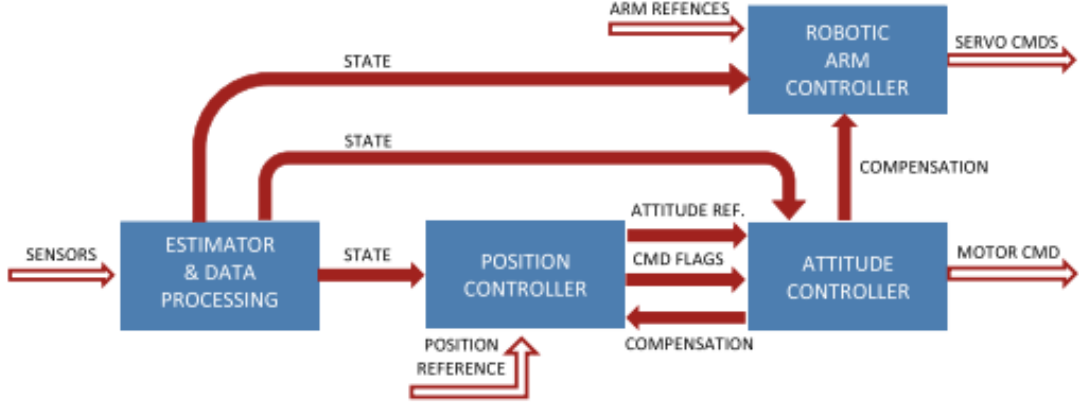


Figure 1.2: Block diagram for behavioural motion controller [21]

In the attitude control, there were two limitations; first, the mechanical limits of the displacement compensation system that restricted the movement and second, the mechanical compensation was not quick to yield desirable performance due to physical and software limitations [21]. To resolve these, the use of brush-less motors was recommended for quicker response and added an appropriate fixed value to the control to compensate the effect of the manipulators weight on the vehicle [21].

Islam and Okasha, [30] provide a comparative study between the linear quadratic regulator (LQR), model predictive control (MPC) and proportional-derivative (PD) controllers on the quaternion orientation quadrotor rather than the Euler angles orientation. The study compared the strategies and recommended appropriate environment for each strategy's capability. The LQR was defined as an optimal control method applicable to linear systems that allow the tuning of proportional-gains to minimise a performance function [30]. The LQR was demonstrated to offer quick responses and efficient tracking for a disturbance free system and was recommended for indoor applications. This is because when the disturbance was introduced, the LQR and PD showed lower tracking efficiency compared to the MPC [30]. The MPC was found to be slightly more efficient than the LQR with respect to control effort during the disturbance. Constraints handling at the control input once more, the MPC required lower control effort and rendered lower fluctuations in comparison to the LQR and PD strategies.

A differential flatness-quaternion technique was proposed by Choutri et al., [22] to enable the quadrotor to track a reference optimal path while avoiding any discontinuities that may occur. The LQR controller guaranteed the trajectory tracking through a double loop control structure for position regulating and the attitude strategy. The position controller outputs the position values which are used to compute the desired quaternion via the quaternion error for the attitude strategy [22].

Reyes-Valeria et al., [10] presented a quadrotor dynamics under the Newton-Euler algorithm based on quaternions for attitude and the corresponding control used the LQR with gain scheduling technique. Here, the LQR was used to obtain two gain matrices, for when the quadrotor was far from the reference trajectory and when the quadrotor was already tracking the reference trajectory. This is because higher gains are obtained for approaching the reference trajectory in one dimension but also results in an oscillatory error in another dimension [10]. Hence the use of gain scheduling technique to switch between the two gains depending on the set tracking error limit. The proposed control was tested through simulation with real parameters and demonstrated that unit quaternions can effectively substitute Euler angles for attitude control [10].

Gain scheduling is further analysed by Okasha et al., [32] to improve hindrances when using the LQR controller by investigating saturation limits of actuators, altering operating points continuously for complex trajectories and possible control input's chattering. The quadrotor dynamics were extracted through the Newton-Euler algorithm based on Euler angles representation of the attitude. The quadrotor model considered the aerodynamics drag and moments, and the gyroscopic rotor dynamics [32] to fairly model a real application. This model was linearised 12 times for different values of the heading angle to form 12 operating points for the gain-scheduled control and a continuous gain control. The former was demonstrated to have discontinuities and scattering in the control inputs during the transitions operating point to the next. The latter control evaluates the 12 gains and executing them as a continuous control function proven to eliminates the scattering and discontinuity problems in the control inputs [32]. Tracking steady state errors were present in all the simulated cases. Reduction of these errors can be achieved through proper selection of Q and R matrices, and an integral action can be incorporated in the LQR control to obtain zero steady state errors [32].

1.2.3 Manipulator controls

More than 15 years ago, Indrawanto et al., [15] presented a robust control (that adapts the impedance) for industrial robot manipulators to obtain fast and accurate force tracking for high productivity using the model reference adaptive control to attain a stable and converging tracking errors. A mechanical impedance is the regulation of relationship between position and force, and it is represented as second order impedance in Laplace domain as:

$$F(s) = (Ms^2 + Bs + K)X(s) \quad (1.1)$$

where $F(s)$ and $X(s)$ are force and position respectively. The impedance parameter K , B , and M being the desired stiffness, desired damping, and desired inertial of the manipulator respectively. From Equation (1.1), it can be seen that the desired impedance is linear and of second order. This is because second order systems are well understood and attaining measurements used for control loop pertaining to higher state variables is challenging for higher order systems [15]. The proposed controller improved force tracking capabilities, with oscillation problems due to derivative noisy force signal. The recommended solution was to implement a force estimator to predict the derivative force signal instead of using a simple derivative filter.

Duan et al., [9] proposed a force tracking strategy called adaptive variable impedance control (AVIC) for an industrial robot. This exploits the advantages between the adaptive control and the variable impedance laws. The proposed control performed dynamic force tracking in uncertain environment [9]. The control algorithm was also capable for dynamic desired force (cosine) tracking in addition to tracking the constant and linear desired forces. Their simulations did not consider any disturbances but an abrupt change of the environmental stiffness was considered three times within 10 s of the simulation. This was to affirm the robustness of the classic impedance control through uncertain environmental stiffness. The robot end-effector was modelled as a mass-damper system while the environment was considered to be rigid (only a spring system).

Lin et al., [45] also introduces the second order impedance control on a complaint flange system for force tracking, where MATLAB/SIMULINK platform was employed to validate the proposed control thereafter. The methodology that was used started with the establishment of the model, attaining the complaint system and verifying its stability. This was followed by understanding the correlation between the end-effector and the environment and the selection of control algorithm. Then, theoretical analysis and simulations were performed for the exploration of varying of impedance parameters within the system [45]. Finally, experiments were carried out using the adjusted parameters from simulation results and the obtained outcomes showed good response performance.

Yeol et al., [19] presented a tentacle system devised for mobile manipulation applicable for mini-AUVs and kinematic analysis for single DOF tentacle system. The operational space formulation was employed on the end-effector for a non-redundant manipulator, taking into account the gravitational forces, centrifugal and Coriolis forces of the end-effector [3]. A hyper redundant manipulator was designed for manipulation and considering manipulator effects as disturbances to the quadrotor with the hovering configuration [43]. Newton's law was employed for the dynamics of the quadrotor in a body-fixed coordinates. Inverse kinematics computes the appropriate joints of the manipulator corresponding to the identified positions such that the end-effector reaches the desired pose with less solving time.

1.2.4 Aerial Manipulator

Samadikhoshkho et al., [48] presented four different non-linear control strategies for aerial manipulators to analyse the efficiency and performance level in terms of robustness, accuracy and control effort for each strategy. The proposed semi-optimal strategy essentially consists of an outer loop based on LQR and inner loop based on feedback linearisation applied on a dynamic model derived from the Euler-Lagrangian formulation. Simulations on a quadrotor fitted with a 2-DOF manipulator were performed to illustrate the comparison in the with and without modelling uncertainties for complex manoeuvres. It was found that all strategies showed similar performance when restricting control effort without system uncertainties, but in terms of accuracy, the hierarchical strategy showed low performance because it presented substantial fluctuations in the roll and yaw motions [48]. In the presence of system uncertainties, the hierarchical strategy does not perform satisfactorily, although it was able to control the second joint of the manipulator better than the other three strategies. The sliding mode strategy handled the uncertainties better for control of the first joint of the manipulator. The results further showed that the sliding mode and the semi-optimal strategies provided good performance for the quadrotor control [48]. Further analysis demonstrated that the semi-optimal and sliding mode strategies outperformed the other strategies in accuracy and robustness, as well as the control inputs. Nonetheless, the other strategies also can be used in low levels of uncertainty to provide reasonable performance [48], especially for control of quadrotors.

A dual quaternion-based model was presented by Abaunza et al., [14] for a quadrotor with a 2-DOF robotic arm attached at the bottom of the vehicle. The dynamic model was extracted through the Newton-Euler algorithm. The use of dual quaternions resulted in a simplified calculations of the inverse kinematics since multiple transformations could be reworked by simply using only the quaternion multiplication and additions [14]. Simulation results showed that the end-effector was stable and converged to the desired values and a flight experiment was performed outdoor considering only for translational velocity in the X and Y axes. This was because it was challenging to perform outdoor, since substantial perturbations would affect the quadrotor behaviour [14]. The quadrotor was also claimed to compensate the effects from the manipulator.

Looking deeper in the literature, the aerial UAV’s mass distribution changes during manipulation tasks performed by the manipulator, which result in varying the dynamics of the system. For contact inspection, the manipulator’s end-effector has to be in contact with the environment long enough to take sufficient measurement readings. Recent work showed a possibility of the aerial manipulator being capable of this requirement using VPIB control technique. Further exploration can be investigated with employing the impedance control theory on the manipulator’s end-effector for projecting a desired impedance while interacting with the environment, which permits the varying of impedance parameters separately from the dynamics model [34].

Impedance control theory is mostly used to permit industrial robot manipulators to interact with the surroundings. Many robot applications such as de-burring and polishing, require contour following motions [15]. The manipulator is expected to apply a desired force normal to the contour, whose geometry and location are partially known, while moving along the contour. Currently, ground robots are well equipped to perform such tasks without human intervention [9],[15]. Hence, the recent increase on research for aerial manipulation of which most are fully-actuated multi-rotors UAVs [13],[3].

Now to the use of aerial manipulators for contact inspection, the literature is focused more on the robotic arm doing the inspection while the UAV is on-hover mode. The aerial manipulator flies forward toward a surface and hovers, allowing the robotic arm to do inspection through contact with the surface [29], [3], [44]. Depending on the robotic arm’s available DOF and length of each link, there could be certain limitations on the area to be inspected. In fact, the end-effector has to be off-contact after inspecting one point and allow the vehicle to stabilize, then in-contact again for the next point in the limited area. In this manner, some points would be missed for inspection of an environment such as bridge surface, and the process would take long to finish the job. The end-effector can be set do a continuous inspection on the limited area, while the vehicle tracks a given contour not just sliding or docking on the wall or surface. Yüksel et al., [7] in 2015 have shown such interaction with an elastic joint in aerial physical interaction, the experiments did not discuss or show that force tracking was achieved. The goal was to have “... a stable flight, while quadrotor equipped with the arm is sliding on the blue surface in a stable manner.” The set-up of this aerial manipulator was not considered as full system in terms of the coupling due to the dynamical effects of the quadrotor on the arm and vice versa, but considered these effects only as disturbances [7]. Again, Yüksel et al., [6] in 2019 presented physical interaction of a quadrotor fitted with a rigid tool of one DOF sliding on an uneven dome but this time measuring the contact forces using a FT sensor fitted on board of the quadrotor. The goal was to establish that reshaping system’s inertia can result in a better contact profile, while the quadrotor slides on the dome. This can be applicable for rapid surface inspection, cleaning or painting task.

In the literature, Alexis et al., [20] investigate interaction between aerial robots and their environments through contact for docking and sliding on walls. A pentagon airframe AIRobots CoaXial-rotor was developed for docking and can withstand impact forces. The proposed control strategy achieved optimal manoeuvring and minimised undesired behaviours. Stable docking and sliding on walls in a controlled manner while retaining contact were experimentally verified [20]. Fumagalli et al., [29] also presented an aerial manipulator with a custom-made manipulator devised remote inspection through contact interacting with an industrial plant. The passivity-based control maintained the end-effector in the desired positions (lateral and altitude), and it docked on a wall-like surface through the end-effector applying a certain force [29]. A gazebo environment was presented to model aerial robot, performing physical interaction with the surrounding virtual walls by Naldi et al., [39], using impedance controller on a ducted fan prototype. On these works, the aerial manipulator is either a linear actuator or a delta manipulator, but not the serial manipulator. Furthermore, the vehicle must tilt to wield the desired force to the wall due to the under-actuated nature of the system dynamics.

Many of the aerial robots mentioned above have the manipulation system that slide on a vertical surface for inspection, but not much for the robotic arm that would allow for contact inspection of a curvy surface such as boilers, bridges, etc. Yuksel et al., [6] presented a case that slides on an uneven surface for a quadrotor with 1-DOF rigid tool attached on top. The proposed control was considered as a low-level control, only responsible for regulating the aerial physical interaction but not developed for force or position trajectory tracking [6]. Nonetheless, the controller could be used together with a high-level (PD) controller for the quadrotor position tracking.

In this dissertation, the AVIC strategy used for industrial robot manipulator to achieve dynamic desired force tracking will be utilized for aerial manipulator. The AVIC for force tracking of the 2-DOF serial manipulator on top of the quadrotor, while implementing the integral LQR (quadrotor) and PD (manipulator) controllers for navigation control. The AVIC has a technique to compensate for environmental uncertainties by varying the impedance parameter (damping coefficient) on-line corresponding to the force tracking error. This strategy has been utilised effectively in industrial robot manipulators for stable force tracking performance, but it has not been applied to aerial manipulators. This strategy will improve contact inspection for dynamic force tracking in uncertain environment without using learning methods. It's a simpler adaptive impedance control to implement by adjusting one gain on-line through the force feedback. It was demonstrated through experiments that the AVIC achieved force tracking on a complex surface which consisted of a sine surface and a flat surface, which had different stiffness [9]. This application can mirror a contact inspection of an infrastructure surface that contains different stiffness like concrete, bricks and cement. The LQR controller has not been used frequently for aerial manipulators, but it has been utilised numerous times for quadrotors' navigation control without incorporating a manipulator. This may be due to

LQR limitations for stability in the presence of model uncertainties, but Reyes-Valeria et al., [10] recommended some measures to address steady state tracking error by implementing the integral action and a proper selection process of matrices Q and R . The LQR is an optimal control that is easy to implement for linear systems while respecting the control effort. Since the manipulator has 2-DOF, it can be used to compensate the end-effector position tracking when the LQR falls short to within 0.5 m of the desired quadrotor position. The dynamics of the moving aerial manipulator (quadrotor and manipulator during inspection) will be more complex and will require much consideration for the quadrotor's stability. The uncertainties on the environment, namely dynamic environmental location and/or unknown environmental stiffness, may introduce dynamical inputs to the control dynamics. Therefore, these unknown dynamical inputs will be considered to simulate a more practical situation.

1.3 Problem statement

Contact inspection using aerial manipulators is useful because they can be deployed to substitute or assist crew members in various applications that are considered very dangerous, in restricted areas, and/or very costly due to the necessity of highly trained specialists. In addition, they can avoid halting of work due to closers to the infrastructure being inspected resulting in less inconveniences and high productivity. Current contact inspection methods use ground robots, crew members with specialised trucks to perform the inspection with various NDT tools. Aerial manipulators have been proposed to do such contact inspection but the majority of the scenarios is for sensor installation that only requires the quadrotor to hover during contact and tracking constant forces. To track desired dynamic forces while the end-effector follows a contour in-contact with the environment introduces control difficulties for both the quadrotor and end-effector position, especially for serial manipulator as chosen for this dissertation. Therefore there is a need for improved dynamic force tracking in the presence of environmental uncertainties and the position controls should be able to compensate accordingly the resultant coupling forces.

1.4 Aim and objectives of dissertation

In the field of robotics, there are developments to fully automate the inspection system to improve quality of products or services in manufacturing, or inspection of dangerous infrastructures. This study aims to track a desired contact force through the utilization of the AVIC strategy for the aerial manipulator to follow a certain contour to be inspected. Can the integral LQR controller regulate the UAV during the manipulator's end-effector physical contact interaction with a rigid surface and following a certain contour that requires the UAV to be off-hover mode? The following can be listed as research objectives:

- To implement an adaptive variable impedance control for aerial manipulator (with the manipulator at the top) to obtain force tracking for contact inspection.

- To assess the AVIC's effectiveness in uncertain environments including for unknown stiffness and/or uncertainty to the environmental location during contact inspection.
- To evaluate the quadrotor's position tracking (including slope and cosine surfaces) with and without contact inspection.
- To compare results obtained from the proposed control approach and the classical impedance control.

The main contributions are the implementation of an AVIC together with the integral LQR and PD controllers for aerial manipulator to achieve better force tracking performance in comparison to the classical impedance control. And the demonstration of the quadrotor's position tracking capabilities using the integral LQR controller during contact inspection.

The scope of this dissertation is limited to modelling and simulations only, physical prototype is for future investigations. As such, simplifications were assumed during the modelling like neglecting aerodynamic drag and moments acting on the quadrotor, propeller's air density, and the gyroscopic effects. The manipulator has two links that are assumed to be identical and it's base is presumed to be collinear with the quadrotor's center of mass. The simulation scenarios were devised in such a way that the reference trajectories can be sent directly to the manipulator to perform certain movements while the quadrotor remains stable at desired position. Considered constant and dynamic contact forces during physical interaction with the environment and introduced environmental location uncertainty for two simulation scenarios.

1.5 Organisation of Dissertation

The dissertation is divided into six chapters and is organized as follows. Chapter 2 treats the quadrotor and the manipulator as individual systems to model their dynamics, then combines the two systems by treating the quadrotor as the floating-base for the manipulator.

The next chapter, Chapter 3 presents forward kinematics of the aerial manipulator, establishing reference frames and transformation matrices between the frames to obtain positions and velocities of the respective reference frames.

In Chapter 4, controller design is presented covering the linearisation of the unit quaternions for the quadrotor and its integral LQR controller and the AVIC strategy for force tracking between the manipulator's end-effector and the environmental surface.

Chapter 5 presents the different simulation scenarios and discussions regarding the results from the simulations. Finally, in Chapter 6 some conclusions are presented and future works on the aerial manipulator are proposed.

Chapter 2

System Kinematics

The aerial manipulator has to be described and modelled within a coordinate system and frame of reference to represent its position and orientation. In this chapter, the motion of the aerial manipulator will be explained geometrically rather than considering the forces and moments that induce the motion. The first section covers the forward kinematics that dictate the end-effector's position and orientation with respect to the given values of the joint variables of the aerial manipulator.

2.1 Forward Kinematics

Denote the body frame with its origin at the quadrotor centre-of-mass (COM) as frame $\{B\}$ depicted in Figure 2.1. And its position relative to the world-fixed or frame $\{W\}$ given as a (3×1) vector $\vec{r} = [x \ y \ z]^T$, while the quadrotor's orientation in the frame $\{W\}$ is described by a quaternion $\bar{q} = [q_0 \ q_i \ q_j \ q_k]^T$. Denoting the angular velocity vector $\vec{\omega} = [\omega_x \ \omega_y \ \omega_z]^T$ used to express the quadrotor's rotational speeds in frame $\{B\}$. The rotation matrix R_W^B in Equation (2.1) rotates an object from frame $\{B\}$ to frame $\{W\}$ [22], [46].

$$R_W^B = \begin{bmatrix} q_0^2 + q_i^2 - q_j^2 - q_k^2 & 2(q_i q_j - q_0 q_k) & 2(q_i q_k + q_0 q_j) \\ 2(q_i q_j + q_0 q_k) & q_0^2 - q_i^2 + q_j^2 - q_k^2 & 2(q_j q_k - q_0 q_i) \\ 2(q_i q_k - q_0 q_j) & 2(q_j q_k + q_0 q_i) & q_0^2 - q_i^2 - q_j^2 + q_k^2 \end{bmatrix} \quad (2.1)$$

This transformation is easily reversed by taking an inverse of the rotation matrix ($R_B^W = (R_W^B)^{-1}$). Additionally, the expression $(R_B^W)^{-1} = (R_W^B)^T$ is satisfied because the rotation matrix R_W^B is orthogonal [22]. Since two quaternion components are multiplied together each time, the quaternion $\bar{q}$ and its conjugate $-\bar{q}$ will both be perform the same rotation. Therefore, there can only be two possible real quaternion entry solutions. For any given rotation, the solutions are: the particular quaternion and its conjugate [46]. If the real (scalar) component q_0 is constrained to be ≥ 0 , then there is only one possible solution, except when $q_0 = 0$. In contrast, the quaternions have no discontinuity problems that are faced by the well-known Euler angles to perform similar transformation.

Unit quaternions are superior when performing theoretical derivations related to orientation but they are not very intuitive for choosing problematic orientations in comparison with Euler angles coordinate system. As such, the direction cosine matrix in Equation (2.2) can be utilised whenever necessary to use Euler angles representation [46], although it was not used in this dissertation.

$$R_W^B = \begin{bmatrix} C_\theta C_\psi & -C_\theta S_\psi & S_\theta \\ C_\phi S_\psi + S_\phi S_\theta C_\psi & C_\phi C_\psi - S_\phi S_\theta S_\psi & -S_\phi C_\theta \\ S_\phi S_\psi - C_\phi S_\theta C_\psi & S_\phi C_\psi + C_\phi S_\theta S_\psi & C_\phi C_\theta \end{bmatrix} \quad (2.2)$$

where C and S is short for $\cos(\cdot)$ and $\sin(\cdot)$ respectively.

$\phi \in \Re$ is the quadrotor's roll (X rotation) angle,

$\theta \in \Re$ is the quadrotor's pitch (Y rotation) angle, and

$\psi \in \Re$ is the quadrotor's yaw (Z rotation) angle.

The vector of Euler angles $\Phi_b = [\phi \ \theta \ \psi]^T$ is the roll-pitch-yaw angles for the quadrotor in the world frame $\{W\}$ [46].

Denoting frame $\{E\}$ as a frame welded to the manipulator's end-effector [11]. Hence, the position of $\{E\}$ with respect to $\{W\}$ is written as:

$$p_e = \vec{r} + R_W^B \vec{r}_{EB}^B \quad (2.3)$$

where $\vec{r}_{EB}^B$ is the vector that describes the position of frame $\{E\}$ with respect to body frame $\{B\}$ expressed in body frame $\{B\}$. The orientation of frame $\{E\}$ is defined by the following rotation matrix R_E in Equation (2.4).

$$R_E = R_W^B R_B^E \quad (2.4)$$

where R_B^E describes the rotation matrix of frame $\{E\}$ with respect to body frame $\{B\}$.

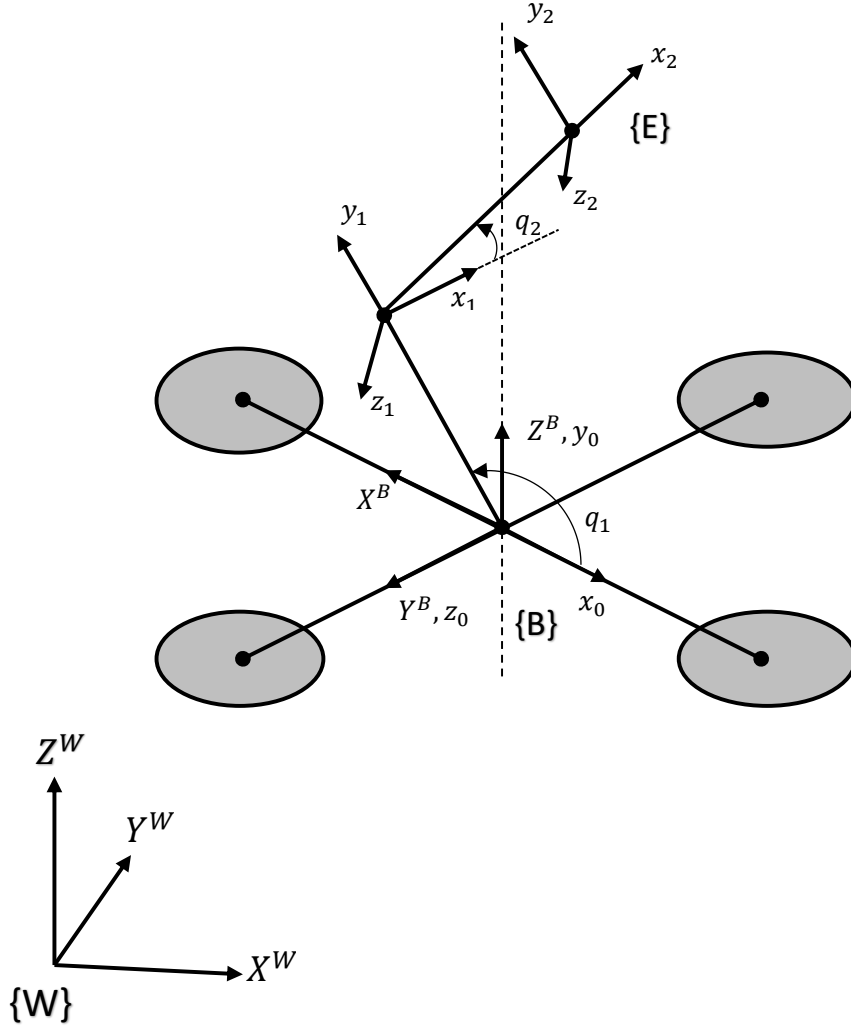


Figure 2.1: Aerial manipulator with a two-link manipulator, adapted from [46],[47]

The linear velocity $\dot{p}_e$ of frame $\{E\}$ in the world frame $\{W\}$ can be attained by taking time derivative of Equation (2.3) as:

$$\dot{p}_e = \dot{\vec{r}} - \text{Skew}(R_W^B \vec{r}_{EB}^B) \bar{\omega} + R_W^B \dot{\vec{r}}_{EB}^B \quad (2.5)$$

where $\text{Skew}(\cdot)$ is the (3×3) skew-symmetric matrix operator [35] and $\bar{\omega} = R_W^B \bar{\omega}$ denotes the angular velocity of the quadrotor in the world frame $\{W\}$. The angular velocity ω_e of frame $\{E\}$ is expressed as:

$$\omega_e = \bar{\omega} + R_W^B \omega_{EB}^B \quad (2.6)$$

where ω_{EB}^B denotes the angular velocity of frame $\{E\}$ relative to body frame $\{B\}$ and expressed in $\{B\}$.

Let $q = [q_1 \ q_2]^T$ denote the (2×1) vector of the manipulator joint angles [11]. Then denoting $v_{EB}^B = [\dot{\vec{r}}_{EB}^B \ \omega_{EB}^B]^T$ as a (6×1) vector for an augmented velocity of the end-effector with respect to body frame $\{B\}$. This velocity can be determined with the joint velocities $\dot{q}$ via the manipulator Jacobian $J(q)$ such that:

$$v_{EB}^B = J(q) \dot{q} \quad (2.7)$$

From Equation (2.5) and (2.6), then the augmented end-effector velocity $v_e = [\dot{p}_e \ \omega_e]^T$ could be written as:

$$v_e = J_B v_b + J_{EB} \dot{q} \quad (2.8)$$

where

$$v_b = \begin{bmatrix} \dot{\vec{r}} \\ \vec{\omega} \end{bmatrix}, J_B = \begin{bmatrix} \mathbf{I}_3 & -Skew(R_W^B \vec{r}_{EB}^B) \\ \mathbf{O}_3 & R_W^B \end{bmatrix}, J_{EB} = \begin{bmatrix} R_W^B & \mathbf{O}_3 \\ \mathbf{O}_3 & R_W^B \end{bmatrix}$$

where $\mathbf{I}_k$ and $\mathbf{O}_k$ denote $(k \times k)$ identity and $(k \times k)$ zeros matrices, respectively [28]. Whenever the attitude of the quadrotor has been expressed in Euler angles, then Equation (2.8) would be:

$$v_e = J_B Q_B X_b + J_{EB} \dot{q} \quad (2.9)$$

with

$$X_b = \begin{bmatrix} \vec{r} \\ \Phi_b \end{bmatrix}, Q_B = \begin{bmatrix} \mathbf{I}_3 & \mathbf{O}_3 \\ \mathbf{O}_3 & T_B \end{bmatrix}$$

where T_B ascribes the rotation matrix that determine the angular velocity $\vec{\omega}$ from Euler rates $\dot{\Phi}_b$ [46], and is can be expressed:

$$T_B = \begin{bmatrix} C_\theta C_\psi & -S_\psi & 0 \\ C_\theta S_\psi & C_\psi & 0 \\ -S_\theta & 0 & 1 \end{bmatrix} \quad (2.10)$$

Alternatively, $\dot{\Phi}_b$ can be found by multiplying $(T_B)^{-1}$ with a set of angular velocities $\vec{\omega}$. But there exist a singularity at $\theta = \pm \frac{\pi}{2}$ radians that is inevitable with Euler XYZ coordinate system.

2.2 Chapter summary

The system kinematics were presented, setting the frames of reference that help to describe and model the aerial manipulator. The rotation matrix R_W^B also known as directional cosine matrix was determined using the unit quaternion representation. Forward kinematics that relate the end-effector dynamics to the quadrotor's COM were given together with the transformation matrix T_B that determines the angular velocity vector $\vec{\omega}$ from the Euler rates $\dot{\Phi}_b$. The next chapter will discuss the modelling of the aerial manipulator.

Chapter 3

Dynamic Model

The complete model of the aerial manipulator is a coupled system between the quadrotor (floating-base) and the manipulator. But the quadrotor and the manipulator can be treated as individual systems to model their dynamics. The quadrotor dynamics are extracted through the Newton-Euler algorithm using quaternion-based orientation instead of Euler angles. The manipulator dynamics are extracted from the recursive Newton-Euler equations for purposes of handling the floating-base effect.

3.1 Quadrotor Dynamics

The development of the mathematical model for the quadrotor dynamics using the Newton-Euler algorithm follows the Thesis by Neeley [46]. First, the orientation representation was a quaternions-based instead of Euler angles based. A quaternion is a type of hyper-complex elements that can be used to define an axis in a 3-D space after-which, a single rotation can be performed around the defined axis. While Euler angles make use of three sequential rotations around non-consecutive pre-defined axes to represent orientation [46], unit quaternions do not have inherent singularities when expressing the quadrotors orientation [22]. Thus with appropriate restrictions, there is only one way to represent the orientation. During differentiation, the unit quaternion elements start to vanish or reduces to a constant, hence they are computationally cheaper. This is because they mostly depend on additions and multiplications and rarely on divisions [46]. On the other hand, trigonometric functions are necessary when working with Euler angles. Later in the section, the conversion from unit quaternions to Euler angles will be given because unit quaternions are a bit alien to work with in terms of choosing orientations.

Then, the position of the quadrotor can be represented in the world frame $\{W\}$ as shown in Figure 3.1, having an X-right, Y-forward and Z-up coordinate system. The quadrotor's translational position vector and velocity vector expressed in $\{W\}$ are:

$$\vec{r} = \begin{bmatrix} x \\ y \\ z \end{bmatrix} \quad (3.1)$$

$$\dot{\vec{r}} = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix} \quad (3.2)$$

Typically, the quaternion [46] to represent the quadrotor's orientation in the world frame $\{W\}$ can be expressed with four hyper-complex components as in Equation (3.3).

$$\bar{q} = \begin{bmatrix} q_0 \\ q_i \\ q_j \\ q_k \end{bmatrix} \quad (3.3)$$

The resulting quaternion norm can be calculated as:

$$||\bar{q}|| = \sqrt{q_0^2 + q_i^2 + q_j^2 + q_k^2} \quad (3.4)$$

A unit quaternion can be defined as a quaternion with a norm of one [46]. Equation (3.5) denotes a vector for angular velocities of the quadrotor in the body frame $\{B\}$.

$$\vec{\omega} = \begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} \quad (3.5)$$

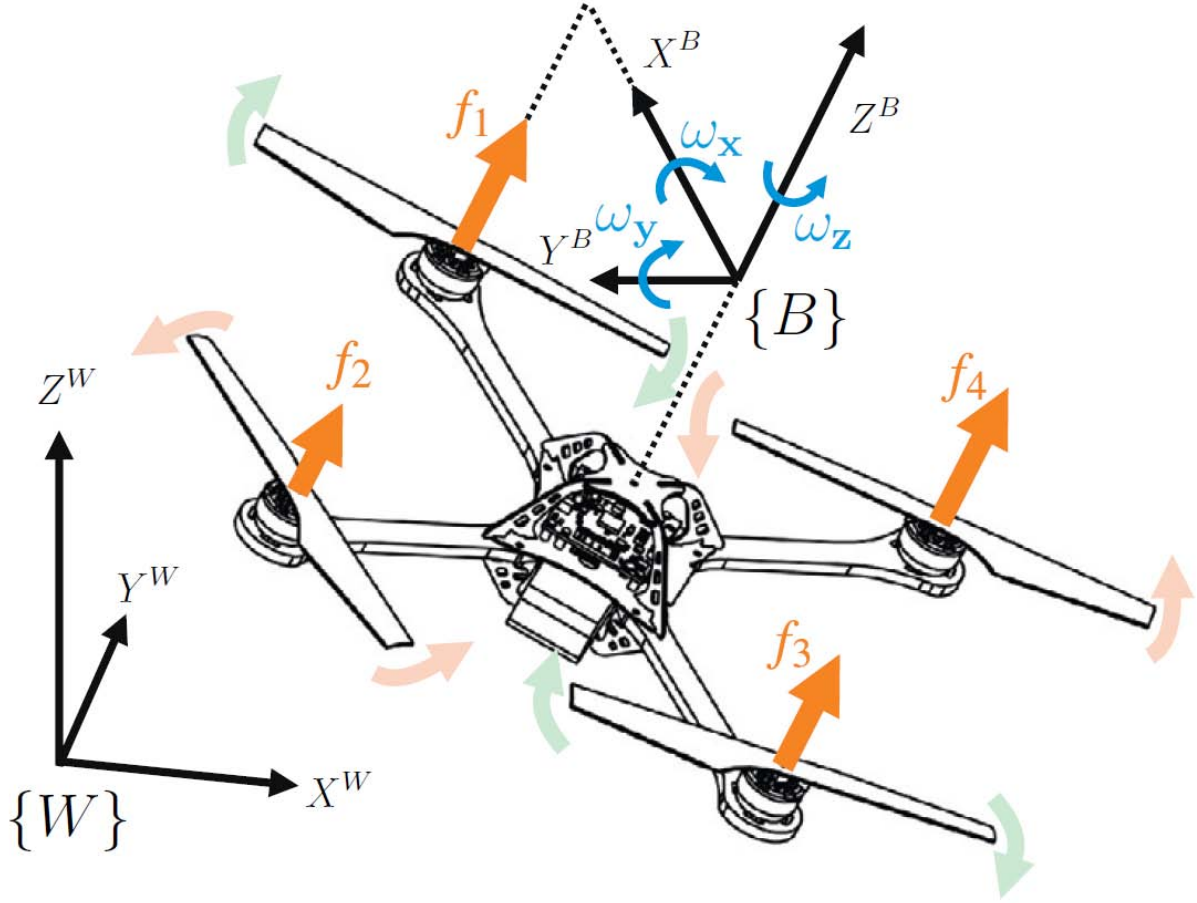


Figure 3.1: Quadrotor sketch with labelled axes, rotational motions and motors [46]

The quadrotor thrust and torques that are produced by a certain set of propeller speeds can be attained by the quadrotor parameters, namely propeller's thrust coefficient (b_t), propeller's drag coefficient (b_k), the length of the quadrotor's arm l and an angular velocity the propeller is spinning at (Ω_i). Equation (3.6) [46] is a result of combining the total thrust generated with roll U_2 , pitch U_3 and yaw U_4 torques generated by systematically changing the propeller speeds. In reality these should depend on air density.

$$\begin{aligned}
 \begin{bmatrix} U_1 \\ U_2 \\ U_3 \\ U_4 \end{bmatrix} &= \begin{bmatrix} b_t & b_t & b_t & b_t \\ 0 & b_t l & 0 & -b_t l \\ -b_t l & 0 & b_t l & 0 \\ b_k & -b_k & b_k & -b_k \end{bmatrix} \begin{bmatrix} \Omega_1^2 \\ \Omega_2^2 \\ \Omega_3^2 \\ \Omega_4^2 \end{bmatrix} \\
 &= M_u \begin{bmatrix} \Omega_1^2 \\ \Omega_2^2 \\ \Omega_3^2 \\ \Omega_4^2 \end{bmatrix} \tag{3.6}
 \end{aligned}$$

There are 13 states for the quadrotor depicted in Equation (3.7) but for unit quaternions, q_0 is restricted in terms of q_i , q_j , and q_k to fulfil the norm requirement. Thus there are 12 free states which will be used to linearise the quadrotor model.

$$\begin{aligned}\vec{X} &= [\vec{r}^T \quad \vec{q}^T \quad \dot{\vec{r}}^T \quad \vec{\omega}^T] \\ &= [x \quad y \quad z \quad q_0 \quad q_i \quad q_j \quad q_k \quad \dot{x} \quad \dot{y} \quad \dot{z} \quad \omega_x \quad \omega_y \quad \omega_z]^T\end{aligned}\quad (3.7)$$

Equation (3.8) gives the vector inputs to the quadrotor in terms of the net thrust and torques as determined from Equation (3.6).

$$\vec{U} = [U_1 \quad U_2 \quad U_3 \quad U_4]^T \quad (3.8)$$

Using the Newton-Euler algorithm requires treating the quadrotor as a rigid body under external forces applied to its COM [10]. Assuming that there are only two forces acting on the quadrotor namely, the gravity and the net thrust generated by the propellers, the translational dynamics can be determined. For simplicity during controller design and linearisation process, the drag affecting the quadrotor is presumed to be of inconsequential. The dynamic equation [10, 46] is represented by Equation (3.9).

$$\ddot{\vec{r}} = R_W^B \begin{bmatrix} 0 \\ 0 \\ U_1/M \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ -g \end{bmatrix} \quad (3.9)$$

When using the rotation matrix R_W^B given in Equation (2.2), the first part of the translational dynamics is reduced to be a factor of the third column of matrix R_W^B multiplied by U_1/M . For the rotational dynamics, the Euler's equation of motion [10, 46] for a system experiencing net torques is given in Equation (3.10). Once again, several effects are deemed to be negligible for simplicity, namely the gyroscopic effects resulting from spinning propellers, torques resulting from accelerating/decelerating propellers, and rotational drag.

$$J\dot{\vec{\omega}} = \vec{\Gamma} - \vec{\omega} \times J\vec{\omega} \quad (3.10)$$

where

$J \in \Re^{3 \times 3}$ is the quadrotor's inertia matrix with respect to body frame $\{B\}$,
 $\vec{\Gamma} = [U_2 \quad U_3 \quad U_4]^T$ is the roll, pitch, and yaw input torques.

Finally, the dynamics of the quadrotor [16, 22, 10, 46] become as follow:

$$\begin{aligned}
\ddot{x} &= \frac{2U_1}{M}(q_0q_j + q_iq_k) \\
\ddot{y} &= \frac{2U_1}{M}(q_jq_k - q_iq_0) \\
\ddot{z} &= \frac{U_1}{M}(q_0^2 - q_i^2 - q_j^2 + q_k^2) - g \\
\dot{q}_0 &= -\frac{1}{2}(q_i\omega_x + q_j\omega_y + q_k\omega_z) \\
\dot{q}_i &= \frac{1}{2}(q_0\omega_x + q_j\omega_z - q_k\omega_y) \\
\dot{q}_j &= \frac{1}{2}(q_0\omega_y - q_i\omega_z + q_k\omega_x) \\
\dot{q}_k &= \frac{1}{2}(q_0\omega_z + q_i\omega_y - q_j\omega_x) \\
\dot{\omega}_x &= (U_2 + \omega_y\omega_z(J_{yy} - J_{zz}))/J_{xx} \\
\dot{\omega}_y &= (U_3 + \omega_x\omega_z(J_{zz} - J_{xx}))/J_{yy} \\
\dot{\omega}_z &= (U_4 + \omega_x\omega_y(J_{xx} - J_{yy}))/J_{zz}
\end{aligned} \tag{3.11}$$

In Equation (3.12), unit quaternion is converted into Euler angles as discussed above for easy decision making towards choosing quadrotor's orientation, while Equation (3.13) shows the formula to transform Euler angles to a unit quaternion [46]. Alternatively, the Euler angles can be found by a single integration of the rates of Euler angle $\dot{\Phi}_b$ obtained from angular velocity $\vec{\omega}$ in the body frame $\{B\}$ by the inverse matrix of transformation T_B in Equation (2.10).

$$\begin{aligned}
\phi &= \arctan\left(\frac{2(q_0q_i - q_jq_k)}{1 - 2(q_i^2 + q_j^2)}\right) \\
\theta &= \arctan\left(\frac{-2(q_0q_j + q_iq_k)}{\sqrt{1 - (2(q_0q_j + q_iq_k))^2}}\right) \\
\psi &= \arctan\left(\frac{2(q_0q_k - q_iq_j)}{1 - 2(q_j^2 + q_k^2)}\right)
\end{aligned} \tag{3.12}$$

$$\begin{aligned}
q_0 &= -\sin(\phi/2) \sin(\theta/2) \sin(\psi/2) + \cos(\phi/2) \cos(\theta/2) \cos(\psi/2) \\
q_i &= +\sin(\phi/2) \cos(\theta/2) \cos(\psi/2) + \sin(\phi/2) \sin(\theta/2) \cos(\psi/2) \\
q_j &= -\sin(\phi/2) \sin(\theta/2) \cos(\psi/2) + \sin(\phi/2) \cos(\theta/2) \cos(\psi/2) \\
q_k &= +\sin(\phi/2) \sin(\theta/2) \cos(\psi/2) + \sin(\phi/2) \cos(\theta/2) \cos(\psi/2)
\end{aligned} \tag{3.13}$$

The model can be tested with some inputs for thrust, roll, pitch and yaw. The center of mass of the quadrotor is simulated to counteract the gravity effect using thrust $U_1 = 18\text{ N}$ and the plot can be seen in Figure 3.2 below. For zero inputs, the quadrotor model simulates a free-fall motion since only gravity is acting on the object. Applying the same thrust value for lift-off, and setting the roll input $U_2 = 0.002\text{ N} \cdot \text{m}$, the quadrotor take-off and lifts for about 2.5 s and then starts to fall. Similarly, the same behaviour is observed when the pitch input $U_3 = 0.002\text{ N} \cdot \text{m}$ is used as well. This is due to the nonlinearity of the quadrotor model, before the linearisation process, one can use the yaw input $U_4 = 0.01\text{ N} \cdot \text{m}$ together with the roll or pitch input to prevent the quadrotor from crushing to the ground. Although, this will introduce a linear angular velocity in the z- direction as well as coupling in the roll and pitch motions. This effect is depicted in Figure 3.3, the yaw motion can be control independently although the same thrust value was applied. It can be seen that the quadrotor COM is lifting vertically when the yaw input is applied. This represents the quadrotor rotating around the z-axis while gaining altitude.

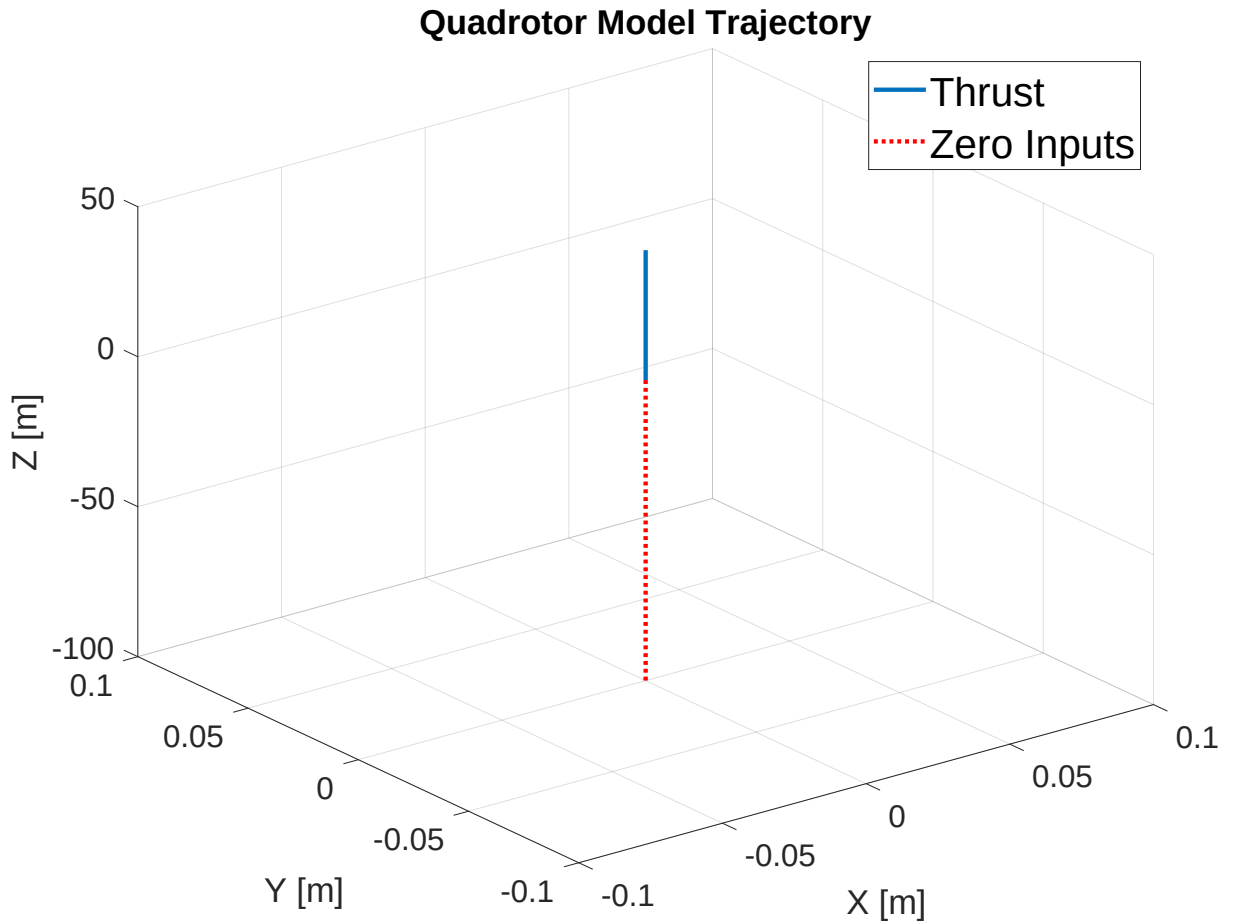


Figure 3.2: Quadrotor validation: Thrust against Zero Inputs

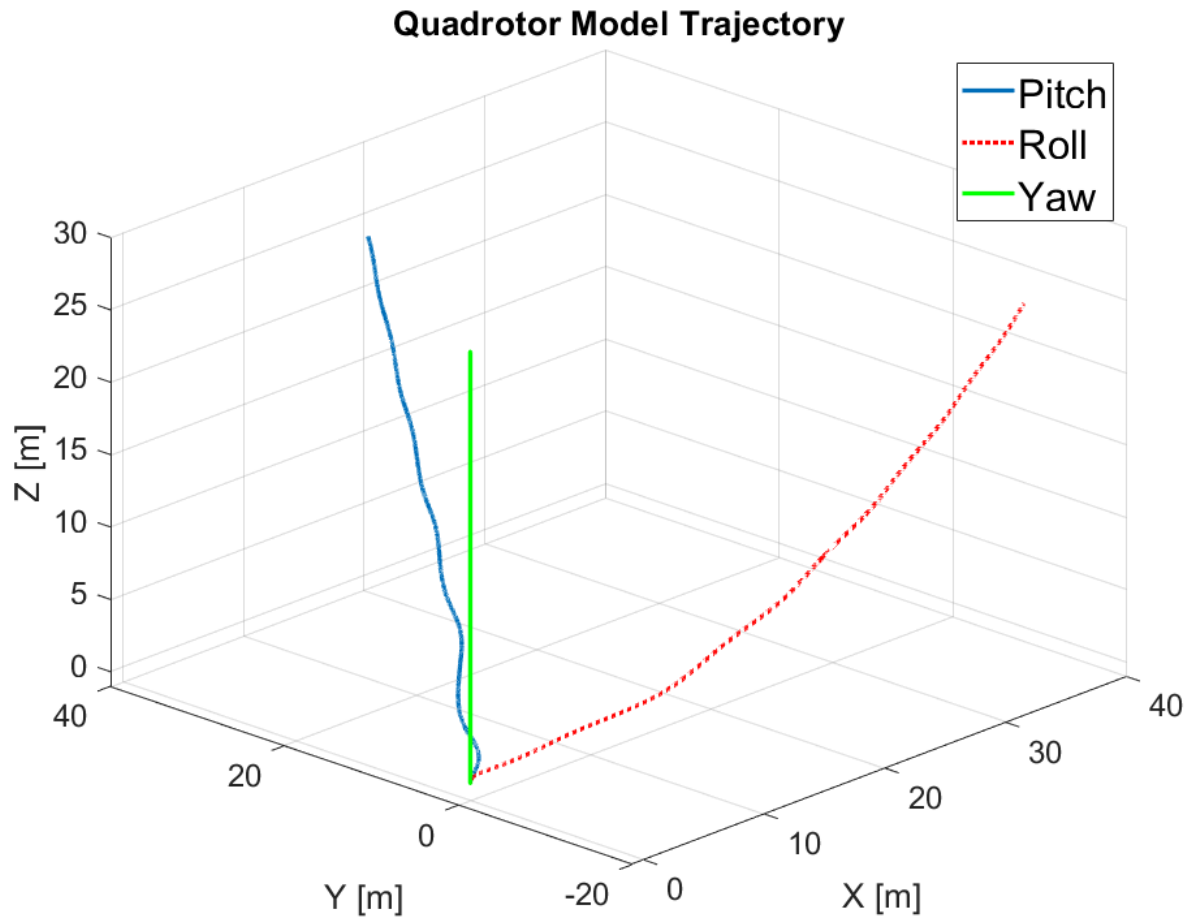


Figure 3.3: Quadrotor validation: Thrust and non-zero input torques

3.2 Manipulator Dynamics

The dynamic analysis of the manipulator can be performed using Lagrangian or Newton-Euler algorithm. In the Lagrangian algorithm, the manipulator is treated as whole (subsystem) using the Lagrangian equation which determines the difference between the total kinetic energy and the total potential energy of the system [35, 44]. The Newton-Euler algorithm however, treats each link of the manipulator subsystem and writing down the equations that describe the linear and rotational motions. These equations of motion for each link have coupling forces and torques that are also present in the equations of motion for the neighbouring links [35].

To derive the equations of motion for an n -link manipulator, frame $\{0\}$ was chosen as the base-fixed frame, and frame $\{i\}$ being rigidly attached to link i for $i \geq 1$ [35]. The goal is to ascertain the correlation between the manipulator joint torques $\tau(\cdot)$ and manipulator joint angles $q(\cdot)$. Commence with f_i being the force plied by link $i - 1$ to link i [35] and can be written as in Equation (3.14).

$$f_i = R_i^{i+1} f_{i+1} + m_i a_{c,i} - m_i g_i \quad (3.14)$$

All forces acting on each link can be solved by successively switching $i = n, n - 1, \dots, 1$ [35, 47]. Then, the torque τ_i plied on link i is given by Equation (3.15)

$$\tau_i = R_i^{i+1} \tau_{i+1} - f_i \times r_{i,ci} + (R_i^{i+1} f_{i+1}) \times r_{i+1,ci} + I_i \alpha_i + \omega_i \times (I_i \omega_i) \quad (3.15)$$

All torques acting on each link can also be found by successively switching $i = n, n - 1, \dots, 1$. Both iterations are running in the direction of descending i which is the operation called backward recursion [47]. To complete the algorithm, a forward recursion is employed to find a relationship between $q, \dot{q}, \ddot{q}$ and $a_{c,i}, \omega_i, \alpha_i$ through iterations running in the direction of ascending i [47]. Equation (3.16) gives the angular velocity ω_i of frame $\{i\}$ in frame $\{i\}$ [35].

$$\omega_i = (R_{i-1}^i)^T \omega_{i-1} + h_i \dot{q}_i \quad (3.16)$$

where

$$h_i = (R_0^i)^T z_{i-1} \quad (3.17)$$

is the axis of rotation of joint i in frame $\{i\}$ and $z_{i-1} = [0 \ 0 \ 1]^T$ [5, 47]. The angular acceleration α_i of frame i in frame $\{i\}$ [35] is given as:

$$\alpha_i = (R_{i-1}^i)^T \alpha_{i-1} + h_i \ddot{q}_i + \omega_i \times h_i \dot{q}_i \quad (3.18)$$

Lastly, the linear acceleration $a_{c,i}$ as an acceleration for the COM of link i in frame $\{i\}$ is given in Equation (3.19).

$$a_{c,i} = (R_{i-1}^i)^T a_{e,i-1} + \dot{\omega}_i \times r_{i,ci} + \omega_i \times (\omega_i \times r_{i,ci}) \quad (3.19)$$

Finding the acceleration for the end of link i can be done by replacing $r_{i,ci}$ with $r_{i,i+1}$ [35] resulting in Equation (3.20).

$$a_{e,i} = (R_{i-1}^i)^T a_{e,i-1} + \dot{\omega}_i \times r_{i,i+1} + \omega_i \times (\omega_i \times r_{i,i+1}) \quad (3.20)$$

The sequence or procedure for the recursive Newton-Euler equations is listed next:

- Begins with the initial constraints for the base link as $\omega_0 = \Omega$, $\alpha_0 = \dot{\Omega}$, $a_{c,0} = a_{e,0} = \dot{v}$ in the forward recursion [47], in Equations (3.16), (3.18), (3.20) and (3.19) sequentially, in an increasing order of i .
- Then followed by the computation of the joint forces and torques for each link commencing from the end-effector to the base frame using link $n + 1$ constraints as $f_{n+1} = 0$, $\tau_{n+1} = 0$ [35] in the backward recursion Equations (3.14) and (3.15) in decreasing order of i .

The recursive Newton-Euler equations were applied to analyse the dynamics of the manipulator shown in Figure 2.1 according to the Spong et al., [35], but for a floating-base instead of the fixed-base (Appendix A details the step by step derivation). Physical interaction of the manipulator with the environment will induce forces and moments $F_e = [f_x \ f_y \ f_z \ n_x \ n_y \ n_z]^T$ at the end-effector. But these moments and forces due to physical interaction were not considered when deriving the dynamic equations. The transpose of the manipulator Jacobian $J(q)$ can relate the end-effector forces and moments to the joint torques as $\tau_{ext} = J(q)^T F_e$ [35]. Hence the external torques τ_{ext} is introduced to the dynamic equation of the manipulator. The closed-form equations [35, 47] are found by solving Equation (3.15) which can be written in a matrix form:

$$D(q)\ddot{q} + C(q, \dot{q})\dot{q} + g(q) = \tau + \tau_{ext} \quad (3.21)$$

where the matrices; $D(q)$ is the inertia matrix, $C(q, \dot{q})$ is the Coriolis and centripetal matrix, $g(q)$ is the gravity vector and τ is the vector of the joint torques. For a 2-link manipulator with two revolute joints, these matrices are given as:

$$D(q) = \begin{bmatrix} d_{11} & d_{12} \\ d_{21} & d_{22} \end{bmatrix}, C(q, \dot{q}) = \begin{bmatrix} c_{11} & c_{12} \\ c_{21} & 0 \end{bmatrix}, g(q) = \begin{bmatrix} \phi_1 \\ \phi_2 \end{bmatrix}, \tau = \begin{bmatrix} \tau_1 \\ \tau_2 \end{bmatrix} \quad (3.22)$$

3.3 Dynamics of the Aerial Manipulator

Presuming that the base of the manipulator is collinear with the quadrotor's COM, consider the initial conditions listed next for the base-fixed frame $\{0\}$ [47].

$$v_0 = R_0^W \dot{\vec{r}}, a_{c,0} = R_0^W (\ddot{\vec{r}} + [0 \ 0 \ g]^T) \quad (3.23)$$

$$\omega_0 = R_0^B \vec{\omega}, \dot{\omega}_0 = R_0^B \dot{\vec{\omega}} \quad (3.24)$$

where $\dot{\omega}_0$, $a_{c,0}$, ω_0 , v_0 are; the angular and linear accelerations, and angular and linear velocities of the quadrotor with respect to the base-fixed frame $\{0\}$ as in Figure 2.1 and matrices R_0^W , and R_0^B can be determined as:

$$R_0^W = R_0^B R_B^W, R_0^B = \begin{bmatrix} -1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix} \quad (3.25)$$

The recursive Newton-Euler equations permit the extraction of forces and torques exerted on the COM of the quadrotor resulting from being coupled with the manipulator. The force f_1 and torque τ_1 obtained in the manipulator dynamics for frame $\{1\}$ can be expressed with respect to the base-fixed frame $\{0\}$ [47] as in Equation (3.26).

$$f_0 = R_0^1 f_1, \tau_0 = R_0^1 \tau_1 \quad (3.26)$$

where R_0^1 is the rotation matrix that transforms an object from $\{1\}$ to base-fixed frame $\{0\}$ [35] and can be computed as follows:

$$R_0^1 = \begin{bmatrix} \cos q_1 & -\sin q_1 & 0 \\ \sin q_1 & \cos q_1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (3.27)$$

The transformation of forces f_0 and torques τ_0 in the base-fixed frame $\{0\}$ to the world frame $\{W\}$ and the body frame $\{B\}$ is done through the rotation matrix R_0^W and matrix R_0^B , respectively. This is because the quadrotor model has foreign forces expressed with respect to the world frame $\{W\}$, while the foreign torques are expressed with respect to the body frame $\{B\}$ [47]. The resulting linear motion, considering the force f_W plied on the quadrotor due to interconnectedness with the manipulator, is given by:

$$\ddot{\vec{r}} = \begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} \end{bmatrix} = \begin{bmatrix} \frac{2T}{M_t}(q_o q_j + q_i q_k) - \frac{f_{W,x}}{M_t} \\ \frac{2T}{M_t}(q_j q_k - q_i q_o) - \frac{f_{W,y}}{M_t} \\ \frac{T}{M_t}(q_o^2 - q_i^2 - q_j^2 + q_k^2) - g - \frac{f_{W,z}}{M_t} \end{bmatrix} \quad (3.28)$$

The rotational motion considering the coupling torque η_B is given by:

$$\dot{\vec{\omega}} = \begin{bmatrix} \dot{\omega}_x \\ \dot{\omega}_y \\ \dot{\omega}_z \end{bmatrix} = \begin{bmatrix} (\tau_x + \omega_y \omega_z (J_{yy} - J_{zz}) - \eta_{B,x}) / J_{xx} \\ (\tau_y + \omega_x \omega_z (J_{zz} - J_{xx}) - \eta_{B,y}) / J_{yy} \\ (\tau_z + \omega_x \omega_y (J_{xx} - J_{yy}) - \eta_{B,z}) / J_{zz} \end{bmatrix} \quad (3.29)$$

where the force vector $f_W = [f_{W,x} \ f_{W,y} \ f_{W,z}]^T$ and the torque vector $\eta_B = [\eta_{B,x} \ \eta_{B,y} \ \eta_{B,z}]^T$ satisfy the the following relations [47]:

$$f_W = (R_0^W)^{-1} f_0, \ \eta_B = (R_0^B)^{-1} \tau_0 \quad (3.30)$$

The complete model can now be simulated and analysed for correctness. First with zero inputs, the center of mass of the aerial manipulator is shown in figure 3.4. It's not a controlled motion, gravity is acting on it as well as the interaction forces and moments are inducing motion in the x and y directions. To begin validation, one must impose constraints on the model like limiting motion in the z- direction to be positive when gaining altitude. Limiting the joint angle q_1 between 15° and 165° to prevent the first link from colliding with the quadrotor base. At this stage, manually setting the inputs to resemble controlled motion can prove to be very difficult. The model is very complex and coupled, hence there is a need to linearise or streamline the control inputs such that the motion can be predicted.

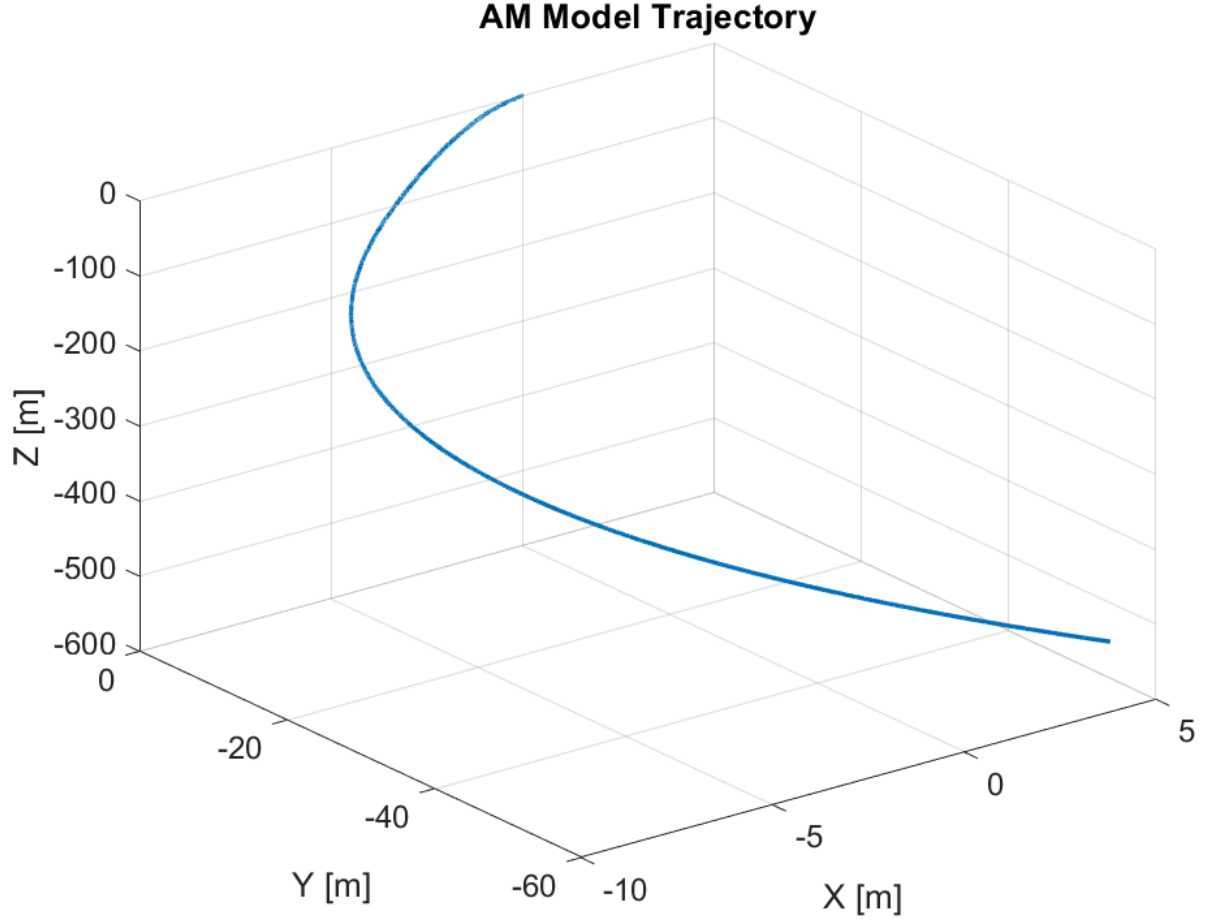


Figure 3.4: Aerial manipulator validation: Motion for zero control inputs

3.4 Chapter summary

The modelling of the aerial manipulator was performed by first treating it as two subsystems, namely the quadrotor and the manipulator. Quadrotor dynamics were derived employing the Newton-Euler algorithm with a quaternion-based orientation for a X-right, Y-forward and Z-up coordinate system. While the 2-DOF manipulator dynamics were analysed using the recursive Newton-Euler equations. The gyroscopic effects resulting from spinning propellers, rotational drag and torques resulting from accelerating/decelerating propellers were deemed to be negligible for the quadrotor dynamics. After obtaining the manipulator dynamics, the base of the manipulator was presumed to be collinear with COM of the quadrotor, in order to model the manipulator with a floating base.

Controller Design

This chapter discusses the control system based on the AVIC technique for the manipulator and an integral LQR control for the quadrotor. Figure 4.1 depicts the overall control architecture for the aerial manipulator. The AVIC strategy establishes a desired dynamic correlation between the end-effector position and the contact force resulting in a modified trajectory that can be fed into the PD position control. The dynamics of the aerial manipulator require a control algorithm that considers the effects from each subsystem (quadrotor or manipulator) to the other due to the interconnectedness. For example, how the quadrotor accelerations and velocities influences the motion of the manipulator, and reversibly how the interconnection forces and moments exerted on the quadrotor's COM from the manipulator influences the behaviour of the quadrotor.

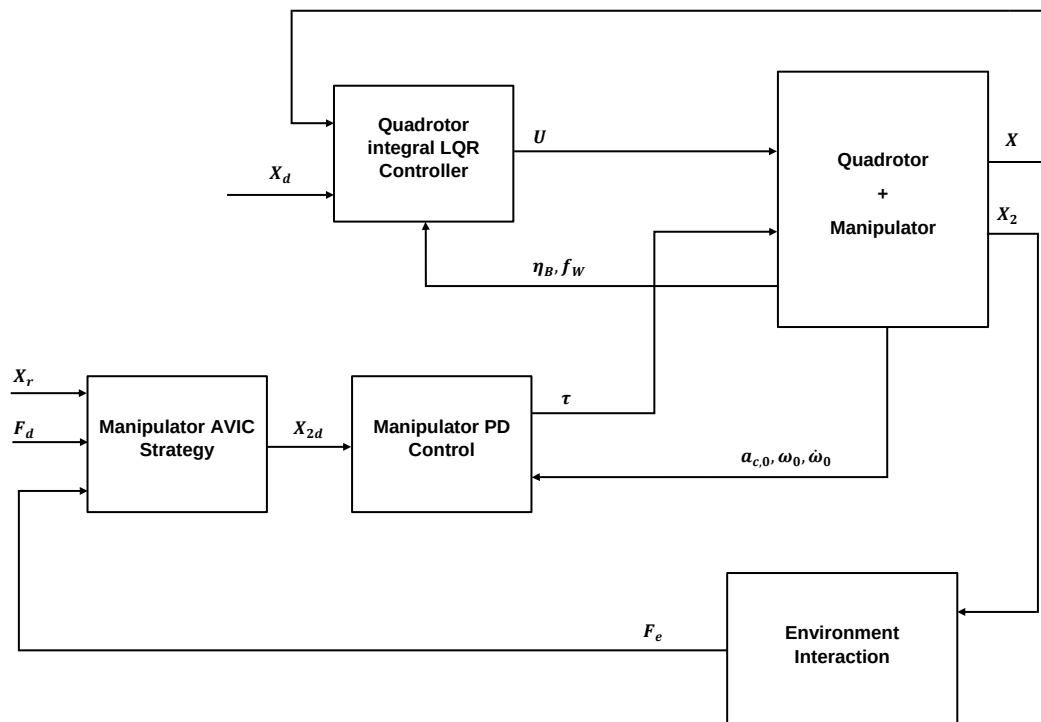


Figure 4.1: The overall control architecture for aerial manipulator

4.1 Quadrotor Control

Since the quadrotor is an under-actuated system due to having only four inputs (rotational speeds from the four propellers) to control the six state variables is not straightforward. Therefore, it is common to control the movement in y and x- direction through controlling the pitch and roll rotation, respectively. The objective for optimal controls is to ascertain a control signal such that the system being regulated can attain physical constraints and minimise a performance function [42]. Here, the well-known linear quadratic regulator will be employed to attain the control input that will minimise the quadratic performance function for the quadrotor's linear model. The linearisation process that was presented by Sabatino, [12] for a quadrotor's dynamics with Euler angles representation was utilised to linearise a quaternion-based orientation system, which begins with a state-space model described below.

State-space model

The state's vector from Equation (3.7) can be re-organised as follows:

$$X = [x \ y \ z \ q_0 \ q_i \ q_j \ q_k \ \dot{x} \ \dot{y} \ \dot{z} \ \omega_x \ \omega_y \ \omega_z]^T \in \mathbb{R}^{13}$$

The dynamics of the quadrotor from Equation (3.11) can be reworked in the state-space form [12]:

$$\dot{X} = f(X) + \sum_{i=1}^4 s_i(X) U_i \quad (4.1)$$

where function $f(X)$ is determined as:

$$f(X) = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \\ -\frac{1}{2}(q_i\omega_x + q_j\omega_y + q_k\omega_z) \\ \frac{1}{2}(q_0\omega_x + q_j\omega_z - q_k\omega_y) \\ \frac{1}{2}(q_0\omega_y - q_i\omega_z + q_k\omega_x) \\ \frac{1}{2}(q_0\omega_z + q_i\omega_y - q_j\omega_x) \\ 0 \\ 0 \\ -g \\ \omega_y\omega_z(J_{yy} - J_{zz})/J_{xx} \\ \omega_x\omega_z(J_{zz} - J_{xx})/J_{yy} \\ \omega_x\omega_y(J_{xx} - J_{yy})/J_{zz} \end{bmatrix}, \quad (4.2)$$

The function $s_i(X)$ is set as:

$$\begin{aligned} s_1(X) &= [0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ s_1^8 \ s_1^9 \ s_1^{10} \ 0 \ 0 \ 0]^T \in \mathbb{R}^{13}, \\ s_2(X) &= [0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ \frac{1}{J_{xx}} \ 0 \ 0]^T \in \mathbb{R}^{13}, \\ s_3(X) &= [0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ \frac{1}{J_{yy}} \ 0]^T \in \mathbb{R}^{13}, \\ s_4(X) &= [0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ \frac{1}{J_{zz}}]^T \in \mathbb{R}^{13}, \end{aligned}$$

with

$$\begin{aligned} s_1^8 &= \frac{2}{M_t} (q_o \ q_j + q_i \ q_k), \\ s_1^9 &= \frac{2}{M_t} (q_j \ q_k - q_i \ q_o), \\ s_1^{10} &= \frac{1}{M_t} (q_o^2 - q_i^2 - q_j^2 + q_k^2). \end{aligned}$$

Linearisation

To linearise the quadrotor model, the control input vector from Equation (3.8) can be renamed as: $U = [U_1 \ U_2 \ U_3 \ U_4]^T \in \mathbb{R}^4$. The process for linearisation is developed around an equilibrium point $\bar{X}$, with the allied fixed input $\bar{U}$ that satisfies the following algebraic system [12, 42]:

$$0 = f(\bar{X}, \bar{U}) \quad (4.3)$$

To perform the linearisation, an equilibrium point for the quadrotor in hover mode is determined as follows [22, 10]:

$$\bar{X} = [0 \ 0 \ 0 \ 1 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0]^T \in \mathbb{R}^{13} \quad (4.4)$$

From the equations of motion (3.11), the equilibrium point in Equation (4.4) requires the constant control input value to be:

$$\bar{U} = [M_t \ g \ 0 \ 0 \ 0]^T \in \mathbb{R}^4 \quad (4.5)$$

This result is the thrust needed to cancel the quadrotor's weight and the manipulator's weight hence M_t (being the aerial manipulator's total mass). Thus, the hovering configuration of the system is guaranteed. After the determination of the equilibrium point $\bar{X}$ with the matching control input $\bar{U}$, the matrices allied with the linear system can be obtained after performing some partial derivatives as the following:

$$\mathbf{A} = \frac{\partial f(X, U)}{\partial X} \Big|_{(X, U) = (\bar{X}, \bar{U})} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{2} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{2} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{2} \\ 0 & 0 & 0 & 0 & 0 & 2g & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -2g & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (4.6)$$

$$\mathbf{B} = \frac{\partial f(X, U)}{\partial U} \Big|_{(X, U) = (\bar{X}, \bar{U})} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ \frac{1}{M_t} & 0 & 0 & 0 \\ 0 & \frac{1}{J_{xx}} & 0 & 0 \\ 0 & 0 & \frac{1}{J_{yy}} & 0 \\ 0 & 0 & 0 & \frac{1}{J_{zz}} \end{bmatrix} \quad (4.7)$$

The forces and torques exerted on the quadrotor due to interconnection with the manipulator can be considered as the disturbance to the quadrotor subsystem control and set as $\mathbf{d}$:

$$\mathbf{d} = [f_{W,x} \ f_{W,y} \ f_{W,z} \ \eta_{B,x} \ \eta_{B,y} \ \eta_{B,z}]^T \in \Re^6 \quad (4.8)$$

$$\mathbf{D} = \frac{\partial f(X, U, d)}{\partial d} \Big|_{(X, U) = (\bar{X}, \bar{U})} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ \frac{1}{M_t} & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{1}{M_t} & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{M_t} & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{J_{xx}} & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{J_{yy}} & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{1}{J_{zz}} \end{bmatrix} \quad (4.9)$$

For implementation purpose of the linear model, the state vector is modified to a (12×1) vector ξ as:

$$\xi = [x \ y \ z \ q_i \ q_j \ q_k \ \dot{x} \ \dot{y} \ \dot{z} \ \omega_x \ \omega_y \ \omega_z]^T$$

Then the linear system is found to be:

$$\dot{\xi} = \mathbf{A} \cdot \xi + \mathbf{B} \cdot U + \mathbf{D} \cdot \mathbf{d} \quad (4.10)$$

$$\ddot{x} = 2gq_j + \frac{f_{W,x}}{M_t}$$

$$\ddot{y} = -2gq_i + \frac{f_{W,y}}{M_t}$$

$$\ddot{z} = \frac{f_{W,z} + U_1}{M_t}$$

$$\dot{q}_i = \omega_x/2$$

$$\dot{q}_j = \omega_y/2$$

$$\dot{q}_k = \omega_z/2$$

$$\dot{\omega}_x = \frac{U_2 + \eta_{B,x}}{J_{xx}}$$

$$\dot{\omega}_y = \frac{U_3 + \eta_{B,y}}{J_{yy}}$$

$$\dot{\omega}_z = \frac{U_4 + \eta_{B,z}}{J_{zz}}$$

(4.11)

The linear system is then simulated with zero inputs to check whether the quadrotor's COM will remain in the initial position without any disturbance. Each torque inputs were tested and the quadrotor behaved as expected, the roll and pitch torques affecting the y and x-direction respectively. Figure 4.2 shows the quadrotor tracking set-points after some tuning of the control inputs. Considering the manipulator disturbances on the quadrotor, the joint angles were set to remain in their initial conditions. Setting zero control inputs to the system results in the motion depicted in Figure 4.3. The aerial manipulator's COM is seen to be gaining altitude although the control inputs are zero, this is because the interacting forces and moments are not always zero during the simulation. $f_{W,z}$ for example, begins at a non-zero value and remains non-zero during the simulation. Due to the coupling between the manipulator and the quadrotor, the manipulator needs to be linearised as well to have the aerial manipulator ready for manoeuvrability.

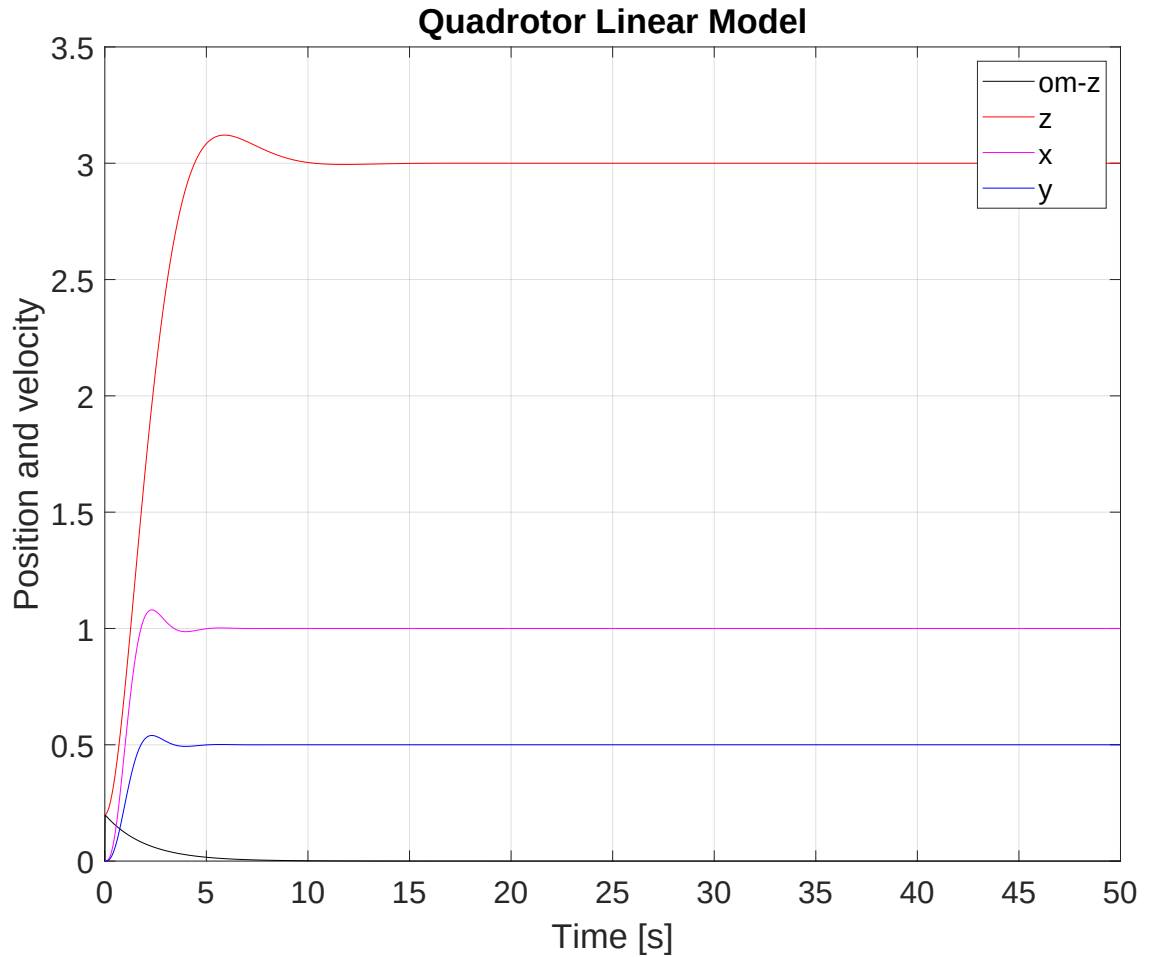


Figure 4.2: Quadrotor validation: Testing the quadrotor linear model

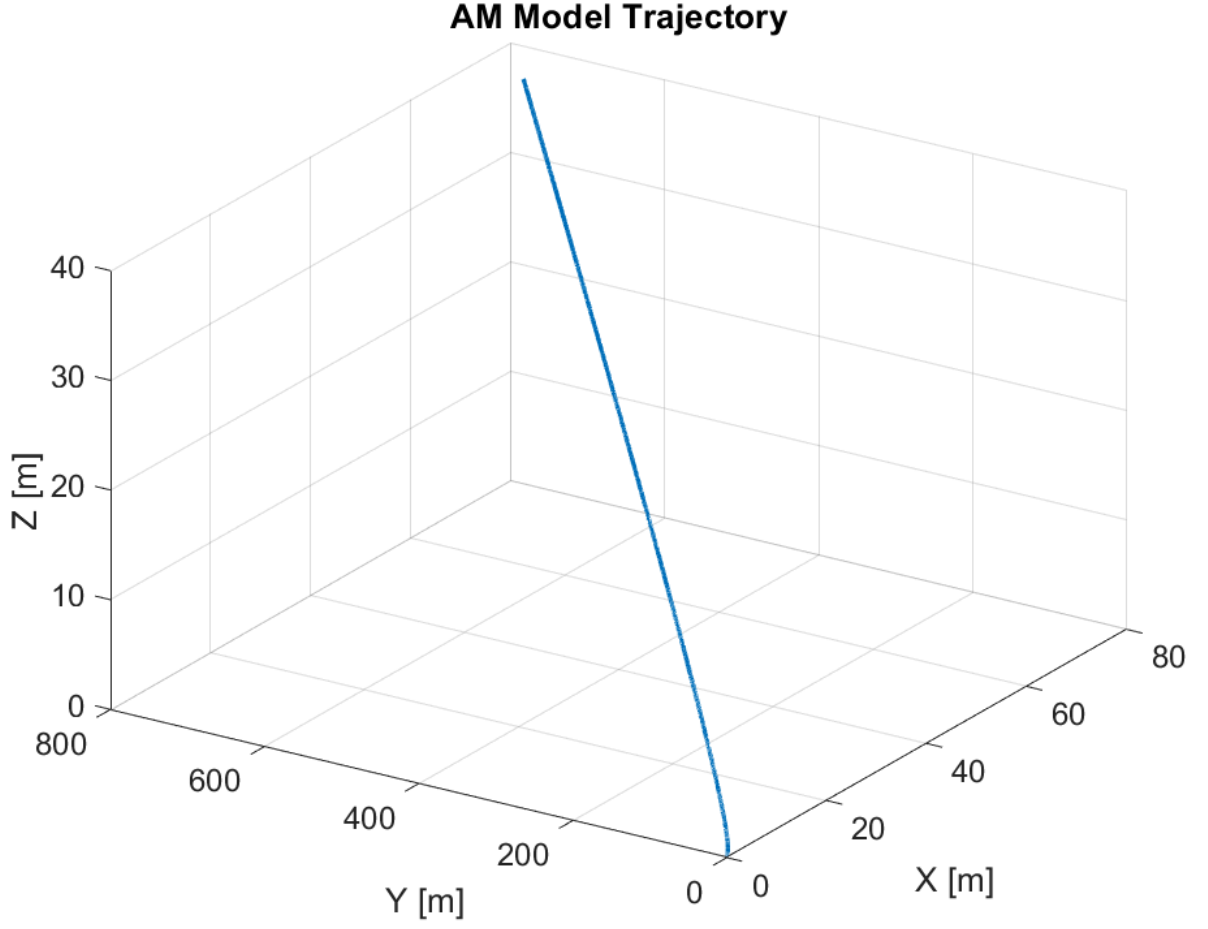


Figure 4.3: Aerial manipulator validation: Zero control inputs to the quadrotor

The quadrotor's subsystem's output for feedback control, Y output vector is chosen as follows:

$$Y = [x \ y \ z \ \omega_z]^T$$

alternatively, state vector can be mapped to the output vector as:

$$Y = \mathbf{C} \cdot \xi \quad (4.12)$$

where

$$\mathbf{C} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad (4.13)$$

Controllability and Observability

In modern control system theory, there are two very important notions, namely controllability and observability that were pioneered by R. Kalman in 1960 [12, 42]. These notions can be defined roughly as in the following:

- Controllability has to do with the possibility to ascertain a system's control input $U(t)$ that can yields any desired final state $\xi(t_f)$ from an initial state $\xi(t_0)$ within a finite time [35]. If it is possible, then the system is deemed to be controllable.
- Observability has to do with the possibility to ascertain the system's state $\xi(t)$ by measuring only the system's output $Y(t)$ over a bounded interval of time $t_0 \leq t \leq t_f$ [35]. If it is possible, then the system is deemed to be observable.

To check observability and controllability for a linear time-invariant (LTI) system can be correlated with an appropriate linear systems of algebraic equations [12, 42], matrix $\mathcal{O}$ and matrix $\mathcal{C}$, respectively. If these matrices are solvable, each will have a solution if and only if the matrix's rank is full. From the linear model in Equation (4.10) the dimensions are: $\xi \in \mathbb{R}^{12}$, $\mathbf{A} \in \mathbb{R}^{12 \times 12}$, $\mathbf{B} \in \mathbb{R}^{12 \times 4}$ and from Equation (4.12) are: $Y \in \mathbb{R}^4$, $\mathbf{C} \in \mathbb{R}^{4 \times 12}$. The controllability matrix $\mathcal{C}$ [12] is expressed as:

$$\mathcal{C}(\mathbf{A}, \mathbf{B}) = \begin{pmatrix} \mathbf{B} & \mathbf{AB} & \mathbf{A}^2\mathbf{B} & \dots & \mathbf{A}^{11}\mathbf{B} \end{pmatrix} \in \mathbb{R}^{12 \times 48}$$

The $\mathbf{B}$ matrix can control the $\mathbf{A}$ matrix if and only if the rank of $\mathcal{C}(\mathbf{A}, \mathbf{B})$ is equal to 12 or has full rank [12, 42]. The observability matrix $\mathcal{O}$ is expressed as:

$$\mathcal{O}(\mathbf{A}, \mathbf{C}) = \begin{pmatrix} \mathbf{C} \\ \mathbf{CA} \\ \mathbf{CA}^2 \\ \vdots \\ \mathbf{CA}^{11} \end{pmatrix} \in \mathbb{R}^{48 \times 12}$$

The $\mathbf{C}$ matrix can observe the $\mathbf{A}$ matrix if and only if the rank of $\mathcal{O}(\mathbf{A}, \mathbf{C})$ is equal to 12 or has full rank [42]. To check the rank of these matrices, simple commands were used in MATLAB, namely $rank(ctrb(\mathbf{A}, \mathbf{B}))$ for controllability status and $rank(observ(\mathbf{A}, \mathbf{C}))$ for observability status. The linear model was found to be observable and controllable, thus the LQR could be implemented for the quadrotor.

4.1.1 Controller implementation

The objective for optimal controls is to ascertain a control input such that the system being regulated can attain the physical constraints and minimise a performance function [35, 12]. As such, the desired performance can be achieved without requiring any maximum magnitude of the control inputs, and thus protecting and/or extending the motors' product life. The controllable linear system of the quadrotor without considering the effects of manipulator coupling can be re-expressed as:

$$\dot{\xi} = \mathbf{A} \cdot \xi + \mathbf{B} \cdot U \quad (4.14)$$

It is known that when using state feedback, one may achieve arbitrary closed-loop poles by choosing the feedback gains appropriately [35]. A gain matrix K can be selected to place eigenvalues of matrix $\mathbf{A} - \mathbf{B}K$, in such a way that the state feedback control input $U(t) = -K[\xi(t) - \xi_{ref}(t)]$ will stabilise the closed-loop system [42] to transfer the system's state $\xi(t)$ to the reference trajectory $\xi_{ref}(t)$. The selection of this K can be performed manually for small and simple systems, but not so much for complex high-order systems with multiple degree of freedoms like the quadrotor.

The linear quadratic regulator (LQR) will be used for the feedback gain selection process. It is a linear quadratic because the performance index is quadratic in nature to control a linear system [35]. The main purpose is to determine the corresponding control input $U(t)$ such that the performance function [10, 12] in Equation (4.15).

$$J = \int_0^\infty \left\{ [\xi(t) - \xi_{ref}(t)]^T \mathbf{Q} [\xi(t) - \xi_{ref}(t)] + U(t)^T \mathbf{R} U(t) \right\} dt \quad (4.15)$$

is minimised, with $\mathbf{Q} \in \mathbb{R}^{12 \times 12}$ and $\mathbf{R} \in \mathbb{R}^{4 \times 4}$ being positive semi-definite matrix and positive definite matrix, respectively [12]. The matrix $\mathbf{Q}$ can be perceived as the penalty and/or reward for the states. The matrix $\mathbf{R}$ can be perceived as the penalty and/or reward for actuators or control effort. Altering matrices $\mathbf{Q}$ and $\mathbf{R}$ appropriately alters the placements of the closed-loop poles [42], therefore influencing the system's dynamic performance.

The control input $U(t)$, which minimises the performance function, is the state linear feedback control:

$$U(t) = -K [\xi(t) - \xi_{ref}(t)] + \bar{U} \quad (4.16)$$

and the gain matrix K is determined as:

$$K = \mathbf{R}^{-1} \mathbf{B}^T S \quad (4.17)$$

The matrix S is the unique symmetric positive definite matrix executing the so-called algebraic Riccati equation [35, 42] of the form

$$0 = \mathbf{A}^T S + S \mathbf{A} - S \mathbf{B} \mathbf{R}^{-1} \mathbf{B}^T S + \mathbf{Q} \quad (4.18)$$

The solution to this algebraic equation can be found by Riccatti's method through the use of LQR function command in the Matlab environment as:

$$K = lqr(\mathbf{A}, \mathbf{B}, \mathbf{Q}, \mathbf{R})$$

The corresponding closed-loop system $\dot{\xi} = (\mathbf{A} - \mathbf{B} \cdot K) \cdot \xi$ is deemed to be asymptotically stable when $K = \mathbf{R}^{-1} \mathbf{B}^T S$ [12]. Thus, the gain matrix K can be used to track a reference $r(t) = [x_{ref}(t), y_{ref}(t), z_{ref}(t), \omega_{zref}(t)]^T$. Considering the output measurements, ξ_{ref} calculated using the reference signal r with:

$$\xi_{ref} = \mathbf{C}^T r,$$

where $\mathbf{C}$ is as given in Equation (4.12). The block diagram is depicted for LQR control architecture in Figure 4.4.

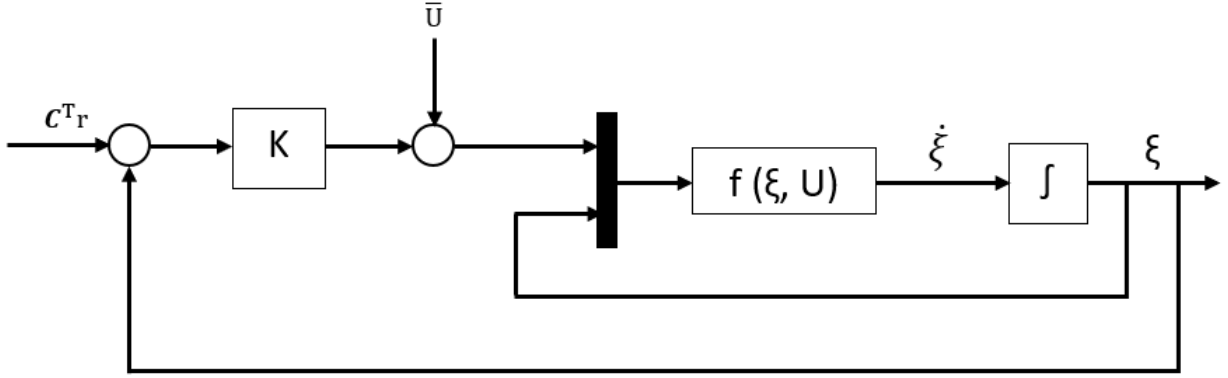


Figure 4.4: Block diagram of the LQR controller, adapted from [42]

Recall that the quadrotor model was linearised about the equilibrium point $(\bar{X}, \bar{U})$ for hovering configuration. The stability of the devised strategy is only guaranteed in the neighbourhood of the equilibrium point. As such, gain scheduling [38, 32, 42] is employed to ensure the stability of the system, for trajectories that would require the quadrotor to deviate away from the equilibrium point. Nonetheless, in this study, the trajectories are assumed to be in the neighbourhood of the equilibrium point.

The matrix $\mathbf{Q}$ and $\mathbf{R}$ were obtained through trial and error during simulation tests and are diagonal with the form:

$$\mathbf{Q} = diag(100, 100, 20, 1, 1, 20, 0, 0, 0, 1, 1, 1) \quad (4.19)$$

$$\mathbf{R} = diag(0.5, 0.1, 0.1, 0.1)$$

where $\mathbf{Q}$ entries represent the penalty amount to be applied to the states in the performance function from Equation (4.15). Entries that correspond with position (x, y) are the highest to penalise the position tracking error more strictly. The entries that correspond to $\dot{x}$, $\dot{y}$ and $\dot{z}$ are set to zero, since they can not be measured directly for most UAVs, but are determined via differentiation of the measured position [46]. The $\mathbf{Q}$ matrix can be used to specify which states are important and/or unmeasurable. On the other hand, the $\mathbf{R}$ matrix can be used to tune the aggressiveness of the controller. Thus, the $\mathbf{R}$ entries represent the penalty amount to be applied to individual control inputs. The entry that corresponds to generated thrust is 5 times the rotational torques inputs, meaning the control effort for thrust is more restricted.

So far, the LQR control is only a corrective effort that is proportional to the error, thus the controller will inherently have steady state error and effect of perturbations more especially for the nonlinear model. To overcome such limitations, the LQR controller is equipped with an integral action by incorporating an extra state variable [38, 26, 42] as:

$$\dot{\nu} = Y - r \quad (4.20)$$

where $Y = [x \ y \ z \ \omega_z]^T$ is the output vector given by Equation (4.12). Augmenting Equation (4.20) with the quadrotor dynamics to devise a feedback control input U for the augmented system in Equation (4.21) to ensure stability.

$$\begin{aligned} \dot{\xi} &= f(\xi, U, \varrho), \\ \dot{\nu} &= \mathbf{C} \cdot \xi - r \end{aligned} \quad (4.21)$$

at the equilibrium point $(\bar{\xi}, \bar{\nu})$, where $\bar{\nu}$ results in the required $\bar{U}$. $\varrho \in \mathbb{R}^p$ represent the unknown constant vector of disturbances, and an assumption can be made that for each pair $(r, \varrho) \in D_r \times D_\varrho$ there is a corresponding pair $(\bar{\xi}, \bar{U})$ [42] that fulfils the following constraints:

$$\begin{aligned} 0 &= f(\bar{\xi}, \bar{U}, \varrho), \\ 0 &= \mathbf{C}\bar{\xi} - r \end{aligned}$$

for any given reference signal $r(t)$. It can be shown that the next control input can be used:

$$U = -K_1(\xi - \bar{\xi}) - K_2\nu + \bar{U}$$

or since $\bar{\xi} = \mathbf{C}^T r$ at all times, then the control input can be expressed as:

$$U = -K_1(\xi - \mathbf{C}^T r) - K_2\nu + \bar{U}$$

that will be used to stabilise the non-linear model given in Equation (4.21) at the equilibrium point $(\bar{\xi}, \bar{\nu})$, where $K_1 \in \mathbb{R}^{4 \times 12}$, $K_2 \in \mathbb{R}^{4 \times 4}$, and can be generalised as $\mathcal{K}$:

$$\mathcal{K} = (K_1 \quad K_2)$$

which is determined such that $\tilde{\mathbf{A}} - \tilde{\mathbf{B}}\mathcal{K}$ matrix [38, 26] is asymptotically stable, with the augmented matrices as:

$$\tilde{\mathbf{A}} = \begin{pmatrix} \mathbf{A} & 0 \\ -\mathbf{C} & 0 \end{pmatrix}, \tilde{\mathbf{B}} = \begin{pmatrix} \mathbf{B} \\ 0 \end{pmatrix} \quad (4.22)$$

and the matrices $\mathbf{A}$, $\mathbf{B}$, $\mathbf{C}$ are attained from Equations (4.10) and (4.12). Figure 4.5 shows the integral LQR control architecture in blocks as in simulink.

where do the manipulator disturbances come in?

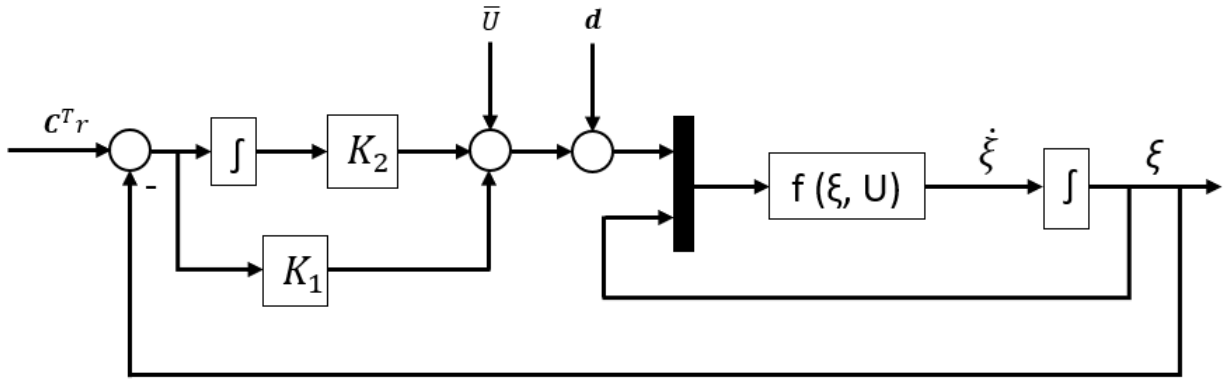


Figure 4.5: Quadrotor's block diagram for the integral LQR controller, adopted from [42]

To attain the matrix $\mathcal{K}$ for the integral LQR, consider the augmented state vector:

$$\zeta = \begin{bmatrix} \xi \\ \nu \end{bmatrix}$$

Following the LQR selection process, to obtain an appropriate gain matrix $\mathcal{K}$, the performance function in Equation (4.15) can be computed as:

$$J = \int_0^\infty [\zeta(t)^T \mathbf{Q}_\zeta \zeta(t) + U(t)^T \mathbf{R} U(t)] dt \quad (4.23)$$

where the matrix $\mathbf{Q}_\zeta$ [42] is expressed as:

$$\mathbf{Q}_\zeta = \begin{bmatrix} \mathbf{Q}_\xi & 0 \\ 0 & \mathbf{Q}_\nu \end{bmatrix}$$

The matrix $\mathbf{Q}_\xi \in \mathbb{R}^{12 \times 12}$ is perceived as the penalty for the state vector ξ which is similar in Equation (4.19), and the matrix $\mathbf{Q}_\nu \in \mathbb{R}^{4 \times 4}$ can be perceived as the penalty for the extra state vector ν . The $\mathbf{Q}_\nu$ entries were also obtained through trial and error and the penalty amounts as:

$$\mathbf{Q}_\nu = \text{diag}(100, 100, 100, 100)$$

To exemplify the effectiveness of the proposed integral LQR control, the first simulation scenario will show trajectory tracking of constant position values considering the interconnection with the manipulator without physical contact with the environment. The next section covers the position control of the manipulator.

4.2 Position control

Finally, after controlling the quadrotor which is the floating-base for the manipulator, the next action is to design a position controller to regulate the manipulator's movement. A control strategy called computed torque controller is applied to regulate the position of the manipulator end-effector [2, 35, 47]. In robotic manipulators, this strategy is an effective feedforward disturbance rejection that compensates through the non-linear system's inertial $D(q)$, Coriolis $C(q)$ and gravitational $g(q)$ forces [35]. The joint torques can be computed [2, 47] depending on feedback linearisation of the coupled system dynamics given by:

$$\tau = D(q)U_q + C(q, \dot{q})\dot{q} + g(q) \quad (4.24)$$

where,

$$U_q = \ddot{q}_d + K_{dq}\dot{e}_q + K_{pq}e_q \quad (4.25)$$

The K_{dq} and K_{pq} are the derivative and proportional gains respectively. These gains form the PD control that accepts the joint angle error between the desired joint angles and the measured joint angles. In Figure 4.6, the simulink control model is displayed with the PD controllers for joint angles and its outputs are sent to a function that compute the required joint torques. The sensed quadrotor dynamics, mainly the angular and translational accelerations and angular velocity are fed back into the control to compensate the floating-base movement for the computed joint torques.

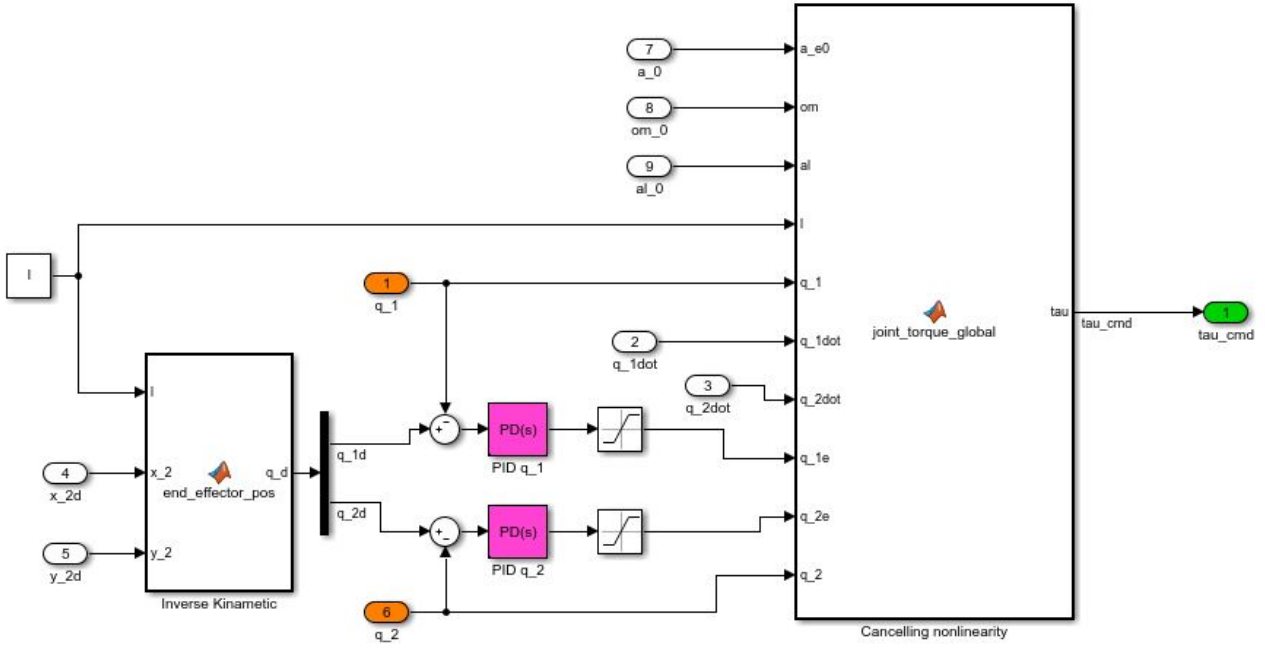


Figure 4.6: Simulink block diagram for the manipulator PD control

The inverse kinematic block in Figure 4.6 computes the desired joint angles from the AVIC's output position (x_{2d}, y_{2d}) as given by Equation (4.26) [35].

$$\begin{aligned}
 w &= \frac{x_{2d}^2 + y_{2d}^2 - l_1^2 - l_2^2}{2l_1l_2} \\
 q_{2d} &= \arctan\left(\frac{\sqrt{1-w^2}}{w}\right) \\
 q_{1d} &= \arctan\left[\frac{y_{2d}}{x_{2d}} - \arctan\left(\frac{l_2 \sin(q_{2d})}{l_1 + l_2 \cos(q_{2d})}\right)\right]
 \end{aligned} \tag{4.26}$$

An attempt was made to use the LQR control for both subsystems. As it will be shown in Chapter 5, the LQR does not respond fast enough to act as an inner-loop for the manipulator's position control. Thus, the force tracking for contact inspection was found to be undesirable. Details of these findings are not included since the force tracking had high overshoots and long settling time.

4.3 The Environment model and Contact Forces

To account for external forces acting on the end-effector, a component of the environment will be expressed in the dynamic system whose interaction forces will influence the quadrotor as well. Widely, the environment is presumed to be rigid and is modelled as a linear spring with a spring constant K_e as in Equation (4.27) [6, 9, 15].

$$F_e = K_e(X_e - X_2) \quad (4.27)$$

where F_e is the 6×1 force vector between the end-effector and the environment, X_e is the location of the environment and X_2 is the end-effector measured position trajectory. This contact forces can be transformed into contact torques using the manipulator Jacobian and hence the term τ_{ext} in the dynamic Equation (3.22) for contact torques can be determined by Equation (4.28).

$$\tau_{ext} = J^T(q)F_e \quad (4.28)$$

where

$$J^T(q) = \begin{bmatrix} -l_1 \sin(q_1) - l_2 \sin(q_1 + q_2) & l_1 \cos(q_1) + l_2 \cos(q_1 + q_2) & 0 & 0 & 0 & 1 \\ -l_2 \sin(q_1 + q_2) & l_2 \cos(q_1 + q_2) & 0 & 0 & 0 & 1 \end{bmatrix} \quad (4.29)$$

$J^T(q) \in \mathbb{R}^{6 \times 2}$ is the transpose of the manipulator Jacobian. The F_e vector resembles a FT sensor, but only the forces in the x_2 and y_2 axes will be simulated.

Figure 4.7 depicts orientation error for contact normal force on a non-planar environment for two-dimensional contour tracking task. The nominal normal direction is along the y_t -axis, but the real contact normal is along y_r -axis, which is due to the angular orientation error θ_t and it's approximate is given in Equation (4.30) [27]. This means that the measured force f is not aligned to the y_t -axis completely, but has f_{x_t} component along the x_t -axis. In practical situation, the f_{x_t} component will be disturbed by surface friction.

$$\theta_t = \arctan\left(\frac{f_{x_t}}{f_{y_t}}\right) \quad (4.30)$$

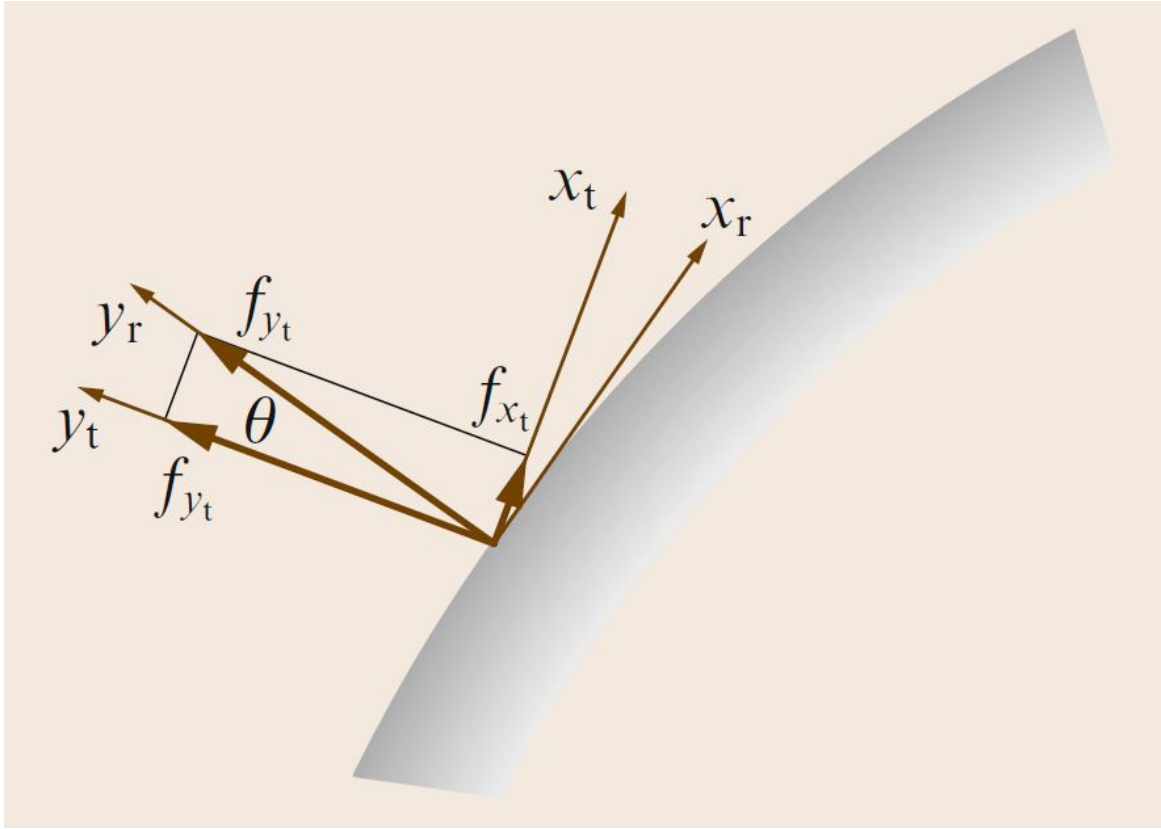


Figure 4.7: Orientation of contact normal force [27]

The orientation error θ_t is exploited to align the desired contact force f_d with the measured contact force f in-line with y_t -axis of the task frame.

It is okay to cast manipulator reaction forces on the quadrotor as disturbances. We however need to know how the quadrotor (base) forces are handled in the control of the manipulator.

4.4 Impedance control of the Manipulator

The aerial manipulator will be interacting with the environment for contact inspection. The goal in this section is the implementation of the AVIC strategy that was proposed by Duan et al., [9] for industrial robots. The block diagram for the AVIC strategy used in force tracking is displayed in Figure 4.8. The inverse kinematic block computes required joint angles q_d with respect to $X_{2d}(t)$ as in Equation (4.26). The end-effector position $X_2(t)$ is computed by applying the forward kinematics on the measured joint angles q .

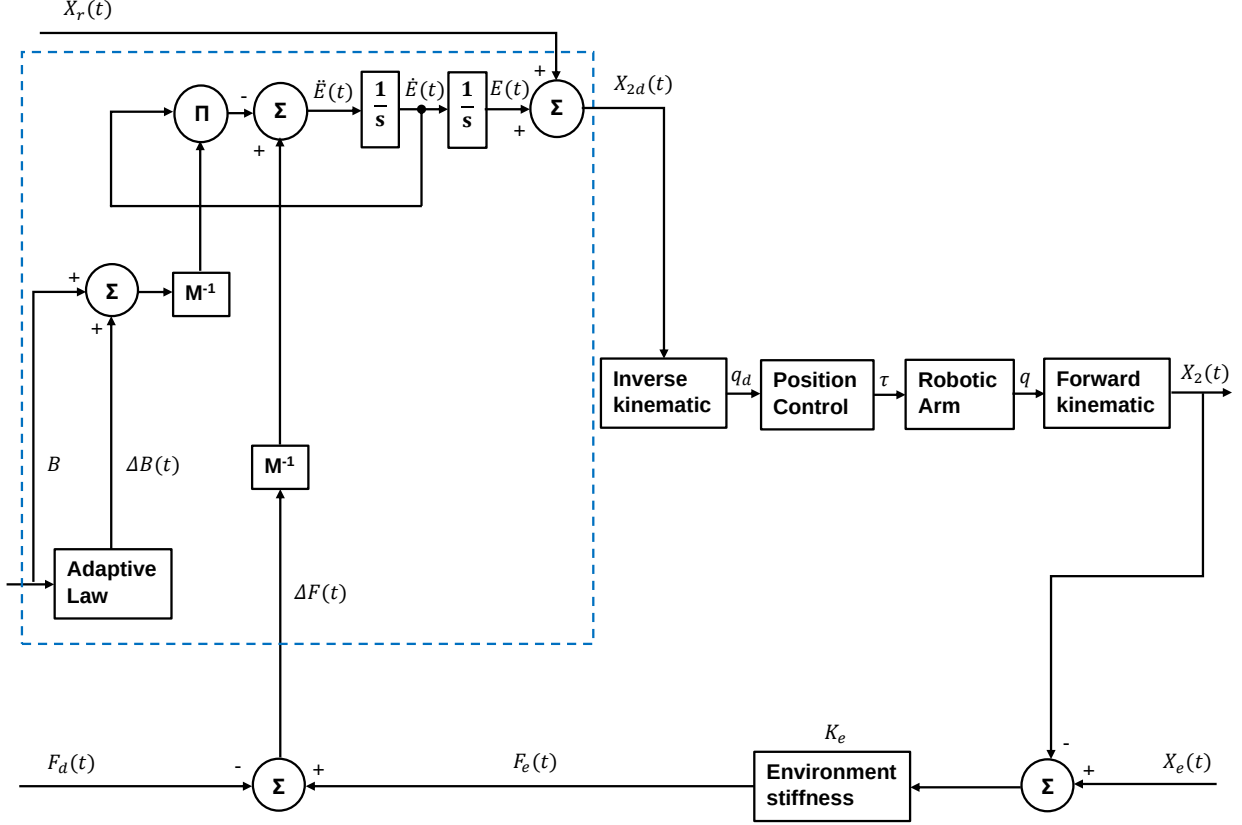


Figure 4.8: The adaptive variable impedance control architecture, adapted from [9]

The blue border in Figure 4.8 represent the AVIC strategy which will be discussed after the classical impedance control has been derived. The $X_r(t)$ and $X_{2d}(t)$ representing the reference trajectory and the modified trajectory, respectively. Assuming that the manipulator PD control guarantees stability such that $X_{2d}(t) = X_2(t)$, the end-effector position error due to physical interaction can be expressed as $E = X_2 - X_r = X_{2d} - X_r$. Therefore, Equation (4.27) can be rewritten as $F_e = K_e(X_e - X_{2d})$. Expressing the desired force as F_d and the force error as $\Delta F = F_e - F_d$. Equation (1.1) can be expressed as a transfer function $Z(s) = 1/(Ms^2 + Bs + K)$ [9], then the target impedance [4] can be computed with:

$$M \frac{d^2 E(t)}{dt^2} + B \frac{dE(t)}{dt} + K E(t) = \Delta F(t) \quad (4.31)$$

where M , B and K are the impedance parameters for the manipulator's end-effector. The reference trajectory $X_r(t)$ consist of x_{2r} and y_{2r} in the end-effector frame $\{E\}$, thus each Cartesian variable is said to be independent [9]. Considering only one-dimension to derive the impedance control, the force tracking error δf can be written as:

$$\delta f = -f_d + f_e = -f_d + k_e(y_e - y_{2d}) = -f_d + k_e y_e - k_e(y_{2r} + z(s) \delta f) \quad (4.32)$$

Substituting $z(s) = 1/(ms^2 + bs + k)$ into the tracking error (4.32), leads to:

$$\delta f(ms^2 + bs + k + k_e) = (ms^2 + bs + k)[-f_d + k_e(y_e - y_{2r})] \quad (4.33)$$

From Equation (4.33), the force tracking error at the steady-state [9, 4] can be determined as:

$$\delta f_{ss} = \frac{k}{k + k_e} [-f_d + k_e(y_e - y_{2r})] \quad (4.34)$$

To ensure zero steady state error, the reference position trajectory must meet the following constraint [9]:

$$y_{2r} = y_e - \frac{f_d}{k_e} \quad (4.35)$$

The constraint in Equation (4.35) requires precise knowledge of the environmental location y_e and the environmental stiffness k_e , for reference position trajectory y_{2r} to compensate for position deviation due to the exerted force on the environment. The environmental information can not be known before hand entirely in practical application [9, 4], which would result in an unstable force tracking situations. Furthermore, these physical interactions will influence the stability of the aerial manipulator. This is the reason for the implementation of the AVIC strategy that guarantees stable interaction between the manipulator end-effector and the environment.

4.4.1 The contact force tracking based on AVIC strategy

To analyse the unknown environmental stiffness: revisit Equation (4.31) to expand it with $E = X_{2d} - X_e$ and $\Delta F = F_e - F_d$, such that the new target impedance is expressed as:

$$F_e - F_d = M(\ddot{X}_{2d} - \ddot{X}_e) + B(\dot{X}_{2d} - \dot{X}_e) + K(X_{2d} - X_e) \quad (4.36)$$

This is a reasonable assumption, since attaining accurate trajectory X_r is a challenging task due to the uncertain environmental properties [9]. Therefore X_r was replaced with the initial environmental location X_e . Once again, considering only one-dimensional interaction force to derive the AVIC strategy, Equation (4.36) becomes:

$$\delta f = m\ddot{e} + b\dot{e} + ke \quad (4.37)$$

where

$$e = y_{2d} - y_e$$

For desired contact force $f_d = 0$, the steady state constraint $\delta f_{ss} = 0$ is satisfied when $y_{2d} = y_e$, indicating that the end-effector just reached the environment with zero contact force f_e . But for non-zero desired force, the manipulator will exert a contact force on the environment resulting in $y_{2d} \neq y_e$, which in turn $\delta f_{ss} \neq 0$ as well, assuming the system remain stable. Setting the stiffness k to zero has been demonstrated as a control law [9, 15] for varying environmental stiffness, then Equation (4.37) becomes:

$$\delta f = m\ddot{e} + b\dot{e} \quad (4.38)$$

Substituting the contact force $f_e = k_e(y_e - y_{2d}) = -k_e e$ into Equation (4.38), the resultant impedance control law is defined as:

$$f_d = -(m\ddot{e} + b\dot{e} + k_e e) \quad (4.39)$$

Since the control law has an approximation of k_e prior to the selection of the impedance parameters m and b , the system achieve $\delta f_{ss} = 0$ as it stabilises even for an uncertain environmental stiffness. Equation (4.39) ensures force tracking for varying or uncertain environmental stiffness [9, 15]. Substituting $e = y_{2d} - y_e$ into Equation (4.38) in order to consider environmental location that is varying:

$$\delta f = m(\ddot{y}_{2d} - \ddot{y}_e) + b(\dot{y}_{2d} - \dot{y}_e) \quad (4.40)$$

For a task on a slope or a cosine surface, y_e will be time-varying which means $\dot{y}_e \neq 0$ and/or $\ddot{y}_e \neq 0$. If the environmental location is estimated as $\hat{y}_e = y_e - \delta y_e$, then the position error becomes $\hat{e} = e + \delta y_e$. The AVIC strategy is proposed as in Equation (4.41) which adjusts the damping coefficient but not modifying the mass coefficient of the impedance model [9].

$$f_e(t) - f_d(t) = m\ddot{\hat{e}}(t) + (b + \Delta b(t))\dot{\hat{e}}(t) \quad (4.41)$$

$\Delta b(t)$ is varied on-line corresponding to the force tracking error, and can be found as:

$$\begin{aligned} \Delta b(t) &= \frac{b}{\dot{\hat{e}}(t)} \Phi(t) \\ \Phi(t) &= \Phi(t - \lambda) + \sigma \frac{f_d(t - \lambda) - f_e(t - \lambda)}{b} \end{aligned} \quad (4.42)$$

where λ is the controller's sampling time and σ being the update rate [9]. The proof for stability and convergence for the AVIC strategy can be found in Appendix B.

4.5 Chapter summary

Three control strategies were discussed, namely the implementation of the AVIC for contact force tracking, PD position control for manipulator's joint torques and the integral LQR for the quadrotor's control inputs. The quaternion-based orientation system of the quadrotor was linearised to attain the linear model necessary to implement the LQR control with an integral action to remove or reduce the steady state errors that are inherent with LQR when there are input perturbations. A brief discussion on the controllability and observability notions for linear system and using the rank command in MATLAB to verify that the linear model was controllable and observable. The LQR presented in this chapter can be considered as the set-point tracking control which achieves perfect tracking for constant set-point, but not for time-varying set-point. Selection of the penalty matrices $\mathbf{Q}_\zeta$ and $\mathbf{R}$ were obtained through trial and error. For physical interaction, the environment was matched as a linear spring. The classical impedance control was derived first, followed by the AVIC strategy for uncertain and varying environmental information by varying the impedance damping coefficient on-line according to the force tracking error. The next chapter demonstrates the effectiveness of these strategies through simulations.

Chapter 5

Simulation Results and Discussion

The system's dynamics of the aerial manipulator and its controllers have been presented in the previous chapters. In this chapter, simulations and results findings are detailed to demonstrate how the objectives for the dissertation have been met. Namely; the implementation of the AVIC strategy for force tracking in contact inspection including the capabilities to handle the effects of uncertain environmental parameters, and the quadrotor's position tracking capabilities using the integral LQR controller during contact inspection. The first section details the simulation studies and the results of these simulations are presented in the sections thereafter as manipulator results focusing on the end-effector in physical contact with the environment and quadrotor results reporting on the stability of the quadrotor's behaviour during the physical interaction. The AVIC strategy will be compared to the classic impedance control (classic in-short) strategy.

5.1 Simulation Studies

The aerial manipulator's equations of motion and the control strategies were simulated in the MATLAB/SIMULINK program. The parameters used for the aerial manipulator can be found in Table 5.1 adapted from [5] mostly for the quadrotor's inertia, with the manipulator links assumed to be identical and their moments of inertia about the joints were $I_1 = I_2 = \frac{1}{3}m_1l_1^2$ [$kg \cdot m^2$] adapted from [47].

Table 5.1: System parameters

Parameter	Value	Unit
M	1.686	kg
J_{xx}	0.0038278	$kg \cdot m^2$
J_{yy}	0.0038288	$kg \cdot m^2$
J_{zz}	0.0076566	$kg \cdot m^2$
m_1	0.4	kg
m_2	0.4	kg
l_1	0.4	m
l_2	0.4	m
K_e	1000	N/m

The research aim is to achieve force tracking for contact inspection using the AVIC strategy. The developed control should handle uncertainties due to the environment, and the quadrotor's stability during the inspection as the environmental forces and moments are being experienced. Four simulation scenarios were devised for the suitability of the proposed control for contact inspection together with the integral LQR controller for the quadrotor's manoeuvrability.

First scenario is trajectory tracking without physical interaction, the desired trajectory was set to track constant values (set-point tracking) for both manipulator and quadrotor's position and the yaw rate was set to zero, as the following:

$$\left\{ \begin{array}{l} x_{ref} = 1 \, m \\ y_{ref} = 0.5 \, m \\ z_{ref} = 3 \, m, \\ \omega_{zref} = 0 \, rad/s \\ x_{2r} = -0.2 \, m \\ y_{2r} = 0.399 \, m \end{array} \right.$$

Second scenario, the manipulator's position track the same set-point as in the first scenario and the quadrotor is set to track a helix trajectory for the desired x position x_{ref} , the desired y position y_{ref} , and the desired z position z_{ref} in meters with respect to time t in seconds:

$$\begin{cases} x_{ref} = 0.4 \sin\left(\frac{\pi}{10}t\right) \\ y_{ref} = 0.4 \cos\left(\frac{\pi}{10}t\right) - 0.4 \\ z_{ref} = (1 - e^{-5t}) \cdot 0.25t \end{cases} \quad (5.1)$$

The desired yaw rate remains unchanged at zero for all simulation scenarios. The end-effector of the manipulator was simulated to being in contact with the environment from time $t = 25$ s to time $t = 45$ s. In this scenario, the AVIC strategy is compared to the classic strategy for tracking the desired constant force of magnitude 2 N. The end-effector reference trajectory was assumed to be the initial environmental location as discussed in section 4.4.1, therefore an uncertainty of 0.1% is applied to the environmental location for this scenario and the next one. This simulates the fact that the environmental information (location and/or stiffness) can not be known precisely.

In the third scenario, the two strategies are compared again for contact inspection with a desired force that increases linearly (at a rate of 0.4 N/s) from 0 N at time $t = 15$ s to 4 N at time $t = 25$ s and maintain the high force till time $t = 45$ s. Meanwhile, the quadrotor and manipulator are tracking the desired trajectories generated from cubic polynomials. The way-points used to form the desired trajectory for the quadrotor are given in Table 5.2 while the end-effector desired trajectory is determined using Equation 5.2. The end-effector reference position (x_{2r}, y_{2r}) has three segments, namely initial position (plane surface) for time $t \leq 20$ s, cubic (slope and curve surface) trajectory $p_2(t)$, and the final position (plane surface) from time $t > 28$ s.

Table 5.2: Way points used for quadrotor's desired trajectory generation using cubic polynomials

No.	t [s]	x_{ref} [m]	y_{ref} [m]	z_{ref} [m]
1	0	0	0	0.2
2	1	0	0	0.2
3	8	0	0	5
4	11	0	0	5
5	18	5	0	5
6	19	5	0	5
7	24	5	2	5
8	30	5	2	5
9	35	5.5	2	5.5
10	40	6	2	6
11	45	6.5	2	5.5
12	50	7	2	5

$$\begin{cases} x_{2r} = -0.2, y_{2r} = 0.399; & \text{if } t \leq 20 \\ p_2(t) = c_1 + c_2t + c_3t^2 + c_4t^3; & \text{if } 20 < t \leq 28 \\ x_{2r} = 0.2, y_{2r} = 0.6; & \text{if } 28 < t \leq 50 \end{cases} \quad (5.2)$$

The quadrotor's desired trajectories are essentially line segments joined together with the cubic polynomials to ensure continuous trajectories that meet desired requirements like the initial and final positions, and the initial and final velocities for each line segment. For the quadrotor, a velocity boundary was set at 0.05 m/s between way-point 9 and way-point 11 and then back to zero for the last way-point. The resultant 3D plot of the desired quadrotor position is depicted in Figure 5.1. The desired end-effector trajectory $p_2(t) = [x_{2r} \ y_{2r}]^T$, its initial velocity was set to zero while the final velocity was set to 0.001 m/s . The coefficients for $p_2(t)$ Equation 5.2 were found to be:

$$p_2(t) = \begin{cases} 19.625 - 2.6012t + 0.1114t^2 - 0.0015t^3 \\ 10.274 - 1.2953t + 0.0555t^2 - 0.0008t^3 \end{cases}$$

The resultant 2D plot for the end-effector desired position is depicted in Figure 5.2.

Finally, the fourth scenario has the desired trajectories similar to the third scenario while the force tracking is similar to the second scenario except that the uncertainty of environmental location is increased to 5%. This scenario tests contact inspection when the environmental location is more uncertain, while the previous two scenarios had small uncertainty to the location.

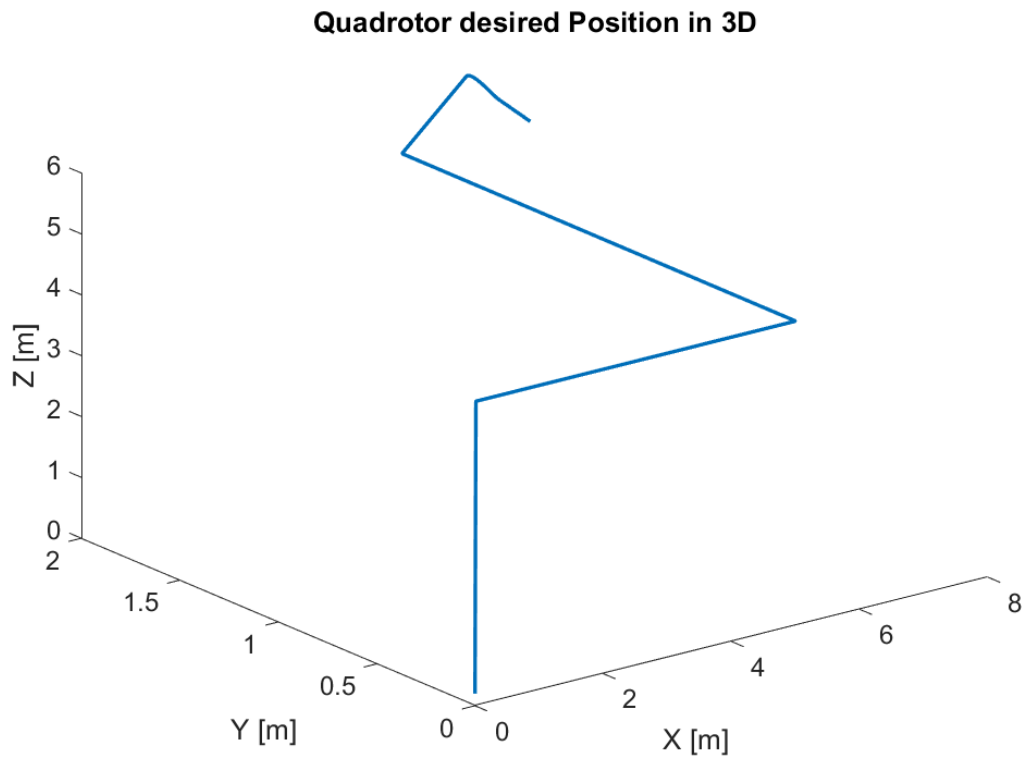


Figure 5.1: Quadrotor desired position - Scenario 3

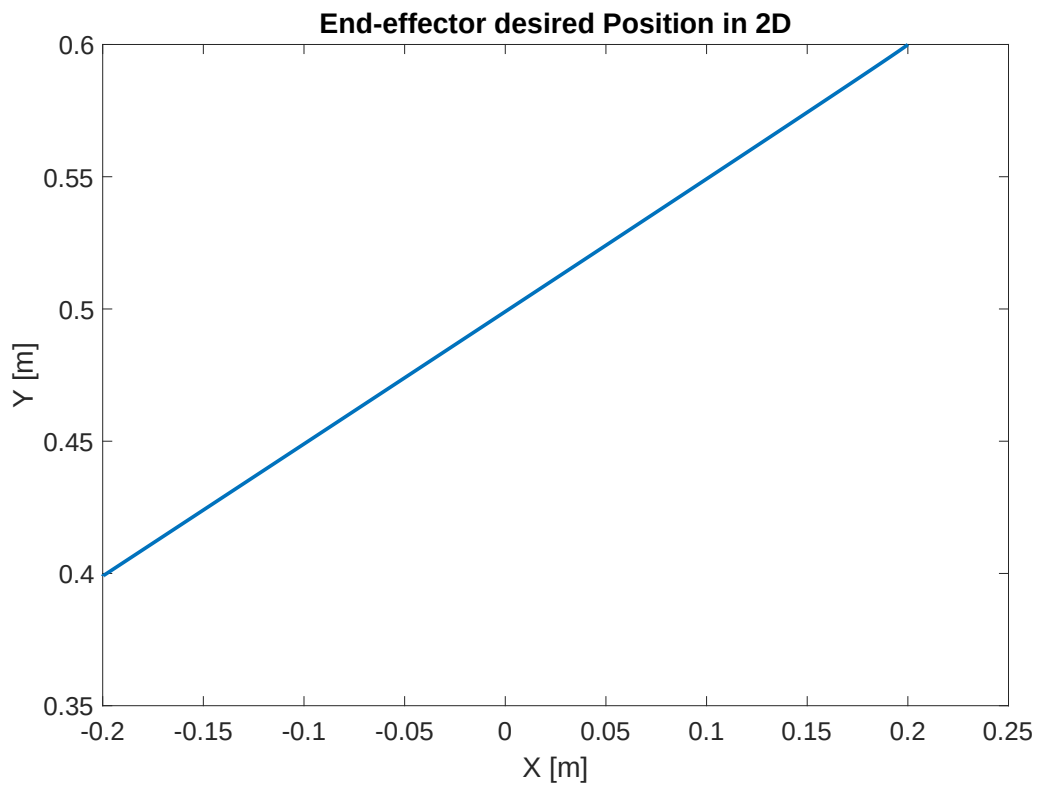


Figure 5.2: Manipulator's end-effector desired position - Scenario 3

5.2 Manipulator Results

The desired end-effector trajectory $p_2(t)$ in Equation (5.2) was obtained through cubic polynomials, as seen in Figure 5.2. The end-effector moves from initial position to final position which resembles plane-curve surface to slope surface then curve-plane surface. For contact inspection: the damping coefficient is initially guessed as $b = 18 [Ns/m]$ and the update rate as $\sigma = 0.5$ while the desired mass coefficient is set to $m = 0.8 [kg]$. In addition, the desired stiffness was set to $k = 0.8 [N/m]$ only for the classic strategy.

The first scenario, has the end-effector tracking a plane (initial position) without physical contact with the environment as shown in Figure 5.3. The desired end-effector position (x_{2r}, y_{2r}) is fed into the inverse kinematics block to compute the required manipulator joint angles q_d . These joint angles are track by the PD controller together with the computed torques. The gain parameters were then tuned in Simulink and are given in Table 5.3 for the 2-DOF manipulator.

Table 5.3: Manipulator Control Parameters

Parameter	Value
Kp_{q_1}	90
Kd_{q_1}	40
Kp_{q_2}	80
Kd_{q_2}	30

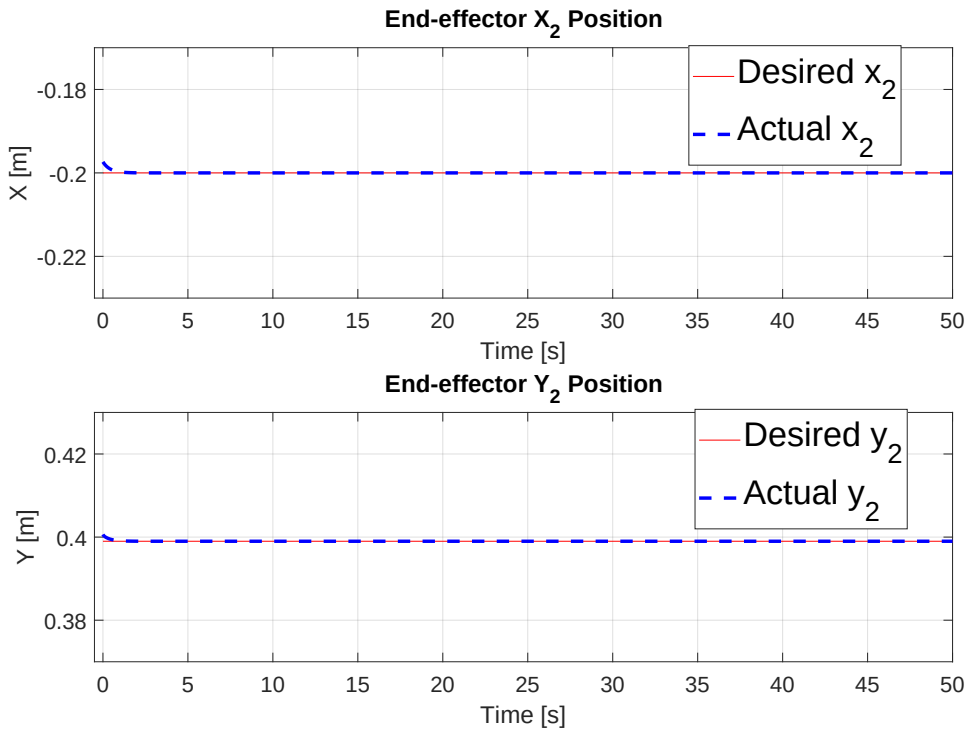


Figure 5.3: End-effector position tracking without physical interaction - Scenario 1

It can be stated that the set-point tracking can be achieved successfully, and Figure 5.4 further shows the tracking errors of the manipulator joint angles q converging to zero quickly. Figure 5.5 shows the manipulator control input torques τ necessary to achieve the plane surface tracking.

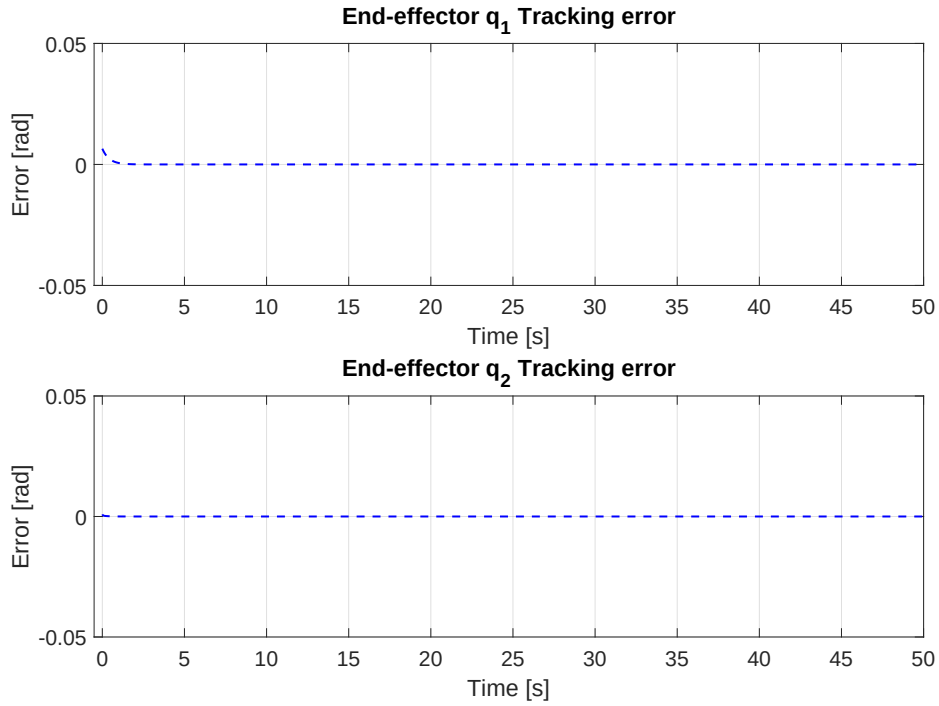


Figure 5.4: Manipulator joint angle tracking errors - Scenario 1

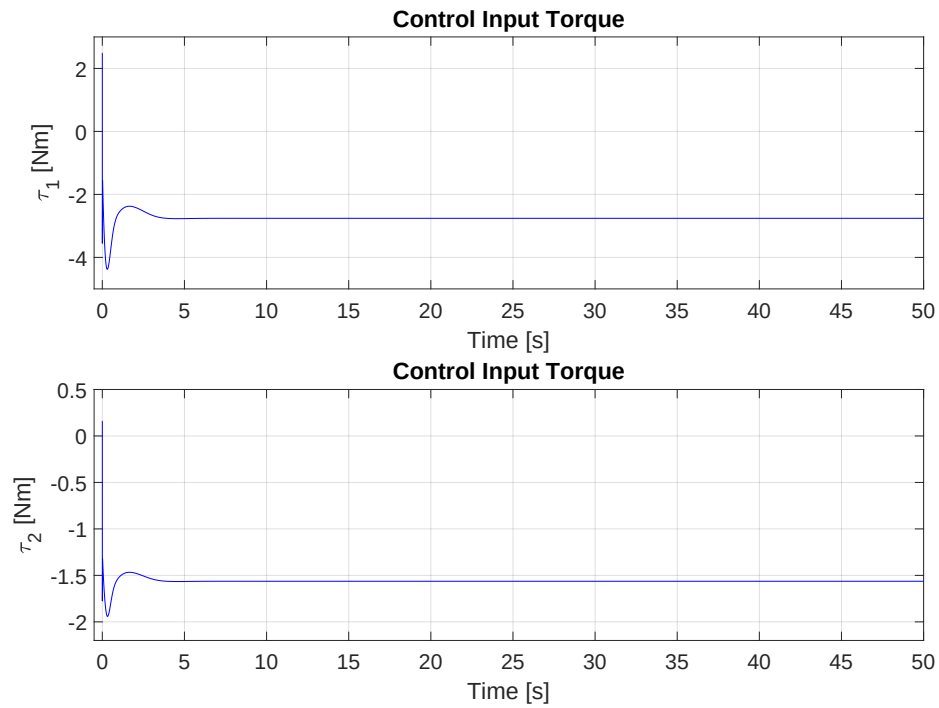


Figure 5.5: Manipulator joint inputs from computed torques - Scenario 1

The control effort required for this scenario can be easily achieved by the actuators of the manipulator. The second scenario considers physical interaction while the end-effector performs the same set-point tracking as in the first scenario. Figure 5.6 depicts the end-effector position tracking with the actual x_2 signal in the first sub-plot indicating to deviate from the desired x_2 signal. This is because during physical interaction, the inverse kinematics block receives a signal that is the modified trajectory rather than the desired end-effector position (x_{2r}, y_{2r}) and then required joint angles q_d will correspond with the modified trajectory. The contact inspection is simulated from time $t = 25\text{ s}$ to $t = 45\text{ s}$.

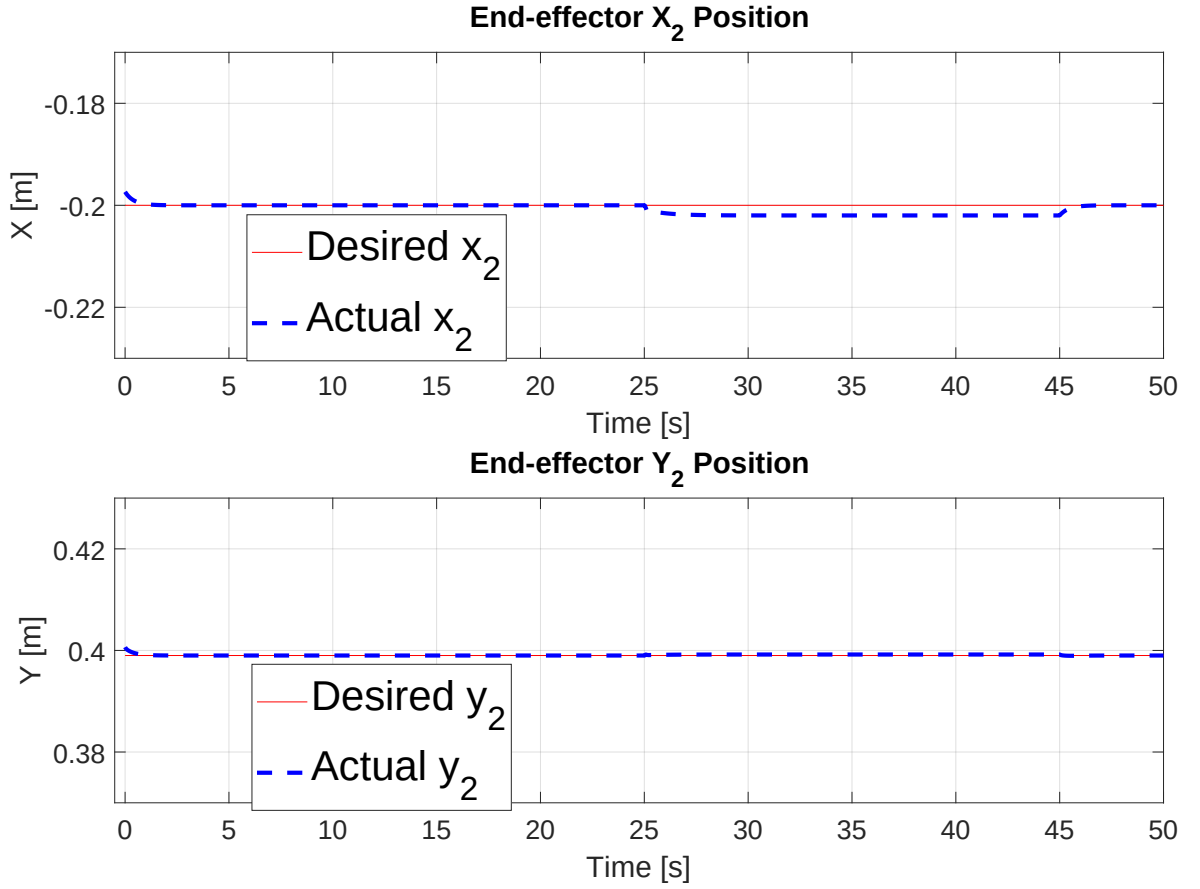


Figure 5.6: End-effector position tracking during contact inspection - Scenario 2

The comparison between the AVIC and classic strategy can be seen in Figure 5.7 in terms of the joint angle tracking errors. This shows that both strategies start to modify desired end-effector position (x_{2r}, y_{2r}) at time $t = 25\text{ s}$ so that the desired contact force can be attained. This force tracking performance is depicted in Figure 5.8. The resulting contact force being applied on the end-effector by the environment as in Equation (4.27) is for a two dimensional task. This resultant force f was calculated according to the well-known Pythagoras theorem as follows:

$$f = \sqrt{f_x^2 + f_y^2} \quad (5.3)$$

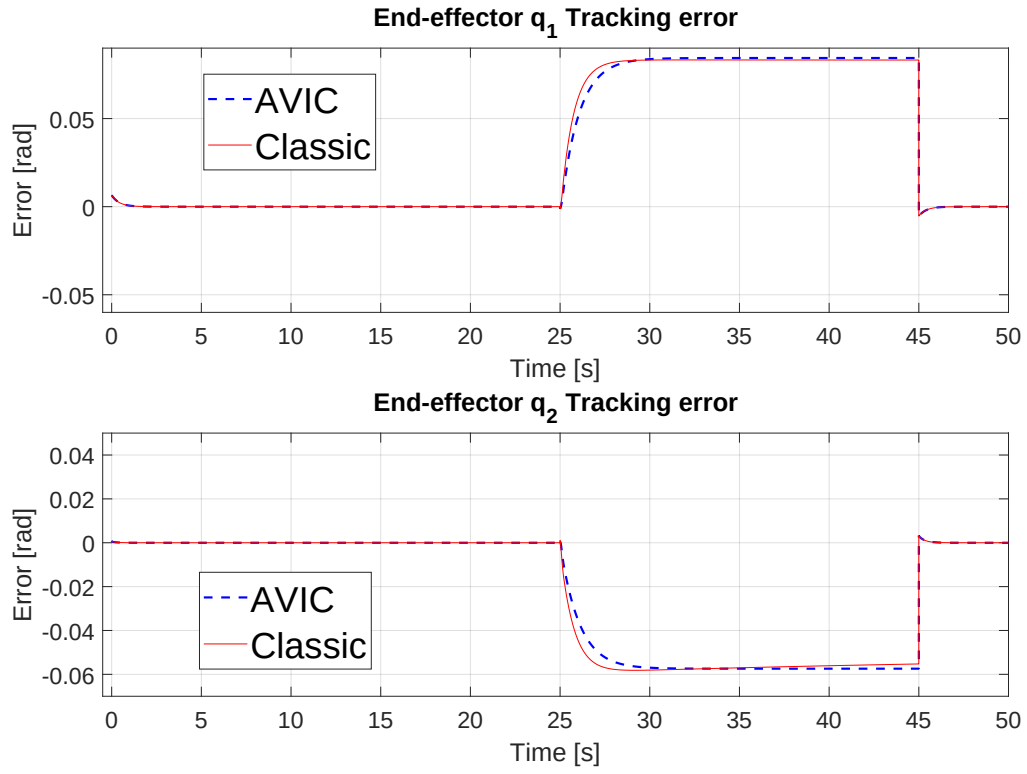


Figure 5.7: Manipulator joint angle tracking errors - Scenario 2

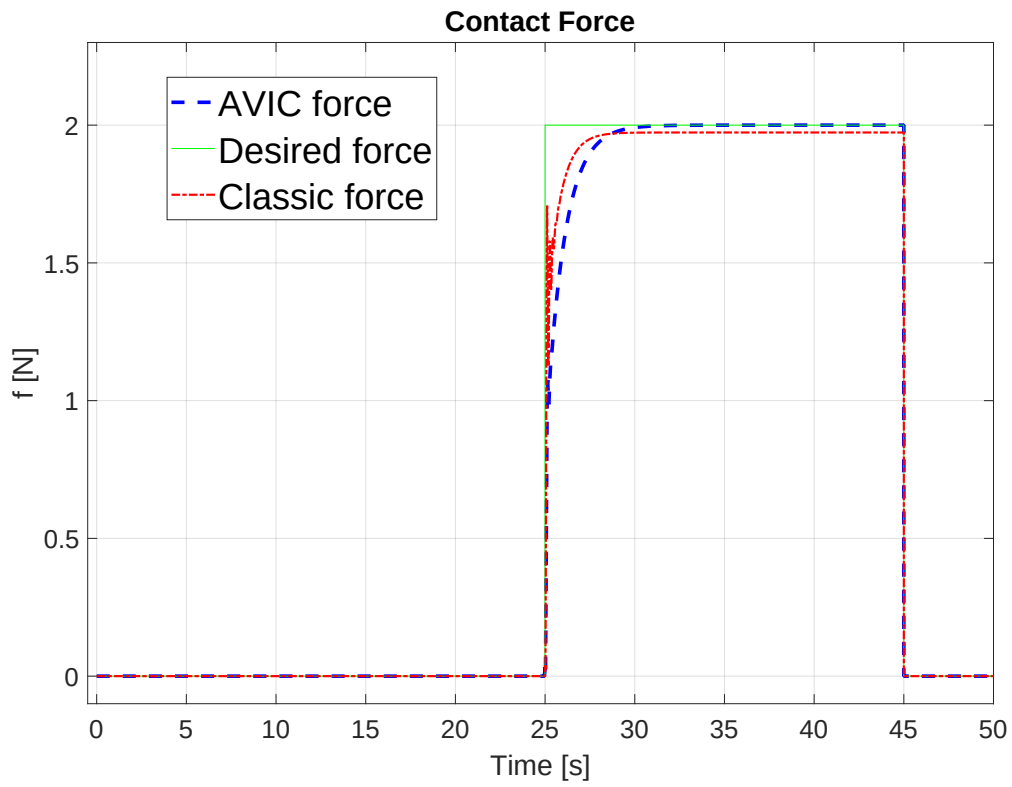


Figure 5.8: Contact force tracking between AVIC and classic strategy - Scenario 2

The force tracking of 2 N was fairly achieved by both strategies within 3 s of contact with the environment. However, there is a steady state error and some force jumps when the contact begins for the classic strategy. As can be seen from Figure 5.8, force spikes are present at time $t = 25\text{ s}$ for about 1 s and the force steady state error of less than 2% , while the AVIC presents zero steady state error and no such spikes.

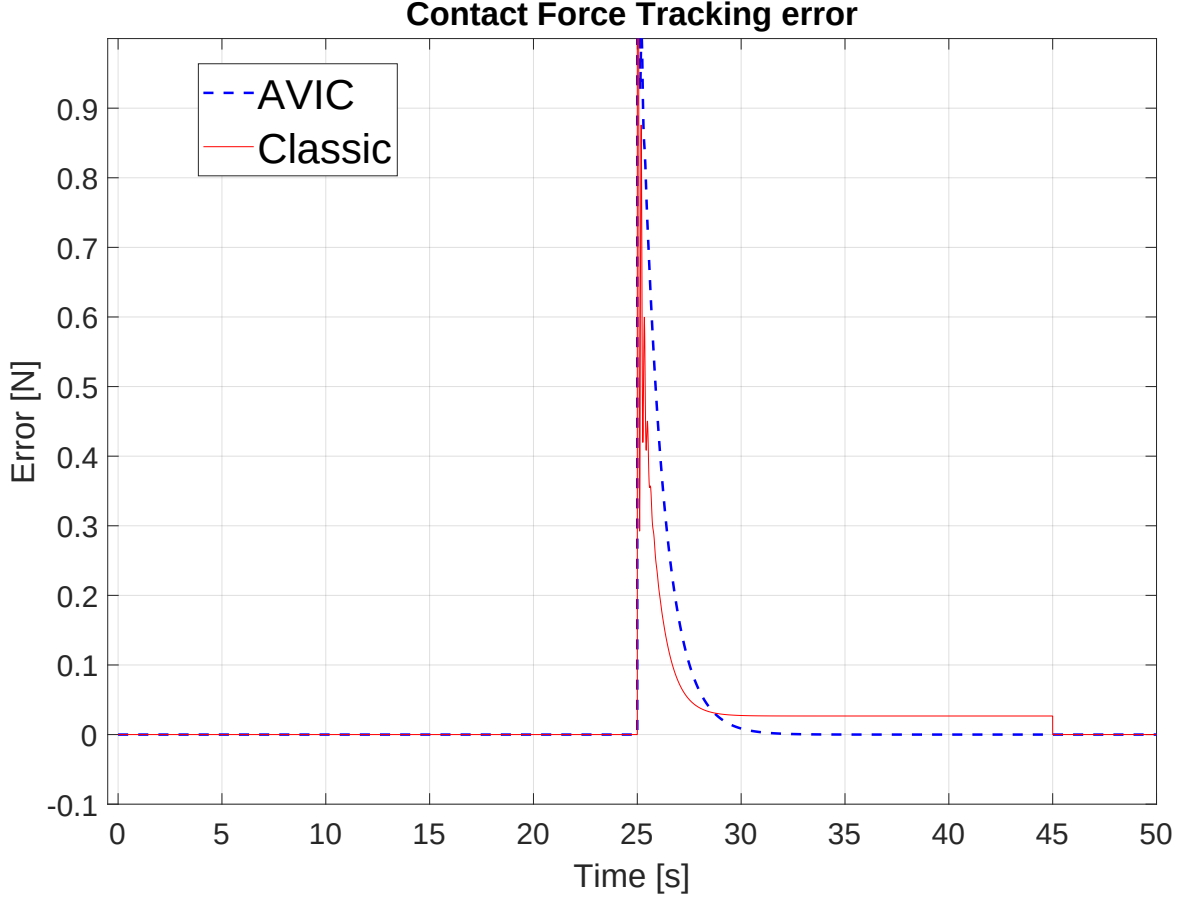


Figure 5.9: Contact force tracking errors - Scenario 2

From Figure 5.10, the control input torques τ and the resultant contact torques τ_{ext} are shown to be a little aggressive at time $t = 25\text{ s}$ and time $t = 45\text{ s}$. At time $t = 0\text{ s}$, the control input torques can be said to be aggressive. In practical terms, the actuators of the manipulator can still deliver these torques to regulate the end-effector during such contact inspection.

The third scenario is about tracking a desired force that increases at a rate of 0.4 N/s from time $t = 15\text{ s}$ to time $t = 25\text{ s}$ reaching force magnitude of 4 N and maintaining it upto time $t = 45\text{ s}$ when physical interaction stops. During this force tracking, the end-effector moves from initial position to final position following the desired trajectory $p_2(t)$ in Equation (5.2) and is plotted in Figure 5.11. The end-effector changes position between time $t = 20\text{ s}$ and $t = 28\text{ s}$. The final velocity for $p_2(t)$ was set to 0.001 m/s during trajectory generation, which means the end-effector will reach the final position moving at 1 mm/s then stops.

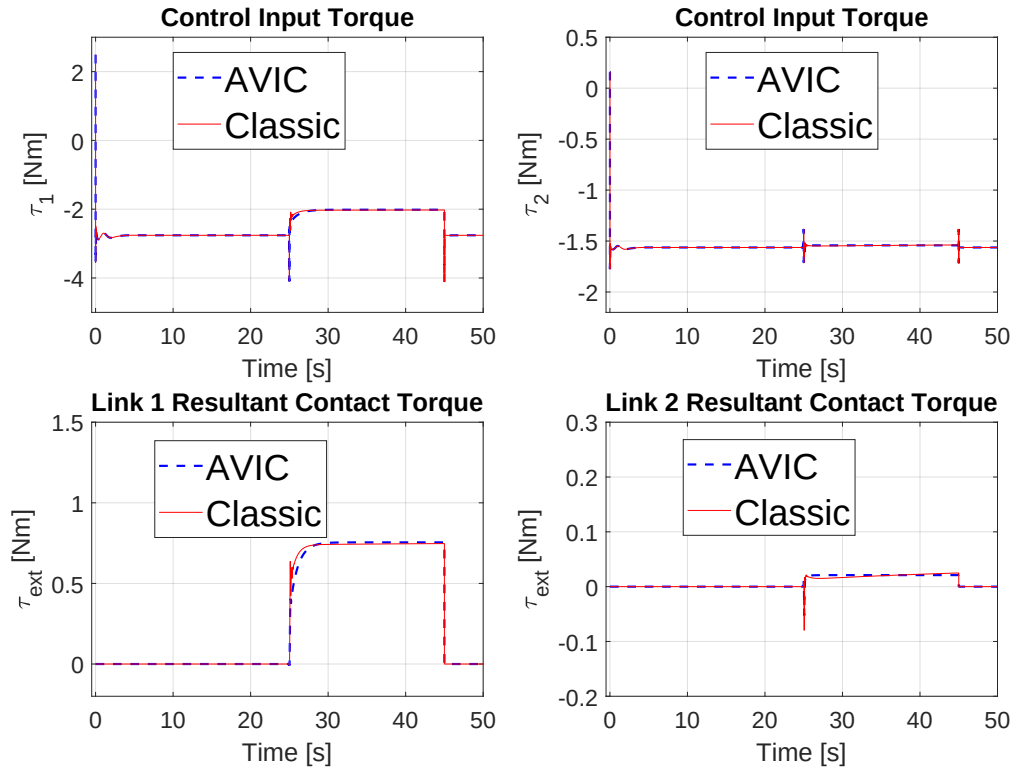


Figure 5.10: Manipulator joint inputs and contact torques - Scenario 2

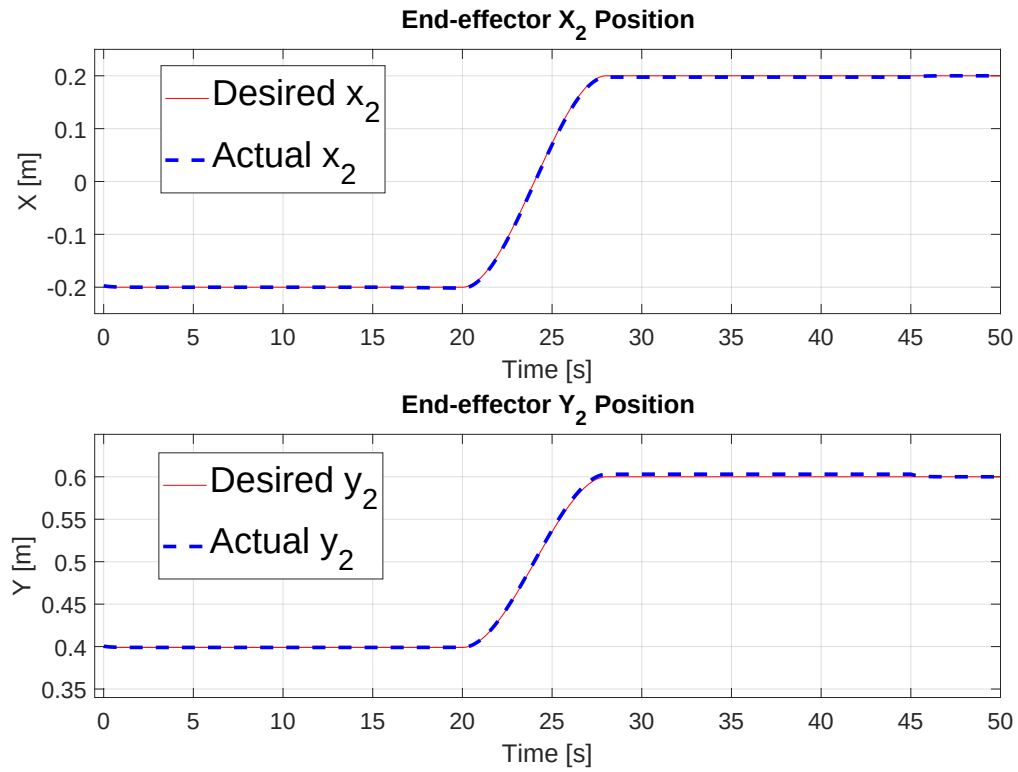


Figure 5.11: End-effector position tracking - Scenario 3

Figure 5.12 and Figure 5.13 compare the AVIC to the classic strategy with respect to joint angle tracking errors and the contact force tracking, respectively.

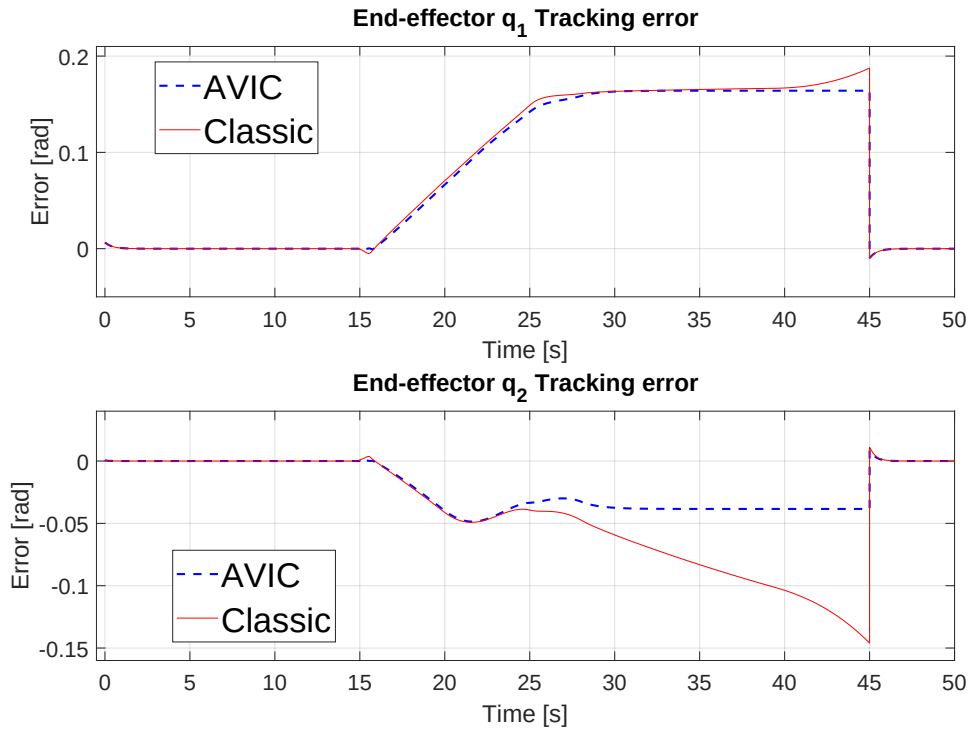


Figure 5.12: Manipulator joint angle tracking errors - Scenario 3

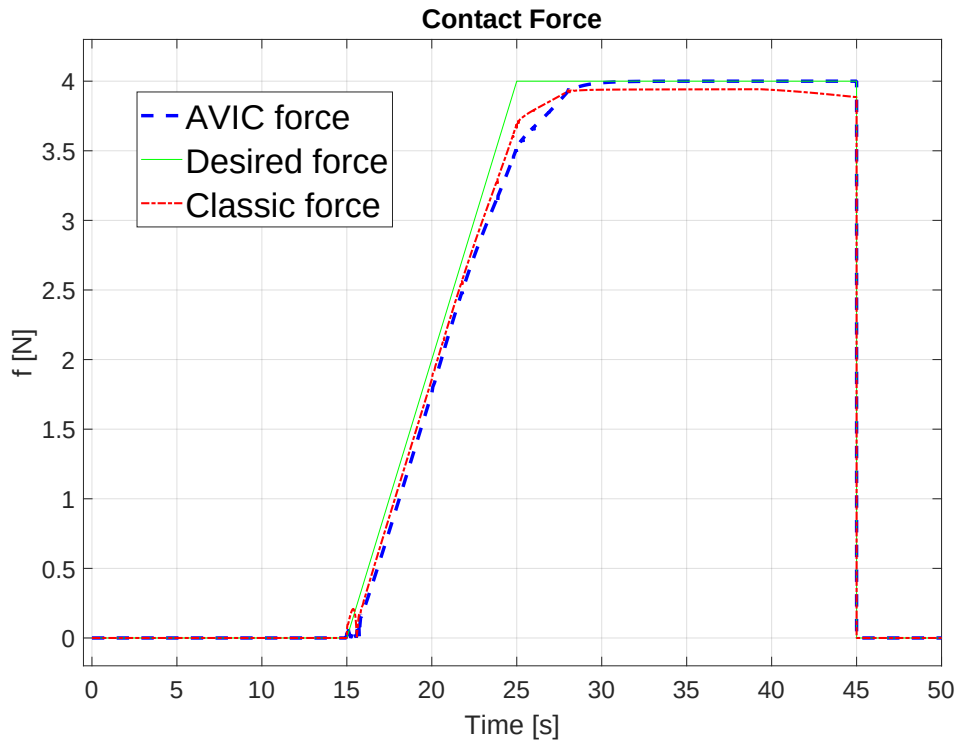


Figure 5.13: Contact force tracking between AVIC and classic strategy - Scenario 3

In this scenario, the damping coefficient b was not updated leaving the AVIC strategy to adapt and improve the force tracking error. As seen in Figure 5.13, force tracking can be attained while the AVIC adjust accordingly to reach zero steady state in the constant force region.

In Figure 5.14, the spikes just after time $t = 20\text{ s}$ can be seen for both strategies, however the AVIC strategy have high tracking error when the force is still increasing but quickly reduces from $t = 25\text{ s}$ to zero error in 5 s . On the hand, the classic strategy does try to track but there exist a steady state error that seems to be increasing from $t = 40\text{ s}$. This was expected as discussed in Equation (4.35) regarding the classic strategy requiring precise knowledge of the environmental properties. Retuning impedance parameters, namely damping coefficient b and desired stiffness k could improve the force tracking for the classic strategy. For AVIC strategy only guess the initial damping coefficient b and the adaptive law will adjust over time. It is worth stating that, the quadrotor which is the base of the manipulator is moving from $t = 11\text{ s}$ to $t = 18\text{ s}$ and from time $t = 19\text{ s}$ to $t = 24\text{ s}$, corresponding to quadrotor change in the desired x position x_{ref} and the desired y position y_{ref} , respectively. The values of these transitions can be found in Table 5.2, specifically from way-point 4 to way-point 7. The cubic polynomial trajectory was used to reduce discontinuities during line segments transition.

The control input torques τ and the resultant contact torques τ_{ext} are shown in Figure 5.15 indicating similar aggressiveness as in the previous scenario.

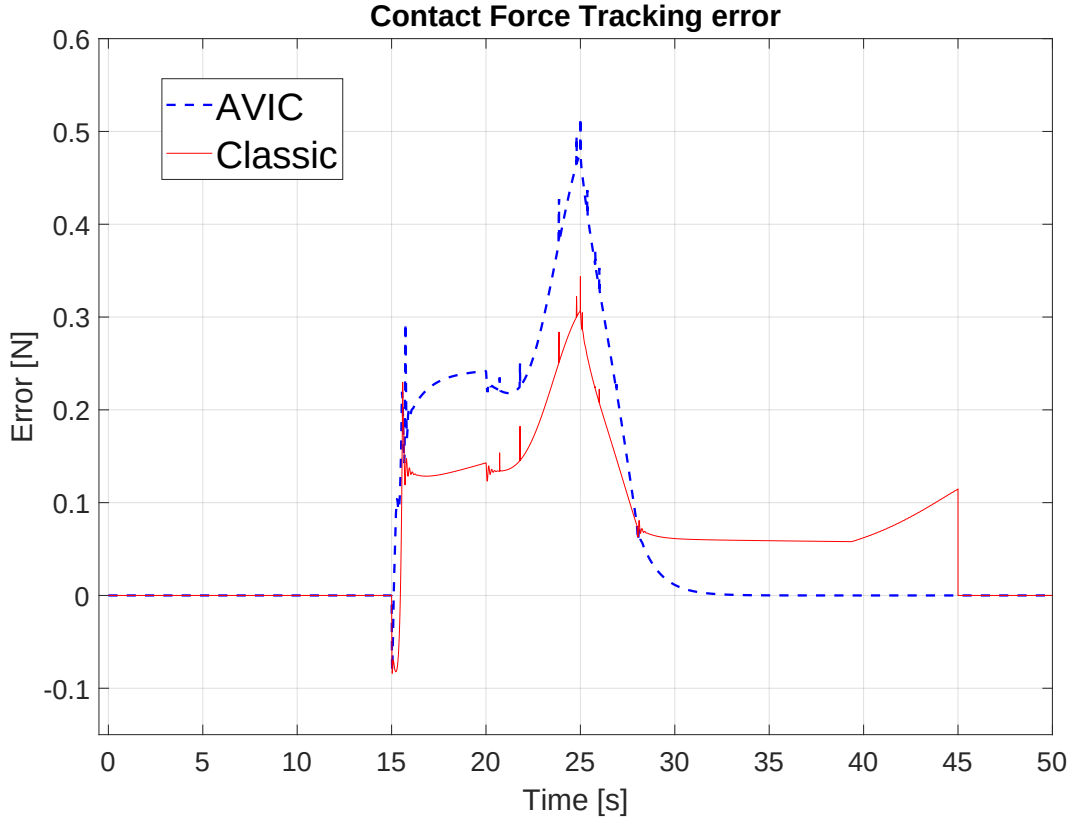


Figure 5.14: Contact force tracking errors - Scenario 3

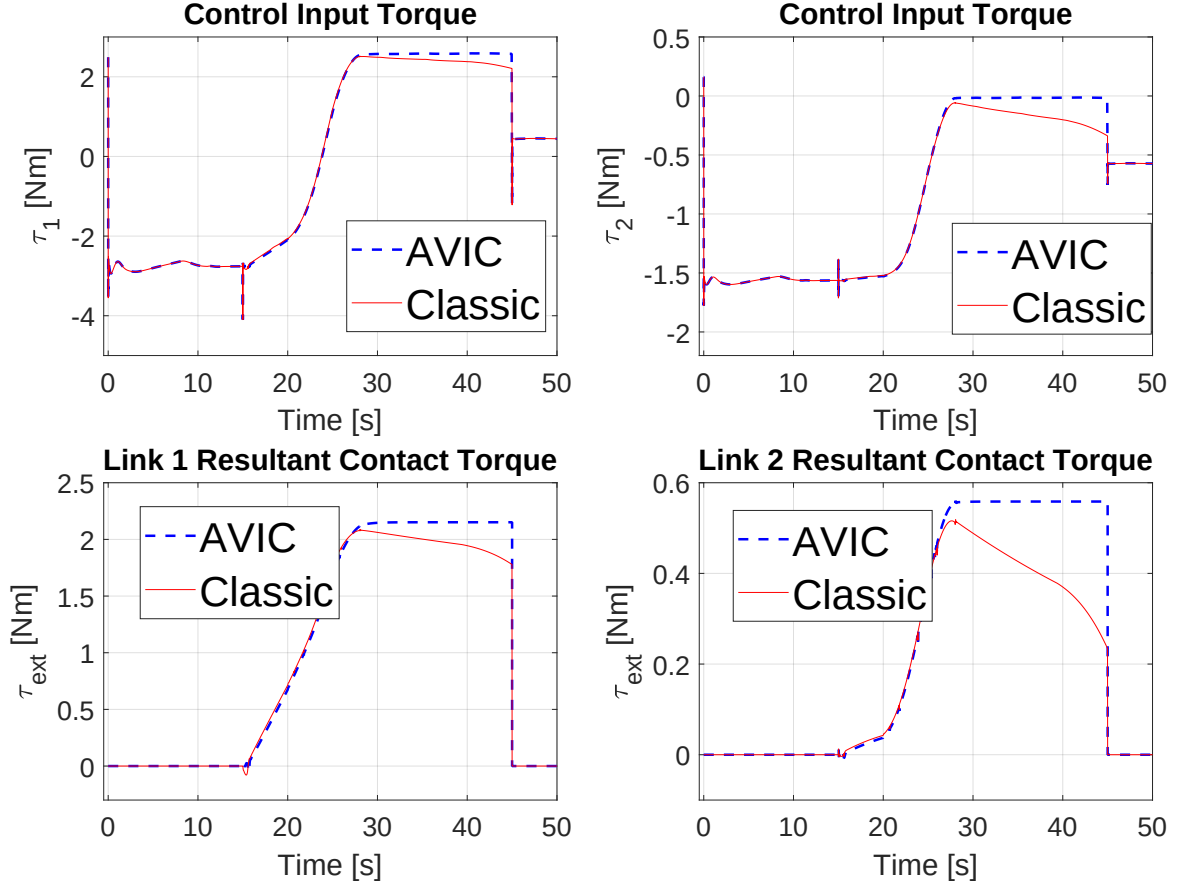


Figure 5.15: Manipulator joint inputs and contact torques - Scenario 3

Finally, the last scenario test the constant force tracking performed in the second scenario while the environmental location uncertainty is 5 % instead of 0.1 % as in the last two scenarios. The desired end-effector trajectory $p_2(t)$ as in the previous simulation scenario.

The force tracking and the resulting tracking error for both strategies are depicted in Figure 5.16 and Figure 5.17, respectively. Force tracking is achieved despite the 5 % uncertainty of the environmental location. But at time $t = 25\text{ s}$ the forces are not so smooth yet, are smooth thereafter. The initial damping coefficient was guessed to be $b = 12\text{ Ns/m}$ for the AVIC and the classic strategy to obtain such contact force tracking. The steady state error has definitely increased to 3 % for the classic strategy compared to less than 2 % attained in the second simulation scenario.

Figure 5.18 shows the joint angle tracking errors for the manipulator while Figure 5.19 depicts the control input torques and the contact torques on the manipulator. It can be observed that the desired end-effector trajectory $p_2(t)$ get modified from time $t = 25\text{ s}$ in both figures but the control input torques from Figure 5.19 indicate that they do start to regulate the movement of the manipulator from time $t = 20\text{ s}$ following $p_2(t)$

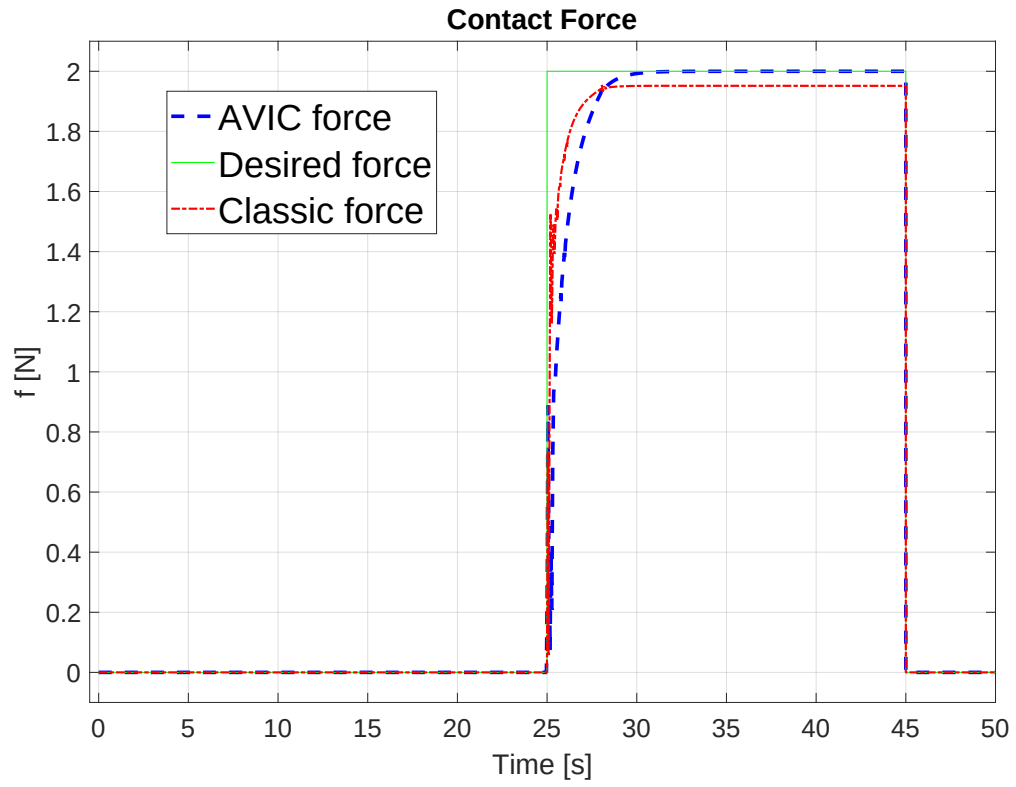


Figure 5.16: Contact force tracking between AVIC and classic strategy - Scenario 4

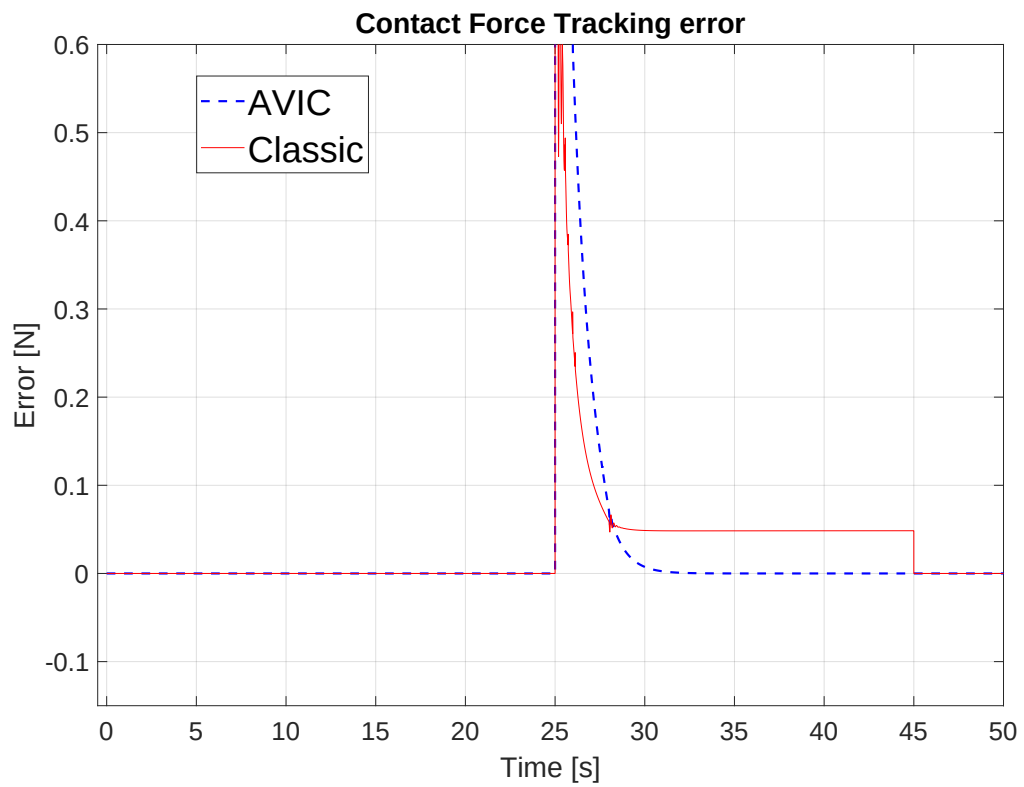


Figure 5.17: Contact force tracking errors - Scenario 4

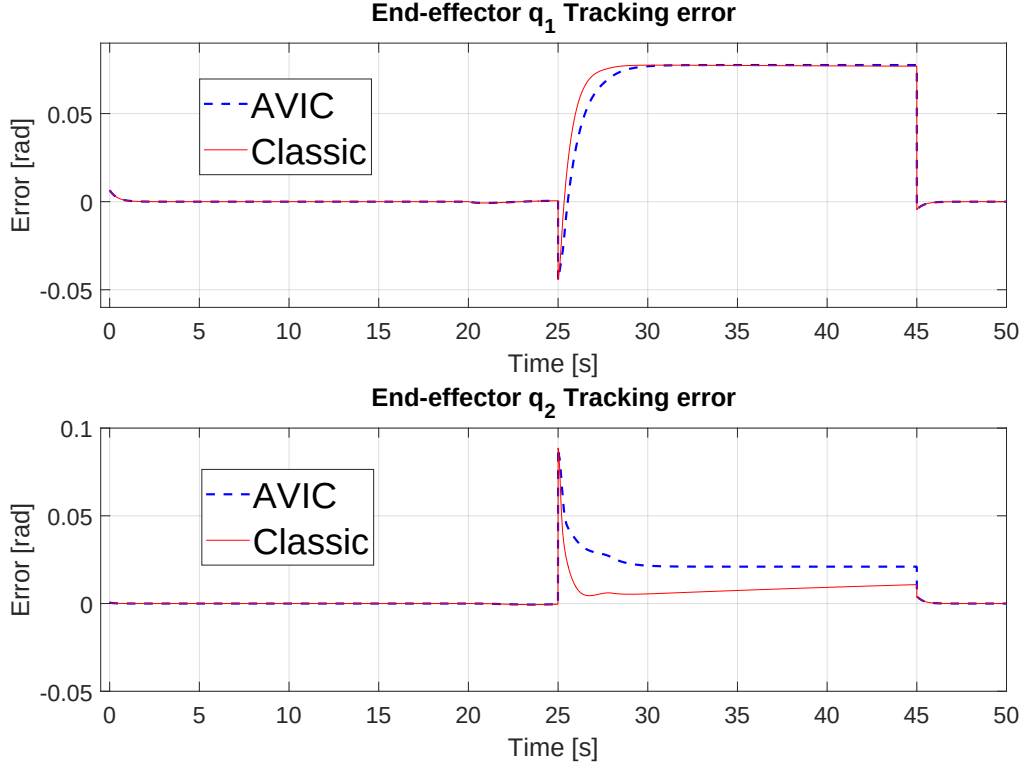


Figure 5.18: Manipulator joint angle tracking errors - Scenario 4

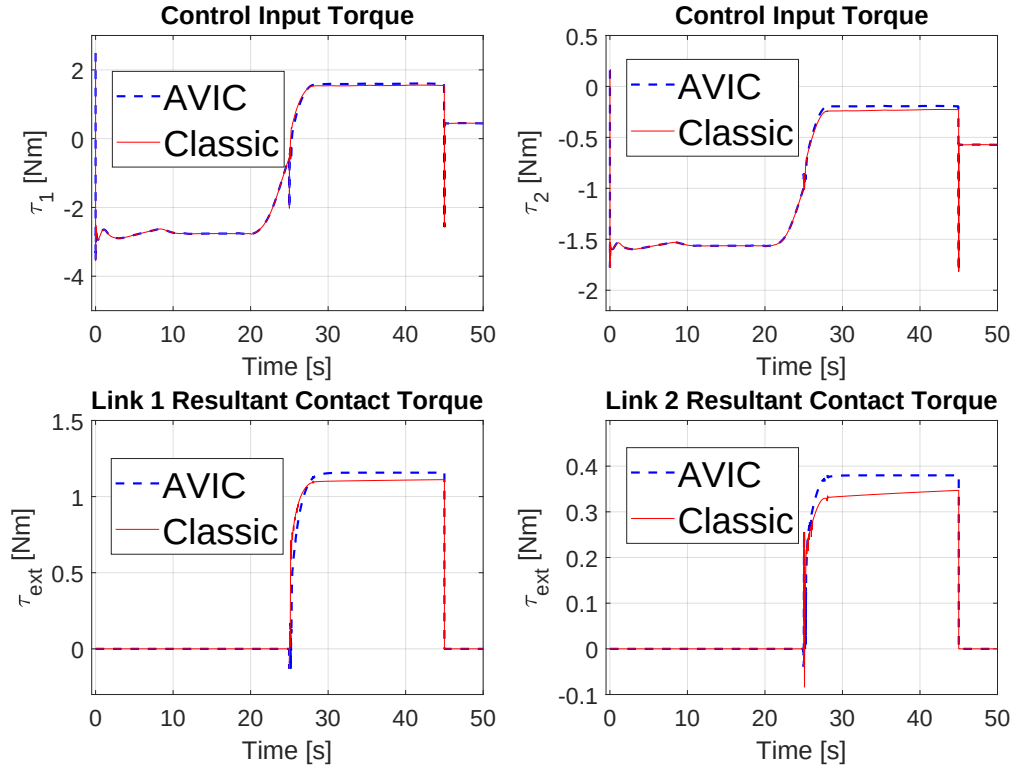


Figure 5.19: Manipulator joint inputs and contact torques - Scenario 4

The AVIC strategy outperformed the classic strategy as demonstrated through the three different physical interaction scenarios that can be utilised in a real-world applications. The environmental forces and moments exerted on the manipulator's end-effector are also sensed by the quadrotor dynamics according to Equations (3.28) and (3.29) and these effects will be assessed in the next section.

5.3 Quadrotor results

The quadrotor as an under-actuated system, has four inputs versus six state variables to be controlled. It is common to control the movement in y and x-directions through controlling roll (ϕ) and pitch (θ) rotations respectively. The quadrotor model was linearised about the equilibrium point for hovering configuration. Then the integral LQR controller developed for quadrotor dynamics was utilised for the altitude motion, for the pitch and roll motions, and for the yaw motion. The penalty amounts for $\mathbf{Q}_\zeta$ and $\mathbf{R}$ matrices were obtained through trial and error method. Then these matrices were used to determine the appropriate gain matrix $\mathcal{K}$ which minimises the augmented performance function J in Equation (4.23). The matrix $\mathcal{K}$ consists of 12 proportional gains Kp_ξ for state variables and 4 integral gains Ki_Y for measured outputs. Table 5.4 shows the gains which were obtained through the LQR function command in Matlab that solves the algebraic Riccati equation. The proportional gains are comparable with the gains obtained by Cariño et al., [16].

First scenario check set-point tracking capabilities of the integral LQR controller without physical interaction. The quadrotor was set to track for desired x position $x_{ref} = 1\text{ m}$, desired y position $y_{ref} = 0.5\text{ m}$, desired z position $z_{ref} = 3\text{ m}$, desired yaw rate $\omega_{zref} = 0\text{ rad/s}$. Figure 5.20 depicts the quadrotor trajectory tracking with a 3D plot for the quadrotor position while Figure 5.21 shows the resulting tracking errors. It can be seen that the controller takes about 5 s to reach steady state of zero.

Table 5.4: Quadrotor Control Parameters

Parameter	Value	Parameter	Value
Kp_z	17.4	Kp_x	45.3
Ki_z	-14.1	Ki_x	-31.6
$Kp_{\dot{z}}$	9.3	$Kp_{\dot{x}}$	16.7
Kp_y	-45.3	Kp_{q_j}	64.7
Ki_y	-31.6	Kp_{ω_y}	3.2
$Kp_{\dot{y}}$	-16.7	Kp_{q_k}	24.6
Kp_{q_i}	64.7	Kp_{ω_z}	3.19
Kp_{ω_x}	3.2	Ki_{ω_z}	-31.6

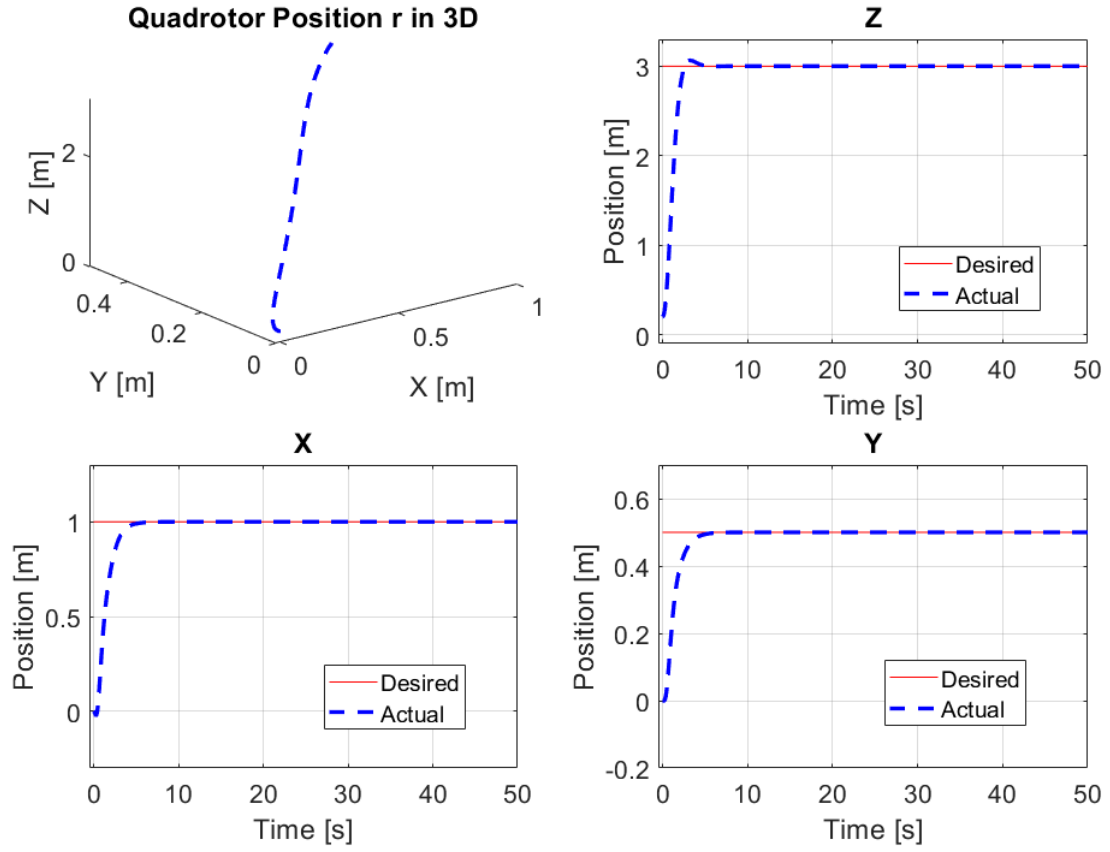


Figure 5.20: Quadrotor trajectory tracking without physical interaction - Scenario 1

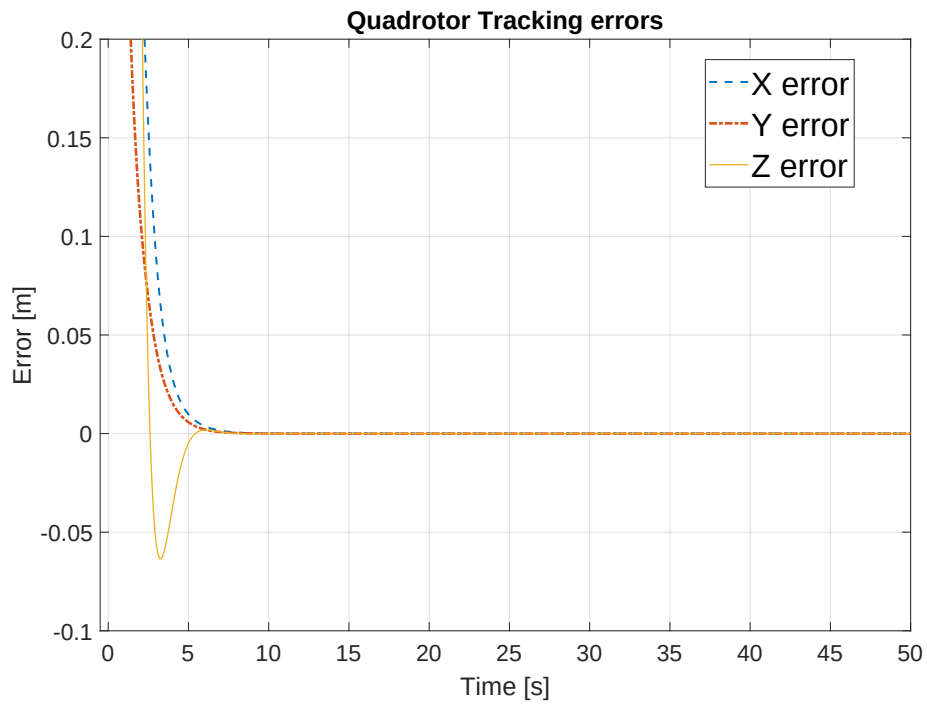


Figure 5.21: Quadrotor position tracking errors - Scenario 1

The rotational motions can be seen in Figure 5.22, that the integral LQR is can to stabilise the quadrotor in about 3 s and the desired yaw rate can be tracked even faster. The fact that a quadrotor is under-actuated can be observed by the movement y and x directions influencing the roll (ω_x) and pitch (ω_y), respectively.

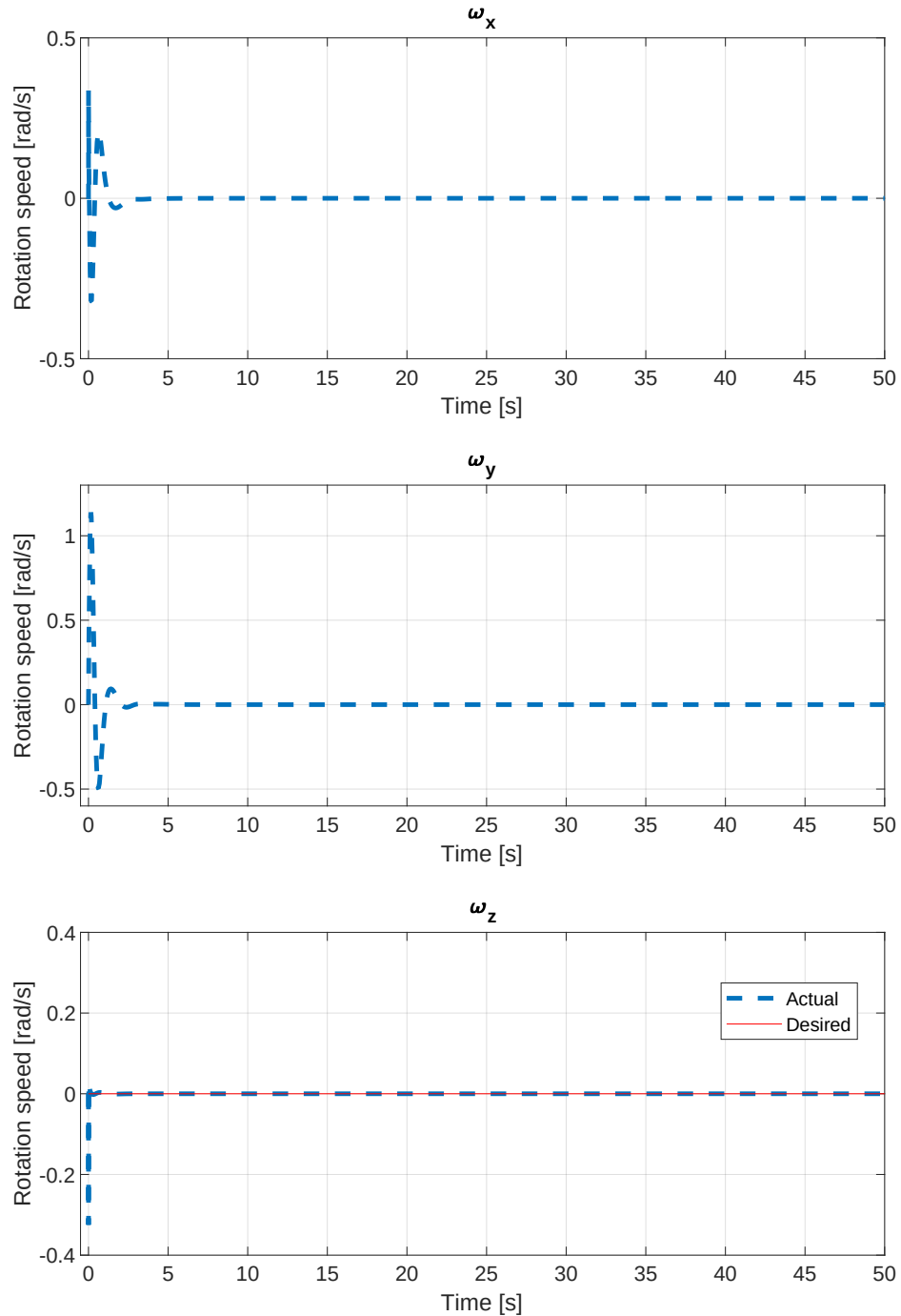


Figure 5.22: Quadrotor's angular velocities - Scenario 1

The quadrotor's control inputs, namely the net thrust U_1 for altitude (z) motion, roll torque U_2 for y position, pitch torque U_3 for x position and the yaw torque U_4 can be observed in Figure 5.23. The control efforts for each inputs are well within physical limitations for many actuators.

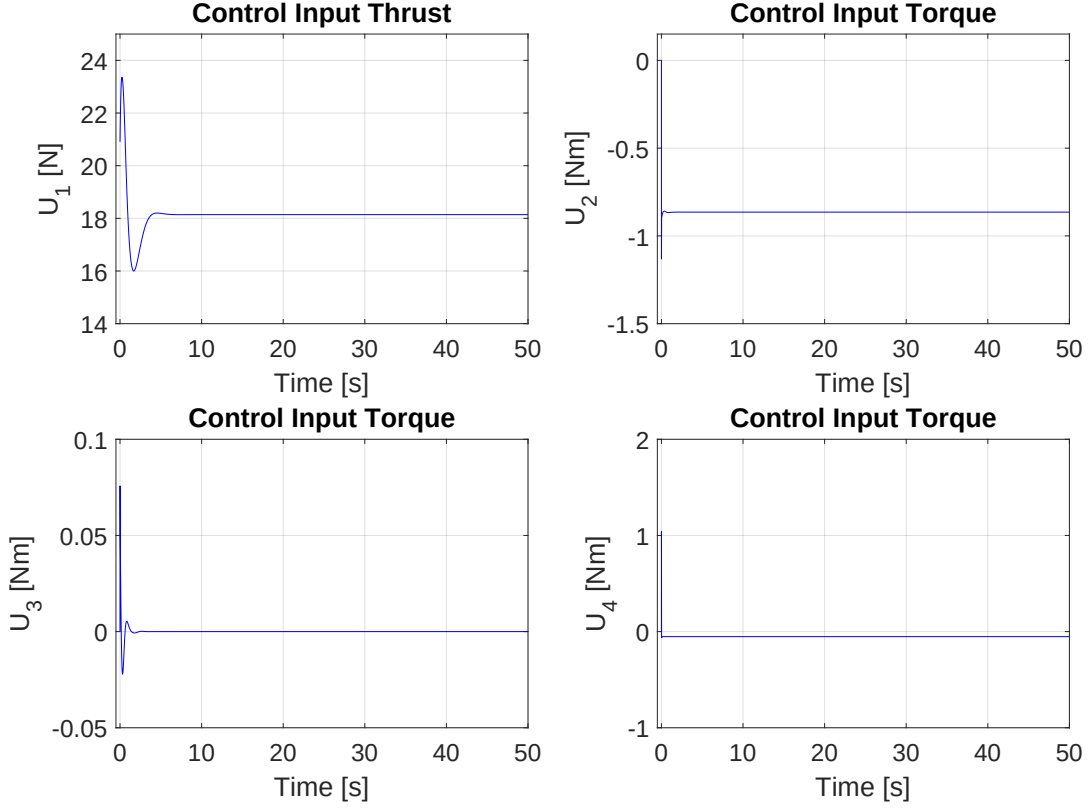


Figure 5.23: Quadrotor's control inputs from the integral LQR - Scenario 1

The second scenario considers physical interaction when the end-effector performs the set-point tracking. Here the quadrotor tracks the desired helix trajectory given in Equation (5.1) and the position trajectory tracking is shown in Figure 5.24. The resulting tracking errors can be seen from Figure 5.25 with steady state error in the altitude motion (Z -axis) this could be attributed to the fact that the designed LQR was for set-point tracking. The LQR can be enhanced for time-varying set-point tracking with a modified control input using the Moore-Penrose pseudo-inversion [38]. Since the error appears to remain constant through out the simulation, the 2-DOF manipulator can compensate the error at the end-effector position. Therefore, the error is considered to be acceptable which indicates good tracking capability of the integral LQR a helix trajectory. It can be seen that the X and Y axes have a delay which could be attributed to the pitch (ω_y) and roll (ω_x) settling time, respectively. Choutri et al.,[22] theorised that such a delay could be due to the altitude settling time.

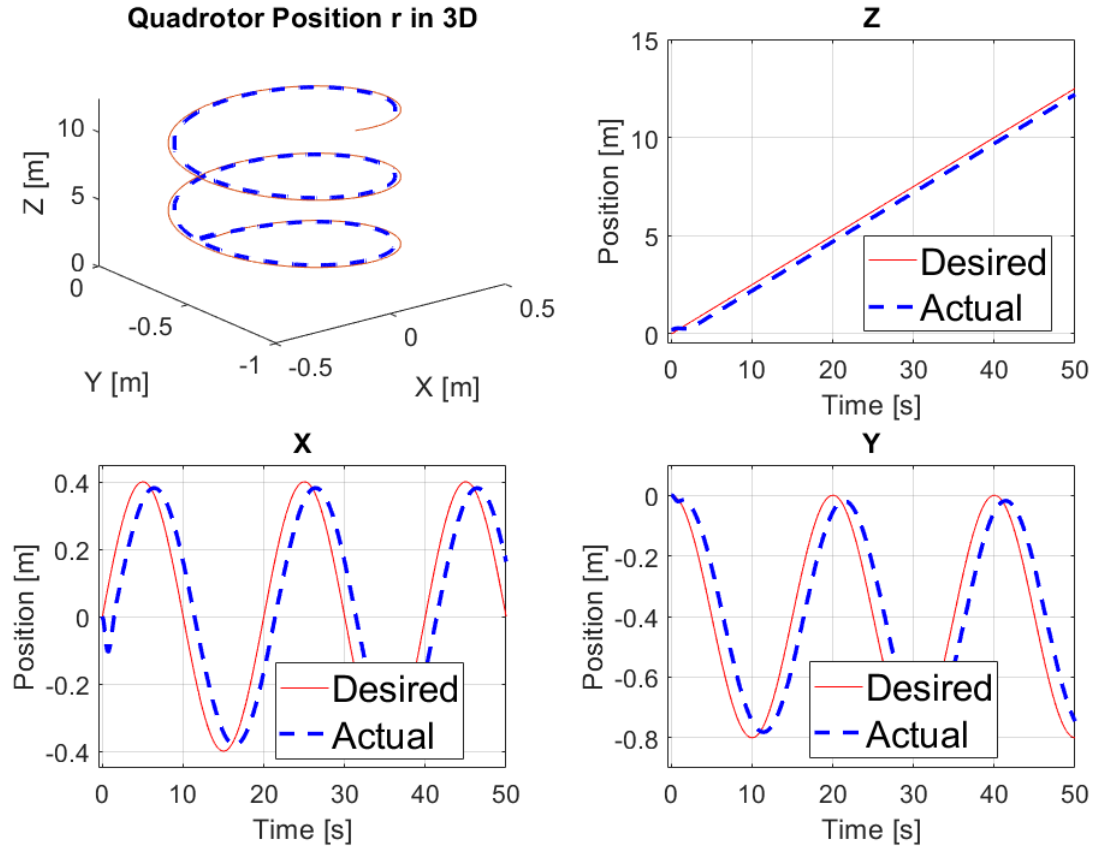


Figure 5.24: Quadrotor trajectory tracking during physical interaction - Scenario 2

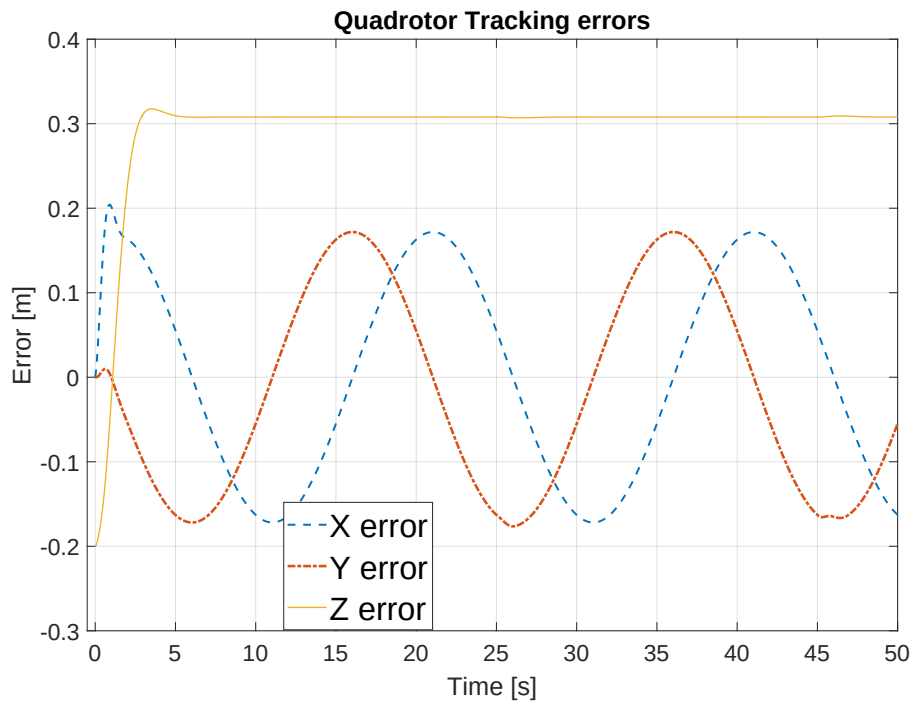


Figure 5.25: Quadrotor position tracking errors - Scenario 2

Figure 5.26 and Figure 5.27 depicts the rotational motions and quadrotor's control inputs, respectively.

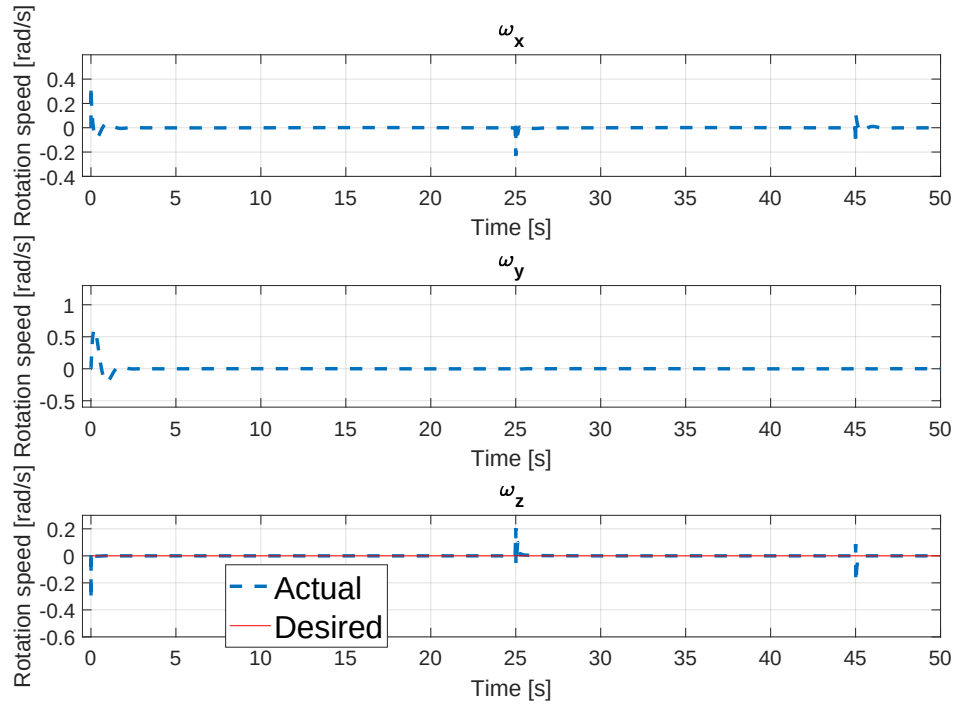


Figure 5.26: Quadrotor's angular velocities - Scenario 2

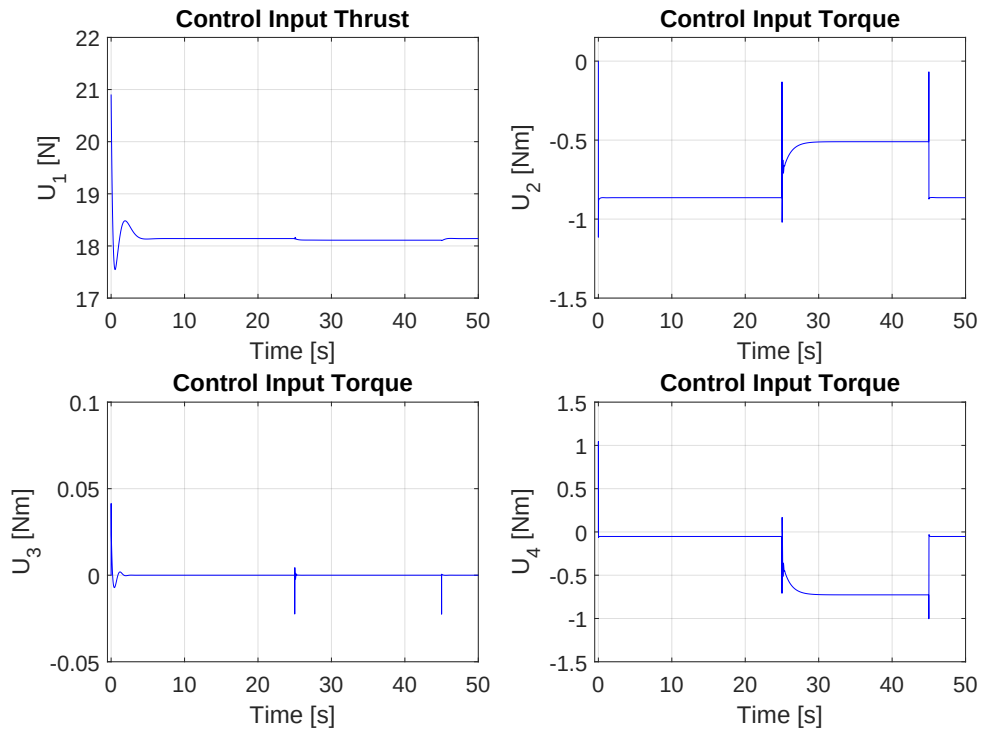


Figure 5.27: Quadrotor's control inputs from the integral LQR - Scenario 2

The physical interaction can be observed in both figures 5.26 and 5.27 from time $t = 25\text{ s}$ to time $t = 45\text{ s}$ with the control input torques having to regulate a bit aggressive for the environmental forces and moment and some velocity spikes in the roll (ω_x) and yaw (ω_z) motions. Figure 5.28 compares resulting Euler angles when the physical interaction is regulated by the AVCI and the classic strategy. The quadrotor's behaviour is controlled by the integral LQR even when the force tracking is performed by the classic impedance control. The initial value of the yaw angle ψ was arbitrarily chosen to 1.57 rad which is approximately 90° while the pitch angle θ and the roll angle ϕ were set to zero initial conditions. The Euler angles are the same for both control force tracking strategies. The yaw angle ψ is affected the most by the physical interaction.

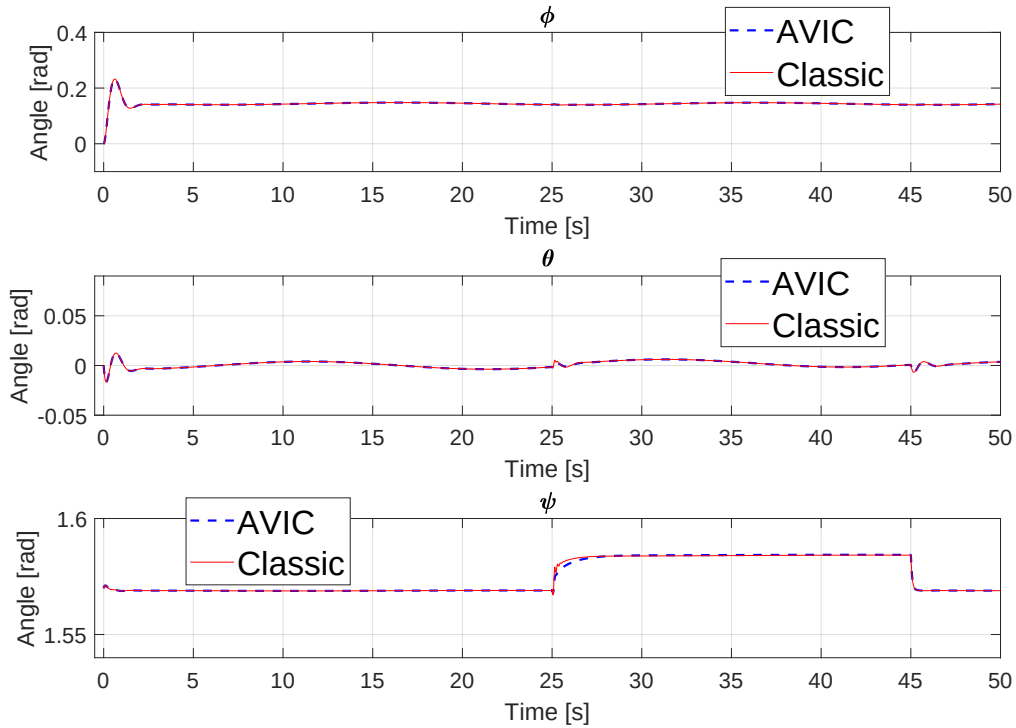


Figure 5.28: Comparing the quadrotor's resultant Euler angles between AVIC and classic strategy - Scenario 2

The third scenario has the quadrotor set to track the desired trajectory computed from cubic polynomials with the way-points given in Table 5.2. While the end-effector following the desired trajectory $p_2(t)$ calculated according to Equation (5.2) during physical interaction with the desired contact force increasing at a rate of 0.4 N/s . The quadrotor trajectory tracking is depicted in Figure 5.29 and it can be seen that there is an error when tracking the desired slopes. A clear representation of such errors can be seen in Figure 5.30, where the position tracking errors exceed 0.7 m . The slopes between way-point 2 and way-point 7 with high errors can be compared to the slopes between way-point 9 and way-point 11 with tracking error less than 0.2 m .

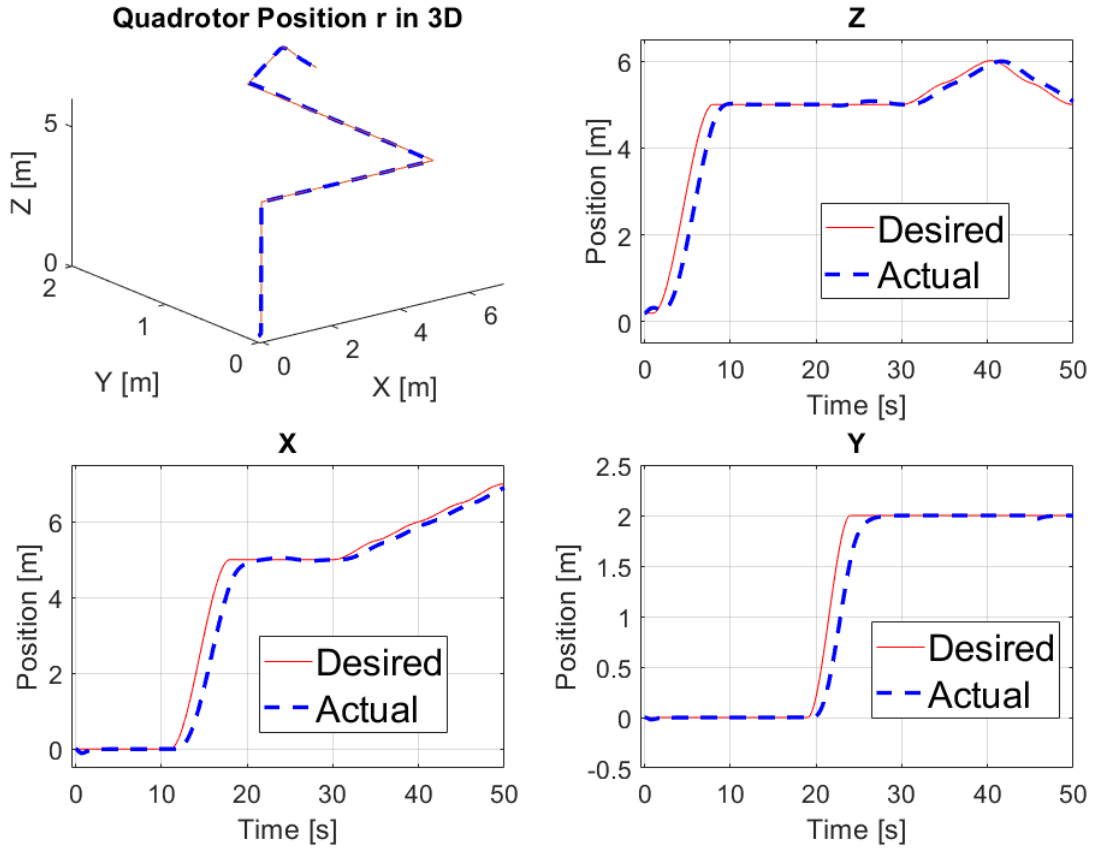


Figure 5.29: Quadrotor trajectory tracking during physical interaction - Scenario 3

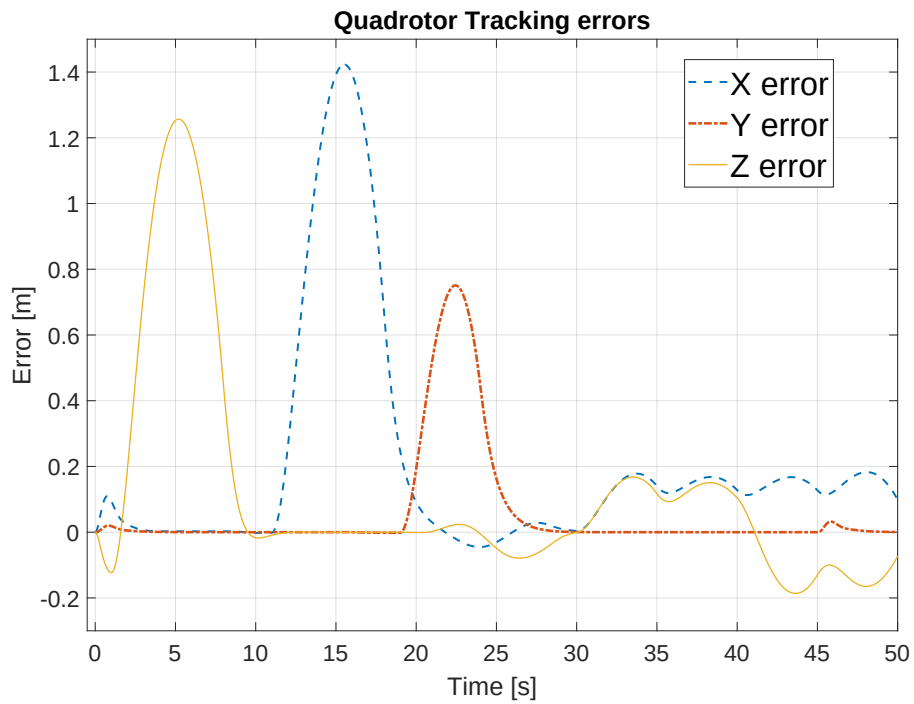


Figure 5.30: Quadrotor position tracking errors - Scenario 3

This is because the desired trajectory had a zero velocity boundary for the former slopes while the latter slopes' velocity boundary was set at 0.05 m/s between way-point 9 and way-point 11. This velocity boundary plays a vital role in trajectory generation with multiple way-points for example, if velocity boundary is set to zero for a given time interval, the necessary velocity will be used between the initial and final position. Figure 5.31 clearly shows that the translational velocities reached 1 m/s for Z and X axes at time $t = 6 \text{ s}$ and at time $t = 16 \text{ s}$, respectively. The Y velocity reached 0.45 m/s at at time $t = 23 \text{ s}$. Despite the high velocities from the desired trajectory, the integral LQR is capable to track the trajectory. To improve the tracking error, the way-points can be modified together with the velocity boundary for a particular scenario. This scenario opened an opportunity to discuss briefly, the problem of trajectory generation.

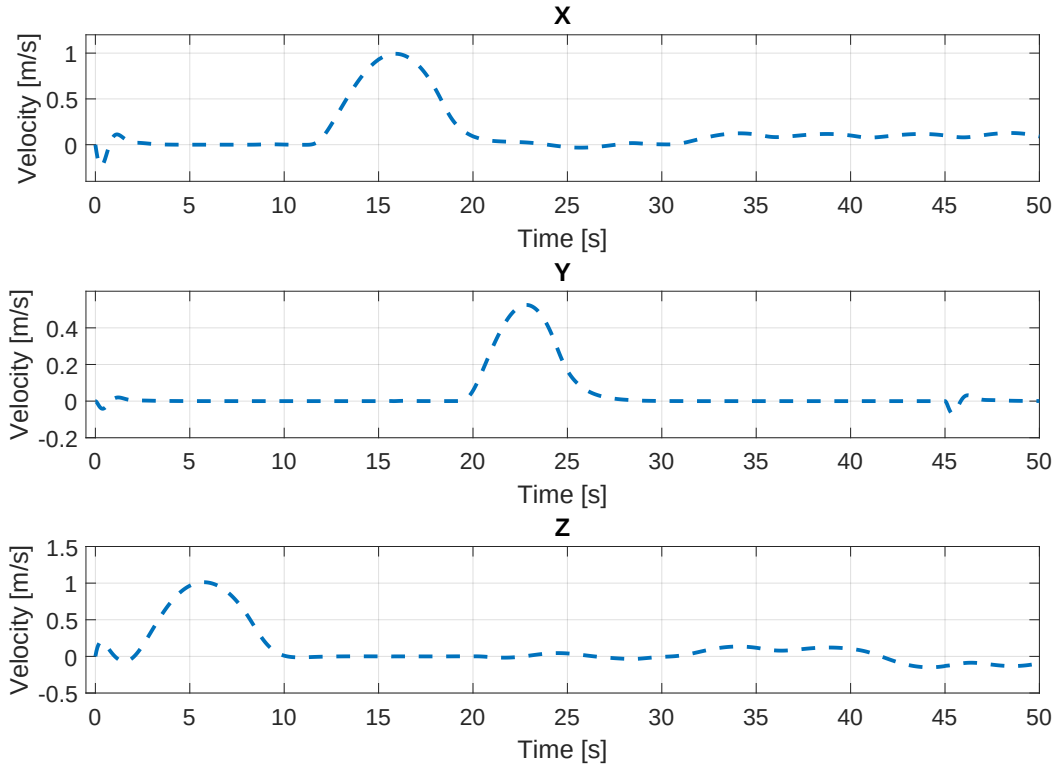


Figure 5.31: Quadrotor's translational velocities - Scenario 3

The effect of physical interaction can be seen in Figure 5.32 for angular velocities and Figure 5.33 for control inputs, along the roll motion (ω_x and U_2) and the yaw motion (ω_z and U_4). The pitch torque U_3 input remains steady during the contact inspection, only spikes at the start ($= 15 \text{ s}$) and end ($= 45 \text{ s}$) of contact. Finally, the resulting Euler angles due to using different force tracking strategies in the third scenario are shown in Figure 5.34. As seen in Figure 5.13, the classic strategy has the steady state error that seemed to increase. Conversely, the quadrotor yaw angle ψ appears to decrease just after 25 s while the AVIC reached zero steady state error and the resulting yaw angle ψ reflect the stable orientation of the quadrotor.

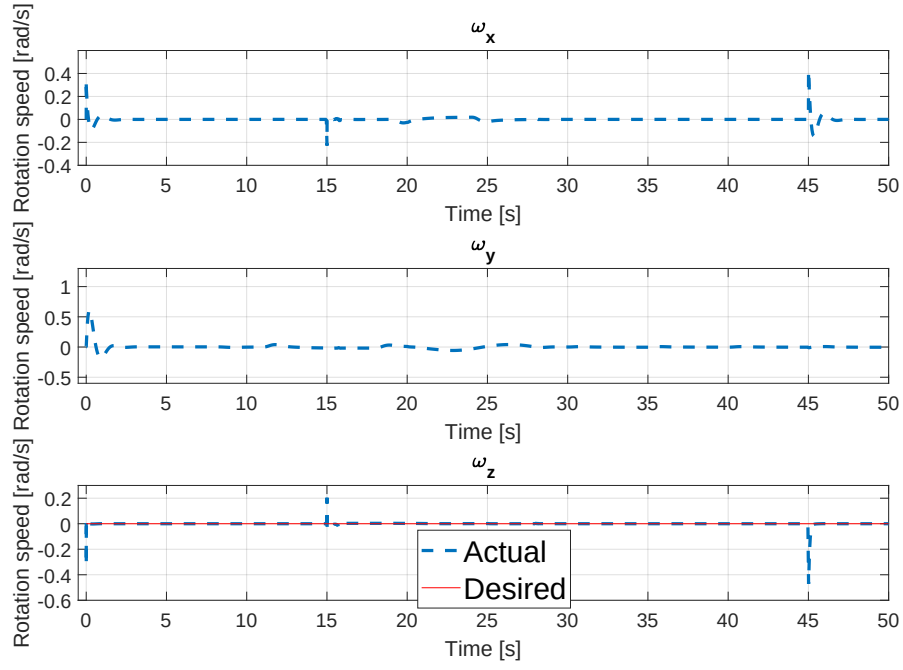


Figure 5.32: Quadrotor's angular velocities - Scenario 3

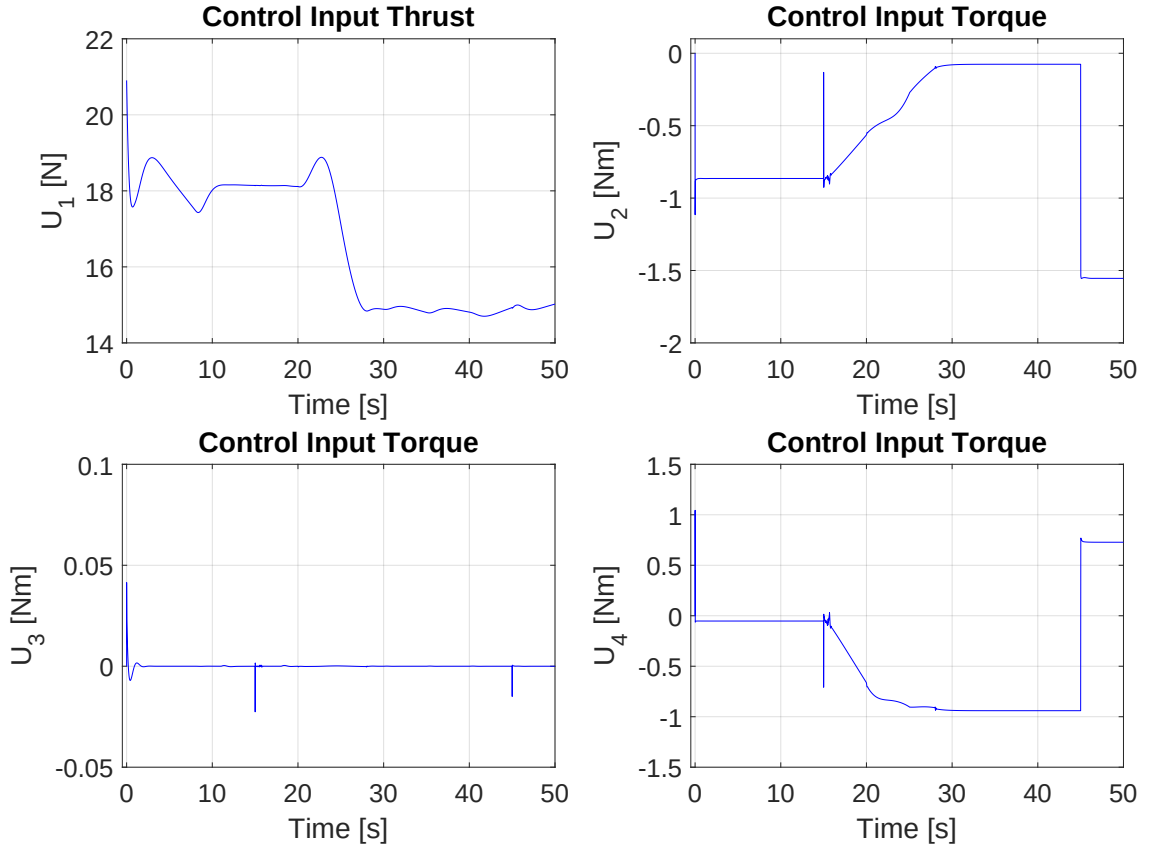


Figure 5.33: Quadrotor's control inputs from the integral LQR - Scenario 3

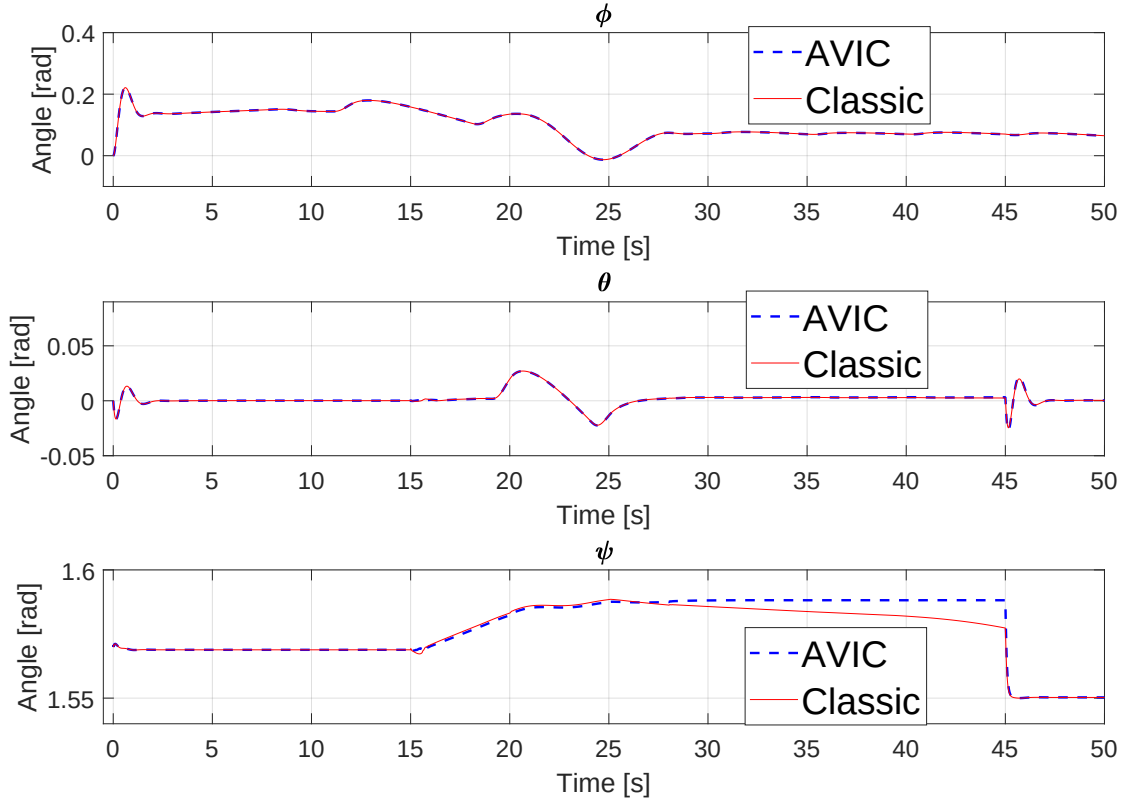


Figure 5.34: Comparing the quadrotor's resultant Euler angles between AVIC and classic strategy - Scenario 3

In the last simulation scenario, the quadrotor performs the same tracking task as in the third scenario. The position tracking and its error were also similar to Figure 5.29 and Figure 5.30, respectively. Therefore, the analysis of the physical interaction effects similar with the second scenario except that the environmental location uncertainty is increased to 5% while exerting contact force of 2 N. The physical interaction is evident with spikes on roll (ω_x) and yaw (ω_z) motions at time $t = 25$ s and $t = 45$ s as seen in Figure 5.35. The physical interaction requires an aggressive control inputs in the roll torque U_2 and yaw torque U_4 as depicted in Figure 5.36 while the thrust input U_1 smoothly reduces to about 15 N. Lastly, the quadrotor's Euler angles are shown in Figure 5.37 and as expected, the resulting Euler angles follow the pattern from the second scenario during physical interaction.

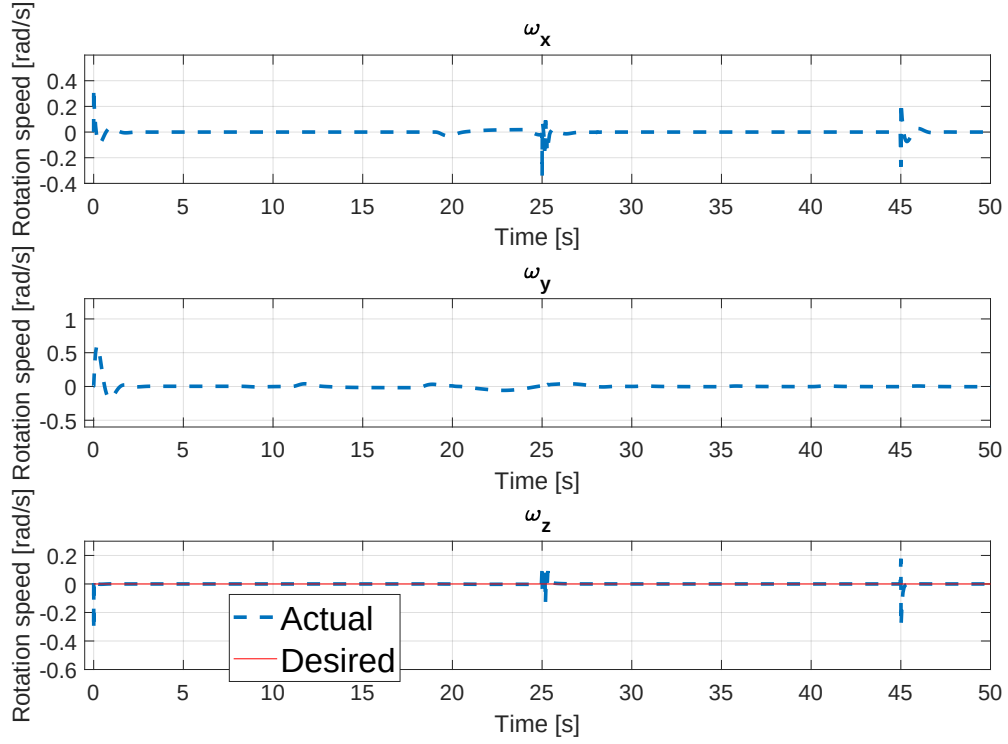


Figure 5.35: Quadrotor's angular velocities - Scenario 4

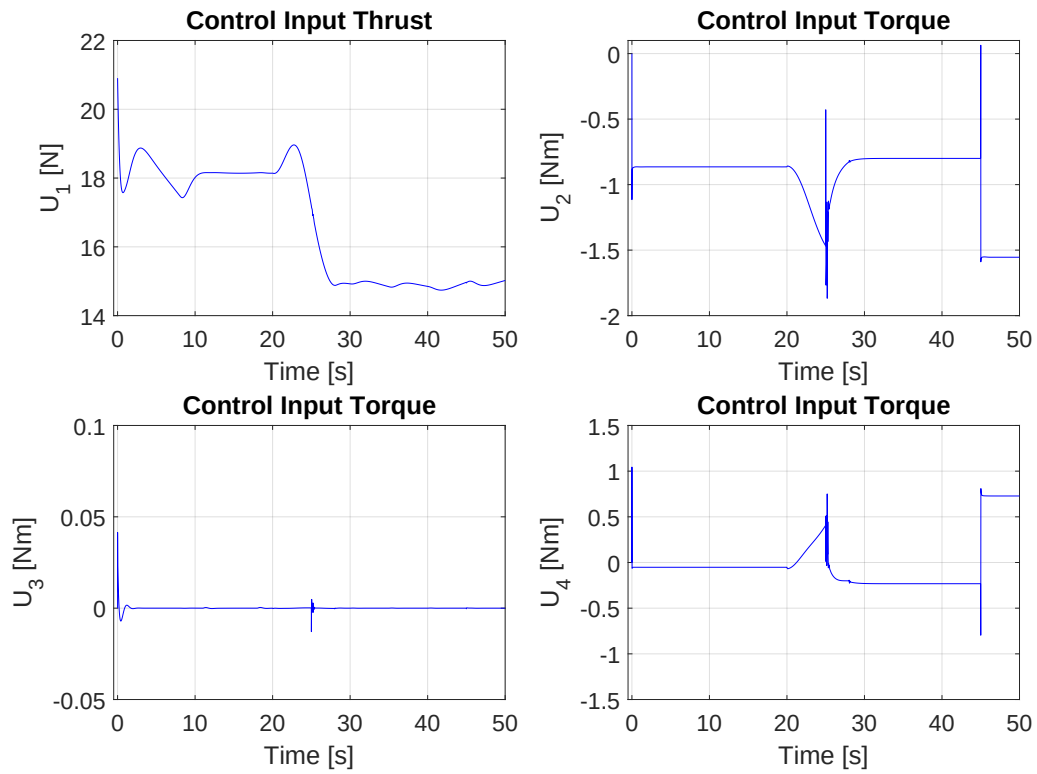


Figure 5.36: Quadrotor's control inputs from the integral LQR - Scenario 4

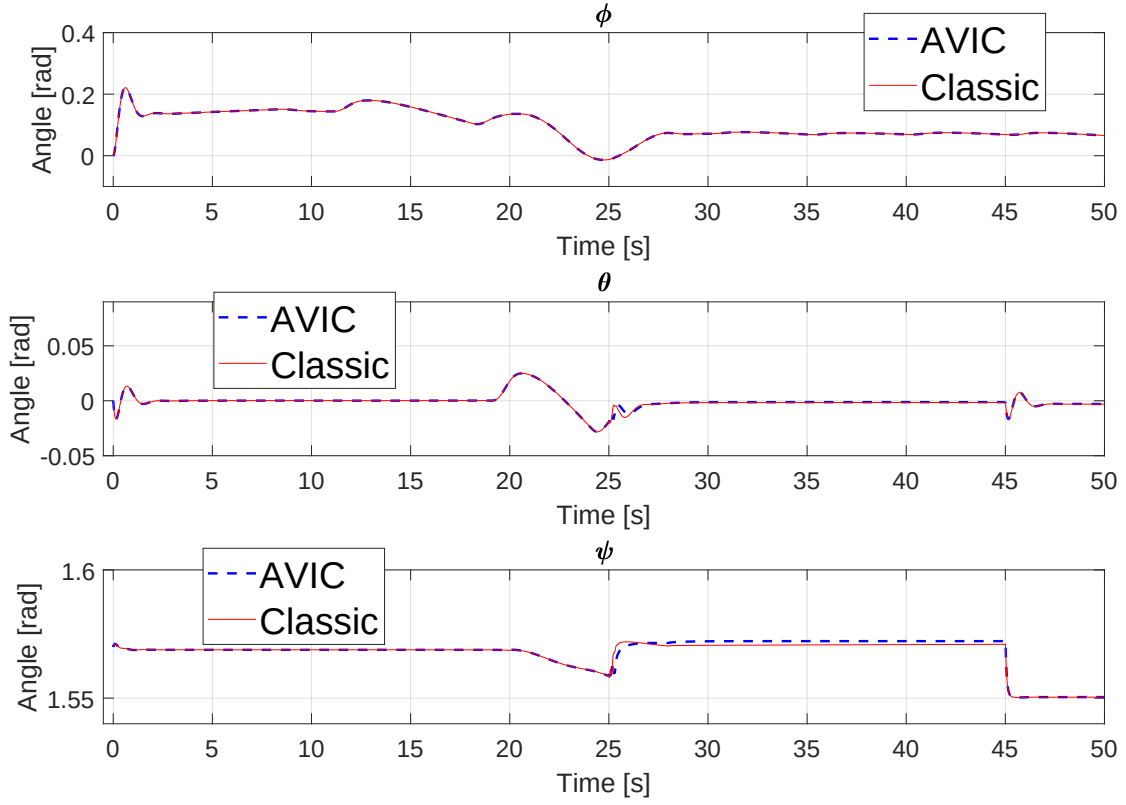


Figure 5.37: Comparing the quadrotor's resultant Euler angles between AVIC and classic strategy - Scenario 4

From the simulations performed, integral LQR is capable of regulating the quadrotor's six variables despite the quadrotor being an under-actuated system. This was achieved after the coupled system dynamics was linearised for hover mode. It was observed that there was steady state error when tracking linear trajectories since the developed control was for set-point tracking which was achieved.

5.4 Chapter summary

The AVIC strategy was implemented to achieve force tracking in contact inspection applications. The quadrotor's behaviour was controlled using the integral LQR. The aerial manipulator was simulated in the MATLAB/SIMULINK program. Four simulation scenarios were devised for the suitability of the AVIC strategy during contact inspection together with the quadrotor's control. First scenario had both manipulator and quadrotor achieving the set-point tracking without physical interaction. Then the quadrotor tracked the helix trajectory with an acceptable steady state error in the slope tracking during the second scenario. The error could be compensated by the 2-DOF manipulator, such that the end-effector position can be reached. Physical interaction for tracking a constant force was achieved by both strategies, although the classic strategy had non-zero steady state error. The third scenario, both subsystems tracked trajectories generated using the cubic polynomials with different velocity boundaries. Physical interaction increased the contact force from 2 N (second scenario) to 4 N at a rate of 0.4 N/s and the AVIC reached zero steady state in the constant force region. The last scenario was some-kind of combination between the second scenario and the third scenario, except that the environmental location uncertainty was increased to 5% . The AVIC strategy achieved force tracking with zero steady state error. All the control efforts necessary for stability were well within physical limitations for many actuators.

In all the simulations, the Euler angles are not used as an input to any function. Rather only the unit quaternion was utilised as the coordinate system to represent the quadrotor's orientation. Two constraints were imposed; first q_0 was restricted to be ≥ 0 to obtain one possible solution when using Equation (2.1), and secondly re-normalisation of the quaternions through dividing each quaternion element with the quaternion norm $\|\bar{q}\|$ from Equation (3.4) after each simulation time step to restrict the unit quaternion to be a unit norm at all times. Advantageously, the unit quaternions can be reworked to Euler angles according to Equation (3.12) but rather Equation (2.10) was used to determine the resultant Euler angles from the angular velocities $\vec{\omega}$. It is worth noting, that the Euler angles must be restricted between $\pm\pi$ to avoid unpredicted behaviours.

Chapter 6

Conclusion and Future Work

In this dissertation, the AVIC strategy was implemented in aerial manipulation for the first time to improve contact inspection while the integral LQR was utilized to regulate the quadrotor's attitude. This chapter gives a summary of the results presented in Chapter 4 and Chapter 5. The section thereafter is a presentation of ideas for future developments.

6.1 Conclusion

The main objective of this dissertation was to achieve force tracking for contact inspection which was obtained using modified impedance control between the manipulator and the environment. The quadrotor position was regulated using LQR with an integral action. The linearisation process that was performed on the non-linear dynamic model of the quadrotor using unit quaternions as the orientation's coordinate system proved to give a fair approximation of the non-linear model. The integral LQR could achieve trajectory tracking to within acceptance errors. The unit quaternion can deviate its unit norm, in most instances, this results in a bizarre behaviour of the quadrotor. To prevent such, one can re-normalise the quaternions simple by dividing the quaternion by the quaternion norm after each simulation time step.

The position controller for the 2-DOF manipulator utilised the computed torque to perform feedback linearisation and the PD control for the joint angles which can be determined with the inverse kinematics to achieve good position control. The Newton-Euler algorithm was employed to extract the equations of motion for the manipulator with it's floating base on the COM of the quadrotor. To achieve force tracking for contact inspection, the AVIC strategy was implemented which performed better than the classic impedance control. The AVIC strategy has an adaptive law for the damping coefficient and it uses a zero stiffness value of the impedance parameters. The proposed strategy was evaluated through four simulation scenarios to demonstrate its effectiveness. The first scenario has the end-effector and quadrotor performing set-point tracking without physical interaction. The second scenario, the end-effector performs the same set-point tracking during physical interaction with desired contact force of 2 N while the quadrotor tracks a helix trajectory. The third scenario is another physical interaction with desired force increas-

ing at a rate of 0.4 N/s to a magnitude of 4 N while the end-effector and quadrotor track desired trajectories generated through cubic polynomials. Then the last scenario, simulates unknown environment by assigning a 5% uncertainty value to the environmental location and the desired force tracking is similar to the second scenario while the trajectory tracking is similar to the third scenario.

6.2 Future Work

The scope covered in this dissertation did not test the control on a physical prototype, but demonstrated that contact inspection can be achieved using an aerial manipulator. A prototype to investigate the proposed controller effectiveness will be the next step of action that will require other considerations that can influence the stability of the quadrotor's behaviour such as obstacle detection and avoidance, limits to motor speeds, altitude sensor ranges, and acceleration and force sensor noises. To extend the integral LQR for tracking a time-varying set point, to improve its robustness and accuracy. To further improve the robustness for the control of aerial manipulator, the backstepping control method can be designed for the entire system to handle model uncertainties and external disturbances.

References

- [1] A. AlAkhras, I.H. Sattar, M. Alvi, M.W. Qanbar, M.A. Jaradat, and M. Alkaddour. “The Design of a Lightweight Cable Aerial Manipulator with a CoG Compensation Mechanism for Construction Inspection Purposes.” *Applied Sciences*, vol. 12, pp. 1-17, 2022.
- [2] A. Alshawhi and S. Kizir. “Impedance control of 2dof serial robot manipulator.” *International Conference on Advanced Technologies, Computer Engineering and Science (ICATCES’ 18)*, pp. 244-249, May 11-13, 2018.
- [3] A.E. Jimenez-Cano, J. Braga, G. Heredia, and A. Ollero. “Aerial manipulator for structure inspection by contact from the underside.” *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 1879-1884, Sep. 28 - Oct. 2, 2015.
- [4] A. Ivanovic, L. Markovic, M. Car, I. Duvnjak, and M. Orsag. “Towards Autonomous Bridge Inspection: Sensor Mounting Using Aerial Manipulators.” *Applied Sciences*, vol. 11, pp. 1-21, 2021.
- [5] A. Khalifa, M. Fanni, A. Ramadan, and A. Abo-Ismael. “Modeling and Control of a New Quadrotor Manipulation System.” *2012 First International Conference on Innovative Engineering Systems (IEEE)*, pp.109-114, Dec. 07-09, 2012.
- [6] B. Yüksel, C. Secchi, H.H. Bühlhoff, and A. Franchi. “Aerial physical interaction via IDA-PBC.” *The International Journal of Robotics Research*, vol. 38(4), pp. 403-421, 2019.
- [7] B. Yüksel, S. Mahboubi, C. Secchi, H.H. Bühlhoff, and A. Franchi. “Design, Identification and Experimental Testing of a Light-weight Flexible-joint Arm for Aerial Physical Interaction.” *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 870-876, May 26-30, 2015.
- [8] C. Coulombe, D. Saussié, S. Achiche. “Modeling and gain-scheduled control of an aerial manipulator.” *International Journal of Dynamics and Control*, vol. 10, pp. 217-229, 2022.
- [9] D. Jinjun, G. Yahui, C. Ming, and D. Xianzhong. “Adaptive variable impedance control for dynamic contact force tracking in uncertain environment.” *Robotics and Autonomous Systems*, vol. 102, pp. 54-65, 2018.
- [10] E. Reyes-Valeria, R. Enriquez-Caldera, S. Camacho-Lara, and J. Guichard. “LQR control for a quadrotor using unit quaternions: Modeling and simulation.” *CONIELECOMP 2013*,

- 23rd International Conference on Electronics, Communications and Computing*, pp. 172-178, 2013.
- [11] F. Caccavale, G. Giglio, G. Muscio, and F. Pierri. “Adaptive control for UAVs equipped with a robotic arm.” *IFAC Proceedings of the 19th World Congress*, pp. 11049-11054, Aug 24-29, 2014.
 - [12] F. Sabatino. “Quadrotor: modeling, nonlinear control design, and simulation.” M.S. Dissertation, KTH Electrical Engineering, Sweden, 2015.
 - [13] G. Nava, Q. Sablé, M. Tognon, D. Pucci, and A. Franchi. “Direct Force Feedback Control and Online Multi-Task Optimization for Aerial Manipulators.” *IEEE Robotics and Automation Letters*, vol. 5(2), pp. 331-338, Apr. 2020.
 - [14] H. Abaunza, P. Castillo, R. Lozano, and A. Victorino. “Quadrotor aerial manipulator based on dual quaternions.” *2016 International Conference on Unmanned Aircraft Systems (ICUAS)*, pp. 152-161, Jun. 7-10, 2016.
 - [15] Indrawanto, J. Swevers, and H. Van Brussel. “Robust Impedance Adaptive Control for Constrained Motions.” *Mesin*, vol. 17(2), pp. 68-79, 2002.
 - [16] J. Cariño, H. Abaunza, and P. Castillo. “Quadrotor quaternion control.” *2015 International Conference on Unmanned Aircraft Systems (ICUAS)*, pp. 825-831, 2015.
 - [17] J. de Gea and F. Kirchner. “Modelling and Simulation of Robot Arm Interaction Forces Using Impedance Control.” in *Proceedings of the 17th World Congress The International Federation of Automatic Control*, pp. 15589-15594, Jul. 6-11, 2008.
 - [18] J.M. Gómez-de-Gabriel, J.M. Gandarias, F.J. Péres-Maldonado, F.J. Garcia-Núñez, E.J. Fernández-Garcia, and A.J. Garcia-Cerezo. “Methods for autonomous Wristband Placement with a Search-and-Rescue Aerial Manipulator.” *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp.7838-7844, 2018.
 - [19] J.W. Yeol, D. Toohey, and Y.W. Hwang. “Design and Analysis of a Multiple Tentacle System for Mobile Manipulation in Micro Aerial Vehicles.” *Procedia Computer Science*, vol. 105, pp. 7-13, 2017.
 - [20] K. Alexis, C. Huerzeler, and R. Siegwart. “Hybrid predictive control of a coaxial aerial robot for physical interaction through contact.” *Control Engineering Practice*, vol. 32, pp. 96-112, 2014.
 - [21] K. Baizid, G. Giglio, F. Pierri, M. Trujillo, G. Antonelli, F. Caccavale, and A. Ollero. “Behavioral control of unmanned aerial vehicle manipulator systems.” *Autonomous Robots*, vol. 41(5), pp. 1203-1220, 2017.

- [22] K. Choutri, M. Lagha, L. Dala, and M. Lipatov. "Quadrotors trajectory tracking using a differential flatness-quaternion based approach." *2017 7th International Conference on Modeling, Simulation, and Applied Optimization (ICMSAO)*, pp. 1-5, 2017.
- [23] K.Y. Chee and Z.W. Zhong. "Control, navigation and collision avoidance for an unmanned aerial vehicle." *Sensors and Actuators A: Physical*, vol. 190, pp. 66-76, 2013.
- [24] L. Ding, G. Zhu, Y. Li, and Y. Wang. "Cable-Driven Unmanned Aerial Manipulator Systems for Water Sampling: Design, Modeling, and Control." *Drones*, vol. 7, pp. 1-28, Jul. 6, 2023.
- [25] L.E. Caraballo, J.M. Diaz-Báñez, I. Maza, and A. Ollero. "Innovative Applications of O.R.: The block-information-sharing strategy for task allocation: A case study for structure assembly with aerial robots." *European Journal of Operational Research*, vol. 260, pp. 725-738, 2017.
- [26] L. Martins, C. Cardeira, and P. Oliveira. "Linear Quadratic Regulator for Trajectory Tracking of a Quadrotor." *IFAC PapersOnLine*, vol. 52(12), pp. 176-181, 2019.
- [27] L. Villani and J. De Schutter. "Force control." *Springer handbook of robotics*, 2008, pp. 161-185.
- [28] M. Fanni and A. Khalifa. "A New 6-DOF Quadrotor Manipulation System: Design, Kinematics, Dynamics, and Control." *IEEE/ASME Transactions on Mechatronics*, vol. 22(3), pp. 1315-1326, Jun. 2017.
- [29] M. Fumagalli, R. Naldi, A. Macchelli, R. Carloni, S. Stramigioli, and L. Marconi. "Modeling and control of a flying robot for contact inspection." *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 3532-3537, Oct. 7-12, 2012.
- [30] M. Islam and M. Okasha. "A Comparative Study of PD, LQR and MPC on Quadrotor Using Quaternion Approach." *2019 7th International Conference on Mechatronics Engineering (ICOM)*, pp. 1-6, 2019.
- [31] M. Kobilarov. "Nonlinear Trajectory Control of Multi-body Aerial Manipulators." *Journal of Intelligent and Robotic Systems - Theory and Applications*, vol. 73, pp. 679-692, 2014.
- [32] M. Okasha, J. Shah, W. Fauzi, and Z. Hanouf. "Gain scheduled linear quadratic control for quadcopter." *IOP Conference Series: Materials Science and Engineering*, vol. 270, Dec. 12, 2017.
- [33] M. Rahman, H. Liu, I.D. Cardenas, A. Starr, A. Hall, R. Anderson. "A Review on the Prospects of Mobile Manipulators for Smart Maintenance of Railway Track." *Applied Sciences*, vol. 13, pp. 1-30, 2023.

- [34] M. Sharifi and H. Sayyaadi. “Nonlinear robust adaptive Cartesian impedance control of UAVs equipped with a robot manipulator.” *Advanced Robotics*, vol. 29(3), pp. 171-186, 2015.
- [35] M.W. Spong, S. Hutchinson, and M. Vidyasagar. *Robot Modeling and Control*. John Wiley & Sons, Inc., 2006, pp. 187-224.
- [36] N. Diolaiti, C. Melchiorri, and S. Stramigioli. “Contact Impedance Estimation for Robotic Systems.” *IEEE Transactions on Robotics*, vol. 21(5), pp. 925-935, Oct. 2005.
- [37] P. Chermprayong, K. Zheng, F. Xiao, and M. Kovac. “An Integrated Delta Manipulator for Aerial Repair.” *IEEE Robotics and Automation Magazine*, pp. 54-66, Mar. 2019.
- [38] P. Ghiglino, J.L. Forshaw, and V.J. Lappas. “OQTAL: Optimal quaternion tracking using attitude error linearization.” *IEEE Transactions on Aerospace and Electronic Systems*, vol. 51(4), pp. 2715-2731, Oct. 2015.
- [39] R. Naldi, A. Macchelli, D. Mengoli, and L. Marconi. “A simulator environment for aerial service robot prototypes.” *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 1893-1898, Nov. 3-7, 2013.
- [40] S. Bouabdallah and R. Siegwart. “Full control of a quadrotor.” *In Intelligent robots and systems (IROS), 2007 IEEE/RSJ international conference*, pp. 153-158, 2007.
- [41] S. Kannan, M.A. Olivares-Mendez, and H. Voos. “Modeling and Control of Aerial Manipulation Vehicle with Visual sensor.” *2nd IFAC Workshop on Research, Education and Development of Unmanned Aerial Systems*, pp. 303-309, Nov. 20-22, 2013.
- [42] S. Sawyer. “Gain-Scheduled Control of a Quadcopter UAV.” M.S. Dissertation, University of Waterloo, Canada, 2015.
- [43] T.W. Danko and P.Y. Oh. “Design and Control of a Hyper-Redundant Manipulator for Mobile Manipulating Unmanned Aerial Vehicles.” *Journal of Intelligent and Robotic Systems*, vol. 73, pp. 709-723, 2014.
- [44] V. Lippiello and F. Ruggiero. “Cartesian Impedance Control of a UAV with a Robotic Arm.” *10th IFAC Symposium on Robot Control*, pp. 704-709, Sep. 5-7, 2012.
- [45] W. Lin, L. Yuan, W. Dong, R. Guan, X. Gu, and C. Qian. “Impedance control based analysis of compliant flange.” *In Automation (YAC), 2017 32nd Youth Academic Annual Conference of Chinese Association of IEEE*, pp. 1222-1227, 2017.
- [46] W. Neeley. “Design and Development of a High-Performance Quadrotor Control Architecture Based on Feedback Linearization.” M.S. Dissertation, University of New Mexico, New Mexico, 2015.

- [47] Y. Stergiopoulos, E. Kontouras, K. Gkountas, K. Giannousakis, and A. Tzes. “Modeling and Control Aspects of a UAV with an Attached Manipulator.” *24th Mediterranean Conference on Control and Automation (MED)*, pp. 653-658, Jun. 21-24, 2016.
- [48] Z. Samadikhoshkho, S. Ghorbani, F. Janabi-Sharifi, and K. Zareinia. “Nonlinear control of aerial manipulation systems.” *Aerospace Science and Technology*, vol. 104, 2020.

Appendix A

Detailed derivation of the Manipulator dynamics

Fixing the reference frames and notations as discussed in section 3.2 for a manipulator that is made up of two revolute joints $i = 1, 2$ as in Figure A.1. Denoting the joint angle with q_i , individual mass of link i as m_i , and l_i being the length of link i . Then the distance from frame $\{i - 1\}$ to the COM of link i is expressed as l_{ci} while expressing I_i as the moment of inertia for link i about the z axis (not shown since it's out of the page) passing through the COM of link i .

Since the focus is to determine the relationship between the manipulator joint torques $\tau(\cdot)$ and the manipulator joint angles $q(\cdot)$. The recursive Newton-Euler equations will be employed to analyse manipulator's dynamics according to the book by Spong et al., [35] but for a floating-base instead of the fixed-base.

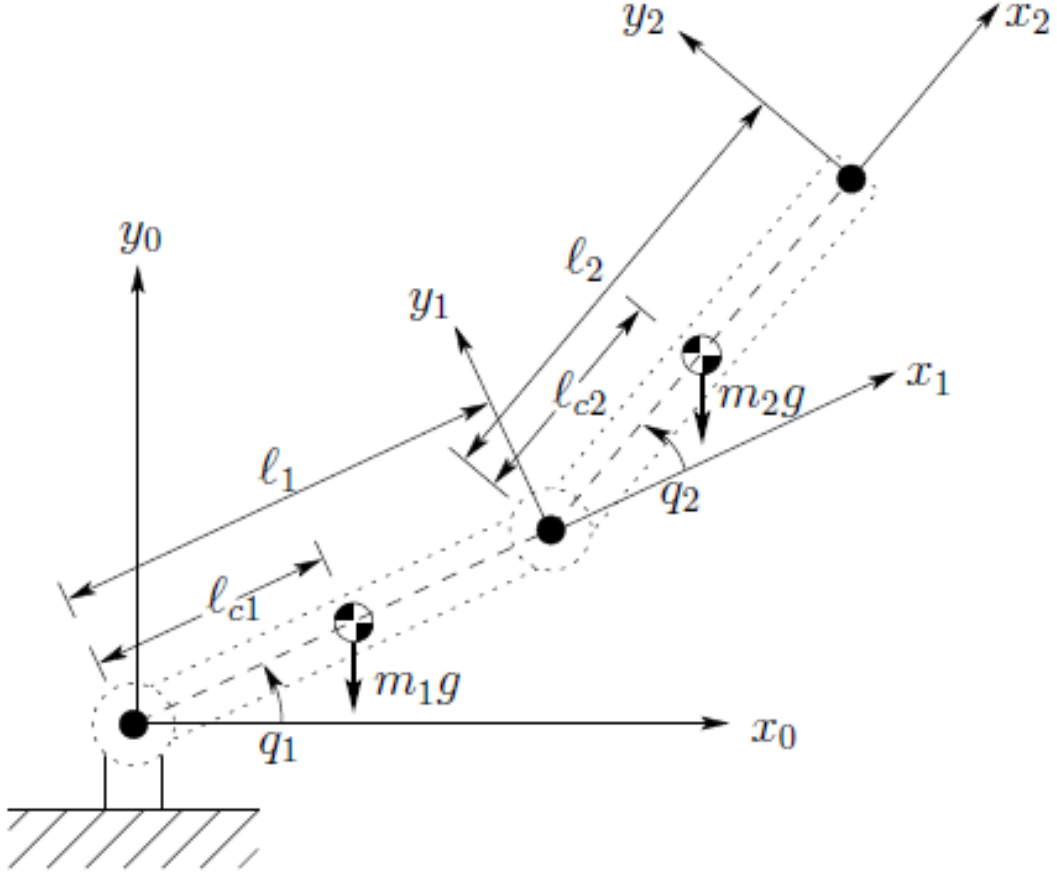


Figure A.1: Two-link revolute joint manipulator [35]

The procedure for recursive Newton-Euler equations starts with the velocities (ω_1, ω_2) and accelerations (α_1, α_2) in terms of q_1, q_2 together with their derivatives. Assuming the initial conditions for the base-fixed frame $\{0\}$ as ω_0 and α_0 being the third components from Equation (3.24), and $a_{e,0} = [a_x \ a_y \ a_z]^T$. This is because the rotation matrix R_0^1 given in Equation 3.27 does not perturb the third component of the vectors, this applies to the generic matrix R_i^{i+1} as well. Using Equations (3.16) and (3.18) for link 1 and link 2, results are:

$$\omega_1 = (\omega_0 + \dot{q}_1)k, \quad \alpha_1 = (\alpha_0 + \ddot{q}_1)k \quad (\text{A.1})$$

$$\omega_2 = (\omega_1 + \dot{q}_2)k, \quad \alpha_2 = (\alpha_1 + \ddot{q}_2)k \quad (\text{A.2})$$

In the forward recursion, the following displacement vectors are necessary:

$$r_{1,c1} = l_{c1}i, \quad r_{2,c1} = (l_1 - l_{c1})i, \quad r_{1,2} = l_1i \quad (\text{A.3})$$

$$r_{2,c2} = l_{c2}i, \quad r_{3,c2} = (l_2 - l_{c2})i, \quad r_{2,3} = l_2i \quad (\text{A.4})$$

A.1 Forward Recursion: Link 1

Now, the acceleration at the COM of link 1 can be found according to Equation (3.19) as:

$$\begin{aligned}
 a_{c,1} &= (R_0^1)^T a_{e,0} + \alpha_1 k \times l_{c1} i + \omega_1 k \times (\omega_1 k \times l_{c1} i) \\
 &= (R_0^1)^T a_{e,0} + l_{c1} \alpha_1 j - l_{c1} \omega_1^2 i \\
 &= \begin{bmatrix} a_x \cos q_1 + a_y \sin q_1 - l_{c1}(\omega_0 + \dot{q}_1)^2 \\ -a_x \sin q_1 + a_y \cos q_1 + l_{c1}(\alpha_0 + \ddot{q}_1) \\ a_z \end{bmatrix}
 \end{aligned} \tag{A.5}$$

Notice the simplicity of this computation when it is done with respect to frame $\{1\}$ compared with similar computation in the base-fixed frame $\{0\}$. Then the gravitational vector:

$$g_1 = -(R_0^1)^T g j = g \begin{bmatrix} \sin q_1 \\ -\cos q_1 \\ 0 \end{bmatrix} \tag{A.6}$$

where g is the gravitational acceleration. It is worth mentioning that the third component of the gravitational vectors (g_1, g_2) are always zero and can be omitted. Likewise, the third component of the force vectors are zero while only the third component of the torque vectors are non-zero. Replacing l_{c1} by l_1 in Equation (A.5) completes the computation for link 1 and finds the acceleration for the end of link 1 as:

$$a_{e,1} = \begin{bmatrix} a_x \cos q_1 + a_y \sin q_1 - l_1(\omega_0 + \dot{q}_1)^2 \\ -a_x \sin q_1 + a_y \cos q_1 + l_1(\alpha_0 + \ddot{q}_1) \\ a_z \end{bmatrix} \tag{A.7}$$

A.2 Forward Recursion: Link 2

The linear acceleration of link 2 is determined as follows:

$$a_{c,2} = (R_1^2)^T a_{e,1} + \alpha_2 k \times l_{c2} i + \omega_2 k \times (\omega_2 k \times l_{c2} i) \tag{A.8}$$

The first term on the right hand side of this equation can be determined as:

$$\begin{aligned}
 (R_1^2)^T a_{e,1} &= \begin{bmatrix} \cos q_2 & \sin q_2 & 0 \\ -\sin q_2 & \cos q_2 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} a_x \cos q_1 + a_y \sin q_1 - l_1(\omega_0 + \dot{q}_1)^2 \\ -a_x \sin q_1 + a_y \cos q_1 + l_1(\alpha_0 + \ddot{q}_1) \\ a_z \end{bmatrix} \\
 &= \begin{bmatrix} \cos q_2(a_x \cos q_1 + a_y \sin q_1 - l_1(\omega_0 + \dot{q}_1)^2) + \sin q_2(-a_x \sin q_1 + a_y \cos q_1 + l_1(\alpha_0 + \ddot{q}_1)) \\ -\sin q_2(a_x \cos q_1 + a_y \sin q_1 - l_1(\omega_0 + \dot{q}_1)^2) + \cos q_2(-a_x \sin q_1 + a_y \cos q_1 + l_1(\alpha_0 + \ddot{q}_1)) \\ a_z \end{bmatrix}
 \end{aligned} \tag{A.9}$$

Then substituting Equation (A.9) into Equation (A.8) gives the linear acceleration of link 2 as:

$$a_{c,2} = [\cos q_2(a_x \cos q_1 + a_y \sin q_1 - l_1(\omega_0 + \dot{q}_1)^2) + \sin q_2(-a_x \sin q_1 + a_y \cos q_1 + l_1(\alpha_0 + \ddot{q}_1)) - l_{c2}(\dot{q}_1 + \dot{q}_2 + \omega_0)]i + [\sin q_2(-a_x \sin q_1 + a_y \cos q_1 + l_1(\alpha_0 + \ddot{q}_1)) - \sin q_2(a_x \cos q_1 + a_y \sin q_1 - l_1(\omega_0 + \dot{q}_1)^2) + l_{c2}(\ddot{q}_1 + \ddot{q}_2 + \alpha_0)]j + [a_z]k \quad (\text{A.10})$$

And the gravitational vector for link 2 is given as:

$$g_2 = g \begin{bmatrix} \sin(q_1 + q_2) \\ -\cos(q_1 + q_2) \\ 0 \end{bmatrix} \quad (\text{A.11})$$

The forward recursion for the 2-DOF manipulator is complete therefore, $a_{e,2}$ calculation is omitted.

A.3 Backward Recursion: Link 2

Now following the computation of the joint forces and torques in the backward recursion and taking $f_3 = 0$ as the terminal condition. Then applying Equation (3.14) with $i = 2$ gives:

$$f_2 = m_2 a_{c,2} - m_2 g_2 \quad (\text{A.12})$$

Substituting $a_{c,2}$ and g_2 , f_2 can be written as:

$$f_2 = m_2 (R_0^2)^T a_{e,0} + m_2 \begin{bmatrix} -l_1 \omega_1^2 \cos q_2 + l_1 \alpha_1 \sin q_2 - l_{c2} \omega_2^2 - g \sin(q_1 + q_2) \\ l_1 \omega_1^2 \sin q_2 + l_1 \alpha_1 \cos q_2 + l_{c2} \alpha_2 + g \cos(q_1 + q_2) \\ 0 \end{bmatrix} \quad (\text{A.13})$$

The term $(R_0^2)^T a_{e,0}$ is a double transformation of $a_{e,0}$ from base-fixed frame $\{0\}$ to frame $\{2\}$ and is determined as:

$$(R_0^2)^T a_{e,0} = \begin{bmatrix} \cos(q_1 + q_2) & \sin(q_1 + q_2) & 0 \\ -\sin(q_1 + q_2) & \cos(q_1 + q_2) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} a_x \\ a_y \\ a_z \end{bmatrix} \quad (\text{A.14})$$

Now, applying Equation (3.15) with $i = 2$ and noting that $\tau_3 = 0$ gives:

$$\tau_2 = I_2 \alpha_2 + \omega_2 \times (I_2 \omega_2) - f_2 \times l_{c2} i \quad (\text{A.15})$$

Then substitute for ω_2 , α_2 from Equation (A.1) and for f_2 from Equation (A.13). The cross-product $f_2 \times l_{c2}i$ has its magnitude being the second component of f_2 , because the cross-product is aligned with the unit vector k . Similarly, the product $\omega_2 \times (I_2\omega_2)$ is zero because both vectors are collinear. Then τ_2 is found to be:

$$\begin{aligned} \tau_2 = & m_2 l_{c2} [a_y \cos(q_1 + q_2) - a_x \sin(q_1 + q_2) + l_1 \cos q_2 (\alpha_0 + \ddot{q}_1) + g \cos(q_1 + q_2) \\ & + l_1 \sin q_2 (\omega_0 + \dot{q}_1)^2 + l_{c2} (\ddot{q}_1 + \ddot{q}_2 + \alpha_0)] k + I_2 (\ddot{q}_1 + \ddot{q}_2 + \alpha_0) k \end{aligned} \quad (\text{A.16})$$

A.4 Backward Recursion: Link 1

Once again, applying Equation (3.14) with $i = 1$ gives the force f_1 as:

$$f_1 = m_1 a_{c,1} + R_1^2 f_2 - m_1 g_1 \quad (\text{A.17})$$

Transformation of f_2 by R_1^2 is as follows:

$$\begin{aligned} R_1^2 f_2 = & m_2 [R_1^2 (R_1^2)^T (R_0^1)^T] a_{e,0} \\ & + m_2 R_1^2 \begin{bmatrix} -l_1 \omega_1^2 \cos q_2 + l_1 \alpha_1 \sin q_2 - l_{c2} \omega_2^2 - g \sin(q_1 + q_2) \\ l_1 \omega_1^2 \sin q_2 + l_1 \alpha_1 \cos q_2 + l_{c2} \alpha_2 + g \cos(q_1 + q_2) \\ 0 \end{bmatrix} \end{aligned} \quad (\text{A.18})$$

After transforming and doing some algebra, Equation (A.18) can be expanded as:

$$\begin{aligned} R_1^2 f_2 = & m_2 [a_x \cos q_1 + a_y \sin q_1 - l_1 \omega_1^2 - g \sin(q_1 + 2q_2) - l_{c2} \omega_2^2 \cos q_2 \\ & - l_{c2} \alpha_2 \sin q_2] i + m_2 [-a_x \sin q_1 + a_y \cos q_1 + l_1 \alpha_1 \\ & - l_{c2} \omega_2^2 \sin q_2 + l_{c2} \alpha_2 \cos q_2 + g \cos q_1] j \end{aligned} \quad (\text{A.19})$$

Applying Equation (3.15) again on link 1. Results in τ_1 being:

$$\tau_1 = R_1^2 \tau_2 - f_1 \times l_{c1} i - (R_1^2 f_2) \times (l_1 - l_{c1}) i + I_1 \alpha_1 + \omega_1 \times (I_1 \omega_1) \quad (\text{A.20})$$

Now, the first $R_1^2 \tau_2 = \tau_2$ because the rotation matrix does not perturb the third component of the vectors. Once again, the the product $\omega_1 \times (I_1 \omega_1)$ is zero. Then substituting the force f_1 from Equation (A.17) into Equation (A.20) results in:

$$\tau_1 = \tau_2 - m_1 a_{c,1} \times l_{c1} i + m_1 g_1 \times l_{c1} i - (R_1^2 f_2) \times l_1 i + I_1 \alpha_1 \quad (\text{A.21})$$

All these products are quite straightforward to do but tedious to follow through. Once again, the cross-product $R_1^2 f_2 \times l_1 i$ has its magnitude being the second component of $R_1^2 f_2$ from Equation (A.19) because the cross-product is aligned with the unit vector k . Finally, the torque τ_1 is calculated as:

$$\begin{aligned} \tau_1 = & \tau_2 + m_1 l_{c1} [l_{c1}(\alpha_0 + \ddot{q}_1) - a_x \sin q_1 + a_y \cos q_1] + m_1 g l_{c1} \cos q_1 \\ & + I_1(\alpha_0 + \ddot{q}_1) + m_2 l_1 [-a_x \sin q_1 + a_y \cos q_1 + l_1(\alpha_0 + \ddot{q}_1) + g \cos q_1 \\ & - l_{c2} \sin q_2 (\dot{q}_1 + \dot{q}_2 + \omega_0)^2 + l_{c2} \cos q_2 (\ddot{q}_1 + \ddot{q}_2 + \alpha_0)] \end{aligned} \quad (\text{A.22})$$

Now substituting τ_2 in τ_1 and collect terms, yields the following equations:

$$d_{11}\ddot{q}_1 + d_{12}\ddot{q}_2 + c_{11}\dot{q}_1 + c_{12}\dot{q}_2 + \phi_1 = \tau_1$$

$$d_{21}\ddot{q}_1 + d_{22}\ddot{q}_2 + c_{21}\dot{q}_1 + \phi_2 = \tau_2 \quad (\text{A.23})$$

where the new coefficients are:

$$d_{11} = m_1 l_{c1}^2 + m_2 (l_1^2 + l_{c2}^2 + 2l_1 l_{c2} \cos q_2) + I_1 + I_2$$

$$d_{12} = d_{21} = m_2 (l_{c2}^2 + l_1 l_{c2} \cos q_2) + I_2$$

$$d_{22} = m_2 l_{c2}^2 + I_2$$

$$h = -m_2 l_1 l_{c2} \sin q_2$$

$$c_{11} = h(\dot{q}_2 + 2\omega_0)$$

$$c_{12} = h(\dot{q}_1 + \dot{q}_2 + 2\omega_0) \quad (\text{A.24})$$

$$c_{21} = -h(\dot{q}_1 + 2\omega_0)$$

$$\begin{aligned} \phi_1 = & m_2 l_{c2} [g \cos(q_1 + q_2) + d_{11}\alpha_0 + a_y \cos(q_1 + q_2) - a_x \sin(q_1 + q_2)] \\ & + (a_y \cos q_1 - a_x \sin q_1)(m_1 l_{c1} + m_2 l_1) + (m_1 l_{c1} + m_2 l_1) g \cos q_1 \end{aligned}$$

$$\phi_2 = m_2 l_{c2} [g \cos(q_1 + q_2) + a_y \cos(q_1 + q_2) - a_x \sin(q_1 + q_2)] + d_{21}\alpha_0 - h\omega_0^2$$

Finally, the dynamical equations of the 2-DOF manipulator can be written in a matrix form known as *Euler-Lagrange* Equations:

$$D(q)\ddot{q} + C(q, \dot{q})\dot{q} + g(q) = \tau \quad (\text{A.25})$$

Where

$$D(q) = \begin{bmatrix} d_{11} & d_{12} \\ d_{21} & d_{22} \end{bmatrix}, C(q, \dot{q}) = \begin{bmatrix} c_{11} & c_{12} \\ c_{21} & 0 \end{bmatrix}, g(q) = \begin{bmatrix} \phi_1 \\ \phi_2 \end{bmatrix}, \tau = \begin{bmatrix} \tau_1 \\ \tau_2 \end{bmatrix} \quad (\text{A.26})$$

Appendix B

Stability and convergence of the AVIC strategy

In this section, the constraint conditions for stability are considered for the AVIC strategy given in Equations (4.41) and (4.42). The tracking error δf can be proven to reach zero if the system can be asymptotically stable [9]. Therefore, substituting $\Delta b(t)$ from Equation (4.42) into Equation (4.41), result with the following:

$$\begin{aligned} f_e(t) - f_d(t) &= m\ddot{e}(t) + b\dot{e}(t) + b\Phi(t - \lambda) \\ + \sigma[f_d(t - \lambda) - f_e(t - \lambda)] &= m[\ddot{e}(t) + \delta\ddot{y}_e(t)] + b[\dot{e}(t) + \delta\dot{y}_e(t)] \\ &\quad + b\Phi(t - \lambda) + \sigma[f_d(t - \lambda) - f_e(t - \lambda)] \end{aligned} \quad (\text{B.1})$$

Reorganising Equation (B.1) to become:

$$\begin{aligned} m\ddot{e}(t) + b\dot{e}(t) + b\Phi(t - \lambda) - [f_e(t) - f_d(t)] \\ + \sigma[f_d(t - \lambda) - f_e(t - \lambda)] &= -m\delta\ddot{y}_e(t) - b\delta\dot{y}_e(t) \end{aligned} \quad (\text{B.2})$$

Differentiating the contact force $f_e = k_e(y_e - y_{2d}) = -k_e e$, leads to:

$$\begin{aligned} \dot{e} &= -\dot{f}_e/k_e \\ \ddot{e} &= -\ddot{f}_e/k_e \end{aligned} \quad (\text{B.3})$$

Then substituting the position error derivatives from Equation (B.3) into Equation (B.2), gives:

$$\begin{aligned} -m\ddot{f}_e(t) - b\dot{f}_e(t) + bk_e\Phi(t - \lambda) - k_e[f_e(t) - f_d(t)] \\ + \sigma k_e[f_d(t - \lambda) - f_e(t - \lambda)] &= -mk_e\delta\ddot{y}_e(t) - bk_e\delta\dot{y}_e(t) \end{aligned} \quad (\text{B.4})$$

Assigning $\hat{f}_e(t) = k_e \delta y_e(t)$, then Equation (B.4) can become:

$$\begin{aligned} m\ddot{f}_d(t) - m\ddot{f}_e(t) + b\dot{f}_d(t) - b\dot{f}_e(t) + bk_e\Phi(t - \lambda) + \sigma k_e[f_d(t - \lambda) - f_e(t - \lambda)] \\ - k_e[f_e(t) - f_d(t)] = m\ddot{f}_d(t) - mk_e\delta\ddot{y}_e(t) + b\dot{f}_d(t) - bk_e\delta\dot{y}_e(t) \\ = m\ddot{f}_d(t) - m\ddot{\hat{f}}_e(t) + b\dot{f}_d(t) - b\dot{\hat{f}}_e(t) \end{aligned} \quad (\text{B.5})$$

Let $c(t) = f_d(t) - f_e(t)$ and also let $r(t) = f_d(t) - \hat{f}_e(t)$, then Equation (B.5) can be written as:

$$m\ddot{c} + b\dot{c} + bk_e\Phi(t - \lambda) + \sigma k_e c(t - \lambda) + k_e c = m\ddot{r} + b\dot{r} \quad (\text{B.6})$$

Following the principle of dispersion, expanding the Φ series to an n^{th} element [9] can be done as follows :

$$b\Phi(t - \lambda) = b\Phi(t - (n + 1)\lambda) + \sigma c(t - (n + 1)\lambda) + \dots + \sigma c(t - 2\lambda) \quad (\text{B.7})$$

Typically, the adaptive compensation $\Phi(t - (n + 1)\lambda)$ is set to zero as the initial value [9]. Hence Equation (B.6) can be re-expressed as:

$$m\ddot{c} + b\dot{c} + k_e c + \sigma k_e (c(t - (n + 1)\lambda) + \dots + c(t - \lambda)) = m\ddot{r} + b\dot{r} \quad (\text{B.8})$$

Equation (B.9) below can be attained through Equation (B.8) in correspondence with the delay property of the Laplace transform $I[f(t - \tau)] = e^{-\tau s} F(s)$ [9].

$$\frac{c(s)}{r(s)} = \frac{ms^2 + bs}{ms^2 + bs + k_e + \sigma k_e (e^{-(n+1)\lambda s} + \dots + e^{-\lambda s})} \quad (\text{B.9})$$

The constraint in Equation (B.10) if met, will ensure Equation (B.9) is stable [9].

$$ms^2 + bs + k_e + \sigma k_e (e^{-(n+1)\lambda s} + \dots + e^{-\lambda s}) = 0 \quad (\text{B.10})$$

Assuming that n is a huge number, Equation (B.9) can be written as:

$$\sum_{n=1}^{\infty} e^{-\lambda ns} = \frac{e^{-\lambda s}}{1 - e^{-\lambda s}} \quad (\text{B.11})$$

For a sufficient sampling rate λ , using Taylor expansion, one can approximate the delayed term as $e^{-\lambda s} \simeq 1 - \lambda s$ [9]. Then rewriting Equation (B.10) as:

$$\lambda ms^3 + \lambda bs^2 + k_e \lambda (1 - \sigma)s + \sigma k_e = 0 \quad (\text{B.12})$$

The Routh array shown in Equation (B.13) was attained in correspondence with the Routh criterion [9]. If the coefficients in the first column of the array together with the coefficients in Equation (B.12) are greater than zero as depicted in Equation (B.14), then the system stability is guaranteed.

$$\begin{array}{rcl}
s^3 & \lambda m & k_e \lambda (1 - \sigma) \\
s^2 & \lambda b & \sigma k_e \\
s^1 & \frac{\lambda b k_e \lambda (1 - \sigma) - \lambda m \sigma k_e}{\lambda b} & 0 \\
s^0 & \sigma k_e & 0
\end{array} \tag{B.13}$$

$$\begin{aligned}
\frac{\lambda b k_e \lambda (1 - \sigma) - \lambda m \sigma k_e}{\lambda b} &> 0 \\
k_e \lambda (1 - \sigma) &> 0 \\
\sigma k_e &> 0
\end{aligned} \tag{B.14}$$

Equation (B.14) can be simplified as:

$$0 < \sigma < \frac{\lambda b}{m + \lambda b} \tag{B.15}$$

Stable system will reach a steady-state error e_{ss} in the Laplace transform. And for convergence, the steady-state error e_{ss} is calculated with the following limits:

$$\begin{aligned}
e_{ss} &= \lim_{t \rightarrow \infty} e(t) = \lim_{s \rightarrow 0} sE(s) = \lim_{s \rightarrow 0} s(c(s) - r(s)) \\
&= \lim_{s \rightarrow 0} s \left[\frac{ms^2 + bs}{ms^2 + bs + k_e + \sigma k_e (e^{-(n-1)\lambda s} + \dots + e^{-\lambda s})} r(s) - r(s) \right]
\end{aligned} \tag{B.16}$$

If a step input with the form as $r(s) = (1/s)$ is used, then Equation (B.16) goes to:

$$e_{ss} = \lim_{s \rightarrow 0} s(c(s) - r(s)) = -1 \tag{B.17}$$

Two conclusions from Equation (B.17) and are stated below as:

$$\lim_{s \rightarrow 0} sc(s) = 0 \tag{B.18}$$

$$\lim_{t \rightarrow \infty} c(t) = 0 \tag{B.19}$$

Consequently, $t \rightarrow \infty$, $f_e \rightarrow f_d$. The convergence of the AVIC strategy is established. In fact, the desired contact force can be tracked even for a dynamic (cosine) $r(s)$ simulated by Duan et al., [9] achieved zero tracking error.