AMERICAN UNIVERSITY OF BEIRUT

ENERGY-OPTIMAL PATH-PLANNING FOR QUADROTORS IN FORESTS

by
CHRISTOPH AOUN

A thesis
submitted in partial fulfillment of the requirements
for the degree of Master of Engineering
to the Department of Mechanical Engineering
of the Faculty of Engineering and Architecture
at the American University of Beirut

Beirut, Lebanon
July 2019

AMERICAN UNIVERSITY OF BEIRUT

ENERGY-OPTIMAL PATH-PLANNING FOR QUADROTORS IN FORESTS

by

CHRISTOPH AOUN

Approved by:

Dr. Naseem Daher, Assistant Professor, Advisor
Department of Electrical and Computer Engineering

Dr. Elie Shammas, Associate Professor, Co-advisor
Department of Mechanical Engineering

Dr. Daniel Asmar, Associate Professor, Member of Committee
Department of Mechanical Engineering

Dr. Imad Elhajj, Professor, Member of Committee
Department of Electrical and Computer Engineering

Date of thesis defense: July 18, 2019

AMERICAN UNIVERSITY OF BEIRUT

THESIS, DISSERTATION, PROJECT RELEASE FORM

Student Name:

AOUN	Christoph	Elias
Last	First	Middle

● Master's Thesis ○ Master's Project ○ Doctoral Dissertation

☒ I authorize the American University of Beirut to: (a) reproduce hard or electronic copies of my thesis, dissertation, or project; (b) include such copies in the archives and digital repositories of the University; and (c) make freely available such copies to third parties for research or educational purposes.

☐ I authorize the American University of Beirut, to: (a) reproduce hard or electronic copies of it; (b) include such copies in the archives and digital repositories of the University; and (c) make freely available such copies to third parties for research or educational purposes after:

One ---- year from the date of submission of my thesis, dissertation, or project.
Two ---- years from the date of submission of my thesis, dissertation, or project.
Three ---- years from the date of submission of my thesis, dissertation, or project.

[signature] August 1, 2019

Signature Date

ACKNOWLEDGMENTS

Special thanks are for Dr. Naseem Daher and Dr. Elie Shammas who advised me throughout the master's program, not only in the technical issues but also in management and discipline.

My recognition and gratitude are addressed to those who helped in accomplishing the work – Vision and Robotics Lab namely Ali Kanso and my colleague Mohammad Tuqan who was always technical and moral support especially as we progressed together throughout our respective researches.

AN ABSTRACT OF THE THESIS OF

Christoph Aoun for Master of Engineering
Major: Mechanical Engineering

Title: Energy-Optimal Path-Planning For Quadrotors In Forests

Motivated by the threat to the Lebanese forests brought upon by the Pine Processionary Moths, a system-level algorithm is proposed to have aerial drones navigate forest environments and visit each tree in an energy-optimal manner. In fact, the developed path-planning algorithm can be generalized for use to visit and inspect various cylindrically-shaped objects (e.g. power poles, concrete structures, and similar) in an energy-optimal manner. Given a map of the domain, an energy-optimal path is established between all pairs of objects (trees in this case) using optimal control theory to arrive at a hybrid solver of different transcription methods including Legendre-Gauss-Radau (LGR) and Hermite-Simpson (H-S) collocation methods. For cases with very large number of trees, a third-order polynomial estimation is established to map the position coordinates to energy consumption, which results in a significant reduction of computation time.

After populating an adjacency matrix from the optimal control solver or polynomial estimation, the problem is defined as a travelling salesman problem (TSP), where the drone is to visit all objects only once, which requires the use of graph theory; Integer Linear Programming (ILP) along with Sub-Tour Elimination Constraints (SEC) are used to develop the general optimal tour.

The algorithm is tailored to the needs of the intended application where prior information from previous scans are leveraged to generate a new set of trees, which includes infected trees in addition to ones with a relatively high probability to be infected based on a proposed probability distribution. The tour is further modified as a second tour is going on where a change of status from infected to non-infected, or vice-versa, results in a change in the probability of infection of each tree, which in turn changes the pool of trees required to visit. This abrupt change in tree sets requires a new path that is satisfied by a Fixed-Start-Fixed-End travelling salesman problem, which is solved using a genetic algorithm that assigns the tree where the drone is located as the initial point, while the base is considered the final point.

CONTENTS

ILLUSTRATIONS

TABLES

CHAPTER I

INTRODUCTION

A. Problem

Unmanned Aerial Vehicles (UAVs) have become the center of attention lately especially with their various uses that arise every day. UAVs have been used in delivering packages, medical kits, and even pizzas. With this potential, quadrotors are opted to gain more attention and have their uses expanded into many fields including agriculture and forestry. Due to the vast areas of forests, it is important to maintain energy optimality in quadrotor motion and trajectory planning.

With the country of Lebanon having vast forests of Lebanese Cedars (Cedrus Libanos), Stone Pines (Pinus Pinea), and Turkish Pines (Pinus Brutia), these tree species have been severely impacted by the rise of Pine Processionary Moths (Thaumetopoea Pityocampa) [1]. The adult moths typically lay their eggs at the south-facing top edges of trees, for maximum sunlight and heat exposure [2]. After hatching, the larva starts feeding on the pine needles, causing the infected branches to lose nutrients and gradually affecting the entire tree's health. The density of these moths has reached severely high levels in Lebanon and around the Middle East and North Africa (MENA) region [2]. This is prompting local researchers (entomologists and engineers) to deal with this issue swiftly, to curb their spread and contain the grave threat that they pose to one of the greatest symbols of Lebanon, cedar trees [1].

Figure 1 Lebanese Stone Pine Forest [2]

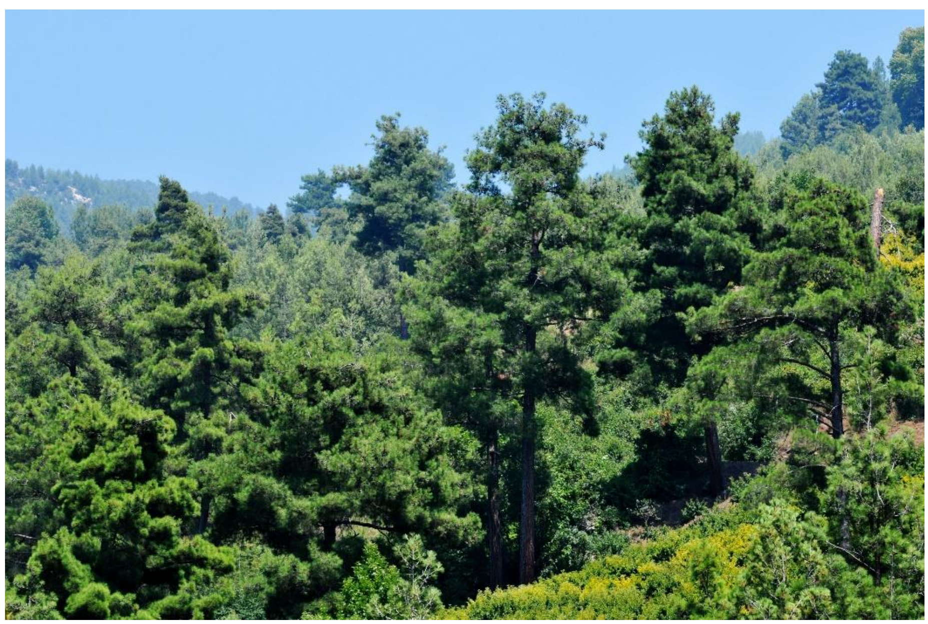

Figure 2 Lebanese Turkish Pine Forest [2]

Figure 3 Lebanese Cedars [2]

A challenge with this type of moths is that the caterpillars can only be detected during a very narrow time slot of two weeks once a year [3]. Detection is achieved through the prevalence of their nests, which occurs during the weakest phase of the moths' life cycle.

Figure 4 Pine Processionary Moth Life Cycle [4]

Since it is impractical to detect and exterminate them via primitive manual methods, and given that spraying pesticides in bulk over the entire forest via helicopters is banned [2], quadrotor UAVs are being proposed to aid with this task given their ability to deliver precise and small amounts of pesticides to infected areas only. In this scheme, quadrotors are released into a forest where they must autonomously navigate through it, visit each tree, and try to detect the developed nests for later actions to be taken. The aim of this research is to generate an energy-efficient path for the quadrotors to effectively and efficiently scan all the trees and be able to detect the nests. The detection (via computer vision) and the termination mechanism (via precise pesticide delivery) are not under the scope of this thesis, and they are being explored by other researchers.

Figure 5 Pine Processionary Moth [4]

Figure 6 Pine Processionary Moth Caterpillar [2]

Figure 7 Pine Processionary Moth Nest [2]

The general problem can be divided into three main parts. The first part entails modeling the domain in which the drones operate, where the proposed tree models lend themselves well for integration into the optimal control problem as constraints. The second part is the optimal trajectory and path-planning between each pair of trees, which solves for an optimal solution by minimizing energy consumption while considering obstacle-avoidance constraints. Finally, the third part generates the sequence for visiting the trees by employing the traveling salesman problem (TSP), where a single salesman (drone) needs to visit all the cities (trees) with a minimum total cost (energy).

B. Literature Review

1. Node-to-Node

Moving from one way-point to another in a three-dimensional environment has been extensively studied, and an array of algorithms exist and could be categorized into five main groups.

The first category is sampling-based algorithms [4], which are based on a priori knowledge of the domain, and they aim at generating a path that avoids obstacles in its course [4]. Available sampling-based algorithms include Rapidly Exploring Random Tree (RRT) [5], Probabilistic Roadmap [6], Voronoi diagrams [7], and Potential Fields [8]. The main downside of these algorithms is that they do not ensure optimality in any form and might be computationally expensive since they deal with a discretized interpretation of the entire domain [4].

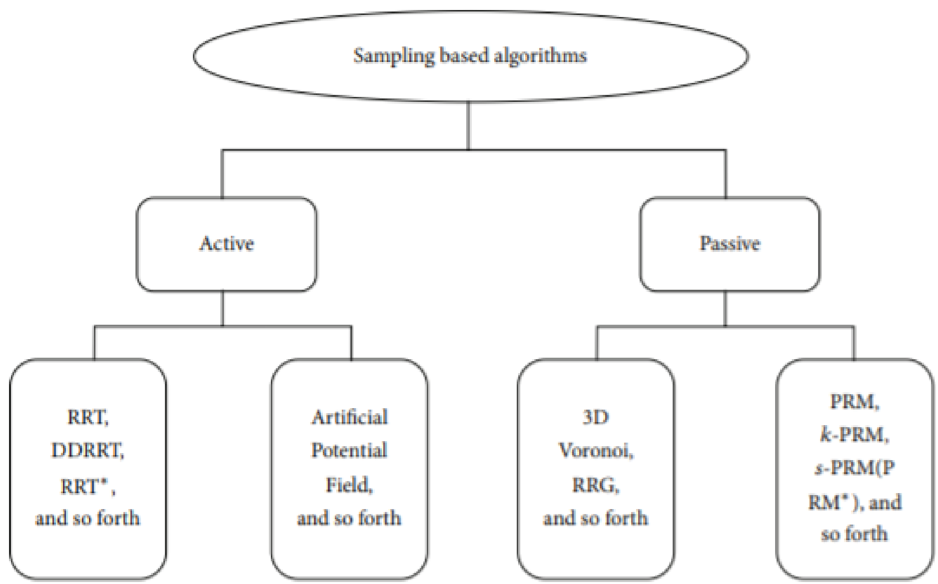

Figure 8 Sampling Based Algorithms [5]

Table 1 Sampling Base Algorithm Comparison

Method	Advantages	Disadvantages
RRT	Fast Searching Ability	Single path Non-optimal Static analysis
PRM	Good for complex environments	Expensive Static analysis Non-optimal
Voronoi	Collision-free and easy	Non-convergence Static analysis Incomplete representation
Potential Fields	Fast convergence	Local minima

The second category includes node-based optimal algorithms such as Dijkstra [9], A-star [10], D-star [9], and many others. Such algorithms require a pre-discretized and known domain, they associate each step with a cost that can be incorporated as a generalized cost-function [4], but they are computationally expensive and might not always lead to global optimality [4].

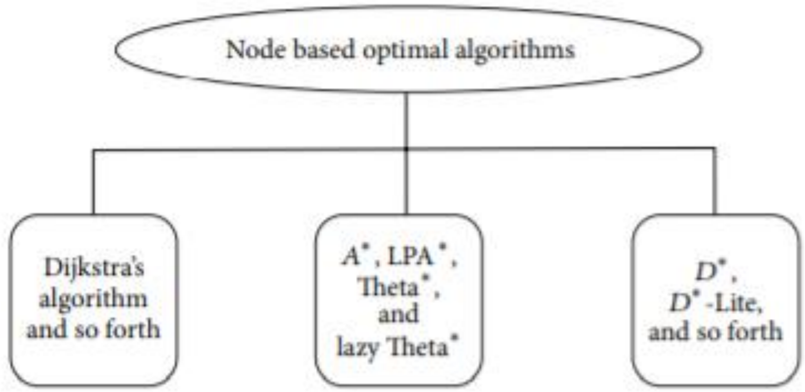

Figure 9 Node Base Path Planning Algorithms

Table 2 Node Based Method Comparison [5]

Method	Advantages	Disadvantages
Dijkstra	Easy to implement	High time complexity Static Analysis
A-star	Fast search	Non-smoothness Static Analysis
D-star	Fast Search Dynamic Environment	Unrealistic distance

The third category includes mathematical model-based algorithms, which formulate the entire domain and cost function into a set of mathematical equations, in addition to several constraints such as initial and final conditions as well as inequalities and differential equations [4]. If the problem can be linearized, it is solved via linear programming [4]; otherwise optimal control theory [11] is used, which tends to be computationally expensive, may not be able to ensure obstacle avoidance, but has the advantage of reaching global optimality.

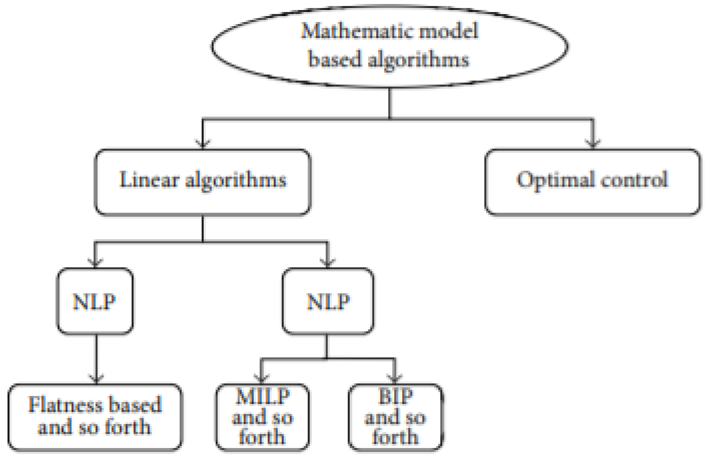

Figure 10 Mathematical Path-planning Algorithms [5]

The fourth category includes bio-inspired algorithms [4], which leverage the bio-mimicry theory of applied mathematics and rely on stochastic approaches. Bio-inspired algorithms include genetic algorithm [12], ant colony optimization [13], neural networks [14], and others. These algorithms can achieve obstacle avoidance, but do not ensure global optimality. They are computationally expensive especially since they are solving an NP-Hard problem [4].

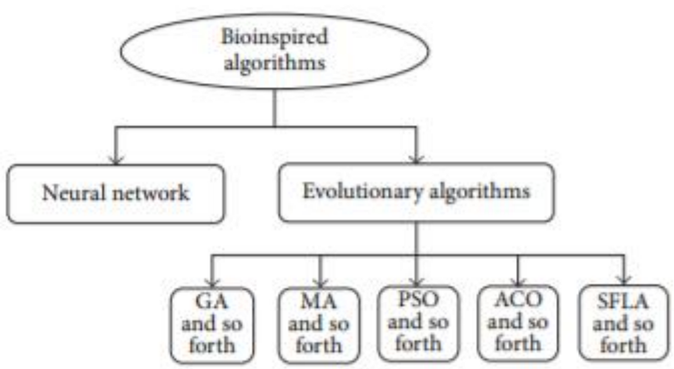

Figure 11 Bio-inspired Path-planning Algorithms [5]

Table 3 Bio-inspired Algorithm Comparison

Method	Advantages	Disadvantages
Genetic Algorithm	Can deal with multi-object problems	Premature convergence
Ant Colony	Continuous planning and multiple objects	High time complexity
Neural Network	Stable again sudden changes	Relies on set rules ad organisms

The fifth category includes mixed algorithms [4], which integrate other categories such as RRT, A-star, and others, with the aim of reaping the benefits of the different categories and overcoming their disadvantages [15].

2. *Travelling Salesman Problem*

For visiting a series of waypoints, node-to-node planners do not suffice, but rather a global planner is required, which is where graph theory comes at hand [16]. The primary aspect of graph theory is the combination of nodes and edge, where a node represents a destination that the salesman (or quadrotor) must reach. An edge represents the connection between two different nodes and includes an associated weight [17]. The traveling salesman problem (TSP) can be formulated using graph theory to produce a global plan for visiting all waypoints. It is important to note that the generalized case of a traveling salesman problem is the asymmetric traveling salesman problem (ATSP), which arises when a pair of nodes do not have the same cost for the edge between them when directed in opposite directions, which results in an asymmetric adjacency matrix [17]. The problem of having an aerial drone visit each tree in a forest lends itself to an asymmetric cost due to obstacles, and especially when it comes to the difference in altitudes that affect energy consumption of quadrotors.

There are several forms of solving the ATSP, which include symmetrizing the adjacency matrix of the ATSP [18] and then using Christofide's algorithm [19] to find a solution. This starts with a minimum spanning tree algorithm [20], but since this creates a single route with branches, a tour cannot be made. Taking recourse to graph theory, the handshake-lemma can be utilized to create a full tour [19]. However, this leads to a requirement that each node should have an even number of edges connected to it. To mitigate this issue, all nodes with the odd number of edges are grouped and the minimum-matching algorithm is used to pair them whilst having a minimum total cost [19]. After having even edges for all nodes, a Eulerian tour could be established, which dictates that all edges should be visited only once. This, however, leads to a problem where nodes are visited several times, which requires it to undergo a cut-off phase where any repeated node is deleted from the sequence thus creating a final Hamiltonian tour [19].

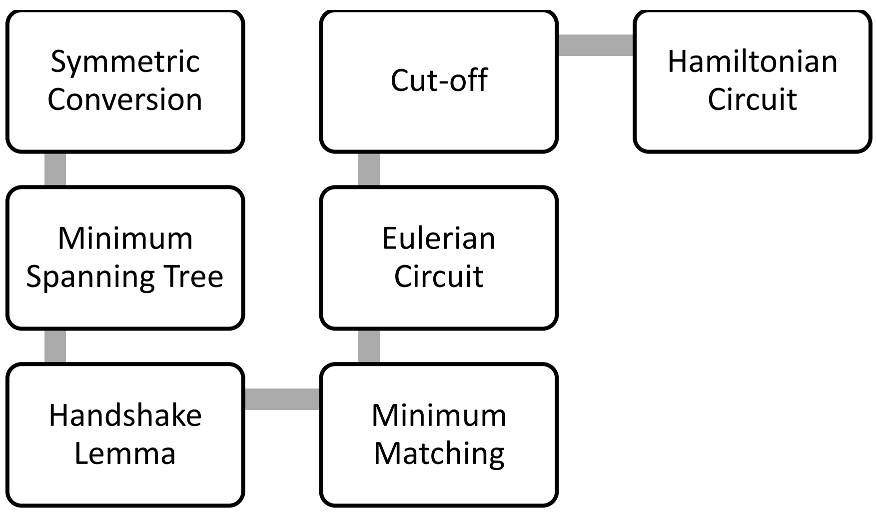

Figure 12 Christofide's algorithm sequence for asymmetric TSP [20]

Another method of solving the ATSP is the Genetic Algorithm, which is based on a cycle approach [21] that starts with a random set of strings representing a series of numbers, which represent a suggested order of node visits. This pool of suggested random items undergoes a fitness test where the ones that pass go on to the next stage [22]. The next stage is composed of mutations and crossovers where a combination of strands that are deemed "fit" is produced. This creates a new "generation" that is added to a new set of randomly generated suggestions to undergo the same cycle. The algorithm ends if the pre-set number of cycles is reached, or if the minimized strand is repeated several times [21].

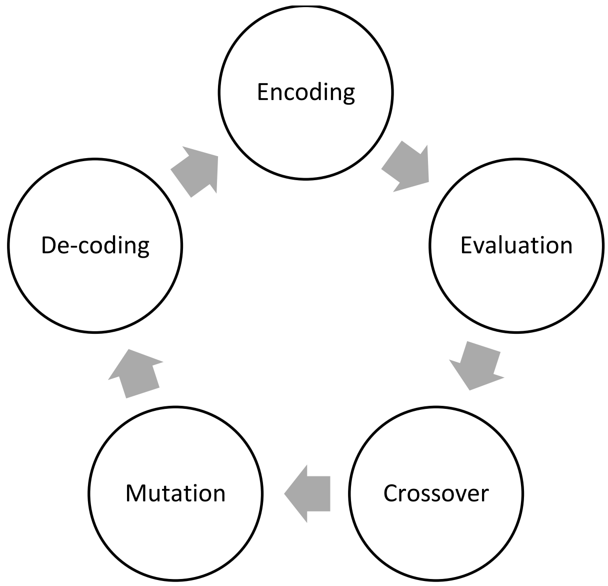

Figure 13 Genetic Algorithm Cycle

Another method is Integer Linear Programming (ILP) with the addition of Sub-tour Elimination Constraints (SEC) [23], which can generally be solved using branching and bounding. These methods vary in computational time and accuracy, with ILP having higher computational cost but guaranteeing global optimality [23].

3. *Transcription and Discretization*

When using optimal control theory to solve the path-planning or trajectory tracking problems, it is important to maintain convergence [24]. Optimal control solvers might not always result in an answer, but when they do, the obtained solutions are surely optimal. This results in an added burden to properly select the method of solving the problem at hand.

Transcription is transforming the set of minimizing equations, constraints, differential equations, and path constraints into a set of nonlinear equations with inequality and equality constraints [25]. This could be done in two different ways that are known as direct and indirect methods. Direct methods are a set of methods where discretization occurs first and then minimization of the object function is sought after [26]. On the other hand, an indirect method includes transcription at first but continues to find the location where the derivative of the objective function is zero.

Table 4 OCP Solver Methods

Indirect	Direct
"Optimize then Discretize"	"Discretize Then Optimize"
More Accurate	Less Accurate
Harder to pose and solve	Easier to pose and solve

After deciding on the general transcription approach, it is important to choose the transcription method itself [27]. There are two general transcription methods: collocation and shooting. The shooting method is similar to a target shooting, where explicit discretization schemes are developed in simulation [27]. This can also be further elaborated upon by using multiple shootings where the domain is split into smaller parts and trial simulations with error estimations are used to find a solution. On the other hand, collocation methods are based on function approximation using implicit forms of integration discretization such as Runge-Kutta. [27]

Table 5 OCP Transcription Methods

Shooting Methods	Collocation Methods
Based on simulation	Based on function approximation
Problems with simple control	Problems with complicated control
No path constraints	Path constraints

After choosing the transcription method, a discretization method is needed. There are several methods for discretizing the domain, which are mainly separated into two main categories that include the h-methods and the p-methods, and their combinations [28]. The h-method generally works on the concept that any domain, if segmented into smaller and smaller parts, will lead to a better and more optimal solution. It has a lower order for estimation in each trajectory and the whole domain is stitched back together to yield the final solution [28]. One of the most significant methods of the h-type is the Hermite Simpson method where the states are represented using a cubic-Hermite spline (third-degree polynomial interpolation), while the dynamics are satisfied using Simpson Quadrature (a numeric approximation of integrals) [29]. The Simpson Quadrature rule is used to estimate the integrated function with the following generalized formula [29]:

$$\int_a^b f(x)dx = \frac{h}{3}(f(a) + (f(b) + \frac{2h}{3}\sum_{k=1}^{n-1} f(x_{2k}) + \frac{4h}{3}\sum_{k=1}^{n} f(x_{2k-1})), \quad (1)$$

$$s.t. h = \frac{b-a}{2n}.$$

This is also obtained where *n* is the number of proposed segments and *h* is the time segments where as *a* and *b* are the initial and final times respectively. Moreover, it is accompanied by a Hermite spline polynomial fit for the states that are described as follows [29]:

$$x(t) = a_0 + a_1 t + a_2 t^2 + a_3 t^3, \quad (2)$$

$$\dot{x}(t) = a_1 + 2a_2 t + 3a_3 t^2. \quad (3)$$

In this assumption, *t* represents time that is between 0 and *m*, which is considered the final time. However, in order to determine the coefficients, the following Hermite rule is used:

$$\begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ -\frac{3}{h^2} & -\frac{2}{h} & \frac{3}{h^2} & -\frac{1}{h} \\ \frac{2}{h^3} & \frac{1}{h^1} & -\frac{2}{h^3} & \frac{1}{h^2} \end{bmatrix} \begin{bmatrix} x(0) \\ \dot{x}(0) \\ x(m) \\ \dot{x}(m) \end{bmatrix}. \quad (4)$$

In the H-S collocation scheme, the collocation point is determined midway within the time domain. The derivative of the state at this point should be equal to the right side of the ordinary differential equations (ODEs) of the OCP. The collocation derivative is expressed as follows:

$$\dot{x}_c = \dot{x}\left(\frac{m}{2}\right) = -\frac{3}{2m}(x_m - x_{m+1}) - \frac{1}{4}[f(x_k, u_k) + f(x_{k+1}, u_{k+1})], \quad (5)$$

$$u_c = \frac{u_k + u_{k+1}}{2}, \quad (6)$$

where *f(x)* is the equation at the right-hand side of the ODE and *u* is the input. The equality equation results in the following [29]:

$$\dot{x}_c - f(x_c, u_c) \approx 0, \quad (8)$$

$$x_k - x_{k+1} + \frac{m}{6}[f(x_k, u_k) + 4f(x_c, u_c) + f(x_{k+1}, u_{K+})] \approx 0. \quad (9)$$

These result in a set of nonlinear equations and finalize as a nonlinear problem (NLP) with the following formulation:

$$\begin{gathered} min.\, \Phi(\boldsymbol{X}, \boldsymbol{U}) \\ \boldsymbol{s.t.} \dot{\boldsymbol{X}} - \boldsymbol{F}(\boldsymbol{X}, \boldsymbol{U}) = \boldsymbol{0} \\ \boldsymbol{X_0} = \boldsymbol{x_0} \\ \boldsymbol{X_f} = \boldsymbol{x_f} \end{gathered} \quad (10)$$

where Φ is the transcribed cost function and $\dot{\mathbf{X}} - \mathbf{F}(\mathbf{X}, \mathbf{U})$ is taken via the equations (8) and (9). $\mathbf{X}_0$ and $\mathbf{X}_f$ are the initial and final values of the said equations

On the other hand, p-methods consider the domain as a single entity and approximate it using higher and higher orders of polynomials for estimation. This creates smoother transitions, but it does not respect actuator or input saturation [28].

Table 6 OCP Collocation Discretization Methods

h-methods	p-methods
A higher number of segments	Single-segment
Low-order estimation	Higher order estimation
Converges by increasing segments	Converges by increasing method order

In general, hybrid methods of p-/hp-methods are used to ensure higher rates of convergence. One of the most significant methods of transcription is the pseudospectral method, which uses a high order estimation for highly nonlinear and coupled dynamics and problems, which also segments the domain into very small sub-parts to be able to solve the whole domain in a discretized manner while taking a global solver into consideration. One of the most significant forms of pseudospectral methods is the Legendre-Gauss-Radau (LGR) method, which is an extension of the Legendre-Gauss (LG) collocation [30].

The LGR method is generally obtained by primarily starting with Legendre polynomials that are used to estimate the entire domain. The Gaussian quadrature is used to discretize the domain, where time is discretized into several segments that might be separated in equal or adequately unequal distances [30]. It is important to note that the general LG method does not include solving for the end and starting states, which leaves the domain as an open interval, while adding the Radau factor results in a half-open domain where the initial states and time is determined using Radau collocation. This enables an open-ended approach to the problem, especially when dealing with free-end-time as an infinite horizon approach.

Primarily, the LGR transcription method starts with the Gauss Quadrature, which estimates the cost function as a sum series of equations, and it is optimal in the standard domain of [-1 1]. To transform the time domain from [0 t_f] to [-1 1], the following equation is used [30]:

$$t = \frac{t_f - t_0}{2}\tau + \frac{t_f + t_0}{2}, \quad s.t. \quad \tau \epsilon [-1, +1], \tag{11}$$

This helps discretize the time domain and thus sets the map straight to start discretizing the rest. After discretizing the domain, the Gaussian quadrature is based on discretizing the integral function into such a format:

$$\int_{-1}^{1} f(x)\, dx \approx \sum_{i=1}^{n} w_i f(x_i), \tag{12}$$

where the $f(x)$ is a polynomial of a degree $2n-1$ or less where n is the number of segments, while w is the weight associated with each function. Under LGR, the Gaussian quadrature function can be assumed as follows [30]:

$$f(x) = (1-x)^{\alpha}(1+x)^{\beta} g(x) \quad s.t. \; \alpha, \beta > -1, \tag{13}$$

where $g(x)$ is a low order degree approximation. However, in order to get accurate weights, Legendre polynomials are added using the following equation:

$$w_i = \frac{2}{(1-x_i^2)[P_n(x_i)]^2}, \tag{14}$$

$$P_n(x) = \frac{1}{2^n n!} \frac{d^n}{dx^n} (x^2-1)^n. \tag{15}$$

This reflects when dealing with the states and their ODE's. The states and inputs could be estimated using the following [30]:

$$x^N(\tau) = \sum_{i=0}^{N} x_i L_i(\tau), \tag{16}$$

$$L_i(\tau) = \prod_{\substack{i=0 \\ j \neq i}}^{N} \frac{\tau - \tau_j}{\tau_i - \tau_j}, \tag{17}$$

where the degree of the polynomial is at most N. Moreover, to estimate their derivatives the following equation is used:

$$\dot{x}^N(\tau) = \sum_{i=0}^{N} D_i x_i, \tag{18}$$

where is D called the Gauss Pseudoscpetral Differential equation where it is also considered the derivative of L at τ. This results in the following matrix when integrating the derivatives [30]:

$$\int_{-1}^{1} \dot{x}^N(\tau) d\tau = \sum_{i=1}^{N} w_i \dot{x}^N(\tau), \tag{19}$$

However, a small change to the Gaussian Quadrature is required to be able to determine the start and endpoint values. This is where the Radau quadrature comes in, which is formulated as follows:

$$\int_{-1}^{1} f(x)\, dx = w_1 f(-1) + \sum_{i=2}^{n} w_i f(x_i), \tag{20}$$

$$s.t. w_1 = \frac{2}{n^2} \; and \; w_i = \frac{1}{(1-x_i)[P_{n-1}(x_i)]^2}. \tag{21}$$

This results in the following formulation of the NLP:

$$\begin{gathered} min. \boldsymbol{\Phi}(\boldsymbol{X_N}) \\ s.t. \boldsymbol{DX} = \boldsymbol{F}(\boldsymbol{X^{LGR}}, \boldsymbol{U^{LGR}}) \\ \boldsymbol{X_0} = \boldsymbol{x_0} \end{gathered} \tag{22}$$

where Φ is the transcribed cost function, $\boldsymbol{D}$ is the Gaussian Pseudospectral Differential Equation Matrix, $\boldsymbol{F}$ is the matrix of discretized and nonlinear equations, and $\boldsymbol{X_0}$ is the initial states matrix. This, however, cannot be solved using conventional methods and requires nonlinear solvers [30].

4. Solvers

After transcribing and discretizing the domain, states, equations, ODEs, and constraints into a set of nonlinear equations that must be solved. This may vary between linearizing the equations or creating Hessian or Jacobin Matrices to solve the nonlinear problem (NLP). One of the more significant forms of solving NLPs is using sequential quadratic programing (SQP), which is generally used to solve lower order problems with a small set of differential equations [31]. SQP linearizes the system and solves using sequential iterations, which may have a high rate of failure when it comes to constraints of high orders including equation, inequality, and state constraints [31].

However, it is important to note that SQP could be used to solve NLP in a more accurate manner, if it is used within a sparse nonlinear optimizer (SNOPT) [32]. This optimization method starts with solving a quadratic model based on an initial guess and starts to iteratively update the solution. It undergoes inexpensive iterations and works without violating constraints. It is important to note that using SNOPT requires more and more constraints since iterations to eliminate sporadic solutions. SNOPT is efficient with highly constrained problems, it can greatly exploit initial guesses, and it needs less evaluations for solving functions and detects infeasibility [32].

On the other hand, another solution method is known as the interior point method (IPM) or interior point optimizer (IPOPT) [33]. IPOPT uses a Newtonian iteration method with a Karush–Kuhn–Tucker (KKT) system to relax the solution. It works in an iterative manner by updating the solution and estimating a relaxation parameter. It generally performs fewer expensive iterations, it heavily relies on linear algebra, it is highly efficient for coupled and nonlinear programs, and it includes simpler interfaces. This type of solvers is generally efficient with higher orders of derivatives, especially second order, which is prevalent in the case under consideration in this work.

One of the most important parts of IPM is eliminating nonlinear inequalities, which is achieved by using a dummy variable instead [34]:

$$g(x) > b \ \Rightarrow \begin{cases} g(x) - b - s = 0 \\ s > 0 \end{cases} \tag{23}$$

This leads to additional equality constraints. The second step is to get rid of the dummy state, which is done using a Natural Logarithm barrier term as an additional constraint to the system, which causes the minimized function to increase in value if approaching the infeasible region based on the inequality constraints [34]:

$$\begin{matrix} \min f(x) \\ x > 0 \end{matrix} \Rightarrow \min f\ (x) - \mu \sum_{i=1}^{n} \ln(x_i), \tag{24}$$

where μ is a varying parameter that is iteratively chosen in order find the optimal solution. However, rather than constantly changing μ, the simplest way is to check where this condition reaches minimum by determining where its derivative is equal to zero. This is where the KKT condition is used, which is stated as follows [33]:

$$\begin{matrix} \min f\ (x) - \mu \sum_{i=1}^{n} \ln(x_i) \\ c(x) = 0 \end{matrix} \Rightarrow \begin{matrix} \nabla f(x) + \nabla c(x)\lambda - \mu \sum_{i=1}^{n} \frac{1}{x_i} = 0 \\ c(x) = 0 \end{matrix} \tag{25}$$

where λ is known as a KKT multiplier. Moreover, it is safe to assume a variable as $z_i = \frac{\mu}{x_i}$ that results in the following KKT finalized condition:

$$\begin{aligned} \nabla f(x) + \nabla c(x)\lambda - z = 0 \\ c(x) = 0 \\ XZe - \mu e = 0 \end{aligned} \tag{26}$$

where e is an array of ones while X and Z are the matrices of x's and z's. This results in the generalized KKT solution with Newton-Raphson [34]:

$$\begin{bmatrix} W_k + \Sigma_k & \nabla c(x_k) \\ \nabla c(x_k)^T & 0 \end{bmatrix} \begin{pmatrix} d_k^x \\ d_k^\lambda \end{pmatrix} = -\begin{pmatrix} \nabla f(x_k) + \nabla c(x_k)\lambda_k \\ c(x_k) \end{pmatrix}, \tag{27}$$

$$s.t. \Sigma_k = X_k^{-1} Z_k, \tag{28}$$

$$Z_k = \begin{bmatrix} z_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & z_n \end{bmatrix}, \tag{29}$$

$$X_k = \begin{bmatrix} x_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & x_n \end{bmatrix}, \tag{30}$$

$$W_k = \nabla_{xx}^2 (f(x_k) + c(x_k)^T \lambda_k - z_k), \tag{31}$$

$$d_k^z = \mu_k X_k^{-1} e - z_k - \Sigma_k d_k^x. \tag{32}$$

This results in a coefficient matrix form of Ax=b where W_k is the second gradient of the Lagrangian, d's are directions that are used in the iterations, and n is the number of variables present. However, the whole d for x, z, and λ is not used within the progression as iterations are made, but rather there is a step size, α, which is used for iterations and are the values iterated as follows:

$$x_{k+1} = x_k + \alpha_k d_k^x, \tag{33}$$

$$z_{k+1} = z_k + \alpha_k d_k^z, \tag{34}$$

$$\lambda_{k+1} = \lambda_k + \alpha_k d_k^\lambda. \tag{35}$$

This is iterated several times until the number of iterations is reached or the following KKT error tolerance level is reached [34]:

$$
\begin{array}{c}
max|\nabla f(x) + \nabla c(x)\lambda - z| \leq \epsilon_{error} \\
max|c(x)| \leq \epsilon_{error} \\
max|XZe - \mu e| \leq \epsilon_{error}
\end{array} \quad . \tag{36}
$$

C. Thesis Objectives:

For the quadrotor UAV to be able to roam around the whole forest in an energy-optimal manner, the problem must undergo several steps that inlcude:

1. Identifying an appropriate optimal control solver that includes a suitable transcription method along with a capable nonlinear solver.
2. Ensuring solution convergence and adequate object avoidance when solving the optimal control problem, whilst respecting downwash restrictions and quadrotor dynamics, along with time and energy optimality.
3. Establishing an optimal tour by solving the TSP using ILP in addition to SECs.
4. Computing a probability of infection for each tree based on prior scans and identification of previously infected trees.
5. Generating a time-varying path, as the quadrotor scans new trees and updates the probability online, by solving the fixed-start fixed-end TSP.

CHAPTER II

PROBLEM FORMULATION

Mapping the environment is a key aspect of the path-planning problem, especially when working offline. For forests considered in this work, the map is considered constant since the environment is not dynamic in the sense that sudden changes are not expected to occur, even at a span of a decade. For mapping a forest, we start by modelling the trees and drones, and establishing zones of operation.

A. Trees

Due to the relatively small change in trunk radius of trees with respect to their height, it is safe to consider a tree trunk as a tall cylinder. However, it is important to note that the radius and the height of a tree trunk is generally governed by a slenderness equation, which is specific to each tree. For Turkish Pines, the following slenderness equation is used [35]:

$$H = 2.4 + 0.45(DBH) - 0.0045(DBH)^2, \tag{37}$$

For Lebanese Cedars, the following slenderness equation is used [36]:

$$H = 1.35(DBH)^{0.72}, \tag{38}$$

For the Stone Pines, the following slenderness equation is used [37]:

$$H = 25 \times DBH. \tag{39}$$

Since the trees at hand are usually tall and have barren trunks, it is safe to presume their trunks as cylinders. Moreover, since trees such as Stone Pines, Turkish Pines, and even Lebanese cedars do not allow peripheral plantations to grow under or around them, it is safe also to presume that the forest at hand is a group of the same trees, which need to be scanned with no objects between them but other trees of the same kind.

B. Quadrotor Downwash Effect

Quadrotors function using aerodynamic thrust forces generated by their propellers. When objects are in the vicinity of a quadrotor, they tend to interfere with the free flow of air through its rotors and surrounding its frame. Quadrotors should thus remain at a certain distance from nearby objects to avoid aerodynamic interference. Due to this aerodynamic nature of quadrotors, they can be generally identified with their downwash ellipsoid, as shown in Figure 14, which tends to simplify the graphical representation of quadrotors [38].

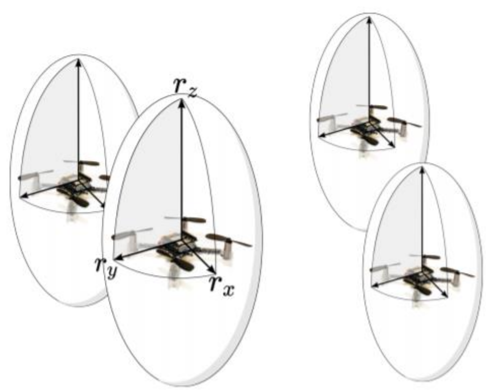

Figure 14 Quadrotor Downwash Inflation

The ellipsoid's radii are equivalent to the quadrotor's chassis dimensions including the radius of the rotor fan, and the downwash radius, which is placed on the vertical body-fixed axis, is calculated using aerodynamic analysis that is specific to the quadrotor under consideration.

C. Zoning

To find out how a quadrotor should be located in order to accurately scan the tree, but yet still satisfy the downwash condition, a zoning scheme is devised. On one hand, the downwash effect requires quadrotors to stay away from nearby objects within a specified distance, on the other hand the intended usage of quadrotors to scan trees via on-board cameras requires them to be within a specified depth based on their resolution. Thus, a zoning of permissible locations, which the quadrotor should operate in, must be established.

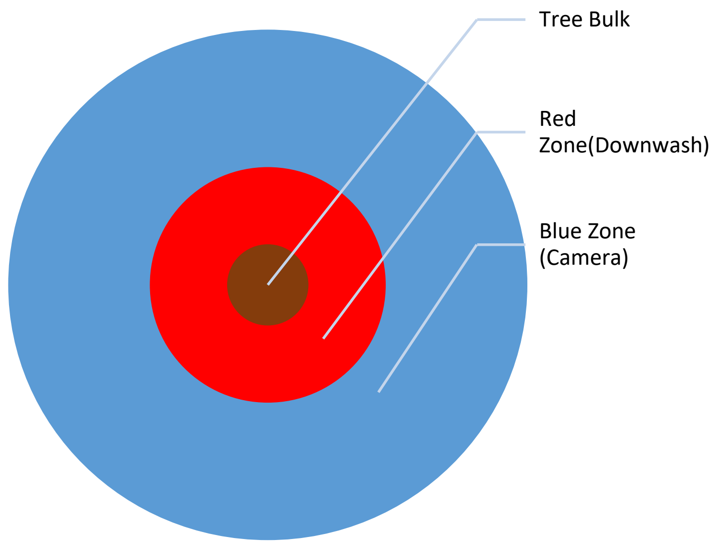

Figure 15 Layers of Cylindrical Zoning

1. *Red Zone*

To accommodate the aerodynamic constraints related to downwash, a no-go zone (red zone) is established around each objective and obstacle. This results in an offset or inflation of cylindrical zone around each object by a distance equal to the quadrotor downwash radius, which could be considered equivalent to the chassis of the quadrotor in addition to the propeller radius in the horizontal plain, and another equal to the downwash radius in the vertical direction, as shown in Figure 15 [38].

2. *Blue Zone*

Since the quadrotors will be equipped with digital cameras for scanning and computer vision purposes, a maximum distance is specified to guarantee accurate scanning of the trees with adequate resolution. This criterion results in a cylinder that is concentric with the tree trunk having a radius equal to the maximum distance allowed for the camera to accurately scan and detect the moths [39].

D. Problem Statement

The main aim of this paper is to create an energy-optimal path to visit every tree and come back to the base. This can be mainly separated into two main parts. The first one is concerned with determining the energy-optimal trajectory and path to travel from one point to another between trees. The second part is determining an energy optimal tour to visit all trees. In this formulation, the trees are always visited from the west where the quadrotor is looking towards the east as shown in the Figure16. This should be done while taking into consideration collision avoidance, in addition to respecting downwash and camera restrictions.

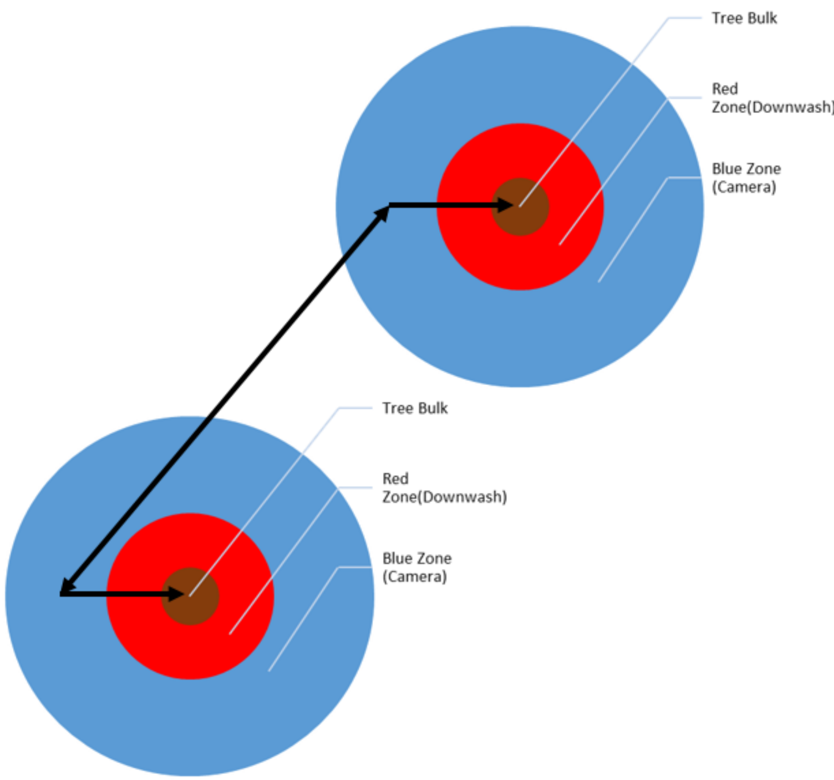

Figure 16 Tree Scanning in Goldilocks Zone and Path-Planning Scheme

CHAPTER III

TREE-TO-TREE PATH PLANNING

A. Transformation Axes

To determine the energy-optimal trajectory and path for quadrotors, the equations of motion that govern their dynamics must be first formulated.

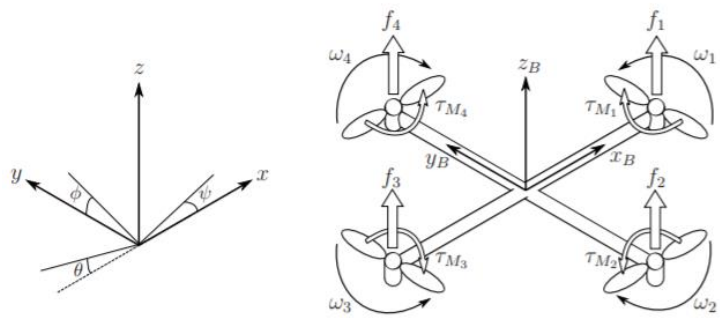

Figure 17 Quadrotor Transformation and rotor direction

Based on the Euler transformation from the global XYZ axes to the local quadrotor's XYZ axes, the following transformation is used and expressed as follows [40]:

$$R = \begin{bmatrix} c_\phi c_\psi - c_\theta s_\phi s_\psi & -c_\psi s_\phi - c_\phi c_\theta s_\psi & s_\theta s_\psi \\ c_\theta c_\psi s_\phi + c_\phi s_\psi & c_\phi c_\theta c_\psi - s_\phi s_\psi & -c_\psi s_\theta \\ s_\phi s_\theta & c_\phi s_\theta & c_\theta \end{bmatrix} \quad (40)$$

The generalized equation is based on the roll, pitch, and yaw motions that are represented by ϕ, θ, and ψ respectively.

B. Motors

The motors used in quadrotors are usually brushless DC motors, which generally represent a circuit like that of an RL circuit**.**

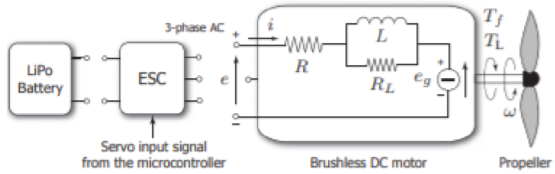

Figure 18 Circuit Representation of a DC- Brushless Motor

The current and potential difference across the motor is expressed as follows [11]:

$$i = \frac{1}{K_T}\left[T_f + T_L\omega + D_f\omega + J\dot{\omega}\right], \tag{41}$$

$$e = R_m i + K_E\omega + L\frac{di}{dt}, \tag{42}$$

where ω is the angular velocity of the motor, K_T is the torque constant, T_f is the motor friction torque, T_L is the speed-dependent friction torque, D_f is the viscous damping coefficient, J is the total inertia of the motor, R_m is the motor resistance, K_E is the back electromotive force constant, and L is the inductance of the motor [41].

E. Forces

To control the altitude and attitude of the quadrotor, generalized forces are present to govern these actions. These forces are either induced using the rotors of the quadrotor or present due to other factors such as gravity, wind, or other external disturbances [42].

The rotors on the quadrotor rotate in pairs where one pair of opposing rotors rotate counterclockwise and the other pair rotates clockwise. Variations and combinations of the speed of each rotor govern the direction in which the quadrotor moves [43].

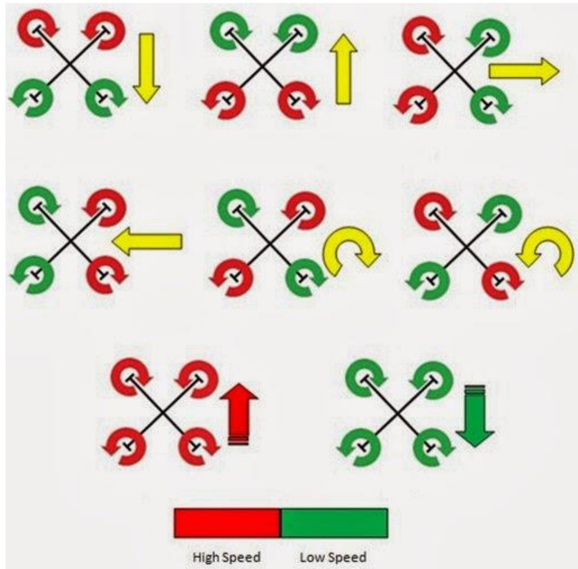

Figure 19 Motor rotation quadrotor combination and motion control

The generalized forces that act on the quadrotor are mostly thrust, gravity, and extraneous forces. To determine the value of the thrust, the thrust of each rotor should be investigated using energy equations. After simplifying the equation of power:

$$P=IV, \tag{43}$$

and after substituting each with its associated equations, it is safe to conclude that the new equation for power from the motor is given by [44]:

$$P = K\tau\omega, \tag{44}$$

where K is a coefficient that can be determined by the internal circuitry of the motor. On the other hand, using the conservation of energy principle, power can be expressed using the following equation [42]:

$$P = Tv_h, \tag{45}$$

where T is the thrust and v_h is the air velocity. However, due to momentum, velocity and thrust have a separate relation, which results in the following equation [44]:

$$v_h = \sqrt{\frac{T}{2\rho A}}, \tag{46}$$

where A is the swept area of the rotor and ρ is the air density. After substituting the equations (43) to (46), the resultant thrust force of each motor can be estimated as follows [40]:

$$T = k\omega^2, \tag{47}$$

where k is an appropriately dimensioned constant, leading the whole thrust force acting on the quadrotor to be the sum of all individual thrusts from each rotor:

$$T_{total} = k\begin{bmatrix} 0 \\ 0 \\ \sum_{i=1}^{4} \omega_i^2 \end{bmatrix}. \tag{48}$$

F. Torques

As there is a thrust force acting parallel to the rotor shaft, there is a perpendicular force acting as well. This is expressed as the drag force:

$$F_D = \frac{1}{2}\rho C_D A(\omega r)^2 \tag{49}$$

where C_D is the drag coefficient and r is the length of each propeller. This leads to the associated torque as follows:

$$\tau_B = \begin{bmatrix} lk(\omega_1^2 - \omega_3^2) \\ lk(\omega_2^2 - \omega_4^2) \\ b_t(\omega_1^2 - \omega_2^2 + \omega_3^2 - \omega_4^2) \end{bmatrix}, \tag{50}$$

where b_t is an appropriately dimensioned constant and l is the distance from the center of the quadrotor to any rotor [44].

G. Equations of motion

After compiling the equations (43) through (50), it is safe to say that the governing inputs, which induce the quadrotor motion, are as follows [11]:

$$\begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \\ u_5 \end{bmatrix} = \begin{bmatrix} k(\omega_1^2 + \omega_2^2 + \omega_3^2 + \omega_4^2) \\ k(\omega_2^2 - \omega_4^2) \\ k(\omega_3^2 - \omega_1^2) \\ b_t(\omega_1^2 - \omega_2^2 + \omega_3^2 - \omega_4^2) \\ \omega_1 - \omega_2 + \omega_3 - \omega_4 \end{bmatrix}. \tag{51}$$

The generalized equations of motion rely on the following basic Newton's equations. [41]

$$\sum F = ma, \tag{52}$$

$$\sum M = I\ddot{\theta}. \tag{53}$$

These result in the following set of equations of motion for the quadrotor:

$$m\ddot{x} = (\sin\phi \sin\psi + \cos\phi \cos\psi \sin\theta)u_1, \tag{54}$$

$$m\ddot{y} = (\cos\phi \sin\theta \sin\psi - \cos\psi \sin\phi)u_1, \tag{55}$$

$$m\ddot{z} = (\cos\theta \cos\phi)u_1 - mg, \tag{56}$$

$$I_x\ddot{\phi} = \left(I_y - I_z\right)\dot{\theta}\dot{\psi} + lu_2 - J\dot{\theta}u_5, \tag{57}$$

$$I_y\ddot{\theta} = (I_z - I_x)\dot{\phi}\dot{\psi} + lu_3 + J\dot{\phi}u_5, \tag{58}$$

$$I_z\ddot{\psi} = \left(I_x - I_y\right)\dot{\phi}\dot{\theta} + u_4. \tag{59}$$

H. Energy Consumption

Since the optimization in this work is based on energy optimality, it is of great importance to determine the proper cost function to guide the path planning process. Energy consumption is one of the most important factors when it comes to planning an efficient quadrotor path. This is calculated using the following generalized equation [11].

$$E_{total} = \int_{t_o}^{t_f} P_{total} dt = \int_{t_o}^{t_f} \sum_{i=1}^{4} e_i(t) i_i(t)\, dt. \tag{60}$$

After combining the two equations of current and voltage in equations (41) and (42), we obtain the following equation after expanding and reducing:

$$E_{total} = \int_{t_o}^{t_f} \sum_{j=1}^{4} \left[c_1 + c_2\omega_j + c_3\omega_j^2 + c_4\omega_j^3 + c_5\omega_j^4 + c_7\dot{\omega}_j^2\right] dt. \tag{61}$$

It is safe to assume that the final state of the rotor speed is the same as the original speed. This is logical since the quadrotor is in hovering mode in both cases as it leaves the previously scanned area and reaches the next scanning area, where each constant is defined as follows [11]:

$$c_2 = \frac{T_f}{K_T}\left(\frac{2R_m D_f}{K_T} + K_E\right),\ c_3 = \frac{D_f}{K_T}\left(\frac{R_m D_f}{K_T} + K_E\right) + \frac{2R_m T_f b_t}{K_T^2},\ c_4 = \frac{b_t}{K_T}\left(\frac{2R_m D_f}{K_T} + K_E\right), c_1 = \frac{R_m T_f^2}{K_T^2}, c_5 = \frac{R_m b_t^2}{K_T^2}, c_7 = \frac{R_m J^2}{K_T^2}.$$

I. Optimal Control Problem (OCP)

To determine the energy-optimal trajectory of the quadrotor, referring to the formulation in, a set of ordinary differential equations is established to link the equations of motion to the motor speeds by defining the following states [41]:

$$x_1 = x, x_2 = \dot{x}, x_3 = y, x_4 = \dot{y}, x_5 = z, x_6 = \dot{z}, x_7 = \phi, x_8 = \dot{\phi}, x_9 = \theta, x_{10} = \dot{\theta}, x_{11} = \psi, x_{12} = \dot{\psi}, x_{13} = \omega_1, x_{14} = \omega_2, x_{15} = \omega_3, x_{16} = \omega_4.$$

This results in the following set of ordinary differential equations [41]:

$$\dot{x}_1 = x_2, \tag{62}$$

$$\dot{x}_2 = \frac{k}{m_{quad}}(\sin x_7 \sin x_{11} + \cos x_7 \cos x_{11} \sin x_9) \textstyle\sum_{k=13}^{16} x_k^2, \tag{63}$$

$$\dot{x}_3 = x_4, \tag{64}$$

$$\dot{x}_4 = \frac{k}{m_{quad}}(\cos x_7 \sin x_9 \sin x_{11} - \cos x_{11} \sin x_7) \textstyle\sum_{k=13}^{16} x_k^2, \tag{65}$$

$$\dot{x}_5 = x_6, \tag{66}$$

$$\dot{x}_6 = \frac{k}{m_{quad}}(\cos x_9 \cos x_7) \textstyle\sum_{k=13}^{16} x_k^2 - g, \tag{67}$$

$$\dot{x}_7 = x_8, \tag{68}$$

$$\dot{x}_8 = \left(\frac{I_y - I_z}{I_x}\right) x_{10} x_{12} + \frac{lk}{I_x}(x_{14}^2 - x_{16}^2) - \frac{J}{I_x} x_{10}(x_{13} - x_{14} + x_{15} - x_{16}), \tag{69}$$

$$\dot{x}_9 = x_{10}, \tag{70}$$

$$\dot{x}_{10} = \left(\frac{I_z - I_x}{I_y}\right) x_8 x_{12} + \frac{lk}{I_y}(x_{15}^2 - x_{13}^2) + \frac{J}{I_y} x_8(x_{13} - x_{14} + x_{15} - x_{16}), \tag{71}$$

$$\dot{x}_{11} = x_{12}, \tag{72}$$

$$\dot{x}_{12} = \left(\frac{I_x - I_y}{I_z}\right) x_8 x_{10} + \frac{b_t}{I_z}(x_{13}^2 - x_{14}^2 + x_{15}^2 - x_{16}^2), \tag{73}$$

$$\dot{x}_{13} = \alpha_1, \tag{74}$$

$$\dot{x}_{14} = \alpha_2, \tag{75}$$

$$\dot{x}_{15} = \alpha_3, \tag{76}$$

$$\dot{x}_{16} = \alpha_4, \tag{77}$$

where α_1 through α_4 are the rotor accelerations that are considered as the control inputs to solve the optimal control problem. The Lagrangian energy term to be minimized is the energy equation, while the Meyer term is the final time to be minimized which is considered an addition non-integral minimization factor. This is especially important since this is a free-end-time OCP.

There are certain constraints that need to be added based on the physical properties of the quadrotor itself relative to maximum pitch and roll angles and rotor speeds, which are formulated as follows [11]:

$$|x_7| \leq \frac{\pi}{2}, \tag{78}$$

$$|x_9| \leq \frac{\pi}{2}, \tag{79}$$

$$0 \leq \omega \leq \omega_{max}.$$

However, in the case under consideration, it is important to note that constraints are not only restricted to states, but also extend to surrounding constraints that include the downwash and object avoidance restrictions of tree trunks. Due to the relatively small size of the quadrotor, it is safe to presume the tree as an infinite cylinder. This assumption results in the following formulation:

$$\sqrt{\left(x_1 - x_{0,tree}\right)^2 + \left(x_3 - y_{0,tree}\right)^2} > r_{tree} + r_{downwash}. \tag{80}$$

Figure 20 Pine Tree Forest from Below

J. Optimal Control Solver

There are several important aspects to be prioritized when selecting the appropriate solver for the optimal control problem under consideration. The primary aspect is that the solver must possess a high percentage of convergence, since the TSP requires the adjacency matrix to be filled by the OCP solver to generate an optimal tour. A secondary aspect of the solver is minimal violation of the defined constraints.

In order to satisfy the priorities, a pool of different types of transcriptions is required, with a hybrid system being the target choice given its discussed advantages. The optimal control solver that is adopted in this work is known as the Imperial College London Optimal Control Solver (ICLOCS) [45], which uses three main transcription methods: h-method based on the Hermite-Simpson direct collocation, p-/hp-method based on the pseudospectral LGR direct collocation, and an auto-direct collocation method that automatically uses a hybrid of both settings and uses appropriate methods at each iteration. The transcribed and discretized domain is then solved using an IPOPT solver [45].

After feeding the dynamics, states, constraints, stage cost (Lagrangian cost), and boundary cost (Mayer cost), each individual combination in the adjacency matrix is solved. This results in the OCP being solved between each pair twice, once in each direction. This results in a total number of n^2-n times to be solved. However, as mentioned before, in order to ensure that convergence, optimality, and constraint violation are respected, each item is solved three times (using three transcription methods) in order to choose the best option.

The choice of the solution is made by checking primarily for convergence, followed by a constraint violation check where the violation should not exceed the inflation placed on the path constraints, and finally the choice is placed upon the one attaining the lowest energy consumption. However, if all three results do not satisfy the first two conditions, the resulting value of the pair is placed as zero, which results in a total amount of solutions to $3(n^2\text{-}n)$ times.

K. Estimation of energy consumption

When solving for a large-scale forest, the computational cost increases by approximately $2n$-2 times to resolve the OCP for all trees. With each tree requiring around 30 seconds or more as a computational time to calculate optimal tours, this might result in an impractical solution from a time-consumption viewpoint. Even though the simulation is only executed once per forest environment, it might still require days/weeks of computation for large-scale applications. To solve the scalability issue, an estimate of the energy consumption with respect to the initial and final positions is developed in this section. In order to determine the order of the required polynomial fit, it is important to note the relationship between the coordinates and the energy consumption.

Primarily we can start with equations (62) to (77). Since the degree of ω in the second derivative equation is of degree two, it reaches four when integrating twice. However, since double integration results in a second order of the states related to forth order of ω, it leads to a second order relationship of ω with respect to the position coordinates (x, y, and z). On the other hand, taking into consideration that the energy equation (61) is of fifth order of ω, the correlation and estimate of the relationship between energy and the coordinates results in a 2.5^{th} order of the coordinates. Thus, it is safe to presume a third order polynomial fit.

Using this type of fit, the computation time is greatly decreased from 30 seconds to less than a second, however, there are errors associated with this assumption especially since it does not account for object avoidance and its associated cost. Moreover, there are peripheral energy consumption that cannot be prevalent in only coordinate representation. Thus, a polynomial fit that overestimates the value is recommended to account for additional energy consumptions and peripheral consuming factors.

CHAPTER IV

COMPLETE TOUR SOLUTION

After computing the route costs between all paired tree combinations, the values are fed into an adjacency matrix where each element represents the cost placed on a directed edge. The ATSP is formulated as an ILP problem using the following set of equations [23]:

$$ATSP\colon min \sum_{i \in N} \sum_{j \in N, i \neq j} c_{ij} x_{ij} \ (j \in N, j \neq i), \tag{81}$$

$$\sum_{j \in N, j \neq i} x_{ji} = 1 \qquad \forall i \in N, \tag{82}$$

$$\sum_{j \in N, j \neq i} x_{ij} = 1 \qquad \forall i \in N, \tag{83}$$

$$x_{ij} \in \{0, 1\} \quad \forall i, j \in N, i \neq j, \tag{84}$$

where c_{ij} is the cost of the edge going from i to j, x_{ij} is a binary variable that represents the sequence of visiting the trees, and i and j belong to a set of integers N. If i directly precedes j in the optimal solution, then $x_{ij} = 1$; otherwise $x_{ij} = 0$ [23].

Nevertheless, the above solution might result in sub-tours where groups of nodes that do not include the origin relate to each other in a separate sequence, which causes disconnections between the tours and a split within smaller groups. To eliminate sub-tours, additional constraints are applied such as the Miller, Tucker, Zemlin Sub-Tour Elimination Constraints (MTZ-SEC) given by [46]:

$$u_i - u_j + (|N| - 1)x_{ij} \leq (|N| - 2) \qquad i, j \in N - \{1\}, i \neq j, \tag{85}$$

$$1 \leq u_i \leq |N| - 1 \qquad \forall i \in N - \{1\}. \tag{86}$$

For any given sub-tour, the number of edges should not equal the number of nodes, thus SEC forces sub-tours that lack the original node to have one less edge than the number of nodes. This, however, is not applicable to the sub-tour that contains the first original node, since this tour expands to engulf other sub-tours, as some sub-tours get iteratively eliminated.

One of the most prominent methods of solving ILP problems are Heuristic methods. These methods include two approaches: one that adds relaxation constraints based on SEC, while the other is known as branching and bounding (B&B). B&B is basically subdivided into two main steps: branching and then bounding. The branching part splits the problem into several sub-problems that branch along several possibilities and try to find individual optima in each case to compare. This may be similar to brute force, but in order to prevent the large computational problems, the bounding aspect is added to prevent it from reaching brute-force status. It is thus important to prune the search space in order to minimize the search, which is done as follows [47]:

- If there is no solution for the function, set it to infinity.
- Initialize a set of possible optimal solutions that contains none of the variables of the problem.
- Each element is taken and compared to the optimal solution so far:
  - If the value of B which is the element value is less than the current optimum, the chosen element replaces it.
  - If the value of B is not less than the current optimum, a new branch B_i is chosen to replace the optimal value.
  - Compare the value of B_i to that of the optimal solution to check if its value is still higher, thus the lower bound in the said node is greater than the upper bound of the problem at hand, which results in the elimination of node B.

The result of the algorithm is a single optimal tour that includes the original first node, and its number of nodes equals to the number of edges.

After taking all the factors of the first part of the energy-optimal tour, it can be summarized using the following flowchart (this involves OCP solution along the TSP solution):

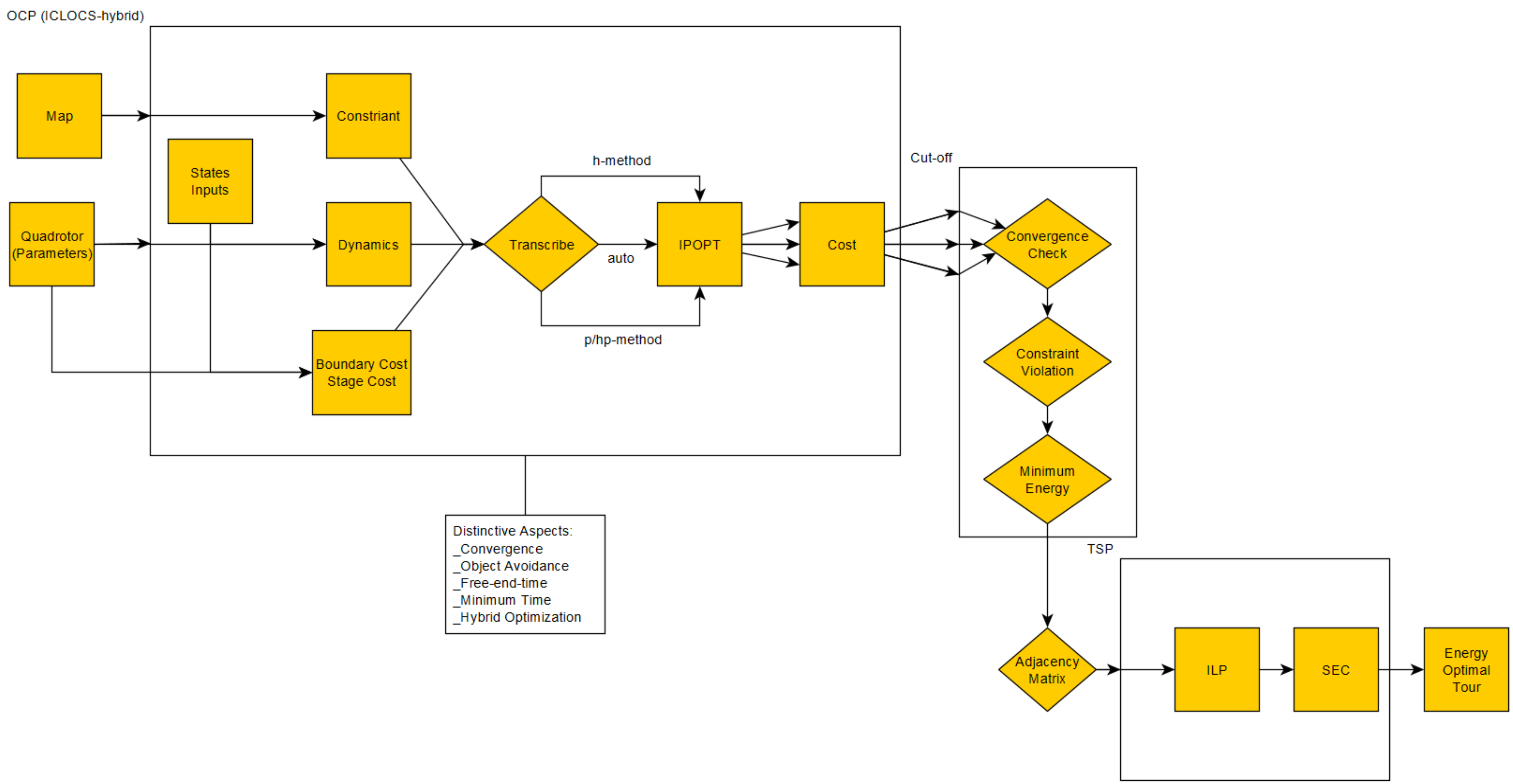

Figure 21 Energy-Optimal Tour Flowchart

CHAPTER V

NEW TOUR WITH PROBABILITY

Since the current situation of the moth-infected trees is alarming, it is important to note that the number of moths has reached the same number of pine trees present in Lebanese forests [48]. However, due to the vastness of the forests and the large number of trees, it is important to prioritize tree visits, which cannot be done using traditional TSP.

In order to maintain energy optimality while receiving the information needed, it is important to prioritize the said trees, which have a higher probability than the rest. Thus, it is safe to say that trees with lower chances of getting infected could be excluded from the set of nodes (trees) that need to be visited after an initial scan. However, in order to presume such a statement an adequate probability of infection (PoI) should be established.

A. Probability distribution

Since there is no possible data to determine the probability density of the presence of moths caused by an infected tree, a rough estimate based on intuitive assumptions is proposed in this work, which serves as a placeholder once the spread of moths is accurately modelled by entomologists.

Since Pine Processionary Moths tend to travel and move to the closest tree possible with a flying distance of a maximum radius of 15km (the most ever recorded), and on an average of 100m radius during their life cycle, the travel distance is an important factor to consider [49]. Another factor to consider is exposure to the sun. The moth itself always tends to the Southern or South-Western orientation when travelling especially since warm winds blow from the south indicating a warmer weather, which is suitable for the moth's reproduction [49]. Moreover, this Southern or South-Western track is most significant since it leads to the maximum sun exposure for the moths especially in the Mount Lebanon region [2].

After considering the above aspects, it is reasonable to assume that the probability could be estimated using a three-dimensional (3D) probability distribution centered around an infected tree. The distance between the trees is identified as a Euclidean distance especially since Pine Processionary moths travel above tree crowns, so there is no object avoidance required. Moreover, it is important to note that since the travel direction of a moth is at a higher rate towards the south, the probability density function differs with direction.

To solve this issue, a 3D probability distribution is proposed, where it is a combination of a lognormal distribution along the vertical axis (North-South) but tending towards the south, and a normal distribution along the horizontal axis (East-West). It is centered at the infected tree and has a maximum magnitude at the infected tree of *1*.

$$f_{lognorm}(x|\mu,\sigma) = \frac{1}{x\sigma\sqrt{2\pi}} e^{\frac{-(\log x-\mu)^2}{2\sigma^2}}, \tag{87}$$

$$f_{normal}(x|\mu,\sigma) = \frac{1}{\sigma\sqrt{2\pi}} e^{\frac{-(x-\mu)^2}{2\sigma^2}}, \tag{88}$$

$$Prob = f_{lognorm}\left(y\middle|\mu_{lognorm},\sigma_{lognorm}\right) \times f_{normal}(x|\mu_{normal},\sigma_{normal}). \tag{89}$$

Figure 22 Complementary Travel Angles for relative probability determination

This results in an adjacency matrix of probability where the rows represent the "from" and the columns represent the "to". For example, if tree 3 is infected, the probability of tree 4 being infected (based on tree 3) is located in row 3 column 4. These probability matrix cells are used such that every time an infected tree is discovered, the row in the matrix linked to that tree is added to the PoI of all trees.

B. Offline solution

After determining the PoI of each tree, it is important to prioritize those that have a higher probability based on a cut-off margin, which occurs due to the fact that a TSP is a priority-blind problem. Even with changing the adjacency matrix, this would not ensure the creation of attracting and repelling points to prioritize the visit of one tree to another, given that TSP is simply a global optimizer based on cost. Thus, it is important to remove any unnecessary trees that do not surpass the cut-off margin of the PoI. This results in a smaller adjacency matrix that is fed into the TSP solver using ILP along with SEC's in order to create a new energy-optimal tour with the smaller subset of nodes.

C. Online solution

Since most of the work is done in a pre-planned offline manner, an additional planner should be added for online or time-varying approaches. This can be associated with the probability distribution and the cut-off scheme. As the drone roams around to scan the trees, the PoI of each tree changes based on the new scan results. Each tree is assessed as a binary output; if the status of the tree, whether it is "infected" or "not-infected", remains the same, there is no change to the PoI of any tree. If a tree was found to be "infected" in the previous scan, but now it is not, the PoI contributed by that tree to all the trees is removed. If the tree was found to be "not-infected" in the previous scan, but now it is, the row associated with that tree in the adjacency matrix is added to the PoI's of the trees.

As the PoI of each tree changes, the set of trees that need to be visited changes along since trees are being added to (or removed from) the subset. However, a simple ATSP cannot solve this issue, especially since the new set is created when the quadrotor has reached a location that is not the base, where it needs to return to, or even has already passed certain trees. In order to solve this problem, a fixed-start fixed-end TSP (FSFE-TSP) is needed.

The quadrotor remains on the tour and path selected by the offline map until a status change occurs, which is directly followed by a possible new set of points. Since a new tree set is created and the start of the new determined path has become the tree with the status change and the final destination is the base, a FSFE-TSP is used to create a path from that tree to the base passing through all trees excluding the ones already visited with an optimal energy approach. This is mainly solved using a genetic algorithm approach where the strand is used as a chromosome with the first item as the start tree and the last item on the chromosome as the base, as per the following flow chart [21]:

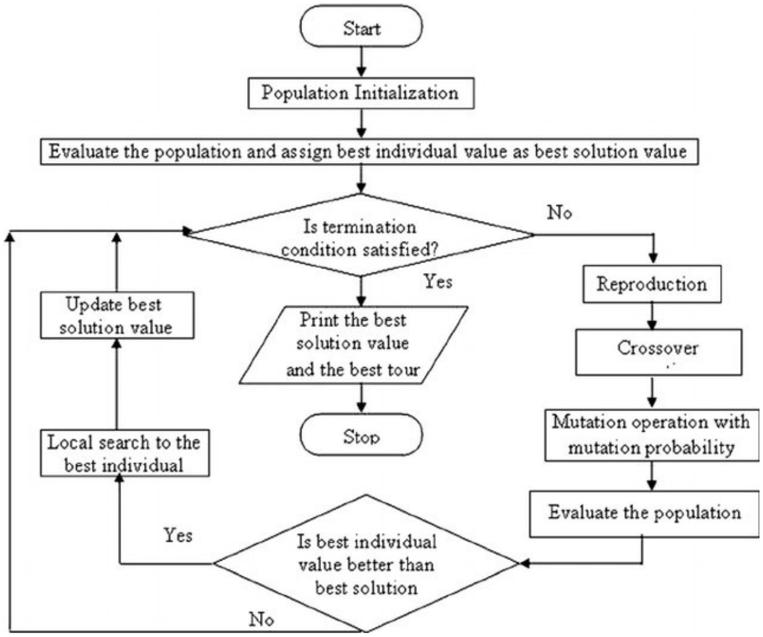

Figure 23 FSFE-TSP Genetic Algorithm Flowchart

The combination of both proposed online and offline solutions is summarized using the following flowchart of Figure 24:

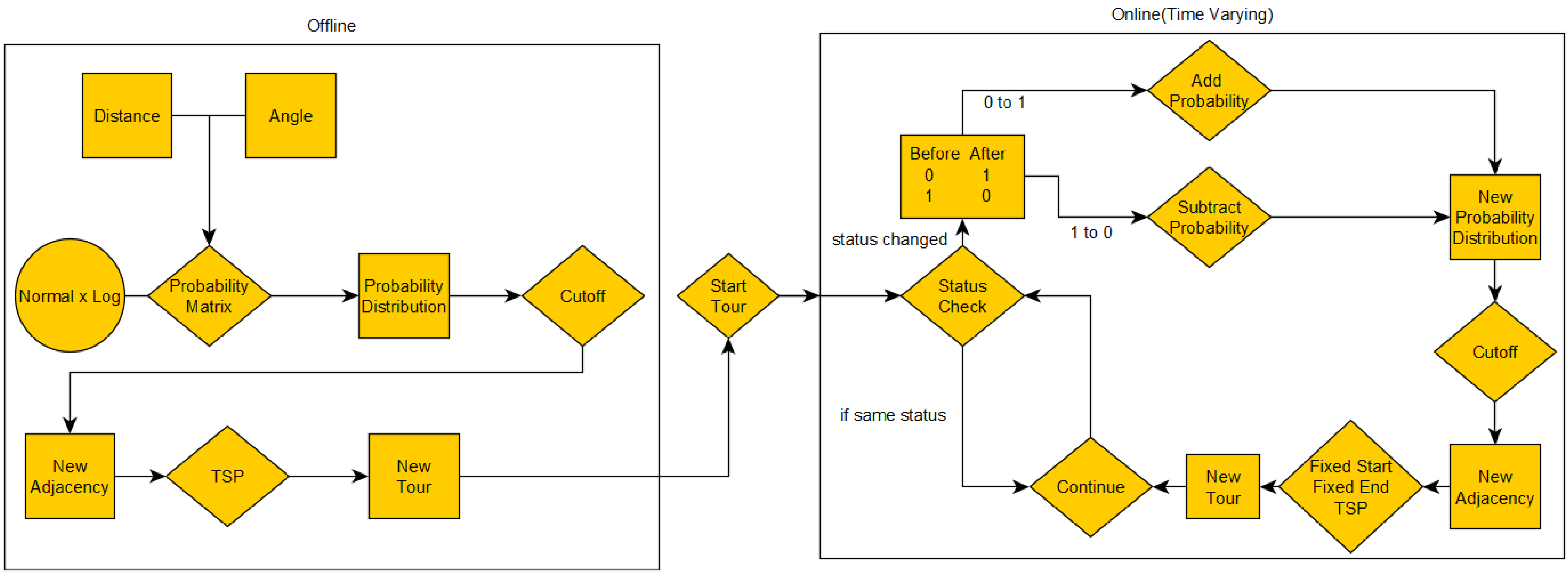

Figure 24 Offline and Online route determination based Probability of Infection

CHAPTER VI

SIMULATION RESULTS

Since the proposed system-level solution is made up of different sections and parts, a separate proof should be presented for each section. This starts with the OCP followed by the full set TSP with ILP and SEC, which is then followed by FSFE-TSP using GA.

A. Optimal Control Solver (ICLOCS vs. ACADO)

To start any solution, it is important to determine the different parameters governing the equations of the OCP. The different parameters are specific to each quadrotor. Parameters of two quadrotors are solved for further insurance of confidence within the program. The parameters listed are for the DJI Phantom2 and the Crazyflie 2.0, respectively.

Table 7 DJI Parameters [11]

K_V=920 rpm/V	K_E=9.5493/K_v Vs/rad	T_f=4×10^{-2} Nm
D_f=2×10^{-4} Nms/rad	R_m=0.2 Ω	m_{quad}=1.3 kg
ω_{max}=1047.197 rad/s	l=0.175 m	J=4.19×10^{-5} kgm^2
k=3.8305×10^{-6}	b_t=2.2518×10^{-8}	K_T=K_E
I_x=0.081 kgm^2	I_y=0.081 kgm^2	I_z=0.142 kgm^2

Table 8 Crazyflie Parameters [50] [51]

K_V=14000 rpm/V	K_E=9.5493/K_v Vs/rad	T_f=1.563383×10^{-5} Nm
D_f=3.5077×10^{-10} Nms/rad	R_m=2 Ω	m_{quad}=30 g
ω_{max}=2513.27 rad/s	l=40 mm	J=1.6833×10^{-7} kgm^2
k=2.4411×10^{-9}	b_t=1.9973×10^{-7}	K_T=K_E
I_x=1.395×10^{-5} kgm^2	I_y=1.395×10^{-5} kgm^2	I_z=2.173×10^{-5} kgm^2

It is important to know that ACADO [52], adopted by Fabio Morbidi in [11], is a solver that uses SQP using a MEX solver for MATLAB, and it requires a set time since it does not accept a free-end-time approach. Moreover, ICLOCS requires an initial and final guess for all of its states including a guess time, which can be considered using the following equation:

$$t_{i,j}^{f} = \sqrt{\left(x_i - x_j\right)^2 + \left(y_i - y_j\right)^2 + \left(z_i - z_j\right)^2}, \tag{90}$$

This presumes that the quadrotor is moving in a straight line at a speed of 1 m/s, which is a reasonable assumption to start with since the result will be close to an almost direct path as assumed. Moreover, it is noted that quadrotor would be hovering at both the start and end points. The hovering rotor speed for each rotor is 912 rad/s and 1,989.675 rad/s for the DJI [11] and Crazyflie, respectively [50]. It is important to note that each quadrotor has a specific battery capacity, which is calculated using the equation of

$$E_{nominal} \times Capacity \times 3.6. \tag{91}$$

This results in an energy capacity of 3.1968 kJ for the Crazyflie2.0 [53] and 207.792 kJ for the DJI Phantom2 [54].

1. *Without object avoidance*

In order to determine which solver is suitable and more capable, it is crucial to compare solvers under the same conditions. The initial position coordinates are (0,0,0) and final coordinates are (4,5,6). For the ACADO solver, the smallest possible time input is determined by trial and error, and free-end-time is not an option when parametrizing time as a variable to be iteratively minimized, which leads to non-convergence. As a result, 5 seconds is the least possible time that does not result in non-convergence. The ACADO solver results in 6.63591 kJ energy consumption, whereas ICLOCS results in 4.2808 kJ with a total time of 2.9776 seconds.

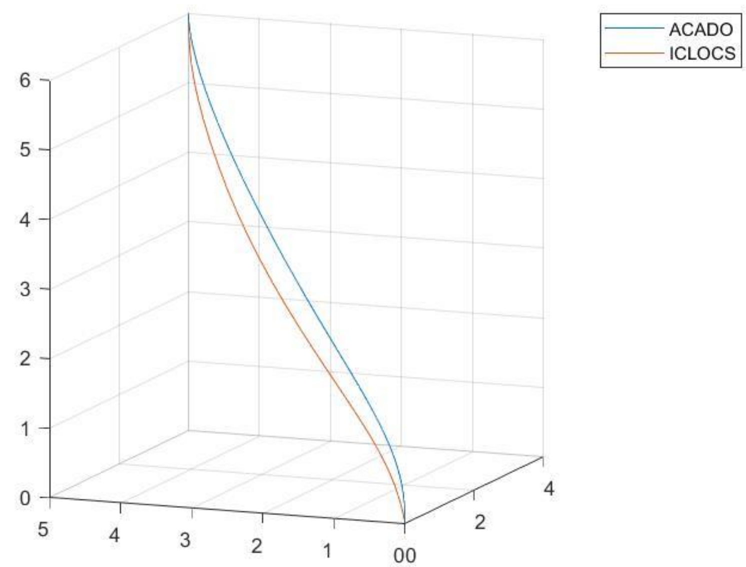

Figure 25 Comparison between ACADO and ICLOCS Trajectory

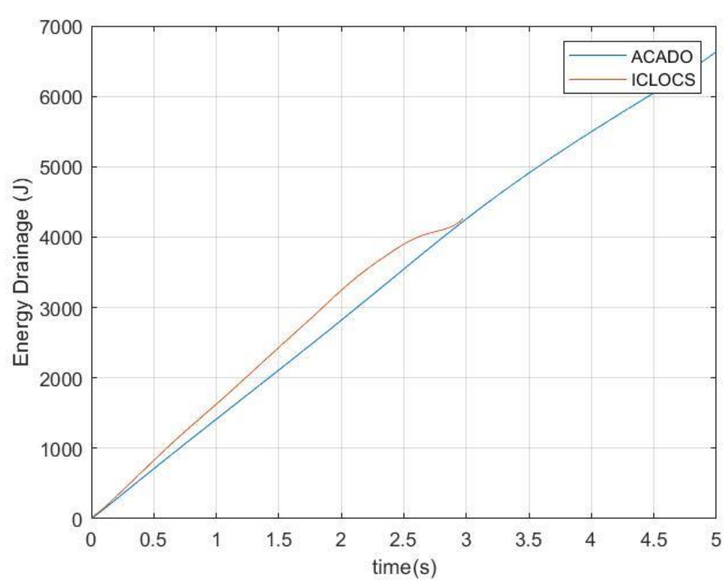

Figure 26 Battery energy drainage versus time

It is noted that both solvers converge at the same endpoint (4,5,6), with ICLOCS resulting in faster battery drainage at its initial stages, whilst ACADO has a constant rate of increase and exceeds the energy consumption of ICLOCS by 55%.

2. *With object avoidance*

One of the most significant factors and advantages of the ICLOCS solver is the ability to converge when object avoidance is present. This is evident even for simple object avoidance, which is placed as a cylinder in the middle of the path. ACADO never converges especially since SQP cannot handle nonlinear path constraints resulting in a failure. On the other hand, ICLOCS converges with extremely low constraint violation. The initial points are (0,0,0) and a final point of (8,8,8) with a Cylinder of center at x=2 and y=2 and a height of 6 with a radius of 1. This results in a final time of 3.5977 seconds and an optimal energy consumption of 5.3301 kJ. It results in the following trajectory:

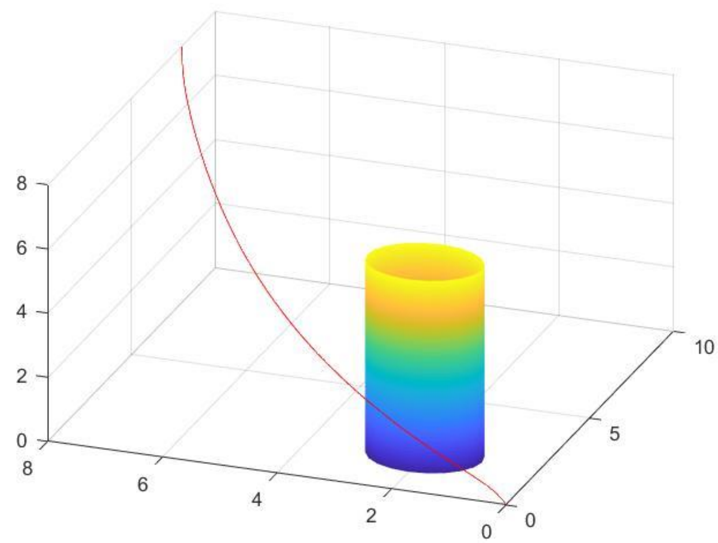

Figure 27 Object Avoidance ICLOCS Trajectory (side view)

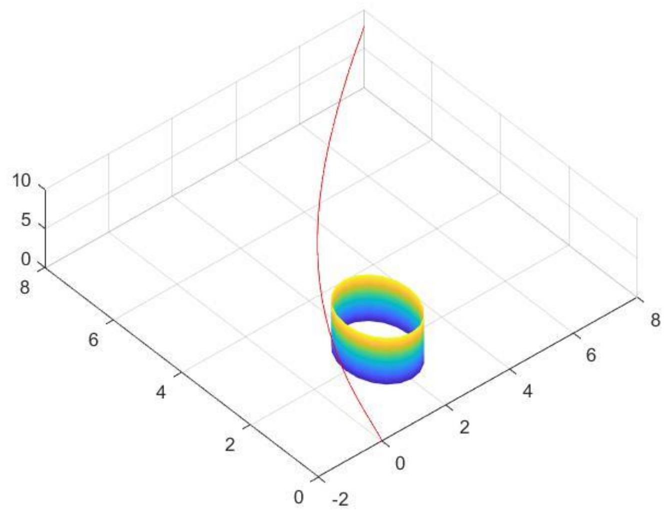

Figure 28 Object Avoidance ICLOCS Trajectory (Top View)

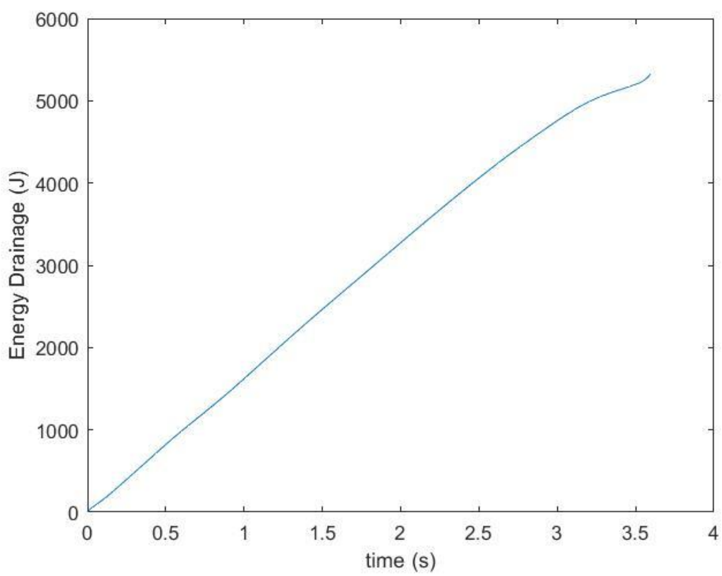

Figure 29 Battery Drainage vs. time (with object avoidance)

B. Travelling salesman solver (ILP/SEC vs. Brute Force)

To test the validity of the ILP solution, it is important to compare it with an adequate solver that has the highest level of confidence. A brute force method compares all possible solution and chooses the lowest one. This requires a large pool of trials, especially with large amounts of permutations (*n-1!)*, from which the option with the least cost is chosen. Since MATLAB only has the capability of generating permutations for up to 9 values before it crashes or freezes, the check for 10 elements is used for validation purposes. This results in 10 nodes but with 9! permutations resulting in 362,880 different results with a computational time of 60 seconds. A randomly generated cost matrix is established along with randomly generated locations for points. It results in the following adjacency matrix:

$$Adj = \begin{pmatrix} 0 & 70.605 & 3.1833 & 27.692 & 4.6171 & 9.7132 & 82.346 & 69.483 & 31.71 & 95.022 \\ 3.446 & 0 & 43.874 & 38.156 & 76.552 & 79.52 & 18.687 & 48.976 & 44.559 & 64.631 \\ 70.936 & 75.469 & 0 & 27.603 & 67.97 & 65.51 & 16.261 & 11.9 & 49.836 & 95.974 \\ 34.039 & 58.527 & 22.381 & 0 & 75.127 & 25.51 & 50.596 & 69.908 & 89.09 & 95.929 \\ 54.722 & 13.862 & 14.929 & 25.751 & 0 & 84.072 & 25.428 & 81.428 & 24.652 & 92.926 \\ 34.998 & 19.66 & 25.108 & 61.604 & 47.329 & 0 & 35.166 & 83.083 & 58.526 & 54.972 \\ 91.719 & 28.584 & 75.72 & 75.373 & 38.045 & 56.782 & 0 & 7.5854 & 5.395 & 53.08 \\ 77.917 & 93.401 & 12.991 & 56.882 & 46.939 & 1.1902 & 33.712 & 0 & 16.218 & 79.428 \\ 31.122 & 52.853 & 16.565 & 60.198 & 26.297 & 65.408 & 68.921 & 74.815 & 0 & 45.054 \\ 8.3821 & 22.898 & 91.334 & 15.238 & 82.582 & 53.834 & 99.613 & 7.8176 & 44.268 & 0 \end{pmatrix}$$

Both Brute Force and the ILP/SEC solver yielded the following result as a sequence of trees to be visited in order: *1→5→4→3→7→9→10→8→6→2→1* with a total tour cost of 151.5715. The point locations are generated randomly and are as follows:

Table 9 TSP trial coordinates

Node Number	X	Y	Z
1	81.472	90.579	12.699
2	91.338	63.236	9.754
3	27.85	54.688	95.751
4	96.489	15.761	97.059
5	95.717	48.538	80.028
6	14.189	42.176	91.574
7	79.221	95.949	65.574
8	3.5712	84.913	93.399
9	67.874	75.774	74.313
10	39.223	65.548	17.119

Both solutions resulted in the following graph:

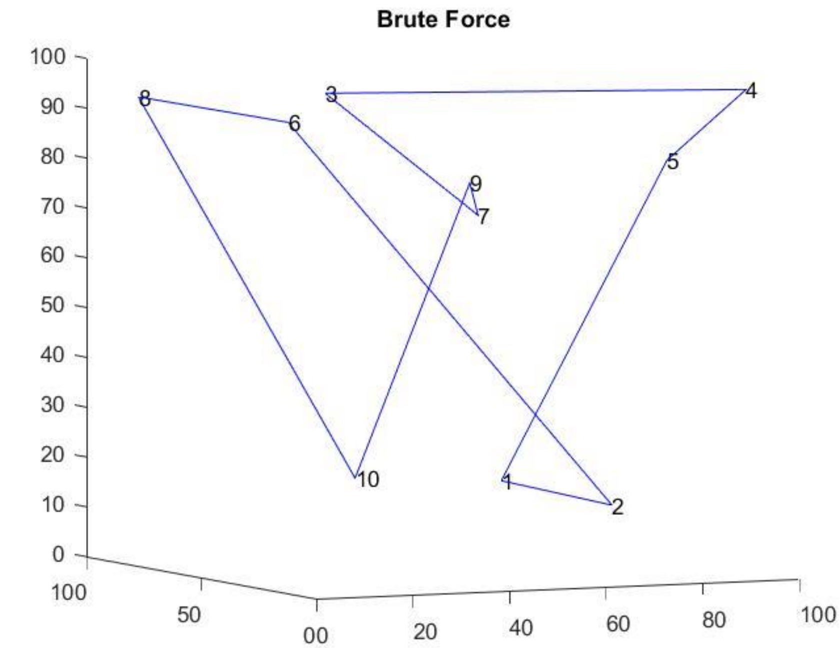

Figure 30 Brute Force Solution

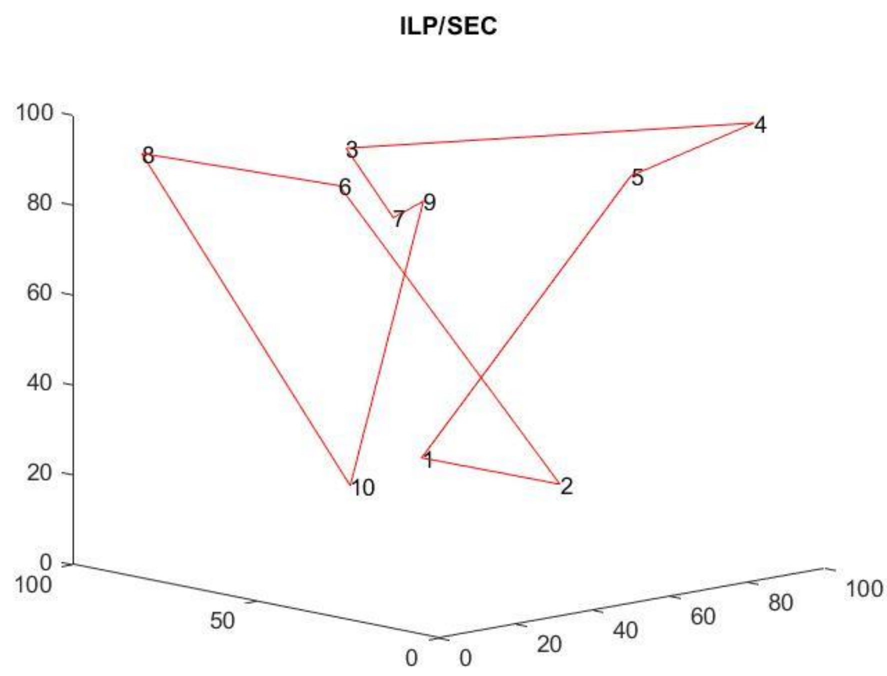

Figure 31 ILP/SEC TSP Solution

C. Fixed-Start Fixed-End TSP (GA vs. Brute Force)

Similar to the validation for the ILP/SEC, it is important to validate the solution using Genetic algorithms. This suggested solution is also compared to the brute force method. GA reduces calculation time since it does not consider the entire pool of options as brute force does, but rather uses different methods of evolution in order to find the optimal result. As mentioned before, the maximum number of permutations allowed by MATLAB is 9. As such, a randomly generated cost matrix is made with randomly generated point locations. It is important to note that the GA solution starts with 1 and ends with 10, whilst the optimization is done with the order of the nodes that go in between the initial and final node. This results in *n-2!* solutions, which equals to 8! or 40,320.

$$Adj = \begin{pmatrix} 0 & 10.676 & 65.376 & 49.417 & 77.905 & 71.504 & 90.372 & 89.092 & 33.416 & 69.875 \\ 19.781 & 0 & 3.0541 & 74.407 & 50.002 & 47.992 & 90.472 & 60.987 & 61.767 & 85.944 \\ 80.549 & 57.672 & 0 & 18.292 & 23.993 & 88.651 & 2.8674 & 48.99 & 16.793 & 97.868 \\ 71.269 & 50.047 & 47.109 & 0 & 5.9619 & 68.197 & 4.2431 & 7.1445 & 52.165 & 9.673 \\ 81.815 & 81.755 & 72.244 & 14.987 & 0 & 65.961 & 51.859 & 97.297 & 64.899 & 80.033 \\ 45.38 & 43.239 & 82.531 & 8.347 & 13.317 & 0 & 17.339 & 39.094 & 83.138 & 80.336 \\ 6.0471 & 39.926 & 52.688 & 41.68 & 65.686 & 62.797 & 0 & 29.198 & 43.165 & 1.5487 \\ 98.406 & 16.717 & 10.622 & 37.241 & 19.812 & 48.969 & 33.949 & 0 & 95.163 & 92.033 \\ 5.2677 & 73.786 & 26.912 & 42.284 & 54.787 & 94.274 & 41.774 & 98.305 & 0 & 30.145 \\ 70.11 & 66.634 & 53.913 & 69.811 & 66.653 & 17.812 & 12.801 & 99.908 & 17.112 & 0 \end{pmatrix}$$

Both Brute Force and GA solvers gave the following result:

1→2→6→5→4→8→3→9→7→10, which sums up to a total cost of 164.8541. The locations of the points are randomly selected, and they were generated as follows:

Table 10 FSFE TSP solver trial Coordinates

Node Number	X	Y	Z
1	8.5516	26.248	80.101
2	2.922	92.885	73.033
3	48.861	57.853	23.728
4	45.885	96.309	54.681
5	52.114	23.159	48.89
6	62.406	67.914	39.552
7	36.744	98.798	3.7739
8	88.517	91.329	79.618
9	9.8712	26.187	33.536
10	67.973	13.655	72.123

Both solutions result in the same graph shown as follows:

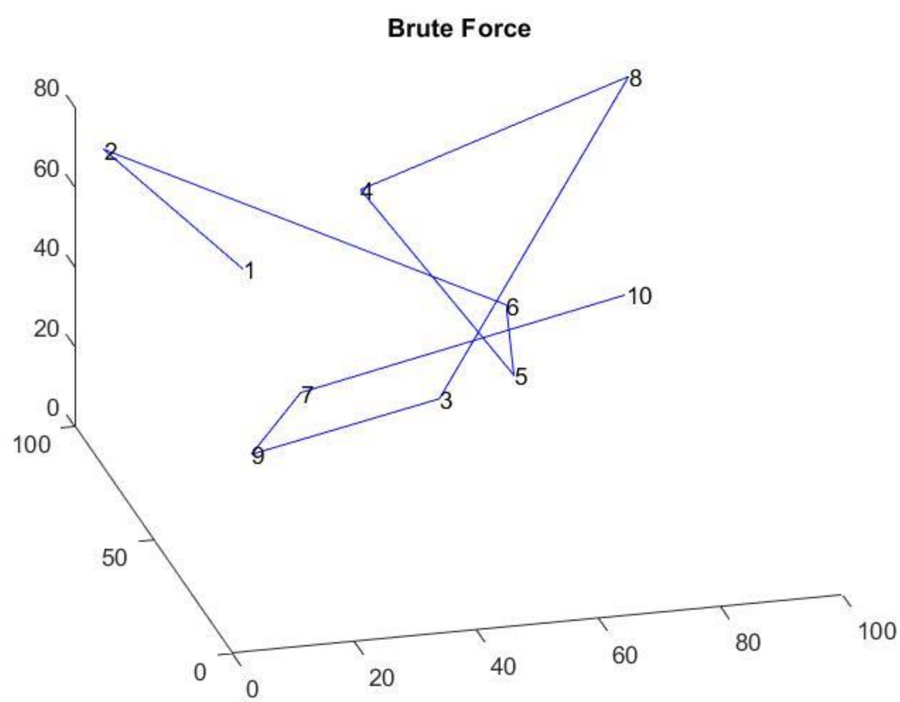

Figure 32 Brute Force Solver Sequence

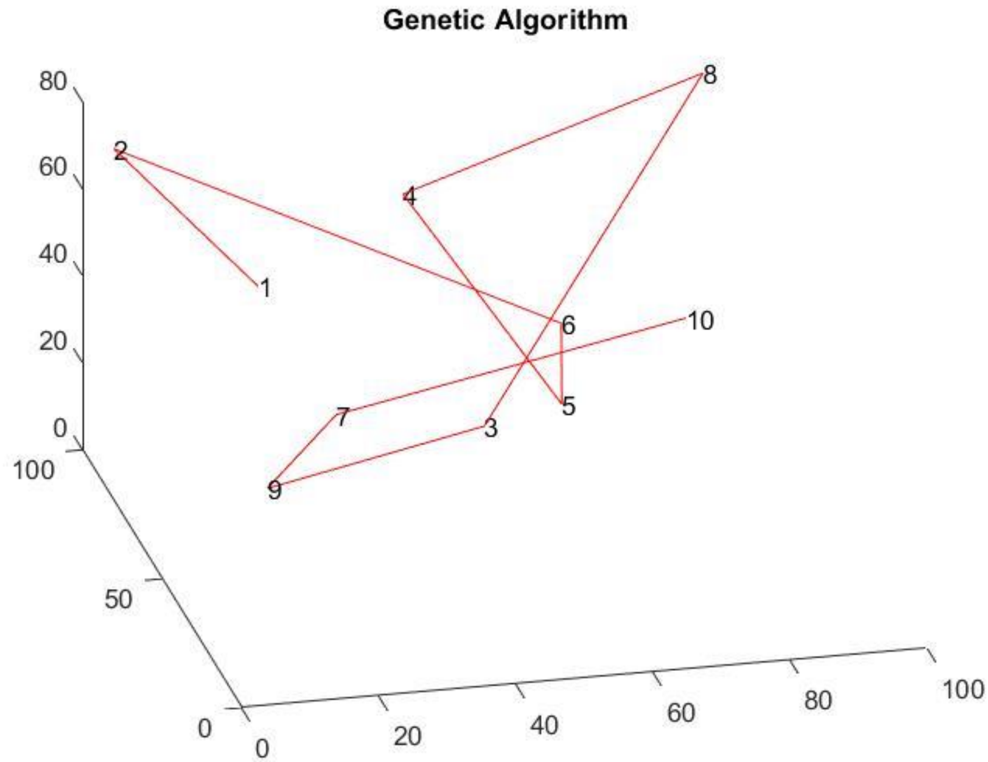

Figure 33 Genetic Algorithm Solver Sequence

D. Full set energy-optimal tour

In order to combine the different aspects of the developed system, a small garden of cylindrically-shaped trees along with their downwash inflation factor is considered. The first trial is using the DJI parameters to navigate a 5-tree forest with the following set of radii and locations:

Table 11 DJI Energy-Optimal Tour Coordinates and Tree Locations

Site	X_{tree}(m)	Y_{tree}(m)	r_{tree}(m)	x(m)	y(m)	z(m)
Base	0	0	0	0	0	0
Tree 1	10	10	0.2	9.4	10	3
Tree 2	8	4	0.35	7.25	4	3
Tree 3	2	8	0.4	1.1	8	5
Tree 4	9	1	0.15	8.45	1	4
Tree 5	2	2	0.1	1.5	2	2

The $\{X_{tree}, Y_{tree}, r_{tree}\}$ are identifying the tree location and radius while $\{x,y,z\}$ is the location of the scanning point. The *x* value is determined by adding an inflation radius of 30cm to the tree radius and add another 10 cm for spacing. This inflation is taken as a sum of the arm length and propeller radius. This is solved using the -p/hp-solver, which uses an LGR transcription method. The highest order for *p* is 10 degrees and the h-sectioning is 5. This results in a full set of optimal control solutions, with a small issue related to constraint violation where the maximum violation reached is 8cm within the extra 30cm inflation rate, which might cause problems with the downwash effect. This might come due to the optimization using IPOPT or transcription especially when it comes to high nonlinear and coupled formulations such as the one at hand. The following adjacency matrix is produced:

$$Adj = \begin{pmatrix} 0 & 5.17 & 4.3 & 4.37 & 4.38 & 2.88 \\ 5.24 & 0 & 3.84 & 4.45 & 4.36 & 6.26 \\ 4.35 & 3.84 & 0 & 4.09 & 3.06 & 3.95 \\ 4.49 & 4.59 & 4.21 & 0 & 4.58 & 3.93 \\ 4.46 & 4.36 & 3.11 & 4.56 & 0 & 4.29 \\ 3.01 & 4.75 & 3.89 & 3.86 & 4.14 & 0 \end{pmatrix} \times 10^3\, J$$

As seen, the cost results in an asymmetric adjacency matrix, which proves that this problem is an ATSP given the variable height that the quadrotors must attain to scan the required areas in the *z*-direction. Using ILP/SEC results in the following sequence: 1→6→4→2→3→5→1 with a total of 22.68 KJ and a total tour time for 15.8s. The energy-optimal tour and path are depicted in Figure 34. This result comes exactly like [55].

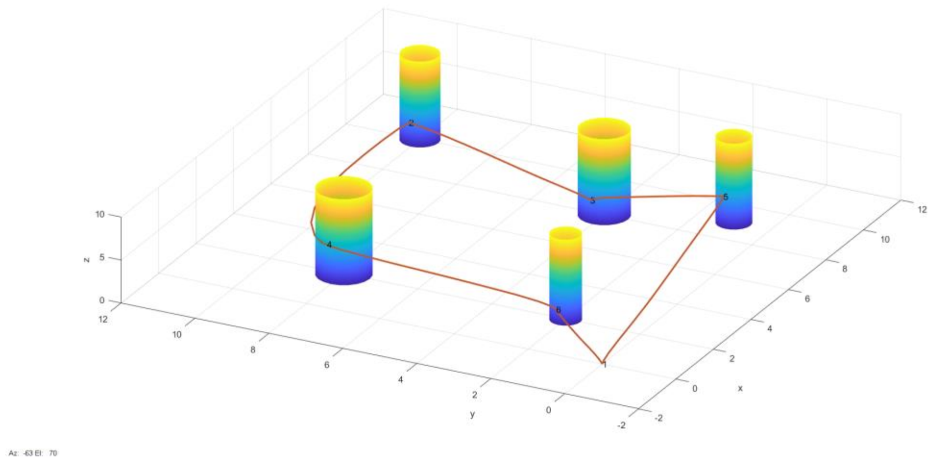

Figure 34 DJI Energy-Optimal Path and Tour

As this has proven to be efficient, it is then important to extrapolate it to a larger number of trees and even to other types of drones. As such the Crazyflie2.0 quadrotor is used which means using the parameters of Table 8. A forest of 10 trees is used, with the following components:

Table 12 Crazyflie2.0 path Coordinates and Tree location and dimensions

Site	X_{tree}(m)	Y_{tree}(m)	r_{tree}(m)	x(m)	y(m)	z(m)
Base	0	0	0	0	0	0
Tree 1	10	10	0.2	9.4	10	3
Tree 2	8	4	0.35	7.25	4	3
Tree 3	2	8	0.4	1.1	8	5
Tree 4	9	1	0.15	8.45	1	4
Tree 5	2	2	0.1	1.5	2	2
Tree 6	5	12	0.1	4.5	12	3
Tree 7	15	3	0.2	14.4	3	4
Tree 8	4	6	0.15	3.45	6	2
Tree 9	6	15	0.1	5.5	15	3
Tree 10	15	15	0.4	14.2	15	4

This is solved using an auto_direct solver, which results in the following adjacency matrix:

$$Adj = \begin{pmatrix} 0 & 129.77 & 110.87 & 115.81 & 116.78 & 90.863 & 126.99 & 132.97 & 106.89 & 135.93 & 148.65 \\ 130 & 0 & 105.09 & 115.94 & 114.53 & 121.27 & 100.64 & 112.11 & 107.67 & 103.59 & 106.11 \\ 111.18 & 104.98 & 0 & 115.03 & 100.68 & 104.54 & 112.51 & 113.69 & 96.606 & 121.33 & 127.5 \\ 115.84 & 115.97 & 115.32 & 0 & 117.36 & 105.54 & 100.65 & 132.84 & 96.609 & 111.65 & 133.21 \\ 117.06 & 114.5 & 100.8 & 144.69 & 0 & 111.33 & 123.04 & 105.77 & 109.16 & 131.5 & 132.81 \\ 91.037 & 121.04 & 104.66 & 105.56 & 111.45 & 0 & 118.5 & 130.3 & 95.42 & 128.38 & 141.39 \\ 127.09 & 107.22 & 112.4 & 100.9 & 137.33 & 126.91 & 0 & 127.76 & 102.98 & 89.407 & 119.53 \\ 133.24 & 112.32 & 0 & 132.59 & 105.61 & 131.25 & 0 & 0 & 124.68 & 132.35 & 123.48 \\ 107.7 & 107.58 & 96.541 & 96.938 & 108.93 & 95.611 & 102.91 & 125.09 & 0 & 114.54 & 130.17 \\ 0 & 103.34 & 121.28 & 111.83 & 131.26 & 128.46 & 89.553 & 132.18 & 114.71 & 0 & 116.83 \\ 0 & 106.36 & 127.66 & 133.17 & 132.92 & 141.55 & 119.28 & 123.47 & 130.26 & 116.58 & 0 \end{pmatrix} J$$

It is important to note that this used 40 sectioning segments for auto transcription. This resulted in almost negligible constraint violations as shown in the following table:

Table 13 Crazyflie Constraint violation (m^2)

From\To	1	2	3	4	5	6	7	8	9	10	11
1	0	0.010458	0	0	0	0	0	0	0	0	0.018536
2	0.010507	0	0	0.0016915	0	0.00010954	0.0007774	0.00012816	6.5378e-05	0.000602	0.00018043
3	0	0	0	0.0040339	0	0.00092536	0	0.00046669	3.8345e-05	0	0
4	0	0.0012261	0.0038834	0	7.3451e-05	0	2.1427e-05	0.013013	7.0717e-05	0	0.0030541
5	0	0	0	0.0076085	0	0.00089086	0.0074373	0.00058828	0.00030661	1.4678e-05	0
6	0	0.0013487	0.00035905	0	0.00016822	0	0	0.0011533	0	0	0.0007107
7	0	0.00013747	0	4.8189e-05	0.010329	0.0077368	0	8.3431e-05	0	0	1.6367e-05
8	0	0.00015035	0	0.013146	0.00038457	0.0016839	0	0	0.010683	0	0
9	0	0.00037675	0.00015829	0.00047399	0.00029254	0	0	0.010688	0	0.00020532	0.001487
10	0	0.00064243	0	0	0.00093293	0	0	0	0	0	0.00063823
11	0	0.00011688	0	0.0039565	0	0.0029589	0.00091278	0	0.0019674	0.001146	0

This result also includes non-diagonal zero elements. These represent the non-converging elements that are directly eliminated from the set. There are no solvers with 100% convergence rate, especially with the given series of equations with high nonlinearities and couplings. Non-convergence and constraint violations occur even when adding a tolerance for the altitude (z-direction) location with a ±5cm and -10cm in the x-

direction. This matrix is placed into the ILP/SEC solver and reaches the following optimal tour: 1→6→9→4→7→10→2→11→8→5→3→1 with a total energy consumption of 1.124 kJ and a total travel time of 17.17 seconds. As such, this results in the following figure path:

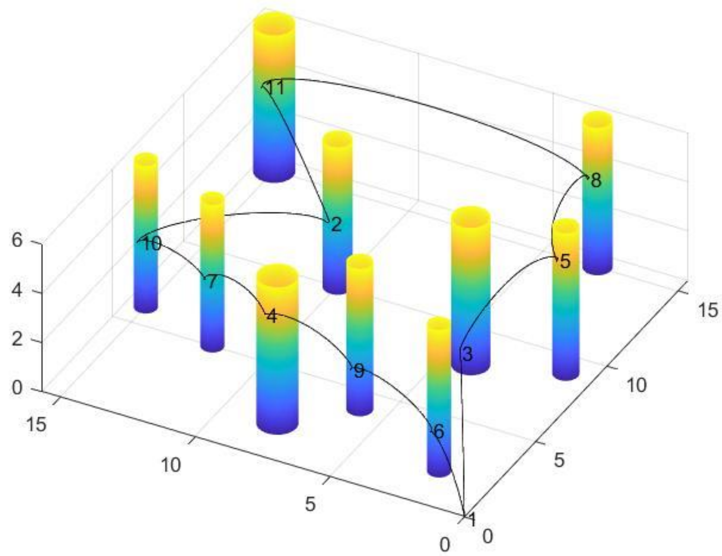

Figure 35 Crazyflie Energy-Optimal Path and Tour

As seen, it is important to point out that the obtained result looks like a 'rubber band' solution where the optimal tour takes the quadrotor around the peripheries. Moreover, it is important to note that non-convergence occurs when avoidance of numerous obstacles is involved, which requires further segmentation, but this is not a pressing issue since the actual solution is almost never the internal option, i.e. visiting the trees from the inside of the contour. That said, it is preferable to have the entire matrix populated in order to ensure that all possibilities are available, and that the obtained tour is the absolute optimal (global) without any doubt. Thus, the hybrid option is used to enable constraint violation criteria to be implemented since comparison can be done between the three different solver options.

E. Closest-Neighbor Approach

In order to determine the efficiency of energy optimality assessment, a closest-neighbor approach, which is generally determined by the Euclidean distance between each pair of locations, is considered and results in the following table:

Table 14 Euclidean Distance (m)

From\To	1	2	3	4	5	6	7	8	9	10	11
1	0	14.048	8.3404	9.4979	9.8693	3.2016	13.162	15.243	7.2043	16.256	21.039
2	14.048	0	6.68	8.7687	9.2684	11.288	5.2924	8.6603	7.239	6.3411	7.0029
3	8.3404	6.68	0	8.3 56	5.142	6.1695	8.6927	7.8181	4.4091	11.316	13.353
4	9.4979	8.7687	8.356	0	10.15	6.7201	5.6178	14.244	4.3038	8.5065	14.887
5	9.8693	9.2684	5.142	10.15	0	7.6356	11.858	6.3563	7.6811	14.447	15.168
6	3.2016	11.288	6.1695	6.7201	7.6356	0	10.488	13.092	4.45	13.638	18.284
7	13.162	5.2924	8.6927	5.6178	11.858	10.488	0	13.417	6.1727	3.1623	10.202
8	15.243	8.6603	7.8181	14.244	6.3563	13.092	13.417	0	11.528	14.974	12.002
9	7.2043	7.239	4.4091	4.3038	7.6811	4.45	6.1727	11.528	0	9.2845	14.162
10	16.256	6.3411	11.316	8.5065	14.447	13.638	3.1623	14.974	9.2845	0	8.7573
11	21.039	7.0029	13.353	14.887	15.168	18.284	10.202	12.002	14.162	8.7573	0

This results in a sequence of 1→6→9→4→7→10→2→5→3→8→11→1. This is used along with the minimum energy approach using the Crazyflie 2.0 results in a total energy consumption of 1.1779 kJ and a total time of 18.896 seconds. It is important to note the increase in total time and energy consumption compared to the tour in the previous section which includes a 50 J increase and a 1 second increase compared to the auto-direct approach. The full path is shown in the following Figure 36 Close Neighbors Energy tour:

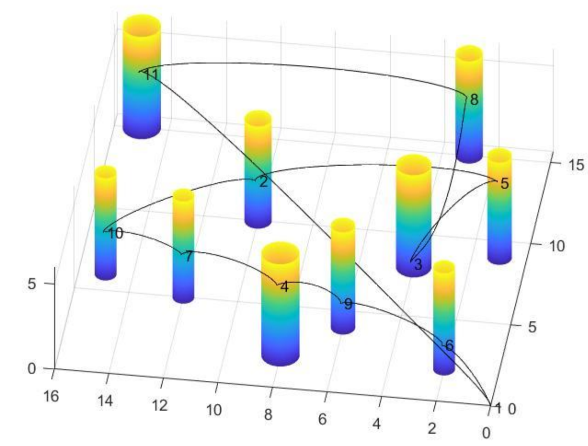

Figure 36 Close Neighbors Energy tour

F. Hybrid solution for Full energy optimal tour

All three solvers are used, and the following hybrid system is used as shown in Figure 21 along with the same map as shown in the Crazyflie solution. This is done for both Crazyflie and DJI. All three options are placed with 40 segments and the order of the Legendre polynomial is set to 10. It is important to note that since the DJI is a relatively heavy quadrotor that is harder to maneuver due to its slower dynamics, it has the advantage of being more stable (due to its larger damping) than the Crazyflie 2.0. This has a drawback that constraint violations are higher when simulating large numbers of path constraints, which require constant hard dynamic maneuvers, as such using all three solvers result in the following results, as shown in Table 15, Table 16, and Table 17.

Table 15 DJI Automatic Direct Collocation Constraint Violation (m^2)

From\To	1	2	3	4	5	6	7	8	9	10	11
1	0	0.0020183	0	0	0	0	0.015462	0	0	0.040486	0.026641
2	0.018529	0	0	0.0051622	0	3.3049e-06	0.0016877	0	0.0017549	0.15921	0.062717
3	0	0	0	0.010983	0	0.0020541	0	0.0010142	0.001371	0	0.0018483
4	0	0	0.0088811	0	9.624e-05	0	0.019753	0.018838	0	0	0.0016067
5	0	0.0027744	0	0.10474	0	0.0024747	0.0067891	0.00087219	0.0011782	0.017364	0
6	0	0.00087667	0.00036786	0	0.00076495	0	0	0.0014358	0	0	3.1468e-05
7	0	0.00038208	0	0.00079396	0	0	0	0.0015182	0	0	0.0011992
8	0	0.00023876	0.0035551	0.028373	0.0021773	0.0041642	0	0	0.18137	0.0014925	0
9	0	0.10833	0.00061565	7.0996e-05	0.00023713	0	0	0.010032	0	0	0.00073658
10	0	0	0	0	0.014673	0.011698	0	0.0010762	0.0014592	0	0.0010713
11	0	0.00047053	0.0012832	0.0046455	0	0.0025344	0.0031948	0	1.0501e-05	0.0040727	0

Table 16 DJI Hermite-Simpson Collocation Constraint Violation (m^2)

From\To	1	2	3	4	5	6	7	8	9	10	11
1	0	0.0020183	0	0	0	0	0.015462	0	0	0.040486	0.026641
2	0.018529	0	0	0.0051622	0	3.305e-06	0.0016877	0	0.0017548	0.15921	0.062717
3	0	0	0	0.010983	0	0.0020541	0	0.0010142	0.001371	0	0.0018479
4	0	0	0.0088811	0	9.6239e-05	0	0	0.018838	0	0	0.0016067
5	0	0.0027744	0	0.10474	0	0.0024747	0.0067892	0.0008722	0.0011782	0.017365	0
6	0	0.00087666	0.00036786	0	0.00076495	0	0	0.0014358	0	0	3.147e-05
7	0	0.00038208	0	0.00079393	0	0	0	0.0015182	0	0	0.0011992
8	0	0.00023879	0.0035551	0.028373	0.0021773	0.0041642	0	0	0.0029908	0.0014925	0
9	0	0.10833	0.00061565	7.1007e-05	0.00023714	0	0	0.010032	0	0	0.00073658
10	0	0	0	0	0.014673	0.011698	0	0.0010762	0.0014592	0	0.0010713
11	0.026933	0.00047053	0.41206	0.0046457	0	0.0025344	0.0031948	0	1.048e-05	0.0040727	0

Table 17 DJI Legendre-Gauss-Radau Collocation Constraint Violation (m^2)

From\To	1	2	3	4	5	6	7	8	9	10	11
1	0	0.081662	0	0	0	0	0.074142	0	0	0.10526	0.089235
2	0.058452	0	0	0.02954	0.010073	6.2304e-05	0.0049639	0.00023889	1.0603e-05	0	0.00090976
3	0	0	0	0.038744	0	0.00691	0	0.030061	0.0078572	0	0.0070673
4	0	0.025421	0.044654	0	0.00053025	0	0	0.093529	0	0	0.00016233
5	0	0.052129	0	0.0040673	0	0.0080024	0	0.012437	0.0014113	0.010597	0
6	0	0.010868	0.0063776	0	0.013007	0	0	0.15764	0	0	0.018666
7	0.022699	0.0081995	0	0	0	0	0	0.0066104	0	0	0.0042268
8	0	0.0052876	0.021025	0.12926	0.0073691	0.011472	0.0090577	0	0.0072079	0.093566	0
9	0	0.00050828	0.00029359	0.0039119	0.00091062	0	0	0.0057769	0	0	0.016905
10	0.002826	0	0	0	0.077973	0	0	0.0045784	0	0	0.0089237
11	0.045938	0.00043196	0.0018372	0.016025	0	0.0014353	0.0067737	0	0.00012368	0.015875	0

This also resulted in the following adjacency matrices for energy, where the 0's identify the absence of the node due to either non-convergence or due to high constraint violations, which has a cutoff margin of 0.1 m that can be calculated by performing $\sqrt{violation}$ since they are in terms of m^2 where as the cut-off criteria is in m. The following are the list of energy matrices:

Table 18 Auto-direct Collocation Energy Matrix (J)

From\To	1	2	3	4	5	6	7	8	9	10	11
1	0	5179	4180	4427.1	4426.3	0	0	5333.9	0	0	0
2	0	0	3913.3	4509.3	4440.3	4790.5	3714.4	4267	4074.9	0	0
3	4266.3	3886.2	0	0	3345.8	3914.5	4330.9	4349.9	3461	4800.1	5077.8
4	4434.5	0	4508.9	0	4538.6	0	0	0	0	4233.8	5381
5	12952	4432.6	3375.7	0	0	4231.9	4892.8	3931	4095.6	0	5373.3
6	0	4735.3	3865.3	0	4204	0	4644.8	5343.7	0	5154.4	5783
7	5091.8	3643.8	0	3635.3	4865.3	4666.2	0	5087.4	3830.6	3049.7	4682.7
8	5399.7	4319.2	4351.2	0	3971.5	5231.6	0	0	0	5363.5	4919.7
9	4015	0	3373.5	3213.2	4024.9	3418.3	3818.1	0	0	4436.4	5214.7
10	5548.8	3788	0	0	0	0	3074.9	5321.1	4464.8	0	4568.4
11	28257	3987.6	5116.3	5407	5399.4	5822.8	4693.8	4919.3	5253.2	4552.8	0

Table 19 Legendre-Gauss-Radau Collocation Energy Matrix (J)

From\To	1	2	3	4	5	6	7	8	9	10	11
1	0	0	4173.9	4385.1	4384.2	2791.5	0	5312.6	3951.6	0	0
2	0	0	3866	0	0	4725.2	3591	4261.7	3975.6	3785.7	3905.4
3	4173.9	3864.1	0	0	3276.9	3791.6	4287.8	0	3315	4775.2	5060.7
4	4380.2	0	0	0	4534.5	3862.9	3543	0	0	4216.8	5332.2
5	4378.7	0	3278.7	4534.6	0	4101.9	4849.9	0	3990.9	0	5366.6
6	2792.3	0	3794.5	3863.6	0	0	4638.7	0	3376.7	5147.8	0
7	0	3592.6	4289.6	3539.2	4853.1	4638.5	0	5072.1	3811.6	3048.3	4601.9
8	5304.7	4261.7	0	0	3849.5	0	5071.5	0	4872.7	0	4914.4
9	3953	3975.9	3322.3	3019.8	3988.6	3376.7	3811.2	4877.5	0	4430	0
10	5507.6	3785.7	4771.6	4217.1	0	5147.3	0	5314.7	4430	0	4453.1
11	0	3905.1	5054.7	0	5367	5765.1	4602.2	4914.4	5179.9	0	0

Table 20 Hermite-Simpson Collocation Energy Matrix (J)

From\To	1	2	3	4	5	6	7	8	9	10	11
1	0	5179	4180	4427.1	4426.3	0	0	5333.9	0	0	0
2	0	0	3913.3	4509.3	4440.3	4790.5	3714.4	4267	4074.9	0	0
3	4266.3	3886.2	0	0	3345.8	3914.5	4330.9	4349.9	3461	4800.1	5077.8
4	4434.5	0	4508.9	0	4538.6	0	0	0	0	4233.8	5381
5	13119	4432.6	3375.7	0	0	4231.9	4892.8	3931	4095.6	0	5373.3
6	0	4735.3	3865.3	0	4204	0	4644.8	5343.7	0	5154.4	5783
7	5091.8	3643.8	0	3635.3	4865.3	4666.2	0	5087.4	3830.6	3049.7	4682.7
8	5399.7	4319.2	4351.2	0	3971.5	5231.6	0	0	4964.6	5363.5	4919.7
9	4015	0	3373.5	3213.2	4024.9	3418.3	3818.1	0	0	4436.4	5214.7
10	5548.8	3788	0	0	0	0	3074.9	5321.1	4464.8	0	4568.4
11	0	3987.6	0	5407	5399.4	5822.8	4693.8	4919.3	5253.2	4552.8	0

After removing all the items that have not converged, or which have not respected the constraint violation cut-off margin, the selection for minimum energy is made between all three. This results in the following adjacency matrix:

$$Adj_{optimal} = \begin{pmatrix} 0 & 5179 & 4173.9 & 4385.1 & 4384.2 & 2791.5 & 0 & 5312.6 & 3951.6 & 0 & 0 \\ 0 & 0 & 3866 & 4509.3 & 4440.3 & 4725.2 & 3591 & 4261.7 & 3975.6 & 3785.7 & 3905.4 \\ 4173.9 & 3864.1 & 0 & 0 & 3276.9 & 3791.6 & 4287.8 & 0 & 3315 & 4775.2 & 5060.7 \\ 4380.2 & 0 & 4508.9 & 0 & 4534.5 & 3862.9 & 3543 & 0 & 0 & 4216.8 & 5332.2 \\ 4378.7 & 4432.6 & 3278.7 & 4534.6 & 0 & 4101.9 & 4849.9 & 3931 & 3990.9 & 0 & 5366.6 \\ 2792.3 & 4735.3 & 3764.5 & 3863.6 & 4204 & 0 & 4638.7 & 5343.7 & 3376.7 & 5147.8 & 5783 \\ 0 & 3592.6 & 4289.6 & 3539.2 & 4853.1 & 4638.5 & 0 & 5072.1 & 3811.6 & 3048.3 & 4601.9 \\ 5304.7 & 4261.7 & 4351.2 & 0 & 3849.5 & 5231.6 & 5071.5 & 0 & 4872.7 & 5363.5 & 4914.4 \\ 3953 & 3975.9 & 3322.3 & 3019.8 & 3988.6 & 3376.7 & 3811.2 & 4877.5 & 0 & 4430 & 0 \\ 5507.6 & 3785.7 & 4771.6 & 4217.1 & 0 & 5147.3 & 3074.9 & 5314.7 & 4430 & 0 & 4453.1 \\ 28257 & 3905.1 & 5054.7 & 5407 & 5367 & 5765.1 & 4602.2 & 4914.4 & 5179.9 & 0 & 0 \end{pmatrix}$$

This results in the following constraint violation matrix:

Table 21 Hybrid Constraint Violation matrix (m²)

From\To	1	2	3	4	5	6	7	8	9	10	11
1	0	0.0020183	0	0	0	0	0	0	0	0	0
2	0	0	0	0.0051622	0	6.2304e-05	0.0049639	0.00023889	1.0603e-05	0	0.00090976
3	0	0	0	0	0	0.00691	0	0	0.0078572	0	0.0070673
4	0	0	0.0088811	0	0.00053025	0	0	0	0	0	0.00016233
5	0	0.0027744	0	0.0040673	0	0.0080024	0	0.00087219	0.0014113	0	0
6	0	0.00087667	0.0063776	0	0.00076495	0	0	0.0014358	0	0	3.1468e-05
7	0	0.0081995	0	0	0	0	0	0.0066104	0	0	0.0042268
8	0	0.0052876	0.0035551	0	0.0073691	0.0041642	0.0090577	0	0.0072079	0.0014925	0
9	0	0.00050828	0.00029359	0.0039119	0.00091062	0	0	0.0057769	0	0	0
10	0.002826	0	0	0	0	0	0	0.0045784	0	0	0.0089237
11	0	0.00043196	0.0018372	0.0046455	0	0.0014353	0.0067737	0	0.00012368	0	0

The results are taken from each solver as follows:

Table 22 Solvers in Hybrid Solution for DJI

From\To	1	2	3	4	5	6	7	8	9	10	11
1	0	auto	hp	hp	hp	hp	0	hp	hp	0	0
2	0	0	hp	auto	auto	hp	hp	hp	hp	hp	hp
3	hp	hp	0	0	hp	hp	hp	hp	hp	hp	hp
4	hp	0	auto	0	hp	hp	hp	0	0	hp	hp
5	hp	auto	hp	hp	0	hp	hp	auto	hp	0	hp
6	hp	auto	hp	hp	auto	0	hp	auto	hp	hp	auto
7	hp	hp	hp	hp	hp	hp	0	hp	hp	hp	hp
8	hp	hp	auto	0	hp	auto	hp	0	hp	auto	hp
9	hp	hp	hp	hp	hp	hp	hp	hp	0	hp	hp
10	hp	hp	hp	hp	0	hp	auto	hp	hp	0	hp
11	auto	hp	hp	auto	hp	hp	hp	hp	hp	hp	0

This results in the following optimal tour:

1→6→9→4→7→10→2→11→8→5→3→1 with a total energy dissipation of 39.6869 kJ and a total time of 27.27 seconds. It is important to note that since the DJI is a large quadrotor, a tolerance margin for *x* is given as an additional 20 cm farther from the tree and a 50 cm margin for altitude. This plot also shows a 'rubber band' like path, which is yet again prevalent in all solutions.

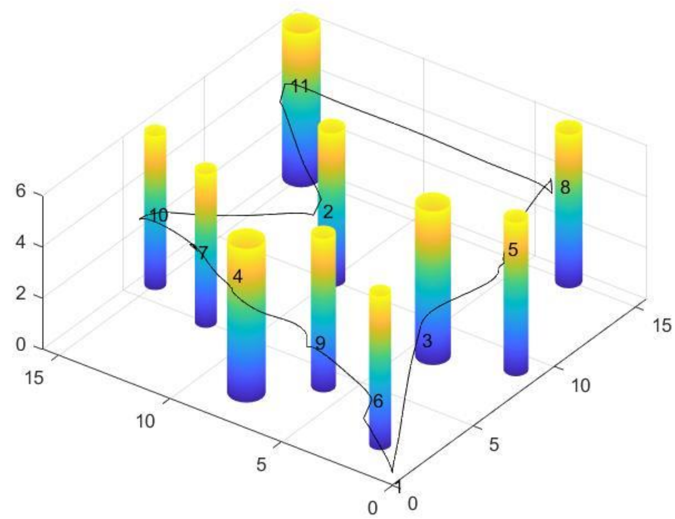

Figure 37 Hybrid DJI Energy-optimal Trajectory and Tour

The same procedure is performed on the Crazyflie 2.0 with the same criteria and inputs. And the following are the results.

Table 23 Crazyflie Auto-Direct Constraint Violation (m^2)

From\To	1	2	3	4	5	6	7	8	9	10	11
1	0	0.010458	0	0	0	0	0	0	0	0	0.018536
2	0.15235	0	0	0.0016915	0	0.00010955	0.0007774	0.00012816	6.5346e-05	0.000602	0.00018043
3	0	0	0	0.0040339	0	0.00092536	0	0.00046669	3.8345e-05	0	0
4	0	0.0012261	0.0038834	0	7.3451e-05	0	2.1427e-05	0.013013	7.0717e-05	0	0.0030541
5	0	0	0	0.0076085	0	0.00089086	0.0074373	0.00058828	0.00030661	1.4678e-05	0
6	0	0.0013487	0.00035905	0	0.00016822	0	0	0.0011533	0	0	0.0007107
7	0	0.00013747	0	4.8189e-05	0.010329	0	0	8.3431e-05	0	0	1.6367e-05
8	0	0.00015035	0.00088538	0.013146	0.00038457	0.0016839	0.0021746	0	0.010683	0	0
9	0	0.00037675	0.00015829	0.00047399	0.00029254	0	0	0.010688	0	0.00020532	0.001487
10	0	0.00064243	0	0	0.00093293	0	0	0	0	0	0.00063823
11	0.018583	0.00011688	0	0.0039565	0	0.0029589	0.00091278	0	0.0019674	0.001146	0

Table 24 Crazyflue Auto-direct Energy Matrix (J)

From\To	1	2	3	4	5	6	7	8	9	10	11
1	0	129.77	110.87	115.81	116.78	90.863	126.99	132.97	0	135.93	148.65
2	0	0	105.09	115.94	114.53	121.27	100.64	112.11	107.67	103.59	106.11
3	111.18	104.98	0	115.03	100.68	104.54	112.51	113.69	96.606	121.33	127.5
4	115.84	115.97	115.32	0	117.36	105.54	100.65	132.84	96.609	111.65	133.21
5	117.06	114.5	100.8	144.69	0	111.33	123.04	105.77	109.16	131.5	132.81
6	91.037	121.04	104.66	105.56	111.45	0	118.5	130.3	95.42	128.38	141.39
7	127.09	107.22	112.4	100.9	137.33	118.59	0	127.76	102.98	89.407	119.53
8	133.24	112.32	113.58	132.59	105.61	131.25	127.93	0	124.68	132.35	123.48
9	107.07	107.58	96.541	96.938	108.93	95.611	102.91	125.09	0	114.54	130.17
10	136.02	103.34	121.28	111.83	131.26	128.46	89.553	132.18	114.71	0	116.83
11	148.83	106.36	127.66	133.17	132.92	141.55	119.28	123.47	130.26	116.58	0

Table 25 Crazyflie Energy Matrix using h-method (J)

From\To	1	2	3	4	5	6	7	8	9	10	11
1	0	129.77	110.87	115.81	116.78	90.863	126.99	132.97	12171	135.93	0.018536
2	130	0	105.09	115.94	114.53	121.27	100.64	112.11	107.67	103.59	0.00018043
3	111.18	104.98	0	115.03	100.68	104.54	112.51	113.69	96.606	121.33	0
4	115.84	115.97	115.32	0	117.36	105.54	100.65	132.84	96.609	111.65	0.0030541
5	117.06	114.5	100.8	144.69	0	111.33	123.04	105.77	109.16	131.5	0
6	91.037	121.04	104.66	105.56	111.45	0	118.5	130.3	95.42	128.38	0.0007107
7	127.09	107.22	112.4	100.9	137.33	118.59	0	127.76	102.98	89.407	1.6367e-05
8	133.24	112.32	113.58	132.59	105.61	131.25	127.93	0	124.68	132.35	0
9	107.07	107.58	96.542	96.938	108.93	95.611	102.91	125.09	0	114.54	0.001487
10	136.02	103.34	121.28	111.83	131.26	128.46	89.553	132.18	114.71	0	0.00063823
11	148.83	106.36	127.66	133.17	132.92	141.55	119.28	123.47	130.26	116.58	0

Table 26 h-method Crazyflie constraint violation (m^2)

From\To	1	2	3	4	5	6	7	8	9	10	11
1	0	0.010458	0	0	0	0	0	0	0	0	0.018536
2	0.010507	0	0	0.0016915	0	0.00010954	0.00077739	0.00012813	6.5378e-05	0.000602	0.00018043
3	0	0	0	0.0040339	0	0.00092535	0	0.00046669	3.8367e-05	0	0
4	0	0.0012261	0.0038834	0	7.3451e-05	0	2.1271e-05	0.013013	7.0654e-05	0	0.0030541
5	0	0	0	0.0076085	0	0.00089086	0.0074373	0.00058828	0.00030657	1.4677e-05	0
6	0	0.0013487	0.00035909	0	0.00016822	0	0	0.0011533	0	0	0.0007107
7	0	0.00013747	0	4.8078e-05	0.010329	0	0	8.3431e-05	0	0	1.6367e-05
8	0	0.00015035	0.00088538	0.013146	0.00038457	0.0016839	0.0021745	0	0.010683	0	0
9	0	0.00037675	0.00015829	0.00047399	0.0002925	0	0	0.010688	0	0.00020532	0.001487
10	0	0.00064243	0	0	0.00093328	0	0	0	0	0	0.00063823
11	0.018583	0.00011684	0	0.0039565	0	0.0029589	0.00091278	0	0.0019674	0.001146	0

Table 27 Crazyflie p/hp-method Energy Matrix (J)

From\To	1	2	3	4	5	6	7	8	9	10	11
1	0	122.2	106.79	110.63	111.42	89.468	120.01	126.44	103.3	127.29	151.03
2	122.87	0	162.91	110.84	109.74	115.5	132.25	107.84	104.04	100.5	102.73
3	107.03	101.65	0	110.07	98.277	101.06	107.97	108.55	94.158	130.43	120.43
4	111.73	111.05	110.74	0	134.27	102.87	98.505	127.5	95.202	107.81	141.02
5	112.53	125.76	98.95	112.35	0	107.42	117.2	102.16	105.67	123.9	154.77
6	89.914	115.31	106.15	102.04	106.65	0	113.21	125.85	93.618	126.16	131.72
7	120.63	132.12	108.46	97.929	129.24	113.43	0	120.79	118.06	88.268	128.92
8	127.41	131.29	109.36	141.05	107.41	123.66	145.12	0	134.09	166.46	117.44
9	103.81	103.75	94.397	94.572	134.15	93.618	117.82	117.92	0	109.87	122.58
10	127.87	100.5	115.86	107.34	123.6	121.56	88.268	126.34	1073.4	0	111.12
11	137.95	102.97	121.02	125.02	154.56	132.11	128.82	117.44	122.97	111.66	0

Table 28 Crazyflie p/hp-method constraint violation (m^2)

From\To	1	2	3	4	5	6	7	8	9	10	11
1	0	0.055188	0	0	0	0	0	0.15973	0	0	0.22485
2	0.0001465	0	0.054129	0.0017261	0	0.0013551	0.014005	0.0040756	0.0029904	0.0013222	3.5037e-05
3	0	0	0	0.021015	0	0.009182	0	0.0041508	0.0012264	0.0024611	4.1054e-05
4	0	0.011112	0.010097	0	0.04418	0	0	0.084034	0.0001873	0	0.0084949
5	0	0.00047424	0	0.00084472	0	0.009472	0	0.0083875	0.0015307	0.060046	0.14961
6	0	1.5037e-05	0.0033991	0	0.0032167	0	0	0.1127	0	0.054247	0.018848
7	0	0.0079104	0	0.0018683	0.078855	0	0	0.0012242	0.010476	0	0.056907
8	0.15249	0.022431	0.00758	0.097165	0.0041951	0.018846	0.035714	0	0.034301	0.042924	0
9	0	0.0081003	0.0041052	0.0031892	0.017123	0	0.0087492	0.014168	0	0.04306	0.0031765
10	0	0.0039376	0	0	0.015764	0	0	0.067669	0.043966	0	0.0096787
11	0.10188	0.001545	0.00043466	0.020043	0.19181	0.013238	0.046996	0	0.00031485	0.010966	0

Table 29 Hybrid Solvers used for Crazyflie

From\To	1	2	3	4	5	6	7	8	9	10	11
1	0	0	hp	hp	hp	hp	hp	h	hp	hp	0
2	hp	0	auto	hp	hp	hp	auto	hp	hp	hp	hp
3	hp	hp	0	0	hp	hp	hp	hp	hp	hp	hp
4	hp	auto	auto	0	0	hp	hp	0	hp	hp	auto
5	hp	hp	hp	hp	0	hp	hp	hp	hp	auto	auto
6	hp	hp	auto	hp	hp	0	hp	h	hp	auto	h
7	hp	auto	hp	hp	0	hp	0	hp	auto	hp	0
8	auto	auto	hp	0	hp	0	auto	0	0	0	hp
9	hp	hp	hp	hp	auto	hp	auto	0	0	0	hp
10	hp	hp	hp	hp	auto	hp	hp	auto	0	0	hp
11	0	hp	hp	0	auto	0	0	hp	hp	auto	0

Table 30 Hybrid Constraint violation (m^2)

From\To	1	2	3	4	5	6	7	8	9	10	11
1	0	0	0	0	0	0	0	0	0	0	0
2	0.0001465	0	0	0.0017261	0	0.0013551	0.0007774	0.0040756	0.0029904	0.0013222	3.5037e-05
3	0	0	0	0	0	0.009182	0	0.0041508	0.0012264	0.0024611	4.1054e-05
4	0	0.0012261	0.0038834	0	0	0	0	0	0.0001873	0	0.0030541
5	0	0.00047424	0	0.00084472	0	0.009472	0	0.0083875	0.0015307	1.4678e-05	0
6	0	1.5037e-05	0.00035905	0	0.0032167	0	0	0.0011533	0	0	0.0007107
7	0	0.00013747	0	0.0018683	0	0	0	0.0012242	0	0	0
8	0	0.00015035	0.00758	0	0.0041951	0	0.0021746	0	0	0	0
9	0	0.0081003	0.0041052	0.0031892	0.00029254	0	0	0	0	0	0.0031765
10	0	0.0039376	0	0	0.00093293	0	0	0	0	0	0.0096787
11	0	0.001545	0.00043466	0	0	0	0	0	0.00031485	0.001146	0

$$Adj_{optimal} = \begin{pmatrix} 0 & 0 & 106.79 & 110.63 & 111.42 & 89.468 & 120.01 & 132.97 & 103.3 & 127.29 & 0 \\ 122.87 & 0 & 105.09 & 110.84 & 109.75 & 115.5 & 100.64 & 107.84 & 104.04 & 100.5 & 102.73 \\ 107.03 & 101.65 & 0 & 0 & 98.277 & 101.06 & 107.97 & 108.55 & 94.158 & 130.43 & 120.43 \\ 111.73 & 115.97 & 115.32 & 0 & 0 & 102.87 & 98.505 & 0 & 95.202 & 107.81 & 133.21 \\ 112.53 & 125.76 & 98.95 & 112.35 & 0 & 107.42 & 117.2 & 102.16 & 105.67 & 131.5 & 132.81 \\ 89.914 & 115.31 & 104.66 & 102.04 & 106.65 & 0 & 113.21 & 130.3 & 93.618 & 128.38 & 141.39 \\ 120.63 & 107.22 & 108.46 & 97.929 & 0 & 113.43 & 0 & 120.79 & 102.98 & 88.268 & 0 \\ 133.24 & 112.32 & 109.36 & 0 & 107.41 & 0 & 127.93 & 0 & 0 & 0 & 117.44 \\ 103.81 & 103.75 & 94.397 & 94.572 & 108.93 & 93.618 & 102.91 & 0 & 0 & 0 & 122.58 \\ 127.87 & 100.5 & 107.34 & 107.34 & 131.26 & 121.56 & 88.268 & 132.18 & 0 & 0 & 111.12 \\ 0 & 102.97 & 0 & 0 & 132.92 & 0 & 0 & 117.44 & 122.97 & 116.58 & 0 \end{pmatrix}$$

This results in an optimal tour with the following optimal sequence 1→3→5→8→11→2→10→4→9→6→1 with a maximum tour of 1.093 kJ and a total time of 17.86s. It is significant to see that the sequence of the trees is the same, but exactly flipped. This comes from the fact that altitude energy dissipation in ascent and descent are almost the same since the Crazyflie has a very small mass. This can also be true when dealing with planar horizontal motion where, if ignoring rotational motion energy dissipation, the energy is the same and is directly related to the distance. Moreover, this results in the Figure 38 Crazyflie Hybrid Energy-Optimal Tour tour.

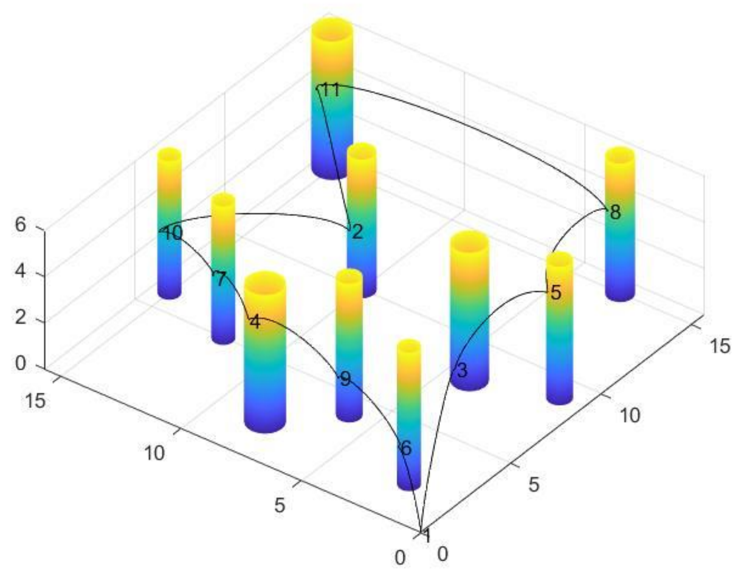

Figure 38 Crazyflie Hybrid Energy-Optimal Tour

One of the most important aspects is to note where the system fails in the cases mentioned above. This can be deduced that the DJI Phantom2 suffers a larger constraint violation due to its slower dynamics whereas Crazyflie2.0 is a more agile and light-weight platform, which can perform more aggressive dynamic maneuvers that in turn results in less constraint violations. Moreover, convergence is affected by how dense the forest is, i.e. how close the trees are to each other in addition to their radii. It is important to note that this greatly effects the rate of convergence as very close and very large trees might result in a no solution, even without a minimization effect, which is due to the fact that this might result in a collision. This is evident when dealing with small-diameter trees and less dense regions, which result in very low or no constraint violation with very high convergence rate. Finally, non-convergence might also result from a close proximity between the start and end points, which is impractical in real-life.

It is important to note the advantages that this hybrid method brings which is an acceptable constraint violation and a higher overall convergence rate which aids in the best possible result relative to results should in the previous section.

G. Full energy optimal tour with polynomial fit estimation

One of the most disadvantageous parts of using optimal control theory to determine the optimal travelling distance is the time needed to find a solution. This increases with the number of segments, path constraints, and order of the polynomial fit. Taking an accurate representation each solution requires around 30 seconds, but this time may vary based on the path and object avoidance. Moreover, the hybrid method proposed before took three hours for the DJI and five hours for the Crazyflie. As mentioned in Chapter III, it is safe to assume a third order polynomial where energy is a function of $\{\Delta x,\Delta y,\Delta z\}$ and Δ represents the difference between initial and final point. Each drone is represented by two equations and they are composed of the descent and ascent ones. After sampling a large number of points in 3-dimensional space, an accurate representation for the Crazylie equation is obtained:

$$
\begin{aligned}
Energy = {} & 3.16018\, z - 0.0163302\, z^2 + 2.14292\, y - 0.0332852\, y\, z + 0.000109225\, y\, z^2 + 0.00245168\, y^2 \\
& + 6.85036 \times 10^{-5}\, y^2\, z + 2.32395\, x - 0.0333332\, x\, z + 0.000100367\, x\, z^2 - 0.0283166\, x\, y \\
& + 0.000212874\, x\, y\, z - 9.90361 \times 10^{-6} x\, y^2 - 0.000802606\, x^2 + 7.55852 \times 10^{-5} x^2\, z \\
& + 3.93758 \times 10^{-5}\, x^2\, y + 85.5608 - 2.25231 \times 10^{-5}\, x^3 - 3.59313 \times 10^{-5} y^3 \\
& + 4.46233 \times 10^{-5}\, z^3
\end{aligned}
$$

$$
where\ z = \Delta z > 0\,, x = |\Delta x|, y = |\Delta y|. \tag{92}
$$

This fit has a R^2 value of 98.45% with a mean absolute error of 0.117 and a mean standard deviation of absolute error of 0.6723. On the other hand, for the descent the equation is shown as follows:

$$
\begin{aligned}
Energy = {} & -4.64959z - 0.0374038\, z^2 + 4.46176\, y + 0.0887795\, y\, z + 0.00119135\, y\, z^2 - 0.0354293\, y^2 \\
& - 0.001546\, y^2\, z + 4.48275\, x + 0.0913392\, x\, z + 0.00113585\, x\, z^2 - 0.181196\, x\, y \\
& + 0.0141618\, x\, y\, z + 0.00152685\, x\, y^2 - 0.0357207\, x^2 - 0.0016975\, x^2\, z + 0.00131784\, x^2\, y \\
& + 67.0816 + 0.000120273\, x^3 + 0.000119289\, y^3 - 0.000124581\, z^3
\end{aligned}
$$

$$
where\ z = \Delta z < 0\,, x = |\Delta x|, y = |\Delta y|. \tag{93}
$$

This fit has a R^2 value of 97.8% with a mean absolute error of 0.1786 and a mean standard deviation of absolute error of 0.7538. The map used for the Crazyflie is used again to find the accuracy of this estimation. This results in the following estimated adjacency matrix:

$$
Adj_{estimated} = \begin{pmatrix}
0 & 133.83 & 112.93 & 119.09 & 120.92 & 99.281 & 128.25 & 134.23 & 111.57 & 135.97 & 153.81 \\
138.42 & 0 & 107.39 & 114.2 & 112.6 & 126.3 & 95.304 & 114.04 & 109.3 & 102.18 & 109.62 \\
113.55 & 108.85 & 0 & 118.82 & 106.53 & 105.76 & 114.17 & 112.62 & 101.41 & 118.34 & 130.98 \\
122.41 & 115.01 & 120 & 0 & 119.411 & 106.02 & 104.44 & 134.1 & 97.712 & 117.09 & 137.32 \\
123.8 & 114.7 & 101.29 & 119.41 & 0 & 111.27 & 128.02 & 103.07 & 116.08 & 135.08 & 137.38 \\
90.453 & 122 & 102.34 & 108.07 & 112.18 & 0 & 116.04 & 122.51 & 91.63 & 124.17 & 143.15 \\
134.55 & 95.304 & 115.87 & 107.45 & 122.62 & 119.76 & 0 & 127.98 & 100.16 & 84.072 & 116.38 \\
141.86 & 115.59 & 112 & 127.76 & 106.2 & 128.43 & 133.65 & 0 & 125.97 & 138.34 & 116.23 \\
111.25 & 110.12 & 95.091 & 104.09 & 115..6 & 91.63 & 103.67 & 121.78 & 0 & 112.04 & 132.19 \\
143.6 & 102.18 & 123.43 & 115.53 & 126.8 & 131.28 & 84.072 & 131.66 & 113.97 & 0 & 108.56 \\
152.24 & 108.69 & 136.16 & 130.88 & 129.52 & 147.94 & 118.85 & 116.23 & 137.42 & 107.41 & 0
\end{pmatrix}
$$

To validate whether these estimations are accurate, this matrix is compared to that of the "auto" result given in the first adjacency matrix of the Crazyflie. This results in the following percent error matrix:

Table 31 Crazyflie Energy Estimation Error Matrix (%)

From\To	1	2	3	4	5	6	7	8	9	10	11
1	N/A	-3.1314	-1.8623	-2.8308	-3.5449	-9.2648	-0.99327	-0.95132	-4.3755	-0.029861	-3.471
2	-6.4801	N/A	-2.185	1.4982	1.6886	-4.1479	5.3022	-1.7193	-1.5093	1.3593	-3.305
3	-2.1302	-3.6899	N/A	-3.2929	-5.8128	-1.1636	-1.4737	0.94439	-4.9773	2.4665	-2.7277
4	-5.6709	0.83117	-4.0567	N/A	-1.7474	-0.45513	-3.7651	-0.94852	-1.1414	-4.8729	-3.0828
5	-5.7541	-0.17259	-0.48985	17.471	N/A	0.04945	-4.0437	2.5536	-6.3399	-2.7205	-3.4396
6	0.64169	-0.79662	2.2182	-2.3788	-0.65417	N/A	2.0754	5.9789	3.9717	3.2807	-1.2454
7	-5.8698	11.114	-3.0873	-6.4937	10.709	5.6361	N/A	-0.17069	2.7429	5.9665	2.6361
8	-6.4708	-2.9096	N/A	3.6432	-0.55996	2.1459	N/A	N/A	-1.0351	-4.5256	5.8699
9	-3.8997	-2.358	1.5017	-7.3771	-6.12	4.1636	-0.7418	2.6462	N/A	2.1788	-1.553
10	N/A	1.1207	-1.7694	-3.3101	3.397	-2.1914	6.1198	0.39369	0.64888	N/A	7.0752
11	N/A	-2.1932	-6.6616	1.7189	2.5546	-4.5112	0.35941	5.8622	-5.4952	7.8632	N/A

After placing the adjacency matrix in the TSP solver, the result is the following tour: 1→3→5→8→11→2→10→4→9→6→1, which is the same as the solution obtained by the hybrid solver. This results in a total estimated cost of 1.195 kJ, which is similar to that of “auto” solution with an error of 0.3% in the total cost estimation.

The same estimation is done for the DJI quadrotor and results in the following equation for ascent:

$$
\begin{aligned}
Energy = {} & +95.439324\, z - 0.078971371\, z^2 + 87.245361\, y - 0.70305548\, y\, z + 0.0014528529\, y\, z^2 \\
& + 0.0064903477\, y^2 + 0.00075131172\, y^2 z + 87.245361\, x - 0.70305548\, x\, z \\
& + 0.0014528529\, x\, z^2 - 0.81620358\, x\, y + 0.0032182792\, x\, y\, z + 0.0012956933\, x\, y^2 \\
& + 0.0064903477\, x^2 + 0.00075131172\, x^2 z + 0.0012956933\, x^2 y + 3159.1738 \\
& - 0.00039275357\, x^3 - 0.00039275357\, y^3 - 0.00032578517\, z^3
\end{aligned}
$$

$$where\ z = \Delta z > 0\,, x = |\Delta x|, y = |\Delta y|. \tag{94}$$

This fit has an R^2 value of 99.89% with a mean absolute error of 0.013 and a mean absolute of standard deviation error of 0.279. While for the descent, the following equation is used:

$$Energy = -80.528041\,z - 0.0039481647\,z^2 + 94.624272\;y + 0.57838575\;y\,z + 0.0010467519\;y\,z^2$$
$$- 0.14898889\;y^2 - 0.00079747153\;y^2 z + 94.624272\,x + 0.57838575\,x\,z$$
$$+ 0.0010467519\,x\,z^2 - 0.75202339\,x\;y - 0.0021665595\,x\;y\,z + 0.0012972654\,x\;y^2$$
$$- 0.14898889\,x^2 - 0.00079747153\,x^2 z + 0.0012972654\,x^2\,y + 3223.9805$$
$$+ 0.00012321658\,x^3 + 0.00012321658\;y^3 + 0.00040678188\,z^3$$
$$where\ z = \Delta z < 0\,, x = |\Delta x|, y = |\Delta y|. \tag{95}$$

This equation has an R^2 value of 99.94% with a mean absolute error of 0.0079 and a mean absolute standard deviation error of 0.0118. This resulted in the following adjacency matrix:

$$Adj_{estimated} = \begin{pmatrix} 0 & 5024 & 4205.3 & 4309.2 & 4419.8 & 3647.8 & 4808.1 & 4976.3 & 4144.7 & 5126.8 & 5844.3 \\ 517206 & 0 & 4134.3 & 4221.1 & 4197.4 & 4735.2 & 3865.4 & 4265.5 & 4215.1 & 4045.7 & 4083.7 \\ 4331.1 & 4039.3 & 0 & 4377.8 & 3891.6 & 3916.2 & 4255.4 & 4133.4 & 3750.5 & 4429.5 & 4914.1 \\ 4445.7 & 4324.9 & 4457.4 & 0 & 4528.8 & 4053 & 4062.6 & 4947.7 & 3864.8 & 4418 & 5096.1 \\ 4477 & 4296.8 & 3929.6 & 4528.8 & 0 & 4191.8 & 4730.8 & 4037.6 & 4368.9 & 4909.6 & 5070 \\ 3709 & 4581 & 4019.4 & 3988 & 4116.5 & 0 & 4356.1 & 4533.3 & 3778.2 & 4685.1 & 5428.7 \\ 4935.4 & 3865.4 & 4363.2 & 3974.3 & 4599.3 & 4489.1 & 0 & 4820.6 & 3957.4 & 3598.8 & 4330.8 \\ 5090.2 & 4396.5 & 4209.8 & 4786.3 & 3933.2 & 4650.5 & 4991 & 0 & 4646.1 & 5160.1 & 4355.4 \\ 4246.1 & 4096.9 & 3841.6 & 3811.5 & 4276.8 & 3778.2 & 3859.8 & 4521.8 & 0 & 4196.4 & 4970.1 \\ 5273.6 & 4045.7 & 4544.6 & 4304.3 & 4772.8 & 4838.3 & 3598.8 & 4980 & 4317.6 & 0 & 4007.8 \\ 6030.3 & 4201.4 & 5053.2 & 4922.5 & 4900.9 & 5621.1 & 4462.4 & 4355.4 & 5132.5 & 4111.6 & 0 \end{pmatrix}$$

This adjacency matrix is compared with the hybrid method developed in the earlier section, and it results in the following error table:

Table 32 DJI Energy Estimation Error Matrix (%)

From\To	1	2	3	4	5	6	7	8	9	10	11
1	N/A	2.9901	-0.75275	-0.11537	-0.81189	-30.676	-Inf	6.3297	-4.8877	N/A	N/A
2	N/A	N/A	-6.8635	6.3918	5.4693	-0.21066	-7.6427	-0.089183	-6.0255	-6.8687	-4.5658
3	-3.7664	-4.534	N/A	N/A	-18.758	-3.2863	0.7551	N/A	-13.136	7.2404	2.8959
4	-1.4945	N/A	1.1432	N/A	0.12467	-4.9218	-14.666	N/A	N/A	-4.7714	4.4278
5	-2.2451	3.0637	-19.852	0.12687	N/A	-2.1908	2.4555	-2.7109	-9.4713	N/A	5.5259
6	-32.83	3.258	-5.9268	-3.2209	2.082	N/A	6.0921	15.166	-11.892	8.9883	6.1261
7	N/A	-7.5947	-1.7161	-12.293	5.2292	3.2209	N/A	4.9588	-3.8251	-18.058	5.8902
8	4.0438	-3.1621	3.2499	N/A	-2.1744	11.108	1.5864	N/A	4.6495	3.792	11.375
9	-7.4139	-3.0425	-15.631	-26.216	-7.2247	-11.892	-1.2748	7.2933	N/A	5.2724	N/A
10	4.2479	-6.8687	4.7576	-2.0668	N/A	6.0028	-17.036	6.2972	2.5369	N/A	10.001
11	78.659	-7.5871	0.02932	8.961	8.6847	2.4971	3.0377	11.375	0.91519	N/A	N/A

After placing the adjacency matrix in the TSP, the result is the following tour: 1→6→9→4→7→10→11→2→8→5→3→1 with a maximum of 43.5475 kJ and a total estimation of total cost error of 9.7% compared to the hybrid approach. It is important to note that the tour is the same as the result for the hybrid one, but there is a small switch between trees 2 and 11, which still maintains the 'rubber band' like path that is normally attained.

As it is evident this method is recommended if total accuracy is not required, but rather a need for a quick and near optima result is needed. It gives a primary guess for the total and specific energy consumption, but does not determine the tree-to-tree route which needs to be done only for the route specified by the tour significantly decreasing computational time.

H. Using time as a minimizing factor

In order to determine the efficiency of the proposed system, it is beneficial to change the minimization objective of the "system of systems" approach. In this section, a minimization of final time is done for the DJI Phantom2 across the same map using "auto-direct" transcription. This also correlates to a minimization of final time on both stages of the system, which includes optimal control solution and the travelling salesman problem. This results in the following tour: 1→6→4→7→10→11→2→8→5→3→9→1. This results in a minimum time of 27.363 seconds with maximum energy consumption of 103.6139 kJ.

$$Adj_{time} = \begin{pmatrix} 0 & 3.3894 & 2.8167 & 3.0409 & 3.04 & 0 & 3.368 & 3.5442 & 0 & 3.6384 & 3.9669 \\ 3.4201 & 0 & 2.6775 & 3.0366 & 3.0018 & 3.1389 & 2.5483 & 2.8569 & 2.7494 & 0 & 2.6378 \\ 2.8709 & 2.6669 & 0 & 3.0157 & 2.3872 & 2.6738 & 0 & 2.9527 & 2.3931 & 3.2114 & 3.3513 \\ 3.042 & 0 & 3.1205 & 0 & 0 & 0 & 0 & 3.6133 & 2.1824 & 2.8519 & 3.5314 \\ 3.0847 & 2.9981 & 2.402 & 0 & 0 & 2.8754 & 3.2562 & 2.6946 & 0 & 3.5176 & 3.5325 \\ 0 & 3.111 & 2.6551 & 2.6942 & 208624 & 0 & 3.1107 & 3.5874 & 2.3495 & 3.4107 & 3.7375 \\ 3.766 & 2.5083 & 0 & 2.4806 & 3.2431 & 3.12 & 0 & 3.3184 & 2.6309 & 2.1519 & 3.1327 \\ 3.5804 & 0 & 2.9391 & 3.5204 & 2.7089 & 4.0776 & 0 & 0 & 11.528 & 3.4906 & 12.001 \\ 7.2043 & 0 & 0 & 2.2676 & 2.7247 & 0 & 2.626 & 3.3468 & 0 & 2.9905 & 3.4036 \\ 3.6457 & 2.5717 & 0 & 2.8715 & 3.4931 & 3.4203 & 0 & 0 & 3.0043 & 0 & 3.0758 \\ 3.9867 & 2.6733 & 3.3691 & 3.5383 & 3.5443 & 3.7548 & 3.1318 & 3.2799 & 3.4268 & 3.0508 & 0 \end{pmatrix}$$

It is evident that the energy consumption is extremely high, especially since the solver requires the quadrotor to operate at maximum allowable power limits to reach the desired waypoints in minimal time, which results in a disregard of energy consumption and a large emphasis on the inputs which is gravely non-optimal when seeking energy optimality. This results in the following path:

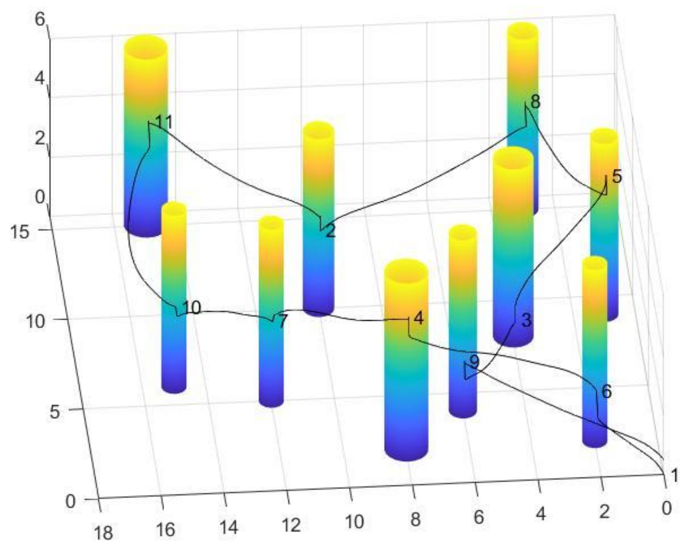

Figure 39 Time optimal path

Table 33 Energy Consumption with minimization of time (J)

From\To	1	2	3	4	5	6	7	8	9	10	11
1	0	73621	0	89713	4.4235e+06	45470	71901	46514	0	44341	2.6329e+05
2	29465	0	59397	47671	58176	1.8193e+05	78694	87752	0	56147	1.0717e+05
3	97156	2.6465e+05	0	2.8281e+05	4.4259e+05	4.0147e+05	2.0603e+05	4.6483e+05	90833	14901	26034
4	2.1784e+05	39284	0	0	18451	36845	90708	1.2363e+05	2.0012e+05	58672	80915
5	0	1.864e+05	18846	0	0	1.3908e+05	28883	48109	1.104e+05	89563	51793
6	1.7599e+05	27262	3.2009e+05	68184	51077	0	3.6806e+05	3.1253e+05	0	29483	0
7	0	1.8808e+05	57716	1.1813e+05	93359	25982	0	31796	1.5019e+05	38375	0
8	24549	92152	0	1.6531e+05	86463	1.5907e+05	2.6389e+05	0	0	49278	30231
9	1.839e+05	48612	85008	92301	50011	1.3825e+05	47445	0	0	1.2935e+05	0
10	44466	2.8073e+05	23751	3.1706e+05	78468	80222	2.6334e+05	25038	54477	0	2.3774e+05
11	0	87868	73388	0	1.2155e+05	0	2.0609e+05	0	1.8678e+05	1.4504e+05	0

Table 34 Final Time Optimization Constraint Violation (m^2)

From\To	1	2	3	4	5	6	7	8	9	10	11
1	0	0.0057638	0	0	0	0	0	0	0	0	0.0014239
2	0.00020031	0	0	0.0044426	0	0.0015211	0.00021535	0.00051877	0.00013696	3.4437e-05	0.00098192
3	0	0	0	0.0013652	0	0.0017622	0	0.00099642	0.0011456	0	0.00024594
4	0	0.00224	0.010834	0	0.00041788	0	0	0.0052602	8.1077e-05	0	0.0032966
5	0	0.005498	0	0.00068024	0	0.0018896	0.017276	0.00077768	0.00030626	0.022062	0
6	0	1.3448e-05	0.00073247	0	0.00071222	0	0	0.026481	0	0	0.038723
7	0	0.00014755	0	0.00036862	0	0	0	1.4221e-05	0	0	0.014605
8	0	0.0015438	0.0034616	0.012211	0.0018794	0.0039366	0.001174	0	0.0045888	7.5016e-05	0
9	0	6.3177e-06	0.00059632	0.0012671	0.0007047	0	0	0.0027481	0	0	0.0014603
10	0	0	0.012671	0	0.012741	0	0	0.0014309	0.0020583	0	0.00082248
11	0.02572	0.0014523	0.0013114	1.7764e-05	0	0.057902	0.0026485	0.00098217	0.0021125	0.0034027	0

I. New tour with probability distribution

To start with, it was important to create a 3D probability distribution with a lognormal and a normal distribution with a max value 1 centered at 0. The following are the constants that govern the probability distribution:

$$\mu_{log} = 3$$

$$\mu_{normal} = 0$$

$$\sigma_{log} = 1$$

$$\sigma_{normal} = 3$$

This results in the Figure 40 Probability distribution (top view) curved probability distribution:

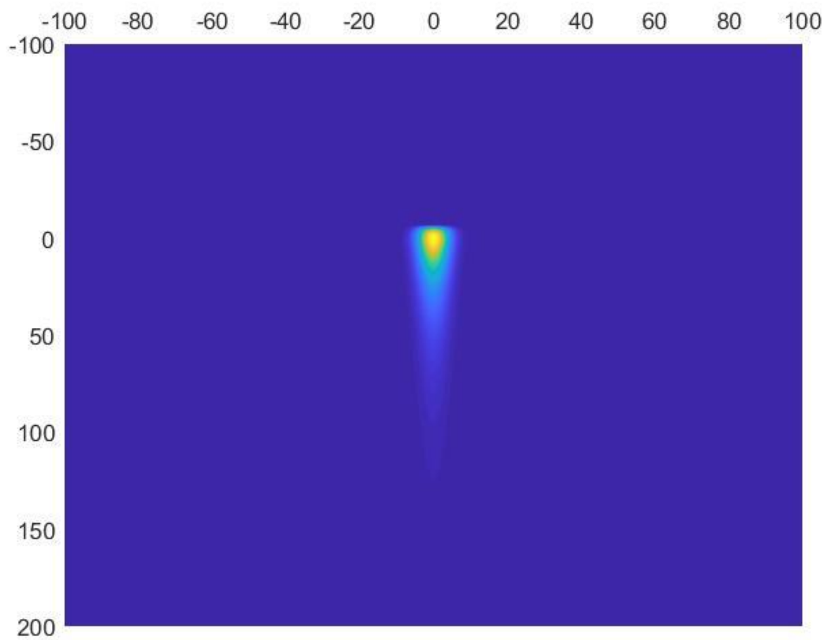

Figure 40 Probability distribution (top view)

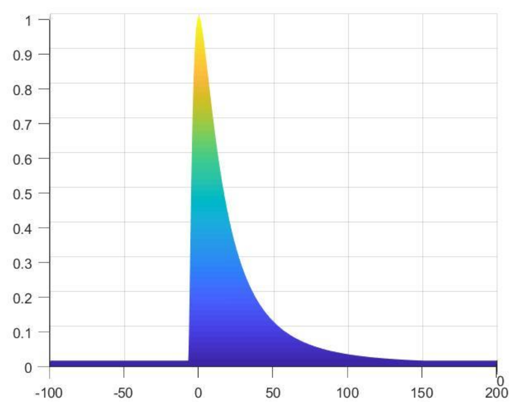

Figure 41 Probability distribution (side view)

In order to plot the location of each point with respect to the other, the Euclidean distance for the distance and the angle between the horizontally projected vector and the (1,0) vector. This results in 2-dimensional cylindrical coordinates and a relative coordinate system centered around the tree at hand. For example, the relative location of all the points with respect to tree 2 is shown as follows:

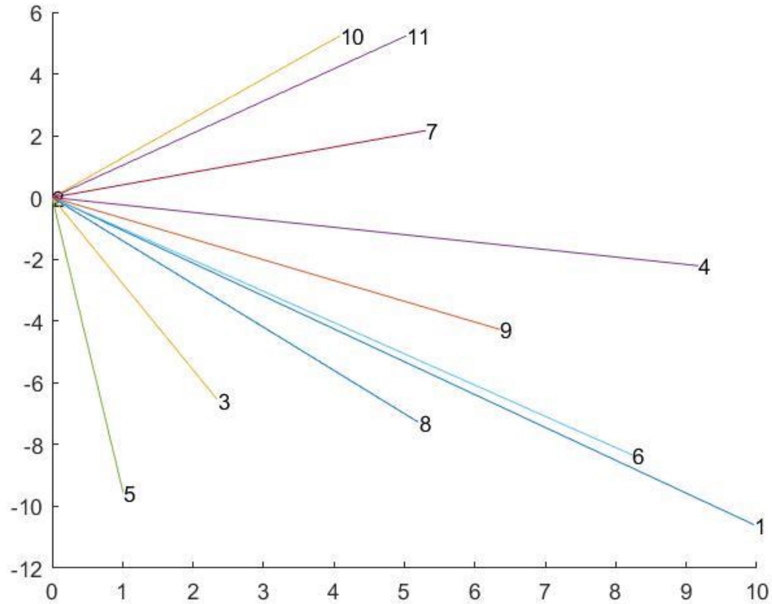

Figure 42 Relative position of all points with respect to node 2

This results in the following generalized probability distribution matrix where each item is the probability of infection of the column item such that the row item is infected.

Table 35 Probability of Infection Matrix

From\To	1	2	3	4	5	6	7	8	9	10	11
1	0	0	0	0	0	0	0	0	0	0	0
2	0	1	0.12327	0.016735	0	0	0.25599	0.0021953	0.098506	0.37584	0.23636
3	0	0.62393	1	0.058351	0.47627	0.14403	0.48513	0.035328	0.41652	0.54879	0.038537
4	0	0.017123	0.041376	1	0.00065703	0.058416	0.43176	2.7025e-05	0.49213	0.25411	5.5453e-05
5	0	0.68352	0.72119	0.03982	1	0.041471	0.26826	0.12924	0.16683	0.34487	0.089704
6	0	0.02316	0.14723	0.8028	0.041338	1	0.41765	7.6812e-05	0.73721	0.244	6.8537e-05
7	0	0.25066	0	0.32956	0	0	1	0	0.21019	0.89258	0.0048127
8	0	0.19554	0.03544	4.4948e-05	0.12644	7.6615e-05	0.0030458	1	0.00098038	0.0075441	0.6265
9	0	0.12244	0.40711	0.52304	0.08043	0.59762	0.78412	0.00090365	1	0.57343	0.0010358
10	0	0.22719	0	0.00037992	0	0	0.82605	0	0	1	0.014114
11	0	0.13965	0	8.0977e-07	0	0	0.0044484	0	0	0.014114	1

Notice that the rows and columns associated with 1 are all zeros since it is the base. This results in the following consecutive changes within, whereas the first sequence is the offline one and the online version is when each item is visited, it shows how the probability distribution and the set is changing when the cut-off PoI is 0.7.

Table 36 Probability based Offline and Time Varying (Online) Status, PoI, and Sequence

1 →3→7→10→11→8→5→1										
1	2	3	4	5	6	7	8	9	10	11
0	0	0	0	1	0	1	1	0	0	1
inf	0.6835	0.7212	0.3296	1	0.0415	1	1	0.2102	0.8926	1
1 →**3**→7→10→11→8→5→1										
1	2	3	4	5	6	7	8	9	10	11
0	0	0	0	1	0	1	1	0	0	1
inf	0.6835	0.7212	0.3296	1	0.0415	1	1	0.2102	0.8926	1
1 →3→**7**→10→11→8→5→1										
1	2	3	4	5	6	7	8	9	10	11
0	0	0	0	1	0	0	1	0	0	1
inf	0.6835	0.7212	0.3296	1	0.0415	0.2683	1	0.1668	0.3449	1
1 →3→7→**11**→8→5→1										
1	2	3	4	5	6	7	8	9	10	11
0	0	0	0	1	0	0	1	0	0	0
inf	0.6835	0.7212	0.0398	1	0.0415	0.2683	1	0.1668	0.3449	0.6265
1 →3→7→11→**8**→5→1										
1	2	3	4	5	6	7	8	9	10	11
0	0	0	0	1	0	0	0	0	0	1
inf	0.6835	0.7212	0.0398	1	0.0415	0.2683	0.1292	0.1668	0.3449	0.0897
1 →3→7→11→8→**5**→1										
1	2	3	4	5	6	7	8	9	10	11
0	0	0	0	1	0	0	0	0	0	1
inf	0.6835	0.7212	0.0398	1	0.0415	0.2683	0.1292	0.1668	0.3449	0.0897
1 →3→7→11→8→5→**1**										

It is important to indicate that the PoI of 1 is artificially always placed as infinity (inf) in order to ensure that is always in the set since it is the base. It is also important to note where tree number 10 is dropped out of the set, especially when it was not yet visited while its PoI dropped below the cut-off margin. These consecutive tests of the online version resulted in the Figure 43 Final tour probability after time varying path changes final tour and energy dissipation of 26.206 kJ and a total time of 17.6785 seconds. While, the original offline path yields a 19.7418 seconds total time and a 29.106 kJ for total energy consumption.

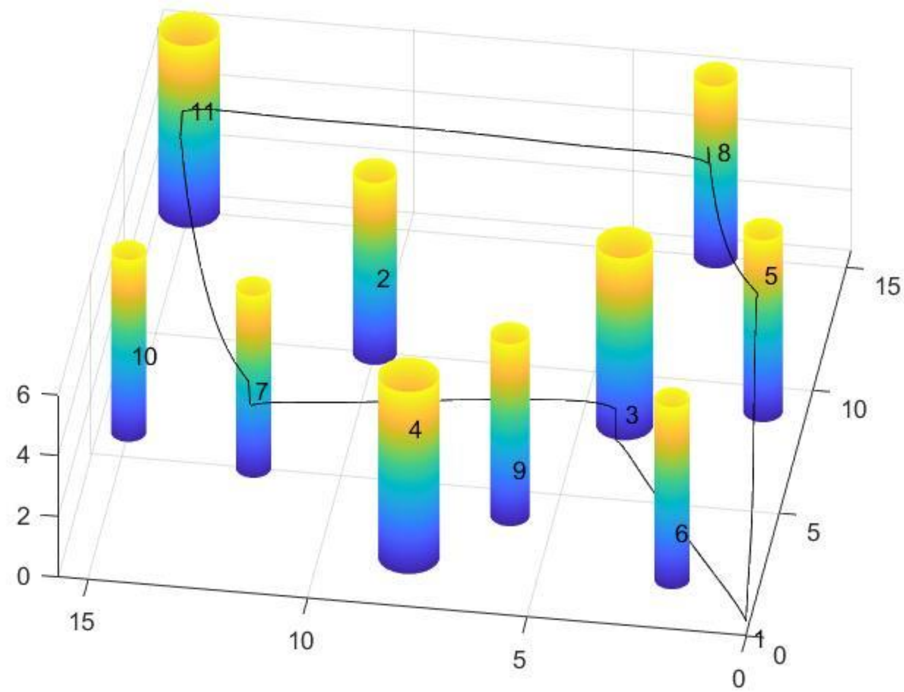

Figure 43 Final tour probability after time varying path changes

CHAPTER VII

CONCLUSION

As a generalized solution, the proposed solver generates energy-optimal tours to visit and scan cylindrical objects. This thesis suggests a *system of systems* that starts with determining individual energy consumption and optimal tour planning between each pair of trees in both directions. This includes a hybrid system of LGR, H-S, and Auto-Direct collocation and transcription methods that result in a minimum energy consumption path and minimal constraint violation. However, as this method or individual solvers may have high computational cost when dealing with large number of trees, energy consumption based on a polynomial fit estimation is provided as an alternative solution with adequate accuracy.

After determining the energy consumption of each route, a travelling salesman problem using ILP/SEC is used to generate an energy optimal tour. Moreover, a probability distribution is used in order to determine a prioritization of objects based on their respective probability of infection, which results in a subset of trees that are required to be visited. However, as the quadrotor visits each object or tree, the probability changes as new events are found, which results in a constantly changing pool of tree subsets.

One of the most important parts in this work is to note that energy consumption is proportional to the energy capacity for each quadrotor's battery. It is evident that when the energy capacity of the battery surpasses that of the entire tour, such an analysis opens the door towards increasing the total number of trees that one drone can visit. It would also yield a higher rate of scanning and can result in using a smaller number of drone when dealing with swarms to scan large forests.

In summary, it is important to point out that the proposed method in this thesis has proven to be effective in determining the optimal sequence and tour for quadrotors to scan cylindrical objects, especially with the contributions it made on the following fronts: collision avoidance, solving with free-end-time, convergence rate, energy consumption estimation based on coordinate inputs, and transforming an offline solution to an online option with real-time implementation prospects.

Given the proposed system-level approach, it is significant to note the contributions that this thesis brings. Primarily, the optimal control solution includes several aspects of new contributions that includes a free-end-time approach in addition to object avoidance, which also results in a converged optimal solution. Obtaining a higher convergence rate using a hybrid method for path planning is of great significance, especially when dealing with a field that places a lot of emphasis on cost inputs. Moreover, establishing a system-of-systems approach results in a varying input and requirements where each level can be optimized based on a certain cost function, such as energy or time or distance travelled, which results in a variation of inputs. In addition, adding a factor of probability with a varying updated route based on newly available information results in a new approach that gives new perspectives while maintaining the most optimal local solution.

As an outlook, future work may involve the development of the proposed algorithm to deal with additional obstacles (such as the terrain in the case of forests and trees) or other obstacles in various applications. Further effort can be exerted to reduce the computation time of the individual components of the algorithm, in addition to increasing the convergence rate and decreasing constraint violation. Last but not least, the results can be further refined via closed-loop parallel programing with mesh refinement.

REFERENCES

[1] N. Mazloum, "Fatal disease threatens Lebanon’s historic parasol pines," NEWSROOM, Beirut, 2017.

[2] M. Zaatari, "Moths pose Threat to pine tree reservation," The Daily Star, Beirut, 2016.

[3] L. Poulsom, "Contingency Plan for the Pine Processionary Moth (Thaumetopoea pityocampa)," Forestry Commission, 2016.

[4] Invasive Species COmpendium, "Thaumetopoea pityocampa (pine processionary)," CABI, [Online]. Available: https://www.cabi.org/isc/datasheet/108482.

[5] J. Q. D. S. J. X. J. H. Y. X. Liang Yang, "Survey of Robot 3D Path Planning Algorithms," *Journal of Control Science and Engineering,* vol. 2016, p. 22, 2016.

[6] A. K. Z. H. Iram Noreen, "A Comparison of RRT, RRT* and RRT*-Smart Path Planning Algorithms," *International Journal of Computer Science and Network Security,* vol. 16, no. 10, 2016.

[7] R. b. Omar, "PATH PLANNING FOR UNMANNED AERIAL VEHICLES USING VISIBILITY LINE-BASED METHODS," Unversity of Leicester Department of Engineering , Leicester, 2011.

[8] W. W. c. H. C. q. X. Y. b. Han Tong, "Path planning of UAV based on Voronoi diagram and DPSO," *Procedia Engineering,* vol. 29, pp. 4198-4203, 2012.

[9] N. A. P. M. T. D. Celia Tazibt, "UAV-based Data Gathering using An Artificial Potential Fields Approach," in *IEEE 88th Vehicular Technology Conference,*, Chicago, 2018.

[10] M. K. P. H. G. M. S. Mohammadreza Radmanesh, "Overview of Path-Planning and Obstacle Avoidance Algorithms for UAVs: A Comparative Study," *Unmanned Systems,* vol. 6, no. 2, pp. 95-118, 2018.

[11] T. Liao, "UAV Collision Avoidance using A* Algorithm," Auburn University, Auburn, 2012.

[12] R. C. D. L. Fabio Morbidi, "Minimum-Energy Path Generation for a Quadrotor UAV," in *IEEE International Conference on Robotics and Automation*, Stockholm, 2016.

[13] A. B. L. G. L. G. P. D. P. Dadios, "Path Planning for Quadrotor UAV Using Genetic Algorithm," in *International Conference on Humanoid, Nanotechnology, Information Technology, Communication and Control, Environment and Management (HNICEM)*, Puerto Princesa City, 2014.

[14] Y. H. Z. Z. L. X. Xiaoyi, "Path planning for indoor UAV based on Ant Colony Optimization," in *Control and Decision Conference*, 2013.

[15] A. L. G. E. P. D. Argel Bandala, "Predicting the motion of quadrotor using neural network," in *2015 International Conference on Humanoid Nanotechnology, Information Technology, Communication and Control, Environment and Management (HNICEM)*, Cebu City, 2015.

[16] C.-S. L. C.-L. L. C.-H. H. Yi-Ju Tsai, "Development of Flight Path Planning for Multirotor Aerial Vehicles," 2015.

[17] C. Brucato, "THE TRAVELING SALESMAN PROBLEM," University of Pittsburgh, Pittsburgh, 2013.

[18] S. S. M. M. Rajesh Matai, "Traveling Salesman Problem: an Overview of Applications, Formulations, and Solution Approaches," in *Traveling Salesman Problem, Theory and Applications*, inTech, 2010.

[19] T. V. Roy Jonker, "Transforming asymmetric into symmetric traveling salesman problems: erratum," *Operations Research Letters,* vol. 5, no. 4, pp. 215-216, 1986.

[20] *Christofides’s Algorithm,* Design and Analysis of Algorithms, 2007.

[21] "Minimum Spanning Trees," pp. 257-252.

[22] N. K. S. I. S. T. N. FOZIA HANIF KHAN, "SOLVING TSP PROBLEM BY USING GENETIC ALGORITHM," *International Journal of Basic & Applied Sciences IJBAS,* vol. 9, no. 10, pp. 79-88, 2010.

[23] P. P. Saloni Gupta, "Solving Travelling Salesman Problem Using Genetic Algorithm," *International Journal of Advanced Research in Computer Science and Software Engineering,* vol. 3, no. 6, 2013.

[24] S. C. S. &. H. D. S. Maichel M. Aguayo, "Solving the single and multiple asymmetric Traveling Salesmen Problems by generating subtour elimination constraints from integer solutions," *IISE Transactions,* vol. 50, no. 1, pp. 45-53, 2017.

[25] D. W. a. G. Wachsmuth, "Necessary conditions for convergence rates of regularizations of optimal control problems," Chemnitz University of Technology, Department of Mathematics,, Chemnitz.

[26] M. P. Kelly, "Transcription Methods for Trajectory Optimization," Cornell University, 2015.

[27] E. B. F. B. M. M. N. Dal Bianco, "Comparison of direct and indirect methods for minimum lap time optimal control problems," *Vehicle System Dynamics,* vol. 0, no. 0, pp. 1-30.

[28] J. T. Betts, A Survey of Numerical Methods for Trajectory Optimization, Seattle: Boeing Information and Support Services,, 1998.

[29] M. Kelly, "An Introduction to Trajectory Optimization: How to Do Your Own Direct Collocation," *SIAM REVIEW,* vol. 59, no. 4, pp. 849-904, 2017.

[30] C. Z. F. Topputo, "Survey of Direct Transcription for Low-Thrust Space Trajectory Optimization with Applications," *Abstract and Applied Analysis,* vol. 2014, p. 15, 2014.

[31] M. P. W. H. A. R. D. B. G. H. Divya Garg, "An overview of three pseudospectral methods for the numerical solution of optimal control problems," University of Florida, 2017.

[32] P. T. Boggs, "Sequential Quadratic Programing," *Acta Numerica,* pp. 1-52, 1996.

[33] D. F. S. A. R. J. V. HANDE Y. BENSON, "A COMPARATIVE STUDY OF LARGE-SCALE NONLINEAR OPTIMIZATION ALGORITHMS," Princeton University, 2002.

[34] A. S. Nemirovski, "Interior-point methods for optimization," *Acta Numerica,* vol. 2008, pp. 191-234, 2008.

[35] J. Hedengren, "Interior Point Methods," Birgham Young University.

[36] M. D. Avsar, "The Relationships between Diameter at Breast Height, Tree Height and Crown Diameter in Calabrian Pines (Pinus brutia Ten.) of Baskonus Mountain, Kahramanmaras, Turkey," *Journal of Biological Sciences,* vol. 4, no. 4, pp. 437-440, 2004.

[37] M. D. A. a. V. Ayyildiz, "The Relationships between Diameter at Breast Height, Tree Height and Crown Diameter in Lebanon Cedars (Cedrus libani A. Rich.) of the Yavsan Mountain, Kahramanmaras, Turkey," *Pakistan Journal of Biological Sciences,* vol. 8, no. 9, pp. 1228-1232, 2005.

[38] M. T. C. A. P. S. F. A. C. D. J. F. P. O. C. a. J. S. P. A. C. Correia, "Biomass allometry and carbon factors for a Mediterranean pine (Pinus pinea L.) in Portugal," *Forest Systems,* vol. 19, no. 3, pp. 418-433, 2010.

[39] J. A. Preiss, W. Hönig, N. Ayanian and G. S. Sukhatme, " Downwash-aware trajectory planning for large quadrotor teams," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Vancouver, 2017.

[40] I. K. a. K. C. Yow, "Object Location Estimation from a Single Flying Camera," in *The Ninth International Conference on Mobile Ubiquitous Computing, Systems, Services and Technologies*, 2015.

[41] A. Gibansky, "Introduction," Quadcopter Dynamics and Simulation, 23 November 2012. [Online]. Available: http://andrew.gibiansky.com/blog/physics/quadcopter-dynamics/.

[42] N. R. O. B. a. M. H. F. Yacef, "Optimization of Energy Consumption for Quadrotor UAV," in *International Micro Air Vehicle Conference and Flight Competition (IMAV)*, Laval, 2017.

[43] A. Gibiansky, "Quadcopter Dynamics and Simulation," 2012.

[44] A. R. H. a. K. A. D. MOHD ARIFFANAN MOHD BASRI, "Stabilization and trajectory tracking control for underactuated quadrotor helicopter subject to wind-gust disturbance," *Sadhana,* vol. 40, no. 5, pp. 1531-1553, 2015.

[45] T. Luukkonen, "Modelling and control of quadcopte," Aalto University, 2011.

[46] E. K. Paola Falugi, Imperial College London Optimal Control Software User Guide (ICLOCS), London: Imperial College London, 2010.

[47] ˙. K. A. G. L. Temel Öncan, "A comparative analysis of several asymmetric traveling salesman problem formulations," *Computers & Operations Research,* vol. 36, pp. 637-654, 2009.

[48] B. W. Taylor, "Integer Programing: The Branch and Bound Method," in *Introduction to Management Science (10th Edition)*, 2006, p. 358.

[49] B. C. A.-M. D.-M. D. P.-C. S. H. J. Margot Régolini, "Effect of host tree density and apparency on the probability of attack by the pine processionary moth," *Forest Ecology and Management,* vol. 334, pp. 185-192, 2014.

[50] J. Forster, "System Identification of the Crazyflie 2.0 Nano Quadrocopter," Swiss Federal Institute of Technology (ETH) Zurich, Zurich, 2015.

[51] M. Greiff, "Modelling and Control of the Crazyflie Quadrotor for Aggressive and Autonomous Flight by Optical Flow Driven State Estimation," Lund University, 2017.

[52] D. Ariens, ACADO for Matlab User's Manua, Optimization in Engineering Center (OPTEC), 2010.

[53] bitcraze, "240mAh LiPo battery including 500mA USB charger," bitcraze, [Online]. Available: https://store.bitcraze.io/products/240mah-lipo-battery-including-500ma-usb-charger.

[54] DJI, "Phantom2 SPECS," DJI, [Online]. Available: https://www.dji.com/phantom-2.

[55] E. S. N. D. Christoph Aoun, "An Energy Optimal Path-Planning Scheme for Quadcopters in Forests," in *58th IEEE Conference on Decision and Control (CDC)*, Nice, 2019.

[56] J. Clausen, "Branch and Bound Algorithms - Principles and Examples.," 1999.