Original Paper

MEASUREMENT AND CONTROL

Measurement and Control
1–14

DOI: 10.1177/00202940231189507
journals.sagepub.com/home/mac
Sage

Control structure design with constraints for a slung load quadrotor system

Halit Ergezer[1] and Kemal Leblebicioğlu[2]

Abstract

We propose a control structure for a quadrotor carrying a slung load with swing-angle constraints. This quadrotor is supposed to pass through the waypoints at specified speeds. First, a cascaded PID autopilot is designed, which adaptively gives attention to position and speed requirements as a function of their errors. Its parameters are found from an optimization problem solved using the PSO algorithm. Second, this controller's performance is improved by adding the Complementary Controller employing an ANN. 5. Training data for the ANN is created by solving optimal control problems. The ANN is activated when the swing angle constraint is about to be violated. It is trained using optimal control values corresponding to the cases where the swing angle falls in a particular band about the upper swing angle constraint. Simulations are performed in a MATLAB environment. Finally, some of the simulation results are validated on a physical system.

Keywords

Slung load quadrotor system, particle swarm optimization, artificial neural networks, autonomous package delivery

Date received: 16 March 2023; accepted: 29 May 2023

Introduction

In recent years, with the technological developments in unmanned aerial vehicles (UAV), studies on package delivery,[1] agricultural applications,[2] search and rescue in dangerous areas,[3,4] and information gathering from desired regions,[5–7] and military applications have gained momentum.[8] Load transportation is vital in all these applications.[8] Two different methods are being studied for carrying loads with rotary-wing UAVs. One can carry a load using a manipulator[9–12] integrated into the vehicle. The other is to connect the load to the vehicle with a cable or rope.[13–15] In vehicles with a manipulator, the vehicle's inertia grows due to components associated with the manipulator – consequently, energy utilization of the vehicle with the manipulator increases. One of the biggest problems for UAVs is the short mission time. In addition, especially when it is required to perform agile movements, the growth in inertia will oppose the maneuvering. For these reasons, hanging loads seem more advantageous. However, using the suspended load method is also challenging because system dynamics change during the lift, hover, and flight. Even though we have not covered it in this study, we would like to emphasize that one of the important issues about load-bearing quadrotors is precise landing. This topic is covered in depth in Xuan-Mung et al.[16] and Xuan Mung et al.[17]

As Khamseh et al.[4] highlighted, a complete mathematical model covering all phases of flight is the biggest challenge. Models are created assuming the cable or rope is constantly tense during flight. Flight profiles that violate these assumptions should be prevented during flight (e.g. the vehicle descends quickly). Usually, there are swings during the flight due to the quadrotor's acceleration. These motions have distorting effects on the quadrotor's dynamics.

A controller should be designed to make the vehicle follow the desired path and minimize these swinging effects. The controller designed in this study is carried out based on these considerations.

In the literature, extensive studies have been published to model the suspended load-bearing rotary-wing UAV systems. Modeling was carried out using the Udwadia-Kalaba Equations in Goodarzi et al.[18] and Almeida and Raffo,[19] the Euler-Lagrange approach in

[1]Mechatronics Engineering Department, Çankaya University, Ankara, Turkey
[2]Electrical and Electronics Engineering Department, Middle East Technical University, Ankara, Turkey

Corresponding author:
Halit Ergezer, Mechatronics Engineering Department, Çankaya University, Eskişehir Yolu 29. Km, Yukarıyurtçu Mahallesi Mimar Sinan Caddesi No:4 06790, Etimesgut, Ankara 06790, Turkey.
Email: halitergezer@cankaya.edu.tr

Lee et al.[20] and Lee et al.,[21] the Newton-Euler method in Jain,[22] and the Kane method in Klausen et al.[23] In Pizetta et al.,[24] the system dynamics are defined as the second-order Euler-Lagrange model while ignoring the aerodynamic effects. In Barikbin and Fakharian,[25] the Euler-Lagrange method was used, but the load's motion was assumed to be planar rather than 3-dimensional. In this study, the controller is designed using the mathematical model detailed in Lee[26] based on the Euler-Lagrange approach.

Measuring and estimating suspended load oscillations is critical for controlling the system. The oscillation frequency caused by the suspended load has been investigated by Geronel et al.,[27] in which the dynamic model is constructed by obtaining dimensionless equations of motion. A method is proposed to determine the vibration frequency of the payload during flight. In recent years, studies have been focused on the controller design for such systems to reduce the oscillations of the suspended load. In Sadr et al.,[28] a nonlinear control algorithm is developed that applies a pulse to the system to prevent oscillation, whose amplitude and time locations are calculated by estimating the system's natural frequency and damping ratio. Due to the quadrotor's movements, the slung load will start to swing while moving to the desired point. This swinging action affects the quadrotor's traveling motion.[29] The suspended load can create unstable oscillations due to its disturbing dynamics and reduce the performance of the UAV. They may even cause crashes in some cases.[30] Therefore, it has been studied for a long time to solve the difficulties of UAVs in carrying suspended loads. Controller and path planning is the most critical parts of these studies. In Jain,[22] a controller is designed for a quadcopter carrying a point-mass load with a rigid cable. Predictive control is considered for carrying the payload from point to point with swing angle constraints along the desired trajectory.[23,24] Guerrero et al.[31] present a passivity based-controller design to minimize the load swing. However, a simplified 2D model is used in this study. In Kusznir and Smoczek,[32] sliding mode control is used for horizontal positioning and payload vibration damping, while a linearizing feedback controller is used for altitude and attitude control. In Yu et al.,[33] the backstepping procedure is applied to design a controller for the load to follow a pre-determined trajectory by defining a virtual thrust force. They find the actual forces using this virtual force. Another controller based on the backstepping method is proposed by Dai et al.[34] to reduce the swing angle of the load. There, the swing angle of the suspended payload is estimated using the harmonically extended state observer. In Shi et al.,[35] an adaptive control scheme with unknown mass transportation is proposed. Lee et al.[36] proposes an adaptive sliding mode controller to achieve the goal of altitude tracking control in the presence of considerable system parameter variations and the ground effect.

After choosing the controller type, its parameters should be determined to optimize the performance. Usually, evolutionary algorithms are preferred for this purpose; PSO (The PSO technique is used to design the PID controller to stabilize the humanoid robot,[37,38]) Ant Colony Optimization (ACO) technique,[39] Genetic Algorithm (GA),[40] Differential Evolution (DE),[41] (the pitch angle controller of a rocket system is optimized algorithm). Comparisons made by El-Ghandour and Elbeltagi[42] have demonstrated through in-depth studies that PSO is more efficient and has lower computational complexity simultaneously.

There are many reasons why Artificial Neural Networks are used in control applications. First, the self-learning ability of neural networks eliminates the use of complex and challenging mathematical analysis that predominates in many adaptive and optimal control methods. Second, it provides a nonlinear mapping capability to solve highly nonlinear control problems by incorporating the Activation Function into hidden neurons. Third, unmodelled dynamics and perturbations in the system are expected to be dealt with using an ANN. Thus, neural controllers can be applied over a broader range of uncertainty.

As ANN has proven to have excellent approximation and learning capabilities, they have become popular tools in control applications of nonlinear systems.[43,44] A unified framework for describing and controlling nonlinear dynamical systems is proposed in Polycarpou.[45] The adaptive nonlinear control and the parametric method of adaptive linear control theory can be applied to perform stability analysis. Typical stable neural network approach control schemes based on Lyapunov training designs are given in Mughees and Mohsin.[39] Training neural networks using a backpropagation algorithm (BP) has dramatically increased the development of these structures in control applications.[46] Many research studies have proven the approximation capabilities of neural networks.[47] In this context, many neural control approaches have been developed with BP neural networks.[45,48,49] Moreover, the ANN structure has been used in different control applications; Guo et al.[50] provides a decentralized robust optimal monitoring control scheme for the class of nonlinear systems interconnected with disturbances and actuator attacks, for which the optimal control problem is solved by NN-based adaptive dynamic programing.

We aim to make a quadrotor carry a suspended load following a given path without the load swinging too much. Here, the slung-load system is supposed to follow a path using the generalized waypoint guidance algorithm (waypoints and desired speeds are specified). The original contributions of this study are described below:

1. An autopilot is designed consisting of cascaded PID controllers for inner and outer loops. This autopilot is supposed to control the position and

velocity of the quadrotor and prevent high swing angles of the suspended load.
2. The primary controller adaptively changes (the percentage of) its attention to speed or position depending on its errors. While the PID coefficients are found, the controller gains associated with positions and velocities are adaptively tuned by gain adjustment (see equations (8)–(11)).
3. Finally, the whole structure is fine-tuned using delta force and torques produced by a Complementary Controller with an ANN to attenuate the oscillations if there is too much swinging.
4. Training data for the ANN is created by solving optimal control problems to calculate the attenuating delta force and torque inputs to the plant when the swing angle is above a certain threshold. These calculations are performed while the vehicle tries to follow specially chosen paths.

To the best of our knowledge, there is no such study for a slung-load quadrotor system.

The rest of the paper is organized as follows; Firstly, the dynamic model of the system is given in Section II. Secondly, the structure and design procedure of the controller are described in Section III. Section IV contains the experimental results. Then, this study is concluded in Section V.

Dynamic model

This section will describe the 3D dynamic model for the slung-load quadrotor system. It starts with the coordinate system definitions in the mathematical model, as shown in Figure 1. The mathematical model used in this study is borrowed from Lee.[26] The detailed derivation of this model can be found there.

This model is differentially flat; the flat outputs are the load's position and the quadrotor's yaw angle. Unlike Lee,[26] the 3D dynamic model is preferred over the flat model since we imposed constraints on the quadrotor's position and attitude. Additionally, different from previous studies,[46,51,52] we do trajectory planning for both the load and the quadrotor to avoid obstacles. The state vector is, $\boldsymbol{x}(t) = \left[\boldsymbol{x_L^T}\ \boldsymbol{p^T}\ \boldsymbol{R}\ \boldsymbol{\dot{x}_L^T}\ \boldsymbol{\dot{p}^T}\ \Omega^T\right]^T \in \mathbb{R}^{18}$ where $\boldsymbol{x_L}$ is the position vector of the load, and $\boldsymbol{p}$ is the unit vector from the quadrotor to the load in the inertial frame. The input vector is,

$\boldsymbol{v} = \left[f\ M_\phi\ M_\theta\ M_\psi\right]^T$ where $f = \mathrm{T1} + \mathrm{T2} + \mathrm{T3} + \mathrm{T4}$ is the lifting force acting on the system $\mathbf{M} = \left[M_\phi\ M_\theta\ M_\psi\right]^T$ are roll, pitch, and yaw moments (see Figure 1).

The dynamic model for the slung-load quadrotor system is valid under the following assumptions:

- The payload is connected with a cable at the quadrotor's center of gravity.

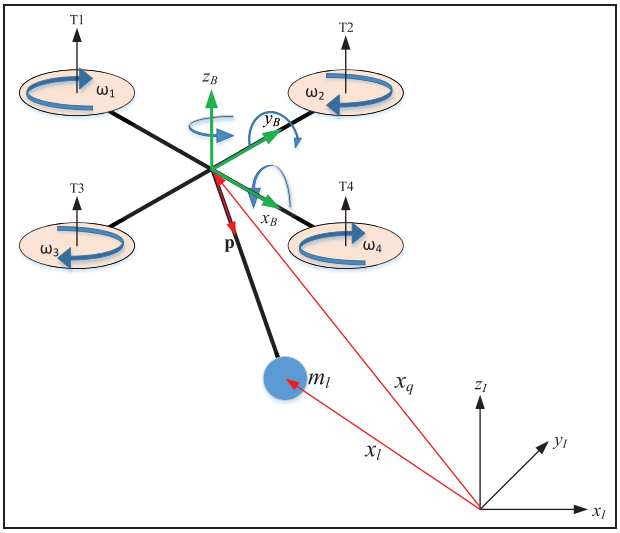

Figure 1. Slung-load quadrotor system coordinate frames.

- The cable is rigid, massless, and inelastic.
- The quadrotor has no external disturbances or interactions, such as wind, except for the load.

The input vector $\boldsymbol{v}$ is found as;

$$\boldsymbol{v} = \begin{bmatrix} f \\ M_\phi \\ M_\theta \\ M_\psi \end{bmatrix} = \begin{bmatrix} K_f & K_f & K_f & K_f \\ 0 & -K_f & 0 & K_f \\ K_f & 0 & -K_f & 0 \\ K_M & -K_M & K_M & -K_M \end{bmatrix} \begin{bmatrix} \omega_1^2 \\ \omega_2^2 \\ \omega_3^2 \\ \omega_4^2 \end{bmatrix} \tag{1}$$

where ω_j, $(j = 1, 2, 3, 4)$ is the angular velocity of rotor j, K_f aerodynamic force constant, K_M aerodynamic moment constant. The mathematical model equations are given below:

$$\frac{d}{dt}\boldsymbol{x_L} = \boldsymbol{\dot{x}_L} \tag{2}$$

$$\frac{d}{dt}\dot{x}_L = \frac{1}{m_q + m_l}\left(\boldsymbol{p}.f\boldsymbol{R}\boldsymbol{e_z} - m_q l||\boldsymbol{\dot{p}}||\right)\mathbf{p} - g\mathbf{e_z} \tag{3}$$

$$\frac{d}{dt}\mathbf{p} = \boldsymbol{\dot{p}} \tag{4}$$

$$\frac{d}{dt}\boldsymbol{\dot{p}} = \frac{1}{m_q l}(\mathbf{p}\times(\boldsymbol{p}\times f\boldsymbol{R}\boldsymbol{e_z}) - ||\boldsymbol{\dot{p}}||\mathbf{p} \tag{5}$$

$$\frac{d}{dt}\mathbf{R} = \mathbf{R}\hat{\boldsymbol{\Omega}} \tag{6}$$

$$\frac{d}{dt}\boldsymbol{\Omega} = \mathbf{I}^{-1}(\mathbf{M} - \boldsymbol{\Omega}\times\mathbf{I}\boldsymbol{\Omega}) \tag{7}$$

where $\mathbf{R}$ is the rotation matrix from Body Frame to Inertial Frame, $\mathbf{e_z}$ is the unit vector along the reference

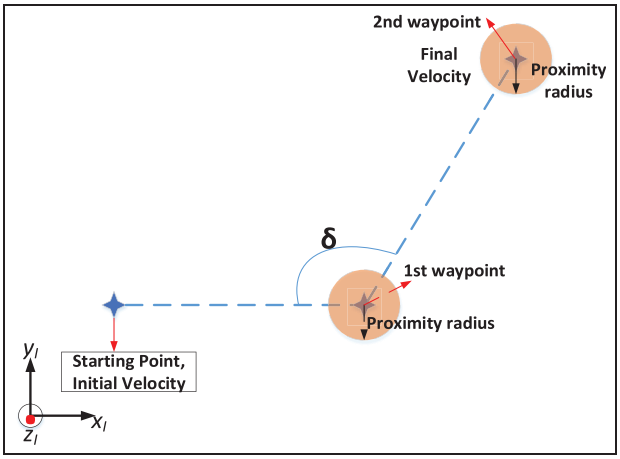

Figure 2. Waypoint geometry used in controller design.

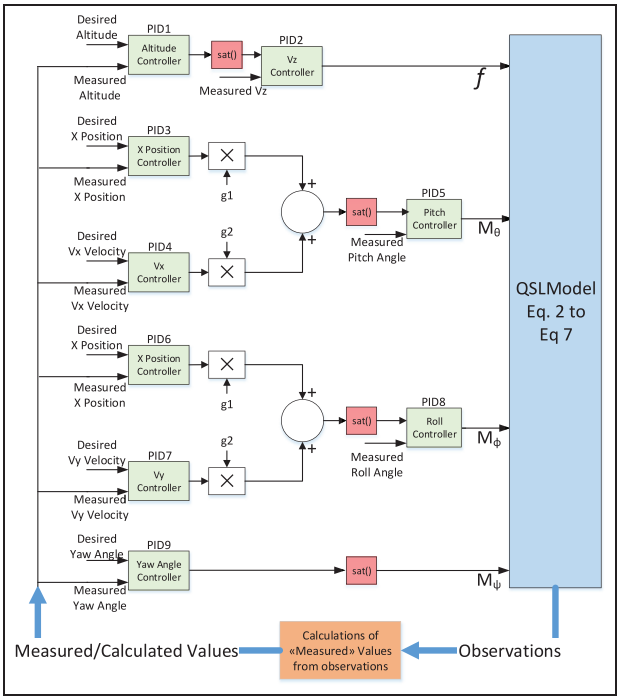

Figure 3. Controller structure without the complementary controller.

z-direction. $\boldsymbol{\Omega}$ is the angular velocity of the quadrotor. $\mathbf{I}$ is the inertia matrix, and m_q and m_l are the mass of the quadrotor and the load, respectively. $\hat{\boldsymbol{\Omega}}$ is the skew-symmetric matrix described in Lee.[53]

Controller design

Our study gives the waypoints the unmanned aerial vehicle should visit and the desired speed values at these points. When the system reaches a certain proximity to a desired waypoint, this point is considered already visited, and a command is produced to go to the next waypoint. Figure 2 shows this structure for two consecutive waypoints. In our study, as in the trajectory tracking approach, the entire path is not given, but only specific waypoints (the dashed lines between the waypoints are drawn to show the approach easily).

The slung-load quadrotor system is an underactuated system having 8 degrees of freedom (DOF) and 4 control surfaces. It is a challenging task to control this system with velocity constraints. The controller structure shown in Figure 3 has been created to overcome this challenge. In this figure, each controller block consists of a PID controller. The vehicle's roll and pitch angles must be changed for position and speed control. Since we want to perform speed and position control, another block for gain adjustment has been added. The gain adjustment block adjusts effectiveness by changing the roll or pitch angles according to speed and position errors. This effect is achieved by calculating the $g1$ and $g2$ parameters at each time step. They are set between 0.2 and 0.8 according to the error in speed and the angle between the current position and the target waypoint adaptively (equations (8)–(11)). Figure 3 shows the controller structure used for position and speed control. This structure does not have a complementary controller, and the swing angle constraint is not expected to be satisfied in scenarios not used to determine PID gains. The gain adjustment parameters $g1$ and $g2$ are calculated as follows:

$$\Delta_{direction} = abs\left(\frac{\arctan\left(\frac{wp_y - cp_y}{wp_x - cp_x}\right)}{\pi}\right) \tag{8}$$

$$\Delta_{speed} = \frac{(V_{desired} - V_{measured})}{V_{max}} \tag{9}$$

$$g1 = \min\left(\max\left(\frac{\Delta_{direction}}{\Delta_{direction} + \Delta_{speed}},\ 0.001\right),\ 1\right) \tag{10}$$

$$g2 = 1 - g1 \tag{11}$$

where V_{max} is the maximum allowable quadrotor's speed, wp_x *and* wp_y are the x *and* y coordinate of the target waypoint, cp_x *and* cp_y are the x *and* y coordinate of the quadrotor's current position. $\Delta_{direction}$ is the normalized angle between the quadrotor's current position and the target waypoint. Δ_{speed} is the normalized speed error. $\Delta_{direction}$ and Δ_{speed} are saturated between 0.001 and 1 to ensure not to get zero division error and to compensate for round-off errors.

Tuning the PID coefficients has been proposed as an optimization problem solved using the PSO algorithm. The problem definition is given below in detail.

Definition of the optimization problem

The control objective is to ensure that all waypoints are visited in a certain order and at the desired speed at the waypoint in minimum time. The system must visit the

waypoints sequentially. A proximity radius is defined for waypoints considered visited when the distance to a waypoint less than the proximity radius is. The proximity radius may change according to the task performed, which is considered one of the problem's input parameters.

$$\begin{aligned} &\min_{K_1,\ldots,\ K_{27}} J(K_1, \ldots, K_{27}) = \\ &(\alpha_1 \sum_j^N perror_j^2 + \alpha_2 \sum_j^N verror_j^2) + \alpha_3 T_{end} \\ &s.t. \\ &\quad |\beta_i| \leqslant \beta_{max}\ i = 0, \Delta t, 2\Delta t \ldots T_{end} \\ &\quad Eq.\ 2\ to\ 7 \end{aligned} \tag{12}$$

where $K_1, \ldots, K_{27}$ are the PID coefficients. For ease of expression, they are enumerated starting from the proportional gain of the PID1. (i.e. K_1 is the proportional gain of the PID1, K_2 is the integral gain of the PID1, and K_{27} is the derivative gain of the PID9). α_1, α_2, *and* α_3 values are appropriately selected, they create a convex sum. β_i is the swing angle at ith timestep, and β_{max} is the maximum allowable swing angle. T_{end} is the duration of the mission. Δt is the simulation time step ($\Delta t = 1msec$).The connection between optimization problem (also Algorithm-1) and the model is that the system's dynamical model is taken as the constraint of the optimization problem given in equation (12).

Details of the objective function calculation are given below.

$$perror_j = \begin{cases} 0\ j = next\ wp\ AND\ d_j \leqslant r_{proximity} \\ d_j\ j = next\ wp\ AND\ d_j > r_{proximity} \\ d_{max}\ j \neq next\ wp \end{cases} \tag{13}$$
$$j = 1, 2, \ldots N$$

where, $r_{proximity}$ is the distance required for the waypoint to be considered visited. It can be determined according to the mission to be performed. As stated before, the system must visit the waypoints in order. Considering this constraint, $perror_j$ is calculated. This calculation includes the system's distance to the waypoints at the time of the visit. d_j is the distance between the jth waypoint and the quadrotor's position. d_{max} is the furthest distance that the system can travel. For waypoints that have not yet been visited and are not the next waypoint in the visiting queue, $perror_j$ is taken as d_{max}.

$$verror_j = \begin{cases} ||v_d - v_a||\ d_j \leqslant r_{proximity} \\ V_{max}\ d_j > r_{proximity} \end{cases} \tag{14}$$

Since the speed objective is valid only within the radius of the proximity area, the function is written in parts, and the velocity error outside this region is considered the maximum allowable velocity value.

PID gains using particle swarm optimization

Our optimization problem (equations (12)–(14)) depends on the Simulink model; we prefer to use a derivative-free evolutionary algorithm. PSO is a population-based iterative algorithm.[49,54] Populations are structured with some topology comprising bidirectional edges connecting particle pairs.

The power of PSO comes from the interaction of individuals and movement to a better objective value that occurs when the individuals interact. All particles in the same neighborhood communicate so they move toward their new position, influenced by the best objective value obtained by the individuals in the topology. Topologies of the neighborhood are scrutinized in Poli et al.[47]

In PSO, many particles are positioned in the problem's search space, each evaluating the objective function at its present location. Each particle in the swarm is composed of three vectors in $\mathbb{R}^{27}$; the current position $\boldsymbol{x_i} \boldsymbol{\epsilon}\ \mathbb{R}^{27}$, the previous best position $\boldsymbol{p_i} \boldsymbol{\epsilon}\ \mathbb{R}^{27}$ and the velocity $\boldsymbol{v_i} \boldsymbol{\epsilon}\ \mathbb{R}^{27}$. At each iteration of the algorithm, $\boldsymbol{x}_i$ vectors are taken as the controller gains. If $\boldsymbol{x}_i$ is better, then the PID gains are stored in the second vector, $\boldsymbol{p_i}$. The best value in the neighborhood is denoted by $\boldsymbol{g} \boldsymbol{\epsilon}\ \mathbb{R}^{27}$

Data generation for ANN

The controller structure given in Figure 3 was tested in many different scenarios. During these trials, if the vehicle is supposed to make a sharp turn between consecutive waypoints, the swing angle of the load may violate the constraints. This problem has been solved by adding a complementary controller, which computes the required controller inputs using an ANN. This structure (see Figure 4) with ANN creates complementary control inputs when necessary. It accepts velocity and position error vectors as inputs and generates necessary force and moments as its outputs. If the swing angle of the load is within the specified range, this structure will not contribute to the overall control inputs.

The input data for training the ANN has been created by running the structure shown in Figure 3 for different and challenging scenarios. Instances that did not meet the angle constraint were found in the results of these runs. Figure 5 shows the plot of a swing angle obtained from a scenario and the points that do not satisfy the constraint value. A particular optimal control problem is solved for each point (indicated by red dots in Figure 5) in the zoomed image. In these optimal control problems, the aim is to obtain the value of the complementary control inputs that will bring the swing angle below the constraint value. A complementary input $\Delta \boldsymbol{v}_{ij}(t)\ \left(t = t_{ij} - \Delta T, \ldots, t_{ij}\right)$ have been added to

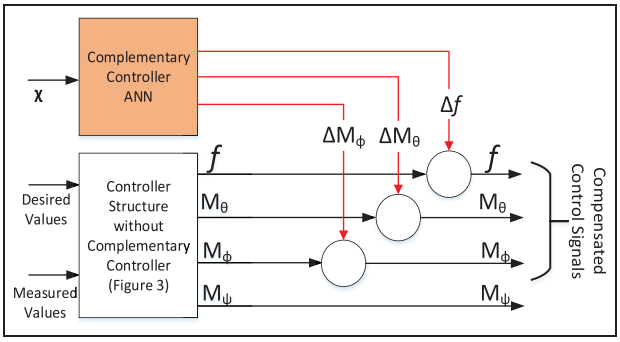

Figure 4. Overall controller structure with ANN.

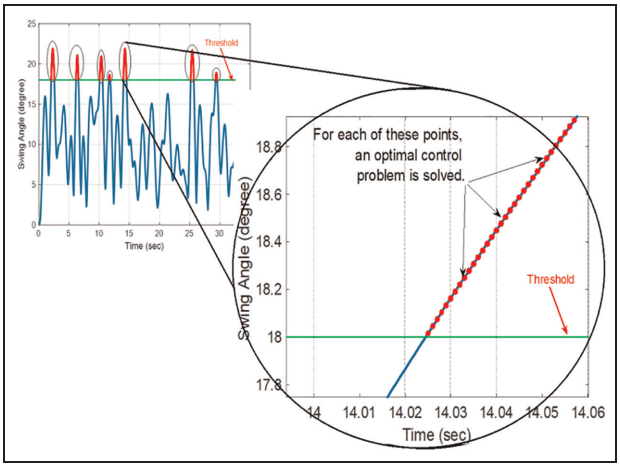

Figure 5. Points at which swing angle constraint is violated.

the model. Then, the optimal control problem is formulated as follows:

$$\min_{\Delta \boldsymbol{v}_{ij}(t)} \int_{t_{ij}-\Delta T}^{t_{ij}} (\max(\beta(t) - \varepsilon))^2 dt \tag{15}$$

subject to

Dynamic constraints of quadrotor slung load system (i.e., Eq. 3 − 8) and PID controllers.

where $i \in \{1, \ldots, N_s\}$ and $j \in \{1, \ldots, N_i\}$ are scenario and instance indices, respectively. The total number of scenarios is N_s, and the total number of instances in scenario i is N_i. So, we have obtained $N_s N_i$ points to train ANN. Eighty percent of these points were taken as training and 20% as validation sets. In equation (15), ε is inserted into the equation to satisfy this strict inequality constraint. The objective of the optimal control problem (equation (15)) is to determine the *complementary inputs* that should additionally be applied to the PID outputs such that the swing angle constraint is satisfied. The $\Delta \boldsymbol{v}_{ij}(t)$ values, which should be applied ΔT steps before the time step when the constraints are violated, were considered the optimal control problem variables. Since applying *complementary inputs* when the swing angle exceeds the constraint value does not guarantee the satisfaction of the constraint, the complementary controller should start contributing some time before a violation occurs. We preferred to use an interval of $\Delta T = 200$ msec (i.e. 200 time-step) by trial and error to take care of the system's inertia. These *complementary input* values obtained were used as ANN outputs during training. In the training process, ANN's inputs and outputs are:

Input: $\chi_{ij}(t) = \left[\phi\ \theta\ p_x\ p_y\ p_z\ V_x\ V_y\ V_z\ p\ q\ r\ \Omega_x\ \Omega_y\ \Omega_z\right]$,

Output: $\Delta \boldsymbol{v}_{ij}(t)$ (*complementary inputs* for force and moments).

Training of the ANN

The scaled conjugate gradient backpropagation algorithm Møller[55] is used because it avoids the line search per learning iteration. Instead, this algorithm uses gradient calculations to update weights instead of the Jacobian calculations used by other methods such as Levenberg-Marquardt and Bayesian regularization. The feedforward calculation has been given as follows;

$$NN_{out} = h\left(\sum\nolimits_{j=1}^{n_H} w_{kj} h\left(\sum_{i=1}^{m} w_{ji}\chi_i + w_{j0}\right) + w_{k0}\right) \tag{16}$$

where the subscript i indexes units on the input layer, j for the hidden; w_{ji} denotes the input-to-hidden layer weights at the hidden unit j, the subscript k indexes units in the output layer (three, in our case) and n_H denotes the number of hidden units (20, in our case). The activation function, h, is the hyperbolic tangent sigmoid activation function;

$$h(s) = \frac{2}{1 + e^{(-2s)}} - 1 \tag{17}$$

Simulation results

Results without using the complementary controller

A sufficiently challenging scenario with a 30° swing angle constraint is suggested to determine the PID coefficients using Algorithm 1. The top view of the resultant paths for the quadrotor and slung load is given in Figure 6. As shown in Figure 7, where the swing angle of the load is plotted with respect to time, the maximum swing angle has been obtained as 9.83°. In Figure 8, the graph of the vehicle's speed is presented, and the speeds at the waypoints are marked. It is observed that the speeds at the waypoints are close to the desired speeds. There are minor deflections from desired speeds due to the swinging of the load. A more challenging scenario with very sharp turning points is used in this

Algorithm 1: PSO-based PID tuning

Input: $w1$, $w2$ ▷ Adjustment Weights
Input : N, S ▷ Swarm Size, Neighbor fraction
Input : W ▷ Inertia Range
Output: $\vec{x}=[K_1, ..., K_{27}]$ ▷ PID Gains
$Stop \leftarrow false$;
GENERATESWARM(N, S);
for *Each Particle-i* **do**
  CHECKBOUNDS($\vec{x}_i$);
  $f_i \leftarrow$ QSLMODEL($\vec{x}_i$)
  $pbest_i \leftarrow f_i$ ▷ initial personel bests
end
$\vec{g} \leftarrow$ FINDBESTNEIGHBOR(**f**); ▷ g: neighbor pos.
while $Stop = false$ **do**
  for *Each Particle-i* **do**
    $\vec{v}_i \leftarrow W\,\vec{v}_i + w_1\,\vec{u}_1 \otimes (\vec{p}_i - \vec{x}_i) + w_2\,\vec{u}_2 \otimes (\vec{g} - \vec{x}_i)$
    $\vec{x}_i \leftarrow \vec{x}_i + \vec{v}_i$
    CHECKBOUNDS($\vec{x}_i$);
    $f_i \leftarrow$ QSLMODEL($\vec{x}_i$)
    if $f_i < pbest_i$ **then**
      $pbest_i \leftarrow f_i$
      $p_i \leftarrow x_i$
    end
  end
  $\vec{g} \leftarrow$ FINDBESTNEIGHBOR (**f**);
  UPDATENEIGHBORS();
  if *Objective Limit is reached* **then**
    $Stop \leftarrow true$
  end
end

new study (Scenario-2). In this scenario, first, PID coefficients are found using an angle constraint of 15°.

The resultant paths followed by the quadrotor and slung load are presented in 3D in Figure 9. The top view of these paths is shown in Figure 10. The swing angle of the load is presented in Figure 11. As observed in this figure, the maximum swing angle is 12.49°. In Figure 12, the vehicle's speed with respect to time is given.

Afterward, we wondered what happens to PID coefficients and the system's performance if the swing angle constraint is relaxed to 30° (using the same scenario with a different swing angle constraint, i.e. Scenario-2_2).

When we run Algorithm 1 with this constraint on this scenario, we obtain the paths given in Figure 13, and the corresponding swing angle is given in Figure 14. The maximum swing angle is 25.81° in this case. We want to point out that by relaxing the system's swing angle constraint (i.e. Figures 11 and 14) from 15° to 30°, task duration is decreased from 35.619 s to 25.764 s. This (i.e. decrease in task duration) is an expected result because as the system is more relaxed, it can behave faster to minimize costs further.

The PID coefficients found in the scenario with 15° swing angle constraint are tested on a new scenario (Scenario-3) shown in Figure 15. The resultant maximum swing angle (14.78°) can be observed in Figure 16, which is not much different than the result in Figure 11.

To check the sensitivity of PID autopilot coefficients concerning the path in the chosen scenario and the

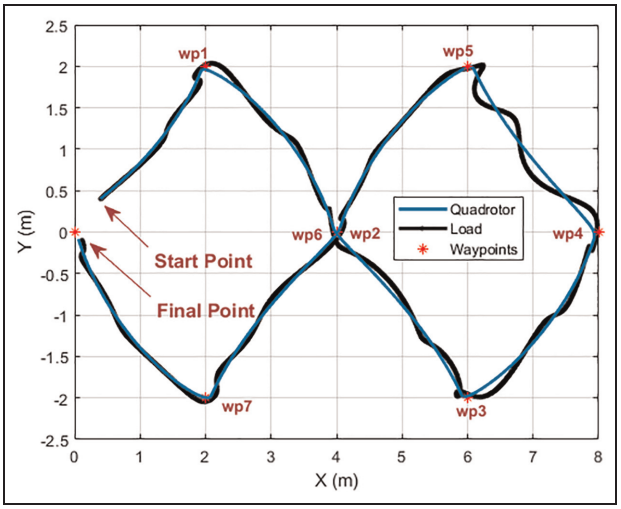

Figure 6. Top view of the path of quadrotor slung-load system (Scenario-1).

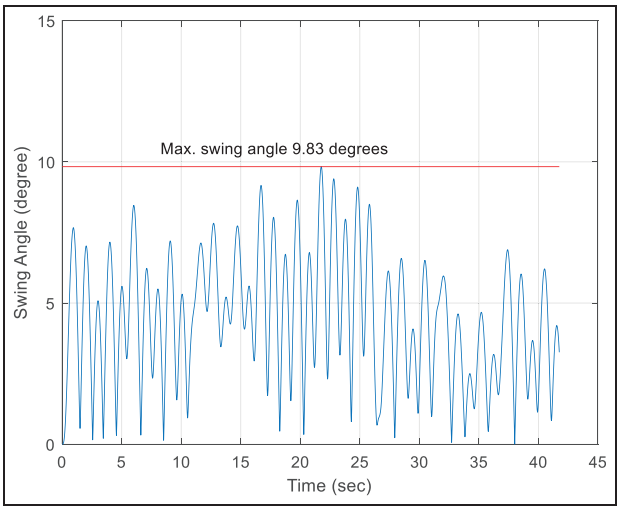

Figure 7. The swing angle of the load versus time (Scenario-1).

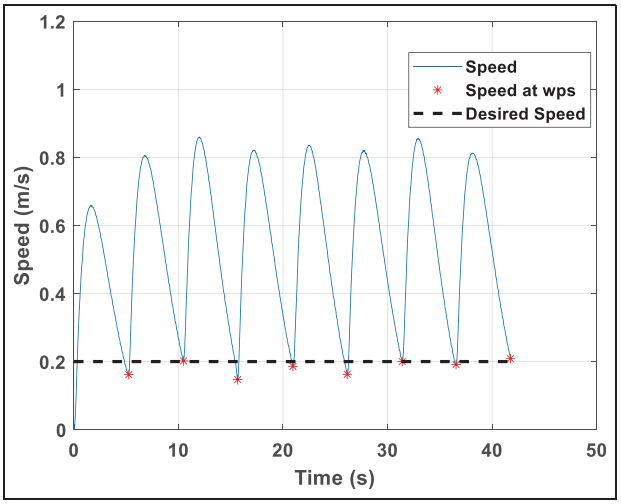

Figure 8. Quadrotor's speed versus time (Scenario-1).

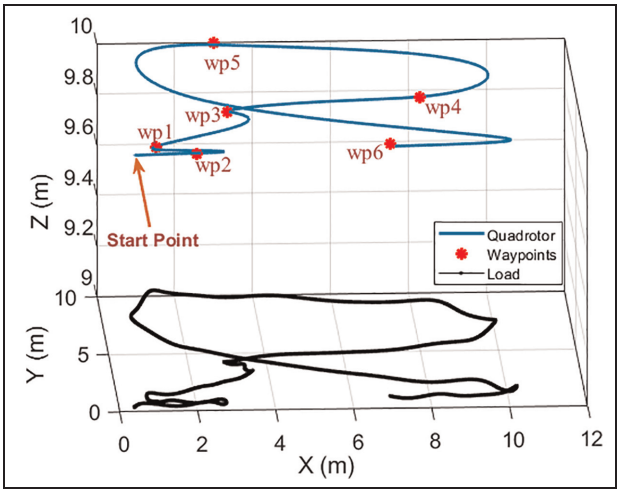

Figure 9. 3D view of the path of quadrotor slung-load system (Scenario-2_1).

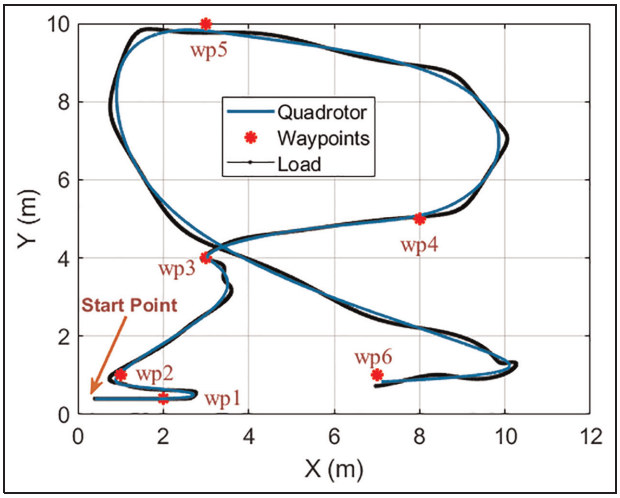

Figure 10. Top view of the path of quadrotor slung-load system (Scenario-2_1).

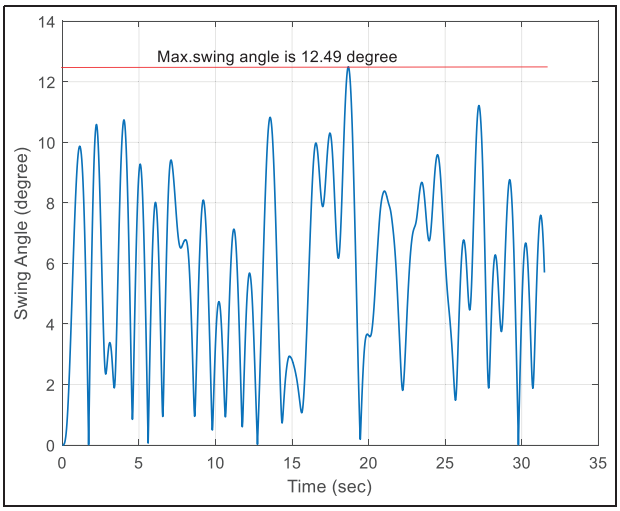

Figure 11. The swing angle of the load versus time (Scenario-2_1).

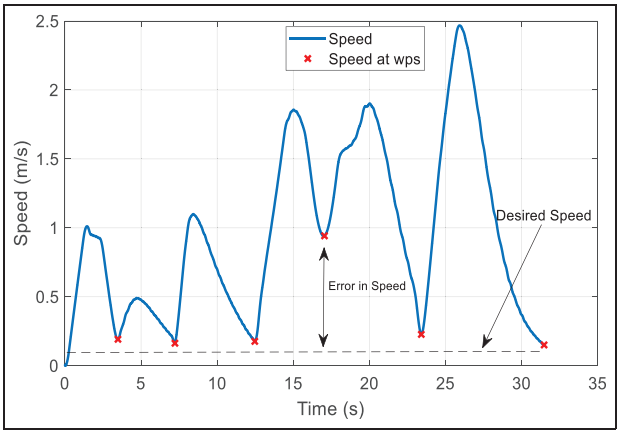

Figure 12. Quadrotor's speed versus time (Scenario-2_1).

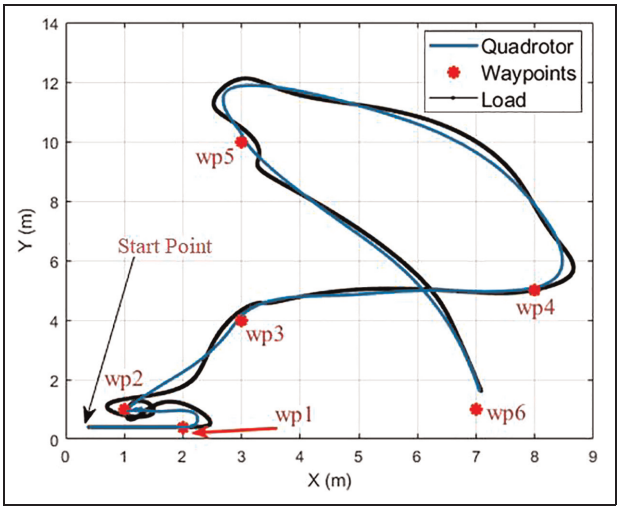

Figure 13. Top view of the path of quadrotor slung-load system (Scenario-2_2).

constraints, we performed the final test on a new scenario (Scenario-4), shown in Figure 17. This scenario has many sharp turns that may cause swing-angle constraint violations. Indeed, when the controller structure in Figure 3 is used, the swing angle of the load is given in Figure 18. The swing angle constraint for this scenario is 30°, but the resultant swing angle is about 38°. Therefore, since optimizing the PID coefficients for every possible path, desired speeds, and swing angle constraint is impossible, we have concluded that the autopilot structure should be extended/complemented in Figure 4.

Results using the complementary controller

The training data for ANN in improved structure is obtained from Figure 18 when the swing angle constraint was 30°. This new structure (with ANN) is tested on the same scenario (Scenario-4) with a swing angle constraint of 20°.

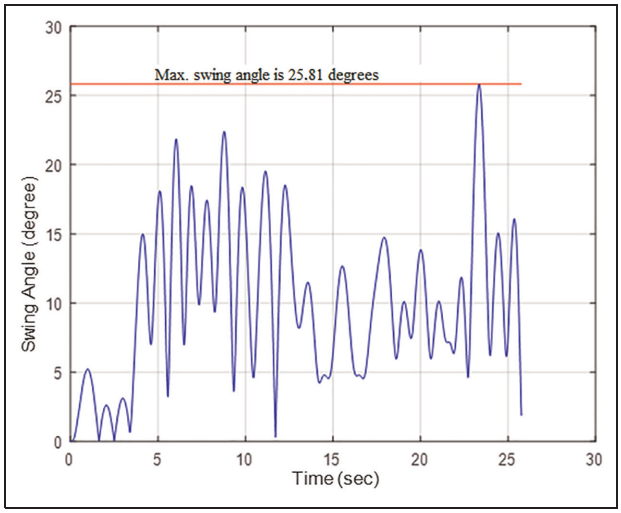

Figure 14. The swing angle of the load versus time (Scenario-2_2).

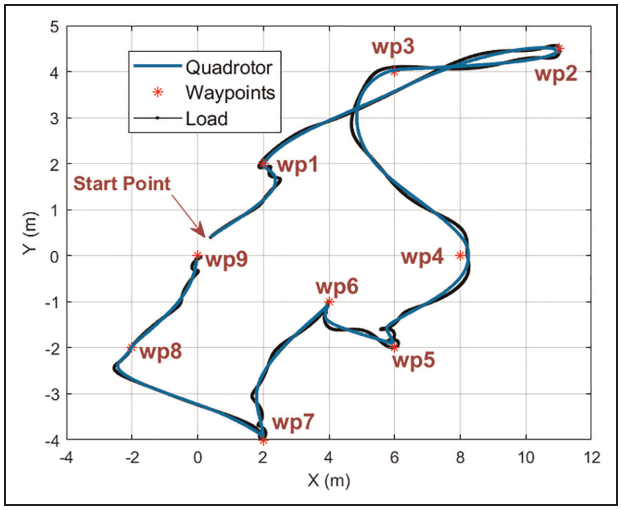

Figure 15. Top view of the path of quadrotor slung-load system (Scenario-3).

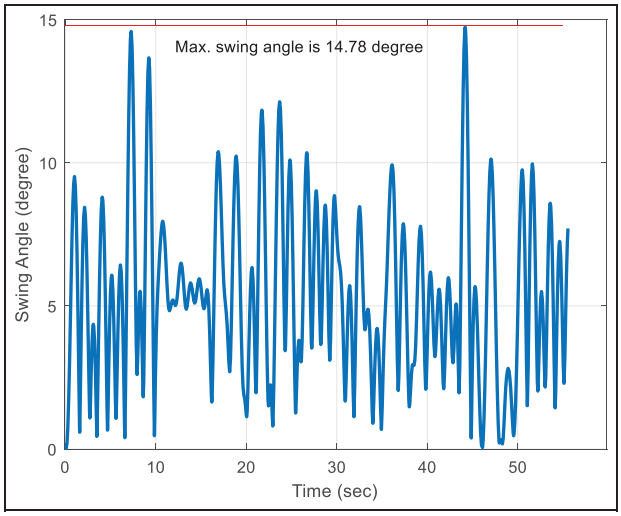

Figure 16. The swing angle of the load versus time (Scenario-3).

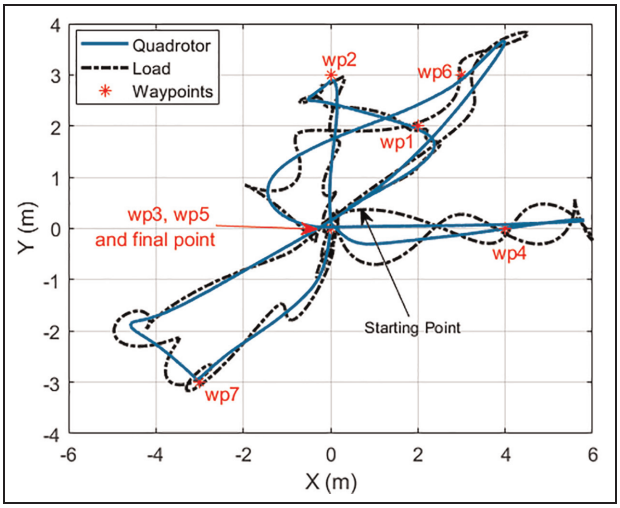

Figure 17. Resultant paths using controller structure without the complementary controller (Scenario-4).

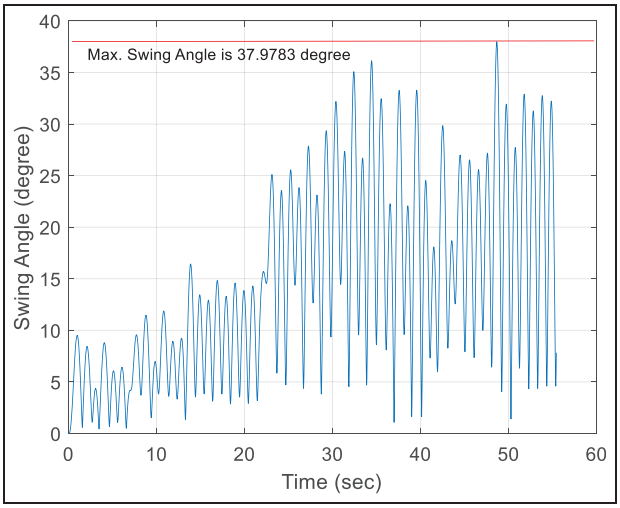

Figure 18. Swing Angles using controller structure without the complementary controller (Scenario-4).

The results are shown in Figure 19; the path with the same set of waypoints is given. In this case, the control structure given in Figure 4 is employed. The ANN part is activated automatically when the swing angle of the load is within 18°–22°. Otherwise, it does not generate any delta inputs. As shown in Figure 20, the constraint is violated around 40th seconds by 0.0077°.

Figures 21 to 23 give delta inputs for thrust, roll, and pitch moments. As can be seen from these figures, the ANN structure is activated when the swing angle is within the defined interval (i.e. between 18° and 22°). Also, delta inputs are saturated, as in the case of PSO. We now change the swing angle constraint from 20° to 17° to better observe the Complementary Controller's performance. Data is collected from the band $[\beta_{lower}\ \beta_{upper}] = [15\ 19]°$ to train the ANN. One can observe the swing angle graph in Figure 24. Note that the maximum value of the swing angle is 18.79°, which

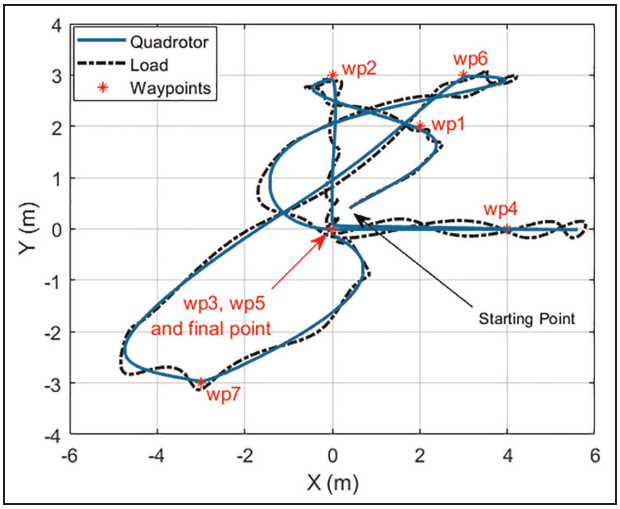

Figure 19. Resultant paths using controller structure with complementary controller (Scenario-4).

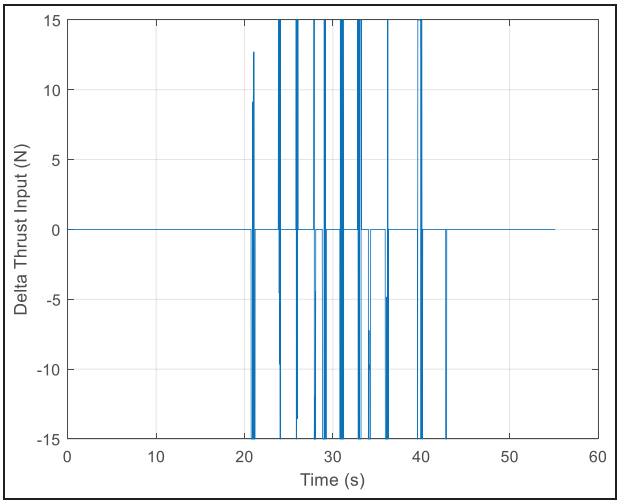

Figure 21. Thrust generated by complementary controller versus time (Scenario-4_1).

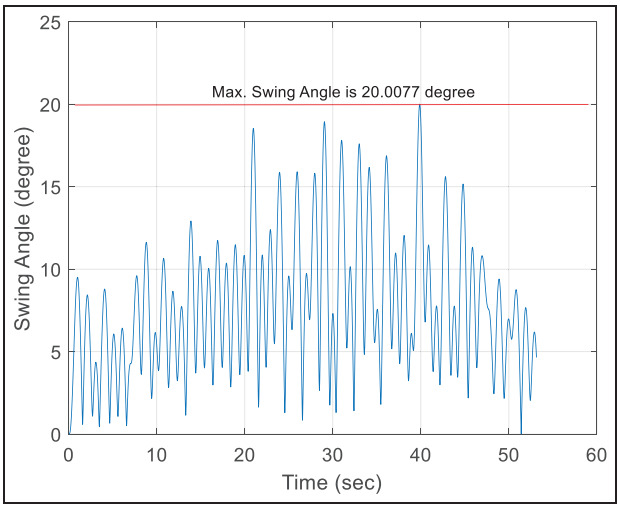

Figure 20. Swing Angles using controller structure with using the complementary controller (ANN has been trained for $[\beta_{lower}\ \beta_{upper}] = [18\ 22]^\circ$ degrees swing angle constraint) (Scenario-4_1).

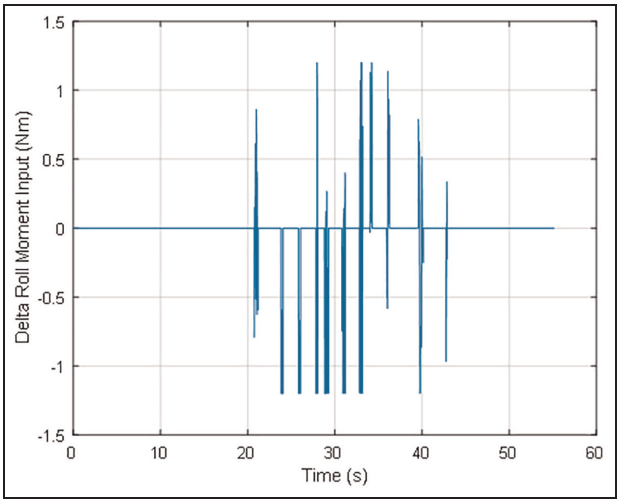

Figure 22. Roll Moment. It was generated by complementary controller versus time (Scenario-4_1).

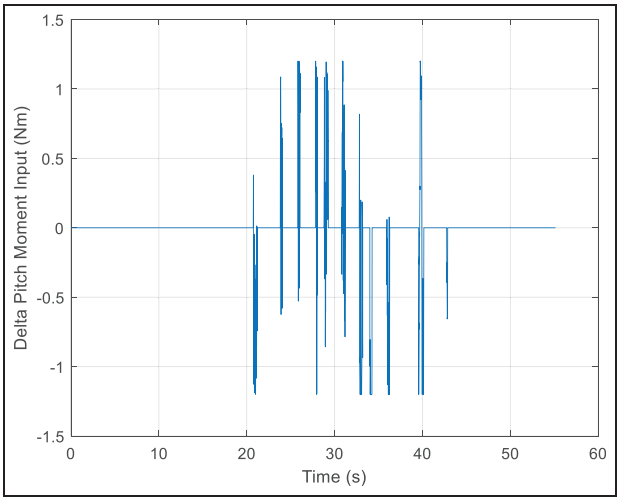

Figure 23. Pitch Moment. It was generated by complementary controller versus time (Scenario-4_1).

is slightly above the constraint for a short duration. You can compare this performance with the case when the complementary controller is trained for the band $[\beta_{lower}\ \beta_{upper}] = [28\ 32]^\circ$; naturally, the resultant controller satisfies the constraint better as the constraint relaxes (Figure 25).

The overall simulation structure, including the state flow diagram (yellow block), is given in Figure 26. In this model, the state flow diagram manages waypoint guidance.

A few of these simulation studies were realized physically in an experimental setup. We use a similar method in Li et.al.[56] to measure swing angles. Firstly, swing angle measurements of slung load were tested in a laboratory medium using the setup in Figure 27. The

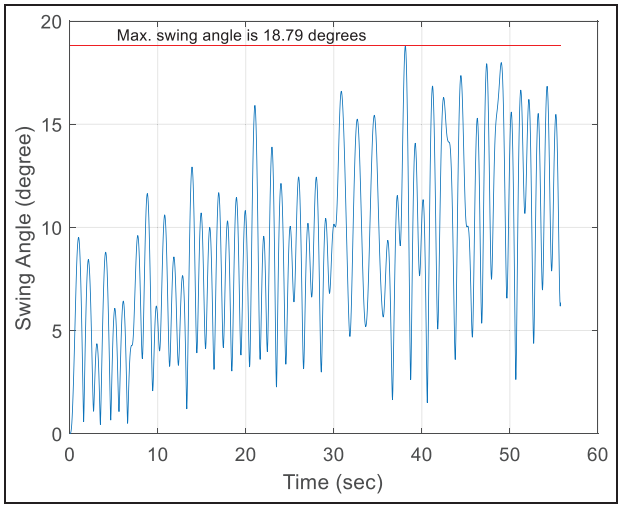

Figure 24. Swing Angles using controller structure with using the complementary controller (ANN has been trained for $[\beta_{lower}\ \beta_{upper}] = [15\ 19]^\circ$ degrees swing angle constraint) (Scenario-4_2).

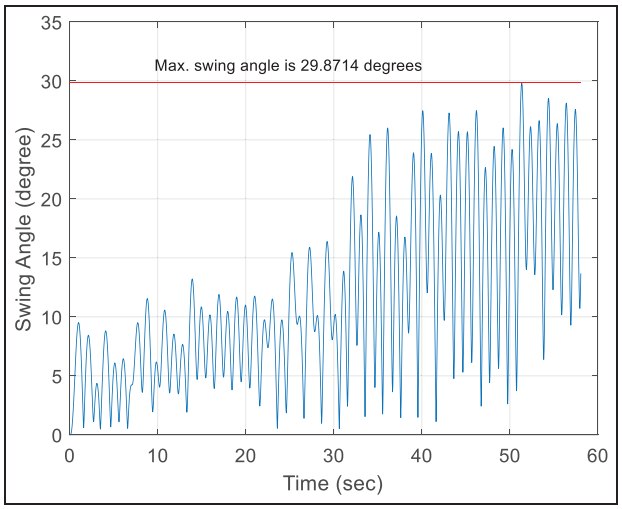

Figure 25. Swing Angles using controller structure with using the complementary controller (ANN has been trained for $[\beta_{lower}\ \beta_{upper}] = [28\ 32]^\circ$ degrees swing angle constraint) (Scenario-4_3).

measurement accuracy was verified using a Single Board Computer (SBC) and a camera at the top of the setup. Next, the angular displacements of the slung load measurements were obtained using the same (Inertial Measurement Unit) IMU in the quadrotor-slung load system. Later, this swing angle measurement setup is integrated into a quadrotor.

Conclusions

This study designed a controller for a quadrotor carrying a suspended load. The quadrotor is supposed to

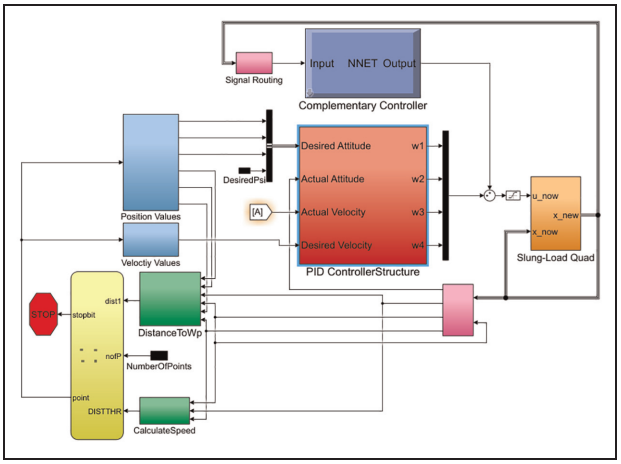

Figure 26. Overall Simulation structure constructed in MATLAB/Simulink.

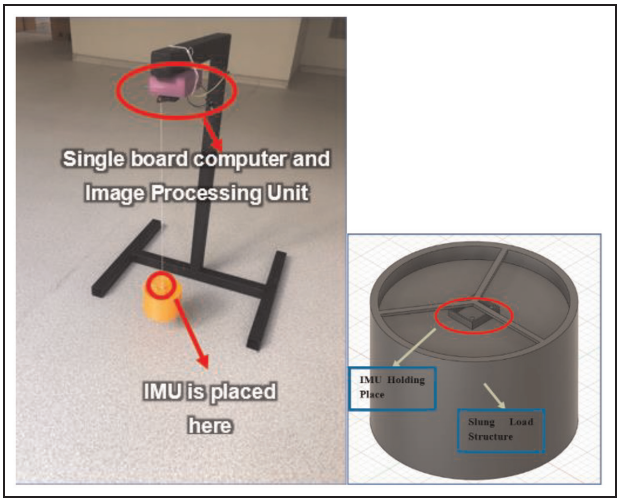

Figure 27. The swing angle measurement setup to collect IMU data.

reach its destination by passing through waypoints with speed constraints. This problem is solved successfully using the proposed complex control structure in Figure 4. In this structure, the primary controller adaptively changes the gains for speed and position as a function of their errors. The Complementary Controller is activated whenever the swing angle constraint is violated. The resultant structure performs much better than the one without the complementary structure. Simulations have been performed in MATLAB/Simulink environment. Measurements of the swing angle are first obtained in a laboratory setup, then tested on a quadrotor in the open field.

Studies on this problem will be extended to the slung load case with multiple quadrotors.

Declaration of conflicting interests

The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Funding

The author(s) received no financial support for the research, authorship, and/or publication of this article.

ORCID iD

Halit Ergezer https://orcid.org/0000-0002-7280-0416

References

1. Villa DKD, Brandão AS and Sarcinelli-Filho M. Load transportation using quadrotors: a survey of experimental results. In: *Proceedings of 2018 Int. Conf. on Unmanned Aircraft Systems (ICUAS), 2018 Jun*, pp.84–93. IEEE.
2. Elmokadem T. Distributed coverage control of quadrotor multi-UAV systems for precision agriculture. *IFAC-PapersOnLine* 2019; 52(30): 251–256.
3. Bernard M, Kondak K, Maza I, et al. Autonomous transportation and deployment with aerial robots for search and rescue missions. *J Field Robot* 2011; 28(6): 914–931.
4. Khamseh HB, Janabi-Sharifi F and Abdessameud A. Aerial manipulation—a literature survey. *Robot Auton Syst* 2018; 107: 221–235.
5. Ergezer H and Leblebicioglu K. Path planning for UAVs for maximum information collection. *IEEE Trans Aerosp Electron Syst* 2013; 49(1): 502–520.
6. Ergezer H and Leblebicioğlu K. 3D path planning for multiple UAVs for maximum information collection. *J Intell Robot Syst* 2014; 73: 737–762.
7. Ergezer H and Leblebicioğlu K. Online path planning for unmanned aerial vehicles to maximize instantaneous information. *Int J Adv Robot Syst* 2021; 18(3): 1–10.
8. Villa DKD, Brandão AS and Sarcinelli-Filho M. A survey on Load Transportation using multirotor UAVs. *J Intell Robot Syst* 2020; 98: 267–296.
9. Ruggiero F, Lippiello V and Ollero A. Aerial manipulation: A literature review. *IEEE Robot Autom Lett* 2018; 3(3): 1957–1964.
10. Mellinger D, Lindsey Q, Shomin M, et al. Design, modeling, estimation and control for aerial grasping and manipulation. In: *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2011, pp.2668–2673. New York: IEEE.
11. Heredia G, Jimenez-Cano AE, Sanchez I, et al. Control of a multirotor outdoor aerial manipulator. In: *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2014)*, 2014, pp.3417–3422. New York: *IEEE*.
12. Suarez A, Soria PR, Heredia G, et al. Anthropomorphic, compliant and lightweight dual arm system for aerial manipulation. In: *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2017, pp.992–997. New York: IEEE.
13. Lee SJ and Kim HJ. Autonomous swing-angle estimation for stable slung-load flight of multi-rotor UAVs. In: *Proceedings of the 2017 IEEE International Conference on Robotics and Automation (ICRA)*, 2017, pp.4576–4581. New York: IEEE.
14. Tang S, Sreenath K and Kumar V. Aggressive maneuvering of a quadrotor with a cable-suspended payload. In: *Proceedings of the Robotics: Science and Systems Workshop on Women in Robotics*, 2014.
15. Son CY, Seo H, Kim T, et al. Model predictive control of a multi-rotor with a suspended load for avoiding obstacles. In: *Proceedings of the 2018 IEEE International Conference on Robotics and Automation (ICRA)*, 2018, pp.1–6. New York: IEEE.
16. Xuan-Mung N, Nguyen NP, Nguyen T, et al. Quadcopter precision landing on moving targets via disturbance observer-based controller and Autonomous Landing Planner. *IEEE Access* 2022; 10: 83580–83590.
17. Xuan Mung N, Nguyen NP, Pham DB, et al. Synthesized landing strategy for quadcopter to land precisely on a vertically moving apron. *Mathematics* 2022; 10(8): 1328.
18. Goodarzi FA, Lee D and Lee T. Geometric Stabilization of a Quadrotor UAV with a Payload Connected by Flexible Cable. In: *Proceedings of the American Control Conference*, Portland, OR, USA, 2014, pp.4925–4930. IEEE.
19. Almeida MM and Raffo GV. Nonlinear Control of a TiltRotor UAV for Load Transportation. *IFAC-PapersOnLine* 2015; 48: 232–237.
20. Lee BY, Hong SM, Yoo DW, et al. Design of a neural network controller for a slung-load system lifted by 1 quadrotor. *J Autom Control Eng* 2015; 3: 9–14.
21. Lee BY, Lee HI, Yoo DW, et al. Study on Payload Stabilization Method with the Slung-load Transportation System Using a Quadrotor. In: *Proceedings of European Control Conference (ECC)*, Linz, Austria, 2015, pp. 2097–2102. IEEE.
22. Jain RPK. *Transportation of Cable Suspended Load Using Unmanned Aerial Vehicles: A Real-Time Model Predictive Control Approach*. Master's thesis, Delft University of Technology, 2015.
23. Klausen K, Fossen TI and Johansen TA. Nonlinear Control of a Multirotor UAV with Suspended Load. In: *Proceedings of the International Conference on Unmanned Aircraft Systems*, Denver, CO, USA, 2015, pp.176–184. IEEE.
24. Pizetta IHB, Brandao AS and Sarcinelli-Filho M. Control and Obstacle Avoidance for an UAV Carrying a Load in Forestal Environments. In: *Proceedings of the International Conference on Unmanned Aircraft Systems*, Dallas, TX, USA, 2018, pp.62–67. IEEE.
25. Barikbin B and Fakharian A. Trajectory tracking for quadrotor UAV transporting cable-suspended payload in Wind Presence. *Trans Inst Meas Contr* 2019; 41(5): 1243–1255.
26. Lee T. Geometric control of quadrotor UAVs transporting a cable-suspended rigid body. *IEEE Trans Control Syst Technol* 2018; 26(1): 255–264.
27. Geronel RS, Dowell EH and Bueno DD. Estimating the oscillation frequency of a payload mass on quadcopter in non-autonomous flight. *J Vib Control* 2023; 29: 236–250.
28. Sadr S, Moosavian SAA and Zarafshan P. Dynamics modeling and control of a quadrotor with swing load. *J Robot* 2014; 2014: 1–12.
29. Ergezer H. Multi-objective trajectory planning for slung-load quadrotor system. *IEEE Access* 2021; 9: 155003–155017.
30. Feng Y, Rabbath CA and Su CY. *Modeling of a micro UAV with slung payload*. Dordrecht: Springer Netherlands, 2015. pp.1257–1272.
31. Guerrero ME, Mercado DA, Lozano R, et al. Passivity based control for a quadrotor UAV transporting a cable-

suspended payload with minimum swing. In: *Proceedings of the 54th IEEE Conference on Decision and Control (CDC)*, 2015, pp.6718–6723. New York: IEEE.
32. Kusznir T and Smoczek J. Sliding mode-based control of a UAV quadrotor for suppressing the cable-suspended payload vibration. *J Contr Sci Eng* 2020; 2020: 1–12.
33. Yu G, Cabecinhas D, Cunha R, et al. Nonlinear backstepping control of a quadrotor-slung load system. *IEEE/ASME Trans Mechatron* 2019; 24(5): 2304–2315.
34. Dai S, Lee T and Bernstein DS. Adaptive Control of a Quadrotor UAV Transporting a Cable-Suspended load with Unknown Mass. In: *Conference on Decision and Control*, 2014. pp.15–17. Los Angeles: IEEE.
35. Shi D, Wu Z and Chou W. Harmonic extended state observer based anti-swing attitude control for quadrotor with slung load. *Electronics* 2018; 7(6): 83.
36. Lee J-W, Xuan-Mung N, Nguyen NP, et al. Adaptive altitude flight control of quadcopter under ground effect and time-varying load: theory and experiments. *J Vib Control* 2023; 29(3-4): 571–581.
37. Kashyap AK and Parhi DR. Particle swarm optimization aided PID gait controller design for a humanoid robot. *ISA Trans* 2021; 114: 306–330.
38. Ateş A, Coşan O, Değirmenci B, et al. Controller Design for Quadrotor-Slung Load System with Swing Angle Constraints Using Particle Swarm Optimization. In: *Paper presented at: International Conference on INnovations in Intelligent SysTems and Applications (INISTA)*, , Kocaeli, Turkey, 2021 Aug 25-27.
39. Mughees A and Mohsin SA. Design and control of magnetic levitation system by Optimizing Fractional Order PID controller using ant colony optimization algorithm. *IEEE Access* 2020; 8: 116704–116723.
40. Lee CL and Peng CC. Analytic Time Domain specifications PID controller design for a class of 2nd Order Linear Systems: A genetic algorithm method. *IEEE Access* 2021; 9: 99266–99275.
41. Lastomo D, Setiadi H and Djalal MR. Optimization pitch angle controller of rocket system using improved differential evolution algorithm. *Int J Adv Intell Inform* 2017; 3(1): 27–34.
42. El-Ghandour HA and Elbeltagi E. Comparison of five evolutionary algorithms for optimization of water distribution networks. *J Comput Civ Eng* 2018; 32(1): 10.
43. Wen G, Ge SS, Chen CLP, et al. Adaptive tracking control of surface vessel using optimized backstepping technique. *IEEE Trans Cybern* 2019; 49(9): 3420–3431.
44. Tu Vu V, Pham TL and Dao PN. Disturbance observer-based adaptive reinforcement learning for perturbed uncertain surface vessels. *ISA Trans* 2022; 130: 277–292. ISSN 0019-0578.
45. Polycarpou MM. Stable adaptive neural control scheme for nonlinear systems. *IEEE Trans Automat Contr* 1996; 41(3): 447–451.
46. Palunko I, Fierro R and Cruz P. Trajectory generation for swing free maneuvers of a quadrotor with suspended payload: A dynamic programming approach. In: *Proceedings of the IEEE International Conference on Robotics and Automation*, 2012. pp.2691–2697. IEEE.
47. Poli R, Kennedy J and Blackwell T. Particle swarm optimization. *Swarm Intell* 2007; 1: 33–57.
48. Rumelhart DE, Hinton GE and Williams RJ. Learning internal representations by error propagation. *Parallel Distrib Process* 1986; 1: 318–362.
49. Shi Y and Eberhart R. A modified particle swarm optimizer. In: *Proceedings of IEEE International Conference on Evolutionary Computation*, 1998, pp.69–73. Amsterdam: CWI
50. Guo B, Dian S and Zhao T. Robust NN-based decentralized optimal tracking control for interconnected nonlinear systems via adaptive dynamic programming. *Nonlinear Dyn* 2022; 110: 3429–3446.
51. Sreenath K, Michael N and Kumar V. Trajectory generation and control of a quadrotor with a cable-suspended load-a differentially flat hybrid system. In: *2013 IEEE International Conference on Robotics and Automation (ICRA), 2013*, pp.4888–4895. New York: IEEE.
52. Tang S and Kumar V. Mixed integer quadratic program trajectory generation for a quadrotor with a cable-suspended payload. In: *IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2015, pp.2216–2222. New York: IEEE.
53. Lee T. Computational geometric mechanics and control of rigid bodies [dissertation]. Ann Arbor (MI): University of Michigan, Dept. Aerosp. Eng.; 2008.
54. Kennedy J and Eberhart R. Particle swarm optimization. In: *Proceedings of IEEE International Conference on Neural Networks*. Vol. IV, 1995, pp.1942–1948. New York: IEEE.
55. Møller MF. A scaled conjugate gradient algorithm for fast supervised learning. *Neural Netw* 1993; 6(4): 525–533.
56. Li G, Tunchez A and Loianno G. PCMPC: Perception-Constrained Model Predictive Control for Quadrotors with Suspended Loads using a Single Camera and IMU. In: *2021 IEEE International Conference on Robotics and Automation (ICRA)*, 2021, pp.2012–2018. New York: *IEEE*.

Appendix

Notation

f	The lifting force.
$\boldsymbol{x_q}$	The position vector of the quadrotor's center of mass with respect to (wrt.) inertial frame.
$\boldsymbol{x_l}$	The position vector of load wrt. Inertial frame.
$\boldsymbol{x}$	State vector
M_ϕ	Roll moment acting on the quadrotor.
M_θ	Pitch moment acting on the quadrotor.
M_ψ	Yaw moment acting on the quadrotor.
ω_j	The angular velocity of rotor j.
K_f	Force-constant.
K_M	Moment-constant.
$\boldsymbol{R}$	The rotation matrix from Body Frame to Inertial Frame,
$\boldsymbol{\Omega}$	The angular velocity.
$\boldsymbol{I}$	The inertia matrix.
m_q	The mass of the quadrotor.
l	Length of Rope.
m_l	The mass of the load.
ϕ	The roll angle.
θ	The pitch angle of the quadrotor.
$\boldsymbol{p^T}$	The unit vector from the quadrotor to load in the inertial frame.
$\boldsymbol{u}$	The vector that defines a Maneuver.
$\boldsymbol{v}$	The input vector of the quadrotor-slung-load system.
T_j	The thrust generated by the jth motor.

(continued)