Trajectory Generation for Swing-Free Maneuvers of a Quadrotor with Suspended Payload: A Dynamic Programming Approach

Ivana Palunko, Rafael Fierro, and Patricio Cruz

Abstract—In this paper, we address the problem of agile swing-free trajectory tracking of a quadrotor with a suspended load. This problem has great practical significance in many UAV applications. However, it has received little attention in the literature so far. Flying with a suspended load can be a very challenging and sometimes hazardous task as the suspended load significantly alters the flight characteristics of the quadrotor. In order to deal with this problem, we propose a technique based on dynamic programming which ensures swing-free trajectory tracking. We start by presenting the mathematical model of a quadrotor with suspended load dynamics and kinematics. A high-level planner is used to provide desired waypoints, and then a dynamic programming approach is used to generate the swing-free trajectory for the quadrotor carrying a suspended load. Effectiveness of this method is demonstrated by numerical simulations and experiments.

I. INTRODUCTION

Unmanned aerial vehicles are increasingly being considered as means of performing complex functions or assisting humans in carrying out dangerous missions within dynamic environments. Possible applications include search and rescue, disaster relief operations, environmental monitoring, wireless surveillance networks, and cooperative manipulation. Dealing with these types of autonomous vehicles places severe demands on the design of control schemes that can adapt to different scenarios and possible changes of vehicle dynamics.

Quadrotor helicopters have become increasingly popular as unmanned aerial vehicle (UAV) research platforms. Many research groups have begun constructing quadrotor UAVs as robotics research tools [1], [2], [3], and several groups are developing quadrotors as general-use UAVs [4] that are becoming more popular in research labs around the world. These aerial vehicles are being used in a wide spectrum of indoor [5], [6] and outdoor [7], [8] applications. Because of their significant application potential, much research has been dedicated to quadrotor modeling and control. Some classical papers in the area are [9], [3].

In this paper we address the problem of quadrotor flying with a suspended load which is widely used for different kinds of cargo transport. Suspended load is also known as either slung load or sling load. Flying with a suspended load can be a very challenging and sometimes hazardous task because the suspended load significantly alters the flight characteristics of the quadrotor. Recent work can be found in [10], [11], [12], [13], [14]. The technique chosen to solve the problem of swing-free trajectory tracking is based on dynamic programming. The proposed technique is an open-loop method used to generate the optimal trajectory which will enable the swing-free flight of the quadrotor carrying a suspended load. A high-level planner is used to provide desired waypoints, and then the optimal swing-free trajectory is generated.

The paper is organized as follows. Section II presents the model of a suspended load. In Section III we describe the DP algorithm used to generate the optimal trajectory for a swing-free maneuver. Section IV provides simulation results and Section V gives experimental results. Finally, we draw conclusions in Section VI.

II. EQUATIONS OF MOTION OF QUADROTOR WITH A SUSPENDED LOAD

The suspended load system shown in Fig. 1 is considered to be a system consisting of two rigid bodies connected by massless straight-line links which support only forces along the link. The system is characterized by the mass and inertia parameters of the rigid bodies, and the suspension’s attachment point locations. The following assumptions are made when representing the suspended load system:

- both bodies are assumed to be rigid - this assumption excludes elastic quadrotor and rotor modes such as flapping etc., and non-rigid loads such as half-filled tanks of liquid, and long, very flexible loads etc.,
- the mass of the cable and aerodynamic effects on the load are neglected,
• the cable is considered to be inelastic.

These assumptions are considered to be sufficient for realistic representation of the quadrotor with a suspended load system used for non-aggressive trajectory tracking.

A. Quadrotor Modeling and Attitude Controller Design

The quadrotor dynamics are derived from first-principles to describe a six degrees of freedom (6 DOF) rigid body model, driven by forces and moments. To this end, we define two coordinate frames as depicted in Fig. 1. The moving coordinate frame \( \{A\} \) is fixed to the quadrotor and is called the aircraft-fixed reference frame. The origin of the aircraft-fixed frame is chosen to coincide with the Center of Gravity (CoG) when the CoG is in the principal plane of symmetry. The motion of the aircraft-fixed frame is described relative to an inertial reference frame. A ground-fixed reference frame \( \{G\} \) is considered to be the inertial frame. The position and orientation of the vehicle are described relative to the inertial reference frame \( \{G\} \) while the linear and angular velocities of the vehicle are expressed in the aircraft-fixed coordinate system \( \{A\} \). The following variables are used to describe quadrotor kinematics and dynamics,

\[
\eta_1 = \begin{bmatrix} x & y & z \end{bmatrix}^T \quad \text{position of the origin of } \{A\} \text{ measured in } \{G\},
\eta_2 = \begin{bmatrix} \phi & \theta & \psi \end{bmatrix}^T \quad \text{angles of roll } (\phi), \text{ pitch } (\theta) \text{ and yaw } (\psi) \text{ that parameterize locally the orientation of } \{A\} \text{ relative to } \{G\},
\nu_1 = \begin{bmatrix} u & v & w \end{bmatrix}^T \quad \text{linear velocity of the origin of } \{A\} \text{ expressed in } \{A\} \text{ (i.e., body-fixed linear velocity)},
\nu_2 = \begin{bmatrix} p & q & r \end{bmatrix}^T \quad \text{angular velocity of } \{A\} \text{ relative to } \{G\} \text{ expressed in } \{A\} \text{ (i.e., body-fixed angular velocity)}.
\]

The quadrotor’s 6 DOF nonlinear dynamic equations of motion can be expressed in a compact form as:

\[
M \dot{\nu} + C(\nu) \nu + D \nu + G(\eta) = \tau + \tau_L, \tag{1}
\]

where

\[
\eta = \begin{bmatrix} \eta_1 & \eta_2 \end{bmatrix}^T \quad \text{the vector of position and orientation},
\nu = \begin{bmatrix} \nu_1 & \nu_2 \end{bmatrix}^T \quad \text{vector of linear and angular velocities},
\text{and } M \text{ is the mass and inertia matrix of the quadrotor. Matrix } C(\nu) \text{ consists of Coriolis and centripetal terms. Using results from [15], we achieve a parameterization such that } C(\nu) \text{ is skew-symmetric. Decomposing the vectors of external forces we obtain three distinct vectors } D \nu, G(\eta), \tau \text{ and } \tau_L. \text{ Dissipative force and torque vector is given by } D \nu, \text{ where } D \text{ is the damping matrix. With } G(\eta) \text{ we denote the vector of gravitational forces and moments. Control inputs are given as vector } \tau \quad \tau(\eta_2, U) = \begin{bmatrix} f_\tau(\eta_2) \\ U_2 \\ U_3 \\ U_4 \end{bmatrix}, \tag{2}
\]

where

\[
U_1, U_2, U_3, U_4 \quad \text{represent control forces generated by four quadrotor rotors ([9])}. \quad \tau_L = [F_H \quad T_H] \quad \text{represents the vector of forces and torques (8) that the load exerts on the quadrotor. A compact representation of the system’s kinematics is}
\]

\[
\begin{bmatrix} \eta_1 \\ \eta_2 \end{bmatrix} = \begin{bmatrix} G_A(\eta_2) \\ 0 \end{bmatrix} \begin{bmatrix} 0 \\ \tau \end{bmatrix}, \tag{3}
\]

where

\[
G_A(\eta_2) \quad \text{and } Q(\eta_2) \quad \text{represent the transformation matrices. Full derivation of the presented model can be found in more detail in [16]. For attitude control of the quadrotor, we designed two low-level controllers: (i) linear cascade PD controllers; and (ii) a nonlinear controller based on input-output feedback linearization described in [16]. The proposed control algorithms are implemented in Matlab and simulation results show their performance are given in Figures 2(a), 2(b) and 2(c).}

B. Suspended Load Model

Very thorough models of single- and multiple-point suspension, as well as multi-lift variations of external suspended load systems can be found in [17]. In more recent publications [18], [19] and [20] the single point suspension type models found in [17] have been implemented and simulated. Considering the models presented in the literature, we outline the model of the single point suspended load, in this subsection.

The external slung load is modeled as a point mass spherical pendulum suspended from a single point. The coordinate systems we use are shown in Figure 1. The unit vectors of the \( \{H\} \) coordinate system always remain parallel to those of the aircraft-fixed coordinate system \( \{A\} \). The motion of the load is described in polar coordinates using angles \( \phi_L \) and \( \theta_L \), where \( \phi_L \) and \( \theta_L \) are measured from the \( Z_H \) axis in direction of \( X_H \) and \( Y_H \), respectively. Therefore, the position vector \( \rho_L \) of the load with respect to the suspension point is given by

\[
\rho_L = R_{Y_H}(\theta_L) R_{X_H}(\phi_L) \begin{bmatrix} 0 \\ 0 \\ l \end{bmatrix}, \tag{4}
\]

where

\[
R_{Y_H}(\theta_L) \quad \text{and } R_{X_H}(\phi_L) \quad \text{are rotational matrices, and } l \text{ is the length of the cable. The position vector } \rho_H \text{ of } \{H\} \text{ with respect to the quadrotor CoG is given by } \rho_H = [x_H \ y_H \ z_H]^T. \text{ The absolute velocity } \nu_L \text{ of the load is given by}
\]

\[
\nu_L = \nu_1 + \dot{\rho} + \nu_2 \times \rho, \tag{5}
\]

where \( \nu_1 \) is the linear velocity of the quadrotor, \( \rho = \rho_L + \rho_H \) is the position vector of the load with respect to the CoG of the quadrotor, and \( \nu_2 \) is the angular velocity of the quadrotor. The absolute acceleration \( \dot{\nu}_L \) of the load becomes

\[
\dot{\nu}_L = \nu_1 + \dot{\rho} + \dot{\nu}_2 \times \rho + 2\nu_2 \times \dot{\rho} + \nu_2 \times (\nu_2 \times \rho), \tag{6}
\]

where \( \dot{\nu}_1 \) is the linear acceleration of the quadrotor. The vector given by \( G_L(\eta) \) represents the vector of gravitational forces and moments

\[
G_L = R_{X_A}(\phi)^{-1} R_{Y_A}(\theta)^{-1} \begin{bmatrix} 0 \\ 0 \\ -m_L g \end{bmatrix}, \tag{7}
\]
where \( \phi \) and \( \theta \) are respectively the roll and pitch angles of the quadrotor, and \( m_L \) is the mass of the load. By enforcing the torque equilibrium about the suspension point, we get

\[
\begin{align*}
\mathbf{f}_\tau &= 0, \\
\mathbf{f}_\tau (\phi_L, \theta_L, \nu, \eta) &= -\mathbf{p}_L \times (-m_L \dot{\nu}_L + \mathbf{f}_{GL}),
\end{align*}
\]

where \( \nu \) and \( \eta \) are vectors of quadrotor states. In expanded form, (7) is a system of three second order equations in Cartesian coordinates in \( \{H\} \) frame. By solving these three equations for \( \dot{\phi}_L \) and \( \dot{\theta}_L \), we obtain the equations of motion for the given system in polar coordinates. The symbolic computation for obtaining these equations is performed using Mathematica, and because of the length of the equations, is omitted. The suspended load introduces additional terms denoted by \( \tau_L \) in the equations of motion of the quadrotor. The force \( \mathbf{F}_H \) that the load exerts on the vehicle and the torque \( \mathbf{T}_H \) are respectively given by

\[
\mathbf{F}_H = -m_L \mathbf{G}_L, \quad \mathbf{T}_H = \mathbf{\rho}_H \times \mathbf{F}_H. \tag{8}
\]

Both \( \mathbf{F}_H \) and \( \mathbf{T}_H \) are functions of \( \phi_L \) and \( \theta_L \), as well as of the quadrotor states \( \nu \) and \( \eta \).

III. TRAJECTORY GENERATION FOR SWING-FREE MANEUVERS

Transport of suspended objects using a robot or a crane is a common application. At the end of a transport motion, the suspended object naturally continues to swing. Suppression of residual oscillation has been a topic of research for many years. Although both open- and closed-loop strategies have been explored, in this paper we focus on an open-loop technique presented in [21] and [22] which we apply for a quadrotor carrying a suspended load.

A. Optimal Trajectory Generation Using Dynamic Programming

Dynamic programming is founded on the principle of optimality [23]. An optimal sequence of decisions has the property that whatever the initial state is, the remaining decisions must be optimal for the remaining problem [24].

With slight abuse of notation the subsequent procedure follows the approach of [21] and [25] and outlines the method of applying dynamic programming to a discrete-time piecewise linear system. We define the general form of the discrete-time system

\[
\mathbf{q}_{k+1} = \mathbf{A}_k \mathbf{q}_k + \mathbf{B}_k \mathbf{u}_k. \tag{9}
\]

The system and input matrices \( \mathbf{A}_k \) and \( \mathbf{B}_k \) can be time varying. Given an initial state \( \mathbf{q}_0 \), we need to find the optimal sequence of inputs that will minimize the scalar objective function:

\[
\Gamma (\mathbf{q}, \mathbf{u}) = \sum_{k=1}^{N} \Gamma_k (\mathbf{q}_k, \mathbf{u}_k). \tag{10}
\]

This objective function can be structured as quadratic in the input \( \mathbf{u}_k \) and state \( \mathbf{q}_k \), as shown in (11), where \( \gamma_k, y_k, z_k, Q_k, R_k, \) and \( S_k \) are time-varying. Given an initial state \( \mathbf{q}_0 \), we need to find the optimal sequence of inputs that will minimize the scalar objective function:

\[
\begin{align*}
\Gamma_k &= \gamma_k + \mathbf{q}_T \mathbf{y}_k + \frac{1}{2} \mathbf{q}_T \mathbf{Q}_k \mathbf{q} + \\
&= 2 \mathbf{q}_T \mathbf{R}_k \mathbf{u}_k + \mathbf{u}_T \mathbf{S}_k \mathbf{u}_k. \tag{11}
\end{align*}
\]

The basis of dynamic programming is the optimal value function. Beginning from any point \( i \), the optimal sequence can be computed recursively backwards to \( i = 1 \). The form of the optimal value function is thereby

\[
\Lambda_i = \min_{u_i} \sum_{k=i}^{N} \Gamma_k (\mathbf{q}_k, \mathbf{u}_k). \tag{12}
\]

Since the objective function is quadratic in form, the optimal value function will also be defined as a quadratic, where \( \zeta_i, \nu_i, \) and \( \mathbf{W}_i \) are the coefficients

\[
\Lambda_i (\mathbf{q}_i) = \zeta_i + \mathbf{q}_T \mathbf{\nu}_i + \frac{1}{2} \mathbf{q}_T \mathbf{W}_i \mathbf{q}. \tag{13}
\]

Using Bellman’s principle of optimality, the backward recursive relation can now be formed

\[
\Lambda_i = \min_{u_i} (\Gamma_i + \Lambda_{i+1}). \tag{14}
\]

Substituting (11) and (12) into (13) yields the recursive relation

\[
\begin{align*}
\zeta_i + \mathbf{q}_T \mathbf{\nu}_i + \frac{1}{2} \mathbf{q}_T \mathbf{W}_i \mathbf{q}_i &= \min_{u_i} \{ \zeta_{i+1} + \mathbf{q}_T \mathbf{\nu}_{i+1} + \\
&= \frac{1}{2} \mathbf{q}_{i+1} \mathbf{W}_{i+1} \mathbf{q}_{i+1} + \gamma_i + \mathbf{q}_T \mathbf{y}_i + \mathbf{u}_T \mathbf{z}_i + \\
&= \frac{1}{2} [\mathbf{q}_T \mathbf{Q}_i \mathbf{q}_i + 2 \mathbf{q}_T \mathbf{R}_i \mathbf{u}_i + \mathbf{u}_T \mathbf{S}_i \mathbf{u}_i]. \tag{14}
\end{align*}
\]

Now, substituting the system equation (9) into (14) and simplifying it, we get

\[
\begin{align*}
\zeta_i + \mathbf{q}_T \mathbf{\nu}_i + \frac{1}{2} \mathbf{q}_T \mathbf{W}_i \mathbf{q}_i &= \min_{u_i} \{ \zeta_{i+1} + \gamma_i + \\
&= \mathbf{q}_T \mathbf{h}_4 + \mathbf{u}_T \mathbf{h}_5 + \frac{1}{2} \mathbf{q}_T \mathbf{H}_1 \mathbf{q}_i + \\
&= 2 \mathbf{q}_T \mathbf{H}_2 \mathbf{u}_i + \mathbf{u}_T \mathbf{H}_3 \mathbf{u}_i \}, \tag{15}
\end{align*}
\]

where

\[
\begin{align*}
\mathbf{H}_{1i} &= \mathbf{Q}_i + \mathbf{A}_i \mathbf{W}_{i+1} \mathbf{A}_i, \\
\mathbf{H}_{2i} &= \mathbf{R}_i + \mathbf{A}_i \mathbf{W}_{i+1} \mathbf{B}_i, \\
\mathbf{H}_{3i} &= \mathbf{S}_i + \mathbf{B}_i \mathbf{W}_{i+1} \mathbf{B}_i, \\
\mathbf{h}_{4i} &= \mathbf{y}_i + \mathbf{A}_i \mathbf{v}_{i+1}, \\
\mathbf{h}_{5i} &= \mathbf{z}_i + \mathbf{B}_i \mathbf{v}_{i+1} + \mathbf{u}_T \mathbf{v}_{i+1}. \tag{16}
\end{align*}
\]

Differentiating the right-hand side of (15) with respect to \( \mathbf{u}_i \), then equating to zero results in

\[
\mathbf{u}_i = -\mathbf{H}_{3i}^{-1} \mathbf{H}_{3i} \mathbf{q}_i + \mathbf{h}_{5i}. \tag{17}
\]
Substituting this solution back into (15) and equating terms of like degree in $q_i$ results in the following recursive equations:

$$
\begin{align*}
\zeta_i &= \zeta_{i+1} + \gamma_i - \frac{1}{2} h_i^T H_i^{-1} h_i,
\nu_i &= h_i - H_i^{-1} h_i^T h_i,
W_i &= H_i - H_i^{-1} H_i^T h_i.
\end{align*}
$$

The initial values for (18) are given by

$$
\zeta_N = \gamma_N, \quad \nu_N = y_N, \quad W_N = Q_N.
$$

The procedure for applying this algorithm is as follows:

1) Determine $v_N$ and $W_N$ using (19). As will be shown, $y_N$ and $Q_N$ can be extracted directly from the objective function.
2) Calculate $v_i$ and $W_i$ recursively for $i = N - 1$ to $i = 1$ using (18), while storing matrices $H_3^{-1}$, $H_2^{-1}$, and $H_1^{-1}, h_i$ in the process.
3) Calculate $u_i$ and $q_i$ recursively for $i = 1$ to $i = N - 1$ using (17) and (9), respectively.

The simplest candidate for the objective function follows a minimum energy principle. Using trajectory accelerations as the input, rather than jerk, we improve computational efficiency while still allowing a zero end point constraint for the trajectory velocities. The state and input of the system are consequently given by

$$
q = [\eta_L, \nu_L, \eta, \nu]^T, \quad u = \dot{\nu},
$$

where $\eta_L$ and $\nu_L$ are load displacement angles and angular velocities, and $\eta, \nu$ and $\dot{\nu}$ are quadrotor attitude, velocity and acceleration vectors. Since the quadrotor exhibits high performance trajectory tracking, as shown in Figure 2, the dynamic model of the quadrotor in the dynamic programming algorithm is replaced by the corresponding kinematic model (double-integrator) in order to improve computational efficiency of the algorithm. To suppress the load oscillations, a penalty weight must be introduced into the objective function. The resulting objective function is given by

$$
\Gamma_k = \frac{1}{2} q_k^T Q_k q_k + 2 q_k^T R_k u_k + u_k^T S_k u_k +
\frac{1}{2} p [x_F - x_N]^T [x_F - x_N].
$$

Matching like terms leads to

$$
\zeta_N = \frac{1}{2} px_F^T x_F, \quad y_N = -px_F, \quad z_N = 0, \quad Q_N = Q_k = q_k I_n,
\begin{align*}
R_N &= R_k, \quad S_N = S_k = I_n,
\end{align*}
$$

where $I_n \in \mathbb{R}^{n \times n}$ is an identity matrix and $n$ is the number of state variables. In [22] and [21], the value of the penalty weight $p$ is found by trial and error, and $q_k$ and $R_k$ are set to zero. The weighted terms are chosen to represent the sum of squares of the final state error. Instead of merely guessing the penalty weight $p$, in this paper we obtain optimal parameters for $p$, $q_k$, and $R_k$ by minimizing the rate of convergence of the dynamic programming algorithm and the duration of the trajectory while still keeping the objective function (11). The computation is performed using the Matlab Optimization Toolbox. The algorithm presented here can be considered as the first step towards developing an intelligent, online learning algorithm. The optimal weighted vectors are found to be

$$
\begin{align*}
p &= [99.69, 99.69, 101.68, 101.68, I_{1 \times 12}],
q_f &= [-9e^{-4}, 2e^{-3}, 9e^{-4}, 2e^{-3}, I_{1 \times 12}],
\end{align*}
$$

$$
\begin{align*}
R_k &= [R_{k42}, O_{1 \times 12}, O_{12 \times 4}],
R_{k42} &= [0, 0.3, 0, 0, 0, 0.3, 0],
\end{align*}
$$

where $p \in \mathbb{R}^{1 \times 16}$, $q_f \in \mathbb{R}^{1 \times 16}$ and $R_k \in \mathbb{R}^{16 \times 6}$.

IV. SIMULATION RESULTS

The algorithm described in the previous section requires the initial and terminal states to be known prior to the optimization. An initial trajectory estimate is also required for the first optimization pass in order to compute the required $A_k$ and $B_k$ matrices. In this paper, we employ cubic polynomial position trajectories (see Fig. 2) for the initial simulation of the system used in the first optimization pass since it has continuous first and second derivatives. The cubic trajectories result in residual oscillations of approximately $6.1^\circ$ for swing (Fig. 3(a)), and $6.15^\circ$ for rock (Fig. 3(b)). Obtained optimal trajectories suppress the residual oscillations to less than 5% of the initial oscillation magnitudes. The algorithm requires three passes before reaching the convergence with computation time of 3.4 s. Furthermore, we present a set of simulations for employing swing-free trajectory tracking in urban environments. By using the swing-free policy presented above, where the displacement angle of the load is minimized, we obtain the optimal trajectory depicted in Figure 4. We can see that the quadrotor performs optimal trajectory tracking without collisions, while using the initial cubic trajectories the displacement angles are too large to avoid collisions. Figure 5 depicts the load displacement angles for both cubic and swing-free trajectories.

V. EXPERIMENTAL VERIFICATION

The experimental verification was performed at the Multi-Agent Robotics Hybrid and Embedded Systems (MARHES) laboratory at the University of New Mexico. We use the AscTec Hummingbird Flight-System as a quadrotor platform [4]. A Vicon motion capture system ([26]) allows to track the attitude of the quadrotor and the suspended load. As a real-time engine, we use the National Instruments CompactRIO that contains the reconfigurable I/O FPGA core, along with the LabVIEW real-time module to execute real-time applications. The NI CompactRIO is a reconfigurable embedded control and acquisition system, is programmed with LabVIEW graphical programming tools, and can be
used in a variety of embedded control and monitoring applications. The real-time embedded controller offers powerful stand-alone embedded execution for LabVIEW Real-Time applications. Using the real-time embedded controller, we are able to send the control values to the quadrotor at a frequency of 100 Hz. For wireless communication we use XBee-PRO ZB embedded RF modules [27] which provide cost-effective wireless connectivity to devices in ZigBee mesh networks. Two of these modules are used in this implementation. One is connected on-board of the quadrotor and the other one is connected by an RS-232 adapter to the Serial port of the CompactRIO. The position controllers for each axis (X, Y, and Z) are implemented as Lead-Lag compensators. The effectiveness of the methodology proposed herein is verified in the experimental setup illustrated in Fig. 6. First, we would like to show robustness of the proposed method with respect to unmodeled actuator dynamics, noise and system delays. In our case the actuator is a quadrotor whose model and attitude control design are presented in Sec. II-A. Since the quadrotor shows almost perfect trajectory tracking (Fig. 2(a), Fig. 2(b) and Fig. 2(c)) in the DP algorithm (Sec. III-A), we assume perfect tracking, i.e., the full nonlinear quadrotor model is replaced by a double integrator. The performance of the swing-free trajectory tracking in an ideal case is shown in Fig. 7(a). However, on Fig. 2(c) we can see that $\ddot{e} \neq 0$ where $\ddot{e}$ represents the acceleration tracking error of a quadrotor. Therefore, in the case when we use the full nonlinear model with an attitude controller for swing-free trajectory tracking, we can see that the performance is different than in the ideal case (Fig. 7(b)) but it is still satisfying. By implementing the proposed method on an experimental system (Fig. 7(c)), we see that the performance deteriorates due to imperfect tracking. On the other hand, the attenuation of the load displacement angles $\phi_L$ and $\theta_L$ is achieved as shown in Fig. 8(a) and Fig. 8(b). Therefore, the proposed method is robust enough and shows good performance even with the lack of perfect trajectory tracking.

VI. CONCLUSIONS

In this paper we propose a technique based on dynamic programming which ensures swing-free trajectory tracking of a quadrotor with a suspended load. The proposed approach is an important step towards developing the next generation of autonomous aerial vehicles. This method enables the quadrotor to perform agile flight maneuvers while performing swing-free trajectory tracking. Future work includes the implementation of the proposed algorithm on a multi-vehicle test bed available at the MARHES Lab. Furthermore, we will investigate robustness of the proposed technique considering the effects of an unbalanced quadrotor, i.e., a displaced center of gravity.

VII. ACKNOWLEDGMENTS

This work is supported by NSF grant ECCS #1027775, and by DOE URPR (University Research Program in Robotics) grant DE-FG52-04NA25590.

REFERENCES

Pass # 0  \( \theta_{L_{max}} = 6.15^\circ \)

Pass # 1  \( \theta_{L_{max}} = 1.81^\circ \)

Pass # 2  \( \theta_{L_{max}} = 0.319^\circ \)

(a) X trajectory and swing.

Pass # 0  \( \phi_{L_{max}} = 6.1^\circ \)

Pass # 1  \( \phi_{L_{max}} = 1.8^\circ \)

Pass # 2  \( \phi_{L_{max}} = 0.315^\circ \)

(b) Y trajectory and rock.

Fig. 3. Optimal and cubic trajectories with load displacement considering one waypoint.

Fig. 4. 3D representation of trajectories considering multiple waypoints with obstacles.

Fig. 5. Load displacement angles considering multiple waypoints with obstacles - the swing-free policy.

Fig. 6. Experimental results: Swing-free trajectory tracking of a quadrotor with suspended load.


Fig. 7. Robustness of the proposed method considering the unmodeled dynamics of the quadrotor.

Fig. 8. Experimental result for swing-free trajectory tracking.