2022 The 22nd International Conference on Control, Automation and Systems (ICCAS 2022)
BEXCO, Busan, Korea, Nov. 27~Dec. 01, 2022

Real-Time Trajectory Generation of a Quadrotor UAV with Load Suspended from a Pulley

Donggeon David Oh[1], Jeonghyun Byun[1], and Dongjae Lee[1]*

[1]Department of Aerospace Engineering, Seoul National University,
Seoul, 08826, Korea (donggeonoh1999@snu.ac.kr, quswjdgus97@snu.ac.kr, ehdwo713@snu.ac.kr) * Corresponding author

Abstract: Quadrotor UAV carrying a payload with a cable can be a prominent setup for aerial transportation. However, it is challenging for the suspended load system to navigate through tunnel-like obstacles if the vertical gap is narrower than the length of the cable. We aim to resolve this problem by adopting the pulley mechanism which controls the cable length. This paper presents a model predictive control based real-time trajectory generation algorithm for a quadrotor with its load suspended from a pulley, or the quadrotor-pulley system. We design a cost function that can be generally applied for avoiding obstacles with rectangular surfaces. Using the cost function, an optimal control problem is formulated, which is then solved by implementing model predictive control algorithm with sequential quadratic programming solver. We employ the state-of-the-art real-time iteration scheme to promote computational efficiency of the algorithm. The presented algorithm is demonstrated with simulations, where trajectories for the quadrotor-pulley system were successfully generated. Collision-free trajectories through tunnels with various heights and lengths were generated in real-time, thus validating the robustness of the algorithm.

Keywords: model predictive control (MPC), trajectory generation, path planning, optimal control, suspended load

1. INTRODUCTION

1.1 Related Works

Connecting the quadrotor UAV and its load with a cable is a well-known method for aerial transportation. Such system, which is also known as the "suspended load" system, retains the quadrotor's agile behavior in terms of attitude control since the moments of inertia are unchanged. Furthermore, the maximum load transportation capability could be utilized because no heavy structures need to be added to the quadrotor.

Geometric controller has been proposed for the suspended load system. In [1], the authors have developed a dynamical model of the differentially flat hybrid system and designed a geometric controller. The robust tracking performance of the controller was verified in [2, 3].

Trajectory generation algorithms for the suspended load system have also been studied by many researchers. Model predictive control (MPC) was shown to be an effective methodology for real-time obstacle avoidance in [4], where its performance was verified with experiments. In [5,6], waypoint tracking performance of linear-quadratic regulator and MPC were compared. The authors of [7] employed mixed integer quadratic programming to generate trajectories through narrow gaps. Direct collocation was performed on the suspended load system, and a method of reformulating the obstacle avoidance constraints was presented in [8]. A novel dynamical model was introduced in [9], where the trajectory optimization problem was posed as a mathematical program with complementarity constraints. In [10], dynamical model of the suspended load system was simplified by considering the control delay, and a real-time optimal trajectory generation algorithm was proposed.

Still, the previous works did not address tunnel-like structures that constrain the position of the load and the quadrotor in the vertical axis. In [7, 10], the quadrotor "swings" its load to a large angle as it moves through a window. For the system to pass through a tunnel, the cable should remain slanted inside the tunnel, thus shortening the vertical distance between the load and the quadrotor. However, we consider such trajectories to be infeasible, since it requires hard acceleration throughout a long time period. The authors of [8] have simulated collision-free trajectories through a tunnel by designating a waypoint at which the load angle must be larger than $\pi/4$, and by exploiting the hybrid modes of the system. Such aggressive maneuvers are unlikely to be reproduced in experiments where physical limitations on the quadrotor should be considered. Moreover, the methods are not applicable for dynamic environments, since waypoint designation and specific parameter settings that depend on obstacle configurations are required.

To overcome the difficulty of navigating through tunnel-like obstacles, we adopt the pulley mechanism proposed in [11]. Although the authors of [11] have successfully designed a geometric controller for a quadrotor with its load suspended from a pulley (hereinafter referred to as the "quadrotor-pulley system"), methods for the system's trajectory generation are yet to be proposed to the best of our knowledge. In this paper, we have employed MPC to generate the quadrotor-pulley system's trajectory, and together with the cost function for tunnel-like obstacle avoidance, the presented algorithm has successfully generated collision-free trajectories through the obstacles in real-time.

1.2 Contribution

The main contributions of this paper are summarized as follows:

1. A real-time trajectory generation algorithm for the quadrotor-pulley system is proposed for the first time, to the best of the authors' knowledge.
2. A cost function which can be generally applied for

978-89-93215-25-0/22/$31.00 ©ICROS

avoiding obstacles with rectangular surfaces is proposed. 3. The presented algorithm was validated with simulations in which the system was able to successfully navigate through narrow tunnels.

2. DYNAMICAL MODEL

The dynamical model of the quadrotor-pulley system was derived in [11]. An additional actuator connected to the pulley alters the cable length, thus allowing the suspended load system to navigate through tight passages without aggressive maneuvers. The system may be interpreted as a hybrid model with its dynamics switching when the tension in the cable becomes zero or when the slack cable becomes taut. In this study, we assume that the cable remains taut in all circumstances because the pulley mechanism is expected to minimize aggressive maneuvers when navigating through obstacles. Furthermore, the radius of the pulley is assumed to be much smaller than the length of the cable.

The coordinate system and variables of the quadrotor-pulley system are depicted in Fig. 1, and the nomenclature is presented in Table 1.

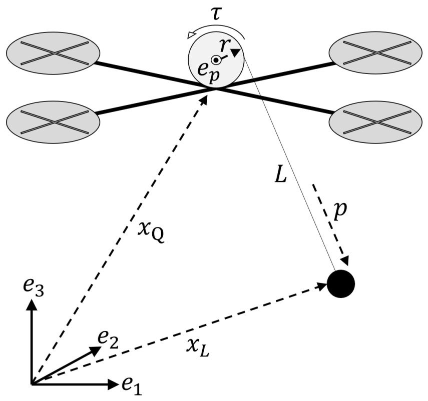


Fig. 1. Quadrotor-pulley system and its variables

The equations of motion for the system are written as

$$\dot{x_L} = v_L, \tag{1}$$

$$\dot{p} = \omega \times p, \tag{2}$$

$$\dot{R} = R\hat{\Omega}, \tag{3}$$

$$m_Q L\dot{\omega} = -p \times fRe_3 - 2m_Q\dot{L}\omega, \tag{4}$$

$$I_Q\dot{\Omega} = M - \tau e_p - \Omega \times I_Q\Omega, \tag{5}$$

$$D\begin{bmatrix} \dot{v_L} + ge_3 \\ \ddot{L} \end{bmatrix} + H = \begin{bmatrix} (p \cdot fRe_3)\, p \\ \tau \end{bmatrix}, \tag{6}$$

where matrix D and vector H are defined as

$$D = \begin{bmatrix} (m_Q + m_L)\, I_3 & -m_Q p \\ m_L r p^T & I_p/r \end{bmatrix},$$

$$H = \begin{bmatrix} m_Q L\,(\dot{p} \cdot \dot{p})\, p \\ 0 \end{bmatrix}.$$

We define the state vector and the input vector as $x(t) = \left[x_L^T, p^T, \Phi^T, v_L^T, \omega^T, \Omega^T, L, \dot{L}\right]^T \in \mathbb{R}^{20}$ and $u(t) = \left[f, M^T, \tau\right]^T \in \mathbb{R}^5$, respectively. Compared to the suspended load system without the pulley mechanism in [1], the length of the cable L and its time derivative $\dot{L}$ are added to the state vector, and magnitude of torque acting on the pulley τ is appended to the input vector.

Table 1. Nomenclature

$\mathcal{B}$	Body-fixed coordinate frame
$\mathcal{I}$	Inertial coordinate frame
$m_Q, m_L \in \mathbb{R}$	Mass of the quadrotor and the suspended load
$I_Q \in \mathbb{R}^{3\times3}$	Inertia matrix of the quadrotor
$I_p \in \mathbb{R}$	Moment of inertia of the pulley
$r \in \mathbb{R}$	Radius of the pulley
$R \in SO(3)$	Rotation matrix from $\mathcal{B}$ to $\mathcal{I}$
$\Phi \in \mathbb{R}^3$	Attitude of the quadrotor described with Euler angle conventions
$\Omega \in \mathbb{R}^3$	Angular velocity of the quadrotor expressed in $\mathcal{B}$
$\omega \in \mathbb{R}^3$	Angular velocity of the suspended load expressed in $\mathcal{I}$
$x_Q, x_L \in \mathbb{R}^3$	Position of the center of mass of the quadrotor and suspended load in $\mathcal{I}$
$v_Q, v_L \in \mathbb{R}^3$	Velocity of the center of mass of the quadrotor and suspended load in $\mathcal{I}$
$f \in \mathbb{R}$	Total thrust of the quadrotor
$M \in \mathbb{R}^3$	Moment input for the quadrotor expressed in $\mathcal{B}$
$\tau \in \mathbb{R}$	Torque input for the pulley
$p \in S^2$	Unit vector from the quadrotor to the suspended load expressed in $\mathcal{I}$
$L \in \mathbb{R}$	Length of the cable, which is altered by the pulley mechanism
$e_1, e_2, e_3 \in \mathbb{R}^3$	Unit vectors along the x, y, z directions of $\mathcal{I}$
$e_p \in \mathbb{R}^3$	Unit vector along the pulley's shaft expressed in $\mathcal{B}$

3. TRAJECTORY GENERATION

The objective of our research is to develop a real-time trajectory generation algorithm so that the quadrotor-pulley system could navigate through tunnel-like obstacles and reach the desired configuration. We employ MPC for its effectiveness in real-time trajectory planning as shown in [4–6, 10]. Formulation of a nonlinear programming problem (NLP) from an optimal control problem (OCP) is introduced in section 3.1. In section 3.2, we present the cost function which enables collision avoidance from obstacles with rectangular surfaces. The sequential quadratic programming (SQP) algorithm used to solve the NLP is discussed in section 3.3.

3.1 NLP Formulation

We apply direct multiple shooting to an OCP to formulate a NLP [12]. The method divides the time horizon into N shooting intervals $[t_0, t_1, ..., t_N]$. At each discrete time point t_k, the state vector is defined as $x_k = x(t_k)$ for $k = 0, 1, ..., N$, and the input vector is defined as $u_k = u(t_k)$ for $k = 0, 1, ..., N-1$. The NLP is described as

$$\min_{x_k, u_k} \sum_{k=0}^{N-1} \frac{1}{2} \left\| h_k(x_k, u_k) \right\|_W^2 + \frac{1}{2} \left\| h_N(x_N) \right\|_{W_N}^2 \tag{7}$$

$$s.t.\ \ x_0 = \hat{x}_0, \tag{8}$$

$$x_{k+1} = F(x_k, u_k), \quad k = 0, ..., N-1, \tag{9}$$

$$\underline{r}_k \leq r_k(x_k, u_k) \leq \overline{r}_k, \quad k = 0, ..., N-1, \tag{10}$$

$$\underline{r}_N \leq r_N(x_N) \leq \overline{r}_N. \tag{11}$$

Eq. (7) is the cost function of the optimization problem. From Eqs. (1) ~ (6), we obtain a continuous-time differential equation $\dot{x}(t) = f(x(t), u(t))$ representing the system dynamics, which is discretized into the form $x_{k+1} = F(x_k, u_k)$ with Runge-Kutta fourth-order method. The discretized dynamics of the system is considered as an equality constraint in Eq. (9). Eq. (8) enforces the state x_0 to be equal to the measurement of the current state $\hat{x}_0$, and Eqs. (10) ~ (11) represent the inequality constraints of the NLP.

3.2 Cost Function and Constraints

Our objective is divided into two subtasks:

1. The UAV should reach the desired configuration.
2. The UAV should avoid collision with the obstacles.

The first subtask is easily accomplished by setting the cost function J_1 as

$$J_1 = \sum_{k=0}^{N-1} \frac{1}{2} \left(\begin{bmatrix} x_k - x_{des} \\ u_k - u_{des} \end{bmatrix} \right)^T Q_1 \left(\begin{bmatrix} x_k - x_{des} \\ u_k - u_{des} \end{bmatrix} \right) + \frac{1}{2} \left(x_N - x_{des} \right)^T L \left(x_N - x_{des} \right), \tag{12}$$

where x_{des} is the desired configuration, and $u_{des} = \left[(m_Q + m_L) g, 0, 0, 0, -m_L r g \right]^T$ is the input that controls the quadrotor-pulley system to hover at a fixed configuration.

For the subtask of obstacle avoidance, we model a tunnel-like obstacle as two parallel two-dimensional plates of finite lengths and widths, as illustrated in Fig. 2. Although an inequality constraint could be formulated using state variables x_L, p, and L to bound the vertical position of the quadrotor-pulley system to be in between the two plates, such constraint should only be applied when the system is inside the tunnel. The constraint should be applied conditionally, depending on the position of the quadrotor-pulley system and the tunnel. Unfortunately, NLP solvers including SQP algorithms rarely support conditional constraints. Even though representing the obstacles as hard inequality constraints would ensure a safe flight without collision, its implementation overcomplicates the problem.

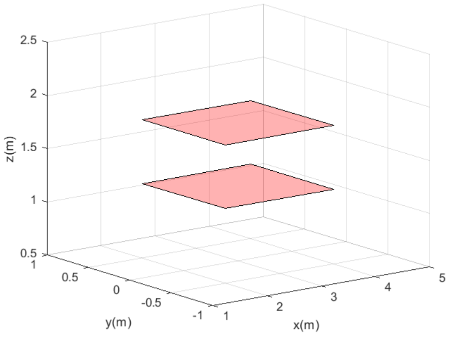


Fig. 2. A tunnel-like obstacle with $x_{o,1} = 2, x_{o,2} = 4, y_{o,1} = -0.5, y_{o,2} = 0.5, z_{o,1} = 1.2, z_{o,2} = 1.8$

An alternative methodology for obstacle avoidance is to use an additional cost function that discourages the system from getting dangerously close to the obstacles. The idea behind this methodology is to design a repulsive potential-like scalar field with its values exponentially increasing as the system approaches the obstacles. By adding this potential-like cost function to Eq. (12), the optimal trajectory of the quadrotor-pulley system is encouraged to keep a safe distance from the obstacles. The effectiveness of using a cost function for obstacle avoidance and thus interpreting the obstacles as soft constraints has been validated with experiments in [4].

We define a plate-like obstacle Obs_1 and the scalar field generated by the obstacle λ_{Obs_1} as

$$\mathrm{Obs}_1 \equiv \left\{ (\mathrm{x}, \mathrm{y}, \mathrm{z}_{\mathrm{o},1}) \,|\, (\mathrm{x}, \mathrm{y}) \in [\mathrm{x}_{\mathrm{o},1}, \mathrm{x}_{\mathrm{o},2}] \times [\mathrm{y}_{\mathrm{o},1}, \mathrm{y}_{\mathrm{o},2}] \right\},$$

$$\begin{aligned} \lambda_{\mathrm{Obs}_1}(x, y, z) = & \exp\left\{ -\left(\frac{\mathrm{x} - (\mathrm{x}_{\mathrm{o},1} + \mathrm{x}_{\mathrm{o},2})/2}{(\mathrm{x}_{\mathrm{o},2} - \mathrm{x}_{\mathrm{o},1})/\beta} \right)^{\alpha} \right\} \\ & \exp\left\{ -\left(\frac{\mathrm{y} - (\mathrm{y}_{\mathrm{o},1} + \mathrm{y}_{\mathrm{o},2})/2}{(\mathrm{y}_{\mathrm{o},2} - \mathrm{y}_{\mathrm{o},1})/\beta} \right)^{\alpha} \right\} \\ & \exp\left\{ -\gamma \left(\mathrm{z} - \mathrm{z}_{\mathrm{o},1} \right)^2 \right\}, \end{aligned} \tag{13}$$

where $x_{o,1}$, $x_{o,2}$ and $y_{o,1}$, $y_{o,2}$ represent the configuration of Obs_1 in the x-axis and y-axis respectively, and $z_{o,1}$ represent the position of Obs_1 in the vertical axis. α, β, and γ are parameters in $\mathbb{R}^+$.

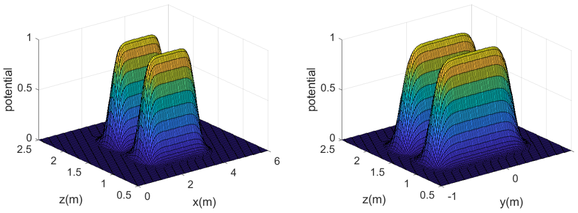
The potential-like field around a tunnel which consists of two plate-like obstacles Obs_1 and Obs_2 is obtained by summing the scalar fields generated by each of the obstacles,

$$\lambda(x, y, z) = \lambda_{\mathrm{Obs}_1}(x, y, z) + \lambda_{\mathrm{Obs}_2}(x, y, z), \tag{14}$$

where Obs_2 is defined as

$$\mathrm{Obs}_2 \equiv \left\{ (\mathrm{x}, \mathrm{y}, \mathrm{z}_{\mathrm{o},2}) \,|\, (\mathrm{x}, \mathrm{y}) \in [\mathrm{x}_{\mathrm{o},1}, \mathrm{x}_{\mathrm{o},2}] \times [\mathrm{y}_{\mathrm{o},1}, \mathrm{y}_{\mathrm{o},2}] \right\}.$$

Fig. 3 visualizes the potential field in Eq. (14). The aforementioned idea of conditional constraints is well reflected in the potential field. Conditionality of the potential field is provided by the first two exponential factors of Eq. (13) in the sense of soft-thresholding. Product of the two factors rapidly converges to zero as the quadrotor-pulley system moves away from the obstacle ($(x,y) \notin [x_{o,1}, x_{o,2}] \times [y_{o,1}, y_{o,2}]$), and maintains a value near 1 if the system is inside the tunnel ($(x,y) \in [x_{o,1}, x_{o,2}] \times [y_{o,1}, y_{o,2}]$).



(a) Potential field at $y = 0$

(b) Potential field at $x = 3$

Fig. 3. Potential field generated by a tunnel-like obstacle, where $(x_{o,1}, x_{o,2}, y_{o,1}, y_{o,2}, z_{o,1}, z_{o,2}) = (2, 4, -0.5, 0.5, 1.2, 1.8)$, $\alpha = 8$, $\beta = 1.6$, and $\gamma = 50$

Larger values of α make the product of the first two factors of Eq. (13) quickly approach 1 as the quadrotor-pulley system enters the tunnel, therefore exhibiting a behavior similar to that of hard-thresholding. Parameter β effects the interval of convergence, and by using smaller values of β, the potential function would maintain a nonzero value further outside the region $[x_{o,1}, x_{o,2}] \times [y_{o,1}, y_{o,2}]$. The "safe distance" between the system and the plate-like obstacle is controlled with parameter γ. Using larger values of γ allows the system to maneuver closer to the obstacle.

The potential function presented in Eq. (13) could be utilized to model rectangular surfaces of three-dimensional obstacles, and is generally applicable for obstacle avoidance tasks in OCPs.

The whole quadrotor-pulley system including the load, cable, and the quadrotor itself should avoid colliding with obstacles. Therefore, we define the internal division points of the cable at a discrete time point t_k as $x_{sys}^i(t_k) = x_L(t_k) - \frac{i}{10} L(t_k) p(t_k)$ for $i = 0, 1, ..., 10$. Finally, we formulate the cost function for tunnel-like obstacle avoidance as

$$J_2 = \sum_{k=0}^{N-1} \frac{1}{2} \Lambda^T(x_k) Q_2 \Lambda(x_k), \tag{15}$$

where $\Lambda(x_k)$ is defined using the potential field in Eq. (14) as

$$\Lambda(x_k) = [\lambda(x_{sys}^0(t_k)), \lambda(x_{sys}^1(t_k)), ..., \lambda(x_{sys}^{10}(t_k))]^T.$$

An inequality constraint shown in Eq. (16) is applied to bound the cable length within a reasonable value.

$$\underline{L} \leq L(t_k) \leq \overline{L}. \tag{16}$$

Utilizing Eqs. (12), (15), and (16), The NLP is reformulated as

$$\min_{x_k, u_k} J_1 + J_2, \tag{17}$$

$$s.t. \;\; x_0 = \hat{x}_0, \tag{18}$$

$$x_{k+1} = F(x_k, u_k), \;\; k = 0, ..., N-1, \tag{19}$$

$$\underline{L} \leq L(t_k) \leq \overline{L}, \;\; k = 0, ..., N-1. \tag{20}$$

3.3 SQP Algorithm

The formulated NLP of Eqs. (17) ∼ (20) is solved using SQP method. For the sake of computational efficiency, a Gauss-Newton Hessian approximate is used instead of second derivatives of the cost function. Rather than algorithms with line search loops, we adopt the state-of-the-art real time iteration (RTI) scheme [13]. The RTI scheme performs only one SQP iteration with a full Newton step, therefore dramatically reducing the computational time. The suboptimal solution of NLP found with the RTI scheme is shown to be effective for MPC applications in [14, 15]. The QP problem is handled with an external solver qpOASES [16] after being condensed.

4. NUMERICAL SIMULATION

The robust performance of the proposed trajectory generation algorithm is validated through numerical simulations. In section 4.1, settings for the simulation are introduced. The simulation results are presented in section 4.2.

4.1 Simulation Settings

The trajectory generation algorithm is implemented using MATMPC [17], a recently-developed MATLAB-based toolbox for nonlinear MPC. Simulations are executed on a laptop equipped with an Intel® Core^{TM} i7-1165G7 and 16GB DDR4 RAM which runs the Windows 10 operating system. Parameters that were used in common for all simulations are listed at Table 2.

Table 2. Parameters used for simulations

$m_Q = 2kg \quad m_L = 0.1kg \quad g = 9.81m/s^2$

$I_Q = diag([0.01, 0.01, 0.02])\, kg \cdot m^2$

$I_p = 0.0003kg \cdot m^2 \quad e_p = [1, 0, 0]^T \quad r_p = 0.03m$

$\alpha = 8 \quad \beta = 1.6 \quad \gamma = 50 \quad \underline{L} = 0 \quad \overline{L} = 1m$

shooting interval time $\Delta t_k = 0.04s$

number of shooting points $N = 60$

total simulation time $t_f = 8s$

$x_{init} = [0, 0, 1, 0, 0, -1, 0, 0, 0,$
$0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0]^T$

$x_{des} = [6, 0, 1, 0, 0, -1, 0, 0, 0,$
$0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0]^T$

$u_{init} = u_{des} = [20.601, 0, 0, 0, -0.02943]^T$

$Q_1 = diag([8, 8, 8, 1, 1, 1, 20, 20, 20, 7, 7, 0,$
$20, 20, 20, 0, 0, 0, 0.8, 8, 0.6, 1, 1, 1, 100])$

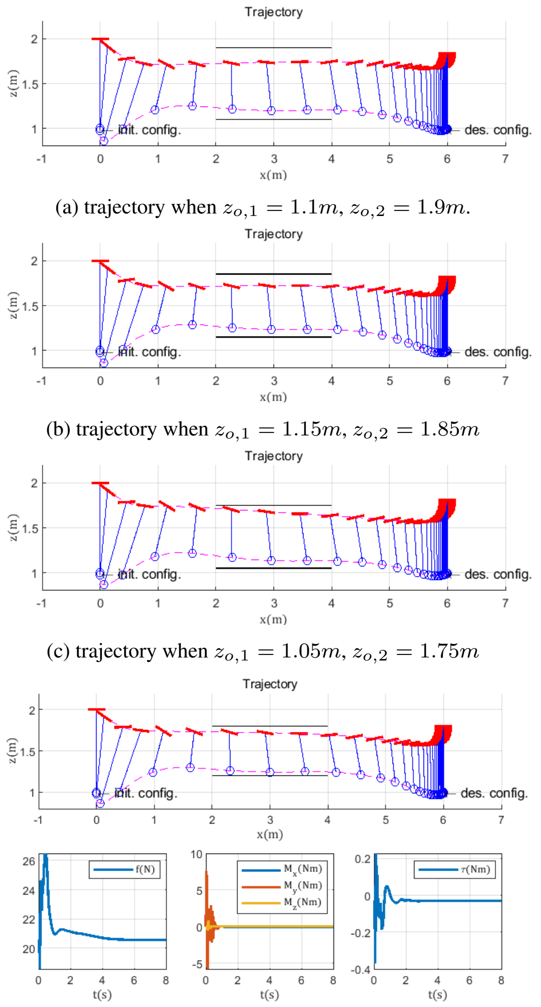
$Q_2 = diag([1.8, 1.8, 1.8, 1.8, 1.8,$
$1.8, 1.8, 1.8, 1.8, 1.8, 1.8])$

$L = diag([500, 500, 500, 1, 1, 1, 10, 10, 10,$
$10, 10, 10, 1, 1, 1, 10, 10, 10, 1, 1])$

4.2 Simulation Results

We present trajectories through tunnels of various heights and lengths, in order to demonstrate the robustness of our algorithm. The initial and desired load positions are fixed at $[0, 0, 1]$ and $[6, 0, 1]$, respectively. Height of the tunnels are set to be shorter than the initial length of the cable. If the quadrotor wasn't equipped with the pulley mechanism, it would have had to make a detour around the obstacle.

Collision-free trajectories of the quadrotor-pulley system through tunnels of various heights and lengths are depicted in Fig. 4 and Fig. 5, respectively. The pulley mechanism is utilized to decrease the vertical distance between the quadrotor and the load before tunnel entrance. Inside the tunnel, the length of the cable is maintained at a certain value, which depends on the height of the passage. As the system exits the tunnel, pulley mechanism unreels the cable to place the load at its desired position.



(a) trajectory when $z_{o,1} = 1.1m$, $z_{o,2} = 1.9m$.

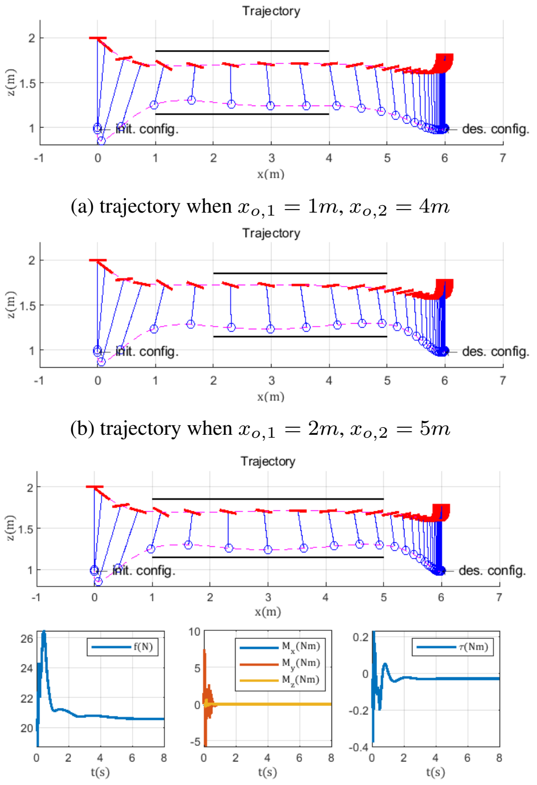
(b) trajectory when $z_{o,1} = 1.15m$, $z_{o,2} = 1.85m$

(c) trajectory when $z_{o,1} = 1.05m$, $z_{o,2} = 1.75m$

(d) trajectory and inputs when $z_{o,1} = 1.2m$, $z_{o,2} = 1.8m$

Fig. 4. Trajectories through tunnels of various heights

Unlike the previous works on trajectory generation through a narrow tunnel, the presented trajectories of the quadrotor-pulley system do not require any aggressive or unrealistic maneuvers. The magnitude of control inputs remain sensible even when navigating through the most challenging obstacles, as shown in Fig. 4d and Fig. 5c.



(a) trajectory when $x_{o,1} = 1m$, $x_{o,2} = 4m$

(b) trajectory when $x_{o,1} = 2m$, $x_{o,2} = 5m$

(c) trajectory and inputs when $x_{o,1} = 1m$, $x_{o,2} = 5m$

Fig. 5. Trajectories through tunnels of various lengths

The proposed trajectory generation method is well applicable for dynamic environments because weight matrix adjustments or predefined waypoints are not required. We also emphasize the ability of the presented method to generate trajectories in real-time. Computational time (CPT) of the NLP corresponding to the case illustrated in Fig. 4a is shown in Table 3. As a comparison, CPT of the problem using a SQP solver without the RTI scheme is also listed.

Table 3. Computational time for solving NLP(ms)

	avg. CPT	max. CPT
with RTI	18.30	68.91
without RTI	177.1	3347

Every time a solution is obtained from Eqs. (17) $\sim$ (20), system trajectory for the next $N\Delta t_k = 2.4s$ is generated. Therefore, the suggested method with the RTI scheme is capable of real-time trajectory generation in dynamic environments, with the average CPT of under 20ms. The solution obtained from the NLP may be suboptimal, but is enough to navigate the system safely through narrow tunnels.

Regarding the application of the proposed trajectory generation algorithm to actual experiments, a fast controller such as the geometric controller presented in [11] would be required to generate the control inputs. Also, swing-angle p could be estimated with the method presented in [18], and L and its derivative are obtainable from the rotation angle of the actuator connected to the pulley mechanism.

5. CONCLUSION

We present a real-time trajectory generation algorithm for a quadrotor with its load suspended from a pulley. MPC algorithm with SQP solver is used to solve the optimal control problem, which is formulated with the cost function that can be generally applied for obstacle avoidance tasks. The RTI scheme is adopted for the real-time capability of the trajectory planning method. The performance of the algorithm is demonstrated with simulations, in which it was able to navigate the system through tunnels of various heights and lengths without aggressive maneuvers. The proposed algorithm is compatible with dynamic environments, since predefined waypoints and weight matrix adjustments are not required.

ACKNOWLEDGEMENT

The authors gratefully appreciate the support from professor H. Jin Kim (Seoul National University).

REFERENCES

[1] K. Sreenath, T. Lee, and V. Kumar, "Geometric control and differential flatness of a quadrotor uav with a cable-suspended load," in *52nd IEEE Conference on Decision and Control*, pp. 2269–2274, IEEE, 2013.

[2] K. Sreenath, N. Michael, and V. Kumar, "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*, pp. 4888–4895, IEEE, 2013.

[3] S. Tang, V. Wüest, and V. Kumar, "Aggressive flight with suspended payloads using vision-based control," *IEEE Robotics and Automation Letters*, vol. 3, no. 2, pp. 1152–1159, 2018.

[4] C. Y. Son, H. Seo, T. Kim, and H. J. Kim, "Model predictive control of a multi-rotor with a suspended load for avoiding obstacles," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 5233–5238, IEEE, 2018.

[5] J. E. Trachte, L. F. G. Toro, and A. McFadyen, "Multi-rotor with suspended load: System dynamics and control toolbox," in *2015 IEEE Aerospace Conference*, pp. 1–9, IEEE, 2015.

[6] S. Notter, A. Heckmann, A. Mcfadyen, and F. Gonzalez, "Modelling, simulation and flight test of a model predictive controlled multirotor with heavy slung load," *IFAC-PapersOnLine*, vol. 49, no. 17, pp. 182–187, 2016.

[7] S. Tang and V. Kumar, "Mixed integer quadratic program trajectory generation for a quadrotor with a cable-suspended payload," in *2015 IEEE international conference on robotics and automation (ICRA)*, pp. 2216–2222, IEEE, 2015.

[8] J. Zeng, P. Kotaru, M. W. Mueller, and K. Sreenath, "Differential flatness based path planning with direct collocation on hybrid modes for a quadrotor with a cable-suspended payload," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 3074–3081, 2020.

[9] P. Foehn, D. Falanga, N. Kuppuswamy, R. Tedrake, and D. Scaramuzza, "Fast trajectory optimization for agile quadrotor maneuvers with a cable-suspended payload," 2017.

[10] C. Y. Son, H. Seo, D. Jang, and H. J. Kim, "Real-time optimal trajectory generation and control of a multi-rotor with a suspended load for obstacle avoidance," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 1915–1922, 2020.

[11] J. Zeng, P. Kotaru, and K. Sreenath, "Geometric control and differential flatness of a quadrotor uav with load suspended from a pulley," in *2019 American Control Conference (ACC)*, pp. 2420–2427, IEEE, 2019.

[12] H. G. Bock and K.-J. Plitt, "A multiple shooting algorithm for direct solution of optimal control problems," *IFAC Proceedings Volumes*, vol. 17, no. 2, pp. 1603–1608, 1984.

[13] M. Diehl, H. G. Bock, J. P. Schlöder, R. Findeisen, Z. Nagy, and F. Allgöwer, "Real-time optimization and nonlinear model predictive control of processes governed by differential-algebraic equations," *Journal of Process Control*, vol. 12, no. 4, pp. 577–585, 2002.

[14] D. Bicego, J. Mazzetto, R. Carli, M. Farina, and A. Franchi, "Nonlinear model predictive control with enhanced actuator model for multi-rotor aerial vehicles with generic designs," *Journal of Intelligent & Robotic Systems*, vol. 100, no. 3, pp. 1213–1247, 2020.

[15] E. Rossi, M. Bruschetta, R. Carli, Y. Chen, and M. Farina, "Online nonlinear model predictive control for tethered uavs to perform a safe and constrained maneuver," in *2019 18th European Control Conference (ECC)*, pp. 3996–4001, IEEE, 2019.

[16] H. J. Ferreau, C. Kirches, A. Potschka, H. G. Bock, and M. Diehl, "qpoases: A parametric active-set algorithm for quadratic programming," *Mathematical Programming Computation*, vol. 6, no. 4, pp. 327–363, 2014.

[17] Y. Chen, M. Bruschetta, E. Picotti, and A. Beghi, "Matmpc-a matlab based toolbox for real-time nonlinear model predictive control," in *2019 18th European Control Conference (ECC)*, pp. 3365–3370, IEEE, 2019.

[18] S. J. Lee and H. J. Kim, "Autonomous swing-angle estimation for stable slung-load flight of multi-rotor uavs," in *2017 IEEE international conference on robotics and automation (ICRA)*, pp. 4576–4581, IEEE, 2017.