ICAS 2016
30th Congress of the International Council of the Aeronautical Sciences
DCC, Daejeon, Korea ; September 25-30, 2016

SAMPLING-BASED MOTION PLANNING FOR COOPERATIVE AERIAL MANIPULATION

Hyoin Kim*, Hyeonbeom Lee*and H. Jin Kim*
***Seoul National University, Seoul, South Korea**

Keywords: *Sampling based algorithm, Rapidly exploring random trees star (RRT*), Aerial manipulation, Cooperative manipulation, Curve parameterization*

Abstract

This paper focuses on the optimal motion planning problem for cooperative aerial manipulation. We use the rapidly exploring random trees star (RRT) algorithm that finds feasible paths quickly and optimizes them. For local planning within RRT*, we developed a trajectory planner using Bezier-curve which utilizes the differential flatness property of the aerial manipulator. Time-parameterization is performed to represent the curve as a function of time.*

1 Introduction

Recently, aerial robots have been popular research platforms since they can perform tasks on sites which are difficult to access. Attached with camera, cable or gripper, the aerial platform can be used in many applications such as exploring, filming the sites or transporting an object [1-3]. Meanwhile, some researchers have investigated the aerial manipulation problem by mounting a multi-DOF robotic arm on aerial robots [4-5], which enables more complex manipulation at high altitudes. However, mounted with a multi-DOF robotic arm, the aerial robots should endure heavier weight, solve the more complex dynamics, and avoid the postures which might cause the platform to drop down.

One of the solutions for those problems is finding an efficient and safe motion trajectories using an optimal motion planner. In this paper, we solve the optimal motion planning problem for aerial manipulators that carry an object cooperatively. Among the various kinds of aerial platforms, we consider multi-copters which can perform the works in narrow and complex environments.

There are many optimal motion planners but only a few compute the solutions for high dimensional problems within practical time. The rapidly exploring random tree star (RRT*) is one of the powerful sampling-based optimal motion planners which can find feasible paths quickly and optimize them [6]. However, RRT* also has a heavy computation issue for non-holonomic systems. In fact, our target platform is non-holonomic, which necessitates an efficient local planner.

Here, we developed a local planner quickly generating motion between sampled nodes within RRT*. The proposed local planner is generated under the consideration of the non-holonomic and differentially flat system. Since most curve fitting methods compute the path with the non-dimensional path parameter, we suggest the additional time-parameterization which can allocate the velocity profile in each desired point and compute the time.

The remainder of the paper is structured as follows: In section 2, we describe problem settings including the dynamic model of an aerial manipulator and kinematics for cooperative missions. RRT* and the proposed local planner are presented in section 3, and simulation results are discussed in section 4. The potential future work and concluding comments are included in section 5.

2 Problem Settings

In this section we described the dynamics and kinematics of aerial manipulators which carry an object cooperatively.

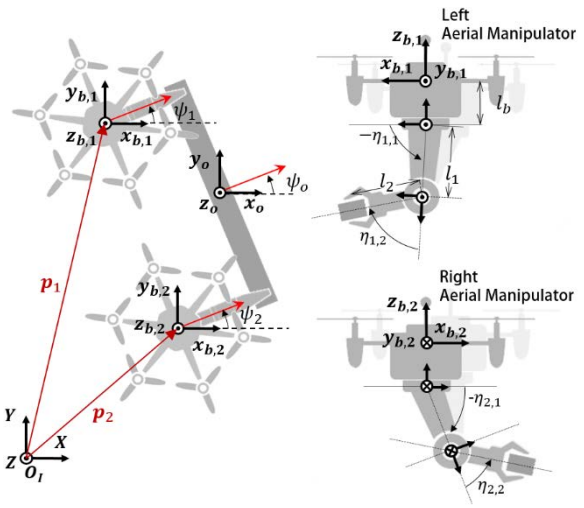

Fig. 1. Configuration of the coordinates for the combined system consisting of a hexacopter and a 2-DOF robotic arm. l_1, l_2 and l_b are lengths of each links and body offset

2.1 Dynamics of single aerial manipulator

Each aerial manipulator is considered as a combined system of 2-DOF robotic arm with gripper and hexa-copter. Then, the general configuration space of single aerial manipulator is represented as $\mathbf{q}_i = [\mathbf{p}_i^\top, \mathbf{\Phi}_i^\top, \boldsymbol{\eta}_i^\top]^\top$ for i=1, 2, where $\mathbf{p}_i = [x_{b,i}, y_{b,i}, x_{b,i}]^\top$ indicates the position of center of body in the Euclidian space. Bold alphabets represent vector quantities. The symbols $\mathbf{\Phi}_i = [\phi_i, \theta_i, \psi_i]^\top$ and $\boldsymbol{\eta}_i = [\eta_{i,1}, \eta_{i,2}]^\top$ denote vectors of Euler angles and joint angles of *i*-th aerial manipulator, respectively. The joint axis of robotic arm is set as shown in Fig. 1. In order to derive the dynamics of the system, the Lagrange-D'Alembert equation is used. The equations of motion can be represented as,

$$\boldsymbol{M}(\mathbf{q}_i)\ddot{\mathbf{q}}_i + \boldsymbol{C}(\mathbf{q}_i, \dot{\mathbf{q}}_i)\dot{\mathbf{q}}_i + \boldsymbol{G}(\mathbf{q}_i) = \boldsymbol{\tau}_i \quad (1)$$

where $\boldsymbol{\tau}_i$ denotes a vector of generalized forces. $M(\mathbf{q}_i) \in \mathbb{R}^{8\times8}$ Indicates the inertia matrix and $C(\mathbf{q}_i, \dot{\mathbf{q}}_i) \in \mathbb{R}^{8\times8}$, $G(\mathbf{q}_i) \in \mathbb{R}^8$ is Coriolis and gravitational matrices. The more detailed equations of motion can be found in [7].

The actuators for hexacopter and robotic arm in practical sense are considered as thrust T, torques of Euler angles $\boldsymbol{\tau}_\Phi = [\tau_\phi, \tau_\theta, \tau_\psi]^\top$, and joint angle $\boldsymbol{\tau}_\eta = [\tau_{\eta_1}, \tau_{\eta_2}]^\top$. From the generalized force $\boldsymbol{\tau}_i$, the actual inputs are derived based on the kinematic relationships as,

$$\begin{bmatrix} \mathrm{T} \\ \tau_{\phi_i} \\ \tau_{\theta_i} \\ \tau_{\psi_i} \\ \tau_{\eta_{i,1}} \\ \tau_{\eta_{i,2}} \end{bmatrix} = \begin{bmatrix} cos\phi_i\, cos\theta_i & 0 & 0 \\ 0 & Q^{-1} & 0 \\ 0 & 0 & I_{2\times2} \end{bmatrix}^{-1} \boldsymbol{\tau}_i \quad (2)$$

where,

$$Q = \begin{bmatrix} 1 & 0 & -sin\theta_i \\ 0 & cos\phi_i & cos\theta_i sin\phi_i \\ 0 & -sin\phi_i & sin\theta_i cos\phi_i \end{bmatrix}$$

2.2 Kinematics for cooperative task

For cooperative manipulation, we assume that each aerial manipulator holds each end of a bar-shaped object with the rigid grasping condition as shown in Fig. 1. From the grasping assumption, aerial manipulators and the object are considered as one system. Let $\mathbf{q}_o = [\mathbf{p}_o^\top, \mathbf{\Phi}_o^\top]^\top$ denote the configuration space of the object. The kinematic constraints from the grasping system are listed as below. The symbol $\bar{\mathbf{p}} = [\mathbf{p}^\top, \mathbf{1}]^\top$ denotes a homogeneous representation of a point $\mathbf{p}$.

$$\begin{aligned} \bar{\mathbf{p}}_{c_i} &= \bar{g}_{oc_i}\bar{\mathbf{p}}_o \\ \bar{\mathbf{p}}_{e_i} &= \bar{g}_{l_{2,i}e_i}\bar{g}_{l_{1,i}l_{2,i}}\bar{g}_{b_i l_{1,i}}\bar{\mathbf{p}}_i \\ \bar{\mathbf{p}}_{e_i} &= \bar{\mathbf{p}}_{c_i} \end{aligned} \quad (3)$$

Here, Each of $\mathbf{p}_o, \mathbf{p}_i, \mathbf{p}_{e_i}$ and $\mathbf{p}_{c_i}$ indicates the position vector of the center of object, *i*-th hexacopter, gripper and both ends of object. $\bar{g}_{l_{2,i}e_i}, \bar{g}_{l_{1,i}l_{2,i}}$ and $\bar{g}_{b_i l_{1,i}}$ represent the matrices of homogeneous rigid transformation [8] which calculate the positions resulted from the joint angles and Euler angles of each aerial manipulator. $\bar{g}_{oc_i}$ is the one that calculates the each tip position of the object. From the grasping assumption, the Euler angles of object and each aerial manipulators are equal.

From the dynamics, we find that the roll and pitch angle ϕ_i, θ_i can be represented by other state variables and their time derivatives. The systems which have this dynamic property are called the differentially flat system and the representing variables are called flat outputs. With Eqn. (3) and utilizing the flat outputs, the configuration space of cooperative aerial

Algorithm 1 The Bezier-RRT* Algorithm

$V \leftarrow v_{init}; E \leftarrow \emptyset; J \leftarrow 0; i \leftarrow 0;$
while $i < N$ **do**
  $v_{rand} \leftarrow \texttt{Sample}(i)$
  $(\mathrm{Z}_{\mathrm{near}}, \mathrm{U}_{\mathrm{near}}) \leftarrow \texttt{Near}(\mathcal{T}, v_{rand})$
  $(v_{min}, v_{new}, u_{new}) \leftarrow \texttt{parent}(\mathrm{Z}_{\mathrm{new}}, \mathrm{U}_{\mathrm{near}})$
  if $\texttt{ObstacleFree}(v_{new})$ **then**
    $i \leftarrow i+1;$
    $(J, V, \mathcal{T}) \leftarrow \texttt{InsertNode}(\mathcal{T}, v_{new}, u_{new})$
    $(J, \mathcal{T}) \leftarrow \texttt{B-Rewire}(\mathcal{T}, \mathrm{Z}_{\mathrm{near}}, v_{new})$
  end if
end while
Return $\mathcal{T}$

Algorithm 2 Near function

Require: $v_{rand}, \mathcal{T}$
Ensure: $\mathrm{Z}_{\mathrm{near}}, \mathrm{U}_{\mathrm{near}}$
$\mathrm{Z}_{\mathrm{range}} \leftarrow \texttt{Range}(v_{rand}, \mathcal{T})$
$(\mathrm{Z}_{\mathrm{near}}, \mathrm{U}_{\mathrm{near}}) \leftarrow \texttt{B-Extend}(\mathrm{Z}_{\mathrm{range}}, v_{rand})$

manipulators is represented as $\mathbf{q} = [\mathbf{p}_o^{\mathsf{T}}, \psi_o, \eta_{1,1}, \eta_{1,2}, \eta_{2,2}]^{\mathsf{T}}$.

3 Motion Planning

Here, we briefly explain the RRT* algorithm and describe the proposed local planning processes using Bezier-curve within RRT*.

3.1 RRT*

RRT*, the outstanding variant of the rapidly exploring random trees (RRTs), keeps the property of quickly searching feasible paths from the original RRTs, and moreover, it achieves asymptotic optimality until the execution finishes. The below are the brief descriptions of the processes in RRT*. Let $\mathrm{Z} \in \mathbb{R}^n$ be the configuration space and the region occupied/vacated by obstacles are denoted $\mathrm{Z}_{\mathrm{obs}}$ and $\mathrm{Z}_{\mathrm{free}} \in \mathrm{Z}/\mathrm{Z}_{\mathrm{obs}}$. The symbols V, E and J are the groups of nodes (or vertices), edges and performance indexes of each node, and v_{init} is the initial configuration. By repeating random sampling v_{rand} and finding the most efficient parent node to reach v_{rand} from near groups with the maximum distance of γ, the nodes and edges form the tree. The additional rewiring process enhances the optimality of local paths by rearranging connections. Here, the local planner, which is required in the tree extension and rewiring processes, computes the control input $u \in U: \{u | \dot{v} = f(v, u)\}$ that leads the system from the start to final node.

3.2 Local planner

*3.2.1 General form or Bezier RRT**

When we neglect the dynamic model in RRT*, the local planner is replaced by a simple straight line connecting the nodes. However, as the manipulator-mounted hexarotor can easily lose safety in some constrained configuration, we should consider the local planner which obeys the dynamic properties and constraints. In addition, as described before, the local planner for RRT* is used in rewiring and extend processes. In both processes, the local planner should compute the paths one or more times in order to pick collision-free and shorter path, as shown Algorithm 1 and 2. Hence, the local planner should compute the path fast enough to avoid the slow completeness of RRT*. Here, we propose the local planning using the Bezier curve with time-parameterization. By using the Bezier curve, we can develop a deterministic path generator with non-dimensional parameters. The basic form of n_b-order of Bezier curve is written as,

$$\mathbf{B}(\xi) = \sum_{i=0}^{n_b} \binom{n_b}{i} (1-\xi)^{n_b - i} \xi^i \mathbf{P}_i \qquad (4)$$

where $\xi \in [0, 1]$ indicates a parameter that does not have any dimension. Each of $\mathbf{P}_0$ and $\mathbf{P}_{n_b}$ denotes start and end states where $\mathbf{P}_i$ for $i = 1,2, \ldots, n_b - 1$ is the i-th control point.
In order to connect the states that have continuous velocity and acceleration, we derived the follow equations of $\mathbf{P}_i$ for $i = 0,1,2, \ldots, n_b$.

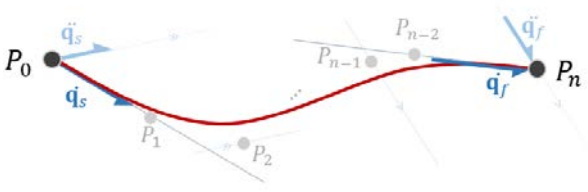

Fig. 2. The general form of local planning with Bezier-curve

$$\mathbf{P}_0 = \mathbf{q_s} \tag{5}$$

$$\mathbf{P}_1 = \mathbf{P}_0 + \frac{\dot{\mathbf{q}}_\mathbf{s}}{n_b \dot{\xi}_s}$$

$$\mathbf{P}_2 = \mathbf{P}_0 + \frac{2\dot{\mathbf{q}}_\mathbf{s}}{n_b \dot{\xi}_s} + \frac{\ddot{\mathbf{q}}_\mathbf{s}}{n_b(n_b-1)\dot{\xi}_s^2} - \frac{\dot{\mathbf{q}}_\mathbf{s}}{n_b(n_b-1)\dot{\xi}_s^3}\ddot{\xi}_s$$

$$\mathbf{P}_{n_b} = \mathbf{q_f}$$

$$\mathbf{P}_{n_b-1} = \mathbf{P}_{n_b} - \frac{\dot{\mathbf{q}}_\mathbf{f}}{n\dot{\xi}_f}$$

$$\mathbf{P}_{n_b-2} = \mathbf{P}_{n_b} - \frac{2\dot{\mathbf{q}}_\mathbf{f}}{n_b \dot{\xi}_f} + \frac{\ddot{\mathbf{q}}_\mathbf{f}}{n_b(n_b-1)\dot{\xi}_f^2} + \frac{\dot{\mathbf{q}}_\mathbf{f}}{n_b(n_b-1)\dot{\xi}_f^3}\ddot{\xi}_f$$

Then we obtain the local planner with design parameters $\dot{\xi}_s, \ddot{\xi}_s, \dot{\xi}_f$ and $\ddot{\xi}_f$. The control points are located with regard to the velocity and acceleration as shown in Fig. 2.

3.2.2 Time parameterization

The time representation of parametric curve of ξ allows us to allocate the desired velocity value at ξ. With velocity information $v(\xi)$ along the ξ, the time can be computed from the equations listed below. Here, $v(\xi)$ should have positive values.

$$ds = v(\xi)dt = \frac{ds}{d\xi}d\xi \tag{6}$$

$$t(\xi) = \int dt = \int \frac{ds}{d\xi}\frac{1}{v(\xi)}\,d\xi$$

In order to satisfy boundary conditions at start and final points, the conditions for $v(\xi)$ are given as follows:

$$v'(\xi=0) = \frac{\ddot{\mathbf{q}}_s^{\top}\dot{\mathbf{q}}_s}{\|\dot{\mathbf{q}}_s\|\dot{\xi}_s} \tag{7}$$

$$v'(\xi=1) = \frac{\ddot{\mathbf{q}}_f^{\top}\dot{\mathbf{q}}_f}{\|\dot{\mathbf{q}}_f\|\dot{\xi}_f}$$

Finally, with the design parameters in section 3.2.1 and velocity $v(\xi)$, the local planner computes the trajectories as a function of time.

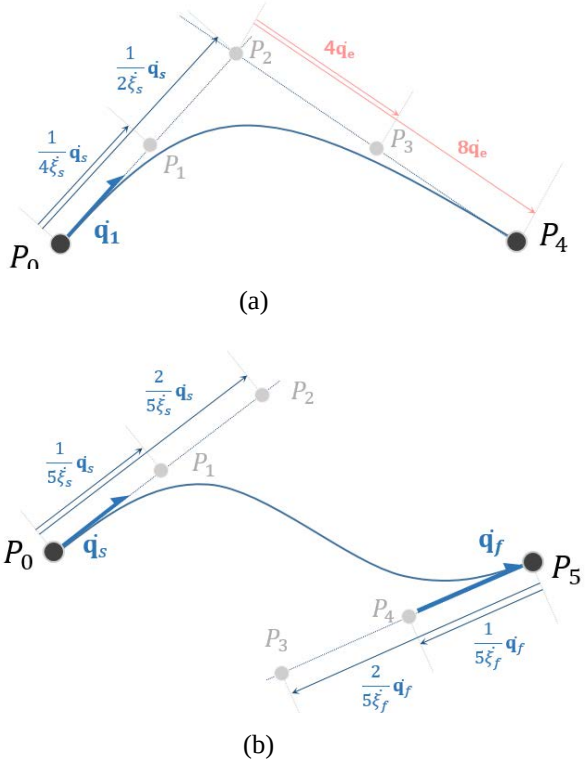

Fig. 3. The form of local planner in (a) extend and (b) rewire processes

3.2.3 Local planning in extend process

In order to reduce the sampling space, we separately design the local planner in the extension and rewire processes.

For extension, it is possible to sample the positions only, because we just need to extend the nodes. Compared to sampling all the state variables with velocity and acceleration, it raises the possibility for picking adequate candidates of parent nodes.

Here, n_b is set to be five. We assume that accelerations at each point are zero. The design parameters for local planner in the extend process are $\dot{\xi}_s, \ddot{\xi}_s, \ddot{\xi}_f$ and $\|\dot{\mathbf{q}}_f\|$. Since there is no bound of velocity at final points, we use $\|\dot{\mathbf{q}}_f\|$ and make the paths end at adequate velocity. For both extension and rewire processes, $\ddot{\xi}_s$ and $\ddot{\xi}_f$ are set to be zero and $\dot{\xi}_s = \max(\|\mathbf{q}_f - \mathbf{q}_s\|, \gamma)/\gamma$. In order to calculate Eqn. (7), we should know the vector elements of $\dot{\mathbf{q}}_f$. $\dot{\mathbf{q}}_f$ is obtained from the following equations.

$$P_2 = \mathbf{q_s} + \frac{\dot{\mathbf{q}}_\mathbf{s}}{2\dot{\xi}_s} = \mathbf{q_f} - \frac{\dot{\mathbf{q}}_\mathbf{f}}{2\dot{\xi}_f} \quad (8)$$

$$\frac{\dot{\mathbf{q}}_\mathbf{f}}{\dot{\xi}_f} = \mathbf{2}(\mathbf{q}_f - \mathbf{q}_s) - \frac{\dot{\mathbf{q}}_s}{\dot{\xi}_f} = \dot{\mathbf{q}}_\mathbf{e} \quad (9)$$

Finally, $\dot{\mathbf{q}}_\mathbf{f}$ and $\dot{\xi}_f$ are determined from the designed $\|\dot{\mathbf{q}}_\mathbf{f}\|$. In Fig. 3(a), the curve for extension process and $\dot{\mathbf{q}}_\mathbf{e}$ are shown. By setting $v(\xi)$ to satisfy Eqn. (7), the trajectories are generated as a function of time.

3.2.4 Local planning in rewire process

Unlike the extension process, the rewiring process should exactly connect the starting and final nodes to satisfy velocity and acceleration conditions at each end point. In order to satisfy the conditions above and avoid a large number of designing parameters, we set n_b to six. Since we set the acceleration to zero at end points as described in 3.2.3, the accelerations for all the nodes are zero. By setting $\dot{\xi}_f = \dot{\xi}_s$ for the rewiring process, the path is generated as shown in Fig. 3(b).

4 Simulation Results

4.1 Simulation Settings

In the simulation scenario, two aerial manipulators transport a common object and put it inside the vertical gap between two obstacles as shown in Fig. 4. For local planning, design parameters are set as described in section 3.2.3 and 3.2.4. The velocity $v(\xi)$ is designed to keep 0.5~0.6 m/s with satisfying Eqn. (7) at each node. The performance index is set to travel time. In order to avoid configurations which cause large moments on the body and consume extra motor torques, we used weighting. Because the range of velocity is designed as 0.5 to 0.6 m/s, we can expect that the path for minimal travel time will be similar to the path for optimal travel length. The actuation limits are considered based on angular velocity criteria from general specifications of the platform. The lower and upper bounds for angular velocity of each *i*-th rotor, ω_i [rad/s] for i=1,2,…6, are denoted as Eqn. (10). Each value of ω_i is obtained from Eqn. (11) where M_o is the motor mapping matrix as listed on [7].

$$\frac{100}{3}\pi < \omega_i < \frac{1000}{3}\pi \quad (10)$$

$$\begin{bmatrix}\omega_1^2\\ \omega_2^2\\ \omega_3^2\\ \omega_4^2\\ \omega_5^2\\ \omega_6^2\end{bmatrix} = M_o^+ \begin{bmatrix}T\\ \tau_\phi\\ \tau_\theta\\ \tau_\psi\end{bmatrix} \quad (11)$$

4.2 Results and discussions

In Fig. 4, the total trajectory in Euclidian space is shown. Red, blue and green lines indicate the body trajectories of object and manipulators. The

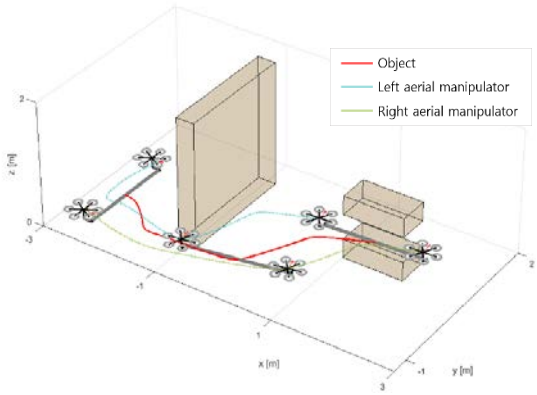

Fig. 4. Optimized path.

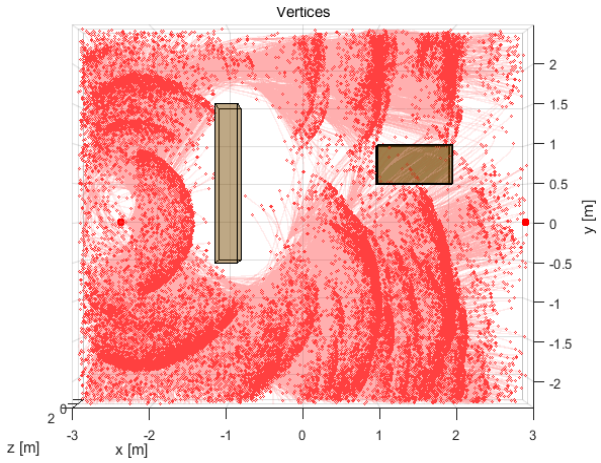

Fig. 5. Vertices and local paths for *N* =30000

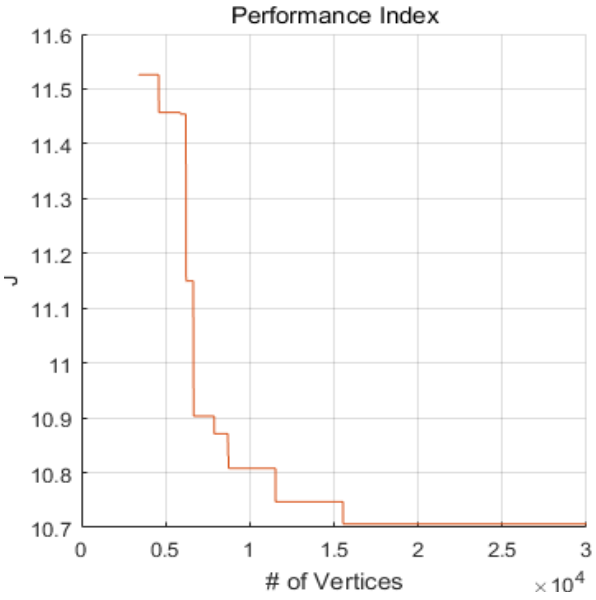

Fig. 6. Performance index as the number of nodes increases.

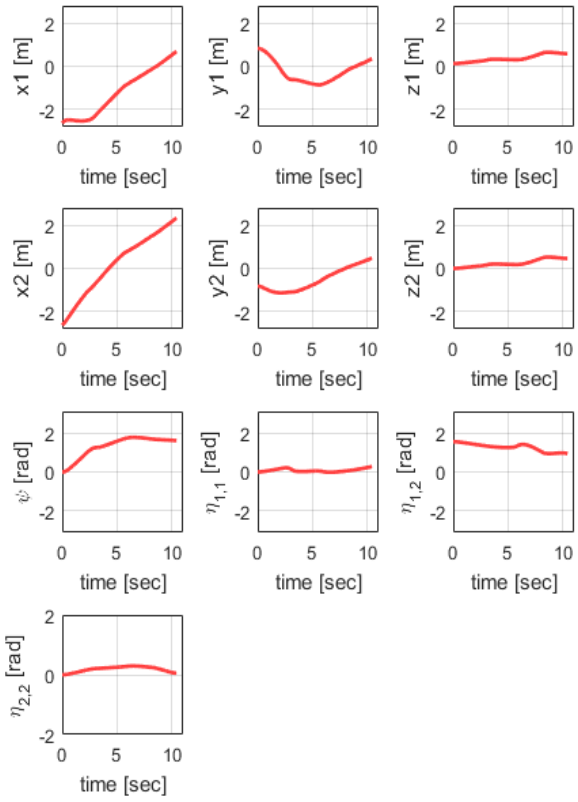

Fig. 7 Time history of state variables

nodes and local paths are shown in Fig. 5. From the branch of paths, we can find that the planner connects the nodes smoothly. In Fig. 6, we found that, as the number of the sampling nodes increases, the performance index of the initial path keeps minimized. The time histories of state variables are listed in Fig. 7.

5 Conclusions and Future work

This paper presents a path planning approach using RRT* for a complex system performing cooperative tasks in 3D space. For our target system, a hexarotor with a multi-DOF robotic arm, we derived the compact sampling space with dynamic constraints and kinematics. For local planning within RRT*, a deterministic local planner was developed using the Bezier curve. With the time-parameterization method we developed, it is possible to allocate the velocity to non-dimensional parametric curve at desired position. This time-parameterization can be exploited other curve fitting methods. The simulation results show the local planner is suitable for the problem. However, even with the fast local planner, we faced the fact that this planner cannot be used as a real-time planner with satisfactory optimized results. Although the initial path is obtained quickly, the optimization may need more iterations generally. Therefore, one of the potential future works is a real-time planning strategy exploiting RRT*.

Acknowledgements

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korean government (MSIP) (2014R1A2A1A12067588 and 0420-20140127)

References

[1] Lindsey, Q., Mellinger, D. and Kumar, V., Construction with quadrotor teams. *Autonomous Robots*, *33*(3), pp.323-336, 2012.

[2] Willmann, J., Augugliaro, F., Cadalbert, T., D'Andrea, R., Gramazio, F. and Kohler, M., Aerial robotic construction towards a new field of architectural research. *International journal of architectural computing*, *10*(3), pp.439-459, 2012.

[3] Palunko, I., Fierro, R. and Cruz, P., Trajectory generation for swing-free maneuvers of a quadrotor with suspended payload: A dynamic programming approach. In *Robotics and Automation (ICRA), 2012 IEEE International Conference on* (pp. 2691-2697). IEEE, 2012, May.

[4] Kim, S., Choi, S. and Kim, H.J., Aerial manipulation using a quadrotor with a two dof robotic arm. In *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems* (pp. 4990-4995). IEEE, 2013, November.

[5] Huber, F., Kondak, K., Krieger, K., Sommer, D., Schwarzbach, M., Laiacker, M., Kossyk, I., Parusel, S., Haddadin, S. and Albu-Schäffer, A., First analysis and experiments in aerial manipulation using fully actuated redundant robot arm. In *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems* (pp. 3452-3457). IEEE, 2013, November.

[6] Karaman, S. and Frazzoli, E., Sampling-based algorithms for optimal motion planning. *The International Journal of Robotics Research*, *30*(7), pp.846-894, 2011.

[7] Lee, H., Kim, H. and Kim, H.J., Path planning and control of multiple aerial manipulators for a cooperative transportation. In *Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on*(pp. 2386-2391). IEEE, 2015, September.

[8] Murray, R.M., Li, Z., Sastry, S.S. and Sastry, S.S., *A mathematical introduction to robotic manipulation*. CRC press, 1994.

Contact Author Email Address

mailto: rlagydls@gmail.com

Copyright Statement