ME579 - Final Project Report
Euler-Lagrange Drone Dynamics and Trajectory Control

Student: Chunhua Ying **ID:**11676923

Introduction:

UAV Express (UAV Express) is an unmanned low-altitude aircraft operated by radio remote control equipment and self-provided program control devices to carry packages and automatically deliver them to their destinations. SF Express currently has it, but it has not been widely used. The automated drone express delivery system uses drones to replace manual delivery of express delivery. It aims to realize the automation, unmanned and informatization of express delivery, improve the delivery efficiency and service quality of express delivery, so as to alleviate the demand for express delivery and express delivery service capabilities. The implementation of this system can effectively cope with the huge increase in order volume, eliminate the danger of express "explosion", improve the service quality of the express industry, reduce the delay rate, damage rate, loss rate, and express complaint rate of express delivery, while also reducing operating costs, warehouse costs, labor costs, etc., enhance industry competitiveness, and make express delivery safer, more reliable, and faster. Through this project, one can combine what have been learned from class and further explore the robot dynamics and kinematic of quadcopters.

Content

1. Introduction to mathematical model of Euler-Lagrange drone dynamics
2. Validation using the example case
3. Trajectory design

1. Introduction to mathematical model of Euler-Lagrange drone dynamics

1.1 Forward kinematics and dynamics

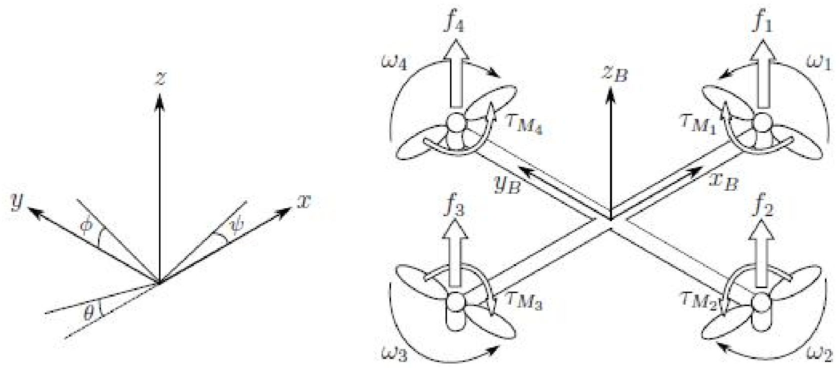


Figure 1: The inertial and body frames of a quadcopter

From the above inertial and body frames definition, coordinates and Euler angles in inertial frame could be defined as follows

$$\xi = \begin{bmatrix} x \\ y \\ z \end{bmatrix}, \quad \eta = \begin{bmatrix} \phi \\ \theta \\ \psi \end{bmatrix}, \quad q = \begin{bmatrix} \xi \\ \eta \end{bmatrix}.$$

The rotation matrix from body frame to the inertial frame is simply

$$R = \begin{bmatrix} C_\psi C_\theta & C_\psi S_\theta S_\phi - S_\psi C_\phi & C_\psi S_\theta C_\phi + S_\psi S_\phi \\ S_\psi C_\theta & S_\psi S_\theta S_\phi + C_\psi C_\phi & S_\psi S_\theta C_\phi - C_\psi S_\phi \\ -S_\theta & C_\theta S_\phi & C_\theta C_\phi \end{bmatrix}$$

To calculate the total thrust T upward and torque tau around three inertial axis, define 4 corresponding angular velocity w_i as shown in figure 1, thus

$$T = \sum_{i=1}^{4} f_i = k \sum_{i=1}^{4} \omega_i^2, \quad T^B = \begin{bmatrix} 0 \\ 0 \\ T \end{bmatrix},$$

$$\tau_B = \begin{bmatrix} \tau_\phi \\ \tau_\theta \\ \tau_\psi \end{bmatrix} = \begin{bmatrix} l k \left(-\omega_2^2 + \omega_4^2\right) \\ l k \left(-\omega_1^2 + \omega_3^2\right) \\ \sum_{i=1}^{4} \tau_{M_i} \end{bmatrix},$$

In which k is lift constant, l is the distance between the rotor and the center of mass. Where $\tau_{Mi}=bw_i^2$, b_i as the drag constant. Note that,

$$\sum_{i=1}^{4} \tau_{Mi} = b(\omega_1^2 - \omega_2^2 + \omega_3^2 - \omega_4^2)$$

Through fundamental kinematics and dynamics derivation, Newton-Euler equations could be obtained as following:

$$m\ddot{\boldsymbol{\xi}} = \boldsymbol{G} + \boldsymbol{R}\boldsymbol{T}_B,$$

$$\begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} \end{bmatrix} = -g \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} + \frac{T}{m} \begin{bmatrix} C_\psi S_\theta C_\phi + S_\psi S_\phi \\ S_\psi S_\theta C_\phi - C_\psi S_\phi \\ C_\theta C_\phi \end{bmatrix}.$$

And,

$$\ddot{\boldsymbol{\eta}} = \boldsymbol{J}^{-1}\left(\boldsymbol{\tau}_B - \boldsymbol{C}\left(\boldsymbol{\eta}, \dot{\boldsymbol{\eta}}\right)\dot{\boldsymbol{\eta}}\right).$$

$$\boldsymbol{C}\left(\boldsymbol{\eta}, \dot{\boldsymbol{\eta}}\right) = \begin{bmatrix} C_{11} & C_{12} & C_{13} \\ C_{21} & C_{22} & C_{23} \\ C_{31} & C_{32} & C_{33} \end{bmatrix},$$

$$\begin{aligned}
C_{11} &= 0 \\
C_{12} &= (I_{yy} - I_{zz})(\dot{\theta} C_\phi S_\phi + \dot{\psi} S_\phi^2 C_\theta) + (I_{zz} - I_{yy})\dot{\psi} C_\phi^2 C_\theta - I_{xx}\dot{\psi} C_\theta \\
C_{13} &= (I_{zz} - I_{yy})\dot{\psi} C_\phi S_\phi C_\theta^2 \\
C_{21} &= (I_{zz} - I_{yy})(\dot{\theta} C_\phi S_\phi + \dot{\psi} S_\phi^2 C_\theta) + (I_{yy} - I_{zz})\dot{\psi} C_\phi^2 C_\theta + I_{xx}\dot{\psi} C_\theta \\
C_{22} &= (I_{zz} - I_{yy})\dot{\phi} C_\phi S_\phi \\
C_{23} &= -I_{xx}\dot{\psi} S_\theta C_\theta + I_{yy}\dot{\psi} S_\phi^2 S_\theta C_\theta + I_{zz}\dot{\psi} C_\phi^2 S_\theta C_\theta \\
C_{31} &= (I_{yy} - I_{zz})\dot{\psi} C_\theta^2 S_\phi C_\phi - I_{xx}\dot{\theta} C_\theta \\
C_{32} &= (I_{zz} - I_{yy})(\dot{\theta} C_\phi S_\phi S_\theta + \dot{\phi} S_\phi^2 C_\theta) + (I_{yy} - I_{zz})\dot{\phi} C_\phi^2 C_\theta \\
&\quad + I_{xx}\dot{\psi} S_\theta C_\theta - I_{yy}\dot{\psi} S_\phi^2 S_\theta C_\theta - I_{zz}\dot{\psi} C_\phi^2 S_\theta C_\theta \\
C_{33} &= (I_{yy} - I_{zz})\dot{\phi} C_\phi S_\phi C_\theta^2 - I_{yy}\dot{\theta} S_\phi^2 C_\theta S_\theta - I_{zz}\dot{\theta} C_\phi^2 C_\theta S_\theta + I_{xx}\dot{\theta} C_\theta S_\theta.
\end{aligned}$$

And Jacobian matrix J calculated as,

$$= \begin{bmatrix} I_{xx} & 0 & -I_{xx} S_\theta \\ 0 & I_{yy} C_\phi^2 + I_{zz} S_\phi^2 & (I_{yy} - I_{zz}) C_\phi S_\phi C_\theta \\ -I_{xx} S_\theta & (I_{yy} - I_{zz}) C_\phi S_\phi C_\theta & I_{xx} S_\theta^2 + I_{yy} S_\phi^2 C_\theta^2 + I_{zz} C_\phi^2 C_\theta^2 \end{bmatrix}.$$

To account for the drag force generated by the air resistance, drag force coefficients A_x, A_y, A_z for velocities in inertial frame are included, the equation can be modified as

$$\begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} \end{bmatrix} = -g \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} + \frac{T}{m} \begin{bmatrix} C_\psi S_\theta C_\phi + S_\psi S_\phi \\ S_\psi S_\theta C_\phi - C_\psi S_\phi \\ C_\theta C_\phi \end{bmatrix} - \frac{1}{m} \begin{bmatrix} A_x & 0 & 0 \\ 0 & A_y & 0 \\ 0 & 0 & A_z \end{bmatrix} \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix}$$

With these equations, quadcopters' forward kinematics and dynamics could be conducted by the input of angular velocity w_i. Next the validation of this method is carried out by 2 examples.

1.2 Trajectory design

From the equation of position in inertial frame, one can obtain an inverse representation of thrust T,

$$\boldsymbol{T}_B = \begin{bmatrix} 0 \\ 0 \\ T \end{bmatrix} = \boldsymbol{R}^{\mathrm{T}} \left(m \left(\ddot{\xi} + \begin{bmatrix} 0 \\ 0 \\ g \end{bmatrix} \right) + \begin{bmatrix} A_x & 0 & 0 \\ 0 & A_y & 0 \\ 0 & 0 & A_z \end{bmatrix} \dot{\xi} \right)$$

And from this, by pre-defined ξ and ψ angle, the left φ, θ and T can be calculated as

$$\phi = \arcsin\left(\frac{d_x S_\psi - d_y C_\psi}{d_x^2 + d_y^2 + (d_z + g)^2} \right),$$

$$\theta = \arctan\left(\frac{d_x C_\psi + d_y S_\psi}{d_z + g} \right),$$

$$T = m \left(d_x \left(S_\theta C_\psi C_\phi + S_\psi S_\phi \right) + d_y \left(S_\theta S_\psi C_\phi - C_\psi S_\phi \right) + \left(d_z + g \right) C_\theta C_\phi \right),$$

in which

$$d_x = \ddot{x} + A_x \dot{x}/m,$$
$$d_y = \ddot{y} + A_y \dot{y}/m,$$
$$d_z = \ddot{z} + A_z \dot{z}/m.$$

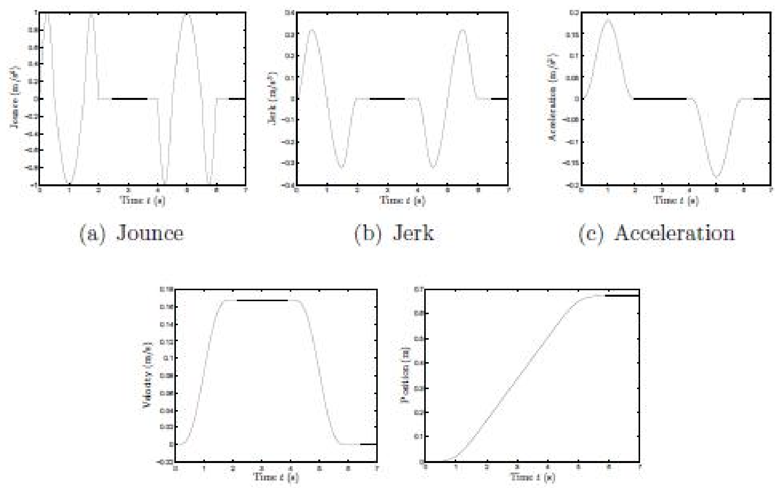
With the angular velocities and accelerations, the torques can be solved from the Newton-Euler equations of Euler angles. When the torques and thrust are known, the control inputs w_i can be calculated from

$$\begin{aligned} \omega_1^2 &= \frac{T}{4k} - \frac{\tau_\theta}{2kl} - \frac{\tau_\psi}{4b} \\ \omega_2^2 &= \frac{T}{4k} - \frac{\tau_\phi}{2kl} + \frac{\tau_\psi}{4b} \\ \omega_3^2 &= \frac{T}{4k} + \frac{\tau_\theta}{2kl} - \frac{\tau_\psi}{4b} \\ \omega_4^2 &= \frac{T}{4k} + \frac{\tau_\phi}{2kl} + \frac{\tau_\psi}{4b} \end{aligned}$$

Following this path, a backward quadcopter kinematics and dynamics could be simply carried out. But the choice of trajectory need to be very careful, as the it not only needs to be continuous but also its forth order derivative needs to be continuous. Here follow the Heuristic method, a Jounce function is defined

$$f(t) = \begin{cases} a \sin\left(\frac{1}{b} \pi t\right), & 0 \le t \le b, \\ -a \sin\left(\frac{1}{b} \pi t - \pi\right), & b \le t \le 3b, \\ a \sin\left(\frac{1}{b} \pi t - 3\pi\right), & 3b \le t \le 4b. \end{cases}$$

From several integration of this function, the acceleration, velocity and position could be obtained. A general process can be seen in figure 2.



(a) Jounce (b) Jerk (c) Acceleration

(d) Velocity (e) Position

Figure 2: position and its derivatives with given jounce

With this Jounce function, any planned distance could be fulfilled, the time needed to achieve this distance could also be adjusted. Here I derived the expression of planned distance S with respect to the parameter a, b, c as,

$$S = (3ab^3(\pi^2 + 4)(4b + c)) / \pi^3$$

A validation has also been shown in the validation section.

2. Validation using the example case

2.1 Forward validation

Following the example in [1], Forward validation consist of ascending row, pitch, yaw, in total 4 process.The results can be seen in figure 3. Parameter are listed in Table 1.

Table 1: Parameter values for simulation

Parameter	Value	Unit
g	9.81	m/s^2
m	0.468	kg
l	0.225	m
k	$2.980 \cdot 10^{-6}$	
b	$1.140 \cdot 10^{-7}$	
I_M	$3.357 \cdot 10^{-5}$	kg m^2

Parameter	Value	Unit
I_{xx}	$4.856 \cdot 10^{-3}$	kg m^2
I_{yy}	$4.856 \cdot 10^{-3}$	kg m^2
I_{zz}	$8.801 \cdot 10^{-3}$	kg m^2
A_x	0.25	kg/s
A_y	0.25	kg/s
A_z	0.25	kg/s

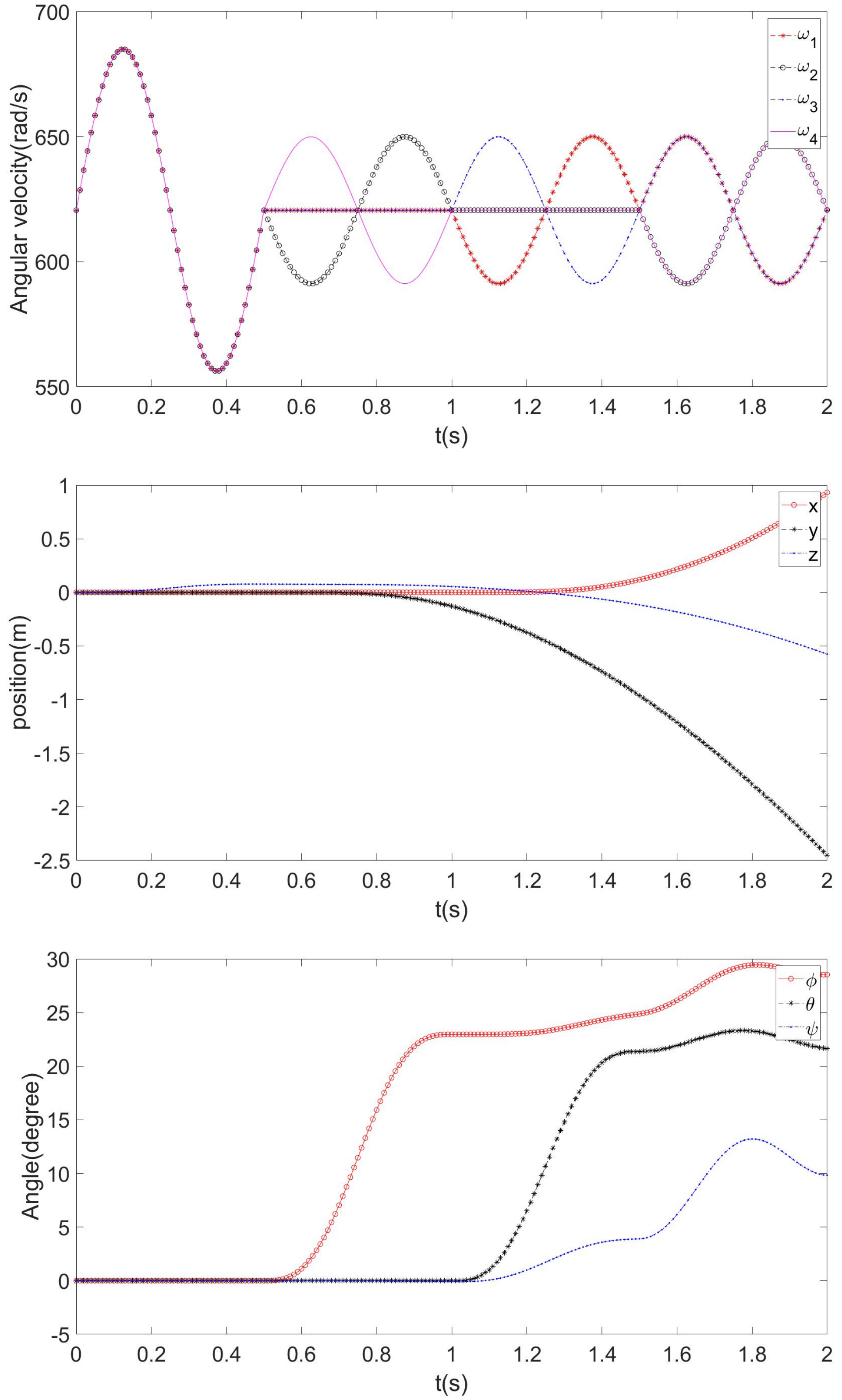


Figure 3: Forward kinematics and dynamics

2.2 Trajectory design validation

Same to the example in [1], the results have been recreated in figure 4.

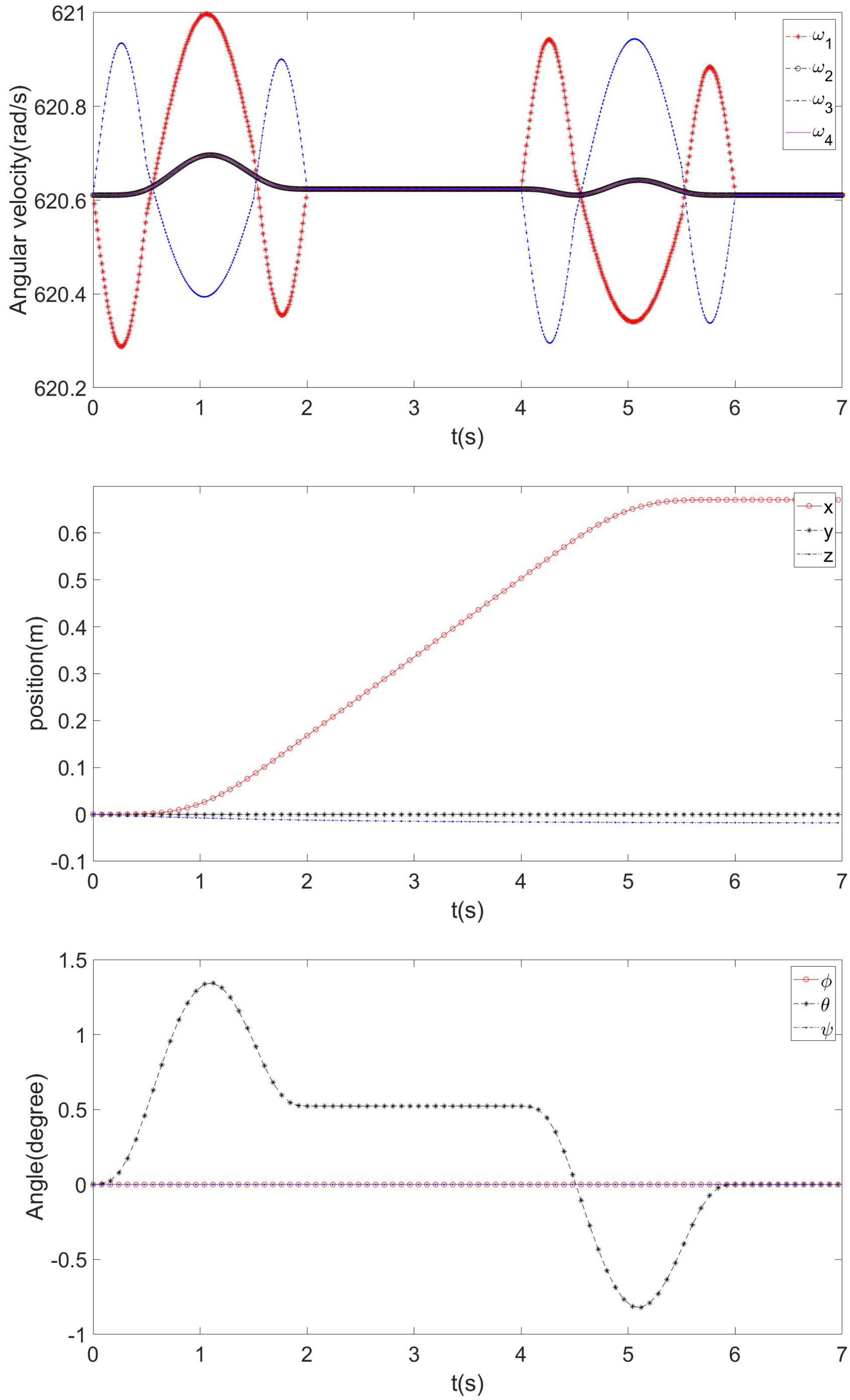


Figure 4: trajectory design test

3. Trajectory design

After previous preparation and validation, it's time to start trajectory design for quadcopter delivering task.

3.1 Basic parameters

When the quadcopter is flying, it needs safe hovering height and droping height.

i) Pre-defined h_0, h_d, W_0, while P_0 as the delivering office station or vehicle position.
ii) Input destination coordinates(customer apartment) P_1 and box weight W_{box}, achieving an safe drone delivering mission.

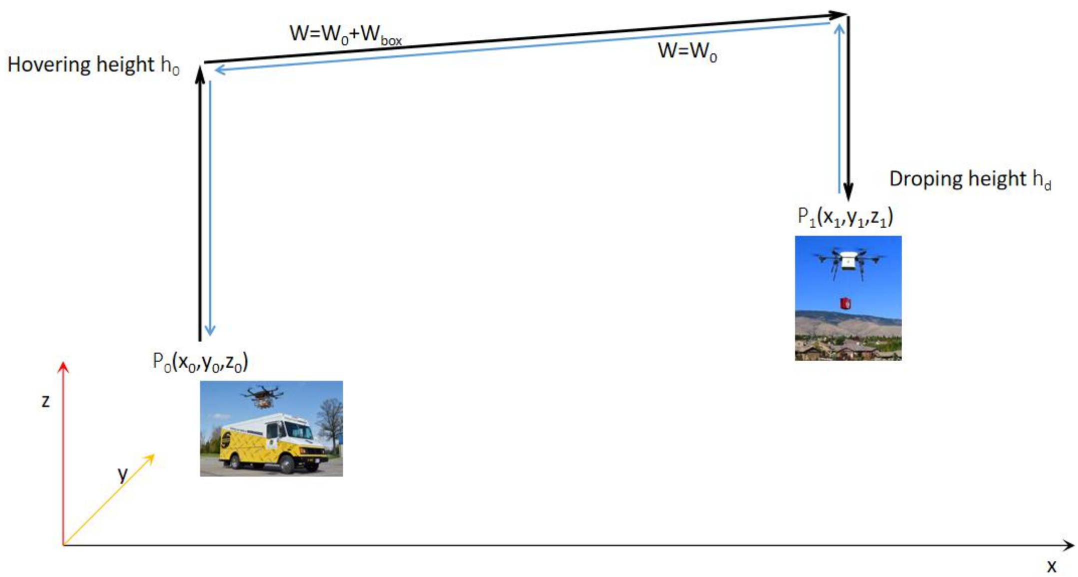


Figure 5: Schematic representation of a delivering UAV

These parameters used in this simulation are listed in Table 2, while other configuration parameters are inherited from Table 1.

Table2: Flight parameters

X_0	Y_0	Z_0	H_0	H_d	X_1	Y_1	W_0	W_{box}	dt
0.0 m	0.0 m	0.0 m	50.0 m	10.0 m	10.0 m	10.0 m	0.468 kg	0.2 kg	0.001s

3.2 Result

Considering a safe flight, the flight process would be consisted of ascending, hovering and descending three parts, total 2 for heading and return. When the drone reaches the hovering height, there are many ways to design the trajectory. The most efficient way would be just fly in the xy plane in a straight line to the final position and descend. This would need xy direction to move interactive, or the drone first yaw to align the body x to final position. Both way would incorporate some interaction error into system, and thus enlarge the existing small error to an unsaveable large error. To prove this, I have implemented the yaw scheme, the result could be obtained in Figure 6. While before xy hovering, the drone ascend to 50 m and yaw to the required angles as planed, but there is a abnormal value to the angular velocity. After the heading process, x and y position have been differed from planed value and the final velocity of both position and angle are not zero, this would leads to huge error for Heuristic method without PDI control according to [1].

Due to time limit, and the unseenable typo in the paper(much time has been put on debuging the code while it turns out there are several serious typo in the paper), I just implement another more direct scheme to fulfill the delivering task.

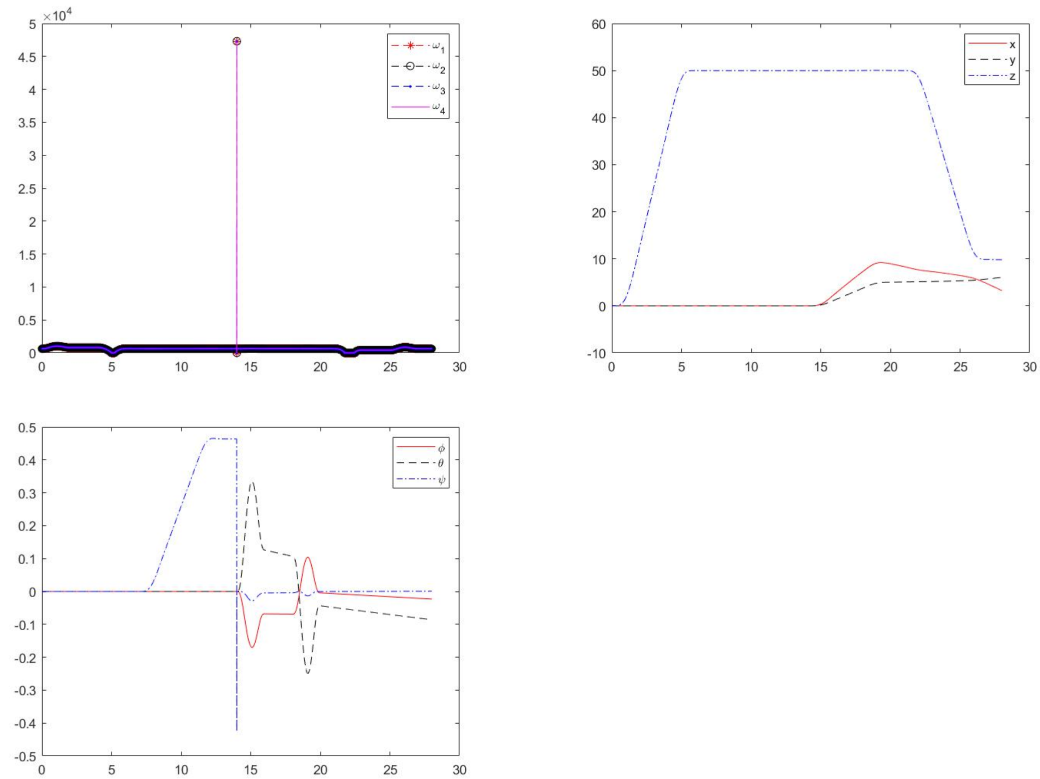
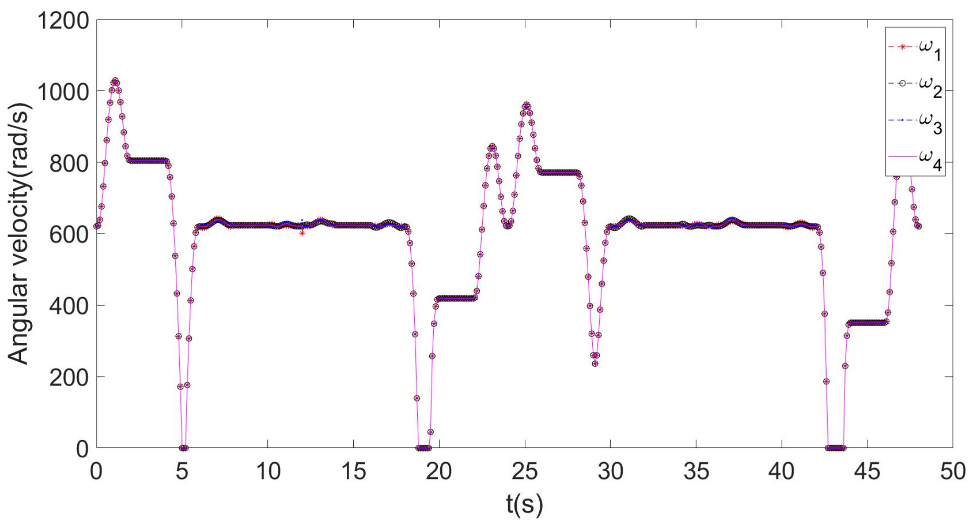


Figure 6: Yaw scheme, from left to right to bottom: angular velocity(rad/s), position(m) and angles(degree). x coordinate is t in seconds.

The other way of doing this hovering process is just first pitch to final x position and then roll to final y position. This scheme turns out to be really robust and the error is unaccountable 0. With this, the total process of the change of angular velocity, position, and angles can be seen in Figure 7. The simulation consist if the heading and returning process didn't consider the change of weight and momentum due to time limit.

From initial position $P_0(x_0,y_0,z_0)$, the quadcopter first ascend to hovering height(=50 m) within the first 6s. Then the pitch angle increase to about 19 degree, causing it to move to x_1(=10 m) in x direction for next 6s. Next, the drone fly to y_1(=10 m) position with corresponding row angle changing. After this, it descends to dropping height 10 m. For each change of flying direction, the quadcopter would have a stabilizing time with about 1s . After about 24s, the quadcopter begins to return to P_0 with a kind of symmetric process.

Further test with different final position, hovering and dropping height has been carried out. The error would be still unaccountable when these value smaller than 200m. While the height can be calculated up to 500m. If the distance range is too large, there would be error and to eliminate this, a smaller dt would be needed and the code code would take longer time be run.



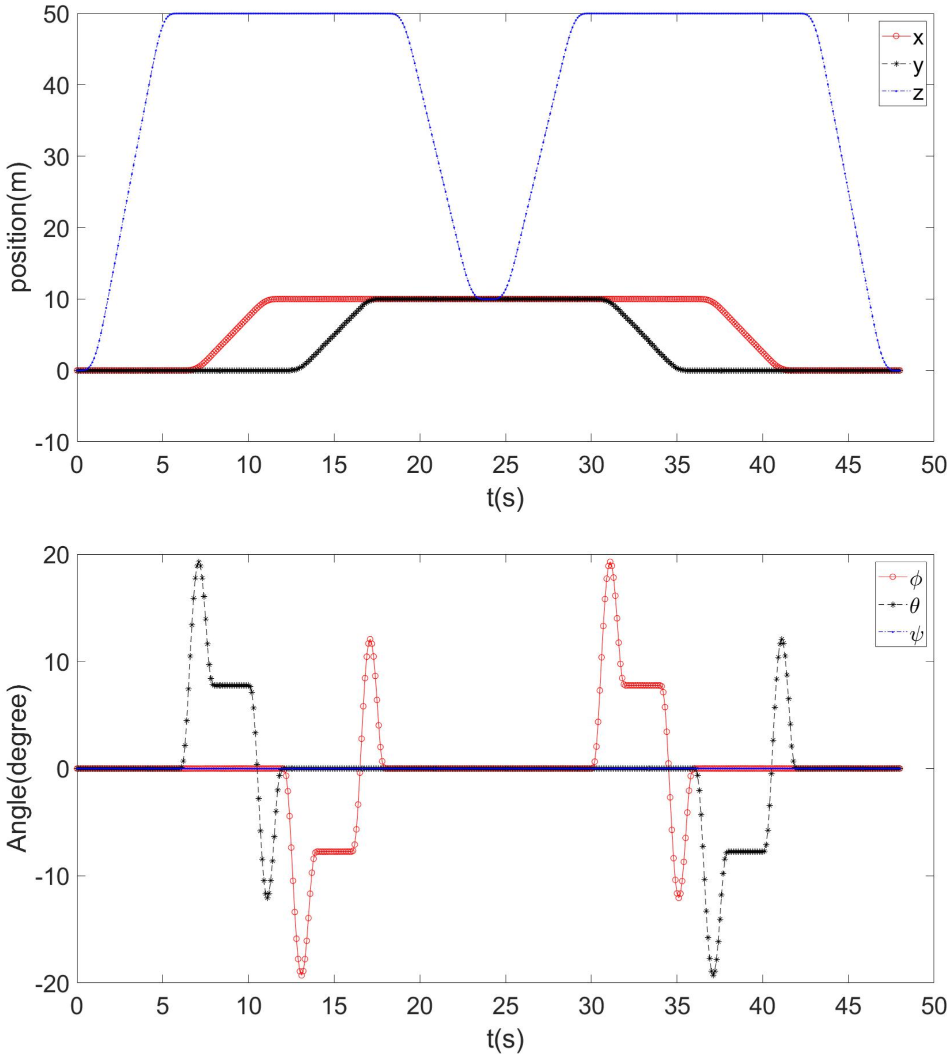


Figure 7: Quadcopter delivering trajectory from direct scheme

4. Conclusion

With the guidance from Luukkonen's paper [1], the forward kinematics & dynamics, a backward trajectory control of quadcopter have been implemented in this project. The final trajectory result satisfied the heading and returning process of a delivering task with unaccountable error if dt is sufficient small. Further work could be done by adding in the PDI controller scheme to make the the yaw process more stable. Also the dropping process, more realistic time consumed by different distance range etc. could also be done.

Reference

[1] Luukkonen, Teppo. "Modelling and control of quadcopter." *Independent research project in applied mathematics, Espoo* 22 (2011).

[2] Zuo, Zongyu. "Trajectory tracking control design with command-filtered compensation for a quadrotor." IET control theory & applications 4.11 (2010): 2343-2355.

[3] Raffo, Guilherme V., Manuel G. Ortega, and Francisco R. Rubio. "An integral predictive/nonlinear H∞ control structure for a quadrotor helicopter." Automatica 46.1 (2010): 29-39.

[4] Bouadi, H., and M. Tadjine. "Nonlinear observer design and sliding mode control of four rotors helicopter." International Journal of Mechanical, Aerospace, Industrial, Mechatronic and Manufacturing Engineering 1.7 (2007): 354-359.