TRIBHUVAN UNIVERSITY
INSTITUTE OF ENGINEERING

AUTONOMOUS TRAJECTORY GENERATION OF UNMANNED AERIAL VEHICLE USING MODEL PREDICTIVE CONTROL

By:
Nitesh Subedi
Prajwal Koirala

A PROJECT
SUBMITTED TO THE DELTA 2.0 FOR MODEL BASED DESIGN COMPETITION IN MATLAB

July, 2021

1 Abstract

Kinematic model of an unmanned aerial vehicle is derived and modeled on MATLAB. Constraints and design parameters of the vehicle are referenced from GeoSurv II UAV project. For autonomous trajectory generation, a cost function is formulated such that its minimization ensures our trajectory requirements. The optimal control problem is then transcribed to Non-linear programming problem which is solved using interior point solver on MATLAB. After every finite time, the trajectory is regenerated using real time information updates. Finally, the simulation is visualized using MATLAB plots and MATLAB toolbox called vrealm.

2 Problem Description

One of the key problems in this field is the autonomous generation of an optimal trajectory for an aerial vehicle. Path Planning in general is a difficult task, especially when considering obstacles. Generally, the main issues that are to be faced while developing a autonomous vehicle exist at three different levels-perception, planning and control. The

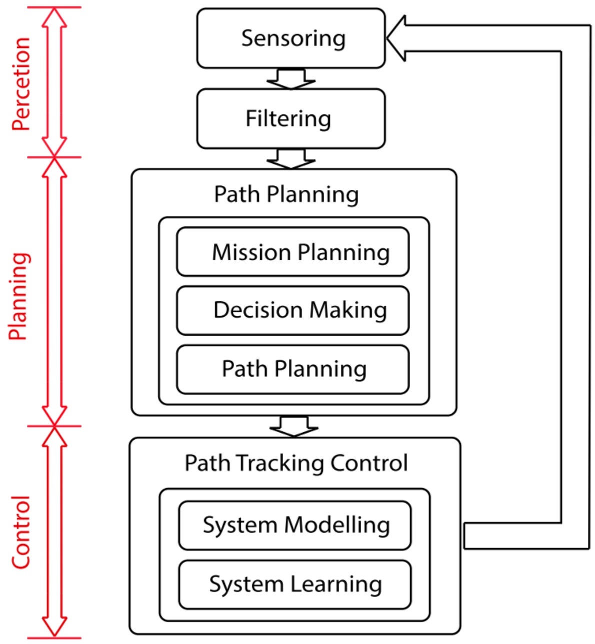


Figure 2.1: Autonomous control architecture at different levels

perception level consists of sensing and filtering the data which may be the surrounding data with one's current location. This system consists of number of sensors which provides data and filter denoises it giving the reasonable estimate for the unmeasureable states. The planning level consists of mainly three tasks, which include mission planning, where the vehicle solves a routing problem in order to complete a task, decision making, where the vehicle chooses an appropriate action for the next time step from an available action set, and path planning, where the vehicle plans its future trajectory as a function of space or time. Finally, the control level receives the signals from the planning level, maintains the stability of the vehicle, and tracks the desired path. All of these levels should operate simultaneously in a control system of a vehicle to operate autonomously. In our project, we try to simulate all three levels with no filtering in the perception level assuming we can get the exact position and all the states are accurately measureable. The algorithm used for the path planning is called Model Predictive Control. A cost function is designed based on the requirements and it is then minimized subjected to constraints of vehicle kinematics, and restrictions in states as well as control inputs. As this is highly non linear system, we use non-linear programming solver called interior point solver (IPOPT) to solve the resulting problem with the help of CasADi framework on MATLAB [2].

3 Model Details

Kinematics of UAV

Kinematic model of the UAV is derived and used for its path planning in this project. Let x, y and z denote the position of the UAV in Cartesian coordinates. γ represents the climb angle (pitch) and ζ represents the heading angle (yaw). Also, v is the heading velocity of the vehicle.

$\dot{x} = vcos(\gamma)cos(\zeta)$

$\dot{y} = vcos(\gamma)sin(\zeta)$

$\dot{z} = vsin(\gamma)$

$\dot{\gamma} = \omega_\gamma$

$\dot{\zeta} = \omega_\zeta$

$\dot{v} = a$

The state vector comprises x, y, z, γ, ζ and v. And the control vector U comprises the control variables ω_γ, ω_ζ and acceleration a. The state space equations can be written as:

$\dot{X} = f(X, U)$

where,

$X = [x, y, z, \gamma, \zeta, v]^T$

$U = [\omega_\gamma, \omega_\zeta, a]^T$

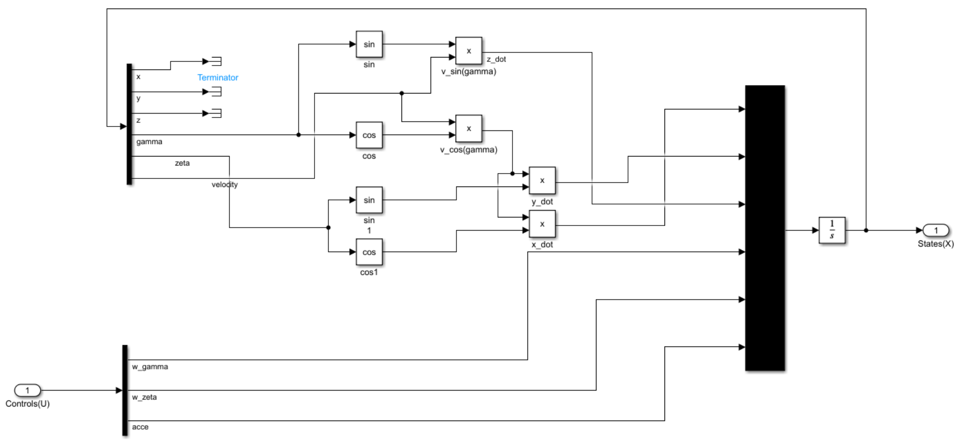


Figure 3.1: Simulink Model of the vehicle kinematics

Cost function

The cost function is a Performance Measure designed such that its minimization ensures a desirable performance of the system [4]. Cost function (J) is designed for following set of problem scenario:

a. The goal location of our UAV is $x(t = t_f) = x_f$.

b. The UAV should not deviate far away from the reference track $y(t) = 0$.

c. The UAV should try to maintain the height of $z(t) = z_f$.

d. The UAV should avoid the obstacle $ln\{h(x, y, z) = (x - x_c)^2/r^2 + (y - y_c)^2/r^2\} = 0$. i.e $r(x, y, x) = e^{\alpha * e^{-h}}$ is to be minimized.

Following cost function is designed to incorporate above objectives:

$$Minimize : J = \int_{t_0}^{t_f} \{c_1[x(t) - x_f]^2 + c_2[y(t)]^2 + c_3[z(t) - z_f] + r(x, y, z)\}dt.$$

Constraints

$\dot{X} = f(X, U)$

$X(t = t_0) = X_0$

$v_{min} \leq v \leq v_{max}$

$a_{min} \leq a \leq a_{max}$

$\omega_{\zeta_{min}} \leq \omega_{\zeta} \leq \omega_{\zeta_{max}}$

$\omega_{\gamma_{min}} \leq \omega_{\gamma} \leq \omega_{\gamma_{max}}$

$\gamma_{min} \leq \gamma \leq \gamma_{max}$

$\zeta_{min} \leq \zeta \leq \zeta_{max}$

$z_{min} \leq z \leq z_{max}$

$h > 0$

Discretization

The optimal control problem is transcribed into a non linear programming problem by discretizing the objective function and the constraints [3]. The values for different unknown terms in the constraints are specific to GeoSurv II UAV [5, 1]. In our case, the following nlp is solved to obtain the optimal state-control pair:

$$Minimize : J = \sum_{k=1}^{N} \{0.02 * [x_k - x_f]^2 + 0.3 * y_k^2 + 0.2 * [z_k - z_f]^2 + r(x_k, y_k, z_k)\}.$$

Subject to:

$X_1 = X_0$

for k = 1:N

$X_{k+1} = X_k + f(X_k, U_k) * \Delta t$

$15 \leq v_k \leq 30$

$-3 \leq a_k \leq 3$

$-0.5 \leq \omega_{\zeta_k} \leq 0.5$

$0.5 \leq \omega_{\gamma_k} \leq 0.5$

$-\pi/4 \leq \gamma_k \leq \pi/4$

$-\pi \leq \zeta_k \leq \pi$

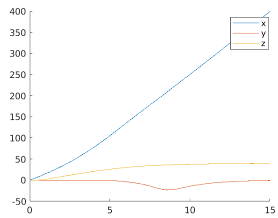
$0 \leq z_k \leq 100$
$h > 0$

Model Predictive Control

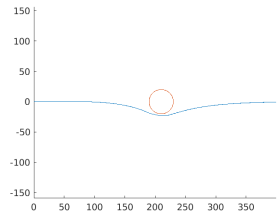
1. Trajectory generation: Solve the above nlp for next 5 seconds $t = t_0$ to $(t_0 + 5)$
2. Follow the generated trajectory for next 0.5 seconds.
3. Update current X as X_0 and the current t as t_0.
4. Repeat step 1.

4 Results

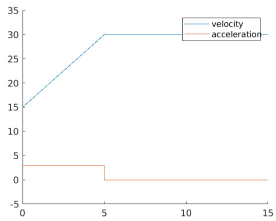
The model is then simulated on MATLAB and graphs for different states and controls are plotted along with the trajectory. It can be quickly observed that the trajectory followed by the UAV is well within the constraints and demonstrate manoeuvres like take-off and obstacle avoidance. Further, a MATLAB toolbox named 'vrealm' is used for the 3D animation and visualization of the trajectory in presence of obstacles.



(a) Position of UAV vs Time

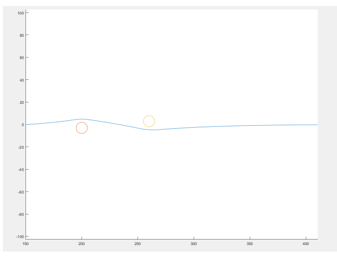


(b) Path of UAV projected on XY plane

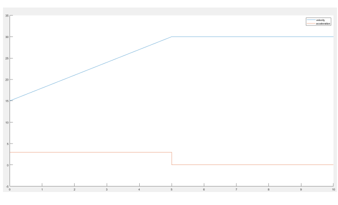


(c) Velocity and acceleration of the UAV

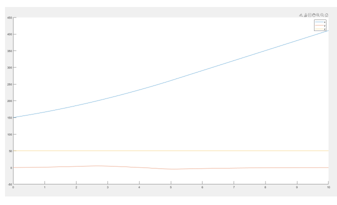
Figure 4.1: Vehicle Trajectory with single obstacle and take-off maneuver



(a) Vehicle x,y location



(b) Velocity and Acceleration



(c) x,y,z plots

Figure 4.2: Vehicle Trajectory with multiple obstacles with constant height

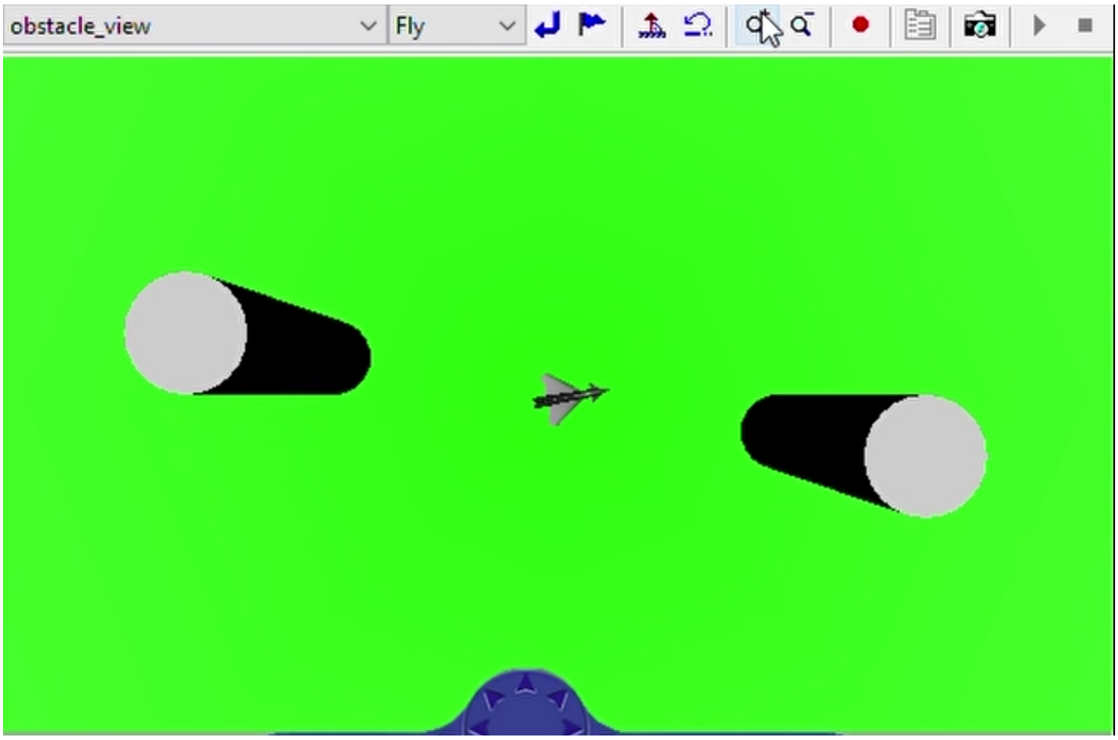


Figure 4.3: Visualization with vrealm

References

[1] Min Prasad Adhikari and B Eng. *Real-time autonomous collision avoidance for unmanned aerial vehicles.* PhD thesis, MS thesis, Dept. of Aerospace Engineering, Ryerson Univ, 2015.

[2] Joel A E Andersson, Joris Gillis, Greg Horn, James B Rawlings, and Moritz Diehl. CasADi – A software framework for nonlinear optimization and optimal control. *Mathematical Programming Computation*, 11(1):1–36, 2019.

[3] Matthew P Kelly. Transcription methods for trajectory optimization: a beginners tutorial. *arXiv preprint arXiv:1707.00284*, 2017.

[4] Donald E Kirk. *Optimal control theory: an introduction.* Courier Corporation, 2004.

[5] Shahaboddin Owlia. *Real-time autonomous obstacle avoidance for low-altitude fixed-wing aircraft.* PhD thesis, Carleton University, 2013.