SMOOTHED HEX-GRID TRAJECTORY PLANNING USING HELICOPTER DYNAMICS

Lukáš Chrpa and Antonín Komenda
Agent Technology Center, Faculty of Electrical Engineering, Czech Technical University in Prague
Prague, Czech Republic

Keywords: Path planning, Planning on grids, Model of dynamics, VTOLs, UAVs.

Abstract: Considering Unmanned Autonomous Vehicles (UAVs) the planning tasks mainly consist of finding paths between given waypoints with respect to given constraints. In this paper we developed a path planning system for flying UAVs (VTOLs and CTOLs) built upon Hexagonal grids which also supports simple dynamics (handling with speed). The planning system is additionally supported by a trajectory smoothing mechanism based on a dynamics model of a helicopter. The model can be also used for the simulation of the helicopter movement.

1 INTRODUCTION

While controlling UAVs[1] we deal mainly with path (trajectory) planning(Wang and Botea, 2009; Bottasso et al., 2008; Meister et al., 2009). VTOLs[2], we will mainly focus on here, are usually helicopter-like assets. Contrary to CTOLs[3] (aircraft-like assets) VTOLs can change their direction or altitude without changing *x* or *y* coordinate. We should mention a state-of-the-art system for air traffic control AgentFly (Šišlák et al., 2008). Plans (paths) provided by the AgentFly are represented by sequences of maneuvers (unit movements). Another state-of-the-art system deals with path planning for VTOLs (Wzorek and Doherty, 2006). In the system, there is used a path planning algorithm based on Probabilistic Roadmaps (Kavraki et al., 1996) which results in a sequence of waypoints interpolated by splines.

In this paper, we introduce our path planning system for VTOLs (its modification can be used also for CTOLs). The planner is built upon Hexagonal grids, A* algorithm (Hart et al., 1968) is used as a planning procedure. The planner supports simple dynamics incorporating speed constraints which are, for instance, good for detecting whether a certain turn maneuver is applicable. Using grids instead of irregular (but more realistic) maneuvers used in AgentFly brings a significant increase of performance because the node comparisons (frequently done during the planning process) is much more easier for grids. Using 'spline planner' does not seem to be possible while taking into account speed limits, because in every point the curvature must be somehow limited depending on the previous part of the spline curve.

Methods for trajectory smoothing and optimal trajectory planning are covered by field of control theory. However, there is a lack of the state-of-the-art works merging classical grid or symbolic planning and trajectory smoothing based on the dynamics of the various vehicles. Method for real-time trajectory smoothing is proposed by (Anderson et al., 2005). Spatial waypoints without speed constraints are input of the method and the output is a generated trajectory for the vehicle. An approach sharing an idea of the planning maneuvers is presented in (Bottasso et al., 2008). The trajectory smoothing process uses motion primitives which are an abstraction of the movement maneuvers.

[1]Unmanned Autonomous Vehicles
[2]Vertical Take-Off and Landing UAVs
[3]Classical Take-Off and Landing UAVs

2 PLANNING FOR VTOLS

Our **basic model**, simply, deals with hex-grid path planning. Since the hexagonal lattice is only 2D, we must consider flight levels. Then, the discretization of the space forms 'honeycomb'.

The *planning problem* deals with finding a path from the starting to ending waypoint, following the directions (if defined) and avoiding the No-flight-zones (NFZs). Let us define a set of primitive ma-

Chrpa L. and Komenda A..
SMOOTHED HEX-GRID TRAJECTORY PLANNING USING HELICOPTER DYNAMICS.
DOI: 10.5220/0003184106290632
In *Proceedings of the 3rd International Conference on Agents and Artificial Intelligence* (ICAART-2011), pages 629-632
ISBN: 978-989-8425-40-9

neuvers (we allow ‘diagonal‘ moves):

Straight — Moves to adjacent (also ‘diagonally adjacent) hexagon according to direction.

Turn — Changes the direction (to the left or right) by an angle $\pi/6$, $\pi/3$ and $\pi/2$.

Pitch — Moves to adjacent upper (resp. lower) flight level.

Additionally, there are several restriction that we applied for plan generation. Every turn maneuver must be followed by the straight maneuver. Every pitch maneuver must be applied between two straight maneuvers. Planning procedure is done by A* algorithm, where the actual distance from the starting waypoint *g* and the estimated cost to the ending waypoint (heuristic) *h* are computed in every open node. The heuristic *h* is computed as Vancouver distance (Yap, 2002) (analog to Manhattan distance). If ‘diagonal‘ moves are allowed, then the heuristic *h* is inadmissible but the optimality of the plans is affected very slightly.

We introduce our **model with simple dynamics** where we have to take into account a new parameter - speed, since the basic model works well only for UAVs that move slowly.

We extend the *planning problem* by a basic speed model, where we define maximal acceleration and deceleration, minimal and maximal speeds with respect to particular maneuvers. From the physics it is well known that a minimal turn radius grows quadratically as a speed grows. We also can define an initial speed (in the starting waypoint) and voluntarily a target speed interval (in the ending waypoint).

Algorithm 1: Computing a speed interval for a newly opened node.

```
{Upper indices − (resp. +) stand for minimal (resp. maximal) speed values.}
if maneuver is the Straight maneuver then
    $v^-_{curr} := MAX\left(v^-, \sqrt{(v^-_{prev})^2 - 2as}\right)$
    $v^+_{curr} := MIN\left(v^+, \sqrt{(v^+_{prev})^2 + 2as}\right)$
end if
if maneuver is the Turn or Pitch maneuver then
    if $\langle v^-_{prev}, v^+_{prev} \rangle \cap \langle v^-_{man}, v^+_{man} \rangle = \emptyset$ then
        return null {Speed constraint cannot be satisfied for the Turn or Pitch maneuver}
    end if
    $v^-_{curr} := MAX(v^-_{prev}, v^-_{man})$
    $v^+_{curr} := MIN(v^+_{prev}, v^+_{man})$
end if
return $v^-_{curr}, v^+_{curr}$
```

Considering speed as a new dimension might result in an explosion of the state space. To avoid this we introduce speed intervals that represent a range of feasible speed values in every explored node. Speed can be changed only during the Straight maneuver according to the speed model. During the Turn or Pitch maneuver the speed remain constant. Any speed limitation (for the Turn or Pitch maneuver) can be easily checked, i.e., if a non-empty subset of the particular speed interval satisfy the particular limits. For the formal description, see alg. 1. We have to back-propagate speed changes (see alg. 2) to follow the dynamics, for instance, the VTOL must begin to slow down at some distance before performing the Turn maneuver.

A feasible plan consists, like in the basic model, a sequence of points and directions (in these points). Maximal values of speed intervals in every point can be set as (exact) speed values. Then, we can simply compute time-stamps in all the points.

Algorithm 2: Propagation of speed intervals.

```
{Upper indices − (resp. +) stand for minimal (resp. maximal) speed values.}
$count_{min} := count_{max} := 0$
$minspd := v^-_{curr}$; $maxspd := v^+_{curr}$
Select the predecessor node as a current node
while $v^-_{curr} < minspd \lor v^+_{curr} > maxspd$ do
    Push the current node into Stack
    if maneuver is not the Turn or Pitch maneuver then
        $maxspd = MIN\left(v^+_{curr}, \sqrt{maxspd^2 + 2as}\right)$
        if $maxspd \neq v^+_{curr}$ then
            $count_{max}++$
        end if
        $minspd = MAX\left(v^-_{curr}, \sqrt{minspd^2 - 2as}\right)$
        if $minspd \neq v^-_{curr}$ then
            $count_{min}++$
        end if
    end if
    Select the predecessor node as a current node
end while
while Stack is not empty do
    Pop the element from the stack as a current node
    if maneuver is not the Turn or Pitch maneuver then
        if $count_{max} > count_{min}$ then
            $maxspd := \sqrt{maxspd^2 - 2as}$; $count_{max}--$
        else if $count_{max} < count_{min}$ then
            $minspd := \sqrt{minspd^2 + 2as}$; $count_{min}--$
        else
            $maxspd := \sqrt{maxspd^2 - 2as}$; $count_{max}--$
            $minspd := \sqrt{minspd^2 + 2as}$; $count_{min}--$
        end if
    end if
    $v^+_{curr} := MIN(v^+_{curr}, maxspd)$
    $v^-_{curr} := MAX(v^-_{curr}, minspd)$
end while
```

3 TRAJECTORY SMOOTHING USING A MODEL OF HELICOPTER DYNAMICS

The model of helicopter dynamics serves two purposes. Firstly, it is used as a smoothing method for the resulted plans from the hex-grid planner (see Section 2). Secondly, it is used during the simulation of the helicopter movement. These two cases differ in two main aspects. The smoothing process runs the model using larger time steps (improving the computational duration). On the other hand, the simulation phase may include additional errors in the movement caused by various real-world effects (e.g. wind, imperfections of the asset hardware, sensory errors and glitches, biased control and others).

We consider 6 DOF[4] model consisting of three spatial and three rotational axes. The spatial position is described by translation vector $\mathbf{p} = (x,y,z)$, and the rotation is described by three euler angles φ, θ, ψ in a quaternion form $\mathbf{q} = (q_0, q_1, q_2, q_3)^T$.

Furthermore, both the spatial and the rotational axes are completed by their first and second derivates, i.e., velocities and accelerations. Spatial velocity and acceleration can be simply derived as follows:

$$\frac{d\mathbf{p}}{dt} = \mathbf{v},\ \ \mathbf{p} = \int \mathbf{v}dt,\ \ \mathbf{p} = \mathbf{p_0} + \mathbf{v}t.$$

The spatial acceleration respects the same pattern.

The first derivate of the rotation (angular velocity $\boldsymbol{\omega}$) described in a quaternion can be derived using differential equation for varying quaternion with angular velocity described in an augmented quaternion: $\overline{\boldsymbol{\omega}}(t) = (0, \omega_1(t), \omega_2(t), \omega_3(t))^T$ as follows:

$$\frac{d\mathbf{q}}{dt} = \frac{1}{2}\mathbf{q} * \overline{\boldsymbol{\omega}},\ \mathbf{q} = \int \frac{1}{2}\mathbf{q} * \overline{\boldsymbol{\omega}},\ \mathbf{q} = \mathbf{q}_0 \exp\left(\frac{1}{2}\overline{\boldsymbol{\omega}}t\right).$$

The second derivate (angular acceleration $\boldsymbol{\alpha}$) is based on a vector derivation of the augmented quaternion describing the angular velocity $\boldsymbol{\omega}$.

The introduced equations describe only dynamics of a mass object in space. Following formulas describe a simplified physical model of a helicopter based on the air lift formula $L(\eta) = \frac{1}{2}\rho v^2 S C_L(\eta)$, where $L[N]$ is the lift force, $\rho[{}^{kg}/{}_{m^3}]$ is air density, $v[{}^{m}/{}_{s}]$ is velocity of the rotor disk causing the lift force, $S[m^2]$ is area of the rotor disk, and $C_L(\eta)$ is coefficient of lift. The lift coefficient C_L is a function of the attack angle η of the rotor blades and it is for the purposes of the model linearized by partially linear function with an ascending part with coefficient k_1 and a descending part with coefficient k_2:

$$C_L(\eta) = \begin{cases} |\eta| < \frac{1}{6}\pi: & \eta k_1, \\ |\eta| \geq \frac{1}{6}\pi: & (\eta - \frac{1}{6}\pi)k_2 + \frac{1}{6}\pi k_1. \end{cases}$$

The spatial acceleration of the object with a weight m is affected by two forces (i) gravity, and (ii) the lift force of the main rotor which is parameterized by the *collective rotor tilt* ν and computed as $\overline{\mathbf{a}} = (0,0,0,-g)^T + \mathbf{q}(0,0,0,\frac{L(\nu)}{m})^T \mathbf{q}^*$. The augmented vector $\overline{\mathbf{a}}$ is defined as $\overline{\mathbf{a}} = (0, \mathbf{a})^T$, $\mathbf{q}^*$ represents an inversion of a quaternion $\mathbf{q}$, and $\mathbf{qpq}^*$ describes rotation of augmented vector $\mathbf{p}$ by a quaternion $\mathbf{q}$.

The angular acceleration is affected by *cyclic rotor tilt* in two dimensions σ_x, σ_y and *anti-torque tail rotor tilt* τ in a following ways $\boldsymbol{\alpha} = (\frac{2L_r(\sigma_x)}{mr}, \frac{2L_r(\sigma_y)}{mr}, 0) + (0, 0, \frac{L_t(\tau)}{mr_t})$. The radius of the main rotor is denoted as r, distance of the tail rotor from the main rotor axis is denoted as r_t. The lift force of one half of the main rotor used by the cyclic tilt is denoted as L_r and the lift force caused by the tail rotor is denoted as L_t. The parameters for the particular lift formulas differs in area of the rotor S and mean velocity of the rotor v. The caused lift force of the rotor cyclic is positioned in the center of the rotor radius $\frac{r}{2}$.

3.1 Model Stabilization

The only explicit stabilization is used for horizontal stabilization of the model. The stabilizers ensure the model will always have tendency to stay horizontally even. The stabilizers are two PID[5] controllers. Each controller stabilizes the model in one axis, i.e. roll φ and pitch θ. The target state of the model rotation is zero angles of rotation. The regulation equations are based on the equation of ideal PID controller and they are discretized and used in iterative manner.

3.2 Movement Regulation

Movement of the model in the main four axes (spatial axes x, y, z and rotational axis ψ) is handled by four movement PID controllers. On the contrary of the stabilization controllers, the movement controllers are designed to fulfill requested height, velocity, direction, and yaw rotation.

The first regulated value is the height z of the model to target height z_t. It is based on PID controller. The affected action control is the collective rotor tilt ν, i.e., strength of the main rotor for climbing/descending.

[4]Degrees of freedom

[5]proportional–integral–derivative

The second two controllers affect the rotor cyclic providing horizontal movement of the helicopter. The formulas are similar to the equation of the height controller and differs in the used parameters and the error definition. The σ_x collective tilt is regulated by error value $(v_{tx} - v_x)$ and σ_y by error value $(v_{ty} - v_y)$. The used parameters are P_v, I_v, D_v. The current velocity $\mathbf{v}$ is regulated towards the target velocity $\mathbf{v_t}$ in the required direction to the target point (typically a trajectory waypoint). The waypoints are switched after reaching a predefined distance to the next waypoint.

The fourth controller is used for pointing of the head of the model towards the target point. Although the model can reach any position using only the previous three spatial axes, the yaw rotation is important if the helicopter carries an orientation dependent sensor (e.g. a range finder, or a camera). The torque controller is also PID with different identified constants.

3.3 Trajectory Smoothing using the Model

The model used for the smoothing of the trajectory produced by the hex-grid planner (see Section 2) uses all the introduced controllers together with the model of helicopter dynamics. The plan begins at ground and increases the height together with forward movement towards the target waypoint. During the flight a NFZ is avoided provided that the hex-grid planner planned the waypoint around the zone.

All the parameters of the controllers (P_s, P_z, P_v, P_r, $I_s, \dots, D_r$) were identified using the Ziegler-Nichols method. The model parameters including weight, rotor area, distance to the tail rotor and others are based on the Skeldar 200 VTOL from SAAB Aerosystems.

4 CONCLUSIONS

We proposed a path planner based on hexagonal grids for VTOLs supported by a trajectory smoothing mechanism based on a dynamics model of a helicopter. The planner can also handle itself simple dynamics considering a simple speed model, where we define certain constraints. The planner-based dynamics is reflected by the smoothing mechanism generating a trajectory usable by a physical model of a helicopter. The computational complexity of the smoothing method is linear as the process is based on integration with a constant step over the space.

For the future we should study how to add a time model, now we can only compute time intervals (according to speed intervals) in every point of the trajectory. It could be formulated as a constraint satisfaction problem and run as a postprocessing task on the found trajectory. We should also investigate another problem - collision avoidance in multi-agent systems. It is necessary to investigate the possibilities of efficient replanning while some collision threatens.

ACKNOWLEDGEMENTS

This work was supported by Czech Ministry of Education, Youth and Sports under Grant MSM6840770038 and by U.S. Army Grant W911NF-08-1-0521 and W15P7T-05-R-P209 under Contract BAA8020902.A.

REFERENCES

Anderson, E. P., Beard, R. W., and McLain, T. W. (2005). Real time dynamic trajectory smoothing for uninhabited aerial vehicles. *IEEE Transactions on Control Systems Technology*, 13(3):471–477.

Bottasso, C. L., Leonello, D., and Savini, B. (2008). Path planning for autonomous vehicles by trajectory smoothing using motion primitives. *IEEE Transactions on Control Systems Technology*, 16(6):1152–1168.

Hart, P., Nilsson, N., and Raphael, B. (1968). A formal basis for the heuristic determination of minimum cost paths. *IEEE Transactions on Systems Science and Cybernetics*, 4(2):100 –107.

Kavraki, L. E., Svestka, P., Latombe, J. C., and Overmars, M. H. (1996). Probabilistic roadmaps for path planning in high-dimensional configuration spaces. *IEEE Transactions on Robotics and Automation*, 12:566–580.

Meister, O., Frietsch, N., Ascher, C., and Trommer, G. F. (2009). Adaptive path planning for vtol-uavs. *Aerospace and Electronic Systems Magazine, IEEE*, 24(7):36–41.

Šišlák, D., Pěchouček, M., Volf, P., Pavlíček, D., Samek, J., Mařík, V., and Losiewicz, P. (2008). *Defense Industry Applications of Autonomous Agents and Multi-Agent Systems*, chapter AGENTFLY: Towards Multi-Agent Technology in Free Flight Air Traffic Control, pages 73–97. Birkhauser Verlag.

Wang, K.-H. C. and Botea, A. (2009). Tractable multi-agent path planning on grid maps. In *Proceedings of IJCAI*, pages 1870–1875.

Wzorek, M. and Doherty, P. (2006). Reconfigurable path planning for an autonomous unmanned aerial vehicle. In *Proceedings of ICAPS*, pages 438–441.

Yap, P. (2002). Grid-based path-finding. In *Canadian Conference on AI*, pages 44–55.