Game of Drones Report: Team Dédale

Charbel Toumieh

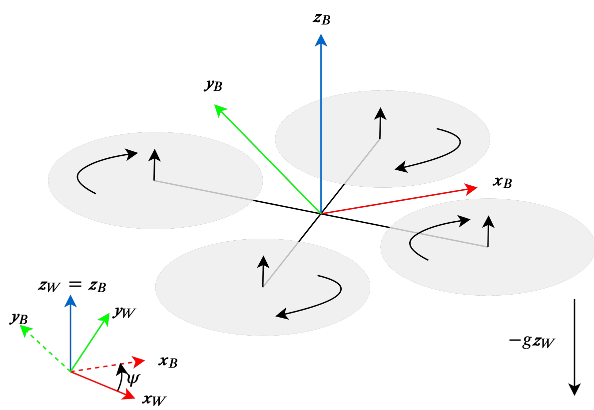

Fig. 1. Schematics of the multirotor model with the used coordinate systems.

***Abstract*— This is an extremely brief description of the methods used by team Dédale, a participant in the Game of Drones competition. We describe the control method used, the planning method used for each tier as well as the gate localization scheme for tiers 2 and 3.**

I. MAV MODEL

TABLE I. Nomenclature

g	gravity
m	multirotor mass
$\boldsymbol{p}$	position vector x, y, z
$\boldsymbol{v}$	velocity vector v_x, v_y, v_z
$\boldsymbol{z}_W$	world frame z
$\boldsymbol{z}_B$	body frame z
$\boldsymbol{R}$	rotation matrix from body to world frame
$\boldsymbol{D}$	drag matrix
ϕ	roll angle
θ	pitch angle
ψ	yaw angle
c_{cmd}	total thrust command
$\|\|.\|\|_2$	euclidean norm

We assume a low level controller allows for controlling the attitude and thrust. The equations of motion are:

$$\dot{\boldsymbol{p}} = \boldsymbol{v} \tag{1}$$

$$\dot{\boldsymbol{v}} = -g\boldsymbol{z}_W + \frac{c_{cmd}}{m}\boldsymbol{z}_B - \boldsymbol{R}\boldsymbol{D}\boldsymbol{R'}\boldsymbol{v}||\boldsymbol{v}||_2 \tag{2}$$

$$\dot{\phi} = \dot{\phi}_{cmd} \tag{3}$$

$$\dot{\theta} = \dot{\theta}_{cmd} \tag{4}$$

$$\dot{\psi} = \dot{\psi}_{cmd} \tag{5}$$

the author is with the Laboratoire de Recherche en Informatique, Université Paris Sud, Université Paris Saclay, France `https://www.lri.fr`

II. CONTROLLER DESIGN

We control our quadrotor using a nonlinear MPC [1], with the ACADOS toolkit [2]. The MPC minimizes the cost function:

$$J = \int_{t=0}^{T} ||\boldsymbol{x}(t) - \boldsymbol{x}_{ref}(t)||^2_{\boldsymbol{Q}_x} + ||\boldsymbol{u}(t) - \boldsymbol{u}_{ref}(t)||^2_{\boldsymbol{R}_u} dt + ||\boldsymbol{x}(T) - \boldsymbol{x}_{ref}(T)||^2_{\boldsymbol{P}} \tag{6}$$

We use the model described in section I with $\boldsymbol{u} = [c_{cmd}\ \dot{\phi}_{cmd}\ \dot{\theta}_{cmd}\ \dot{\psi}_{cmd}]^T$ and $\boldsymbol{x} = [\boldsymbol{p}\ \boldsymbol{v}\ \phi\ \theta\ \psi]^T$. The sampling time is $h = 0.05s$ and the horizon $N_h = 10$ which gives $T = 0.5s$. The weights are:

$$\boldsymbol{P} = \boldsymbol{Q}_x = diag(10, 10, 10, 0.1, 0.1, 0.1, 0, 0, 0.01) \tag{7}$$

$$\boldsymbol{R}_u = diag(0, 0.05, 0.05, 0.05) \tag{8}$$

All parameters are set by approximation/experimentation and may not be optimal. We limit $|\phi| \le 85\ deg$, $|\theta| \le 85\ deg$, $|\dot{\phi}_{cmd}| \le 120\ deg/s$, $|\dot{\theta}_{cmd}| \le 120\ deg/s$ and $|\dot{\psi}_{cmd}| \le 60\ deg/s$.

III. PLANNING METHOD

A. Tier 1

For tier 1, we use a new planning method developed by ourselves. It does not rely on waypoints, but on gate sizes so that it is more optimal i.e. the constraint is that the trajectory passes through the gate and not just a point. The method will determine which optimal point at the gate it will pass through. It uses non linear optimization and gives faster trajectories then the state of the art (at least 30%). It gives up to 3 times faster trajectories then polynomial methods described in [3], [4] and [5].

B. Tiers 2 and 3

We use a method inspired by the method described in [6]. We linearly interpolate between points before and after the gate with a constant speed and let the MPC follow the trajectory to the best of its ability (since it is not absolute feasible).

IV. GATE DETECTION

We use transfer learning with tensorflow (RCNN) to detect gates with a bounding box [7]. We are able to localize the gates given that we know their sizes beforehand (through PnP). However since some gate sizes and shapes vary dramatically, the performance of our method is subpar. We will need at least two images with two detections to calculate their scale. We did not implement this solution and took the fixed size as the one with the most compatible gates. In the future we plan to implement a semantic segmentation scheme [8] to make the localization more accurate.

V. Adversarial tactics

A. Tier 1

Since we are considerably faster then the state of the art, there is, for the most part, no need for adversarial tactics. Mainly we may need it at the start of the race. For this reason we use the following rule: when we are close to the opposing drone ($distance < 0.6$) and if it is closer then us to the next gate, we translate our reference trajectory 20 cm in the direction of the projection of $\boldsymbol{p}_1 - \boldsymbol{p}_2$ on $\boldsymbol{z}_W$ for 5 seconds, with $\boldsymbol{p}_1$ the position of our drone and $\boldsymbol{p}_2$ that of the opposing drone.

This assumes that the trajectory generated offline is at least 40 cm (20 cm for the drone size + 20 cm trajectory translation) away from any obstacle in the $\boldsymbol{z}_W$ direction (correct assumption). It also assumes that we will be ahead of the other drone after 5 seconds with no collision (may not hold).

B. Tier 3

We do not use any adversarial tactics as we did not train our network to detect the adversary drone.

References

[1] M. Kamel, M. Burri, and R. Siegwart, "Linear vs nonlinear mpc for trajectory tracking applied to rotary wing micro aerial vehicles," *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 3463–3469, 2017.

[2] R. Verschueren, G. Frison, D. Kouzoupis, N. van Duijkeren, A. Zanelli, R. Quirynen, and M. Diehl, "Towards a modular software package for embedded optimization," in *Proceedings of the IFAC Conference on Nonlinear Model Predictive Control (NMPC)*, 2018.

[3] C. Richter, A. Bry, and N. Roy, "Polynomial trajectory planning for aggressive quadrotor flight in dense indoor environments," in *Robotics Research*. Springer, 2016, pp. 649–666.

[4] D. Mellinger and V. Kumar, "Minimum snap trajectory generation and control for quadrotors," in *2011 IEEE International Conference on Robotics and Automation*. IEEE, 2011, pp. 2520–2525.

[5] M. Burri, H. Oleynikova, M. W. Achtelik, and R. Siegwart, "Real-time visual-inertial mapping, re-localization and planning onboard mavs in unknown environments," in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2015, pp. 1872–1878.

[6] E. Kaufmann, M. Gehrig, P. Foehn, R. Ranftl, A. Dosovitskiy, V. Koltun, and D. Scaramuzza, "Beauty and the beast: Optimal methods meet learning for drone racing," in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 690–696.

[7] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, *et al.*, "Tensorflow: A system for large-scale machine learning," in *12th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 16)*, 2016, pp. 265–283.

[8] H. Zhao, X. Qi, X. Shen, J. Shi, and J. Jia, "Icnet for real-time semantic segmentation on high-resolution images," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 405–420.