# A robust but easily implementable remote control for quadrotors: Experimental acrobatic flight tests

Maison Clouatre, Makhin Thitsa, Michel Fliess, Cédric Join

# A robust but easily implementable remote control for quadrotors: Experimental acrobatic flight tests

Maison Clouatre*, Makhin Thitsa*, Michel Fliess+†, Cédric Join¶†
*Department of Electrical and Computer Engineering, Mercer University
Macon GA 31207, USA
maison.roland.clouatre@live.mercer.edu, thitsa_m@mercer.edu
+LIX (CNRS, UMR 7161), École polytechnique
91128 Palaiseau, France
Michel.Fliess@polytechnique.edu
¶ CRAN (CNRS, UMR 7039)), Université de Lorraine
BP 239, 54506 Vandœuvre-lès-Nancy, France
cedric.join@univ-lorraine.fr
† AL.I.E.N., 7 rue Maurice Barrès, 54330 Vézelise, France
{michel.fliess, cedric.join}@alien-sas.com

***Abstract*— Experimental flight tests are reported about quadrotors UAVs via a recent model-free control (MFC) strategy, which is easily implementable. We show that it is possible to achieve acrobatic rate control of the UAV, which is beyond the previous standard. The same remote controller is tested on two physical vehicles without any re-tuning. It produces in both cases low tracking error. We show that MFC is robust even when the quadrotor is highly damaged. A video footage can be found at:** `https://youtu.be/wtSLalA4szc`

***Key Words*— Model-free control, intelligent controller, robustness, quadrotor.**

## I. Introduction

Quadrotors are today the most ubiquitous unmanned aerial vehicles (UAVs) (see, *e.g.*, [1]–[3], and the references therein). Many remote control strategies, including machine learning techniques, have been investigated (see, *e.g.*, [1]–[6], and the references therein). According to some authors (see, *e.g.*, [5], and the references therein) the simplicity of PIDs explains why they behave better in practice than more advanced controllers. Nevertheless the well-known shortcomings of PIDs (see, *e.g.*, [7], [8]) remain valid in this UAV context. This is why we suggest here *model-free control* (MFC) in the sense of [9] which might be viewed as an improvement of PIDs. MFC, which has been successfully applied to a number of concrete case-studies (see [9]–[11] and the references therein for most publications until the beginning of 2020), has already been employed several times for UAVs [12]–[18]. This communication might be the first one about experimental tests of MFC with quadrotors.

We are concerned here with the *acrobatic* flight[1] of quadrotors. This type of flight is important to tracking, evasion, and rescue missions, as well as obstacle avoidance in dense environments. We propose a fully MFC-based flight controller for robust rate tracking onboard quadrotor UAVs, which is an improvement beyond the previous standard. The proposed method is validated

- on-board various physical UAVs,
- via highly damaged vehicles.

In both cases

- no modification or re-tuning of the controller is needed,
- the UAVs are always well stabilized around the reference trajectory.

It highlights the method's robustness and data-driven nature. MFC requires little computational power to implement, thus it is an exciting low-cost alternative to PID that is used on most quadrotors today.

The paper is organized as follows. In Section II a short review of MFC theory is provided. Section III outlines the implementation and deployment of MFC on-board quadrotor UAVs. Section IV reports experimental flight data for various vehicles under MFC and compares the flight performance achieved to that of a PID, the most common control technique used on small UAVs. Conclusions and future works follow.

## II. Model-free control: A short review

### A. The ultra-local model for a SISO systelm

It has been proved in [9] that under quite weak assumptions any SISO system, with input $u$ and output $y$, may be well approximated by the *ultra-local* model

$$y^{(m)} = F(t) + \alpha u \tag{1}$$

where

- $m \geq 1$ is the derivation order,
- the time-varying quantity $F$ subsumes not only the unmodeled dynamics, but also the external disturbances,

[1]There is a legal definition for *acrobatic*, or *aerobatic*, flights: `https://www.law.cornell.edu/cfr/text/14/91.303` See, *e.g.*, [19], [20] for brillant and recent contributions.

- the constant $\alpha \in \mathbb{R}$ is such that the three quantities $y^{(m)}$, $F$, $\alpha u$ in Eq. (1) are of the same order of magnitude.

Note that

- the poorly known plant is not necessarily of order $m$: $y^{(\nu)}$, where $\nu > m$ may be sitting in $F$,
- in almost all the numerous concrete case-studies that were encountered until now, $m = 1$, with only some exceptions where $m = 2$,
- it is meaningless to try to estimate $\alpha$ precisely.

### B. MIMO systems

Consider a multi-input multi-output (MIMO) system with $p$ control variables $u_i$ and $p$ output variables $y_i$, $i = 1, \ldots, p$. It has been observed in [21] and confirmed by all encountered concrete case-studies (see, *e.g.*, [22]), that such a system may be regulated via $p$ monovariable ultra-local models:

$$y_i^{(m_i)} = F_i + \alpha_i u_i$$

where $F_i$ may also depend on $u_j$, $y_j$, and their derivatives, $j = 1, \ldots, p$.

*Remark 2.1:* In our example $p = 3$, one with a first (resp. second) order ultra-local model. Previous publications on UAVs [13]–[15] teach us that $m = 2$ is often appropriate (see also [23]–[26]) for other examples).[2]

### C. iP and iPD

*1) $m = 2$:* Eq. (1) becomes

$$\ddot{y} = F(t) + \alpha u. \tag{2}$$

Associate [9] to Eq. (2) the *intelligent Proportional-Derivative* controller, or *iPD*,

$$u = -\frac{\hat{F} - \ddot{y}_r + K_P e + K_D \dot{e}}{\alpha} \tag{3}$$

where

- $\hat{F}$ is an estimate of $F$,
- $y_r$ is the reference trajectory,
- $e = y - y_r$ is the tracking error,
- $K_P, K_D \in \mathbb{R}$ are the feedback gains

It yields

$$\ddot{e} - K_D \dot{e} - K_I e = F - \hat{F}.$$

If the estimate $\hat{F}$ is "good", *i.e.*, $F - \hat{F} \approx 0$, the choice of the gain $K_P$, $K_D$ for ensuring "local" stability around the reference trajectory is straightforward. This is a major difference with classic PIs and PIDs (see, *e.g.*, [7], [8]). Moreover and obviously no anti-windup is needed for iPDs.

*Remark 2.2:* Here $\dot{e}$ in Eq. (3) is given by an appropriate sensor for measuring $\dot{y}$.

*2) $m = 1$:* Eq. (1) becomes

$$\dot{y} = F(t) + \alpha u$$

The corresponding [9] *intelligent Proportional* controller, or *iP*, reads

$$u = -\frac{\hat{F} - \dot{y}_r + K_P e}{\alpha}. \tag{4}$$

The adaptation of Section II-C.1 is straightforward.

[2]According to [9], the choice $m = 2$ is necessary if there is (almost) no friction. The question however is far from being fully understood [27].

TABLE I: Control Parameters

| Parameter: | $m$ | $\alpha$ | $T$ | $K_d$ | $K_p$ |
|---|---|---|---|---|---|
| **Roll:** | 2 | 1 | 0.02s | 0.096 | 3.0 |
| **Pitch:** | 2 | 1 | 0.02s | 0.096 | 2.7 |
| **Yaw:** | 1 | 1 | 0.02s | | 2.7 |

### D. Estimation of $F$

A classic result from mathematical analysis (see, *e.g.*, [28]) states that under a weak integrability condition any function $\Phi : [a, b] \to \mathbb{R}$, $a, b \in \mathbb{R}$, $a < b$, may be approximated by a *step* function, *i.e.*, a piecewise constant function. Then, according to [9], an estimate $\hat{F}$ of $F$ is computed by averaging $F(t) = y^{(m)}(t) - \alpha u(t)$, which is deduced from Eq. (1), on a "short" sliding time window. If $m = 1$, Remark 2.2 tells us that $\dot{y}$ is obtained via a sensor. If $m = 2$, set $\ddot{y}(t) \approx \frac{\dot{y}(t) - \dot{y}(t-T)}{T}$, where $T > 0$ is the sampling period. A *Finite Impulse Response*, or *FIR*, filter is of course used in practice.

*Remark 2.3:* The unavoidable noise corruptions are attenuated via the averaging integral (see [29] for a mathematical explanation, and [30], [31] for applications to parameter identification and signal processing).

## III. Quadrotor Dynamics & Control

Many authors have reported models of quadrotor dynamics [1], [2]. We refer interested readers to said models to build some intuition about the behavior of such a vehicle and to recognize their nonlinearities, which are potent during acrobatic flight. Naturally, these are only approximate models of the quadrotor's dynamics; therefore, a controller that builds its own model from data is highly desirable.

### A. Quadrotor Attitude

A diagram of a quadrotor is shown in Figure 1 and is useful when considering its behavior. We consider the case of a 3 degree-of-freedom (DOF) quadrotor with controllable roll ($\phi$), pitch ($\theta$), and yaw ($\psi$) axes. The UAV's rate of rotation, or attitude, about each of these axes—$\dot{\phi}, \dot{\theta}$, and $\dot{\psi}$—is to be controlled by automatically adjusting the speed of each of the vehicle's four propellers, $\omega_1 - \omega_4$. The exact relationship between the vehicle's attitude and these speeds must be inferred by the controller. Some intuition may be gained from this figure. For instance, if one wanted to roll to the right, they would increase the speed of motor 4 and decrease the speed of motor 2. Pitching forward would entail increasing the speed of motor 3 and decreasing that of motor 1. Finally, yawing counter-clockwise would require increasing the speed of the clockwise rotating motors and decreasing the speed of the counter-clockwise motors. This intuition is used in Section III-B in order to deliver control signals to the motors. Only this low-level knowledge is given to the controller. All other unknown dynamics must be accounted for numerically.

### B. Quadrotor Control Scheme

A single instantiation of MFC will be assigned to each of the quadrotor's degrees of freedom: roll, pitch, and yaw.

Thus, a multi-variable control formulation will be implemented as outlined in Section II-B. Three control inputs—$u_\phi, u_\theta,$ and $u_\psi$—will be used to manipulate the vehicle's attitude. The output of these monovariable controllers must be translated into four separate control signals for each of the vehicle's four motors. The appropriate control signals for motors 1–4 are as follows:

$$\boldsymbol{u}_{\text{motors}} = \begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \end{bmatrix} = \begin{bmatrix} u_t - u_\theta - u_\psi \\ u_t - u_\phi + u_\psi \\ u_t + u_\theta - u_\psi \\ u_t + u_\phi + u_\psi \end{bmatrix}. \tag{5}$$

Here, $u_t$ is the baseline throttle given to the quadrotor by its pilot. If the vehicle were stationary and undisturbed, $u_t$ would be used by the pilot to adjust the height of the quad. A block diagram visualization of the proposed control scheme is shown in Figure 2.

In our physical implementation, iPDs are used to control the pitch and roll axes. Due to the high level of drag associated with the yaw axis, an iP is more appropriate. In traditional linear control of quadrotors, it is common to use PID on the pitch and roll axes while employing a PI controller on the yaw axis. It has been shown that the appropriate replacement to a PI is an iP and a the appropriate replacement to a PID is an iPD [9]. Due to the decoupled nature of control variables, we can use ultra-local models of varying order without fear of cross-talk or adverse effects.

The control parameters used in our physical implementation are listed in Table I. In each installment of MFC, elementary numerical methods are employed for the implementation (see Sect. II-D). The output of each of the three controllers is appropriately saturated so that the commands sent to each of the four motors is reasonable. Rate measurements taken by a MPU6050 gyroscope are passed through a simple complimentary filter. The output of the filter is a convex combination of the current measurement and the previous filter output. The controller must overcome the inherent noise that is allowed through this low-level filter. Thus, quality results will support the notion of robustness of the proposed technique. The controller is implemented on an 8-bit Atmega 328 micro-controller and uses a refresh rate of 250 Hz. The inexpensive hardware used here demonstrates that this control technique can be implemented with relatively little computational power (see also [32])—making it a very attractive nonlinear alternative to traditional PID used profusely on small UAVs.

## IV. Experimental Results[3]

We demonstrate the robustness of the proposed technique by using the same control scheme across vehicles with varying dynamics. The controller is not re-tuned between any of these experiments. All flights were flown outdoors on the campus of Mercer University in Macon, GA, USA, on the 18th of March, 2020. On this day there were southern winds blowing upwards of 9 mph.

[3]A video is available at: `https://youtu.be/wtSLalA4szc`

### A. DJI F450 Quadrotor

The first vehicle flown was a DJI F450 quadrotor equipped with readily available brushless motors and affordable plastic propellers. It is a light-weight drone whose center of gravity is located at the center of the body of the vehicle. The goal of the controller is to minimize the error between rate commands provided by the UAV's pilot and the rates at which the vehicle is rotating. The pilot flew the drone around an open field performing various maneuvers, *i.e.*, the reference trajectories are arbitrary and are generated in real time. The performance of the drone is shown in Figure 3. The mean absolute error for each degree of freedom was under 10 degrees per second, which is remarkable since no high-level filtering was used on sensor measurements.

### B. DJI F450 Minus Half Its Propellers

The dynamics of the quadrotor were then changed by cutting off significant portions of each of the vehicle's four propellers. Not only was the effectiveness of each propeller reduced, the drone became unbalanced and began to vibrate badly. This experiment has a myriad of applications. The most interesting is perhaps in defense scenarios where quadrotors are targeted with directed energy weapons or artillery fire. Once the dynamics of the vehicle change, non-adaptive control techniques tend to fail. Since the proposed technique is data-driven and model-free, it is not affected by the damage. Without re-tuning the controller, the performance shown in Figure 4 was achieved. The mean absolute error for each degree of freedom was under 20 degrees per second. The rate measurements here seem very noisy; however, this is reasonable given the unbalanced nature of the injured vehicle. The authors refer the readers to the provided YouTube video to observe the smooth performance generated by the controller. An image of the vehicle with its damaged rotors is shown in Figure 5.

### C. A new vehicle: Tarot 650 Sport Quadrotor

The controller, without being re-tuned, was transferred to a Tarot 650 Sport Quadrotor. This vehicle was equipped with high-efficiency brushless motors, heavy carbon-fiber propellers, a long-endurance flight battery, and retractable landing gear. Its center of gravity was much lower than that of the DJI. During this flight, one of the pieces of landing gear retracted upon takeoff, while the other was obstructed and remained in its initial position. This added to the complexity of the vehicle's behavior as it was unbalanced in this configuration. The proposed control technique responded well to both the vehicle swap and the unbalanced dynamics of the Tarot Quadrotor. The potential real-world applications of this experiment are again numerous as robotics rarely behave as they are designed to. The results of this flight are shown in Figure 7. Each of the three flights reported here are replicated in the aforementioned footage.

### D. Comparison to PID Control

In order to compare the robustness of MFC to PID, a PID controller was properly tuned and deployed on the large

Tarot quadrotor. The results of this flight are shown in Figure 8. After a successful flight, the controller was transferred to the DJI F450 and flown without re-tuning. The results of this flight are shown in Figure 9. It is clear that MFC is much more robust across vehicles with varying dynamics. This might also suggest the ease of tuning associated with MFC as compared to PID. Note that performance of PID suggests, also, that MFC is a better multi-variable control scheme. This is particularly apparent under PID control when performing robust maneuvers on the roll axis, which seems to degrade the performance of the pitch and yaw controllers.

## V. Conclusions & Future Work

A highly promising control technique for quadrotors has been derived without the need of any mathematical modeling in the usual sense.[4] Even as the dynamics of the vehicle changed—either from having damage inflicted upon it or from changing the vehicle altogether—the proposed technique performed well. The little computational power required to implement this algorithm means that it could be utilized on micro and macro aerial vehicles alike. Our future work is to demonstrate a full-scale Model-Free Control architecture on a 6-DOF quadrotor—the three additional DOF being longitude, latitude, and altitude, which will come from equipping our fleet of UAVs with global positioning systems and barometers. The control scheme will look very much like the one outlined here.

## References


[1] B.L. Stevens, F.L. Lewis, E.N. Johnson, *Aircraft Control and Simulation: Dynamics, Controls Design, and Autonomous Systems* (3rd ed.). Wiley, 2016.

[2] Quan Quan, *Introduction to Multicopter Design and Control*. Springer, 2017.

[3] B. Hassanalian, A. Abdelkefi, "Classifications, applications, and design challenges of drones: A review," *Progr. Aerosp. Sci.*, vol. 91, pp. 99-131, 2017.

[4] A. Zulu, S. John, "A review of control algorithms for autonomous quadrotors," *Open J. Appl. Sci.*, vol. 4, pp. 547-556, 2014.

[5] P. Pounds, R. Mahony, P. Corke, "Modelling and control of a large quadrotor robot," *Contr. Engin. Pract.*, vol. 18, pp. 691-699, 2010.

[6] N.O. Lambert, D.S. Drew, J. Yaconelli, S. Levine, R. Calandra, K.S.J. Pister, "Low-level control of a quadrotor with deep model-based reinforcement learning," *IEEE Robot. Automat. Lett.*, vol. 4, pp. 4224-4230, 2019.

[7] K.J. Åström, T. Hägglund, *Advanced PID Control*. Instrument Soc. Amer., 2006.

[8] K.J. Åström, R.M. Murray, *Feedback Systems: An Introduction for Scientists and Engineers*. Princeton University Press, 2008.

[9] M. Fliess, C. Join, "Model-free control," *Int. J. Contr.*, vol. 86, pp. 2228-2252, 2013.

[10] O. Bara, M. Fliess, C. Join, J. Day, S.M. Djouadi, "Toward a model-free feedback control synthesis for treating acute inflammation," *J. Theoret. Biology*, vol. 448, pp. 26-37, 2018.

[11] M. Fliess, C. Join, "Machine learning and control engineering: The model-free case," *Future Techno. Conf. (FTC)*, Vancouver, 2020. https://hal.archives-ouvertes.fr/hal-02851119/en/

[12] Y. Al Younes, A. Drak, H. Noura, A. Rabhi, A. El Hajjaji, "Robust model-free control applied to a quadrotor UAV," *J. Intel. Robot. Syst.*, vol. 84, pp. 37-52, 2016.

[13] H. Wang, X. Ye, Y. Tian, G. Zheng, N. Christov, "Model-free–based terminal SMC of quadrotor attitude and position," *IEEE Trans. Aero. Electron. Syst*, vol. 52, pp. 2519-2528, 2016.

[14] M. Bekcheva, C. Join, H. Mounier, "Cascaded model-free control for trajectory tracking of quadrotors," *Int. Conf. Unman. Aircr. Syst.*, Dallas, 2018.

[15] J.M.O. Barth, J.-P. Condomines, M. Bronz, G. Hattenberger, J.-M. Moschetta, C. Join, M. Fliess, "Towards a unified model-free control architecture for tail sitter micro air vehicles: Flight simulation analysis and experimental flights," *AIAA Scitech Forum*, Orlando, 2020.

[16] J.M.O. Barth, J.-P. Condomines, M. Bronz, J.-M. Moschetta, C. Join, M. Fliess, "Model-free control algorithms for micro air vehicles with transitioning flight capabilities," *Int. J. Micro Air Vehic.*, 12, 2020. DOI: 10.1177/1756829320914264

[17] Z. Chekakta, M. Zerikat, Y. Bouzid, M. Abderrahim, "Model-free control applied for position control of quadrotor using ROS," *6th Int. Conf. Contr. Decis. Informat. Techno.*, Paris, 2019.

[18] N.D. Özbek, "An evaluation of model-free control strategies for quadrotor type unmanned aerial vehicles," *3rd Int. Conf. Appl. Automat. Indust. Diagnost.*, Elazig, 2019.

[19] Y. Chen, N.O. Pérez-Arancibia, "Controller synthesis and performance optimization for aerobatic quadrotor flight," *IEEE Trans. Contr. Syst. Techno.*, 2019. doi: 10.1109/TCST.2019.2919819

[20] E. Kaufmann, A. Loquercio, R. Ranft, M. Müller, V. Koltun, D. Scaramuzza, "Deep drone acrobatics," *Robot. Sci. Syst.*, 2020. arXiv:2006.05768

[21] F. Lafont, J.-F. Balmat, N. Pessel, M. Fliess, "A model-free control strategy for an experimental greenhouse with an application to fault accommodation," *Comput. Electron. Agricul.*, vol. 110, pp. 139-149, 2015.

[22] Y. Wang, H. Li, R. Liu, L. Yang, X. Wang, "Modulated model-free predictive control with minimum switching losses for PMSM drive system," *IEEE Access*, vol. 8, pp. 20942-20953, 2020.

[23] J. De Miras, C. Join, M. Fliess, S. Riachy, S. Bonnet, "Active magnetic bearing: A new step for model-free control," *52nd IEEE Contr. Dec. Conf.*, Florence, 2013.

[24] L. Menhour, B. d'Andréa-Novel, M. Fliess, D. Gruyer, H. Mounier, "An efficient model-free setting for longitudinal and lateral vehicle control. Validation through the interconnected pro-SiVIC/RTMaps prototyping platform," *IEEE Trans. Intelligent Transport. Syst.*, vol. 19, pp.461-475, 2018.

[25] M. Ticherfatine, Q. Zhu, "Fast ferry smoothing motion via intelligent PD controller," *J. Marine. Sci. Appl.*, vol. 17, pp. 273-279, 2018.

[26] M. Haddar, R. Chaari, S.C. Baslamisli, F. Chaari, M. Haddar, 'Intelligent PD controller design for active suspension system based on robust model-free control strategy," *Instit. Mech. Engin. Part C: J. Mech. Engin. Sci.*, vol. 233, pp. 4863-4880, 2019.

[27] C. Join, E. Delaleau, M. Fliess, C.H. Moog, "Un résultat intrigant en commande sans modèle," *ISTE OpenSci. Contr./Autom.*, vol. 1, 9 pages, 2017. https://hal.archives-ouvertes.fr/hal-01628322/en/

[28] W. Rudin, *Real and Complex Analysis*, McGraw-Hill, 1976.

[29] M. Fliess, "Analyse non standard du bruit," *C.R. Acad. Sci. Paris Ser. I*, vol. 342, pp. 797-802, 2006.

[30] H. Sira-Ramírez, C. García-Rodríguez, J. Cortès-Romero, A. Luviano-Juárez, *Algebraic Identification and Estimation Methods in Feedback Control Systems*, Wiley, 2014.

[31] R. Morales, E. Segura, J.A. Somolinos, L.R. Núñez, H. Sira-Ramírez, "Online signal filtering based on the algebraic method and its experimental derivation," *Mech. Syst. Sign. Proc.*, vol. 66-67, pp. 374-387, 2016.

[32] C. Join, F. Chaxel, M. Fliess, "Intelligent controllers on cheap and small programmable devices," *2nd Int. Conf. Contr. Fault-Tolerant Syst.*, Nice, 2013. https://hal.archives-ouvertes.fr/hal-00845795/en/


[4] See [11] for a connection with Machine Learning.

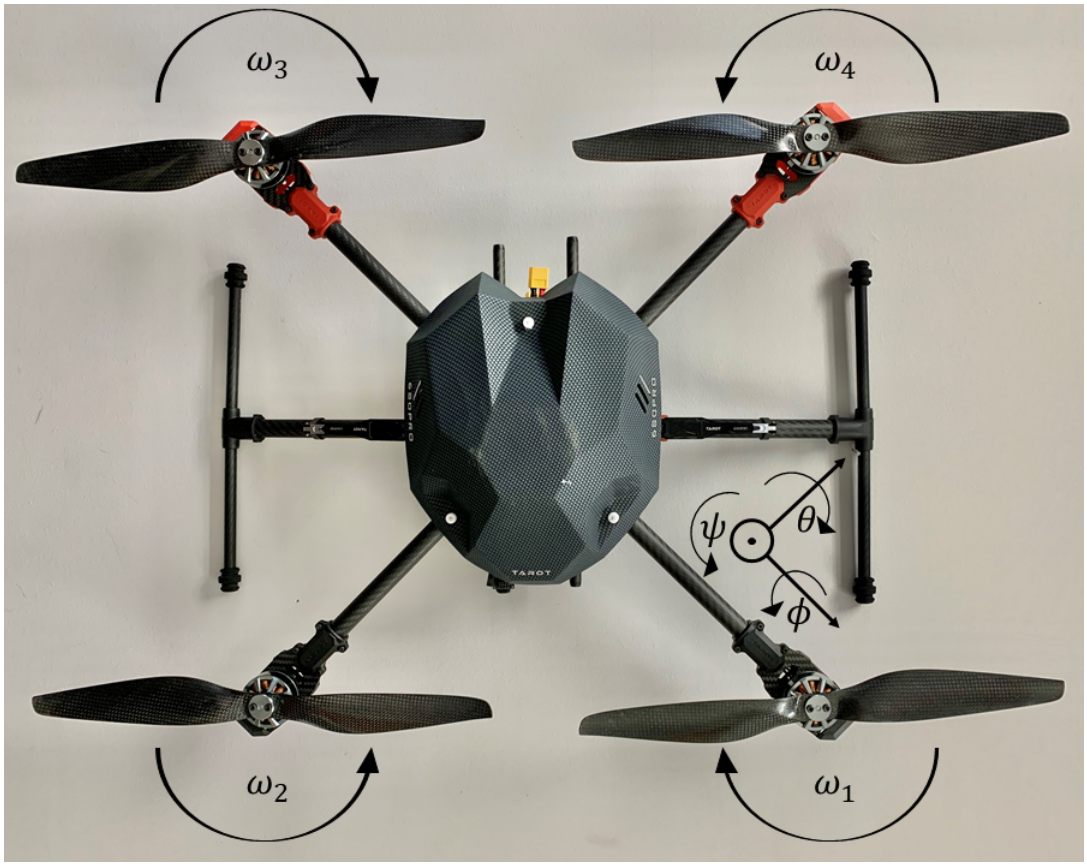


Fig. 1: A Tarot 650 Sport quadrotor. This vehicle is flown in the experiments discussed in Section IV.

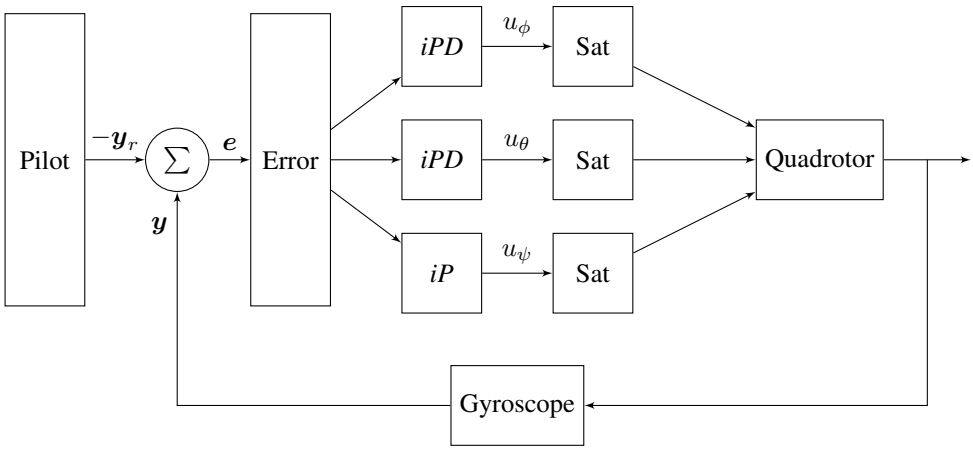


Fig. 2: Model-Free Control Scheme. "Sat" refers to passing the output of each control signal through a saturation function such that only reasonable commands are given to the motors of the quadrotor. The block labelled "Error" represents de-multiplexing the error signal and sending error values to the appropriate controller.

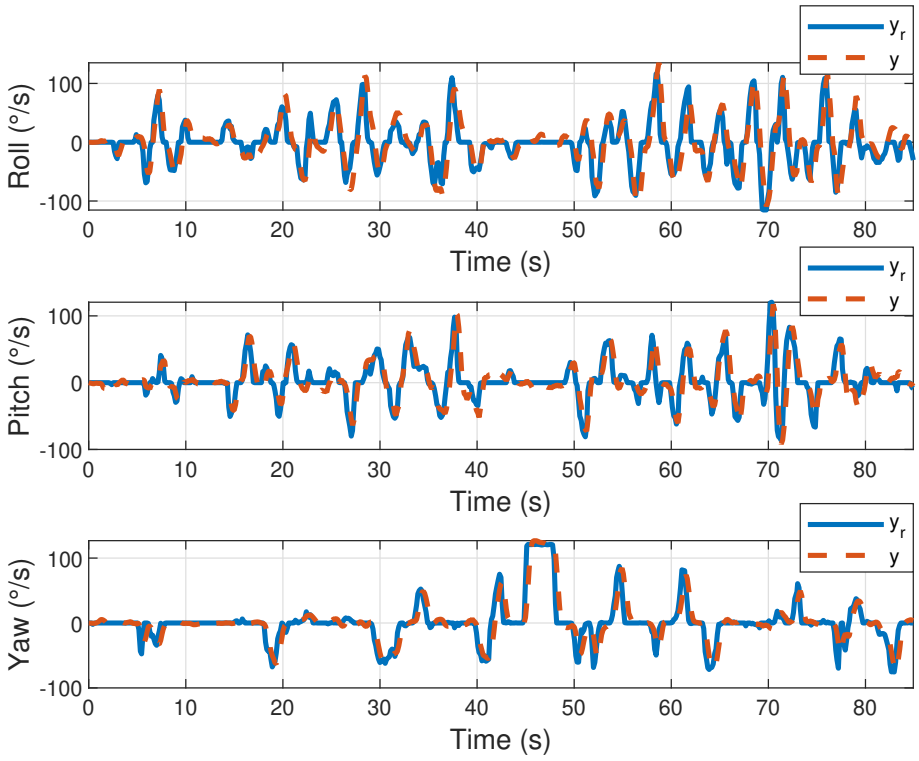


Fig. 3: DJI F450 roll, pitch, and yaw performance.

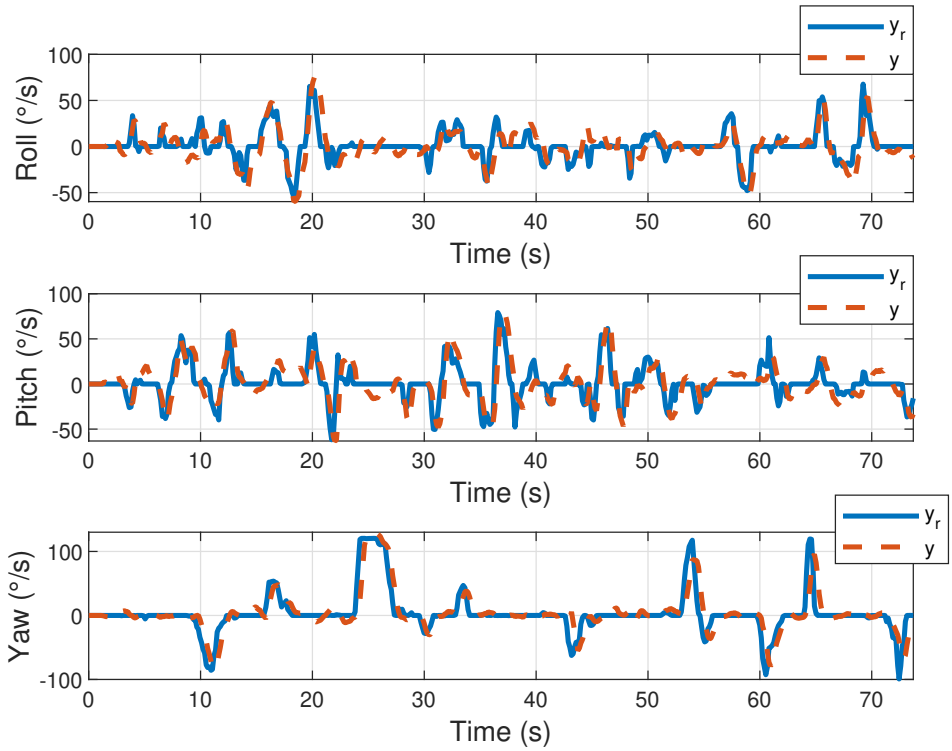


Fig. 4: DJI F450 roll, pitch, and yaw performance with severely damaged propellers.

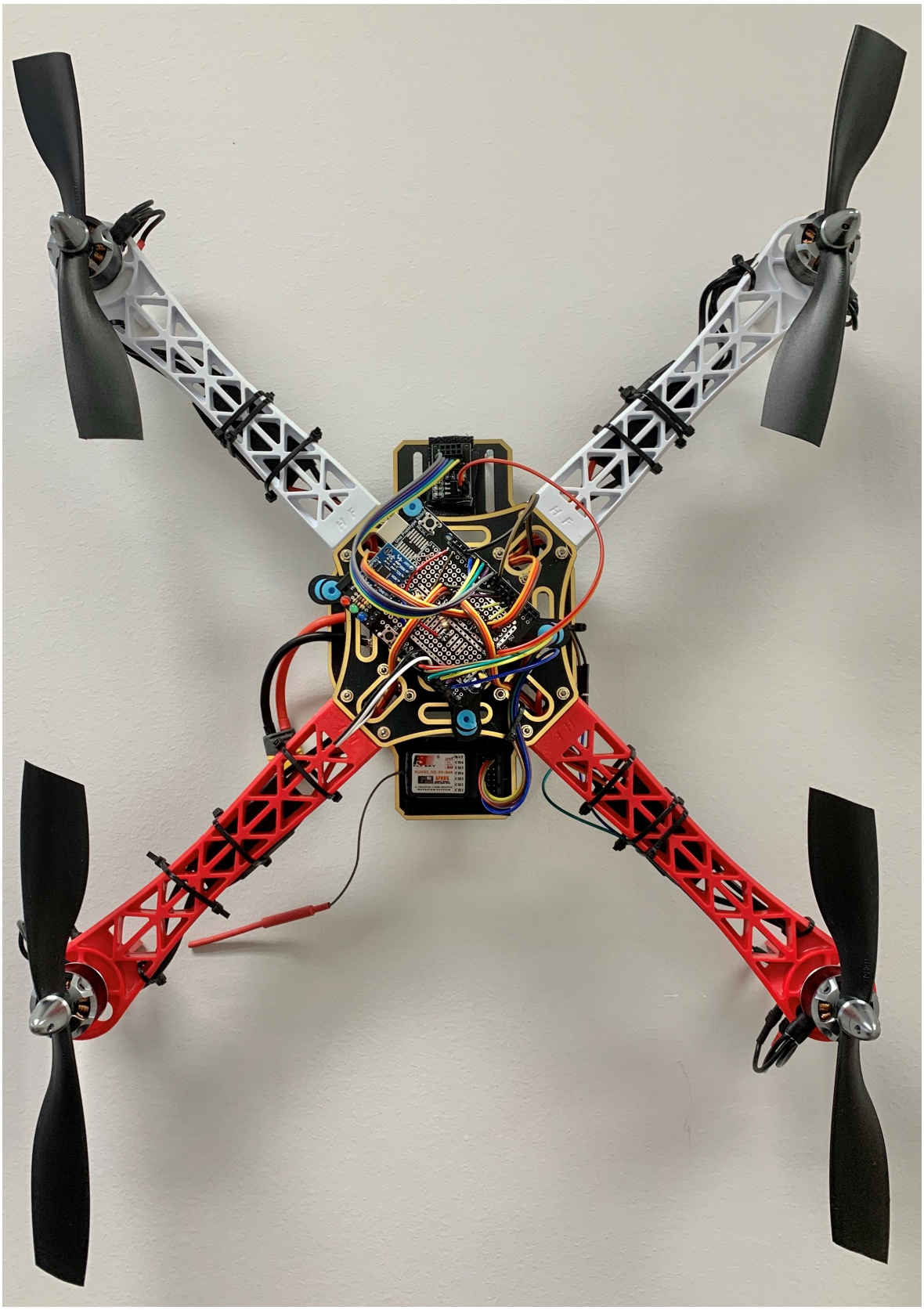

Fig. 5: A DJI F450 with damaged rotors. This serves to simulate a quadrotor being damaged in flight.

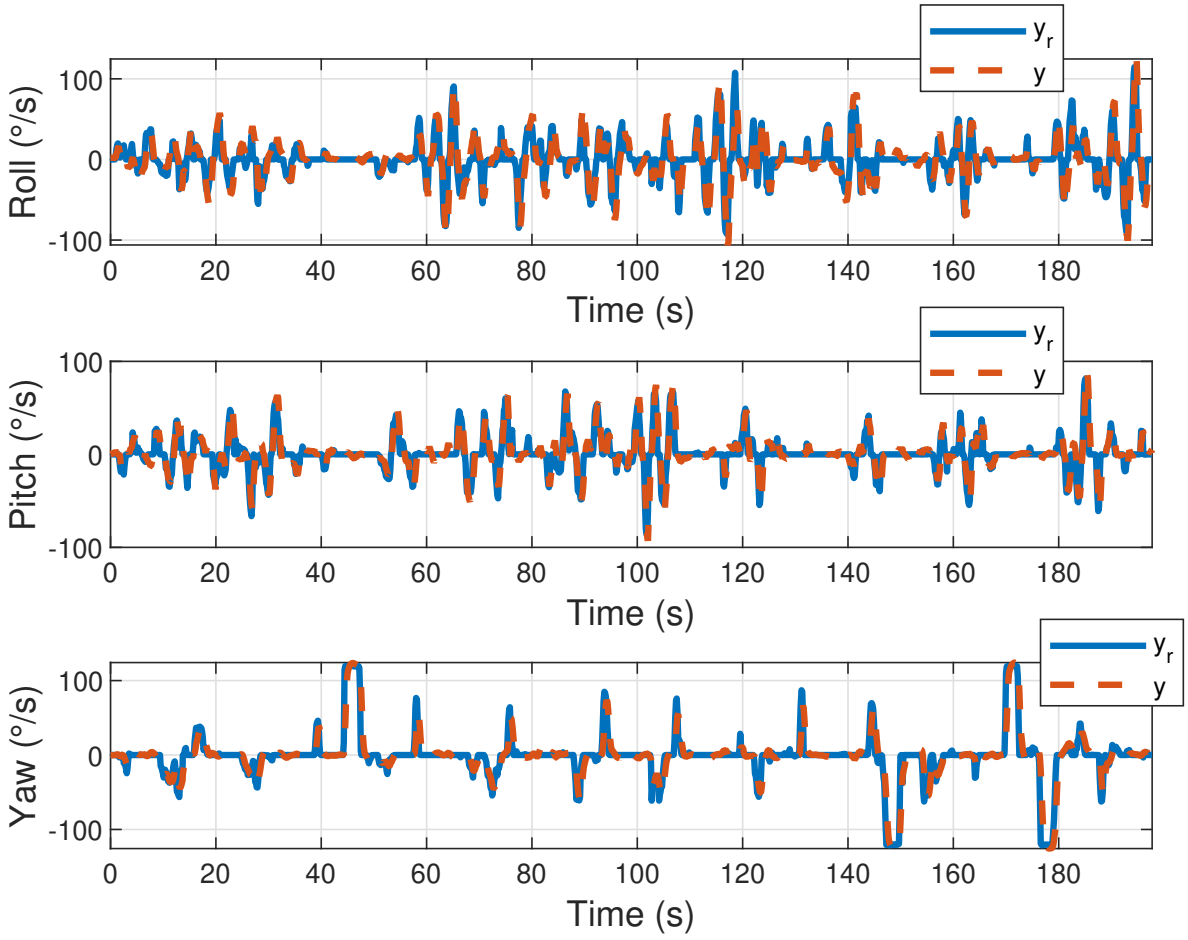


Fig. 6: Tarot roll, pitch, and yaw performance after failed landing gear deployment under MFC.

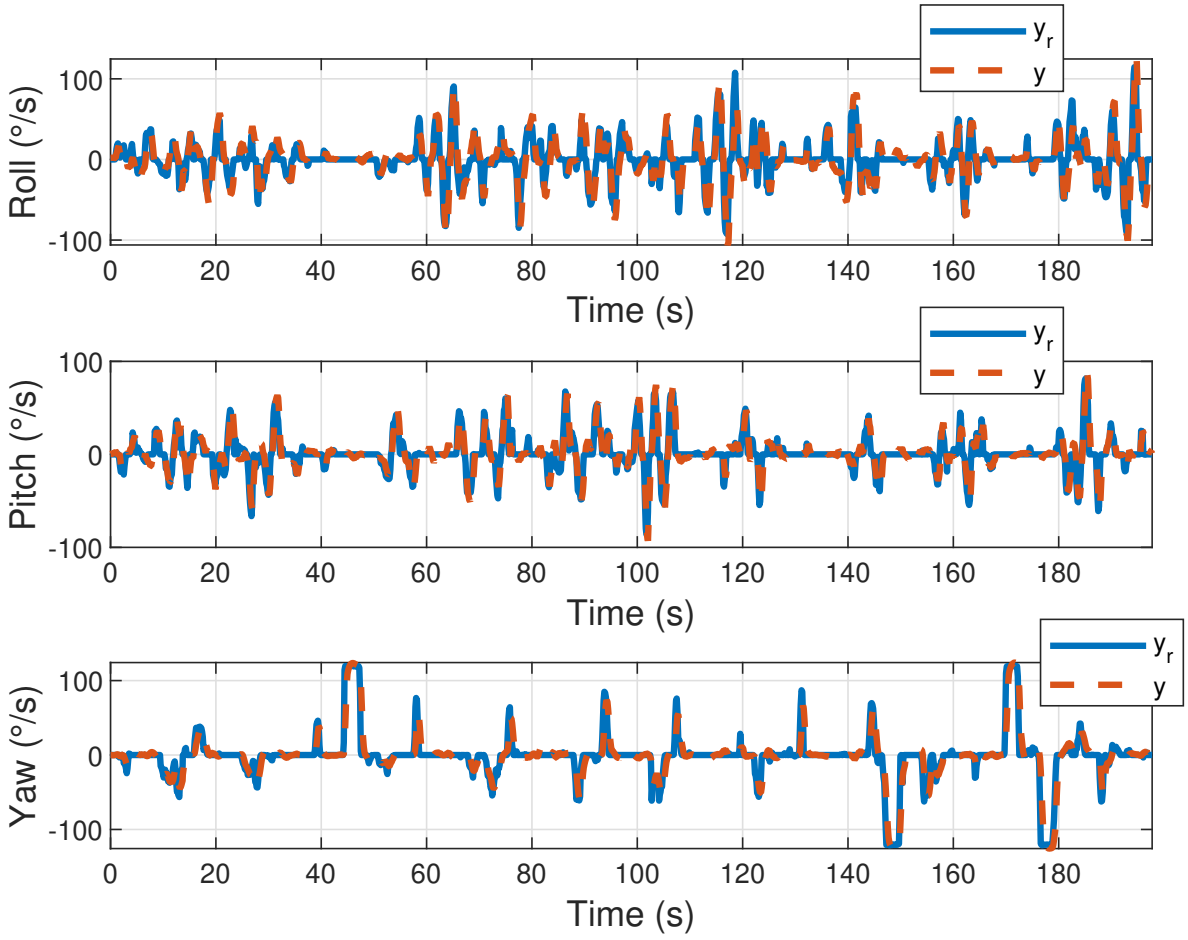


Fig. 7: Tarot roll, pitch, and yaw performance after failed landing gear deployment under MFC.

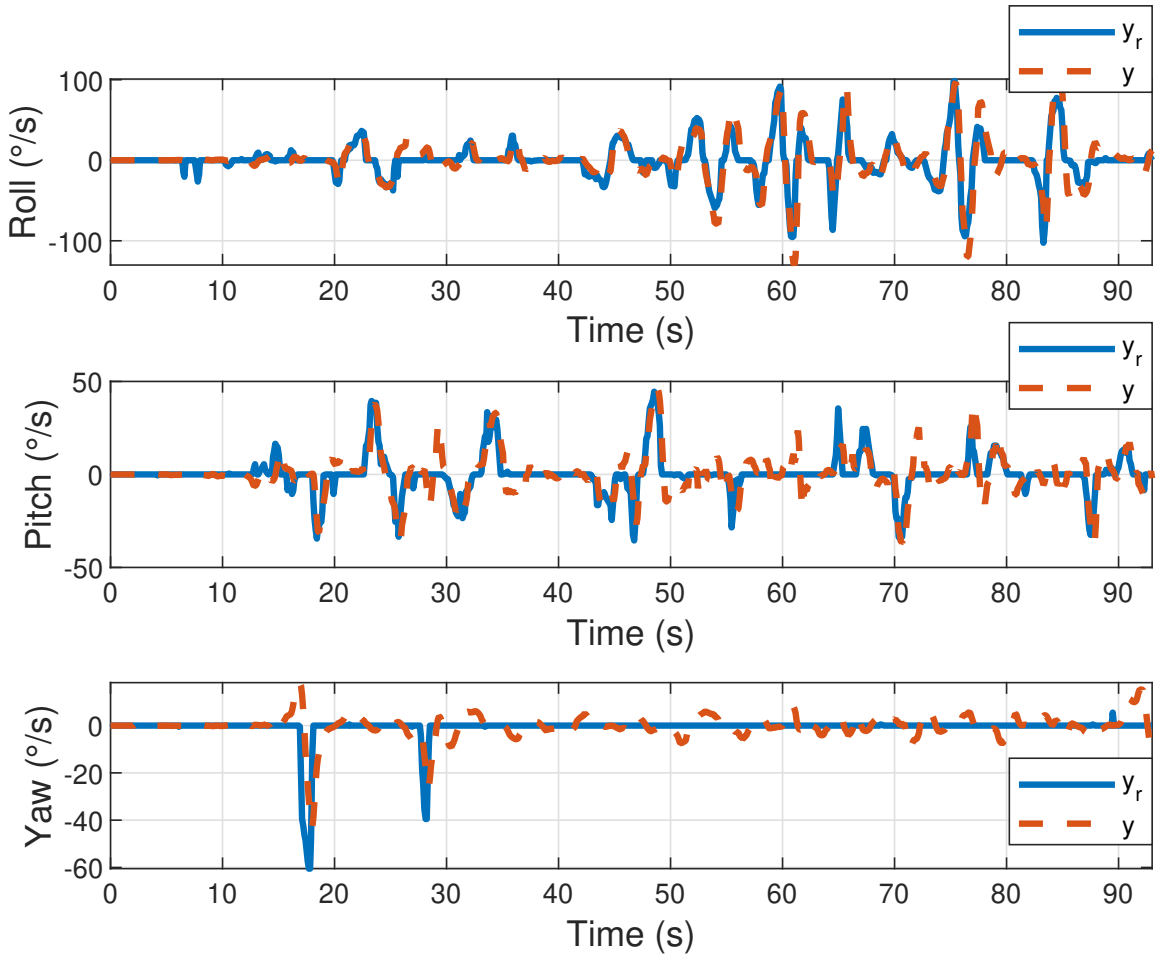


Fig. 8: Tarot roll, pitch, and yaw performance under PID control.

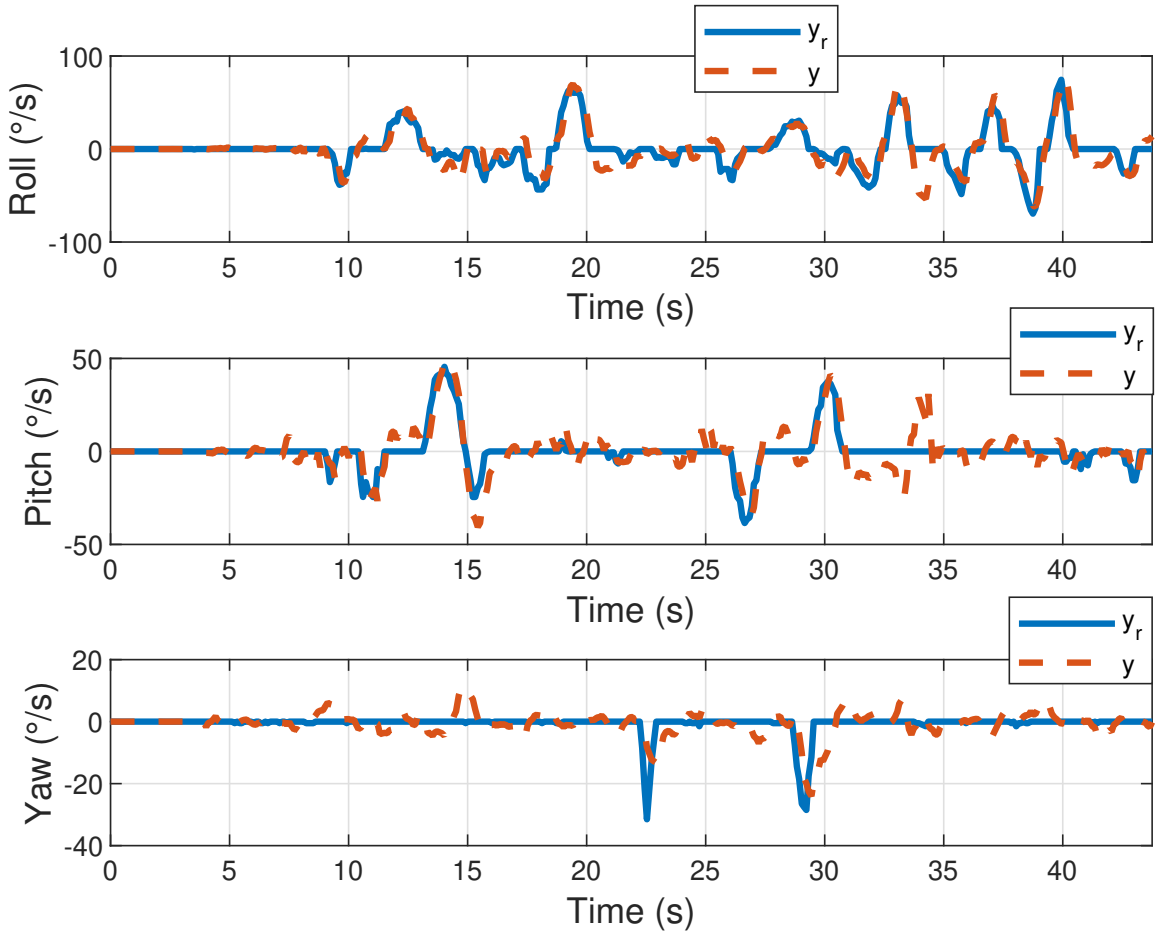


Fig. 9: F450 roll, pitch, and yaw performance under PID control. No re-tuning was conducted after moving the controller from the Tarot.