2018 IEEE International Conference on Robotics and Automation (ICRA)
May 21-25, 2018, Brisbane, Australia

Differential Flatness Transformations for Aggressive Quadrotor Flight

Benjamin Morrell[1], Marc Rigter[1], Gene Merewether[2], Robert Reid[2],
Rohan Thakker[2], Theodore Tzanetos[2], Vinay Rajur[3] and Gregory Chamitoff[1]

***Abstract*— Aggressive maneuvering amongst obstacles could enable advanced capabilities for quadrotors in applications such as search and rescue, surveillance, inspection, and situations where rapid flight is required in cluttered environments. Previous works have treated quadrotors as *differentially flat* systems, and this property has been exploited widely to design simple algorithms that generate dynamically feasible trajectories and to enable hierarchical control. The differentially flat property allows the full state of the quadrotor to be extracted from the reduced dimensional space of x, y, z, yaw and their derivatives. This differential flatness transformation has a number of singularities, however, as well as stability issues when controlling near these singularities. Many methods have been described in the literature to address these; however, they all have limitations when exploring the full flight envelope of a quadrotor, including roll or pitch angles past 90°, and during inverted flight. In this paper, we review these existing methods and then introduce our method, which combines multiple methods to provide a highly-robust differential flatness transformation that addresses most of these issues. Our approach is demonstrated enabling highly-aggressive quadrotor flight in both simulations and real-world experiments.**

I. INTRODUCTION

The development of autonomous quadrotors has seen rapid progress in recent years, with capabilities developing to support a range of applications, including: search and rescue, surveillance, facility inspection, and delivery [1]–[4]. These applications often require operations in environments with numerous obstacles, and limited space to maneuver, such as through buildings, forests, or crowded urban settings. Within these environments, the need for high-speed navigation while reacting to dynamic obstacles, demands an ability to generate and control large linear and angular accelerations. This is often referred to as *aggressive* flight [5]. An example of a trajectory that is aggressive is shown in Fig. 1.

For autonomous navigation, two key components are 1) a trajectory planner and 2) a trajectory tracking controller; these components introduce many challenges when striving for aggressive flight. For quadrotors, each of these components can utilize a property called *differential flatness* [6], [7], which allows direct mapping from the *flat outputs* of x, y, z and ψ (where ψ is yaw), plus their derivatives, through the full quadrotor state to the *flat inputs:* the input revolutions per minute (RPM) squared for each motor. A continuous trajectory planned in the flat output space gives continuous motor RPMs in the flat input space, providing a convenient method to ensure a dynamically feasible trajectory (given RPM magnitudes are within limits).

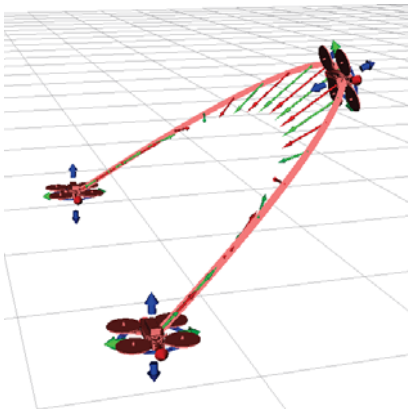

Fig. 1. An aggressive trajectory planned between three waypoints. Arrows represent the direction and relative magnitude of commanded acceleration.

For trajectory tracking controllers, a hierarchical architecture is often used with an outer loop position controller, and an inner loop attitude controller [8]–[10], as depicted in Fig. 2. The outer loop controller gives a desired thrust vector ($\mathbf{T}_{sp}$) that is transformed through the differential flatness transformation to a desired attitude ($\mathbf{q}_{sp}$) and thrust magnitude (T). The inner loop attitude controller tracks this attitude, outputting torques (τ) that are mixed with T to compute target RPMs for each motor controller.

This planning and control architecture utilizing differential flatness has been widely employed to great effect [2], [11]–

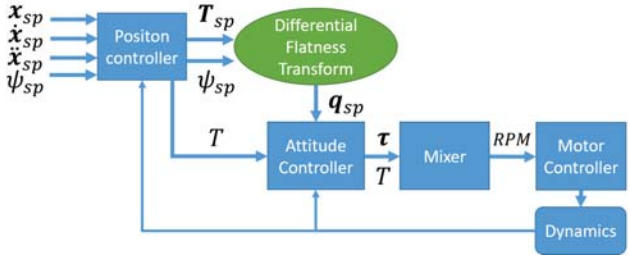

Fig. 2. Block diagram of the hierarchical tracking controller considered in this work. Here, $\mathbf{x}_{sp}$ and ψ_{sp} are the position and yaw set points for the outer position controller, which outputs the thrust set point $\mathbf{T}_{sp}$. The differential flatness transformation provides the attitude set point $\mathbf{q}_{sp}$ to inner attitude controller. The attitude controller outputs torques (τ) that are mixed with thrust magnitude T to compute target motor RPMs.

*This research was carried out at the Jet Propulsion Laboratory, California Institute of Technology, and was sponsored in part by Google. Morrell and Rigter were supported by the University of Sydney, Australia.

[1] School of Aeronautical, Mechanical and Mechatronics Engineering at the University of Sydney, Australia. Corresponding author: benjamin.morrell@sydney.edu.au

[2] Jet Propulsion Laboratory, California Institute of Technology, USA. Corresponding author: theodore.tzanetos@jpl.nasa.gov

[3] College of Computing, Georgia Institute of Technology, USA

[17]. The differential flatness transformation is commonly discussed as a key part for trajectory planning [8], [9] and is a core part of the controller by Lee [10], which has been used in many demonstrations with impressive results [11]–[15].

A. Singularities

Despite the frequent and successful use of the differential flatness transformation, there are known singularities that occur 1) when there is zero thrust (when the desired acceleration is fully achieved by gravity), and 2) when the desired thrust vector is in the xy plane and aligned with the desired direction of travel (e.g. pitched forward at $90°$). The first singularity is a fundamental limitation of the transformation, founded on the notion that the desired thrust direction sets the quadrotor attitude. The second singularity, and the sensitivity of states near this singularity, needs to be managed robustly.

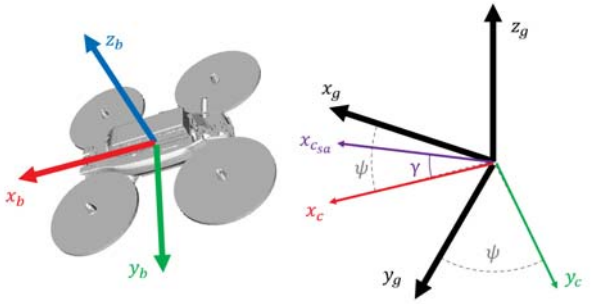

Fig. 3. Quadrotor body axes (subscript b), global axes (subscript g), interim yaw axes (subscript c), and second angle axes (subscript c_{sa}, with angle γ). East, North, Up (ENU) convention.

A thorough description of the differential flatness transformation can be found in references such as [10] and [11]; we repeat the main steps here. The starting assumption is that the thrust vector of the quadrotor sets the direction of the z body axis, $\mathbf{z}_b$ (see Fig. 3). Hence, from a planned trajectory that gives the desired acceleration $\ddot{\mathbf{x}}_{sp}$, we get the thrust vector ($\mathbf{T}$), that sets the direction of the $\mathbf{z}_b$:

$$\mathbf{T} = \ddot{\mathbf{x}}_{sp} - \mathbf{e}_3 g \tag{1}$$

$$\mathbf{z}_b = \frac{\mathbf{T}}{||\mathbf{T}||} \tag{2}$$

Here, g is the acceleration due to gravity, and $\mathbf{e}_3$ the global z unit vector. The first type of singularity occurs in Eq. 2 when the thrust magnitude is zero.

The desired yaw angle, ψ_{sp} sets the vector $\mathbf{x}_c$: a unit vector in the xy plane pointing in the desired heading direction. The cross product of $\mathbf{z}_b$ with $\mathbf{x}_c$ gives an orthogonal vector, $\mathbf{y}_b$, that represents the y body axis direction. This axis is crossed with $\mathbf{z}_b$ to get a third orthogonal vector $\mathbf{x}_b$; this provides the x body axis, giving the overall orientation $\mathbf{R}$:

$$\mathbf{x}_c = \begin{bmatrix} \cos(\psi_{sp}), & \sin(\psi_{sp}), & 0 \end{bmatrix} \tag{3}$$

$$\mathbf{y}_b = \frac{\mathbf{z}_b \times \mathbf{x}_c}{||\mathbf{z}_b \times \mathbf{x}_c||} \tag{4}$$

$$\mathbf{x}_b = \mathbf{y}_b \times \mathbf{z}_b \tag{5}$$

$$\mathbf{R} = \begin{bmatrix} \mathbf{x}_b, & \mathbf{y}_b, & \mathbf{z}_b \end{bmatrix} \tag{6}$$

The second source of singularities occurs in Eq. 4, when $\mathbf{z}_b \times \mathbf{x}_c = 0$, i.e. when $\mathbf{z}_b$ is parallel with $\mathbf{x}_c$. Note that this can occur in two orientations, with both positive $\mathbf{x}_c$ and negative $\mathbf{x}_c$. In addition to this singularity, the transformation is also highly sensitive at states around this singularity, e.g. when $||\mathbf{z}_b \times \mathbf{x}_c||$ is small.

The first singularity (zero thrust) is fundamental to the transformation, and it can be avoided by setting a constraint on the minimum thrust in both the planners and controllers. The second type of singularity must be managed carefully; this paper explores how to do this robustly, and evaluates several methods with the goal of enabling highly aggressive flight. Throughout the rest of this paper, *singularity* refers to this second type.

II. Existing Approaches to Address the Transformation Singularity

In this section we review a range of methods from the literature that address the singularity in the differential flatness transformation. In the subsequent section, we analyze each of these methods and assess their limitations.

A. Standard Method

The Standard transformation, as described in the previous section, is widely used without any modifications, as described in review papers [8], [9], and numerous quadrotor systems [10], [13], [17], [18]. There are many applications that build from the controller by Lee [10], that simply assumes the singularity is not met. These applications have been very effective, such as Allen [13] navigating through crowded indoor environments, and deal with the singularity by operating in a regime far from it: flying with relatively small deviations from the hover state. Falanga [18] and Neunert [17] push the dynamics up to $45°$ and $30°$ of roll, respectively, using the Standard transformation, however their demonstrations are far from the singularity at $90°$.

B. Negative Check Method

Mellinger [11] demonstrates more aggressive flight, with up to $90°$ of roll through a window, operating in states that could be approaching the singularity in the transformation (also subsequently demonstrated by Loianno [14]). They note that the negative x and y body axes can be consistent with the desired yaw angle and desired $\mathbf{z}_b$, and check which of $(\mathbf{x}_b, \mathbf{y}_b, \mathbf{z}_b)$ or $(-\mathbf{x}_b, -\mathbf{y}_b, \mathbf{z}_b)$ are closest to the current orientation. The closest axes set is then used as the output of the transformation. Mellinger's approach has been used very effectively for autonomous navigation in ships [2], and through indoor environments [12], however only with moderate roll or pitch angles. Mellinger and Loianno only demonstrate roll in isolation, and do not exceed $90°$, therefore they do not operate near the sensitive regions of the transformation.

C. Second Angle Method

Thomas [15] uses a method that enables orientations holding at $90°$ for an extended period, and extending past $90°$,

with flight demonstrations of a quadrotor perching on both vertical surfaces and the underside of inclined surfaces. To move away from the singularity (see Fig. 3), their approach employs an additional *working angle*, γ, to rotate the $\mathbf{x}_c$ vector in the vertical plane:

$$\mathbf{x}_c = \begin{bmatrix} \cos(\psi)\cos(\gamma), & \sin(\psi)\sin(\gamma), & \sin(\gamma) \end{bmatrix} \quad (7)$$

This moves $\mathbf{x}_c$ away from the xy plane to avoid being parallel with $\mathbf{z}_b$. While their approach moves the location of singularity, it does not remove it. Thomas' demonstrations show the effectiveness of their approach, however the examples only operate with one axis of rotation (pitch), rather than the full range of quadrotor motion.

D. Inverted flight

None of the previous work described above have demonstrated highly-aggressive flight with 360° rotations through inversion. There are a number of examples that demonstrate such flight, but these tend to employ switching controllers that split the trajectory into simplified segments including launch, ballistic trajectory, constant roll rates, and recovery. These methods either learn a set of parameters for the simplified maneuver, such as multiple flips [5], [19], or fit a model to the flight of an expert pilot [20], [21]. The segmentation and simplification of the maneuvers avoids any singularities, however this approach is restricted to a limited range of dynamics.

E. Recovery

One of the key components of aggressive maneuvers, such as those mentioned above, is the recovery phase. Faessler et al. [16] developed an autonomous recovery controller that manages the differential flatness transformation effectively. Their approach splits control of the pitch and roll rates from control of the yaw rate. First the angular error for pitch and roll is derived, independent of the singularity, to give the correct $\mathbf{z}_b$. Multiplying the errors by a gain gives desired pitch and roll rates. The additional rotation required due to yaw is computed using the standard differential flatness transformation, but as the roll and pitch rates are already determined, the yaw rate can simply be set to zero when the singularity is met. Additionally, if the $\mathbf{z}_b$ axis is pointing down, then negative $\mathbf{x}_c$ is used, very similarly to the default transformation in the PX4 flight controller [22]. This approach was shown to be successful as part of aggressive maneuvers in Falanga's work [18], including from inverted attitudes. Yet these recoveries are to a commanded horizontal attitude, and hence the system is never commanding 90° pitch or roll or inverted orientations, where the sensitivities occur.

F. Summary

Each of the differential flatness transformation methods described above have limitations that can cause the transformation to fail in particular scenarios. While singularities may be avoided, in the next section we show that there are still issues *around* the singularity and in transitions near it, where the transformations give large changes in attitude with a small change in the acceleration vector.

III. Analysis of Differential Flatness Transformations

In this section, we analyze the transformation methods described in the previous section and highlight where issues may arise. We propose new methods that address these issues, and subsequently analyze them. A summary of this analysis is presented in Table I. Our goal is to show the limitations of each method, and the best method for applying differential flatness throughout the entire flight envelope of a quadrotor.

A. Standard

The Standard transformation suffers directly at the singularity, yet of greater concern, is the change in attitude on either side of the singularity. For example; when the $\mathbf{z}_b$ axis transitions from above the xy plane to below in a pitching forward maneuver (Fig. 4), the direction of the $\mathbf{y}_b$ axis coming out of the cross product in Eq. 4 flips 180°. This then flips the $\mathbf{x}_b$ axis, giving an extremely large change in attitude.

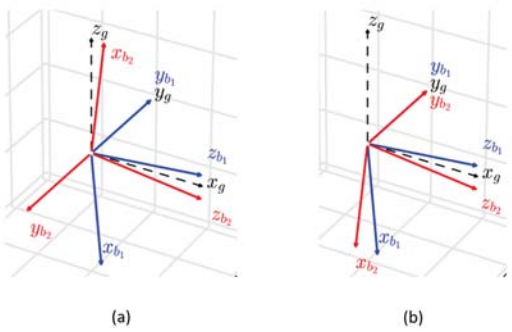

Fig. 4. 10° of pitching through 90°, and the singularity. (a) Standard method failing. (b) Negative Check method succeeding. Axes slightly before 90°, are in blue, and axes slightly after 90° are in red. The black dashed axes represent the global coordinate frame.

B. Negative Check

A solution to this issue, is to take the Negative Check method proposed by Mellinger [11], or the PX4 method [22]. Negating $\mathbf{x}_b$ and $\mathbf{y}_b$, maintains a similar attitude through the transition (see Fig. 4.b). The Negative Check approach is still susceptible at the singularity though, and is sensitive when pitching through 90° near the singularity (Fig. 5.a). Both the Negative Check method and the PX4 method have issues when there are fast dynamics near the singularity.

C. Using Other Axes

Another approach for dealing with the singularity is to recognize that it is possible to use both $\mathbf{x}_c$ and $\mathbf{y}_c$ as the axis in the intermediate frame, when taking the problematic cross

product in Eq. 4. i.e. the body axes could be equivalently determined by:

$$\mathbf{y}_c = \begin{bmatrix} -\sin(\psi), & \cos(\psi), & 0 \end{bmatrix} \tag{8}$$

$$\mathbf{x}_b = \frac{\mathbf{y}_c \times \mathbf{z}_b}{||\mathbf{y}_c \times \mathbf{z}_b||} \tag{9}$$

$$\mathbf{y}_b = \mathbf{z}_b \times \mathbf{x}_b \tag{10}$$

Two methods that can be used to select the best vector to use: 1) taking the vector that is closest to $90°$ from $\mathbf{z}_b$ (the Check Furthest Approach), 2) selecting the vector that gives a resulting $\mathbf{x}_b$ closest to the current $\mathbf{x}_b$. These approaches can be quite effective, but still have sensitive regions: when transitioning through the decision point for $\mathbf{x}_c$ and $\mathbf{y}_c$, and when the body x axis is moving through a vertical orientation.

Alternatively, the current body axes, $\mathbf{x}_b$ and $\mathbf{y}_b$ can be used in place of $\mathbf{x}_c$ and $\mathbf{y}_c$, ensuring that the axes used in the cross products are always be nearly orthogonal to $\mathbf{z}_b$ (referred to here as the Use Current $\mathbf{x}_b$ method). While effective (successful in all scenarios above), this approach sacrifices the ability to control yaw, instead simply maintaining the previous orientation (or drifting with the current attitude estimate).

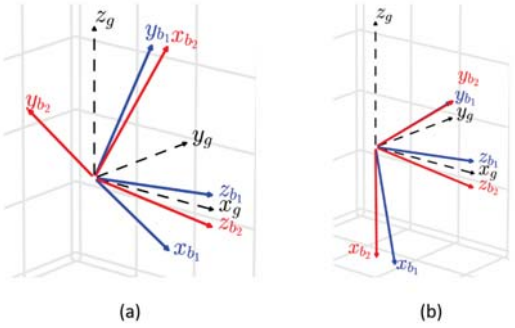

Fig. 5. $10°$ of pitching through $90°$, near the singularity (yaw at $5°$). (a) Negative Check method failing. (b) Combined methods succeeding. Axes before the transition are in blue and axes after the transition are in red. The black dashed axes represent the global coordinate frame.

D. Second Angle

The Second Angle approach described by Thomas [15], is successfully able to avoid the singularity by moving where it is encountered, but introduces challenges in controlling the second angle, γ (the best method for doing so was not fully explained by the authors). Setting a constant angle simply shifts the singularity. Not controlling γ correctly could also lead to discontinuous jumps. We could take the approach to set γ so that $\mathbf{x}_c$ is always chasing or leading $\mathbf{z}_c$. In this case, a $360°$ pitching maneuver can successfully be described. Yet this same approach will fail when rolling through $90°$. If this approach is combined with checking the negative set of the result, then the second angle approach is able to perform well in most scenarios.

E. Pitch and Roll only

If the application is agnostic to the yaw angle, then a method inspired by Faessler's [16] angular error calculations can be used. The required rotation from the global z axis, $\mathbf{z}_g$, to the desired body $\mathbf{z}_b$ axis can be described by a quaternion ($\mathbf{q}$), giving the attitude of the vehicle. This avoids any issues with singularities or sensitive regions, but does not grant any control of yaw.

F. Combined Methods

Our proposed methods to address the issues with the singularity along with the ability to set yaw is to combine methods. The underlying philosophy is to compute the transformation with multiple different approaches, and then select the result giving the smallest change in orientation. The body axes can be computed with $\mathbf{x}_c$, $\mathbf{y}_c$ and the second angle approach (using a constant, fixed $\gamma = 15°$), giving three possible solutions, plus the three from taking the negative $\mathbf{x}_b$ and $\mathbf{y}_b$ of their results, giving a total of six axes sets. Each of these axes sets can be compared against the previously computed orientation[1] and the closest matching result selected.

Good performance for the scenarios above can be achieved by using just four axes sets with the positive and negative results from $\mathbf{x}_c$ and $\mathbf{y}_c$. Yet still can encounter issues, as depicted in Fig. 6.a in a scenario of pitching when at $90°$ roll.

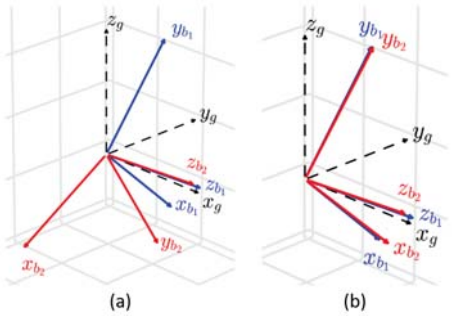

Fig. 6. $5°$ of pitching while at $90°$ roll (yaw at $90°$). (a) Four Axes Combined method failing. (b) Six Axes Combined method succeeding. Axes before the transition are in blue and axes after the transition are in red. The black dashed axes represent the global coordinate frame.

Introducing more axes, from the second angle method, gives an orientation option of lower error in the region of sensitivity. Fig. 7 shows the selection of the closest axes result, and the error for each axes set throughout a trajectory. At the point where $90°$ of roll is reached, there is a large change in the axes using $\mathbf{x}_c$, $\mathbf{y}_c$, as shown by the large changes in the axes errors (Fig. 7.a center), but the second angle method is not as sensitive in that same region, and is able to maintain a feasible trajectory, (Fig. 7.b center).

This Six-Axes Combined method is robust enough to cover the spread of possible dynamic transitions tested without large discrete changes in orientation.

[1]In a controller this is the previous attitude set-point. Alternatively the current attitude could be used, however, this was found to be inferior in flight tests.

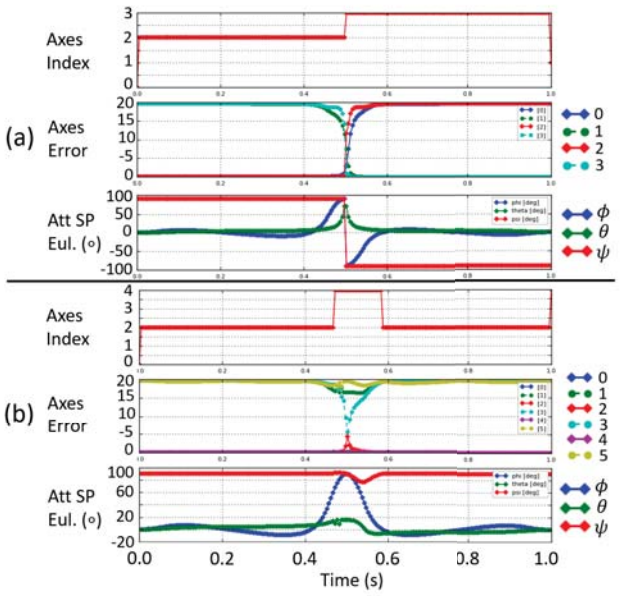

Fig. 7. Closest axes selection and error between each axes set and the current orientation. Values presented through an aggressive roll maneuver. Two methods compared: (a) Four Axes method, (b) Six Axes method For (a) and (b), **top**: selected axes index, **center**: error for each axes set and **bottom**: resulting orientation as Euler angles. The Four Axes method results in 180° yaw rotation. The Six Axes method does not and has a smaller error throughout. Axes order for errors is: 0: $\mathbf{x}_c$, 1: $\mathbf{x}_c$ negated, 2: $\mathbf{y}_c$, 3: $\mathbf{y}_c$ negated, 4: second angle, 5: second angle negated.

The performance of each of the approaches for a range of challenging scenarios is summarized in Table I. A method is deemed to have failed (noted by the 0 in the table), if there is a discontinuous jump or an extremely rapid, and large, change in attitude. Across the full range of possible dynamics, the Six Axes Combined method is the best approach, being the only method to succeed in all cases and still enable control of yaw.

TABLE I

PERFORMANCE SUMMARY OF DIFFERENTIAL FLATNESS METHODS.

	Test Case							
Method (Paper Section)	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**All**
Standard (II.C)	0	0	0	0	1	0	1	0
Neg. Check [11] (II.B)	0	1	0	1	1	0	1	0
PX4 [16], [22]	1	1	1	1	0	1	0	0
2nd Angle [15] (II.C)	1	1	1	1	1	1	0	0
Check Furthest (III.C)	1	1	1	1	0	0	1	0
Check Current $\mathbf{x}_b$ (III.C)	1	1	0	1	1	0	0	0
Use Current $\mathbf{x}_b$* (III.C)	1	1	1	1	1	1	1	1
Pitch and Roll* (III.E)	1	1	1	1	1	1	1	1
4 Axes Comb. (III.F)	1	1	1	1	1	1	0	0
6 Axes Comb. (III.F)	1	1	1	1	1	1	1	1

For each Method and Test Case, 1 is pass and 0 is fail. **Test Case 1**: Singularity. **2**: Pitch 90° through singularity. **3**: Pitch 90° yaw 5° **4**: Roll through 90° with 180° yaw. **5**: Yaw with thrust in xy plane. **6**: Pitch backwards through 90°. **7**: Pitch at 90° roll, through x axis. *Full control of yaw not possible.

G. Discussion

The analysis of the different methods for dealing with the singularity issues in the quadrotor differential flatness transformation is hoped to provide insight into the limitations of the different approaches, to enable selection of the most appropriate method for a given application.

While the Six Axes Combined method performs the best over the range of scenarios, it comes with approximately three times greater computational expense than the simple, Standard method. If the application for a quadrotor will not have any large expected roll or pitch angles, then the Standard method may be sufficient. If control of yaw is not important in the application, then the Pitch and Roll Only method is a good selection, providing a simple and quick method to compute the required orientation.

For highly aggressive trajectories through 90° and inversion, we choose the Six Axes Combined method and demonstrate it in both simulated and real-world flights in the next section.

IV. APPLICATION TO AGGRESSIVE TRAJECTORIES

The benefit of having a differential flatness transformation that is effective across all orientations, is that it is possible to command highly aggressive trajectories that pass through 90° of pitch or roll. We demonstrate this capability in both simulation and flight with an example aggressive trajectory planned with a minimum snap formulation similar to Bry et al. [12].

A. Software in the Loop Simulation

A sharp pitching maneuver is tested in a software in the loop simulator, RotorS [23], running PX4 with the 3DR Iris quadrotor model from 3DR. The default PX4 transformation is compared to the Six Axes Combined method. The PX4 method fails near to 90° pitch, with a 180° change in the direction of the x axes (see Fig. 8). The graphs show the orientation set-point as a quaternion ($\mathbf{q}_d$) coming out of the differential flatness transformation in the position controller. While the drone does not crash, the behavior is highly undesirable when striving for close tracking of aggressive trajectories. The Six Axes Combined method, in contrast does maintain continuous set-points throughout

B. Flight Experiments

Various flight experiments were performed using our custom "TORQ" 250 mm quadrotor. The TORQ quadrotor uses Qualcomm's Snapdragon Flight platform [24] to execute the PX4 flight controller and a tightly-integrated visual-inertial localization and mapping system (all localization is performed onboard). Minimum-snap polynomial trajectories [11] are flown using the hierarchical tracking controller shown in Fig. 2. Our system is demonstrated in this video: `https://youtu.be/SrqrGweKQAU`, while videos of the flight experiments described here can be found here: `https://youtu.be/M-1jA1KCqb8`.

To evaluate the various methods for computing the differential flatness transformation, fast trajectories were planned between three waypoints, as shown in Fig. 1. Three methods were tested: a) the standard PX4 method, b) the Four Axes Combined method and c) the Six Axes Combined

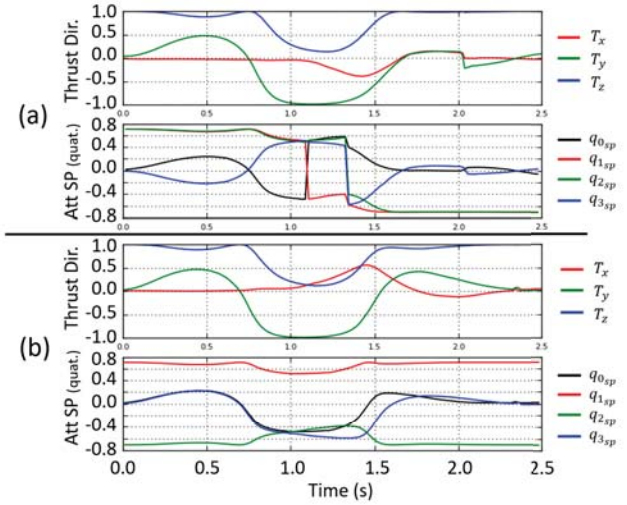

Fig. 8. Software in the loop simulation results for pitching trajectory. (a) using the standard PX4 controller, showing discontinuous jumps. (b) using the Six Axes Combined method. For both (a) and (b): **top** is the thrust set-point and the **bottom** is the output attitude set-point (yaw is constant).

method. Fig. 9 plots the results from these flight experiments, including the input to the differential flatness transformation: the thrust set-point (in addition to a constant yaw of 90°), and the output: the desired attitude as a quaternion set-point. Discontinuities are visible in the quaternion set-points for both the PX4 and Four Axes Combined methods, which is not ideal when accurate tracking is desired.

The Four Axes Combined method also has some discontinuous jumps, an interesting result, given that the planned trajectory worked smoothly for that method (Fig. 10). Flying the trajectory introduces more variables, with tracking errors and disturbances changing the desired thrust that is output from the position controller. An important point here is that even though the problem areas for a transformation method might not be expected to be encountered, disturbances from the nominal trajectory could push the quadrotor into such areas.

A limitation that is evident in the flight results is that it is possible for the yaw to move to 180 degrees of error. At orientations of 90° roll or pitch, a yaw of 0° and 180° become equivalent. If the 180° pathway is taken, it will continue to be tracked because the methods are checking against the last orientation (which is now at 180°). Nonetheless, the Six Axes Combined method enables highly aggressive trajectories to be tracked, such as demonstrated in Fig. 11, in a pitching maneuver beyond 90° where the thrust vector passes below the xy plane (z component of the thrust vector less than zero). Running the standard PX4 transformation on the same thrust set-point produces multiple discrete jumps in orientation (Fig. 11.c).

These flight test results show that the types of orientations discussed in Section III that cause differential flatness transformations to fail, can indeed occur in flight when attempting aggressive maneuvers. Hence it is important to employ an adequate transformation method for applications striving for aggressive flight.

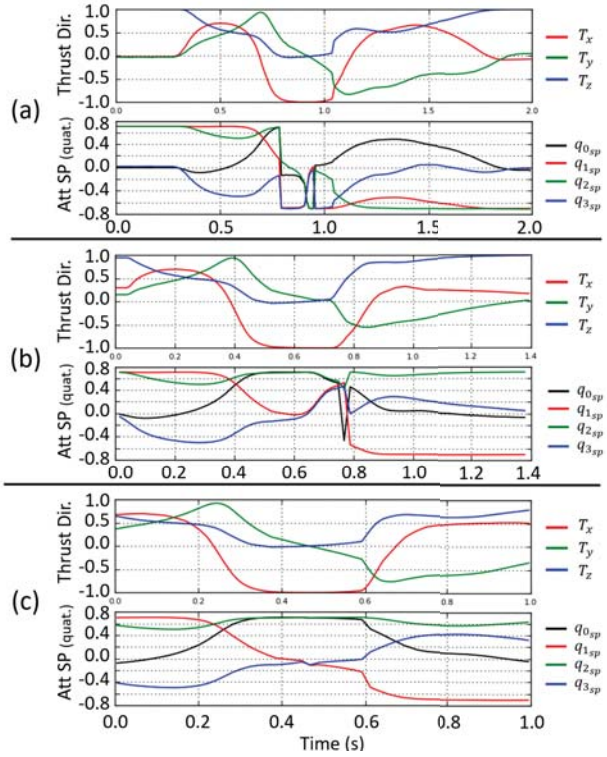

Fig. 9. Aggressive trajectory flight results for three differential flatness transformations: (a) standard PX4 method. (b) Four Axes Combined method, (c) Six Axes Combined method. In each of (a), (b), and (c): **top**: thrust direction set-point, **bottom**: quaternion set-point. These values are the input and output, respectively, of the differential flat transformation.

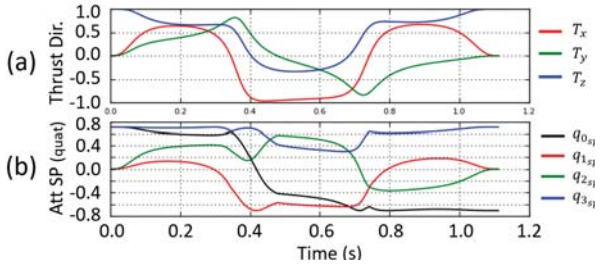

Fig. 10. Planned aggressive trajectory acceleration and corresponding attitude set points as quaternions. (a) Acceleration for the trajectory (b) Quaternion from Four Axes Combined method.

V. CONCLUSIONS

Autonomous navigation systems for quadrotors are becoming more and more capable, with many of the systems being developed relying on the quadrotor's differential flatness property in trajectory planners and controllers. The differential flatness property allows trajectories to be planned in the flat output space of x, y, z and yaw, and facilitates the link between an outer loop position controller and an inner loop attitude controller. A key aspect of the differential flatness property is the transformation from flat outputs to the flat inputs of the motor RPMs squared.

With a push towards highly aggressive flight, progressing through 90° pitch or roll, and inverted flight, the commonly used transformation becomes susceptible to the singularity when the desired thrust vector aligns with the desired heading

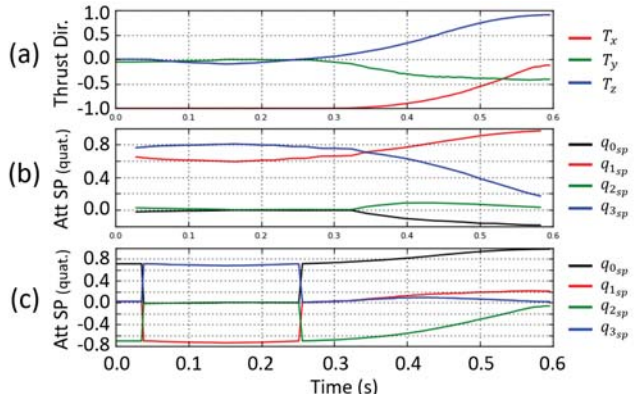

Fig. 11. Flight results from highly aggressive trajectory pitching beyond 90°. (a) Thrust direction for the trajectory (b) Quaternion set-point from controller in flight, using Six Axes Combined method (c) Quaternion from PX4 method applied to the flight data.

in the xy plane. Numerous methods have been proposed to deal with the singularity, but each have limitations arising from sensitivities *around* the singularity, and in transitions near it; these can cause more issues than the singularity itself.

An analysis of existing and newly proposed transformation methods highlights the limitations that are present. A new method that checks six body axes sets against the previously computed axes is shown to be the most stable method through all orientations. A susceptibility exists, however, for the yaw to have a 180° error after completing aggressive maneuvers. Different methods may be more suitable for a given application, hence the analysis presented here may serve as a resource for others to understand the characteristics and limitations of the various methods. Simulation and flight tests with aggressive trajectories show that the problematic flight conditions for the transformations *can* be experienced and *do* cause issues, demonstrating a need to carefully consider the differential flatness transformation method when designing an quadrotor system for highly-aggressive flight.

ACKNOWLEDGMENT

This research was carried out at the Jet Propulsion Laboratory, California Institute of Technology, and was sponsored in part by Google. Morrell and Rigter were visiting scholars supported by the University of Sydney, Australia. The authors wish to thank Russell Smith, Dillon Azzam, Thomas Paynter and Larry Matthies for their hardware and programmatic support.

REFERENCES

[1] S. Gupte, P. I. T. Mohandas, and J. M. Conrad, "A survey of quadrotor unmanned aerial vehicles," in *Southeastcon, 2012 Proceedings of IEEE*. IEEE, 2012, pp. 1–6.

[2] Z. Fang, S. Yang, S. Jain, G. Dubey, S. Roth, S. Maeta, S. Nuske, Y. Zhang, and S. Scherer, "Robust autonomous flight in constrained and visually degraded shipboard environments," *Journal of Field Robotics*, vol. 34, no. 1, pp. 25–52, 2017.

[3] D. Droeschel, M. Nieuwenhuisen, M. Beul, D. Holz, J. Stückler, and S. Behnke, "Multilayered mapping and navigation for autonomous micro aerial vehicles," *Journal of Field Robotics*, vol. 33, no. 4, pp. 451–475, 2016.

[4] M. Faessler, F. Fontana, C. Forster, E. Mueggler, M. Pizzoli, and D. Scaramuzza, "Autonomous, vision-based flight and live dense 3d mapping with a quadrotor micro aerial vehicle," *Journal of Field Robotics*, vol. 33, no. 4, pp. 431–450, 2016.

[5] D. Mellinger, N. Michael, and V. Kumar, "Trajectory generation and control for precise aggressive maneuvers with quadrotors," *The International Journal of Robotics Research*, vol. 31, no. 5, pp. 664–674, 2012.

[6] R. M. Murray, M. Rathinam, and W. Sluis, "Differential flatness of mechanical control systems: A catalog of prototype systems," in *ASME international mechanical engineering congress and exposition*, 1995.

[7] M. Van Nieuwstadt and R. M. Murray, "Real time trajectory generation for differentially flat systems," *IFAC Proceedings Volumes*, vol. 29, no. 1, pp. 2301–2306, 1996.

[8] R. Mahony, V. Kumar, and P. Corke, "Multirotor aerial vehicles," *IEEE Robotics and Automation magazine*, vol. 20, no. 32, 2012.

[9] V. Kumar and N. Michael, "Opportunities and challenges with autonomous micro aerial vehicles," *The International Journal of Robotics Research*, vol. 31, no. 11, pp. 1279–1291, 2012.

[10] T. Lee, M. Leoky, and N. H. McClamroch, "Geometric tracking control of a quadrotor uav on se (3)," in *Decision and Control (CDC), 2010 49th IEEE Conference on*. IEEE, 2010, pp. 5420–5425.

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

[12] A. Bry, C. Richter, A. Bachrach, and N. Roy, "Aggressive flight of fixed-wing and quadrotor aircraft in dense indoor environments," *The International Journal of Robotics Research*, vol. 34, no. 7, pp. 969–1002, 2015.

[13] R. E. Allen and M. Pavone, "A real-time framework for kinodynamic planning with application to quadrotor obstacle avoidance," Ph.D. dissertation, Stanford University, 2016.

[14] G. Loianno, C. Brunner, G. McGrath, and V. Kumar, "Estimation, control, and planning for aggressive flight with a small quadrotor with a single camera and imu," *IEEE Robotics and Automation Letters*, vol. 2, no. 2, pp. 404–411, 2017.

[15] J. Thomas, M. Pope, G. Loianno, E. W. Hawkes, M. A. Estrada, H. Jiang, M. R. Cutkosky, and V. Kumar, "Aggressive flight with quadrotors for perching on inclined surfaces," *Journal of Mechanisms and Robotics*, vol. 8, no. 5, p. 051007, 2016.

[16] M. Faessler, F. Fontana, C. Forster, and D. Scaramuzza, "Automatic re-initialization and failure recovery for aggressive flight with a monocular vision-based quadrotor," in *Robotics and Automation (ICRA), 2015 IEEE International Conference on*. IEEE, 2015, pp. 1722–1729.

[17] M. Neunert, C. de Crousaz, F. Furrer, M. Kamel, F. Farshidian, R. Siegwart, and J. Buchli, "Fast nonlinear model predictive control for unified trajectory optimization and tracking," in *Robotics and Automation (ICRA), 2016 IEEE International Conference on*. IEEE, 2016, pp. 1398–1404.

[18] D. Falanga, E. Mueggler, M. Faessler, and D. Scaramuzza, "Aggressive quadrotor flight through narrow gaps with onboard sensing and computing using active vision," in *Proc. of the IEEE International Conference on Robotics and Automation (ICRA)*, 2017.

[19] S. Lupashin, A. Schöllig, M. Sherback, and R. D'Andrea, "A simple learning strategy for high-speed quadrocopter multi-flips," in *Robotics and Automation (ICRA), 2010 IEEE International Conference on*. IEEE, 2010, pp. 1642–1648.

[20] P. Abbeel, A. Coates, and A. Y. Ng, "Autonomous helicopter aerobatics through apprenticeship learning," *The International Journal of Robotics Research*, vol. 29, no. 13, pp. 1608–1639, 2010.

[21] E. Frazzoli, M. A. Dahleh, and E. Feron, "A hybrid control architecture for aggressive maneuvering of autonomous helicopters," in *Decision and Control, 1999. Proceedings of the 38th IEEE Conference on*, vol. 3. IEEE, 1999, pp. 2471–2476.

[22] L. Meier, P. Tanskanen, L. Heng, G. H. Lee, F. Fraundorfer, and M. Pollefeys, "Pixhawk: A micro aerial vehicle design for autonomous flight using onboard computer vision," *Autonomous Robots*, vol. 33, no. 1-2, pp. 21–39, 2012.

[23] F. Furrer, M. Burri, M. Achtelik, and R. Siegwart, "Rotorsa modular gazebo mav simulator framework," in *Robot Operating System (ROS)*. Springer, 2016, pp. 595–625.

[24] Qualcomm, "Snapdragon flight 801 processor," 2017. [Online]. Available: https://developer.qualcomm.com/hardware/snapdragon-flight