An Interactive Tool for Designing Quadrotor Camera Shots
Supplementary Material

Niels Joubert* Stanford University
Mike Roberts* Stanford University
Anh Truong Stanford University
Floraine Berthouzoz Adobe Research
Pat Hanrahan Stanford University

1 Deriving the Quadrotor Camera Manipulator Matrices

In this section, we derive the manipulator matrices for our quadrotor camera model. At a high level, our strategy will be to relate our quadrotor camera's degrees of freedom to our control inputs.

In the derivation that follows, we assume that our quadrotor and gimbal are *kinematically coupled* [Kondak et al. 2013], in the sense that moving the position of the quadrotor also moves the position of the gimbal. However, we assume that our quadrotor and gimbal are not *dynamically coupled* [Kondak et al. 2013], in the sense that torques acting on the gimbal do not induce reactive torques on the quadrotor. These assumptions simplify the modeling of our system. Moreover, we feel they are justified, since in the cameras mounted on quadrotors tend to be very lightweight relative to the quadrotors themselves. For example, on our hardware platform, our camera and gimbal are roughly $25\times$ lighter than our quadrotor.

Relating the Quadrotor's Position to the Control Inputs We assume that there are two forces acting on our quadrotor camera: (1) a *thrust force* induced by the quadrotor's propellers; and (2) an *external force* that models any other forces acting on the quadrotor, such as gravity, wind, and drag. Based on this assumption, we use Newton's Second Law to relate the linear acceleration of the quadrotor in the world frame, $\ddot{\mathbf{p}}$, to the control input applied at each of the quadrotor's propellers, $\mathbf{u}_q$, as follows,

$$m\ddot{\mathbf{p}} = \mathbf{f}_e + \mathbf{R}_{\mathcal{W},\mathcal{Q}}\mathbf{M}_\mathbf{f}\mathbf{u}_q \tag{1}$$

where m is the mass of the quadrotor camera; $\mathbf{f}_e$ is the external force; $\mathbf{M}_\mathbf{f}$ is the matrix that maps the control input at each of the quadrotor's propellers into a net thrust force oriented along the quadrotor's local $\mathbf{y}$ axis; and $\mathbf{R}_{\mathcal{W},\mathcal{Q}}$ is the rotation matrix that represents the quadrotor's orientation in the world frame (i.e., the rotation matrix that maps vectors from the body frame of the quadrotor into the world frame). We define $\mathbf{M}_\mathbf{f}$ as follows,

$$\mathbf{M}_\mathbf{f} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix} \tag{2}$$

Relating the Quadrotor's Angular Acceleration and Angular Velocity to the Control Inputs We use Euler's Second Law to relate the angular acceleration and angular velocity of the quadrotor in the body frame, $\dot{\omega}_q$ and ω_q respectively, to the control input applied at each of the quadrotor's propellers, $\mathbf{u}_q$, as follows,

$$\mathbf{I}_q\dot{\omega}_q + \omega_q \times \mathbf{I}_q\omega_q = \mathbf{M}_\tau\mathbf{u}_q \tag{3}$$

where $\mathbf{I}_q$ is the inertia matrix of the quadrotor camera; and $\mathbf{M}_\tau$ is the matrix that maps the control input at each of the quadrotor's propellers into a net torque acting on the quadrotor in the body frame. We define $\mathbf{M}_\tau$ as follows,

$$\mathbf{M}_\tau = \begin{bmatrix} ds_\alpha & ds_\beta & -ds_\beta & -ds_\alpha \\ \gamma & -\gamma & \gamma & -\gamma \\ -dc_\alpha & dc_\beta & dc_\beta & -dc_\alpha \end{bmatrix} \tag{4}$$

*Niels Joubert and Mike Roberts contributed equally to this work.

where d, α, β, and γ are constants related to the physical design of a quadrotor: d is the distance from the quadrotor's center of mass to it's propellers; α is the angle in radians that the quadrotor's front propellers form with the quadrotor's positive $\mathbf{x}$ axis; β is the angle in radians that the quadrotor's rear propellers form with the quadrotor's negative $\mathbf{x}$ axis; γ is the magnitude of the in-plane torque generated by the quadrotor propeller producing 1 unit of upward thrust force; $c_a = \cos a$ and $s_a = \sin a$. Our definition for $\mathbf{M}_\tau$ assumes: (1) the quadrotor's propellers are co-planar with its center of mass; and (2) a constant relationship between the magnitude of the in-plane torque generated by the quadrotor propeller, and the magnitude of the upward thrust force generated by the propeller.

Relating the Gimbal's Angular Acceleration to the Control Inputs We assume that our 3 degree-of-freedom gimbal is fully actuated, and has very large actuator limits. Since fully actuated systems are *feedback equivalent* [Tedrake 2014] to double integrator systems, and we assume that our gimbal has very large actuator limits, we model the gimbal as a double integrator for simplicity. We relate the angular acceleration of the gimbal in the body frame of the quadrotor, $\dot{\omega}_g$ to the *feedback linearized* [Tedrake 2014] control input applied at the gimbal, $\mathbf{u}_g$, as follows,

$$\dot{\omega}_g = \mathbf{u}_g \tag{5}$$

Relating Euler Angle Time Derivatives to Angular Velocity and Angular Acceleration At this point, we have related the angular velocities and accelerations of our system to our control inputs. However, to derive the complete equations of motion for our system, we need to relate the Euler angle time derivatives of our system to our control inputs. To do this, we must relate Euler angle time derivatives to angular velocities and angular accelerations.

Let $\mathbf{e} = [\theta\ \psi\ \phi]^T$ be a vector of our Euler angles. Let us define the rotation matrix $\mathbf{R}$ in terms of the Euler angles θ, ψ, and ϕ as follows,

$$\begin{aligned}\mathbf{R} &= \mathbf{R}_\mathbf{y}^\psi\mathbf{R}_\mathbf{z}^\theta\mathbf{R}_\mathbf{x}^\phi \\ &= \begin{bmatrix} c_\psi & 0 & s_\psi \\ 0 & 1 & 0 \\ -s_\psi & 0 & c_\psi \end{bmatrix}\begin{bmatrix} 1 & 0 & 0 \\ 0 & c_\theta & -s_\theta \\ 0 & s_\theta & c_\theta \end{bmatrix}\begin{bmatrix} c_\phi & -s_\phi & 0 \\ s_\phi & c_\phi & 0 \\ 0 & 0 & 1 \end{bmatrix}\end{aligned} \tag{6}$$

where $c_a = \cos(a)$ and $s_a = \sin(a)$.

We can straightforwardly compute the time derivative of this expression to get an (admittedly unpleasant) expression for $\dot{\mathbf{R}}$ in terms of our Euler angles and their time derivatives. We omit this step for brevity.

We make the observation that $\dot{\mathbf{R}} = (\omega)_\times\mathbf{R}$, where ω is the angular velocity of a rotating body in the non-rotating frame; and the notation $(\mathbf{a})_\times$ refers to the skew-symmetric matrix, computed as a function of the vector $\mathbf{a}$, such that $(\mathbf{a})_\times\mathbf{b} = \mathbf{a} \times \mathbf{b}$ for all vectors $\mathbf{b}$.

From the above expression, we immediately get $\dot{\mathbf{R}}\mathbf{R}^T = (\omega)_\times$. We observe that the entries of $\dot{\mathbf{R}}\mathbf{R}^T$ are linear in $\dot{\psi}$, $\dot{\theta}$, and $\dot{\phi}$. Therefore, there is a matrix $\mathbf{A}$ that relates $\dot{\mathbf{e}}$ to ω as follows,

$$\mathbf{A}\dot{\mathbf{e}} = \omega \tag{7}$$

We can take the time derivative of both sides of this expression using the product rule to get the following expression,

$$\dot{\mathbf{A}}\dot{\mathbf{e}} + \mathbf{A}\ddot{\mathbf{e}} = \dot{\omega} \tag{8}$$

We define the matrix $\mathbf{A}$ that relates $\dot{\mathbf{e}}$ to ω according to the linear relationship $\mathbf{A}\dot{\mathbf{e}} = \omega$, and its time derivative $\dot{\mathbf{A}}$, as follows,

$$\begin{aligned} \mathbf{A} &= \begin{bmatrix} c_\psi & 0 & s_\psi c_\theta \\ 0 & 1 & -s_\theta \\ -s_\psi & 0 & c_\psi c_\theta \end{bmatrix} \\ \dot{\mathbf{A}} &= \begin{bmatrix} -s_\psi \dot{\psi} & 0 & -s_\psi s_\theta \dot{\theta} + c_\psi c_\theta \dot{\psi} \\ 0 & 0 & -c_\theta \dot{\theta} \\ -c_\psi \dot{\psi} & 0 & -s_\psi c_\theta \dot{\psi} + s_\theta c_\psi \dot{\theta} \end{bmatrix} \end{aligned} \tag{9}$$

where $c_a = \cos a$ and $s_a = \sin a$.

Relating the Quadrotor's Orientation to the Control Inputs We can rotate equations (7) and (8) into the body frame of the quadrotor, and substitute them into equation (3), to get the following expression for the quadrotor's rotational dynamics,

$$\begin{aligned} \mathbf{I}_q(\mathbf{R}_{\mathcal{Q},\mathcal{W}}\dot{\mathbf{A}}_q\dot{\mathbf{e}}_q + \mathbf{R}_{\mathcal{Q},\mathcal{W}}\mathbf{A}_q\ddot{\mathbf{e}}_q) + \\ \mathbf{R}_{\mathcal{Q},\mathcal{W}}\mathbf{A}_q\dot{\mathbf{e}}_q \times \mathbf{I}_q\mathbf{R}_{\mathcal{Q},\mathcal{W}}\mathbf{A}_q\dot{\mathbf{e}}_q = \mathbf{M}_\tau\mathbf{u}_q \end{aligned} \tag{10}$$

where $\mathbf{e}_q$ is the vector of Euler angles representing the quadrotor's orientation in the world frame; $\mathbf{R}_{\mathcal{Q},\mathcal{W}}$ is the rotation matrix that maps vectors from the world frame into the body frame of the quadrotor; and $\mathbf{A}_q$ is the matrix that relates the quadrotor's Euler angle time derivatives to its angular velocity in the world frame.

Relating the Gimbal's Orientation to the Control Inputs Similarly to our approach in the previous subsection, we can substitute equation (8) into equation (5) to get the following expression for the gimbal's rotational dynamics,

$$\dot{\mathbf{A}}_g\dot{\mathbf{e}}_g + \mathbf{A}_g\ddot{\mathbf{e}}_g = \mathbf{u}_g \tag{11}$$

where $\mathbf{e}_g$ is the vector of Euler angles representing the orientation of the gimbal in the body frame of the quadrotor; and $\mathbf{A}_g$ is the matrix that relates the gimbal's Euler angle time derivatives to its angular velocity in the body frame of the quadrotor.

Defining the Manipulator Matrices We define the layout of our degree-of-freedom vector $\mathbf{q}$, and our control vector $\mathbf{u}$, as follows,

$$\mathbf{q} = \begin{bmatrix} \mathbf{p} \\ \mathbf{e}_q \\ \mathbf{e}_g \end{bmatrix} \quad \mathbf{u} = \begin{bmatrix} \mathbf{u}_q \\ \mathbf{u}_g \end{bmatrix} \tag{12}$$

Based on this layout, we can express equations (1), (10), and (11) in manipulator form. In doing so, we get the following expressions for our quadrotor camera manipulator matrices,

$$\begin{aligned} \mathbf{H}(\mathbf{q}) &= \begin{bmatrix} m\mathbf{I}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & \mathbf{I}_q\mathbf{R}_{\mathcal{Q},\mathcal{W}}\mathbf{A}_q & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{A}_g \end{bmatrix} \\ \mathbf{C}(\mathbf{q},\dot{\mathbf{q}}) &= \\ &\begin{bmatrix} \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & \mathbf{I}_q\mathbf{R}_{\mathcal{Q},\mathcal{W}}\dot{\mathbf{A}}_q - \left(\mathbf{I}_q\mathbf{R}_{\mathcal{Q},\mathcal{W}}\mathbf{A}_q\dot{\mathbf{e}}_q\right)_\times \mathbf{R}_{\mathcal{Q},\mathcal{W}}\mathbf{A}_q & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \dot{\mathbf{A}}_g \end{bmatrix} \\ \mathbf{G}(\mathbf{q}) &= \begin{bmatrix} -\mathbf{f}_e \\ \mathbf{0}_{3\times1} \\ \mathbf{0}_{3\times1} \end{bmatrix} \\ \mathbf{B}(\mathbf{q}) &= \begin{bmatrix} \mathbf{R}_{\mathcal{W},\mathcal{Q}}\mathbf{M}_\mathbf{f} & \mathbf{0}_{3\times3} \\ \mathbf{M}_\tau & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times4} & \mathbf{I}_{3\times3} \end{bmatrix} \end{aligned} \tag{13}$$

where $\mathbf{0}_{p\times q}$ is the $p \times q$ zero matrix; and $\mathbf{I}_{k\times k}$ is the $k \times k$ identity matrix. In our definition for the manipulator matrices above, we assume that $\mathbf{f}_e$ can depend on $\mathbf{q}$, but cannot depend on $\dot{\mathbf{q}}$. We make this assumption for simplicity, although it could be relaxed by making minor modifications to $\mathbf{C}$ and $\mathbf{G}$ above.

2 Proof that B is Always Full Column Rank

To prove that the matrix $\mathbf{B}$ is full column rank, consider the following matrix,

$$\mathbf{M} = \begin{bmatrix} \mathbf{R}_{\mathcal{W},\mathcal{Q}}\mathbf{M}_\mathbf{f} \\ \mathbf{M}_\tau \end{bmatrix} \tag{14}$$

It will suffice to show that $\mathrm{rank}(\mathbf{M}) = 4$. We assume without loss of generality that $0 < \alpha, \beta < \frac{\pi}{2}$ and that $d, \gamma > 0$. We can clearly see that $\mathrm{rank}(\mathbf{M}_\tau) = 3$.

Suppose for the sake of contradiction that $\mathrm{rank}(\mathbf{M}) = 3$. Then we must be able to represent any row of $\mathbf{R}_{\mathcal{W},\mathcal{Q}}\mathbf{M}_\mathbf{f}$ as a linear combination of the rows of $\mathbf{M}_\tau$. Let the row $\mathbf{r} = [r\ r\ r\ r]$ be one such row of $\mathbf{R}_{\mathcal{W},\mathcal{Q}}\mathbf{M}_\mathbf{f}$. Representing $\mathbf{r}$ as a linear combination of the rows in $\mathbf{M}_\tau$, we get the following system of equations,

$$\begin{aligned} r = \ & ids_\alpha & + j\gamma & - kdc_\alpha \\ r = \ & ids_\beta & - j\gamma & + kdc_\beta \\ r = \ & -ids_\beta & + j\gamma & + kdc_\beta \\ r = \ & -ids_\alpha & - j\gamma & - kdc_\alpha \end{aligned} \tag{15}$$

for some i, j, k. Equating the first and last of these equations above, we get $ids_\alpha = -j\gamma$. Equating the middle two of these equations above, we get $ids_\beta = j\gamma$. Substituting these two new expressions into the first two equations above, we get $c_\alpha = -c_\beta$. However, this is only possible if α or β is outside the range $(0, \frac{\pi}{2})$. Since we had previously assumed that α and β are both inside the range $(0, \frac{\pi}{2})$, we have arrived at a contradiction. This completes our proof.

3 Empirical Justification for Minimizing the 4th Derivative of 7th Degree Polynomials

In this section, we evaluate alternative polynomial representations for quadrotor camera paths. We find that minimizing the 4th derivative of 7th degree piecewise polynomials produces the smoothest and most reasonably bounded control signals for quadrotors (see Figures 1 and 2 in this document).

When evaluating alternative polynomial representations, we chose to limit our experiments to odd degree (i.e., even order) polynomials, since these tend to produce smoother curves than even degree (i.e., odd order) polynomials [Goshtasby et al. 1990].

References

GOSHTASBY, A., CHENG, F., AND BARSKY, B. A. 1990. B-spline curves and surfaces viewed as digital filters. *Computer Vision, Graphics, and Image Processing 52*, 2.

KONDAK, K., KRIEGER, K., ALBU-SCHAEFFER, A., SCHWARZBACH, M., LAIACKER, M., MAZA, I., RODRIGUEZ-CASTANO, A., AND OLLERO, A. 2013. Closed-loop behavior of an autonomous helicopter equipped with a robotic arm for aerial manipulation tasks. *International Journal of Advanced Robotic Systems 10*, 145.

TEDRAKE, R., 2014. Underactuated robotics: Algorithms for walking, running, swimming, flying, and manipulation (course notes for MIT 6.832). `http://people.csail.mit.edu/russt/underactuated/`.

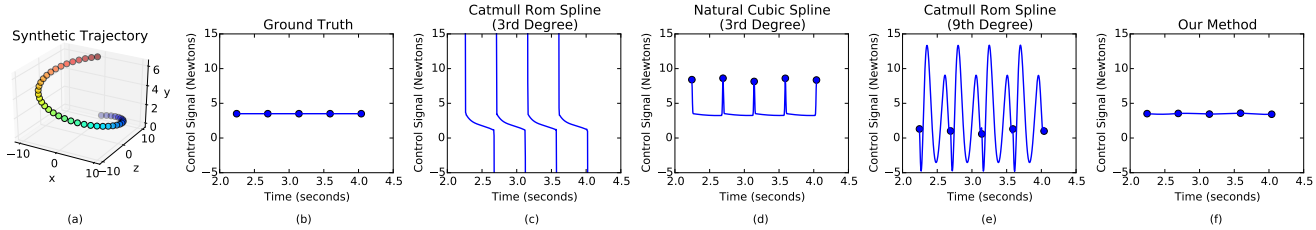

Figure 1: *Comparison of the quadrotor control signals resulting from different keyframe interpolation methods. We construct a simple synthetic traejectory (a). We use the control signals required for a quadrotor to follow the synthetic trajectory as ground truth (b). We sample the synthetic trajectory with 15 evenly spaced keyframes, and we construct an interpolated trajectory from these keyframes using different interpolation methods. We plot the control signals produced by each interpolation method. We require that the control signals be continuous, and we prefer control signals that are as close as possible to the ground truth. We show the middle of these control signal plots to highlight their periodic behavior without boundary artifacts. For each interpolation method, we show the control signal for the front right propeller, and we indicate the control signal value at each keyframe with a blue dot. 3^{rd} degree Catmull Rom Splines are C^1 continuous, and therefore produce discontinuous control signals (c). Natural Cubic Splines are C^2 continuous, and therefore also produce discontinuous control signals (d). 9^{th} degree Catmull Rom Splines are C^4 continuous, so they produce continuous control signals, but with large periodic excursions (e). Our method is C^4 continuous and produces control signals with minimal excursions (f).*

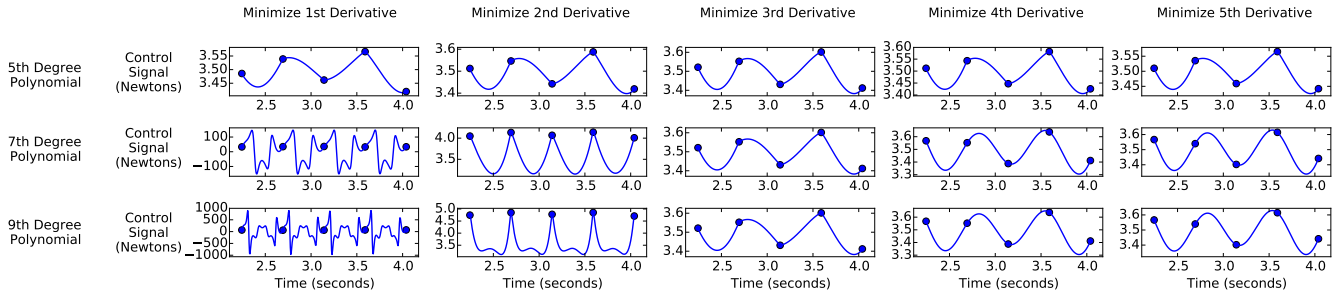

Figure 2: *Comparison of the quadrotor control signals resulting from different variations of our method when interpolating the synthetic trajectory described in Figure 1. We show the middle of these control signal plots to highlight their periodic behavior without boundary artifacts. For each variation of our method, we show the control signal for the front right propeller, and we indicate the control signal value at each keyframe with a blue dot. Note that the vertical scaling varies in each subplot, and is different from the vertical scaling in Figure 1. Minimizing the 4^{th} derivative of 7^{th} degree polynomials (also shown, with different vertical scaling, in Figure 1f in this document) is the simplest variation of our method that produces smooth and reasonably bounded control signals. Note that C^4 continuous position trajectories are only guaranteed to produce C^0 continuous control signals for quadrotors, which is why some of the control signals in this figure are jagged.*