Attitude Control for a Vehicle in Free Fall

Brian Bittner, Quan Nguyen

Abstract—Wheeled systems are broadly used throughout field robotics. Path planning for these systems is straightforward, but the dynamics are difficult to predict in the presence of contact forces that are highly dependent on surface texture and geometry. The nature of field robotics is to interact in unstructured environments, so sensory information is used to compensate for dynamic uncertainties. Pose estimation systems are generally implemented with stereo vision, inertial measurement units, and Lidar. The gravitational field not only causes these unpredictable contact forces, but invalidates symmetries of the vehicle’s Lagrangian. Albeit a rare occurrence, free fall offers an opportunity to exploit geometric symmetries of wheeled systems. Here we propose a method for attitude control of a vehicle in free fall. The shape space of our vehicle will consist of the angular position of rotors, which provide the system’s actuated degrees of freedom. In order of increasing complexity, we model and control a planar vehicle, followed by a fully-actuated and under-actuated 3D vehicle. Coordinate optimization for the underactuated case allows us to design gaits that obtain net rotation aligned with the lie bracket motion via sinusoidal steering. These insights combined with geometric control provide a basis for attitude control of an underactuated 3D vehicle in free fall.

I. INTRODUCTION

Control of rigid bodies in SO(3) is a topic that has been approached using a broad range of techniques and system representations. The nonlinearity of transformations along this group manifold require sophisticated control architectures and careful consideration of system properties. Geometric control has emerged as a powerful tool for controlling systems composed of rigid bodies. Controlling behavior directly on SO(3) enables cooperative load transportation by quadrotors [1] and bipedal locomotion, as shown by Atrias. [2] For this work, we look to extend such geometric techniques toward attitude control of an underactuated vehicle in free fall.

This work is inspired by Boston Dynamic’s SandFlea robot, which demonstrated pitch control during substantial free fall (from the roof of a building.) To control a fully actuated vehicle, geometric control is perfectly sufficient. In this paper, we present control of a fully actuated vehicle state in SO(3) and can ensure convergence to a target orientation prior to contact with the ground.

Conversely, underactuated control of SO(3) cannot be achieved with continuous, time invariant feedback. To extend to the underactuated case, further insight is needed from geometric mechanics. Particularly we look at Sastry’s shape input dynamics for satellites [3] and Travers derivation of coordinate choice for sinusoidal steering [4]. Such prior work allow us to execute large amplitude motions to achieve net rotation in the underactuated degree of freedom, while minimizing nonconservative contribution to rotation in the actuated degrees of freedom.

This work makes the following contributions:

- Geometric Control has yet to be applied to vehicular dynamics.
- Implementation on a real vehicle would extend successful pitch control of Boston Dynamics’ Sand Flea by offering full control over SO(3).
- Optimization of a sinusoidal steering policy complements standard geometric control for the underactuated vehicle.

In this paper we build our control up from the simplest case to the most complex. In Section II we discuss a pitch controller for a planar vehicle. In Section III we apply geometric control to a vehicle in 3D. In Section IV we present our controller for an underactuated vehicle. Section V discusses future work.

II. PLANAR VEHICLE MODEL

A. System Dynamics

For this system and all subsequent systems, invariance in Euclidean space is a key to the simplification of the vehicle dynamics. Because the center of mass (COM) of each vehicle does not change with respect to the vehicle’s rotary shape inputs, the translational dynamics are entirely decoupled from the rotary dynamics. The vehicle’s translational dynamics involve simple, ballistic trajectories entirely dependent on
initial conditions of the body’s COM and the gravitational field. Rotational dynamics are dependent on initial conditions and inputs from the system’s various rotors.

\[ L = \frac{1}{2} J_B \omega_B^2 + \frac{1}{2} J_1 \omega_1^2 + \frac{1}{2} J_2 \omega_2^2 \]  

(1)

\[ \frac{\partial L}{\partial \theta_B} - \frac{\partial}{\partial t} \left( \frac{\partial L}{\partial \omega_B} \right) = 0 \]  

(2)

\[ \Rightarrow J_B \dot{\omega}_B + J_1 \dot{\omega}_1 + J_2 \dot{\omega}_2 = 0 \]  

(3)

\[ \Rightarrow \dot{\omega}_B = -\frac{1}{J_B} \left[ J_1 \dot{\omega}_1 + J_2 \dot{\omega}_2 \right] \]  

(4)

Wheel dynamics

\[ J_1 \dot{\omega}_1 = u_1 \]  

(5)

\[ J_2 \dot{\omega}_2 = u_2 \]  

(6)

B. Controller

From the model (4), we can apply a PD controller:

\[ \dot{\omega}_B = -\left[ K_P \quad K_D \right] \begin{bmatrix} \theta_B - \theta_B^d \\ \omega_B - \omega_B^d \end{bmatrix} \]  

(7)

C. Simulation

We simulate the system with three different initial conditions:

\[ \begin{bmatrix} x \\ \dot{x} \\ y \\ \dot{y} \\ \theta_B \\ \dot{\theta}_B \\ \omega_B \\ \theta_1 \\ \omega_1 \\ \theta_2 \\ \omega_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 8 \\ 10 \\ 3 \\ \frac{\pi}{4} \\ -\frac{\pi}{2} \\ 0 \\ 0 \\ 20 \\ 0 \\ 20 \end{bmatrix} \]  

(10)

where:

- \((x, \dot{x}, y, \dot{y})\) are position and velocity of the car’s COM in X and Y axis,
- \((\theta_B, \omega_B)\) are angle and angular velocity of the car in the global frame.
Fig. 4: Simulation of planar vehicle: Case 2 ($\theta_B = -\pi/2$)

- $(\theta_1, \omega_1, \theta_2, \omega_2)$ are angles and angular velocities of the two wheels in the global frame.

Fig. 3, 4, 5 show the simulation for these three cases. As you can see from these figures, the attitude trajectories converge exponentially to zero with different initial conditions.

In the figures displaying two position trajectories, the dotted green line represents the COM trajectory, while the dotted blue line represents the trajectory of a point in the front for the car. The difference between these two lines will show us how the car rotates around its COM.

In case 1 (Fig.3) and 3 (Fig.5), we set the initial velocity of the car’s COM in X and Y direction are $\dot{x} = 8 \text{ m/s}$, $\dot{y} = 3 \text{ m/s}$. Therefore, the COM trajectory in these cases is a parabola. While in case 2 (Fig. 4), we assume that the car is dropped with zero velocity, that implies a straight-line trajectory for the COM.

Especially, in case 3 (Fig.5), we run an interesting simulation case, when we set the initial angle of the car $\theta_B = 10\pi$, which means that we want the car must rotate around five times before landing. You can easily observe this behavior from the dotted blue line in Fig.5.

The pitch controller for this planar vehicle has been shown to be successful. We will now extend to a car in 3D, whose orientation acts on $SO(3)$.

III. FULLY-ACTUATED 3D VEHICLE MODEL

A. System Dynamic

The model for fully-actuated 3D vehicle is shown in Fig.6. The rotational dynamics of a rigid body [?] are as follows:

$$J\dot{\Omega} + \Omega \times J\Omega = M = \begin{bmatrix} \dot{u}_x \\ \dot{u}_y \\ \dot{u}_z \end{bmatrix}$$  \hspace{1cm} (11)

$$\dot{R} = R\dot{\Omega}$$  \hspace{1cm} (12)

where

- $J \in \mathbb{R}^{3 \times 3}$ is the inertia matrix of the car body (not including the wheels).
- $\Omega \in \mathbb{R}^{3}$ is the angular velocity of the car in the global frame.
Vehicle.

Fig. 7: Geometric Control Diagram for Fully-actuated 3D Vehicle.

- \( u_x, u_y, u_z \) are moments at 3 wheel axes,
- \( R \in SO(3) \) in the rotational matrix from the global frame to the body frame.

B. Controller

For this problem, we use Geometric Control to drive the system to desired rotation \( R_d \), and angular velocity \( \Omega_d \). This type of controller was presented in [?]. We first need to define configuration errors

\[
e_R = \frac{1}{2} (R_d^T R - R^T R_d)^V, \quad (13)
\]

\[
e_{\Omega} = \Omega - R^T R_d \Omega_d \quad (14)
\]

We have

\[
e_R = 0 \implies R = R_d \quad (15)
\]

\[
R = R_d; e_{\Omega} = 0 \implies \Omega = \Omega_d \quad (16)
\]

where \( R_d; \Omega_d \) are desired orientation and angular velocity.

The following geometric controller will drive the errors \( e_R \) and \( e_{\Omega} \) to zeros:

\[
M = -k_Re_R - k_{\Omega}e_{\Omega} + \Omega \times J\Omega - J(\hat{\Omega}R^T R_d \Omega_d - R^T R_d \hat{\Omega}_d) \quad (17)
\]

\[
- k_Re_R - k_{\Omega}e_{\Omega} + \Omega \times J\Omega - J(\hat{\Omega}R^T R_d \Omega_d - R^T R_d \hat{\Omega}_d) = 0 \quad (18)
\]

The control diagram (Fig.7) will give you a better idea about how the geometric controller (17) works. To be more specific, when we apply the geometric controller (17) to the system dynamic (23), we will have:

- the term \( \Omega \times J\Omega \) in the dynamic and the controller will be canceled out,
- the term \( J\Omega R^T R_d \Omega_d \) in the controller will give us the difference between \( R \) and \( R_d \), since we have \( R = R\hat{\Omega} \),
- the difference between the term \( J\Omega R^T R_d \hat{\Omega}_d \) in the controller and the term \( J\hat{\Omega} \) will give us the difference between \( \Omega \) and \( \hat{\Omega}_d \).
- \( e_R \) represents the difference between \( R \) and \( R_d \),
- \( e_{\Omega} \) represents the difference between \( \Omega \) and \( \hat{\Omega}_d \).

Therefore, if we represent the difference between these terms as "\( -\)", and plug the controller to the dynamics, the geometric controller can be described in a locomotive context:

\[
-k_R(R - R_d) - J(\hat{R} - \hat{R}_d) - k_{\Omega}(\Omega - \Omega_d) - J(\hat{\Omega} - \hat{\Omega}_d) = 0. \quad (19)
\]

We need to notice that the above equation is not mathematically correct, we the symbol "\( -\)" here represents the difference between two term, not simply a mathematical subtraction. However, this way of representation helps us to explain intuitively the principles of geometric control.

Because this geometric control can cancel out the nonlinear dynamics, we can therefore track the target orientation very well, even with large initial error. This property will be shown in the following simulation results.

C. Simulation

We simulate the system with two different initial conditions for roll, pitch, yaw of the car:

\[
\begin{bmatrix}
\theta \\
\phi \\
\psi \\
\end{bmatrix} = \begin{bmatrix}
-\pi/2 \\
\pi \\
\pi/4 \\
\end{bmatrix}; \begin{bmatrix}
-2\pi/3 \\
2\pi/3 \\
\pi/4 \\
\end{bmatrix} \quad (20)
\]

and we want our controller to drive the car to

\[
\begin{bmatrix}
\theta_d \\
\phi_d \\
\psi_d \\
\end{bmatrix} = \begin{bmatrix}
0 \\
0 \\
0 \\
\end{bmatrix} \quad (21)
\]

or

\[
R_d = I = \begin{bmatrix}
1 & 0 & 0 \\
0 & 1 & 0 \\
0 & 0 & 1 \\
\end{bmatrix} \quad (22)
\]

The simulation for these two cases is shown in Fig.8, and Fig.9. The result show that the geometric controller (17) can effectively drive the car to the desired configuration from very large initial errors.

However, this type of geometric control requires a fully actuated system, so that the value of control inputs \( M \) can be arbitrarily assigned by the controller (17). In the next section, we will introduce a solution for control of an underactuated vehicle.
IV. Under-actuated 3D Vehicle Model

A. System Dynamic

The fully actuated and underactuated system dynamics are identical for these systems, except for the roll rotor’s contribution. It’s absence in the underactuated system means no input torque can directly induce a roll in the system’s body frame. The lack of contribution produces:

\[ J\dot{\Omega} + \Omega \times J\Omega = M = \begin{bmatrix} 0 \\ u_y \\ u_z \end{bmatrix} \]  \hspace{1cm} (23)

\[ \dot{R} = R\hat{\Omega} \]  \hspace{1cm} (24)

B. Controller

The geometric controller from the previous system fails in this case, due to an inability to stabilize the system’s roll. Our approach to handling this inadequacy is to first resolve this roll error by obtaining a net rotation in the unactuated degree of freedom, then switching to the geometric controller to resolve pitch and yaw error before the vehicle lands.

It is important to note that because there are only two rotors, there is no ability to control along a trajectory of states in SO(3). There is one target state in SO(3) which the system must stabilize to before contact. As in the previous cases, this is the goal of the controller.

Next we will discuss a method for obtaining net rotation in our body roll frame.

1) Sinusoidal Steering in SO(3): Sinusoidal steering has been used effectively for locomotion strategies, such as snake sidewinding. Our goal is to use such insights to obtain our lie bracket motion in a planning environment conducive to large amplitude oscillations. Connection Vector Fields and Height functions offer key insight to determining the proper strategy.

Recent work has suggested a coordinate frame \( \beta \) in SO(3) which minimizes the nonconservative contribution of gaits to displacement in the system’s degrees of freedom. \( \beta \), a function of shape inputs \( \alpha_1 \) and \( \alpha_2 \), provide a frame in which to obtain net roll displacement while minimizing perturbations in yaw and roll. Sinusoidal steering in frame \( \beta \) will allow us to directly achieve target roll, which will form a discontinuous, time-variant component of our controller.

Next we will show a procedure to arrive at a state dependent transformation \( \beta \) which minimizes perturbations for our system’s yaw and pitch.

2) Finding Optimal Coordinate Frame Beta for Various Inertial Systems: Recent work on minimum perturbations on SO(3) [4] provides the reconstruction equation for transformation is SO(3) \( \beta \) and the cost function to describe perturbations across the shape input space. The reconstruction equation must be linearized to the following form, since transformations in SO(3) are unavoidably nonlinear.
\[ \xi_{new} = (I - \dot{\beta})\xi_{old} + \beta \]  
\[ \xi_{old} = -A_{new}(r)p \begin{bmatrix} 0 & 0 \\ -J_y & 0 \\ 0 & -J_z \end{bmatrix} \begin{bmatrix} \dot{\alpha}_1 \\ \dot{\alpha}_2 \end{bmatrix} \]  
\[ (25) \]

Here the cost function presents a summation of the squared perturbation integrals across the roll, pitch, and yaw connection vector fields.

\[ J = \int \int_{\Omega}||\vec{A}_\xi \dot{\beta}_z - \beta_y \vec{A}_\xi \dot{\beta}_x + \nabla \beta_z||^2 \delta \Omega \]
\[ + \int \int_{\Omega}||\vec{A}_\xi \dot{\beta}_y - \beta_z \vec{A}_\xi \dot{\beta}_x + \nabla \beta_y||^2 \delta \Omega \]
\[ + \int \int_{\Omega}||\vec{A}_\xi \dot{\beta}_z - \beta_x \vec{A}_\xi \dot{\beta}_y + \nabla \beta_y||^2 \delta \Omega \]  
\[ (27) \]

Transformation \( \beta \) is constructed as a third order polynomial for each of its three rotation elements. A formal derivation of valid \( \beta \) transformations, and how to optimize coordinates for a generalized system are provided here. [5]

\[ \beta_x = w_1 \alpha_1^3 + w_2 \alpha_1^2 + w_3 \alpha_1 + w_4 \alpha_2^3 + w_5 \alpha_2^2 + w_6 \alpha_2 \]  
\[ \beta_y = w_7 \alpha_1^3 + w_8 \alpha_1^2 + w_9 \alpha_1 + w_{10} \alpha_2^3 + w_{11} \alpha_2^2 + w_{12} \alpha_2 \]  
\[ \beta_z = w_{13} \alpha_1^3 + w_{14} \alpha_1^2 + w_{15} \alpha_1 + w_{16} \alpha_2^3 + w_{17} \alpha_2^2 + w_{18} \alpha_2 \]  
\[ (28) \]
\[ (29) \]
\[ (30) \]

Optimizing for these weights, a new frame \( \beta \) is defined for shape inputs \( \alpha_1 \) and \( \alpha_2 \). This induces the following changes in the connection vector fields for \( \xi_x, \xi_y, \) and \( \xi_z \).

3) Numerical Optimization of Sinusoidal Steering Policy: Using insights from the previous section, we hope to build a control strategy which quickly resolves our roll error, while minimizing perturbations in the vehicle’s pitch and yaw frame. Based on the height function provided in Fig.12, a gait will be planned moving along an ellipse. Clockwise paths will induce a positive net rotation in the \( \beta_x \) frame. An numerical optimization procedure is conducted to find an the length of the semi-major (\( A \)) and semi-minor axis (\( B \)), along with the frequency (\( \omega \)) at which the path should be traveled. This will form the policy for the sinusoidal steering controller.

Using a three dimensional parameter space (\( A, B, \omega \)), a 10 second simulation is run with the steering policy. The objective function consists of the squared net angular displacement of the vehicle’s roll over the duration of the simulation.

\[ \text{criterion}(A, B, \omega) = -\theta(t_f)^2 \]  
\[ (31) \]

Optimization produced the following policy:

\[ A = 1, B = 1, w = 30 \]  
\[ (32) \]
4) Discontinuous Sinusoidal Steering Branch in Geometric Controller: To integrate this steering policy with the geometric controller used in the previous section, a switch is set to toggle between the two strategies. If the roll error is above threshold $\delta_{roll}$ then the sinusoidal steering controller is called to overcome this error. Once within the threshold, geometric control takes over to resolve remaining error in yaw and pitch. The vehicle stabilizes to the ideal orientation before contact with the ground.

Fig. 13: Sinusoidal Steering-Geometric Controller

C. Simulation

We simulate the system with two different initial conditions for roll, pitch, yaw of the car:

$$
\begin{bmatrix}
\theta \\
\phi \\
\psi
\end{bmatrix} = \begin{bmatrix}
-\pi/2 \\
\pi \\
\pi/4
\end{bmatrix}
$$

and we want our underactuated controller to drive the car to

$$
\begin{bmatrix}
\theta_d \\
\phi_d \\
\psi_d
\end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}
$$

or

$$
R_d = I = \begin{bmatrix}
1 & 0 & 0 \\
0 & 1 & 0 \\
0 & 0 & 1
\end{bmatrix}
$$

In these simulations, the red rotor can no longer enact a torque on the system. In both cases the sinusoidal steering resolves the initial error, followed by quick convergence to steady state guided by the geometric controller. The roll error for both cases has opposing sign, so the elliptical path is traveled in the reverse direction for the respective trials. These simulations show promising results, showing that the system has overcome substantial initial error to right itself before landing.

V. Future Work

The plan moving forward is to formally prove stability for the underactuated controller, and test it on a real platform in the Biorobotics Lab at Carnegie Mellon University.

We are also interested in minimizing the velocity each rotor obtains throughout the duration of the simulation. This will make implementation on a real system, which has motor saturation constraints, more feasible.

VI. DISCUSSION

Transformation $\beta$ and its evolution over the duration of the simulation are vitally important to the success of the sinusoidal steering technique. At the beginning of each simulation the wheel velocities always start from rest. Therefor the motor controllers sinusoidal input oscillates the wheels back and forth about a set point, keeping $\beta$ well aligned with the local body frame. If the system obtains higher rotor velocities as a consequence of the geometric control, when the controller attempts to reinstate the sinusoidal steering, there will be no oscillation about an angular setpoint. In this situation, $\beta$ is likely not aligned with the body frame, making the steering produce unpredictable dynamical behavior.

Our suggestion is to implement another level of control which sets wheel position directly, rather than sinusoidal torque inputs.
VII. CONCLUSION

Precise and stable control of the planar and fully actuated 3D vehicular platforms are proven analytically and validated in simulations. From here we develop a technique for underactuated control in SO(3) that has been successful in preliminary simulation trials. Minimum perturbation coordinates provide an ideal region of the shape space in which to execute sinusoidal steering strategies. Coupling this insight to the manifold structure with numerical optimization using the rigid body dynamics, we develop a sinusoidal steering policy that effectively resolves initial errors in the roll of the body frame. With this error resolved, the remaining pitch and yaw error is effectively resolved with the pitch controller. This provides a piecewise, discontinuous controller which combines powerful insights from differential locomotive strategies and standard geometric control.

REFERENCES

[1] K. Sreenath and V. Kumar, “Dynamics, control and planning for cooperative manipulation of payloads suspended by cables from multiple quadrotor robots,” in Robotics: Science and Systems (RSS), 2013, this paper won the RSS Best Paper Award 2013.