2016 IEEE 55th Conference on Decision and Control (CDC)
ARIA Resort & Casino
December 12-14, 2016, Las Vegas, USA

# A Stabilizing Gyroscopic Obstacle Avoidance Controller for Underactuated Systems

Gowtham Garimella[1], Matthew Sheckells[2] and Marin Kobilarov[1]

*Abstract*— **This work addresses the problem of controlling an underactuated system to a goal state in a cluttered environment. To this end, a gyroscopic obstacle avoidance controller is proposed for a class of underactuated systems, which take as inputs body torques and a thrust force along some known body-fixed axis. The Lyapunov stability of the controller in the presence of obstacles is shown in theory and verified through simulations. Care is taken to ensure stability even when the system has only a finite obstacle detection radius, and a novel gyroscopic obstacle avoidance gain selection is designed which smoothly attenuates the steering force as the robot heads away from an obstacle. Furthermore, a gyroscopic steering force tailored to two primitive obstacles, namely cylinders and spheres, is proposed. Simulations of a quadrotor system and a nanosatellite system validate that the controller is stable under a large number of obstacles and smoothly converges to the goal state.**

## I. Introduction

Navigation of underactuated dynamic robotic systems in a cluttered environment is considered a nontrivial task. Challenges arise due to the inability of the system to instantaneously produce a force in any desired direction. The class of underactuated systems considered in this work are rigid bodies with controls given by body torques and a thrust force along with some body-fixed axis. These types of systems are common in robotics and include quadrotors, satellites, underwater and surface vehicles. For instance, quadrotors are becoming important for a variety of applications such as search and rescue [1], mapping [2], and package delivery [3]. Quadrotors have four axis aligned rotors which provide thrust along the rotor's axis and torque along the body axes. Besides, small satellites such as cubesats [4] are enabling low-cost testbeds for applications such as formation flying [5] and other autonomous operations [6]. The CubeSats considered in this work have an attitude control system (ACS) and a single thruster which produces the thrust force along some body-fixed direction.

Many control techniques have been developed for such types of vehicles. These include deterministic feedback linearizing controllers for quadrotors in [7], [8], [9], [10] as well as Lyapunov-stable controllers in the presence of bounded external disturbances [11], [12], [13], [14].

In addition to standard point stabilization, requiring provably stable obstacle avoidance significantly complicates the control design. While a backstepping obstacle avoidance controller has been proposed in [15], it focused on two schemes, namely a mass point model and a safety ball model to generate waypoints away from an obstacle. The proposed controller, therefore, does not explicitly include obstacle constraints, and no proof of obstacle-aware convergence is available. Traditionally, obstacle avoidance for fully actuated systems has been considered through gyroscopic avoidance and navigation field [16] approaches. The navigation field approach generates an artificial potential field with a global minimum at the goal and a maximum at the obstacles, but the design of such a potential field is non-trivial. The navigation field approach has been used for obstacle avoidance of quadrotors in [17], [18]. Unlike the potential field methods, the dynamic window approach [19] is a local method that merges a Model Predictive Control (MPC) approach and potential field methods to find a control trajectory in the accessible space that maximizes a utility function.

In contrast to requiring a potentially complex nonlinear potential function, the gyroscopic avoidance approach handles obstacles by applying a steering force to the robot without increasing the Lyapunov energy of the system [20]. The applied steering force, therefore, does not affect the Lyapunov stability of the controller. This method is semi-globally convergent to the goal state in the presence of unknown convex obstacles. Gyroscopic avoidance has been successfully applied to create flocking behavior in a multi-agent system [21] and to control an Unmanned Ground Vehicle (UGV) [22]. A gyroscopic force added to a potential field approach was applied to quadrotor swarm formation in [23], but only a simplified kinematic model for the quadrotor was considered.

This work extends the gyroscopic avoidance approach [20] (originally developed for fully actuated systems) to underactuated dynamical systems in 3D workspaces through a backstepping technique. To ensure convergence in the presence of multiple obstacles, a novel obstacle avoiding steering function has been designed to enable smooth transitions between colliding and non-colliding directions of motions. Furthermore, to ensure stability even when the system has a finite obstacle detection radius, a smooth obstacle control gain is employed. Two types of 3D obstacles are considered: cylinders and spheres. Many real world obstacles can be modeled using a combination of these primitives.

We demonstrate the ability of the method to perform obstacle avoidance in two challenging simulated scenarios. The controller is first employed on a quadrotor and shown to converge to a goal position while avoiding a forest of

[1]Gowtham Garimella and Marin Kobilarov are with the Department of Mechanical Engineering Johns Hopkins University, 3400 N Charles Str, Baltimore, MD 21218, USA ggarime1|marin@jhu.edu

[2]Matthew Sheckells is with the Department of Computer Science Johns Hopkins University, 3400 N Charles Str, Baltimore, MD 21218, USA mshescke1@jhu.edu

trees modeled as cylinders with spherical canopies. A similar example involving a nanosatellite has shown convergence to a goal position while avoiding space debris modeled as spheres. To the best of the authors' knowledge, this is the first controller providing convergence for these types of underactuated systems in complex scenarios.

The rest of the paper is organized as follows. In §II we specify the dynamics of the class of underactuated systems considered. In §III, a desired gyroscopic controller is designed for the translational dynamics. This controller is extended to the class of underactuated systems through backstepping in §IV. Next, the design of gyroscopic obstacle avoidance gains specific to cylindrical, and spherical obstacles are specified in §V. Finally, simulations of a quadrotor and nanosatellite in non-trivial scenarios are shown in section VI. The proof for stable collision avoidance is derived in Appendix.

## II. Dynamics of Underactuated systems

We consider underactuated systems modeled as rigid bodies with position $p \in \mathbb{R}^3$ and velocity $\dot{p}$ in a fixed inertial frame, orientation matrix $R \in SO(3)$ and body-fixed angular velocity $\omega \in \mathbb{R}^3$. The control inputs for the system are the body torques $\tau \in \mathbb{R}^3$ and thrust force $u_t \in \mathbb{R}$ in some known body-fixed direction $e \in \mathbb{R}^3$. The system is subject to known external forces given by $f \in \mathbb{R}^3$ and no external torques. The dynamics is

$$m\ddot{p} = Reu_t + f, \tag{1}$$

$$\dot{R} = R\widehat{\omega}, \tag{2}$$

$$J\dot{\omega} = J\omega \times \omega + \tau, \tag{3}$$

where $m$ is the mass and $J$ is the rotational inertia.

Our goal is to design a Lyapunov-stable controller achieving a given desired goal position $p_d$ with zero velocity $\dot{p}_d = 0$ while avoiding obstacles. To accomplish this, we first design a gyroscopic obstacle avoidance controller for the translational dynamics of the underactuated system. The resulting "desired" control forces for this controller cannot be directly achieved due to under actuation. Therefore, we perform a backstepping procedure which closes the loop in stages and ultimately achieves stability. We next describe the translational and gyroscopic parts of the controller derivation.

## III. Gyroscopic Avoidance

We first design a gyroscopic avoidance controller for the position coordinates. Let $g \in \mathbb{R}^3$ denote the translational input force. For clarity, let the system's translational state combining position and velocity be denoted by $x = [p^T, \dot{p}^T]^T$. The translational dynamics is then given by

$$\dot{x} = Ax + B(g + f), \tag{4}$$

$$A = \begin{bmatrix} 0 & I_{3\times3} \\ 0 & 0 \end{bmatrix}, \quad B = \begin{bmatrix} 0 \\ \frac{1}{m} I_{3\times3} \end{bmatrix}. \tag{5}$$

The underactuated dynamics considered in (1-3) is an extension of this subsystem, where the control force is given as the thrust vector $g = Reu$ and the thrust direction $Re$ is controlled by the rotational dynamics of the system.

The design of the controller starts with defining the error $z_0 \in \mathbb{R}^6$ between the state $x$ and the desired state $x_d = [p_d^T, 0^T]^T$, given by

$$z_0 = x - x_d.$$

For a standard linear system considered in (4-5), a Lyapunov stable feedback control law can be achieved using a desired force $g_d \in \mathbb{R}^3$ given by

$$g_d = -Kz_o - f, \tag{6}$$

$$K = [K_p,\ K_v], \tag{7}$$

where $K_p, K_v \in \mathbb{R}^{3\times3}$ are positive definite matrices corresponding to proportional and derivative gains, respectively.

Gyroscopic avoidance is equivalent to adding force terms which steer the system around the obstacles. This obstacle avoidance is achieved by adding the desired force perpendicular to both the current velocity and the steering axis of the obstacle. The control law is thus augmented according to

$$g_d = -Kz_0 - f - G(x)\dot{p}, \tag{8}$$

where the matrix $G(x) \in \mathbb{R}^{3\times3}$ is skew-symmetric. This matrix instantaneously rotates the velocity away from an obstacle. Equivalently, the *steering force* $G(x)\dot{p}$ can be regarded as being perpendicular to the velocity of the robot.

For stability analysis it will be useful to define the storage function

$$V_0 = \frac{1}{2} z_0^T P z_0, \tag{9}$$

where

$$P = \begin{bmatrix} K_p & 0 \\ 0 & mI_{3\times3} \end{bmatrix}. \tag{10}$$

We then have

$$\dot{V}_0 = \frac{1}{2} z_0^T P(Ax + B(g+f)) + \frac{1}{2}(Ax + B(g+f))^T P z_0.$$

If the system if fully actuated, the input force $g$ can be set to the desired control force $g_d$ and it would be possible to show asymptotic stability. Yet, the input force for an underactuated system is restricted only along the single body direction $Re$ and cannot be set to the desired force $g_d$. Hence, the above controller is extended to the full underactuated system dynamics using a backstepping procedure as described next.

## IV. Backstepping Procedure

We next close the loop in stages using backstepping. The controller designed in (6) requires the control force on the system $g$ to be equal to a desired control force $g_d$ that stabilizes the system. The desired force $g_d$ cannot be achieved directly for underactuated systems since the control force can only be applied along some known body direction $Re$. Instead, the difference between the applied force $g$ and desired force $g_d$ is considered as an error $z_1 \in \mathbb{R}^3$ defined by

$$z_1 = g - g_d,$$

to be further suppressed in the backstepping procedure. With this definition, we have

$$\dot{V}_0 = z_0 Q z_0 + (B^T P z_0)^T (g - g_d),$$

where

$$Q = \begin{bmatrix} 0 & 0 \\ 0 & -K_v \end{bmatrix}.$$

Next, a new Lyapunov candidate is defined which includes error $z_1$ as

$$V_1 = V_0 + \frac{1}{2} z_1^T z_1,$$

with time-derivative given by

$$\dot{V}_1 = z_0 Q z_0 + z_1^T (\dot{g} - \dot{g}_d + B^T P z_0),$$

which can be expressed as

$$\dot{V}_1 = z_0 Q z_0 - k_{z_1} z_1^T z_1 + z_1^T z_2,$$

where $z_2 = \dot{g} - a_d$ and $a_d = \dot{g}_d - B^T P z_0 - k_{z_1} z_1$. The variable $a_d$ is a desired value for $\dot{g}$ which cannot be achieved by the underactuated systems. Thus, continuing the backstepping procedure a new Lyapunov candidate which includes the error between the desired and actual values of $\dot{g}$ is given as

$$V_2 = V_1 + \frac{1}{2} z_2^T z_2,$$

with derivative

$$\dot{V}_2 = z_0 Q z_0 - k_{z_1} z_1^T z_1 + z_2^T (\ddot{g} - \dot{a}_d + z_1).$$

The desired value of $\ddot{g}$ which ensures $\dot{V}_2$ is negative semi-definite is denoted as $b_d$, and is computed as

$$b_d = \dot{a}_d - z_1 - k_{z_2} z_2,$$
$$b_d = \ddot{g}_d - B^T P \dot{z}_0 - k_{z_1} \dot{z}_1 - z_1 - k_{z_2} z_2,$$

where

$$\ddot{g}_d = -[0 \;\; \ddot{G}(x)] z_0 - 2[0 \;\; \dot{G}(x)] \dot{z}_0 - K \ddot{z}_0,$$

and

$$\ddot{z}_0 = A^2 x + AB(f + g) + B\dot{g}.$$

For the underactuated systems, the derivative $\ddot{g}$ can be expanded as

$$\ddot{g} = R[e\ddot{u} + \hat{\dot{\omega}} e u + 2\hat{\omega} e \dot{u} + \hat{\omega}^2 e u].$$

Finally, the control inputs $\tau, \ddot{u}$ can be chosen to satify the desired $b_d$ by setting

$$\tau = \mathbb{J}[e \times (R^T b_d - \hat{\omega}^2 e u - 2\hat{\omega}\dot{u})/u] - \mathbb{J}\omega \times \omega, \tag{11}$$
$$\ddot{u} = e^T (R^T b_d - \omega^2 e u - 2\hat{\omega}\dot{u}). \tag{12}$$

Note that this controller does not directly control the thrust force $u$, but instead controls $\ddot{u}$. The state of quadrotor system is extended by $u, \dot{u}$ to account for this. The controller for torque $\tau$ shown in (11) has a singularity when the thrust force goes to zero $u = 0$. This is not a problem in practice since the vehicle always produces positive force $u$ while navigating.

*Proof of Stability:*

Let the error state for the system be given by $z = [z_0^T, z_1^T, z_2^T]^T$. Using the control law described in (11), the derivative of Lyapunov Function $V_2$ is evaluated as

$$V_2 = \frac{1}{2}(z_0^T P z_0 + z_1^T z_1 + z_2^T z_2), \tag{13}$$
$$\dot{V}_2 = z_0^T Q z_0 - k_{z_1} z_1^T z_1 - k_{z_2} z_2^T z_2 = z^T K z, \tag{14}$$
$$\text{with } K = \begin{bmatrix} Q & 0 & 0 \\ 0 & k_{z_1} & 0 \\ 0 & 0 & k_{z_2} \end{bmatrix}, \tag{15}$$

where the matrix $K$ is negative semidefinite. The largest invariant set [24] with respect to the quadrotor dynamics, when $\dot{V}_2 = 0$ is given by $z = 0$. Assuming $V_2$ is a $C^1$ smooth function and assuming the initial error state $z(t = 0)$, controls $\tau(t), u(t)$, and states $x(t)$ are bounded, the system will asymptotically converge to $z = 0$ according to Lasalle's invariance principle [24].The boundedness of the states and controls is ensured only if the robot does not collide with an obstacle. The proof for collision avoidance is provided in section VII-A which guarantees boundedness of the control inputs. The smoothness of $\dot{V}_2$ is ensured by a smooth control law, which implies the steering forces must be $C^2$ smooth. The design of proper steering forces is discussed in the following section.

## V. Obstacle Avoidance Coefficients

In the backstepping controller shown in (11), the specific form of gyroscopic avoidance matrix $G(x)$ is not specified. In this section, we discuss the form of $G(x)$ suitable for obstacle avoidance.

The obstacle avoidance matrix $G(x)$ is decomposed into a sum of obstacle avoidance matrices $G_i(x)$ corresponding to individual obstacles. Each individual obstacle avoidance matrix is composed of angular obstacle avoidance gain $k_1(\theta_i)$, radial obstacle avoidance gain $k_2(d_i)$, and the steering axis $e_i(x)$. In summary, we have

$$G(x) = \sum G_i(x), \;\; G_i(x) = k_1(\theta_i) k_2(d_i) \hat{e}_i(x). \tag{16}$$

where the hat operator $\hat{}$ maps the steering axis to a skew symmetric matrix as

$$\hat{e} = \begin{bmatrix} 0 & -e_3 & e_2 \\ e_3 & 0 & -e_1 \\ -e_2 & e_1 & 0 \end{bmatrix}.$$

The angular obstacle avoidance gain reduces the magnitude of steering force as the robot heads away from the obstacle, while the radial obstacle avoidance gain forces the magnitude of steering force to be inversely proportional to the distance between the robot and obstacle. In addition, the steering force should be negligible beyond a user specified distance from the obstacle known as the detection radius, since the obstacle avoidance is not active beyond this distance. The second order derivatives of $G$ are used in the backstepping control law (11). Therefore, the obstacle avoidance gains and steering axis direction should be $C^2$ smooth functions.

The obstacles considered in this work are either cylinders or spheres. The choice of obstacle avoidance gain and steering axis for the two obstacles are discussed next.

### A. Cylinders

A cylinder is specified by the major axis $a$, radius $r$, and a point on the major axis $o_p$. The cylinder obstacles are assumed to have infinite length. The steering axis $e(x)$ for a cylindrical obstacle is chosen along the major axis $a$ which applies steering correction around the major axis of the cylinder. There is an ambiguity on the sign of the principal axis of the cylinder, which corresponds to driving to the right or left of the cylinder. The sign of the steering axis is determined according to (17-20). According to this approach, the robot avoids the cylinder by steering right if it is heading to the right of the cylinder, and the opposite when heading left of the cylinder. The steering axis $e(x)$ is mathematically computed as

$$v = \dot{p} - (a^T \dot{p})a, \tag{17}$$

$$\Delta p = o_p - p, \tag{18}$$

$$d = \Delta p - (a^T \Delta p)a, \tag{19}$$

$$e(x) = \text{sign}(a^T (d \times v))a. \tag{20}$$

The velocity $v$ is the projected robot velocity onto the plane perpendicular to the major axis of the cylinder $a^{\perp}$. Similarly, the displacement vector $d$ is the projection of the vector connecting the robot center to the point on the major axis of the cylinder onto the plane $a^{\perp}$.

The sign function is not differentiable, hence it is not suitable for the backstepping control law. A scaled sigmoid function which is a smooth approximation of the sign function is chosen instead. The smooth steering axis is given by:

$$e(x) = (2S(a^T (d \times v)) - 1)a,$$

$$S(t) = \frac{1}{1 + e^{-kt}}.$$

The angular obstacle avoidance gain is chosen to decay exponentially with the angle between the robot heading and obstacle $\theta$. The mathematical form of the angular gain is described as

$$k_1(\theta) = \exp\left[k_{att}\,(\theta - 1)\right], \tag{21}$$

$$\theta = \frac{d^T v}{\|d\|\|v + \lambda a\|}. \tag{22}$$

The variable $\theta$ is the cosine of the angle between the projected velocity $v$ and the projected displacement vector $d$, and $k_{att} \in \mathbb{R}^+$ adjusts the sensitivity of the steering force to the robot's heading. When the magnitude of the projected velocity approaches zero, the angle between $d$ and $v$ is undefined. A velocity vector in the null space of projected velocity is added to the projected velocity in (22) to avoid this singularity. This regularizing term also ensures that the derivatives of $k_1(\theta)$ remain bounded.

The specific form of the radial obstacle avoidance gain is shown below:

$$k_2(d) = k_{obs} \frac{S(r_d + r - \|d\| - \varepsilon)}{\|d\| - r}. \tag{23}$$

A sigmoid function is used as an smooth approximation of the step function to suppress the radial gain beyond the finite detection radius $r_d$. The value of $\varepsilon$ is adjusted such that the radial obstacle avoidance gain at the detection radius is negligible. The gain $k_{obs}$ scales the steering effort. An appropriate value for $k_{obs}$ can guarantee obstacle avoidance as discussed in appendix V.

The complete obstacle avoidance matrix for a cylindrical obstacle is written as

$$G(x) = k_{obs} \frac{e^{k_{att}(\theta - 1)}}{\|d\| - r} S(r_d + r - \|d\| - \varepsilon) S(a^T (v \times d))\hat{a}.$$

### B. Spheres

A sphere is specified by it's center $o_p$ and radius $r$. Unlike a cylinder, the steering axis for a sphere is not constant, and is chosen based on the robot velocity and displacement vector from the robot to the center of the sphere. Given the velocity of the robot $v = \dot{p}$, the distance to the sphere $d = o_p - p$, the form of steering axis chosen as

$$e(x) = \frac{d \times v}{\|d \times v\|}.$$

When the robot is heading towards the obstacle, the cross product of $d \times v$ goes to zero, which creates unbounded derivatives of the steering axis. To avoid this issue, a regularized value of norm of the cross product is used. The norm of the cross product can be written as

$$\|d \times v\| = \sqrt{\|d\|^2 \|v\|^2 - (d^T v)^2}.$$

The regularized value can then be formulated as

$$\|d \times v\|_{reg} = \sqrt{\|d\|^2 \|v\|^2 (1 + k_{reg}) - (d^T v)^2}.$$

The gain $k_{reg}$ ensures that the norm $\|d \times v\|_{reg}$ is non-zero even when the robot is heading towards the obstacle.

The radial obstacle avoidance gain $k_2(d)$ is the same as that of cylinder obstacle. Similar to the cylinder case, the steering gain chosen for a spherical obstacle as $k_1(\theta) = \exp\left[k_{att}(\theta - 1)\right]$, where $\theta$ the angle between the robot heading and obstacle is given by

$$\theta = \frac{d^T v}{\|d\|\|v\|}.$$

The steering obstacle avoidance gain and the steering axis are not defined when the robot velocity is zero. This singularity is avoided by setting the obstacle avoidance to zero when the velocity of the robot is below a threshold $\delta$. The complete obstacle avoidance matrix $G$ for a sphere is written as

$$G(x) = \begin{cases} k_{obs} \frac{\exp[k_{att}(\theta - 1)]}{\|d\| - r} \frac{S(r_d + r - \|d\| - \varepsilon)}{\|d \times v\|_{reg}} \widehat{d \times v} & \text{if } \|v\| > \delta, \\ 0 & \text{otherwise.} \end{cases}$$

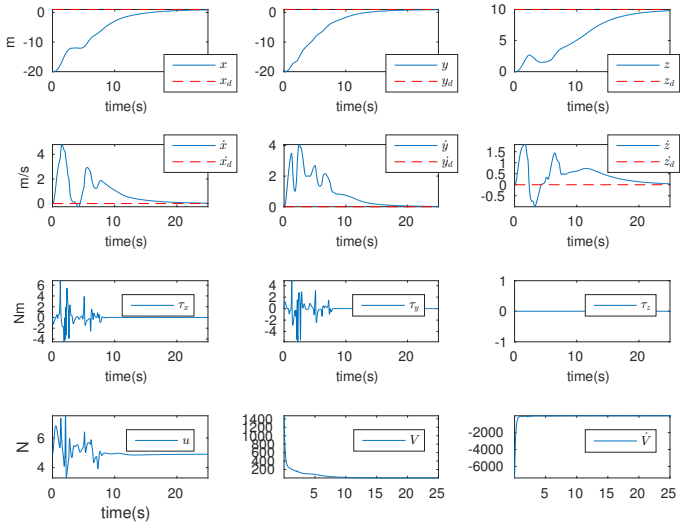


Fig. 2. History of the quadrotor position, velocity, controls, and Lyapunov function $V$ for the quadrotor simulation

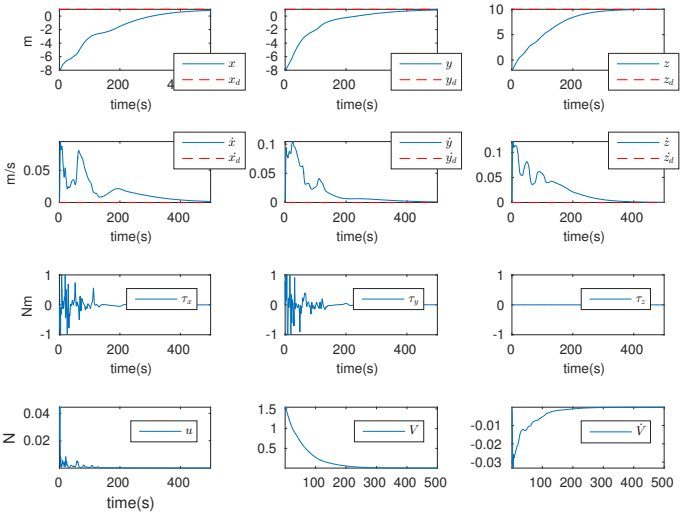


Fig. 3. History of the nanosatellite position, velocity, controls, and Lyapunov function $V$ for satellite simulation

## VI. Numerical Simulations

The backstepping controller designed in section IV is tested on a quadrotor and a nanosatellite in simulation. The underactuated dynamics in (1) has been discretized using a semi-implicit scheme [25] and integrated using Euler integration.

A finite detection radius of 3 meters is applied to the obstacles. The backstepping parameters have been chosen to ensure the controls are bounded, and the system converges smoothly to the goal.

### A. Quadrotor in a Dense Forest

A standard quadrotor model is used, with the thrust aligned with the body-fixed $z$-axis of the system and with gravity as the only external force, i.e., $f = (0,0,-9.81m)$. The mass of the quadrotor is 0.5 Kg and the moment of inertia is chosen as a diagonal matrix $J = diag([.003,.003,.005])$. The obstacle scene used for testing is that of a forest with tree obstacles as shown in Fig. 1. Each tree obstacle consists of a cylindrical trunk and a spherical canopy. A total of 23 trees are generated on a grid spanning a 20m$\times$20m region. The trees are perturbed randomly from the grid centers. The goal of the controller is to reach the desired position shown in Fig. 1 starting from the opposite side of the grid.

One can observe that the quadrotor smoothly reaches the goal while avoiding obstacles as illustrated in Fig. 1. The state, control, and Lyapunov energy history of the quadrotor trajectory are shown in Fig. 2. Note that the Lyapunov energy function asymptotically approaches zero with no discontinuities, even though the obstacle detection radius of the system is finite.

### B. Satellite among Space Debris

For the second example, consider a nanosatellite equipped with an attitude control system and a single thruster. The satellite is placed in an environment of space debris modeled as spheres, and no external forces are included in the simulation. The satellite is tasked to navigate to the desired position shown in 1 while avoiding the obstacles. A total of 48 obstacles are generated using a cuboid grid in a cubic space of 9m$\times$9m$\times$9m. The obstacles are randomly perturbed around the grid centers.

Similarly to the quadrotor scenario, the satellite is able to stabilize to the goal while avoiding obstacles as demonstrated in Fig. 1. A detailed state, control, and Lyapunov energy history of the satellite are shown in Fig. 3. The Lyapunov energy function asymptotically approaches zero even as new obstacles enter or leave the detection radius of the system.

## VII. Conclusions

A backstepping Lyapunov stable controller using gyroscopic obstacle avoidance has been designed for underactuated systems. The ability of the controller to handle finite obstacle detection radius and the underactuated dynamics in the presence of obstacles has been shown in theory, and simulations verified the capability of the controller to avoid obstacles and converge to a goal in complex scenarios. Furthermore, the obstacle coefficient design approach employed in this work can be extended to include new primitives such as finite-length cylinders, finite planes, and ellipsoids. The backstepping procedure can also be extended to stabilize the system under bounded external disturbances as explained in [11]. Future work will concentrate on implementing the gyroscopic obstacle avoidance controller on a real system and showing that the convergence guarantees hold.

## Appendix

### A. Proof of Obstacle Avoidance

In this section, the appropriate choice of scaling gain on the distance obstacle avoidance $k_{obs}$ shown in (23) required to guarantee obstacle avoidance of the system is presented. Several assumptions are required for finding the gain. To start with it is assumed that there is a single obstacle in the

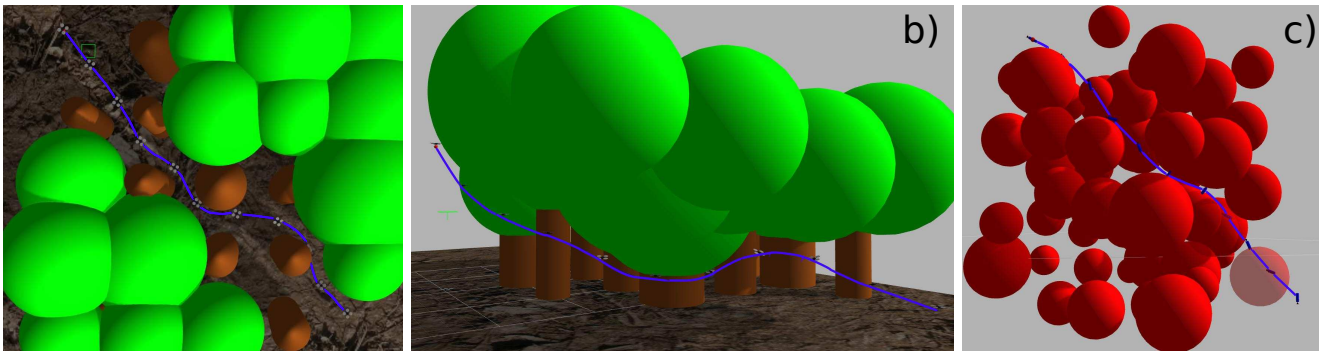


Fig. 1. The figures (a) and (b) show a quadrotor navigating through a dense cluster of trees using the proposed controller with a finite obstacle detection radius. Figure (a) shows the top view of the quadrotor, where the obstructing obstacles have been removed for a clear view of the trajectory. Figure (b) shows a side view of the same trajectory swooping under the canopies to reach the goal. A satellite navigating through space debris using the proposed controller is illustrated in (c). The thrust vector of the satellite is shown by a pointed cone at the bottom of the satellite.

environment. Further, the gain matrices $K_p$ and $K_v$ used in the desired force $g_d$ in (8) are assumed to be constant matrices as in $k_p I_{3\times3}, k_v I_{3\times3}$, and we set $m=1$ for simplicity. If the robot is not moving $\dot{p}=0$, it is assumed that the robot will not collide with an obstacle. The proof for obstacle avoidance is explained through the principle of contradiction similar to the proof shown in [20].

Let the system collide with the obstacle at $t_c$ with non-zero velocity $\dot{p}(t_c)\neq 0$. The dynamics before the collision during the interval $I=[t_c-\Delta t, t_c^-]$ is considered. The closed loop translational dynamics of the underactuated system can be written as

$$\ddot{p} = g+f = -k_p(p-p_d)-(k_v I_{3\times3}+G)\dot{p}+z_1. \tag{24}$$

Integrating the dynamics for the interval $I$ gives $\dot{p}(t_c^-)$ as

$$\dot{p}(t_c^-) = e^{-k_v\Delta t}R_G(t_c^-, t_c^- -\Delta t)\dot{p}(t_c-\Delta t)+e(\Delta t), \tag{25}$$

$$e(\Delta t) = \int_{t_c^- -\Delta t}^{t_c^-} e^{-k_v(t_c^- -\tau)}R_G(t_c^-,\tau)(-k_p\Delta p+z_1)d\tau, \tag{26}$$

where $R_G(t,\tau)$ is given by solving $\frac{d}{dt}R_G(t,\tau) = -G(x)R_G(t,\tau)$ with $R_G(\tau,\tau)=I$ [26]. The obstacle avoidance matrix $G(x)$ has the form shown in (16). The steering axis $e_i(x)$ is assumed to be constant during the small time $\Delta t$. For cylindrical obstacle, the steering axis is chosen to be the major axis of cylinder, hence it is constant during $\Delta t$. For a sphere, this is a valid approximation assuming the displacement vector does not change during the interval $I$. Under this assumption, the rotation matrix $R_G(t,\tau)$ can be simplified as

$$\begin{aligned}
&\frac{d}{dt}R_G(t,\tau) = -k_1(\theta)k_2(d)\hat{e}R_G(t,\tau),\\
&R_G(t,\tau) = e^{-\left[\int_\tau^t k_1(\theta)k_2(d)dt\right]\hat{e}} = R_e(\psi(t,\tau)),\\
&\psi(t,\tau) = \int_\tau^t k_1(\theta)k_2(d)dt.
\end{aligned}$$

The form of $R_e(\psi(t,\tau))$ implies that the obstacle avoidance matrix induces a rotation about the steering axis $e$, where $e$ is assumed to be constant during $\Delta t$ and the amount of rotation is given by the angle $\psi(t,\tau)$.

Let the initial value of the Lyapunov function (13) be $V_{max} \triangleq V_2(t=0)$. It has been shown in (14), that the Lyapunov energy is non-increasing over time. The error vector $e(\Delta t)$ from (26) is shown to be $O(\Delta t)$ (i.e the norm of the vector is $O(\Delta t)$) as follows

$$\begin{aligned}
\|e(\Delta t)\| &\leq \int_{t_c-\Delta t}^{t_c^-}\|(-k_p\Delta p+z_1)\|d\tau,\\
&\leq \int_{t_c-\Delta t}^{t_c^-} k_p\|\Delta p\|+\|z_1\|d\tau,\\
&\leq \sqrt{V_{max}}(\sqrt{2k_p}+1)\Delta t, \quad \text{using (10,13).}
\end{aligned}$$

Using this result, the velocity at $t_c^-$ from (25) can be written as

$$\dot{p}(t_c^-) = e^{-k_v\Delta t}R_e(\psi(t_c^-, t_c^- -\Delta t))\dot{p}(t_c-\Delta t)+O(\Delta t). \tag{27}$$

Since the Lyapunov energy is non-increasing over time and $\frac{1}{2}m\|\dot{p}\|^2 < V(t) \leq V_{max}$ according to (13,10), the norm of the velocity of the system is upper bounded as $\|\dot{p}(t)\| < \sqrt{2V_{max}}$. This implies the projected velocity $v(t)$ shown in (17 should also have the same property, i.e. $\|v(t)\| \leq \sqrt{2V_{max}}$. The derivative of the projected distance vector $d(t)$ is evaluated to be the negative of the projected velocity $\dot{d}(t) = -v(t)$. The projected distance at the time of collision is equal to the radius of the obstacle $\|d(t_c)\| = r$. The projected distance at $t_c-\Delta t$ can be bounded as

$$\|d(t_c-\Delta t)\| \leq \sqrt{2V_{max}}\Delta t + r. \tag{28}$$

The projected distance is assumed to be continuously getting closer to the obstacle as in

$$\|d(t)\| \leq \|d(t_c-\Delta t),\| \ \ \forall t\in I. \tag{29}$$

Since the robot is heading towards the obstacle, the absolute value of the heading of the robot from the obstacle will be less than $\pi/2$. Hence, the angular avoidance gain during the interval $I$ can be bounded as $k_1(\theta(t)) \geq e^{-k_{att}}$. The robot is also assumed to be within the detection radius. Hence, the obstacle avoidance gain is simplified as $k_2(d(p(t))) = k_{obs}/(\|d(p(t))\| - r)$. A lower bound on the

rotation $\psi(t_c^-, t_c^- - \Delta t)$ is found using above results as

$$\begin{aligned}\psi(t_c^-, t_c^- - \Delta t) &= \int_{t_c-\Delta t}^{t_c^-} k_1(\theta(t))k_2(d(t))dt, \\ &\geq \int_{t_c-\Delta t}^{t_c^-} e^{-k_{att}} \frac{k_{obs}}{\|d(t)\| - r} dt, \\ &\geq e^{-k_{att}} \frac{k_{obs}}{\sqrt{2V_{max}}}, \quad \text{using (28-29).}\end{aligned}$$

If $k_{obs}$ is chosen as $k_{obs} \geq \pi e^{k_{att}} \sqrt{2V_{max}}$, the lower bound on the rotation induced by the obstacle avoidance matrix is given by $\psi(t_c^-, t_c^- - \Delta t) \geq \pi$. This bound is not dependent on time left to collision $\Delta t$. Using (27), the velocity at the time of collision is rotated around the obstacle axis by more than 180 degrees from the time $t_c - \Delta t$ when the robot is expected to be heading towards the obstacle. Thus at time $t_c$ when the robot collision occurs, the robot is not heading towards the obstacle which is a contradiction to our initial assumption that the robot collided with the obstacle at nonzero velocity for which the heading angle needs to be towards the obstacle. Even with a small time to collision, the robot can completely avoid the obstacle with the appropriate gain selection of $k_{obs}$. In practice, this a very conservative gain and smaller values than this have achieved satisfactory collision avoidance.

## REFERENCES


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

[2] S. Siebert and J. Teizer, "Mobile 3d mapping for surveying earthwork projects using an unmanned aerial vehicle (UAV) system," *Automation in Construction*, vol. 41, pp. 1–14, 2014.

[3] "Prime air," http://www.amazon.com/b?node=8037720011, 2015.

[4] S. Rainer, "Status and trends of small satellite missions for earth observation," *Acta Astronautica*, vol. 66, no. 1, pp. 1–12, 2010.

[5] G. C. Caillibot Eric and D. Kekez, "Formation flying demonstration missions enabled by canx nanosatellite technology," in *Proceedings of the AIAA/USU Conference on Small Satellites, 13th Annual Frank J. Redd Student Scholarship Competition*, 2005. [Online]. Available: http://digitalcommons.usu.edu/smallsat/2005/all2005/42/

[6] M. Kobilarov and S. Pellegrino, "Trajectory planning for cubesat short-time-scale proximity operations," *Journal of Guidance, Control, and Dynamics*, vol. 37, no. 2, pp. 566–579, 2014.

[7] E. Altuğ, J. P. Ostrowski, and R. Mahony, "Control of a quadrotor helicopter using visual feedback," in *2002 IEEE International Conference on Robotics and Automation. Proceedings. ICRA'02.*, vol. 1. IEEE, 2002, pp. 72–77 vol.1.

[8] S. A. Al-Hiddabi, "Quadrotor control using feedback linearization with dynamic extension," in *6th International Symposium on Mechatronics and its Applications, 2009. ISMA'09.* IEEE, March 2009, pp. 1–3.

[9] K. H. J. Lee. Daewon and S. Shankar, "Feedback linearization vs. adaptive sliding mode control for a quadrotor helicopter," *International Journal of control, Automation and systems*, vol. 7, no. 3, pp. 419–428, 2009.

[10] H. Voos, "Nonlinear control of a quadrotor micro-UAV using feedback-linearization," in *2009 IEEE International Conference on Mechatronics. ICM 2009.* IEEE, April 2009, pp. 1–6.

[11] M. Kobilarov, "Trajectory tracking of a class of underactuated systems with external disturbances." in *2013 American Control Conference (ACC)*, June 2013, pp. 1044–1049.

[12] R. Mahony and T. Hamel, "Robust trajectory tracking for a scale model autonomous helicopter," *International Journal of Robust and Nonlinear Control*, vol. 14, no. 12, pp. 1035–1059, 2004.

[13] S. Bouabdallah and R. Siegwart, "Backstepping and sliding-mode techniques applied to an indoor micro quadrotor," in *2005 IEEE International Conference on Robotics and Automation, 2005. ICRA 2005.* IEEE, 2005, pp. 2247–2252.

[14] M. G. O. G. V. Raffo and F. R. Rubio, "Backstepping/nonlinear H ∞ control for path tracking of a quadrotor unmanned aerial vehicle," in *2008 American Control Conference*. IEEE, June 2008, pp. 3356–3361.

[15] S. H. Geng Qingbo and Q. Hu, "Obstacle avoidance approaches for quadrotor UAV based on backstepping technique," in *2013 25th Chinese Control and Decision Conference (CCDC)*. IEEE, May 2013, pp. 3613–3617.

[16] E. Rimon and D. E. Koditschek, "Exact robot navigation using artificial potential functions," *IEEE Transactions on Robotics and Automation*, vol. 8, no. 5, pp. 501–518, 1992.

[17] H. K. Chang Kai, Xia Yuanqing and M. Dailiang, "Obstacle avoidance and active disturbance rejection control for a quadrotor," *Neurocomputing*, vol. 190, pp. 60 – 69, 2016. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0925231216000801

[18] A. Budiyanto, A. Cahyadi, T. B. Adji, and O. Wahyunggoro, "UAV obstacle avoidance using potential field under dynamic environment," in *2015 International Conference on Control, Electronics, Renewable Energy and Communications (ICCEREC),*, Aug 2015, pp. 187–192.

[19] P. Ogren and N. E. Leonard, "A convergent dynamic window approach to obstacle avoidance," *IEEE Transactions on Robotics*, vol. 21, no. 2, pp. 188–195, April 2005.

[20] D. E. Chang and J. E. Marsden, "Gyroscopic forces and collision avoidance with convex obstacles," in *New trends in nonlinear dynamics and control and their applications*. Springer, 2003, pp. 145–159.

[21] D. E. Chang, S. C. Shadden, J. E. Marsden, and R. Olfati-Saber, "Collision avoidance for multiple agent systems," in *42nd IEEE Conference on Decision and Control (CDC), 2003.*, vol. 1, December 2003, pp. 539–543 Vol.1.

[22] D. Vissiere, D. E. Chang, and N. Petit, "Experiments of trajectory generation and obstacle avoidance for a ugv," in *2007 American Control Conference (ACC)*, July 2007, pp. 2828–2835.

[23] F. S. Haibo Min and F. Niu, "Decentralized UAV formation tracking flight control using gyroscopic force," in *IEEE International Conference on Computational Intelligence for Measurement Systems and Applications, 2009. CIMSA '09.*, May 2009, pp. 91–96.

[24] H. K. Khalil and J. Grizzle, *Nonlinear systems*. Prentice hall New Jersey, 1996, vol. 3.

[25] M. Kobilarov, "Discrete optimal control on lie groups and applications to robotic vehicles," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*, May 2014, pp. 5523–5529.

[26] C.-T. Chen, *Linear system theory and design*. Oxford University Press, Inc., 1995.