Modeling and Control of Apical Extension Robots Through Hybrid Dynamics

Vedad Bassari

March 23, 2023

1 Introduction

Apical extension robots, colloquially known as vine robots, are a novel class of soft robots [1]. Vine robots are constructed from a thin-walled tube of flexible material inverted into itself, creating an outer body and a length of internal (inverted) material (Figure 1). When the body is pressurized, this internal material is pulled to the tip where it everts and causes the vine robot to grow (lengthen). To retract (shorten) the vine robot, inversion at the tip is achieved by pulling internal material away from the tip towards the base of the robot.

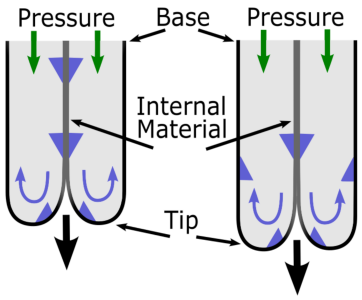

Figure 1: A pressurized vine robot grows as internal material (gray) is pulled to the tip.

These two modes of operation, namely growth and retraction, are the available tools for controlling the robot's position. A common scheme for positional control of vine robots is to grow roughly to the desired point in space before reducing pressure to the point that allows tail (internal material) tension to prevent further growth of the robot. However, this strategy is ineffective in large-body vine robots because the rate of pressure modulation and the accuracy of pressure sensing are severely limited by fluid losses and fan performance. These factors render precise control of pressure impractical and therefore limit the viability of finding an exact equilibrium position in either growth or retraction regimes. Instead,

approximate control of position can be achieved by repeatedly toggling between growth and retraction about the desired position. This ability is particularly desirable for industrial inspection robots, since the users are often interested in precise examination of a specific point of the asset being monitored. An alternative proposal is to use brakes to prevent movement, but electrical brakes commonly used in similar applications suffer from thermal limitations when engaged for extended periods of time. The objective of this discussion is to formalize the proposed control strategy by treating apical extension robots as a hybrid dynamical system.

2 Methods

Hybrid dynamics is a framework for modeling dynamical systems that have both continuous-time and discrete-time behavior [2]. Systems with explicit logical modes - in this case the growth and retraction of the vine robot - are commonly treated as hybrid systems. The following sections formulate a hybrid model of the vine robot and attempt to make assertions about the stability of the desired position under the proposed controller. Additionally, simulation studies are presented to examine the robustness of the controller. We focus our attention on the special case of controlling a robot's linear position in a straight path, but the presented model can be extended to any path and robot configuration without loss of generality.

2.1 Hybrid System Formulation

Hybrid systems are represented as

$$\mathcal{H} = (C, F, D, G) \tag{1}$$

where C is the flow set, F is the flow map, D is the jump set, and G is the jump map [2]. Because hybrid systems display both continuous and discrete time behavior, it is necessary to split the state space into a flow-set C in which the state "flows" continuously according to a flow-map F and a jump-set D in which the state "jumps" discretely according to a jump-map G. The data-set (C, F, D, G) encapsulates the dynamics of the system.

To establish a hybrid model of the vine robot, let us first define the state vector:

$$x \coloneqq \begin{bmatrix} z \\ p \\ q \end{bmatrix}$$

where $z \in [0, z_{max}]$ is the linear position of the vine robot limited to one-dimension for simplicity, with z_{max} being the fully extended length of the robot. $p \in [0, p_{max}]$ is the internal

pressure of the robot where p_{max} is dictated by the fan capacity. Lastly, $q \in \{1, 2\}$ is the logic variable corresponding to the operating mode such that mode 1 indicates growth and mode 2 indicates retraction.

Next, we need to divide the state-space between the flow-set and the jump-set. We first Consider the desired attractor for the system: let us define this based on a reference position z_{goal},

$$\mathcal{A} \coloneqq [z_{goal} - \delta, z_{goal} + \delta] \times [0, p_{max}] \times \{1, 2\}. \tag{2}$$

Note that the attractor, as currently defined, allows for any pressure or mode as long as the position is within the desired band. This is due to the fact that a specific equilibrium pressure is not required for the toggling-controller. In fact, vine robots frequently have vastly different growth and retraction pressures; this will be evident when the growth and retraction dynamics are introduced. Moreover, δ is arbitrarily defined as the positional tolerance that is desired for a given task.

With the attractor defined, it is possible to outline the control scheme introduced above in more detail: If the robot is at a position smaller than the goal ($z < z_{goal} + \delta$), the growth mode is engaged. Growth continues until the robot grows past $z = z_{goal} + \delta$, at which point the mode is changed to retraction in order to stay within the attractor. This operating principle can be mathematically represented as the flow and jump sets for mode 1:

$$\begin{aligned} C_1 &\coloneqq [0, z_{goal} + \delta] \\ D_1 &\coloneqq [z_{goal} + \delta, z_{max}]. \end{aligned} \tag{3}$$

In a similar manner, the flow and jump sets of the second mode are defined in a way that allows for retraction if the position is greater than the goal ($z > z_{goal} - \delta$) but toggles a switch to growth otherwise:

$$\begin{aligned} C_2 &\coloneqq [z_{goal} - \delta, z_{max}] \\ D_2 &\coloneqq [0, z_{goal} - \delta]. \end{aligned} \tag{4}$$

Finally, the full jump and flow sets can be constructed from the union of the sets for each mode. Figure 2 provides a visual overview of the flow and jump sets, with all sets being

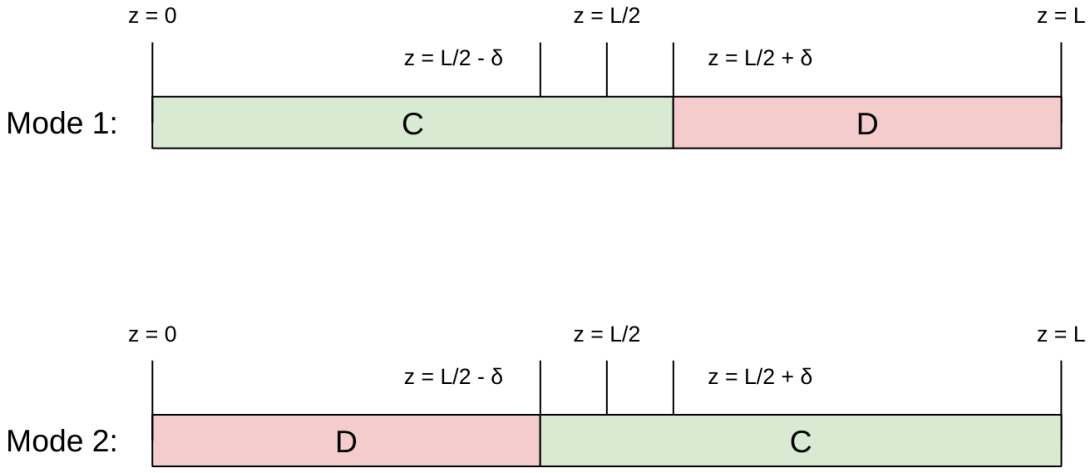

Figure 2: Visual representation of the jump and flow sets in both modes.

closed.

$$\begin{aligned} C &:= ([0, z_{goal} + \delta] \times \{1\}) \cup ([z_{goal} - \delta, z_{max}] \times \{2\}) \\ D &:= ([z_{goal} + \delta, z_{max}] \times \{1\}) \cup ([0, z_{goal} - \delta] \times \{2\}) \end{aligned} \tag{5}$$

The flow maps of the constructed system are based on existing analytical models for the growth and retraction of a vine robot [3] [4].

$$\begin{aligned} v_{growth} &= \left[\frac{1}{2}p - \frac{\mu w z}{A} - Y - \sum_i \frac{C}{A} e^{\frac{\mu L_i}{R_i}}\right]^n \phi \\ v_{retraction} &= -\left[-\frac{1}{2}p - \frac{\mu w z}{A} - Y - \sum_i \frac{C}{A} e^{\frac{\mu L_i}{R_i}} + \frac{T}{A}\right]^n \phi \end{aligned} \tag{6}$$

In this expression, A is the cross-sectional area of the robot, w is the weight per-length of the robot body, μ is the coefficient of friction, Y is the force associated with robot deformation, T is the tail tension, and L_i, R_i correspond to the radii and lengths of the turns in

the robot's path. Lastly, n and ϕ are empirical velocity-dependent values, and n has been shown to take a value between 1 and 1.5. It is evident that these expressions result from simple force balances arising from pressure and various sources of friction.

For the case study at hand, the expressions in (7) can be simplified to obtain the flow maps in both modes.

$$F_1 \coloneqq \begin{bmatrix} (max\{0, \frac{1}{2}p - kz - c\})^n \\ k_p|z_{goal} - z| \\ 0 \end{bmatrix} \tag{7}$$

$$F_2 \coloneqq \begin{bmatrix} -(max\{0, \frac{1}{2}p - kz - c + T\})^n \\ -k_p|z_{goal} - z| \\ 0 \end{bmatrix} \tag{8}$$

Note that the flow of z is set up in such a way to account for the adversarial nature of the friction forces (i.e friction forces in growth cannot be larger than the driving force). The pressure is set to flow proportional to the error in position, and the mode does not change during periods of flow.

The jump maps are more straight-forward to construct, since the jump dynamics do not alter pressure or position and only serve to toggle the mode. We can summarize the system data in the following format.

$$\begin{aligned} C &\coloneqq ([0, z_{goal} + \delta] \times \{1\}) \cup ([z_{goal} - \delta, z_{max}] \times \{2\}) \\ F &\coloneqq F_q \\ D &\coloneqq ([z_{goal} + \delta, z_{max}] \times \{1\}) \cup ([0, z_{goal} - \delta] \times \{2\}) \\ G &\coloneqq \begin{bmatrix} z \\ p \\ 3 - q \end{bmatrix} \end{aligned} \tag{9}$$

2.2 Stability Analysis

To examine the stability of the attractor under the control action, candidate Lyapunov functions for each mode are examined.

1. We start by considering the growth mode, with the attractor in growth defined as

$$\mathcal{A}_1 = \{z_{goal} + \delta\} \times \{2(k(z_{goal} + \delta) + c)\} \tag{10}$$

Note that the z value of the attractor is the endpoint of the flow map for the growth mode and the pressure value of the attractor is defined to set the z-derivative equal to zero at this position.

- Start with a candidate $V = (p_1 - p) + (z_{goal} + \delta - z)$, where $p_1 = 2(k(z_{goal} + \delta) + c)$.
- The function is selected to yield a derivative proportional to the distance from the attractor set:

$$\begin{aligned} < \nabla V, F_1 > &= -k_p|z_{goal} - z| - (\frac{1}{2}p - kz - c)^n \qquad (11) \\ -k_p|z_{goal} - z| - (\frac{1}{2}p - kz - c)^n &\leq -\rho(\sqrt{(p_1 - p)^2 + (z_{goal} + \delta - z)^2}) \end{aligned}$$

  with ρ a continuous, positive definite function. Since $p_1 - p \leq p_{max}$ and $z_{goal} + \delta - z \leq z_{max}$, a function of the form $\rho = rs$ with a real constant $r(p_{max}, z_{max})$ can be found for the system to fulfill (12) while the system is in the flow set of the first mode. This is made possible by the appearance of the same terms (neglecting the small magnitude of δ on both sides of the inequality. Note that the second term of the Lie derivative is positive semi-definite and does not adversely affect the flow behavior.
- Next we consider the boundedness requirement of the Lyapunov function.

$$\begin{aligned} \alpha_1(\sqrt{(p_1 - p)^2 + (z_{goal} + \delta - z)^2}) &\leq (p_1 - p) + (z_{goal} + \delta - z) \qquad (12) \\ &\leq \alpha_2(\sqrt{(p_1 - p)^2 + (z_{goal} + \delta - z)^2}) \end{aligned}$$

  where α_1, α_2 are κ_∞ functions [5]. Again, by construction of the Lyapunov function, a simple function of the form $\alpha = rs$ fulfills both the lower-bound and the upper bound. This is possible because the Lyapunov function is positive-definite and zero only when x is in the attractor, allowing the bounding functions to remain zero-at-zero.
- Lastly, because the jump map does not affect p or z we can show $V(g) = V(x)$ for all x in the jump set. This completes the Lyapunov prove for the global asymptotic stability of $\mathcal{A}_1$ in the growth mode.

2. For the retraction mode, consider the modified attractor set:

$$\mathcal{A}_2 = \{z_{goal} - \delta\} \times \{k(z_{goal} - \delta) + c + \frac{1}{2}p\} \tag{13}$$

- Following the same procedure as mode 1, define a candidate $V = (p - p_2) + (z - z_{goal} + \delta)$, where $p_2 = k(z_{goal} - \delta) + c + \frac{1}{2}p$.
- Consider the Lie derivative of the Lyapunov function.

$$\begin{aligned} < \nabla V, F_2 >= -k_p|z_{goal} - z| \quad &- \quad (\frac{1}{2}p - kz - c + T)^n \\ -k_p|z_{goal} - z| - (\frac{1}{2}p - kz - c + T)^n \quad &\leq \quad -\rho(\sqrt{(p_2 - p)^2 + (z_{goal} - \delta - z)^2}). \end{aligned} \tag{14}$$

It is evident, following an identical reasoning as above, that the flow is well-behaved according to Lyapunov conditions. This is once again caused by the fact that the term $k_p|z_{goal} - z|$ appears directly on the left-hand side of the inequality by the choice of control law.

- We note that the modification of the signs of the terms allows for the following inequality:

$$\begin{aligned} \alpha_1(\sqrt{(p_2 - p)^2 + (z_{goal} - \delta - z)^2}) \quad &\leq \quad (p - p_2) + (z - z_{goal} + \delta) \\ &\leq \quad \alpha_2(\sqrt{(p_2 - p)^2 + (z_{goal} - \delta - z)^2}). \end{aligned} \tag{15}$$

Once again, the Lyapunov function is constructed to be positive-definite and zero when the state is in the attractor. Because the terms appearing an all sides of the inequalities are identical, it is trivial to show the existence of class-kappa bounding functions.

- Finally, $V(g) = V(x)$ concludes the Lyapunov stability of the attractor $\mathcal{A}_2$ under the retraction regime.

3. We have shown the global asymptotic stability of $\mathcal{A}_1$ in growth and $\mathcal{A}_2$ in retraction. Next, note that $\mathcal{A}_1 \subset D_1$ and $\mathcal{A}_2 \subset D_2$. This implies that any period of flow is followed by a mode switch, with the state oscillating between the two mode-specific attractors.

Since $\mathcal{A}_1$ and $\mathcal{A}_2$ are on the interior of $\mathcal{A}$, the full-system attractor $\mathcal{A}$ is rendered globally asymptotically stable.

4. Because the system data satisfy the hybrid basic conditions (in simplistic terms closed sets and continuous maps), we can further argue that the global asymptotic stability implies uniformly global asymptotic stability (UGAS).

5. The well-posedness granted by the hybrid basic conditions additionally guarantees semi-global practical asymptotic stability of the system under small perturbations to the data. This robustness is a particularly desirable outcome, since variable pressure losses are virtually guaranteed to act as small pressure-perturbations for vine robots.

2.3 Simulation Studies

Simulations studies were carried out in MATLAB using the hybrid equation toolbox [6] to corroborate the stability analysis presented above. Three case studies were conducted to examine stability and robustness:

1. Simulation of the trajectory for the regular system as defined by (10) for 200 continuous time seconds starting at $z = 0$. This simulation serves to broadly validate the consistency of the system's behavior with our analytical predictions.

2. Simulation of the system with sinusoidal perturbations to the pressure, resulting in the following flow maps:

$$F_1 \coloneqq \begin{bmatrix} (max\{0, \frac{1}{2}p + \rho sin(t) - kz - c\})^n \\ k_p|z_{goal} - z| \\ 0 \end{bmatrix}$$

$$F_2 \coloneqq \begin{bmatrix} -(max\{0, -\frac{1}{2}p - \rho sin(t) - kz - c + T\})^n \\ -k_p|z_{goal} - z| \\ 0 \end{bmatrix}$$

   The objective of this study is to specifically examine the robustness properties predicted by well-posedness.

3. Simulation of the system with an on-off pressure controller replacing the proportional control law.

$$F_1 \coloneqq \begin{bmatrix} (\max\{0, \frac{1}{2}p - kz - c\})^n \\ \dot{p}_{max} \\ 0 \end{bmatrix}$$

$$F_2 \coloneqq \begin{bmatrix} -(max\{0, -\frac{1}{2}p - kz - c + T\})^n \\ -\dot{p}_{max} \\ 0 \end{bmatrix}$$

Where $\dot{p}_{max}$ is the maximum rate of change supported by the fan. This simulation is more pragmatic in nature, motivated by the fact that many fans are not capable of operating with an analog input.

Table 1 enumerates the constants used in the simulation studies.

k ($\frac{N}{m^3}$)	c ($\frac{N}{m^2}$)	L (m)	z_{goal} (m)	δ (m)	n	p_{max} (kPa)	p_{min} (kPa)	k_p	$\dot{p}_{max}$ (kPa)
0.1	0.1	100	50	1	1.1	20	0	1.0	1.0

Table 1: Default parameters used in simulation studies.

3 Results

Figure 3 shows the simulation results from the basic setup with the tail tension varied between 15-25N. The lower-bound on the tail tension is imposed by the fact that the retraction motors are selected to have sufficient torque to retract a fully unspooled robot. This means that the motor torque is greater than the minimum torque required for retraction when the robot is not fully unspooled. Similarly, figure 5 displays a similar parametric sweep using the on-off pressure controller instead of the proportional controller. Lastly, figure 4 is produced by varying the magnitude of pressure perturbation ρ. The implications of these results are discussed next.

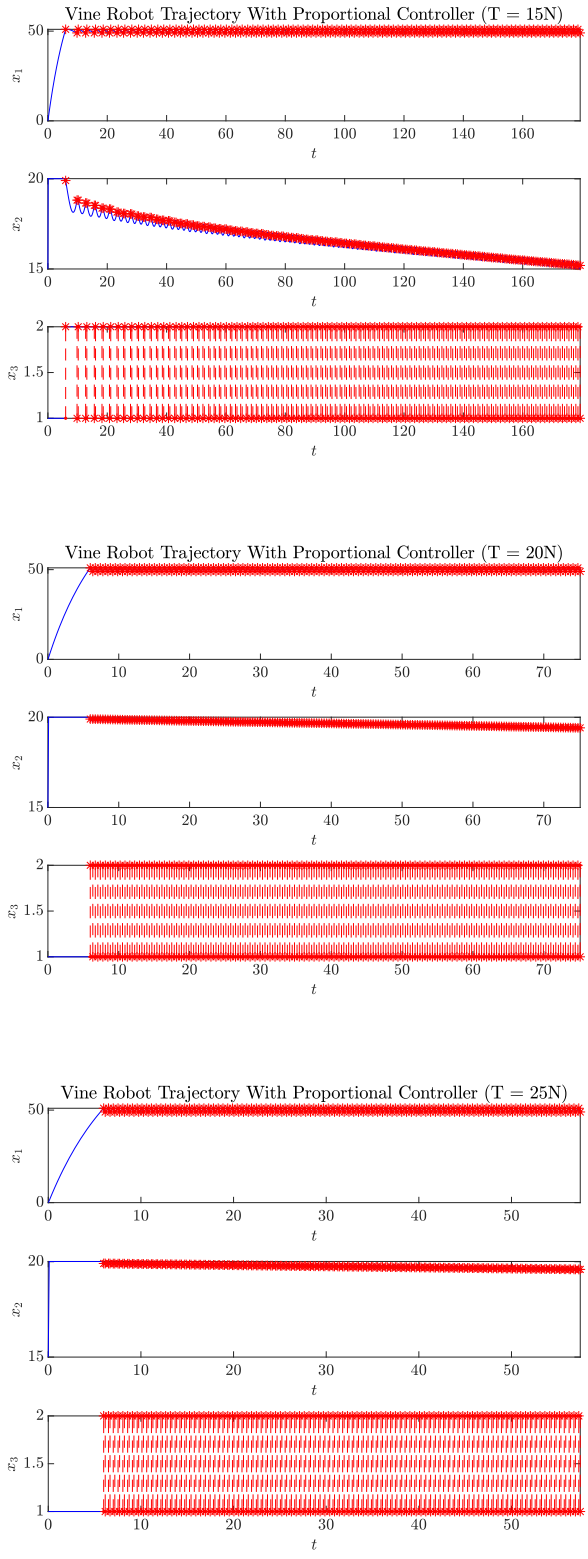

Figure 3: Predicted vine robot trajectories for 200 seconds under proportional pressure control at different tail tensions.

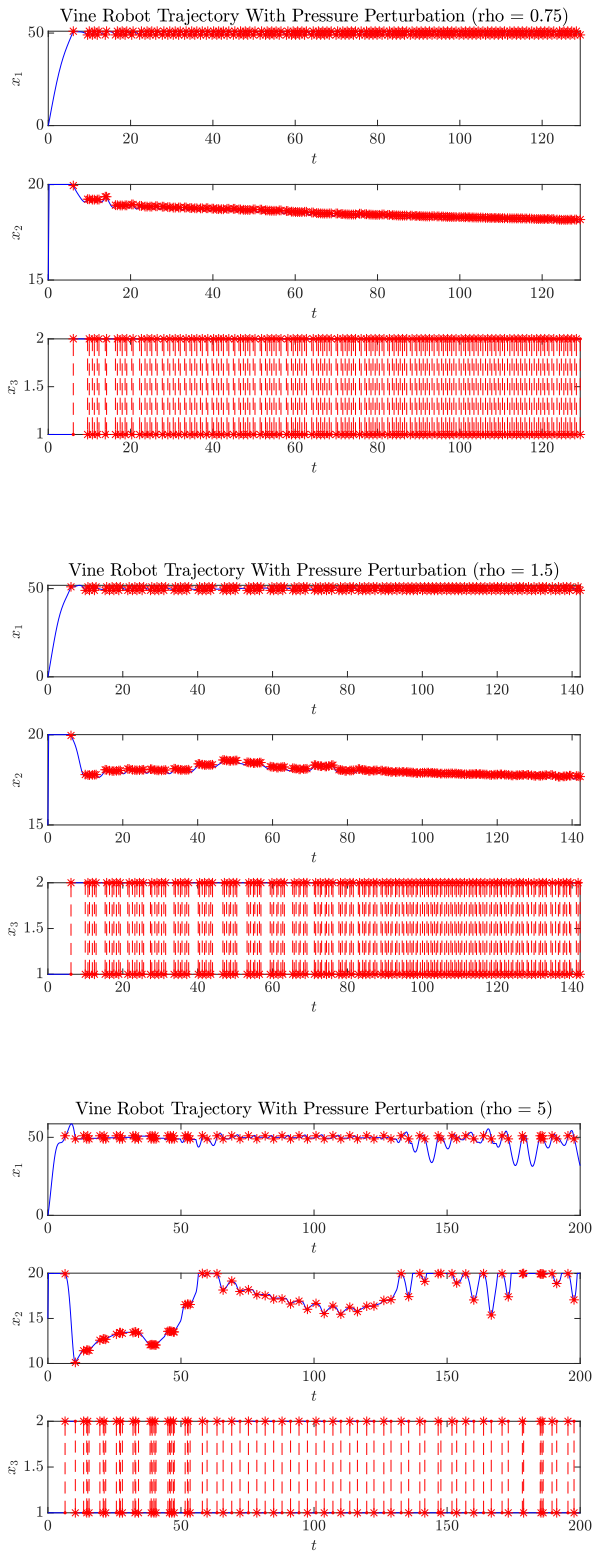

Figure 4: Predicted vine robot trajectories for 200 seconds under proportional pressure control at different magnitudes of pressure perturbation.

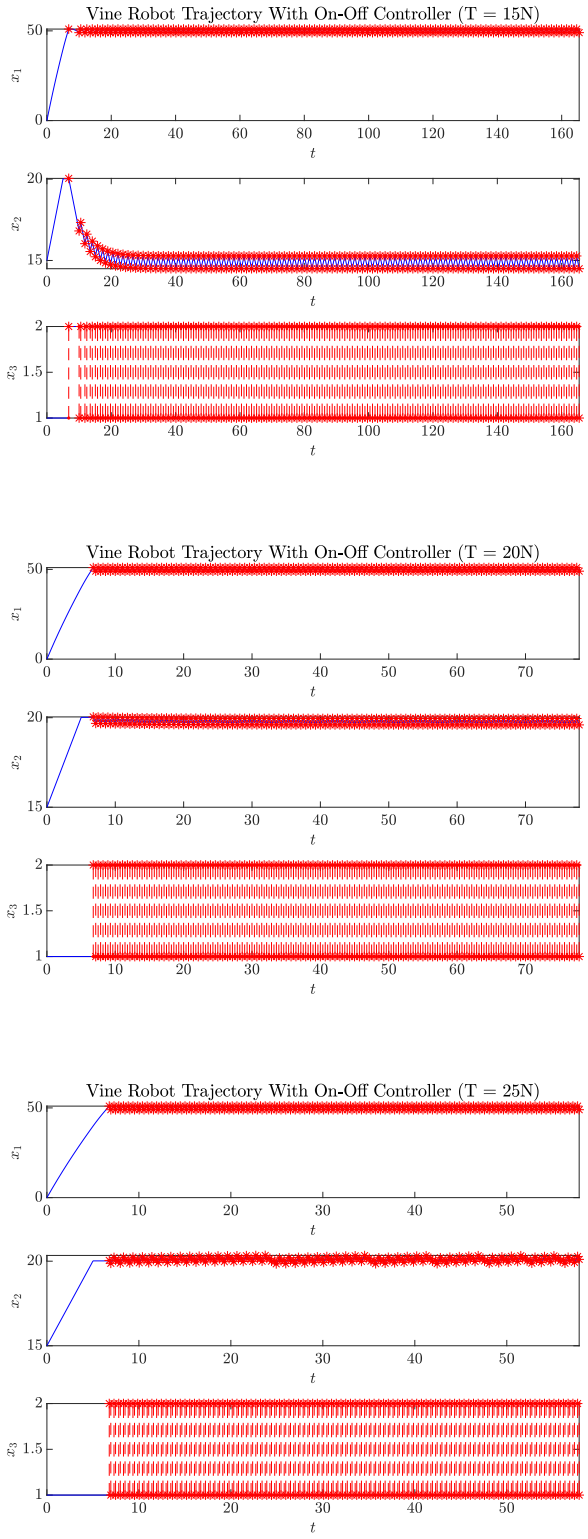

Figure 5: Predicted vine robot trajectories for 200 seconds under on-off pressure control at different tail tensions.

4 Discussion

The simulation results confirm the robot's convergence to the desired position under the action of the controller: starting from $z = 0$, the robot grows until reaching $z = 51$ and then continuously switches between growth and retraction while staying in the desired range of positions. A symmetric response is seen if the simulation is started at $z = 100$. Figure 3 highlights the increase in switching frequency caused by increasing the tail tension. The dependence of (8) on tail tension predicts this behavior, as higher tail tensions increase the magnitude of the positional derivative. Additionally, this shortening of the retraction intervals results in a higher steady-state pressure.

Figure 4 illustrates the robot's robustness to small pressure perturbations. With $\rho = 0.75$, almost no visible disturbance is made to the robot's position and $\rho = 1.5$ causes a modest variability in the switching periods. However, $\rho = 5$ is a large enough disturbance to negate the robustness under small perturbations that is guaranteed by well-posedness. Despite leaving the attractor under sudden pressure variation, the robot returns to the desired range of positions. This suggests that the controller maintains the global recurrence of the attractor - a weakened form of asymptotic stability due to the absence of forward invariance, but a desirable property nonetheless - even in presence of larger pressure perturbations. Presence of leaks that may vanish and re-appear periodically during growth and retraction is a major concern in control of vine robots; the results of this simulation study indicate the robot's resilience to such operating conditions. Lastly, Figure 5 illustrates the similarities between the performance of the on-off controller and the proportional controller. Note the widened band of steady state pressures in Figure 5 due to the simplified pressure control law.

5 Conclusion

We presented a simplified model of the dynamics of an apical extension robot, or vine robot, by treating it as a hybrid dynamical system. A switching control scheme was introduced to account for the limitations in sensing and actuation for large-diameter vine robots and a brief stability analysis of the controller was presented using the hybrid dynamics nomenclature. Lastly, simulation studies corroborated the analytical predictions and highlighted the robustness of the system to pressure perturbations.

This model can be further improving by representing the control loop as a hybrid sample-and-hold system. Additionally, more complex paths and tasks with time-varying attractors need to be examined to make this analysis more directly applicable to the tasks commonly performed by vine robots. However, the foregoing discussions confirm that the non-trivial switching dynamics of vine robots makes this modeling framework valuable for developing higher-level controllers and motion planners.

6 Appendix A: Hybrid Equation Toolbox Class

```
classdef vineRobot_LimitedK < HybridSystem
    %Modeling apical extension robot as a hybrid system.

    %Define variable properties that can be modified.
    properties
       k = 0.1; %Constant representing linear friction [N/m^3]
       c = 0.1; %Constant representing misc. friction interactions [N/m^2]
       t = 15; %Constant representing tension on the robot tail [N/m^2]
       L = 100; %Maximum length of the robot [m]
       zGoal = 50; %Desired attractor z-position [m]
       delta = 1; %Hysterisis band about the goal [m]
       kP = 0.5; %Proportional control constant [-]
       pMax = 20; %Maximum pressure accesible [kPa]
       pMin = 10; %Minimum pressure accesible [kPa]
       n = 1.1; %Empirical constant
    end

    %Define constant properties
    properties(SetAccess = immutable)
       zIndex = 1; %The index of position in the state vector
       pIndex = 2; %The index of pressure in the state vector
       qIndex = 3; %The index of operating mode in the state vector
    end

    methods
        function this = vineRobot_LimitedK()
            %Constructor for vineRobot class
            stateDim = 3; %Pass the number of state vector dimensions
            this = this@HybridSystem(stateDim);
        end

        function xdot = flowMap(this,x)
            z = x(this.zIndex); %Extract position
            p = x(this.pIndex); %Extract pressure
```

```
q = x(this.qIndex); %Extract mode
if q == 1
    %Impose maximum allowed pressure
    if p >= this.pMax && this.kP*(this.zGoal - z) >= 0
        if 0.5*p - this.k*z - this.c>0 %No retraction in growth
            xdot = [(0.5.*p - this.k.*z - this.c).^this.n;...
                0;...
                0]; %Define flow map in growth mode
        else
           xdot = [0;0;0];
        end
    %Impose minimum allowed pressure
    elseif p <= this.pMin && this.kP*(this.zGoal - z) <= 0
        if 0.5*p - this.k*z - this.c>0 %No retraction in growth
            xdot = [(0.5.*p - this.k.*z - this.c).^this.n;...
                0;...
                0]; %Define flow map in growth mode
        else
            xdot = [0;0;0];
        end
    else
        if 0.5*p - this.k*z - this.c>0 %No retraction in growth
            xdot = [(0.5.*p - this.k.*z - this.c).^this.n;...
                this.kP.*(this.zGoal - z);...
                0]; %Define flow map in growth mode
        else
            xdot = [0;this.kP.*(this.zGoal - z);0];
        end
    end
else
    %Impose maximum allowed pressure
    if p >= this.pMax && this.kP*(this.zGoal - z) >= 0
        if 0.5*p - this.t + this.c + this.k*z < 0
        %No growth in retraction
            xdot = [(0.5.*p - this.t + this.c + this.k.*z)
            .^this.n;...
```

```
                    0;...
                    0]; %Define flow map in retraction mode
                else
                    xdot = [0;0;0];
                end
            %Impose minimum allowed pressure
            elseif p<= this.pMin && this.kP*(this.zGoal - z) <= 0
                if 0.5*p - this.t + this.c + this.k*z < 0
                %No growth in retraction
                    xdot = [(0.5.*p - this.t + this.c +
                    this.k.*z)
                    .^this.n;...
                        0;...
                        0]; %Define flow map in retraction mode
                else
                    xdot = [0;0;0];
                end
            else
                if 0.5*p - this.t + this.c + this.k*z < 0
                %No growth in retraction
                    xdot = [(0.5.*p - this.t + this.c + this.k.*z)
                    .^this.n;...
                        this.kP.*(this.zGoal - z);...
                        0]; %Define flow map in retraction mode
                else
                    xdot = [0;this.kP.*(this.zGoal - z);0];
                end
            end
        end
    end

    function xplus = jumpMap(this,x)
       z = x(this.zIndex); %Extract position
       p = x(this.pIndex); %Extract pressure
       q = x(this.qIndex); %Extract mode
       xplus = [z;p;3-q]; %Toggle mode using simple switching rule
```

```
        end

        function C = flowSetIndicator(this,x)
            z = x(this.zIndex); %Extract position
            q = x(this.qIndex); %Extract mode

            %Choose flow-set about the goal point according to mode
            if ( (z <= (this.zGoal + this.delta) ) && (q == 1) ||...
                    (z >= (this.zGoal - this.delta) ) && (q == 2))
                C = 1;
            else
                C = 0;
            end
        end

        function D = jumpSetIndicator(this,x)
           z = x(this.zIndex); %Extract position
           q = x(this.qIndex); %Extract mode

           %Choose jump-set about the goal point
           if ( (z >= (this.zGoal + this.delta) ) && (q == 1) ||...
                    (z <= (this.zGoal - this.delta) ) && (q == 2))
               D = 1;
           else
               D = 0;
           end
        end
    end
end
```

7 Appendx B: MATLAB Script

```
    %Run an instance of the vine robot hybrid system class

%Re-define variables
k = 0.1; %Constant representing linear friction [N/m^3]
c = 0.1; %Constant representing misc. friction interactions [N/m^2]
t = 15; %Constant representing tension on the robot tail [N/m^2]
L = 100; %Maximum length of the robot [m]
zGoal = 50; %Desired attractor z-position [m]
delta = 1; %Hysterisis band about the goal [m]
kP = 0.5; %Proportional control constant [-]
pMax = 20; %Maximum pressure accesible [kPa]
pMin = 10; %Minimum pressure accesible [kPa]
n = 1.1; %Empirical constant

%Create a vineRobot object
sys = vineRobot_LimitedK();

%Check that C and D are correct
xGrowthC = [10; 0; 1]; %C in growth
sys.assertInC(xGrowthC);
sys.assertNotInD(xGrowthC);

xRetractionC = [90; 0; 2]; %C in retraction
sys.assertInC(xRetractionC);
sys.assertNotInD(xRetractionC);

xGrowthD = [90; 0; 1]; %D in growth
sys.assertNotInC(xGrowthD);
sys.assertInD(xGrowthD);

xRetractionD = [10; 0; 2]; %D in retraction
sys.assertNotInC(xRetractionD);
sys.assertInD(xRetractionD);
```

```
%Compute a solution

%Initial condition
x0 = [0, 15, 1];
%Time spans
tspan = [0, 200];
jspan = [0, 200];

%Specify solver options
config = HybridSolverConfig('AbsTol', 1e-3, 'RelTol', 1e-7);

%Compute solution
sol = sys.solve(x0, tspan, jspan, config);

%Plot the solutions
figure(1)
clf
hpb = HybridPlotBuilder();
hpb.title('Vine Robot Trajectory With Proportional Controller')...
    .subplots('on')...
    .plotFlows(sol)
```

References

[1] M. M. Coad, L. H. Blumenschein, S. Cutler, J. A. R. Zepeda, N. D. Naclerio, H. El-Hussieny, U. Mehmood, J.-H. Ryu, E. W. Hawkes, and A. M. Okamura, "Vine robots," *IEEE Robotics & Automation Magazine*, vol. 27, no. 3, pp. 120–132, 2020.

[2] R. Goebel, R. G. Sanfelice, and A. R. Teel, "Hybrid dynamical systems," *IEEE Control Systems Magazine*, vol. 29, no. 2, pp. 28–93, 2009.

[3] M. Coad, R. Thomasson, L. Blumenschein, N. Usevitch, E. Hawkes, and A. Okamura, "Retraction of soft growing robots without buckling," 10 2019.

[4] L. H. Blumenschein, A. M. Okamura, and E. W. Hawkes, "Modeling of bioinspired apical extension in a soft robot," in *Living Machines*, 2017.

[5] R. Goebel, R. G. Sanfelice, and A. R. Teel, *Hybrid Dynamical Systems: Modeling, Stability, and Robustness.*, 2012.

[6] R. Sanfelice, D. Copp, and P. Nanez, "A toolbox for simulation of hybrid systems in matlab/simulink," in *Proceedings of the 16th international conference on Hybrid systems: computation and control - HSCC '13*. ACM Press, 2013. [Online]. Available: https://doi.org/10.1145%2F2461328.2461346