Dynamic Modeling and Control of Aerial Manipulator for Phytobiopsy

Daniel S. Cohen

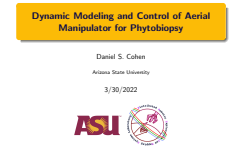
Arizona State University

3/30/2022



2023-09-16

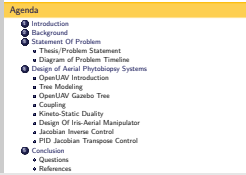
Your Short Title



1. Thank you Dr. Das for the introduction, it my pleasure today to present my thesis on Dynamic modeling and Control of an Aerial Manipulator for Phytobiopsy

Agenda

1. Introduction
2. Background
3. Statement Of Problem
   - Thesis/Problem Statement
   - Diagram of Problem Timeline
4. Design of Aerial Phytobiopsy Systems
   - OpenUAV Introduction
   - Tree Modeling
   - OpenUAV Gazebo Tree
   - Coupling
   - Kineto-Static Duality
   - Design Of Iris-Aerial Manipulator
   - Jacobian Inverse Control
   - PID Jacobian Transpose Control
5. Conclusion
   - Questions
   - References



1. We will start off by going into the background then cover my thesis and problem statement
2. Then we will cover the design process of the Aerial Phytobipsy system
3. We will then rap up with the conclusion and will take questions.

Introduction

- Aerial Manipulation
- Phytobiopsy



Figure 1: Phytobiopsy Performed by UPenn GRASP Lab [3]

Introduction
- Aerial Manipulation
- Phytobiopsy

Figure 1: Phytobiopsy Performed by UPenn GRASP Lab [3]

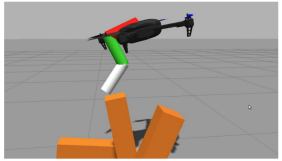
1. At the highest level an Aerial Arboreal Manipulator will need to be able to exhibit the ability manipulate objects while in loft while while also maintaining the connection to the tree long enough to perform data acquisition

Background

- Phytobiopsy: Sampling tree leaves by grasping and holding the leaf while the sample is prepared.
  - Maximizing the economic benefits to farms while minimizing impact to environment [1].
- Aerial Manipulator: Aerial manipulators are aerial robot systems furnished with a robotic manipulator that is capable of performing manipulation tasks while in flight.
- Aerial Arboreal Manipulator that must work with Aerial Arboreal Physical Interactions (AAPhI).



Pistachio Tree In Need Of Health Quantification



Aerial Arboreal Manipulator

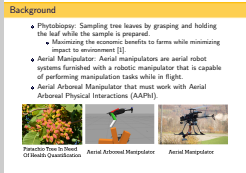


Aerial Manipulator

2023-09-16

Your Short Title
└Background

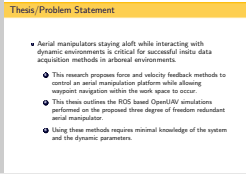
└Background



1. Phytobiopsy: Sampling tree leaves by grasping and holding the leaf while the sample is prepared
2. This leads to having an impact economically to farm and to the environment
3. This leads to Aerial Arboreal Physical interactions with the Aerial Arboreal Manipulator

Thesis/Problem Statement

- Aerial manipulators staying aloft while interacting with dynamic environments is critical for successful insitu data acquisition methods in arboreal environments.
  1. This research proposes force and velocity feedback methods to control an aerial manipulation platform while allowing waypoint navigation within the work space to occur.
  2. This thesis outlines the ROS based OpenUAV simulations performed on the proposed three degree of freedom redundant aerial manipulator.
  3. Using these methods requires minimal knowledge of the system and the dynamic parameters.



1. To achieve an aerial manipulator for phytobiopsy this thesis purposes velocity and force feedback methods with a 3-DOF redundant manipulator build upon the ROS based Open UAV framework. It should also be noted methods do not reacquire knowledge of the dynamic parameters of the tree

Design of Aerial Manipulators

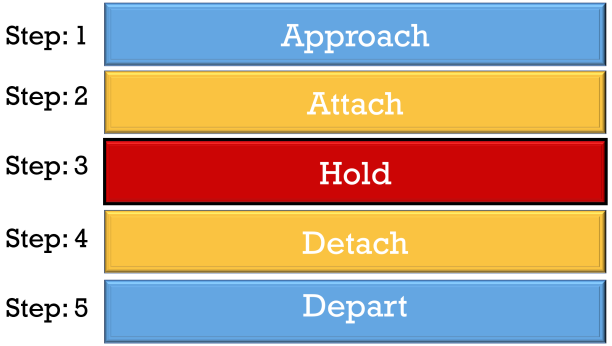
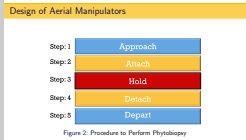


Figure 2: Procedure to Perform Phytobiopsy



1. The general procedure is to approach, attach, hold, detach then depart.
2. This thesis will covers approaching the tree, the attaching to it and then purposed methods for holding onto the tree.

Dynamic Modeling Via Euler Lagrange

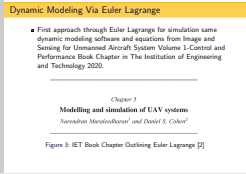
- First approach through Euler Lagrange for simulation same dynamic modeling software and equations from Image and Sensing for Unmanned Aircraft System Volume 1-Control and Performance Book Chapter in The Institution of Engineering and Technology 2020.

Chapter 5

Modelling and simulation of UAV systems

Narendran Muraleedharan[1] and Daniel S. Cohen[2]

Figure 3: IET Book Chapter Outlining Euler Lagrange [2]



1. The first approach was to use Euler Lagrange to simulate the full system based on my previous publications while looking into the framework of the problem

Design of Aerial Manipulators Cont Euler Lagrange

- Generalized coordinates to form state vector/establish body fixed frames

$$\gamma = \left[x, z, \vartheta, \alpha_1, \alpha_2\right]^T \tag{1}$$

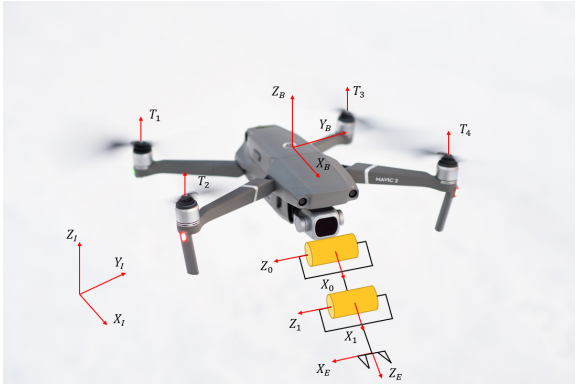
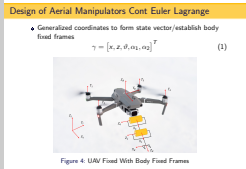


Figure 4: UAV Fixed With Body Fixed Frames



1. First generalized coordinate need to be established, this is the state vector gamma. Frames were also established at each of the chosen locations

Euler Lagrange

- Starting with the state vector and establish body fixed frames.

$$ {}^{I}_{I}R_1 = {}^{I}_{I}R_{UAV} + {}^{I}T_{UAV}{}^{UAV}T_1{}^{UAV}_{UAV}R_1 \tag{2} $$

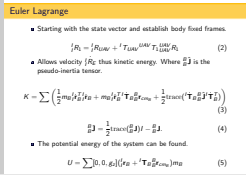
- Allows velocity ${}^{I}_{I}\dot{R}_E$ thus kinetic energy. Where ${}^{B}_{B}\hat{\mathbf{J}}$ is the pseudo-inertia tensor.

$$ K = \sum \left(\frac{1}{2} m_B {}^{I}_{I}\dot{\mathbf{r}}^{T}_{B} {}^{I}_{I}\dot{\mathbf{r}}_B + m_B {}^{I}_{I}\dot{\mathbf{r}}^{T}_{B} {}^{I}\dot{\mathbf{T}}_B {}^{B}_{B}\mathbf{r}_{cm_B} + \frac{1}{2}\mathrm{trace}({}^{I}\dot{\mathbf{T}}_B {}^{B}_{B}\hat{\mathbf{J}} {}^{I}\dot{\mathbf{T}}^{T}_{B}) \right) \tag{3} $$

$$ {}^{B}_{B}\hat{\mathbf{J}} = \frac{1}{2}\mathrm{trace}({}^{B}_{B}\mathbf{J})I - {}^{B}_{B}\mathbf{J}. \tag{4} $$

- The potential energy of the system can be found.

$$ U = \sum [0, 0, g_z]({}^{I}_{I}\mathbf{r}_B + {}^{I}\mathbf{T}_B {}^{B}_{B}\mathbf{r}_{cm_B}) m_B \tag{5} $$



1. Once the kinematics of the robotics system can found through IRRI the velocity IIRI dot can be found all the way to the end-effector. This can done by chain differentials
2. This allows the kinetic of the energy to be found to each body fixed frame
3. Where BBJhat is the pseudo inertia tensor
4. This leads to the findings the potential energy of the robotic system

Euler Lagrange Cont.

$$\mathbf{f} = \left(\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{\gamma}}\right) - \frac{\partial L}{\partial \gamma}\right)^T \tag{6}$$

$$\mathbf{H}(\gamma) = \left(\frac{\partial}{\partial \dot{\gamma}}\left(\frac{\partial K}{\partial \dot{\gamma}}\right)^T\right) \tag{7}$$

$$\mathbf{d}(\gamma, \dot{\gamma}) = \left(\frac{\partial K}{\partial \gamma}\left(\frac{\partial K}{\partial \dot{\gamma}}\right)^T\right)\dot{\gamma} - \left(\frac{\partial K}{\partial \gamma}\right)^T \tag{8}$$

$$\mathbf{g}(\gamma) = \left(\frac{\partial u}{\partial \gamma}\right)^T \tag{9}$$

$$\mathbf{f} = \mathbf{H}(\gamma)\ddot{\gamma} + \mathbf{d}(\gamma, \dot{\gamma}) + \mathbf{g}(\gamma) \tag{10}$$

$$\begin{bmatrix}\gamma, & \dot{\gamma}\end{bmatrix}^T = \begin{bmatrix}x, z, \vartheta, \alpha_1, \alpha_2, \dot{x}, \dot{z}, \dot{\vartheta}, \dot{\alpha_1}, \dot{\alpha_2}\end{bmatrix}^T \tag{11}$$

Euler Lagrange Cont.
$\mathbf{f} = \left(\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{\gamma}}\right) - \frac{\partial L}{\partial \gamma}\right)^T$ (6)
$\mathbf{H}(\gamma) = \left(\frac{\partial}{\partial \dot{\gamma}}\left(\frac{\partial K}{\partial \dot{\gamma}}\right)^T\right)$ (7)
$\mathbf{d}(\gamma, \dot{\gamma}) = \left(\frac{\partial K}{\partial \gamma}\left(\frac{\partial K}{\partial \dot{\gamma}}\right)^T\right)\dot{\gamma} - \left(\frac{\partial K}{\partial \gamma}\right)^T$ (8)
$\mathbf{g}(\gamma) = \left(\frac{\partial u}{\partial \gamma}\right)^T$ (9)
$\mathbf{f} = \mathbf{H}(\gamma)\ddot{\gamma} + \mathbf{d}(\gamma, \dot{\gamma}) + \mathbf{g}(\gamma)$ (10)
$\begin{bmatrix}\gamma, & \dot{\gamma}\end{bmatrix}^T = \begin{bmatrix}x, z, \vartheta, \alpha_1, \alpha_2, \dot{x}, \dot{z}, \dot{\vartheta}, \dot{\alpha_1}, \dot{\alpha_2}\end{bmatrix}^T$ (11)

1. This culminates in subtracting between the kinetic and potential to find the lagrange of the system.
2. Though a common formulation but the picture can be improved by separating out the compensates
3. Where H is the system mass matrix, d is the vector of Coriolis and centripetal and g is the vector of gravitation terms
4. This leads to the classic equation commonly refered to as the robotics equation
5. Once Gamma double dot was isolated I wrote my own RK45 dynamics simulator to find gamma and gamma dot over time.

UAV and 2-DOF Simulation with RK45 with Euler Lagrange (Arm Angle:45 deg,45 deg)

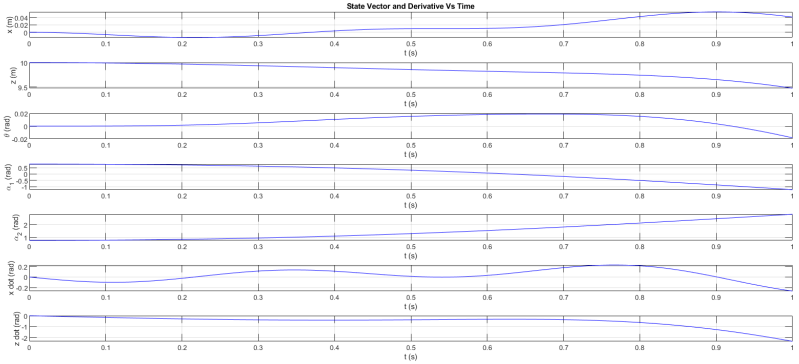
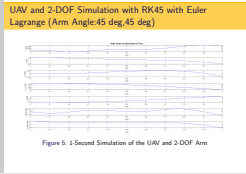


Figure 5: 1-Second Simulation of the UAV and 2-DOF Arm



1. The states were simulated with the UAV with two the two joints at 45 degrees up. While a constant force applied in the Z direction.
2. This is powerful method but would take 15 mins to produce one second of simulation, and this does not include the third DOF and the need for the paffian constraint for the tree contact and the tree.
3. To take this further, matlab would be too slow for real time simulations, but can still have value.

OpenUAV Introduction

- Open browser based design studio built from the Robot Operating System (ROS 1 Melodic),Gazebo, and PX4 autopilot flight stack. First of it kind.



Middleware



Robot Simulation

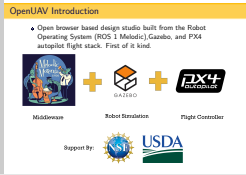


Flight Controller

Support By:



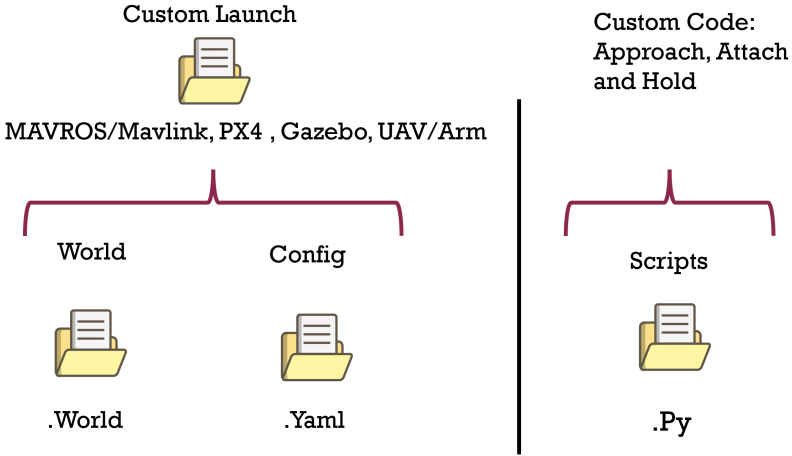


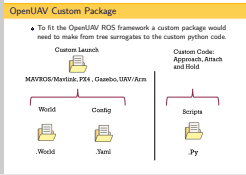


1. To perform the real time simulation the first of its kind OpenUAV ROS based design studio
2. The software stack is comprised of the ROS Melodic as the middleware, Gazebo for the dynamic simulations and Robot Controls. Along with the PX4 Flight Controller
3. This worked has has been supported by NSF and USDA

OpenUAV Custom Package

- To fit the OpenUAV ROS framework a custom package would need to make from tree surrogates to the custom python code.





1. Building out and testing the custom aerial arboreal package based on the OpenUAV can now be laid out block by out
2. The core of this package is the custom Iris UAV XML Xacro and the python code use to run the full system, but this also means building out the custom world and the yaml file to tie unique transmission types controllers in URDF Unified Robot Description Format (URDF) XML Xacro to the python code and Gazebo

Tree Modeling

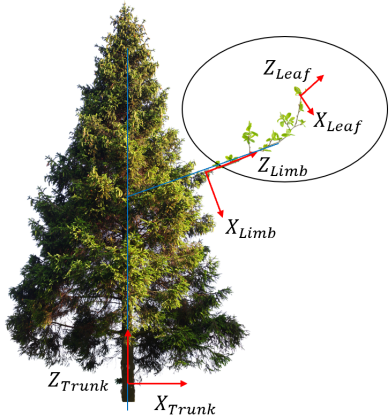


Figure 6: Tree Overview/Frames

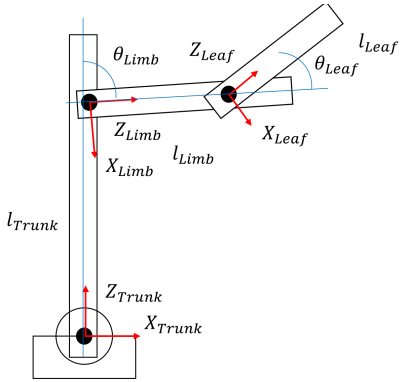
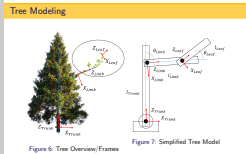


Figure 7: Simplified Tree Model



1. To model then Aerial Arboreal Physical interactions tree models were constructed by breaking the tree into three critical system, The trunk, limb and the leaf
2. The literature typically shows the kinematic model to clarify the connection, as for to utilize this model draw out the dynamics they typically outline a one degree of freedom system

Tree Modeling

- Modeling the tree as a Liner Time Invariant System

$$\dot{x}(t) = Ax(x) + Bu(t) \quad (12)$$

$$y(t) = Cx(x) + Du(t) \quad (13)$$

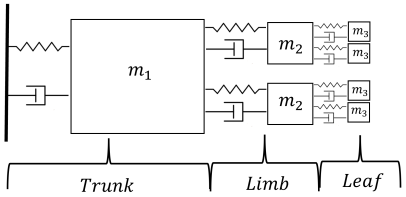


Figure 8: 7-DOF Spring Mass Damper System

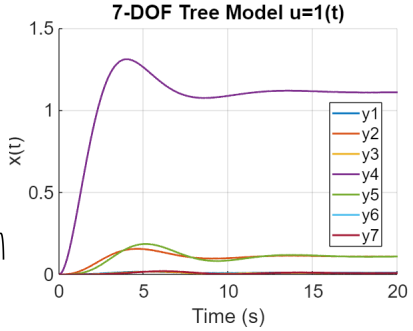
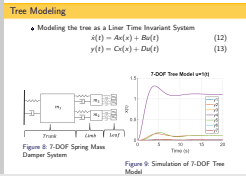


Figure 9: Simulation of 7-DOF Tree Model



1. It is more common to model the system as spring mass damper LTI system and that what did by building the 7 DOF Spring mass damper model where an excitation was place on the top most leaf mass.
2. The

OpenUAV Gazebo Tree

- Two Tree System Designed From URDFs XML and then merged into the Gazebo World File.
- The Static Tree represents connecting to truck of the tree
- The Gazebo Linear Spring Tree: 7.0 (Nm/rad), 0.7 (Nm/rad), and 0.07 (Nm/rad)

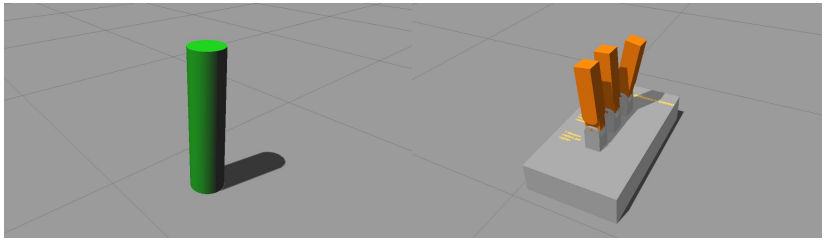


Figure 10: Static Tree

Figure 11: Linear Spring Tree

OpenUAV Gazebo Tree
- Two Tree System Designed From URDFs XML and then merged into the Gazebo World File.
- The Static Tree represents connecting to truck of the tree
- The Gazebo Linear Spring Tree: 7.0 (Nm/rad), 0.7 (Nm/rad), and 0.07 (Nm/rad)



Figure 10: Static Tree Figure 11: Linear Spring Tree

1. This yields two testing situations,the static tree and the constant mass less spring
2. The linear spring 7.0 (Nm/rad), 0.7 (Nm/rad), and 0.07 (Nm/rad)

Tree Coupling

- Strong, Loose, and Momentary
- Manipulator End-effecor is attached to the tree

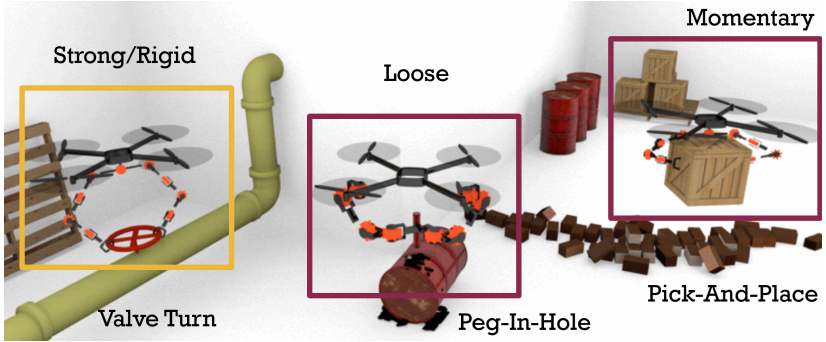
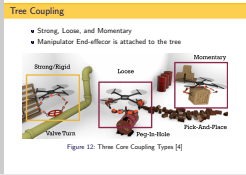


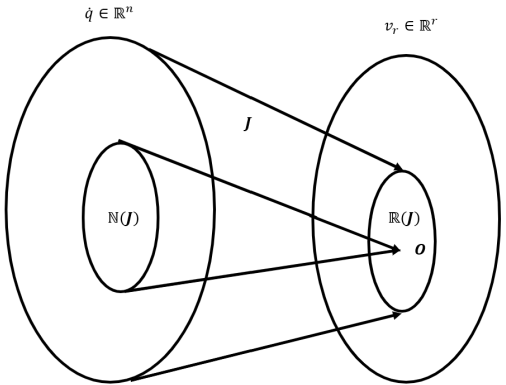
Figure 12: Three Core Coupling Types [4]

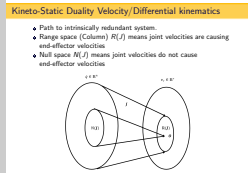


1. Coupling to the tree is also a critical design choice, typically there are thee main coupling types, Strong, where the system is more rigidly coupled to the object and the can pivot.
2. Loose coupling is more prevalent in peg-in-hole applications, where the robot can bounce between physical and no constraint
3. Lastly, there is momentary coupling, like the right most robot where the CG shifts but the robot can still move in all directions freely.
4. For my work,I consider the Strong coupling between the Aerial Manipulator and the tree system.

Kineto-Static Duality Velocity/Differential kinematics

- Path to intrinsically redundant system.
- Range space (Column) $R(J)$ means joint velocities are causing end-effector velocities
- Null space $N(J)$ means joint velocities do not cause end-effector velocities

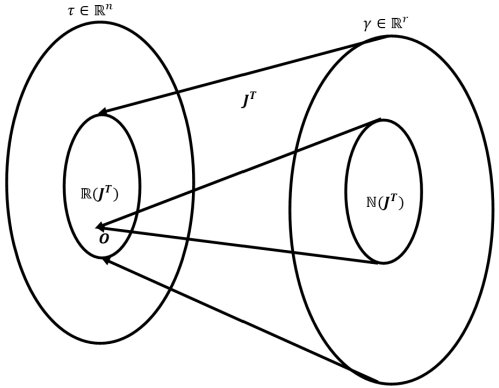


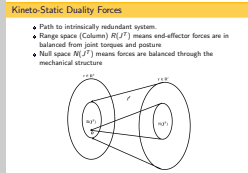


1. Developing methods to allow the UAV to performed well under these conditions was the next goal. The path to an an intrinsically redundant manipulator and control system was through the classic differential kinematics and the kineto-static duality
2. The start of the path is with the differential kinematics mapping through the jacobian.Where the joint velocities can be linearly mapped to the end-effecctor velocities
3. Where the Column space means the joint velocities would be causing end-effector velocities and the null space would mean the joint velcoties do not cuase and end-effector velocities

Kineto-Static Duality Forces

- Path to intrinsically redundant system.
- Range space (Column) $R(J^T)$ means end-effector forces are in balanced from joint torques and posture
- Null space $N(J^T)$ means forces are balanced through the mechanical structure





1. A linear relationship can also be drawn between the joint torques and the end-efector forces and torques, this is done with the jacobian transpose
2. The column space would be the end-effector forces are balanced via the joint torques and the posture
3. Where as the null space would beam the forces balance through the mechanical structure

Design Of Iris-Aerial Manipulator

- Intrinsically Redundant for Task.

$$dim\{Operational\ Space\} < dim\{Joint\ Space\} \qquad (14)$$

ASU

Figure 15: 2-DOF with position controller

ASU

Figure 16: 3-DOF with position controller

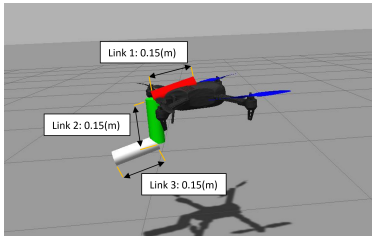
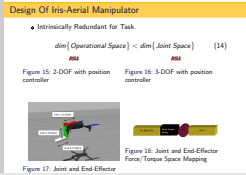


Figure 17: Joint and End-Effector



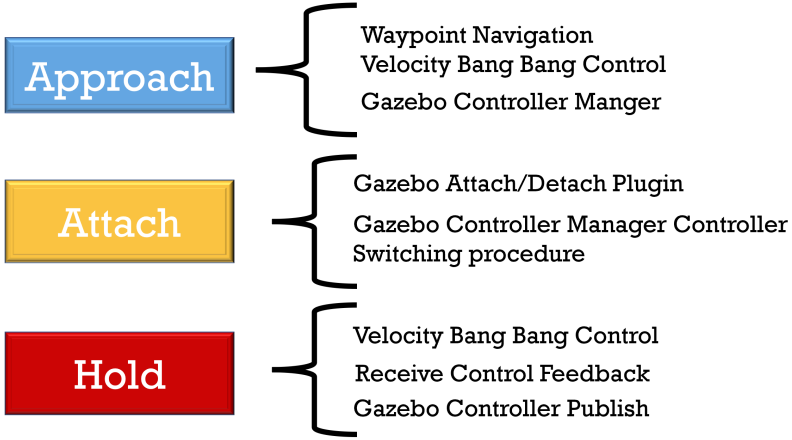
Figure 18: Joint and End-Effector Force/Torque Space Mapping

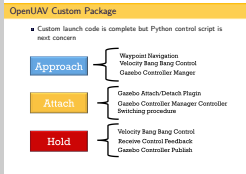


1. The quadcopter really operates in the works space of x,z. I it can pitch but that does not add to the space. A three-DOF planar arm is functional redundant if the End-effector position is specified as there is more then one configuration the robot can be in to meet the workspace demands. In the static tree case, pitch is less of concern and is not something we design for in the steady state to maintain. Best control authority is over going vertical. Thus we will conecct with with a linear velocy bang bang control and allow the PX4 flight contoler to handel this while the three DOF arm pluse the one DOF tree to allow the are to move in the necessary manner.
2. Thus a 3 DOF arm can be made in gazebo and a Force Torque sensor with the. One issue is that due to the ridge connection the gazebo force torque sensor show zero forces so it was placed between the base link or link 1

OpenUAV Custom Package

- Custom launch code is complete but Python control script is next concern





1. The launch codes and controller now now be developed
2. At this point, once the PX4 flight system has been armed and switched to off board mode my custom python code could then actively allows UAV to approach the tree and to boot up the gazebo controller in the base mode before it switch during attachment
3. Once python code calculated that it close enough attachment the gazebo attach/detach plugin is set a ros-service request to start, causing the strong coupling between the last link of the robot and the tree.
4. Once attached the holding control code for the UAV and arm could then be activated. That is the last piece to the frame worked needed

Jacobian Inverse Control

$$J = \begin{bmatrix} -a1sin(b)-a1sin(a)-a3sin(c) & -a2sin(b)-a3sin(c) & -a3sin(c) \\ a1cos(b)+a1cos(a)+a3cos(c) & a2cos(b)+a3cos(c) & a3cos(c) \\ 1 & 1 & 1 \end{bmatrix} \quad (15)$$

$$a=q1,\ b=q1+q2,\ c=q1+q2+q3 \quad (16)$$

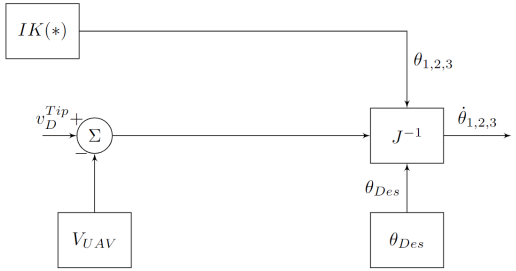
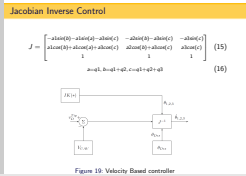


Figure 19: Velocity Based controller



1. This leads to the development of a jacobian that would describe the manipulator and be usable for control systems.
2. The first control system outlined in my thesis is the jacobian inverse condole with the feedback from the UAV linear velocities. This can be seen in Figure 19
3. This control system is actively works to negate the UAV velocities given the desired zero tip velocities with also a desired total theta (adding up each joint angle)
4. Though simple,this system works in static test and in the 0.7 (Nm/rad)

Jacobian Inverse Control (Static Test 2)

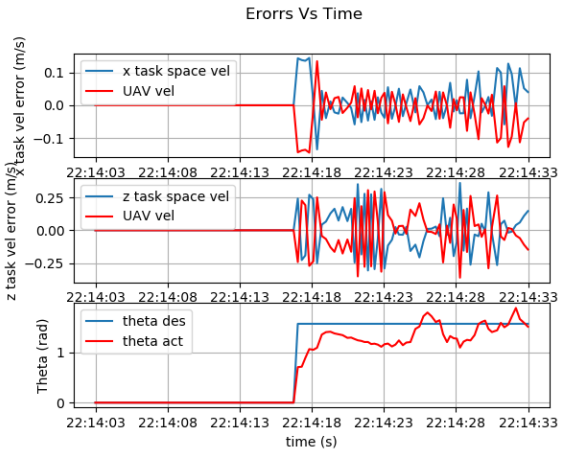
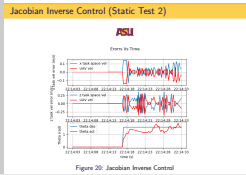


Figure 20: Jacobian Inverse Control



1. This control system can now be fully simulated in the OpenUAV framework from a central python document.
2. For all testing cases the UAV was asked move half of a link length away.
3. Play video
4. In each case the Z axis is globally north and the X axis is with the arm and towards the tree surrogate

Jacobian Inverse Control (High Test 4)

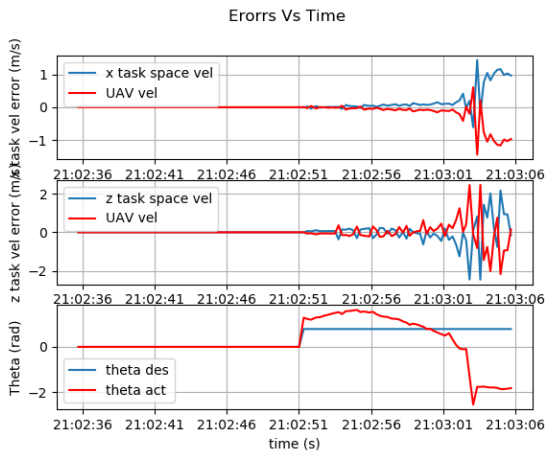
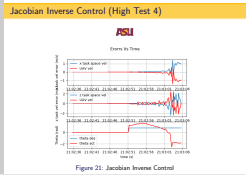


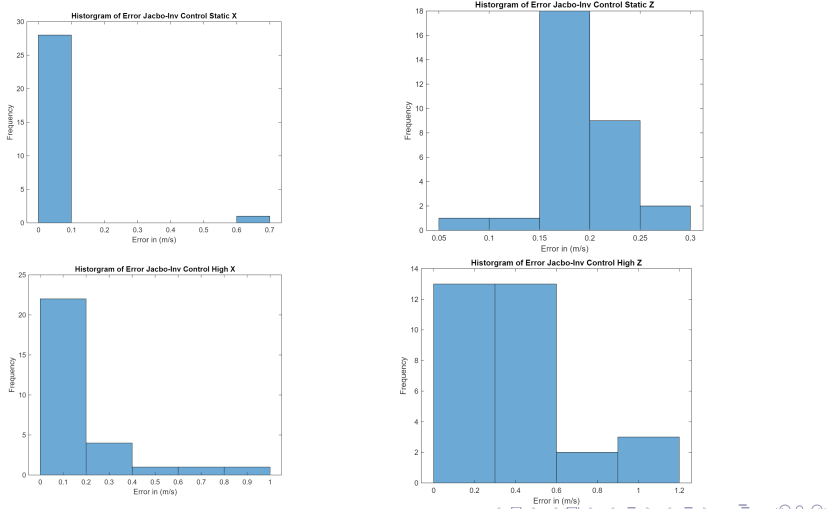
Figure 21: Jacobian Inverse Control

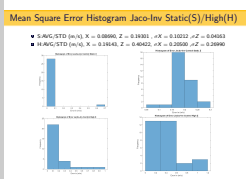


1. In the high test case 0.7 (Nm/rad) the aerial manipulator was able to attach and work to nullify the UAV velocities. This can be seen in the X and Z axis, furthermore the total theta be seen working hard to track to descried but is system can still able to function

Mean Square Error Histogram Jaco-Inv Static(S)/High(H)

- S:AVG/STD (m/s), X = 0.08690, Z = 0.19301 , $\sigma X = 0.10212$,$\sigma Z = 0.04163$
- H:AVG/STD (m/s), X = 0.19143, Z = 0.40422, $\sigma X = 0.20500$,$\sigma Z = 0.26990$





1. After thirty test for each of the working testing conditions the histograms of the error can be analyzed
2. The top two figures show the histograms of the errors found in the X axis and Z axis in meters per second
3. In this case you can see in the X axis error is relatively low and of the type error, whereas the Z axis is more of the classic bell curve
4. The bottom two figures show the error histograms for the High tree system 0.7(Nm/rad) with the Error in the meters per second.
5. The X-error is characteristic of the static case but the Z error is much falter, this shows the system starting to have some problems with the manipulation task as the Med and low testing case was not able succeed in long term testing.

PID Jacobian Transpose Control

$$\tau = J^T(q_{1,2,3})(K_p Fe + K_i \int F_e dt + K_d F_e) \quad (17)$$

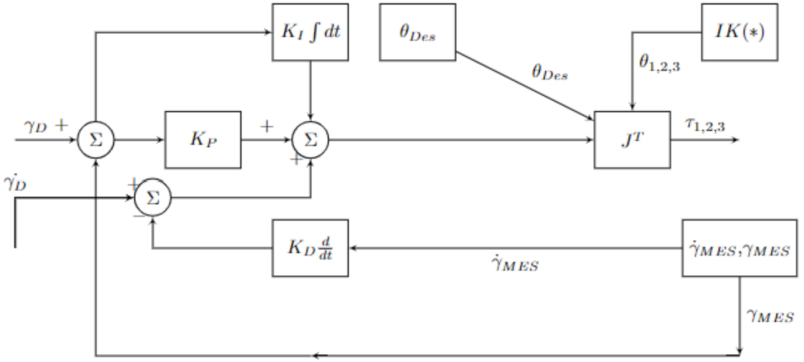
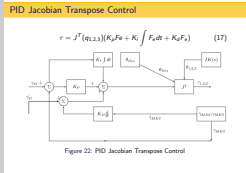


Figure 22: PID Jacobian Transpose Control



1. Using the kentio-static dualitiy the second control system was build to decrease the forces between the robot and the UAV using Gazebo Based force torque sensor for control feedback. This feedback can then be passed through the PID gain block and then Jacbobain transpose to generate the responding torques

Plots of PID Jacobian Transpose Control (Static Test 14)

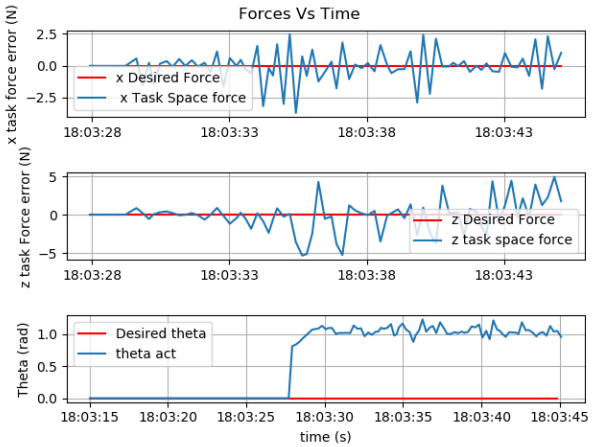
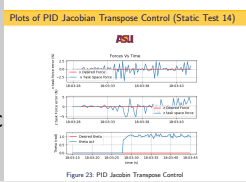


Figure 23: PID Jacobin Transpose Control



1. The using this control system and the framework that has been built up to this point the UAV can now fully fly to tree attach and then work decrease the forces felt between the UAV and the arm.
2. Play video
3. The desired and measured forces in newtons for both axis can now be seen over time verse and measured and desired total arm angle
4. Even though the UAV is moving the x-axis forces still have the general harmonics but is control system it working to mitigate the forces felt
5. In the Z case the control system has to more rapidly respond the UAV with the bang bang flight controller
6. The static case shows the the shows that the success in the conditions is less tied directly to the desired theta can be tuned, however one tun was used over all tests

Plots of PID Jacobian Transpose Control (High Test 29)

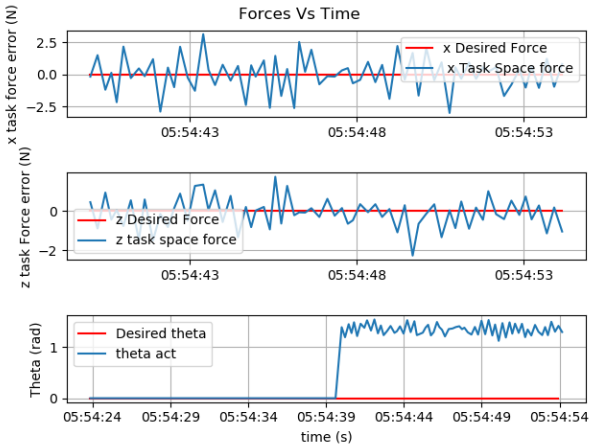
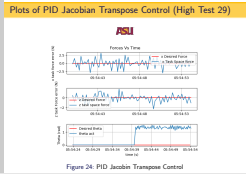


Figure 24: PID Jacobin Transpose Control



1. The same control system with the same control system tune can be seen in Figure 24.
2. Play video
3. Where the system can better compensate for the tree ability to pivot but once again the arm does not exert to much force to push the system back to the zero set point but is close to the roll over condition where the 2pi total goes to zero

Plots of PID Jacobian Transpose Control (Med Test 19)

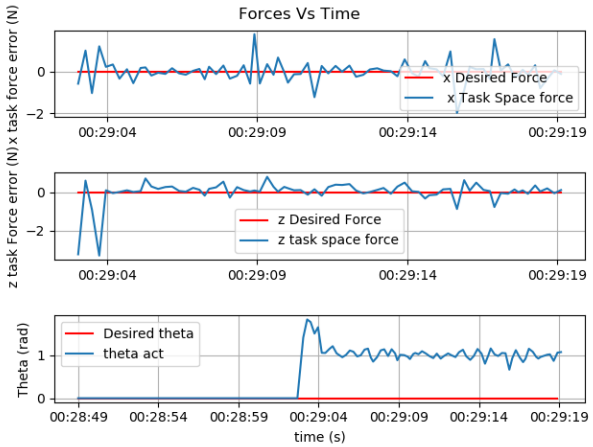
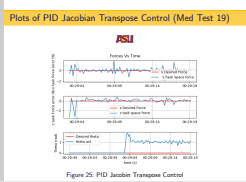


Figure 25: PID Jacobin Transpose Control



1. The same control system with the same control system tune can be seen in Figure 25.
2. Play video
3. The force error plots for the X and Z axis both show the larger force to start but as time goes on the PID Transpose control system is able to work towards lowers the forces felt between the UAV and the arm.
4. In this case the spring based tree is still applying more constant force over time and the arm and adjust better but is still not able to bring the total arm angle, but that is perfectly fine.

Plots of PID Jacobian Transpose Control (Low Test 5)

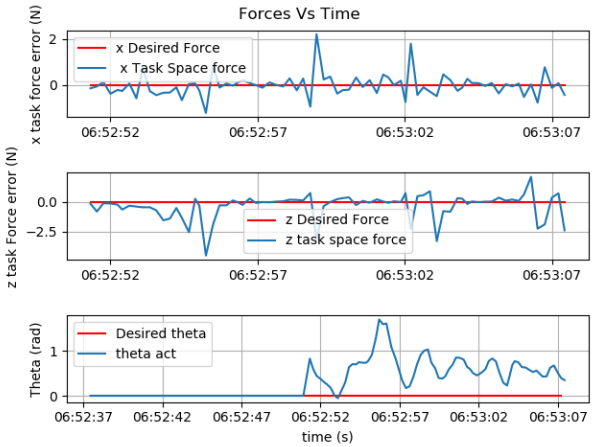
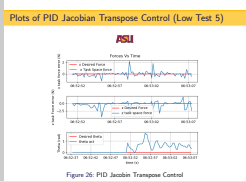


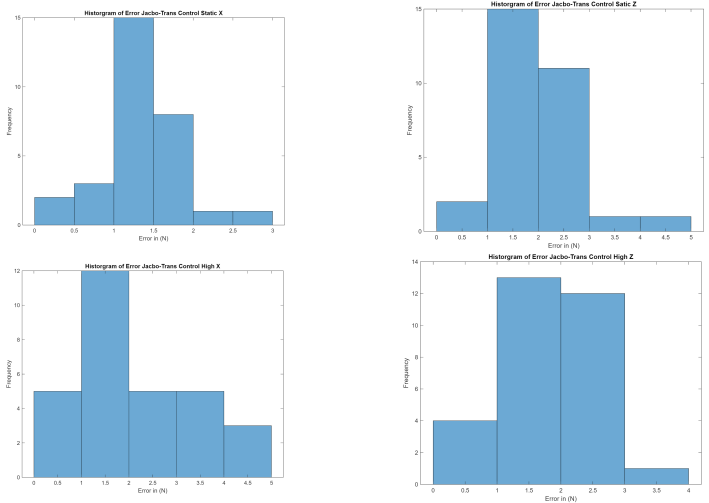
Figure 26: PID Jacobin Transpose Control

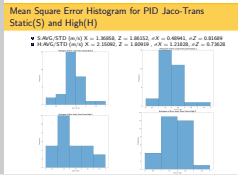


1. The same control system with the same control system tune can be seen in Figure 26.
2. Play video
3. In the lowest sping tree case of 0.07 Nm/rad the UAV and arm now have much more control over the system but the larger gain values used to Handel the other cases is starting to effect the error dynamics but it is still able to work toward both goals of decreasing the forces in the system while working to move the arm back to a inital zero arm angle.
4. In this low tree case the arm is less compliant then it could be to but fit all testing cases the tune still is able to be effective and keep the UAV aloft
5. The derivative gain now could now be clearly decreased in this case

Mean Square Error Histogram for PID Jaco-Trans Static(S) and High(H)

- S:AVG/STD (m/s) X = 1.36858, Z = 1.86152, $\sigma X = 0.48941$, $\sigma Z = 0.81689$
- H:AVG/STD (m/s) X = 2.15092, Z = 1.80919 , $\sigma X = 1.21828$, $\sigma Z = 0.73628$

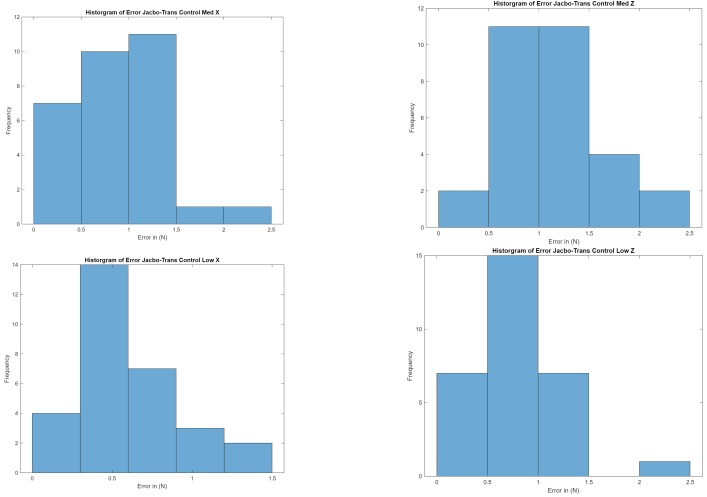


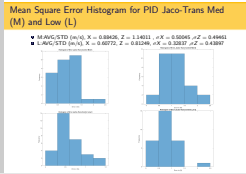


1. The MSE Histograms for the PID Jaco-Trans control system can now be shown after thirty tests for each control system the Histograms of the error in newtons can be shown
2. The top two histograms show the error of the static of the static tree case. The avg and std of these testing conditions can also be found in top bullet point
3. The static testing histograms show the error being more centered over 2.5 N but with some skewness toward the zero error line
4. The bottom two histograms show the error of the static of the high tree case. The avg and std of these testing conditions can also be found in lower bullet point
5. The high testing histograms measured in newtons show the error being more centered over 1.5 in the X-axis and over 2 in N for the Z-axis.
6. Like in many testing certain attachment configurations can cause larger error values while other place the arm in a better initial

Mean Square Error Histogram for PID Jaco-Trans Med (M) and Low (L)

- M:AVG/STD (m/s), X = 0.88426, Z = 1.14011 , $\sigma X = 0.50045$,$\sigma Z = 0.49461$
- L:AVG/STD (m/s), X = 0.60772, Z = 0.81249, $\sigma X = 0.32837$,$\sigma Z = 0.43897$



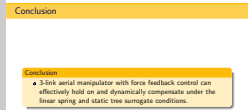


1. The MSE Histograms for the PID Jaco-Trans control system can now be shown after thirty tests for each control system the Histograms of the error in newtons can be shown The avg and std of these testing conditions can also be found in top bullet point
2. The Med test can be seen in the top two figs, where the control system of the arm in the X has the highest skewnesss toward the zero error line on the far left. Where as the Z is more exhibits the more typical bell curve
3. The bottom two histograms show the error of the low tree case. The avg and std of these testing conditions can also be found in lower bullet point
4. The low testing histogram shows the forces decreasing with the less stiff spring but now the tune in the arm may be causes more error, namely from the derivative term

Conclusion

Conclusion

- 3-link aerial manipulator with force feedback control can effectively hold on and dynamically compensate under the linear spring and static tree surrogate conditions.



1. The main conclusion is that by integrating a three link manipulator onto the iris UAV with the force feedback the system was able to handle each of the testing conditions, this success is more limited with the velocity based control system
2. Together with custom OpenUAV package this allows aerial arboreal research to continue.

Future Research

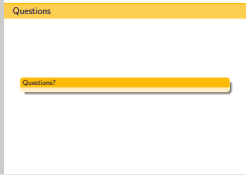
Future Research

- Increase dynamic parameters to find balance between control authority.
- Add in gravity compensation term in and dynamic compensation.
- Test under loose coupling by allowing end-effector pivoting and using the attach/detach in quick succession.
- Compliance / Impedance control.

Future Research
Future Research
- Increase dynamic parameters to find balance between control authority.
- Add in gravity compensation term in and dynamic compensation.
- Test under loose coupling by allowing end-effector pivoting and using the attach/detach in quick succession.
- Compliance / Impedance control.

Questions

Questions?



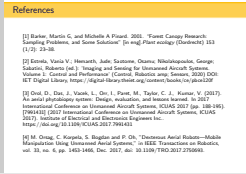
References

[1] Barker, Martin G, and Michelle A Pinard. 2001. "Forest Canopy Research: Sampling Problems, and Some Solutions" [in eng].*Plant ecology* (Dordrecht) 153 (1/2): 23–38.

[2] Estrela, Vania V.; Hemanth, Jude; Saotome, Osamu; Nikolakopoulos, George; Sabatini, Roberto (ed.): 'Imaging and Sensing for Unmanned Aircraft Systems. Volume 1: Control and Performance' (Control, Robotics amp; Sensors, 2020) DOI: IET Digital Library, https://digital-library.theiet.org/content/books/ce/pbce120f

[3] Orol, D., Das, J., Vacek, L., Orr, I., Paret, M., Taylor, C. J., Kumar, V. (2017). An aerial phytobiopsy system: Design, evaluation, and lessons learned. In 2017 International Conference on Unmanned Aircraft Systems, ICUAS 2017 (pp. 188-195). [7991431] (2017 International Conference on Unmanned Aircraft Systems, ICUAS 2017). Institute of Electrical and Electronics Engineers Inc.. https://doi.org/10.1109/ICUAS.2017.7991431

[4] M. Orsag, C. Korpela, S. Bogdan and P. Oh, "Dexterous Aerial Robots—Mobile Manipulation Using Unmanned Aerial Systems," in IEEE Transactions on Robotics, vol. 33, no. 6, pp. 1453-1466, Dec. 2017, doi: 10.1109/TRO.2017.2750693.


References

[1] Barker, Martin G, and Michelle A Pinard. 2001. "Forest Canopy Research: Sampling Problems, and Some Solutions" [in eng].Plant ecology (Dordrecht) 153 (1/2): 23–38.

[2] Estrela, Vania V.; Hemanth, Jude; Saotome, Osamu; Nikolakopoulos, George; Sabatini, Roberto (ed.): 'Imaging and Sensing for Unmanned Aircraft Systems. Volume 1: Control and Performance' (Control, Robotics amp; Sensors, 2020) DOI: IET Digital Library, https://digital-library.theiet.org/content/books/ce/pbce120f

[3] Orol, D., Das, J., Vacek, L., Orr, I., Paret, M., Taylor, C. J., Kumar, V. (2017). An aerial phytobiopsy system: Design, evaluation, and lessons learned. In 2017 International Conference on Unmanned Aircraft Systems, ICUAS 2017 (pp. 188-195). [7991431] (2017 International Conference on Unmanned Aircraft Systems, ICUAS 2017). Institute of Electrical and Electronics Engineers Inc.. https://doi.org/10.1109/ICUAS.2017.7991431

[4] M. Orsag, C. Korpela, S. Bogdan and P. Oh, "Dexterous Aerial Robots—Mobile Manipulation Using Unmanned Aerial Systems," in IEEE Transactions on Robotics, vol. 33, no. 6, pp. 1453-1466, Dec. 2017, doi: 10.1109/TRO.2017.2750693.

Backup Slides

Backup Slides