Thrust Vector Control: Design, Code, and Construction on a Budget

Lundon Salley
College of Engineering
Embry-Riddle Aeronautical University
Prescott, Arizona
Email: salleyl@my.erau.edu

***Abstract*—Project Ananke is an in-progress project at Embry-Riddle Aeronautical University that is working on creating a rocket capable of autonomous flight using thrust vector control (TVC). By using two linear actuators to control the gimbal ring, the rocket engine is able to maneuver the rocket in both pitch and yaw. Autonomous control and monitoring of the rocket's attitude can be achieved through positional quaternions to adjust for simple attitude changes and deep neural networks for more computationally expensive or complex maneuvers. Re-purposing linear actuators from TV mounts can be used as a cheaper alternative for extremely expensive custom linear actuators. Finally, a simple Raspberry Pi can function as the primary micro-controller, handling the positional quaternions and pre-trained deep neural networks (DNNs).**

***Keywords*—** Thrust Vector Control, Quaternion, Neural Network

I. Introduction

Thrust Vector Control (TVC) has become increasingly popular for propulsion systems in the 21st century, as it allows for high-speed maneuvers, quick attitude changes, and even land rockets back on the ground for re-use [1]. Unfortunately for undergraduate students, TVC isn't cheap for non-model rockets because it requires powerful computing chips to run on-board calculations, precise sensors to determine the attitude of the rocket, custom machined parts, and extremely durable actuators to move the gimbal. On top of the construction costs, the programming requirements are no small feat either: using 4D coordinates to manipulate an object in 3D using only 2 linear actuators as seen in "Fig 1" [2]. Being able to successfully create a TVC module gives undergraduate students relevant experience in the aerospace engineering field and an opportunity to grow their skills for their future jobs.

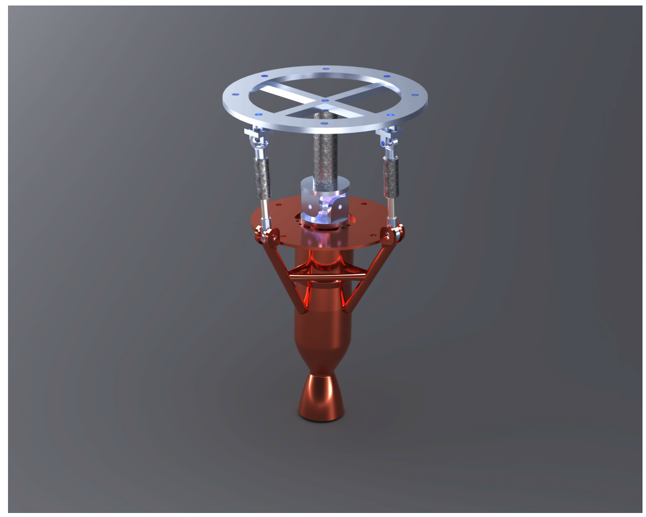


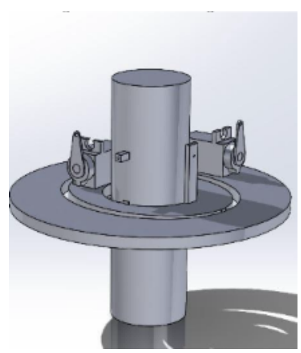
Fig. 1. Dual-Linear Actuator TVC Gimbal Design

II. Designing a Thrust Vector Control Gimbal

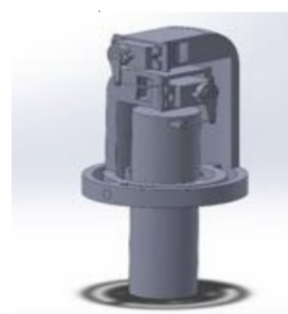
There are many different ways to configure a gimbal for a TVC system, as seen in "Fig 2", each with their own set of advantages and disadvantages. For example, "Fig 2 (a)" was rejected due to its small half-angle, large rings, and lack of stability, while "Fig 2 (b)" was accepted due to its more compact, durable, and much larger half-angle [3]. "Fig 1" is the design used as a rough reference in this research paper, which was chosen due to its mount to the rocket engine, and its simplicity.

"Fig 1" shows many similarities to "Fig 3," using linear actuators to manipulate the gimbal and a universal joint (U-Joint) to secure the engine in place. The advantage of this configuration is that the engine is putting its force into a secure central rod instead of suspending the engine as seen in "Fig 2".

Giving the engine a secure U-Join to push against allows for a much more powerful engine, such as the one to be used in "Fig 1". U-Joints work similarly to a ball joint, allowing freedom to move in its pitch and yaw directions, but limits the roll to prevent the actuators from ripping the whole module apart. It should be noted that many other TVC mechanisms generally use gimbal mounts, which allow rotation as well as pitch and yaw, but those functions were deemed unnecessary and too expensive for this project.



(a) Iteration 1



(b) Iteration 2

Fig. 2. Alternate Configurations for Gimbal [3]

Another struggle with using TVC is the heat transfer up the linear actuator's shaft. Normally, the divergent part of the rocket's nozzle is fully disconnected from the body, with only the throat connecting them. This poses an issue if the actuators are connected to enough heat for the shafts or connectors to deform. According to the findings of NASA's Jet Propulsion Laboratory (JPL), "The heat-transfer coefficient is a minimum upstream of the throat" with a "substantial decrease in heat transfer existed downstream of the point

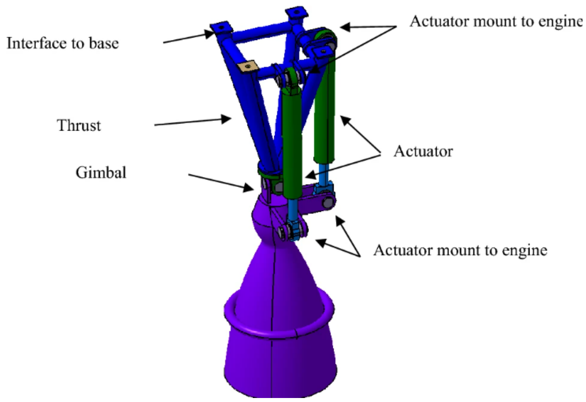


Fig. 3. Similar Gimbal Design to "Fig 1" [2]

of flow separation" as well [4]. Since the rocket fuel flows from the body of the engine, into the combustion tank, and out the nozzle, JPL's research shows that the insulation of the combustion chamber is significantly higher. This parallels with the extreme heat created in the combustion chamber, which could easily melt or deform the struts holding on the gimbal if not properly protected [5].

III. Understanding Quaternions

Programming actuators for TVC is not quite as straightforward as it may seem, due to an unfortunate fault in Euler's coordinate system. When approaching the manipulation of a gimbal using Euler's angles, a phenomenon called gimbal lock will occur, and will forever combine 2 axes into a single axis [6]. Needless to say, this less than ideal. Fortunately, there is a much simpler and more elegant solution using Sir William Rowan Hamilton's number system from 1843: quaternions. Quaternions are an entirely different coordinate system that uses 4 dimensions on 4 axes consisting of: 1, i, j, and k. Each letter represents $\sqrt{-1}$, just as i normally does, but instead as a vector that heads in the $\hat{\imath}$, $\hat{\jmath}$, and $\hat{k}$ (x, y, and z) directions respectively, just as they would a normal vector on Cartesian coordinates. The 1 works as the quaternion's identity element.

↓ X →	1	i	j	k
1	1	i	j	k
i	i	-1	k	-j
j	j	-k	-1	i
k	k	j	-i	-1

Fig. 4. Quaternion Multiplication Table

"Fig 4" displays the quaternion multiplication table, demonstrating how multiplication is non-commutative for quaternions. Other than preventing gimbal lock, quaternions help reduce the computational load that using matrix multiplication with trigonometric functions would use.

IV. Using Quaternions for Control

Firstly, some quaternion definitions

$$q = q_w + q_x\boldsymbol{i} + q_y\boldsymbol{j} + q_z\boldsymbol{k} \tag{1}$$

$$q = \begin{bmatrix} q_w & q_x & q_y & q_z \end{bmatrix}^T \tag{2}$$

$$\bar{q} = \begin{bmatrix} q_w & -q_x & -q_y & -q_z \end{bmatrix}^T \tag{3}$$

(1) and (2) are the most common representations of quaternions, with (3) displaying the quaternion's conjugate [7].

$$p \otimes q = \begin{bmatrix} p_w q_w & - & p_x q_x & - & p_y q_y & - & p_z q_z \\ p_w q_x & + & p_x q_w & + & p_y q_z & - & p_z q_y \\ p_w q_y & - & p_x q_z & + & p_y q_w & + & p_z q_x \\ p_w q_z & + & p_x q_y & - & p_y q_x & + & p_z q_w \end{bmatrix} \tag{4}$$

(4) is a product of quaternions p and q, called the Hamilton product which is denoted by $\otimes$ [7]. As shown in "Fig 4", $p \otimes q \neq q \otimes p$ because quaternions are non-commutative.

To program the controller for the rocket will require a few different inputs from the onboard sensors to create an observed rocket attitude quaternion, denoted as q_{obs}. The location quaternion will be the desired attitude the rocket should be at, which will be denoted as q_{loc}. Finally, the difference between those attitudes will be denoted as q_{err}. The goal quaternion can be found by taking the Hamilton product of the observed attitude and the error between them. Since the gimbal will be moving by q_{err}, solving for q_{err} gives us:

$$q_{err} = \bar{q}_{obs} \otimes q_{loc} \tag{5}$$

Next, the shortest rotation path must be calculated using a simple if-else statement.
if ($q_{w_{err}} < 0$):

$$\begin{aligned} q_{w_{err_m}in} &= -q_{w_{err}} \\ q_{x_{err_m}in} &= -q_{x_{err}} \\ q_{y_{err_m}in} &= -q_{y_{err}} \\ q_{z_{err_m}in} &= -q_{z_{err}} \end{aligned} \tag{6}$$

else:

$$\begin{aligned} q_{w_{err_m}in} &= q_{w_{err}} \\ q_{x_{err_m}in} &= q_{x_{err}} \\ q_{y_{err_m}in} &= q_{y_{err}} \\ q_{z_{err_m}in} &= q_{z_{err}} \end{aligned} \tag{7}$$

Using the quaternion derivative $\dot{q}_{loc}$ to represent the rate of rotation of attitude proportional to the error. Introducing proportional gain denoted as K_p, the equation below appears.

$$\dot{q}_{loc} = K_p \cdot q_{err_{min}} \tag{8}$$

Using the rate of rotation as an angular rate vector, the above values can be substituted in, given that q_u is an unrotated quaternion:

$$\omega_{loc} = 2\bar{q}_u \otimes \dot{q}_{loc} \tag{9}$$

Finally, formulas (8) and (9) can be manipulated to create equation (10), as seen below.

$$\begin{bmatrix} \omega_{x_{loc}} \\ \omega_{y_{loc}} \\ \omega_{z_{loc}} \end{bmatrix} = 2 \cdot K_p \cdot \begin{bmatrix} q_{x_{err_{min}}} \\ q_{y_{err_{min}}} \\ q_{z_{err_{min}}} \end{bmatrix} \tag{10}$$

The equation in (10) takes the inputs from error quaternions and turns them into rates of rotation for each axis, allowing the gimbal move accordingly. Since quaternions generally give the shortest rotation with the shortest path, using them can drastically lowering the computational requirements for the on-board computer.

V. Artificial Intelligent Control

Implementing Artificial Intelligent Control (AIC) would require a massive amount of training data and computing power to fully control the entire rocket, but implementing it for specific parts of its flight can be beneficial. One benefit of using AIC is that it can account for the lacking real-time performance of sequential convex optimization algorithms [8]. Using a Deep Neural Network (DNN), it can be trained to “approximate the optimal actions and steer the spacecraft to the target in real time” [9]. DNNs can vastly improve upon normal algorithms by discovering patterns about what actions should be done at certain times, which can easily outperform a clunky algorithm.

DNNs are not currently planned to be used on this project, but it is something that may be added on if landing proves to be an issue. DNNs can be easily implemented using PyTorch, a Python library built around using neural networks, tensors, and other various machine learning techniques.

VI. Construction

Major aerospace companies such as SpaceX or NASA that use TVC have access to many different methods to create the parts needed to make their system. They can machine the parts on-site, order them from another company, or request another company to make the part for them. Fortunately for most large companies, most of these issues can be solved with money, which is something that college students famously do not have.

A. Linear Actuators

Two linear actuators are used to maneuver the gimbal in “Fig 1”, mounted 90° apart to allow control in both pitch and yaw. Due to the thrust forces of the rocket engine, the linear actuator need to be able to withstand high forces, and be able to change direction very quickly to react to the volatility of the rocket. Finally, the half-angle requirement for this project is 11°, meaning that the linear actuators need to be long enough to move the gimbal enough to meet the 11° requirement. Progressive Automation’s High Force Industrial Linear Actuator adequately fulfills the requirements, costing only around $500 each and having many different length and force options to choose from.

B. Micro-controller

Arduino and Raspberry Pi are two of the most popular micro-controllers for students, and for a good reason. Code can be programmed and ran off of both of them, all without spending hundreds of dollars on custom PCB boards. Choosing which one to choose can be a challenge, as they both share many similarities. Arduino boasts of its easy-to-use interface and massive open source libraries, while Raspberry Pi displays its higher computational power and its extensive adaptability [10]. Due to the better mathematical computational capabilities of the Raspberry Pi, it will be used on this project as the primary micro-controller.

VII. Conclusions

Building a thrust vector controlled rocket on a tight budget isn’t easy, but can be done through vigorous research, building off of successes and failures of others that have tried similar approaches. Designing a robust TVC unit required using two linear actuators mounted 90° apart with a universal joint holding the rocket engine to the body. The placement of the two linear actuators at a 90° angle allows the attitude of the rocket to change proportionally to the length of the actuators with the help of the universal joint to allow the multi-directional movement. Programming the linear actuators to autonomously control the rocket’s attitude can be achieved by using quaternions to keep computation expenses to a minimum and by applying mixed learning algorithms such as deep neural networks to help assist with the more difficult sections. Finally, the gimbalated system can be constructed from repurposed TV mount linear actuators, a Raspberry Pi, a universal joint, and various other parts.

Acknowledgments

I would like to thank Ryan Coder for the wonderful CAD model in “Fig‘1” and for helping me understand the finer details of TVC. I would also like to acknowledge YouTube channel “3Blue1Brown” and Ben Eater for their incredible videos and interactive websites for helping understand quaternions. Finally, I want to give a huge shout out to everyone on the Project Ananke for their phenomenal work and effort put into to complete the project.

References

[1] Z. Wang, Y. Jia, L. Jin, and J. Duan, “Thrust vector control of upper stage with a gimbaled thruster during orbit transfer,” *Acta Astronautica*, vol. 127, pp. 359–366, 2016, iD: 271447. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0094576516305264

[2] A. Krammer, L. Blecha, and M. Lichtenberger, “Fin actuation, thrust vector control and landing leg mechanisms design for the retalt vtvl launcher,” *CEAS space journal*, vol. 14, no. 3, pp. 577–591, 2022. [Online]. Available: https://link.springer.com/article/10.1007/s12567-021-00421-0#citeas

[3] C. H. Reddy, Y. S. Shreeya, P. V. Reddy, P. S. Rao, and P. S. Raju, “Design and analysis of thrust vector control system,” *AIP Conference Proceedings*, vol. 2659, -10-07 2024.

[4] L. H. Back, P. F. Massier, and H. L. Gier, “Convective heat transfer in a convergent-divergent nozzle,” *International Journal of Heat and Mass Transfer*, vol. 7, no. 5, pp. 549–568, 1964, iD: 271451. [Online]. Available: https://www.sciencedirect.com/science/article/pii/0017931064900523

[5] W. Wessels, “How rocket engines stay cool and don’t melt,” *Headed for Space*, 04/04/ 2022. [Online]. Available: https://headedforspace.com/how-rocket-engines-stay-cool/

[6] D. S. Brezov, C. D. Mladenova, and I. M. Mladenov, “New perspective on the gimbal lock problem,” in *New perspective on the gimbal lock problem*, vol. 1570. Melville: American Institute of Physics, 2013, p. 374. [Online]. Available: https://go.exlibris.link/qGlWkypp

[7] M. Gołabek, M. Welcer, C. Szczepański, M. Krawczyk, A. Zajdel, and K. Borodacz, “Quaternion attitude control system of highly maneuverable aircraft,” *Electronics*, vol. 11, no. 22, p. 3775, 2022. [Online]. Available: https://www.proquest.com/scholarly-journals/quaternion-attitude-control-system-highly/docview/2739418908/se-2

[8] W. Li, Y. Song, L. Cheng, and S. Gong, “Closed-loop deep neural network optimal control algorithm and error analysis for powered landing under uncertainties,” *Astrodynamics*, vol. 7, no. 2, pp. 211–228, 2023, iD: Li2023. [Online]. Available: https://doi.org/10.1007/s42064-022-0153-1

[9] L. Cheng, Z. Wang, and F. Jiang, “Real-time control for fuel-optimal moon landing based on an interactive deep reinforcement learning algorithm,” *Astrodynamics*, vol. 3, no. 4, pp. 375–386, 2019, iD: Cheng2019. [Online]. Available: https://doi.org/10.1007/s42064-018-0052-2

[10] geeksforgeeks, “Difference between arduino and raspberry pi,” *geeks for geeks*, 04 Sep, 2024. [Online]. Available: https://www.geeksforgeeks.org/difference-between-arduino-and-raspberry-pi/