Technology injection in amateur rocket design and fabrication – an exploratory study of 3D printing, thrust vector control, onboard flight computer and rocket motor in the design and fabrication of DIY rockets

Deemo CHEN

Ningbo Hanvos Kent School, Ningbo, Zhejiang, China

E-mail: deemosea@gmail.com

Abstract

An exploratory study of injecting new technologies in the design and fabrication of amateur rockets is performed. Four emerging technologies are considered, namely 3D printing, thrust vector control, onboard flight computer and re-engineered rocket motors. It is shown that lightweight, custom-made parts manufactured by 3D printing accelerate amateur rocket building at a lower cost. The thrust vector control equips the rocket with enhanced capability to maintain a steady posture during ascending and landing. Autonomous controls are enabled by the onboard flight computer, and rocket motors capable of providing sustained thrust are argued to play a key role in securing a smooth landing. Through this heavily heads-on project, a valuable set of experience is obtained. Furthermore, the project demonstrates that technology injection has a fascinating potential in revolutionizing norms adopted in the design and fabrication of traditional amateur rockets.

Table of Contents

Table of Contents

1 Introduction

As the payload fairing jettisons, the bright sunshine lights up the spaceman in a spacesuit literally speeding in a glittering Tesla roadster (more than 6000 miles per hour), with our very home planet earth glowing blue in the background: the scene captured by the onboard camera taken in space is indistinguishable from a science fiction blockbuster (Fig. 1 (a)). Yet behind the scene, the two side boosters, which just accelerated the payload into space, quietly descended back to Cape Canaveral, with controlled pace and elegant style, and landed firmly next to the launch pad (Fig. 1 (b)). Sending objects into space has become a mature technology [1], while reusing rocket cores and boosters have only become feasible in recent years. The enablers are undoubtedly new technologies, most notably advanced manufacturing of lighter and more durable parts [2] and complicated autonomous control [3] that helps rockets and boosters self-stabilize during their re-entry and landing. The injection of new technology pioneered by SpaceX has re-energized the slow-paced, exclusively government-backed and somewhat "less exciting" rocket industry, and inspired many competitors, both domestic (Blue Origin, United Launch Alliance) and international (Ariane Group, LinkSpace) [4]. More significantly, the success of SpaceX is a live testimony that technology injection empowers a paradigm shift, in that norms followed in the past may no longer be the most ideal in the presence of emerging technology.

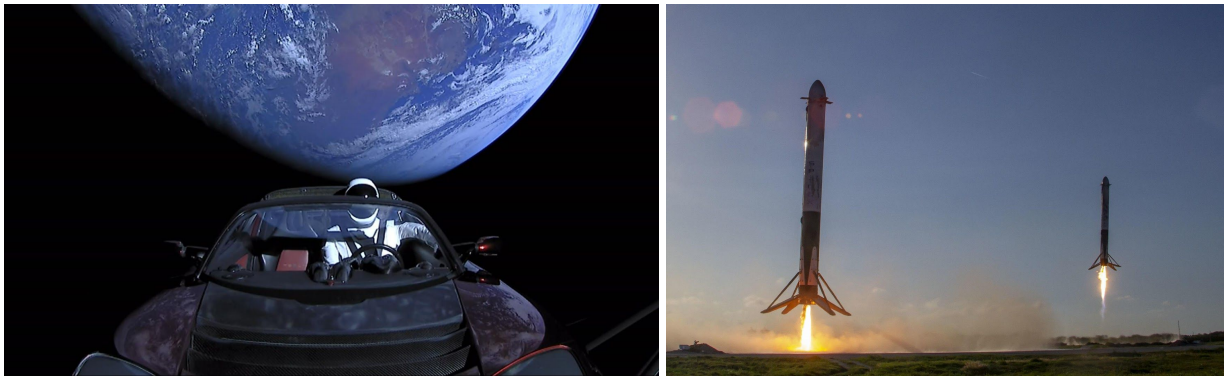

Fig. 1 (a) Tesla roadster with the spaceman launched by SpaceX's Falcon Heavy on Feb. 06, 2018 (b) two side boosters of SpaceX's Falcon heavy landing regulated by the entry burn

One norm that warrants some thoughts and motivates this study is the design and fabrication of amateur rockets by rocket enthusiasts. The complexity of amateur rockets ranges from pumping air into a big inverted coke bottle filled with water (known as a water rocket) that barely reaches the roof to sophisticated rockets using liquid or solid fuels capable of climbing a few thousand feet. Regardless of the complexity, the majority of the amateur rockets have a few characteristics in common. First, they often

use out-of-the-box materials (coke bottles) or hand-made ones (such as hammered and soldered frames and glued-together parts). Second, the amount of control is limited as the primary attention is given to improving thrust performance. Third, the level of autonomy is low, arguably because few tasks are expected for the amateur rockets in the first place due to the lack sufficient control. Forth, the rocket motors (thrust provider, the real rocket engine equivalent) used usually feature high thrust in the beginning, which declines quickly in the end, thus leaving limited potential for final-stage maneuvers.

The norms in the design and fabrication of amateur rocket can be readily challenged, and even possibly updated by the technology injection explored in this work, namely 3D printing, thrust vector control and onboard flight computer. Three-dimensional (3D) printing allows fast fabrication of custom-made parts that succinctly implement a design. For example, 3D printing produces structurally-critical parts only, eliminating redundant components and reducing weight. Free from soldering and gluing, 3D-printed parts also have less weak points while preserving structural integrity. Thrust vector control (TVC) allows the rocket motor to point in an arbitrary direction within a spherical cone, as opposed to the vertical direction only. The onboard flight computer processes data from the gyroscope (roughly speaking, a device that measures or maintains the flight orientation/status), adjusts the thrust vector control and the rocket motor accordingly and completes the control loop. The grand combination of the technologies delivers a custom-made amateur rocket that can self-stabilize at any given altitude with real-time control, with flexible mission payload and target. Infused with "intelligence", such a smart amateur rocket can be extremely useful if climate data or photos at a specific height are to be taken, or catalysts for artificial precipitation need to be deployed at designated location in the clouds for maximum efficacy. Our prototype and tests, as detailed in this technical report, provide strong support that technology injection has great potential in opening up many opportunities that have barely been dreamed of.

The technical report is organized as follows. In Section 2, we present a brief overview of the design and fabrication of traditional amateur rockets, and highlight where new technologies lead to improvement. Section 3 gives detailed descriptions of the new technologies chosen, namely 3D printing, thrust vector control and onboard flight computer. In particular, we provide technical details to explain the working principles of each technology, and comprehensively document design considerations in the evolution of each design generation. In Section 4, we discuss how the prototype attempted and tested in this work can benefit the field of amateur rocket in general.

2 Background of Amateur Rockets

Amateur rockets realize the layman's desire into space, and the basic idea is simple: fill up the rocket body with fuel, hit the sky and get recovered by a parachute. Amateur rockets are built just to

replicate and salute their famous predecessors, such as Saturn V that powers the Apollo Project, while others are engineered to perform specific tasks or just to aim high. Even though procedures of fabricating amateur rockets have been established and practiced, emerging technologies explored in this work could lead to a big step forward. 3D printing widens the producibility of rocket parts previously hindered by craftsmanship, money and/or time, delivering lightweight and yet sturdy parts overnight so that the design iterates and evolves fast. Thrust vector control (TVC) works in conjunction with the specially designed rocket motor, both regulated by the flight computer to achieve self-stabilization, altitude control and autonomous landing. As a result, parachute used for rocket recovery is no longer necessary, and the amateur rocket can hit the sky with more payload and better control. The autonomous landing also calls for re-engineered rocket fuels that can give more sustained thrust.

2.1 Anatomy of Traditional Amateur Rockets

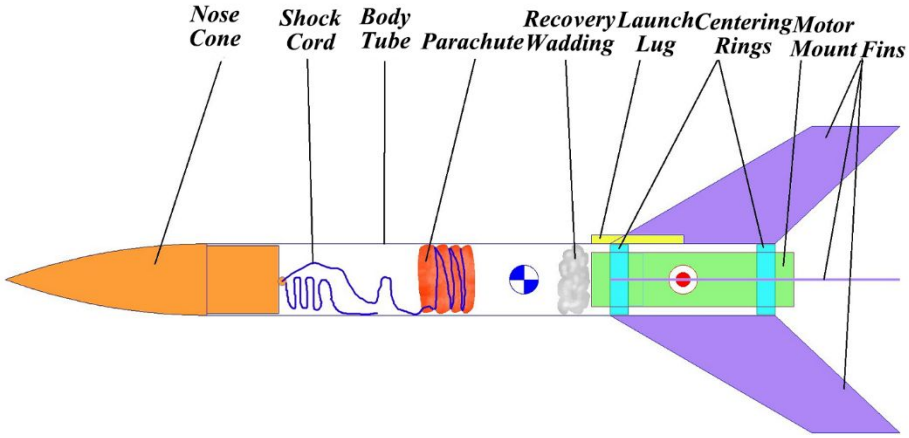

Fig. 2.1: anatomy of an amateur rocket

A traditional amateur rocket is no more than a missile in early war era, but with the warhead replaced by a parachute. Fig. 2.1 shows the anatomy of a typical amateur rocket. The exterior consists of a pointed nose cone which reduces aerodynamic drag, a body tube usually made of light materials that encapsulates payloads and on-board equipment, and fins which stabilize the rocket in the mid air and avoid excessive rolling. A launch lug is also common, which helps maintain a vertical posture on the launch pad. The interior contains the recovery system and the rocket motor, in addition to the payload. The recovery system is composed of a parachute connected to the nose cone by a shock cord, which can withstand huge extensive straining force. As the rocket completes its mission, the nose cone jettisons, pulling the shock cord and releasing the parachute. The successful deployment of a parachute is key to the recovery of on-board equipment. The rocket motor is installed at the bottom of the rocket, and its installation can be

adjusted by centering rings to align the thrust in the vertical direction. To prevent excessive heat from melting the payload and the parachute, some recovery wadding can be added. Amateur rockets as shown here usually only have one goal - fly high [5].

2.2 Opportunities for New Technology Injection

2.2.1 3D printing

3D printing has become a relatively mature technology nowadays and has been widely applied in various fields, from fabricating parts to printing a house [2]. The fact that 3D printing could quickly create models with high accuracy is favored by many engineers. As an example, traditionally model building often involves sculpture making, basically cutting the model from a large clay material (Fig. 2.2 (a)), which demands a high level of craftsmanship and is timing-consuming. In comparison, 3D printing (Fig. 2.2 (b)) produces accurate models, having no bearing on the designer's craftsmanship.

Common types of 3D printer include Fused Deposition Modeling (FDM), Stereolithography (SLA) and Digital Light Processing (DLP), etc. FDM is the most popular type given its good overall metrics of printing speed and accuracy, structural strength, and satisfactory user support. As a result it is chosen for this project. With 3D printing, model building starts with a computer-aided design (CAD) model, which is then "sliced" into layers by a slicing software. The sliced model can be readily interpreted by 3D printer so that it completes printing layer by layer. In addition to the custom-designed shape, precise weight control of the printed parts is possible by using materials with different fill-in ratios (Fig. 2.2 (c)). The best model is likely to result from one that uses intermediate fill-in ratio (i.e. a medium amount of cavities), which achieves a perfect balance between strength and weight, as both are key factors in amateur rocket design.

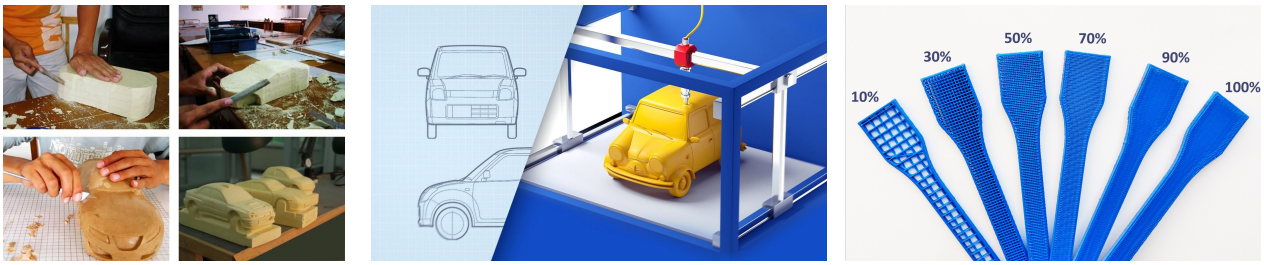

Fig. 2.2 (a) traditional model making (b) 3D printing (c) materials fed into 3D printer, the labels show different fill-in ratios.

2.2.2 Thrust Vector Control (TVC)

TVC is a system that allows the thrust to be directed in any direction within a spherical cone. It is first used in the engine nozzles of performant fighter jets (Fig. 2.3 (b)), and has also been adopted in the rocket

engines by SpaceX (Fig. 2.3 (a)). The working principle is fairly straight-forward (Fig. 2.3 (c)): by adjusting the angle of the thrust, a torque about the center of gravity is produced and can be used to correct any inclination of the rocket body. Inclinations of the rocket are not uncommon as the rocket body is subject to forces from wind gusts during ascending. Traditionally, amateur rocket is deliberately made to undergo fast spinning upon launch to enhance the vertical stability [6], but such a strategy may fail the mission purpose, such as high altitude photo-taking, and definitely makes autonomous landing impossible. TVC comes as a perfect solution: it ensures that the rocket follows the intended climbing trajectory, and stabilizes the rocket during landing. Sophisticated TVC system obsoletes parachutes and the side fins, thus saving extra weight and room for the payload.

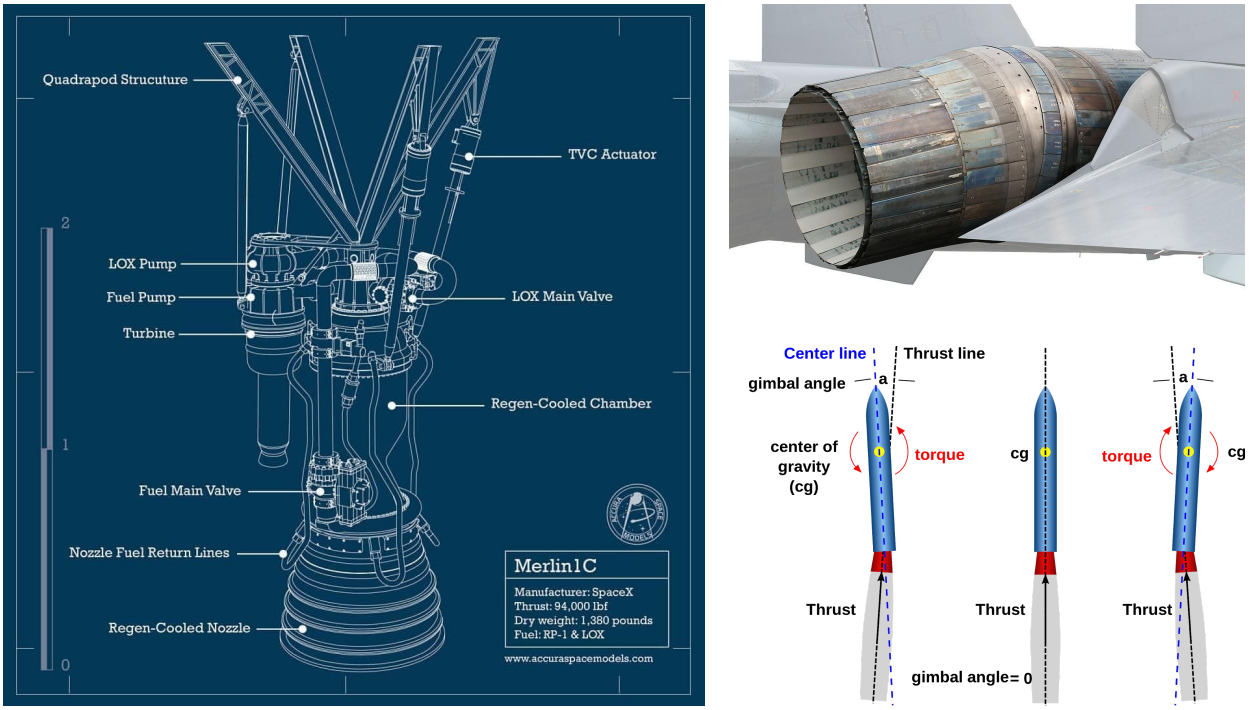

Fig. 2.3: (a) thrust vector control (TVC) of the Merlin 1C rocket engine used by SpaceX [7] (b) thrust vectoring nozzle on a Russian Sukhoi Su-35 (c) illustration of how TVC system corrects inclination and maintains vertical status

2.2.3 flight computer and rocket motor

As the name suggests, flight computer serves as the central control and flies with the rocket. In our case, it brings "intelligence" to an amateur rocket and enables self-stabilization and landing. Key components of a flight computer include the brain (the central processor), the sensor (a gyroscope that obtains information on acceleration, and thus altitude) as well as interconnects (pins that connect the processor, the gyroscope, the servo motors and the battery. Following instructions of the code stored in the processor, the flight

computer reads data from the gyroscope, processes it with the central processor, and then sends commands to the servo motors to maintain the position and orientation of the rocket. Other extendable components can also be added, such as LEDs and buzzers for displays and notifications, SD cards for flight data storage, cameras for visual recordings, and bluetooth or Wi-Fi equipment that establishes connections with the phone, or scientific instruments. Compared to traditional amateur rockets, flight computer is a true game-changer: empowering amateur rocket with the ability to self-control, record and communicate remotely, flight computer revolutionizes amateur rockets to stand out in the bland game of "see who flies the fastest and highest".

The rocket motor provides the indispensable thrust so that the rocket can lift off in the first place. Thrust is produced by burning rocket fuel, and different fuels give distinct performance. Fuels used in amateur rockets usually dump enormous thrust which lasts for a certain period of time and then declines rapidly. Plotting thrust as a function of time, these typical fuels render the "thrust curve" as having a sustained peak thrust at early times, but sudden and sharp decrease in magnitude at final stages. However to ensure the maneuverability of "smart" amateur rockets during landing, it is critical for the thrust curve to flatten out (i.e. more steady burning), reserving enough thrust for late times. The technical details of forming the specific composition of the actual fuel are postponed to Section 3.4.

3 Descriptions of the Implementations of New Technologies

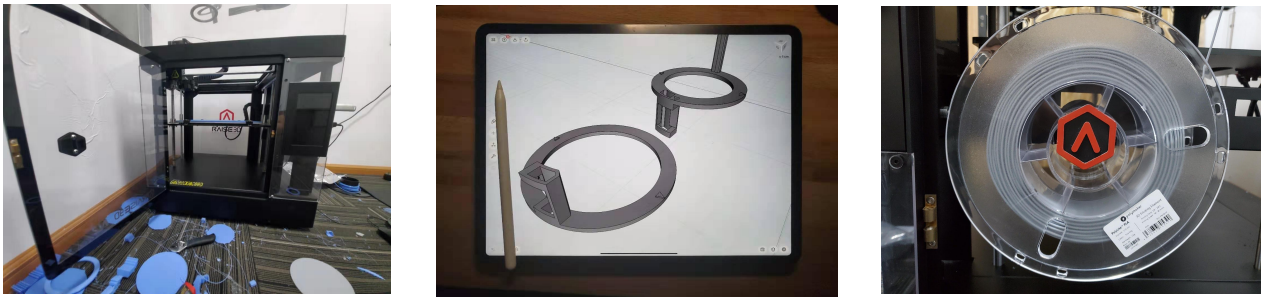

Fig. 3.1 (a) the Raise3D printer (b) the Shapr3D design software (c) polylactic acid (PLA) filament

Rome wasn't built in one day, nor is an amateur rocket, especially when multiple new technologies are injected. For each of the new technologies, several iterations have been performed. In this section, we document the implementation details, including procedures, rationales and challenges.

3.1 3D printing

3.1.1 introduction

The 3D printer used in this work is a powerful Raise3D (Fig. 3.1 (a)), which features a $12 \times 12 \times 12$ inch build platform, and a minimum of 0.01 mm layer height. The build platform has ample room that can cater a single large piece, without the need to disassemble the digital model into multiple parts for separate printing. The small layer height gives a fine finish, saving the extra work of polishing. Shapr3D (Fig. 3.1 (b)) is used as the 3D design software, as it works perfectly with the latest iPad Pro 11" where the use of Apple pencil greatly facilitates the component design to a whole different level. High-quality polylactic acid (PLA) filament (Fig. 3.1 (c)) is chosen as the printing material after much trial and error. PLA strikes a good balance of structural strength, weight and printing speed. Furthermore, high-quality PLA melts smoothly and avoids material jams at the 3D printer nozzle.

3.1.2 procedures

3D printing starts from model building, which nowadays relies heavily on computer-aided design (CAD) tools. Fig. 3.2 (a) shows a screenshot of the Shapr3D CAD software where the rocket motor mount (shown in blue) is virtually assembled with the outer rings/gimbals (shown in gray). Even though a CAD model can be designed in any way that imagination permits, practical concerns on the model's "printability" play an important role in shaping the actual design. For example, any dimension of the model cannot be smaller than the finest layer height that the 3D printer can handle. Also, certain holes can be better obtained from post-processing of drilling, rather than printing. Once a "printable" model is ready, it is important to separate the full assembly and print each part individually, because this reduces the risk of parts interfering with each other during printing.

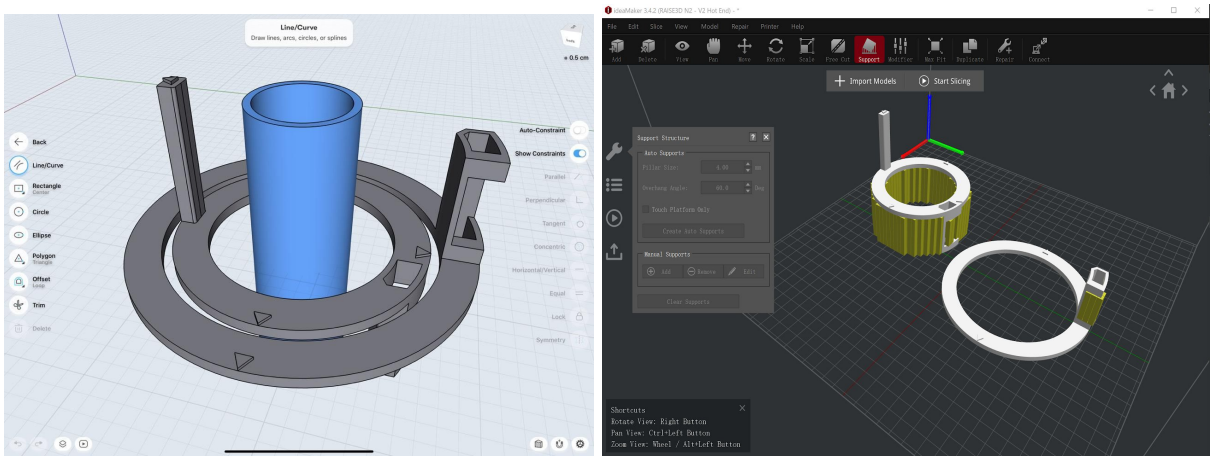

Fig. 3.2 screenshots of (a) the Shapr3D design software (b) the ideaMaker slicing software

Since 3D printer does not understand the CAD model, a slicing software is needed to translate the CAD model to something comprehensible by the 3D printer. The slicing software analyzes the CAD model, breaks it into layers and sets up the route for the 3D printer nozzle to traverse. Specifically, CAD model is first saved in a universal file type (STL format) for the best compatibility. Then the STL file is transferred to the slicing software, which usually bandles with the 3D printer. For Raise3D, ideaMaker serves as the slicing software (Fig. 3.2 (b)). In the slicing software, first scale the model to the right size and double check the units. Then add auto-supports (seen as yellow parts in Fig. 3.2 (b)) so that the model does not crush during printing. Finally set up printing specifics and wrap up all settings into a .gcode file, which can be readily read by the 3D printer. For this project, the optimal settings have the layer height as 0.1 mm, shell thickness as 2.0mm, fill-in density (of the PLA filament) as 15% and the fill-in speed as 60 mm/s. These settings ensure good structural integrity, lightweight and a smooth finish for parts that can be printed/delivered overnight.

The final step is just to launch the 3D printing. The nozzle head needs to be well calibrated to the origin of the printing panel for consistent product quality. Also make sure the 3D printer warms up to the specified temperature that the printing material requires (for PLA filament, it is 220 degree Celsius). A small trick from the hands-on experience is to apply a layer of tape on the printing panel to increase the surface roughness, because it facilitates easy scraping of the printed model from the panel. In the final product inspection, use a file or sandpaper to polish the model if needed.

3.1.3 performance comparisons and discussions

Compared to the "sculpture style" model-making shown in Fig. 2.2 (a), 3D printing has multiple advantages, as it not only allows extremely fast iteration, but also is more cost-effective and environmentally-friendly. For example, modifying a design feature only takes minutes in Shapr3D while traditional models entail cutting, gluing and refining, which are all time-consuming. Moreover, leeway for major changes is extremely limited for traditional model-making. As for the economy perspective, the 3D printed parts shown in Fig. 3.2 cost about $4 of the PLA filament while the sculpture style can easily exceed $20 of material cost, not even accounting for the cost of human labor. Since traditional model-making tends to produce much waste due to cutting and refining, 3D printing can help reduce material waste and help conserve the environment.

A few suggestions on 3D printing are in place which may benefit future enthusiasts. First, the treatment of holes in parts deserve extra care. In this project, holes are needed for metal rod connection and servo motor fixation, and yet the hole diameter of 1.2mm is too small to use auto-support but too big that the model can crush during printing. In scenarios like this, hole drilling is more viable. Second, high quality printing material is critical to avoiding the nozzle jam. At one time nozzle jam constantly occur, and it

took a while to exclude possibilities of nozzle head malfunction or temperature setting, and finally pinpoint the exact cause of the issue - quality of printing materials.

3.2 Thrust vector control (TVC)

3.2.1 introduction

To facilitate the discussions, Table 1 below gives definitions of key components of the thrust vector control system, accompanied by a visual illustration in Fig. 3.3.

name	definition
Outer X-gimbal	A ring-like structure that holds the x-axis servo motor. It connects the inner Y-gimbal's rod and controls its rotation about the x-axis.
Inner Y-gimbal	A ring-like structure that holds the y-axis servo motor. It connects the motor holder and controls its rotation about the y-axis.
Motor holder	A circular tube where motor of cylindrical shape can be installed.
Thrust vector control (TVC)	A combination of the above, plus the servo motors on each gimbal. TVC can rotate within a 5 degree spherical cone.

Table 1: definitions of key components of the thrust vector control system

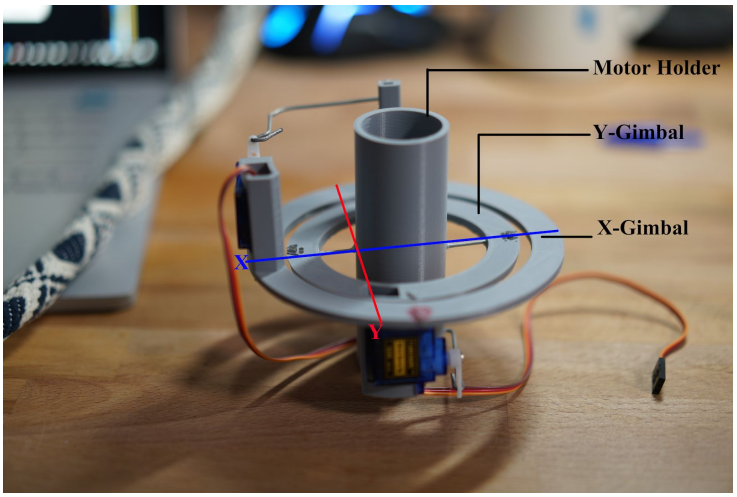

Fig. 3.3 illustration of key components of the thrust vector control system

3.2.2 self-stabilization algorithm

All the delicate hardware needs to be regulated by an algorithm that implements self-stabilization, for orientation and position adjustment during both ascending and landing. A mature language for microcontrollers is Arduino [8], and we use the Arduino 1.8.9 platform.

Fig. 3.4 gives an outline of the algorithm. For the set-up, the gyroscope provides 6 data of the acceleration and altitude in the x, y, z directions and feeds them into the Arduino central processor. Two pins of the Arduino microcontroller are dedicated to control the servo motor on the outer X-gimbal and inner Y-gimbal respectively. In the main loop, the algorithm keeps track of the rotation of each servo motor, and should the rotation values change, the algorithm updates the rotation value (stored in the servo motors) with data freshly obtained from the gyroscope. In this way, the rocket motor can always be directed in the direction that helps restore a vertical orientation. The code snippet is pasted in the appendix, and the whole code is also available on my blog and Github [15].

```
Pseudo code:

In the main loop:

    Call the mpu to get the acceleration and attitude data

    For outer: set the range of acceleration from -17000 to 17000, and attitude from 0 degrees
    to 179

    For inner: set the range of acceleration from -17000 to 17000, and attitude from 179
    degrees to 0

    If the previous value does not equal to the current value, write current value
```

Fig. 3.4 an outline of the main loop in the self-stabilizing algorithm

3.2.3 demonstration

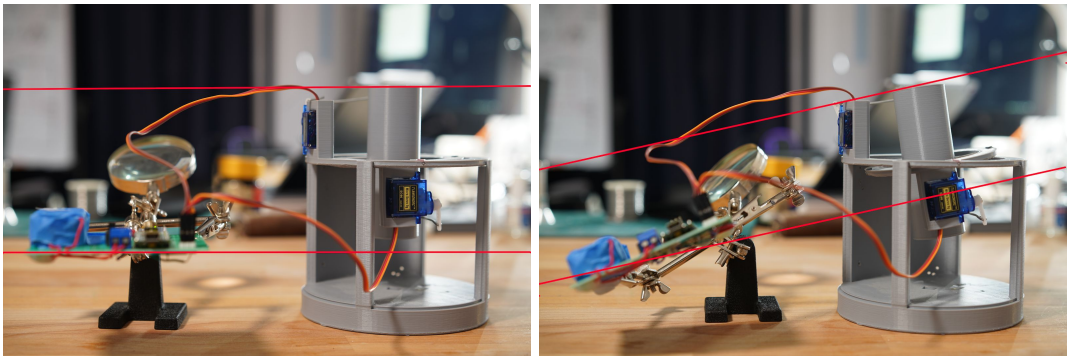

Figure 3.5 (a) default position where both the gyroscope (in blue) and the motor holder are vertical (b) motor holder inclines in accordance with the gyroscope

In this subsection, we demonstrate how the self-stabilizing mechanism works. In actual configurations, the gyroscope and the central processor (Arduino microcontroller) are attached with the Thrust Vector Control (TVC) system. However, for demonstration purposes, both components are taken detached from the TVC, as shown in Fig. 3.5. Once the gyroscope is inclined, the motor holder is instantly rotated by the sermo motors by the same rotational angle, as marked by the two parallel red lines.

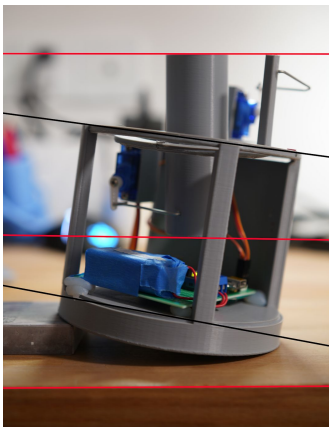

Figure 3.6 motor holder keeps vertical while the test platform inclines

Fig. 3.6 shows the TVC system in its actual usage state where the gyroscope and central processor are installed with the TVC system. It demonstrates again that the motor holder always points vertical to the ground regardless of the orientation of the rocket body. In the real atmosphere, wind gusts create disturbances that can tilt the rocket body, and the TVC system counteracts the disturbances and corrects the rocket trajectory, regardless of ascending or landing.

3.2.4 design iterations and discussions

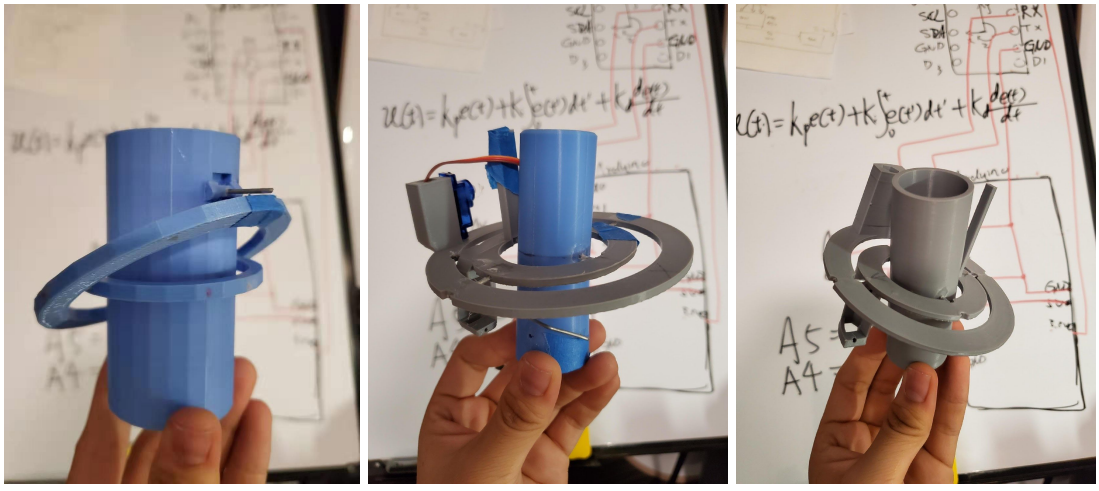

Figure 3.7 the three generations of TVC systems in chronological order from left to right

The TVC system in its current form actually undergoes three generations of design iteration, as shown in Fig. 3.7. The first generation successfully serves as a concept proof, but has various design issues. For example, limited installation space was reserved for servo motors and the rod connections, and the weight was too high. The second generation solves partial problems by using lighter materials with smaller thickness (in gray) and leaving more space for servo motors. Furthermore, it separates the rigid connection between the inner Y-gimbal and the motor holder, making the motions in the x and y directions truly independent. While the third generation replaces the motor holder (in blue) with lighter materials, there was still dangling issues that inner and outer gimbals lack sufficient clearance to avoid any clashes. Eventually the design is further optimized, and the final layout is shown in Fig. 3.6.

3.3 The flight computer

3.3.1 introduction

Generally speaking, the flight computer is an electronic component of amateur rocket that processes signals from various sensors and issues commands to maintain the correct position and orientation of the rocket. Since one essence of the project is the thrust vector control (TVC), we focus on designing and building a flight computer that enables active TVC.

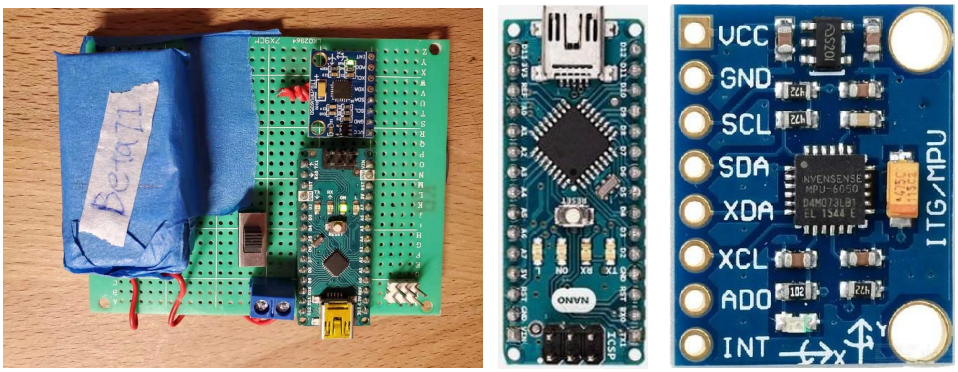

Fig. 3.8 (a) the full look of the flight computer (b) zoomed-in view of the Arduino Nano microcontroller (c) zoomed-in view of the gyroscope MPU6050

3.3.2 construction

Fig. 3.8 (a) gives a full look of the flight computer in its most current state, which houses the microcontroller, gyroscope, battery and a power switch. The brain of the flight computer is the Arduino Nano microcontroller (Fig. 3.8 (b)), which is a more compact version of its predecessor of Arduino Uno. The gyroscope measures acceleration and calculates the altitude, and it acts as a key sensor to help maintain the posture of the rocket. The model MPU 6050 (Fig. 3.8 (c)) used is precise and light-weight, and receives good support from a large user community. The power supply is a 9 Volt lithium battery

rated at 650mAh. Equipped with a micro USB charging port, the lightweight reusable battery can power the entire flight computer for a sustained period of time.

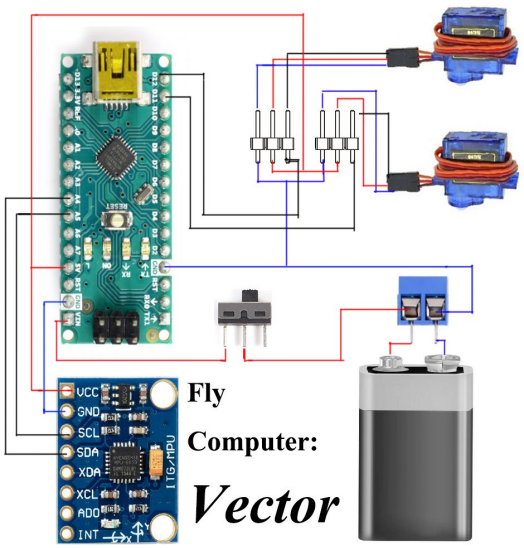

Fig. 3.9 the circuit schematic diagram of the flight computer dubbed “Vector”

To wire together different components, the following procedures can be followed. Take Fig. 3.9 as the reference, where red and blue colors denote positive and negative wires respectively. To power the Arduino Nano, connect the battery to the KF301-2P Blue Screw Terminal, and install a power switch on the red wire. Then ground (GND) the negative wire and connect the positive wire to VIN to complete the power set-up. To connect the MPU6050 (gyroscope), connect the 5 Volt output pin to VCC on the MPU6050 and GND to GND. To establish the data connection between Arduino Nano and MPI6050, connect the SCL (A5) and SDA (A4) ports from Arduino Nano to the corresponding ports on MPU6050. The connections entail some soldering (Fig. 3.10 (a)), and the correctness of the connection needs to be confirmed through testing (Fig. 3.10 (b)).

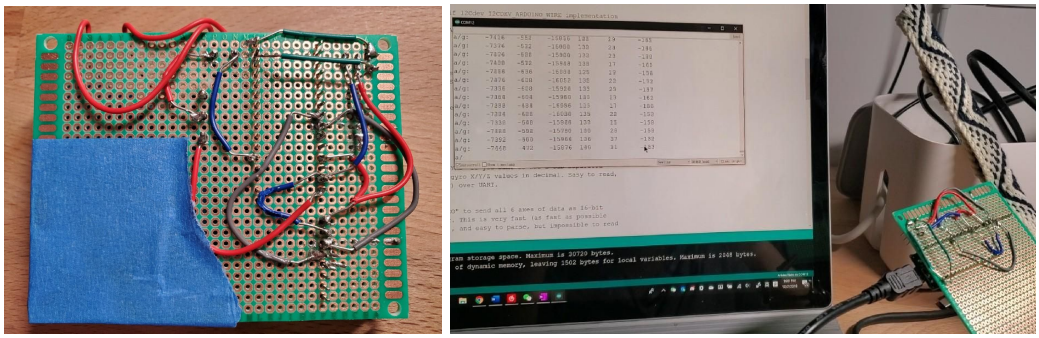

Fig. 3.10 (a) the soldering on the back of the flight computer (b) testing gyroscope data reading

3.3.3 design iterations and discussions

Fig. 3.11 shows the three generations of flight computer. The first two generations use a PWM control board that houses the bigger Arduino Uno microcontroller, and as a result much more space was occupied (to be compared against Fig. 3.8 (a)). In addition to the issue of excessive size, technical issues lingered for the first generation. Most notably, the Arduino Uno microcontroller did not allow data uploading from the computer while its certain pins were plugged. Such a nuisance prevents efficient testing, and motivates the switch to Arduino Nano in the current version (right-most of Fig. 3.11).

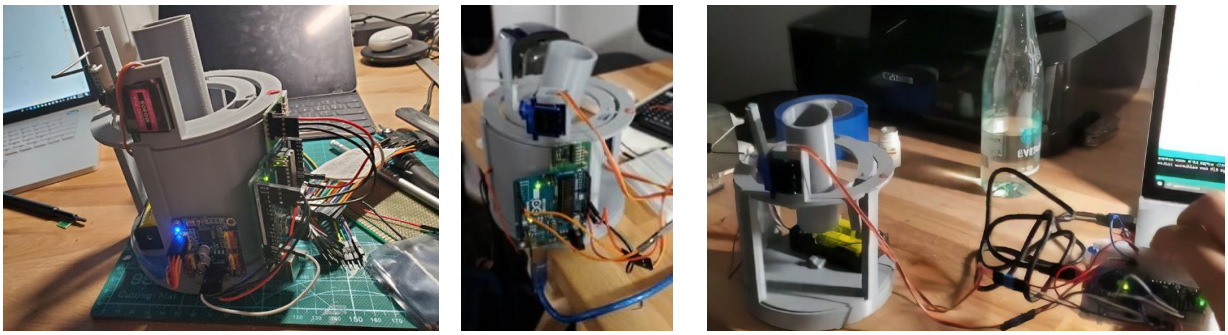

Fig. 3.11 the three generations of the flight computer in chronological order from left to right

A key upgrade in the second generation (middle of Fig. 3.11) was the removal of the PWM control board, because experiments showed that Arduino could provide enough voltage to power the servo motors. Yet it still used an older version of the gyroscope which had compatible glitches. The third generation is the current version, which is strikingly more compact. Furthermore, to allow easy plug in and out of the connection between the servo motors and the flight computer, the current flight computer uses two 3-pin headers, instead of soldering the connection. As a result, small components of the flight computer are modularized, making maintenance and future upgrades simple.

3.4 The Rocket Motor

3.4.1 introduction

The rocket motor provides thrust to the entire rocket by burning rocket fuel/propellant. Opposed to fixing the rocket motor to the rocket body in traditional amateur rocket, in this project the motor is installed inside the motor holder, and is controlled by the thrust vector control (TVC) system. As noted previously, TVC system requires a relatively long burn time and sustained thrust output in order to self-stabilize the rocket during ascending and landing. Because buying rocket motor is forbidden in mainland China, as a last resort I have to build the rocket motor and “brew” the specific rocket fuel, all by myself.

3.4.2 background and theoretical calculations

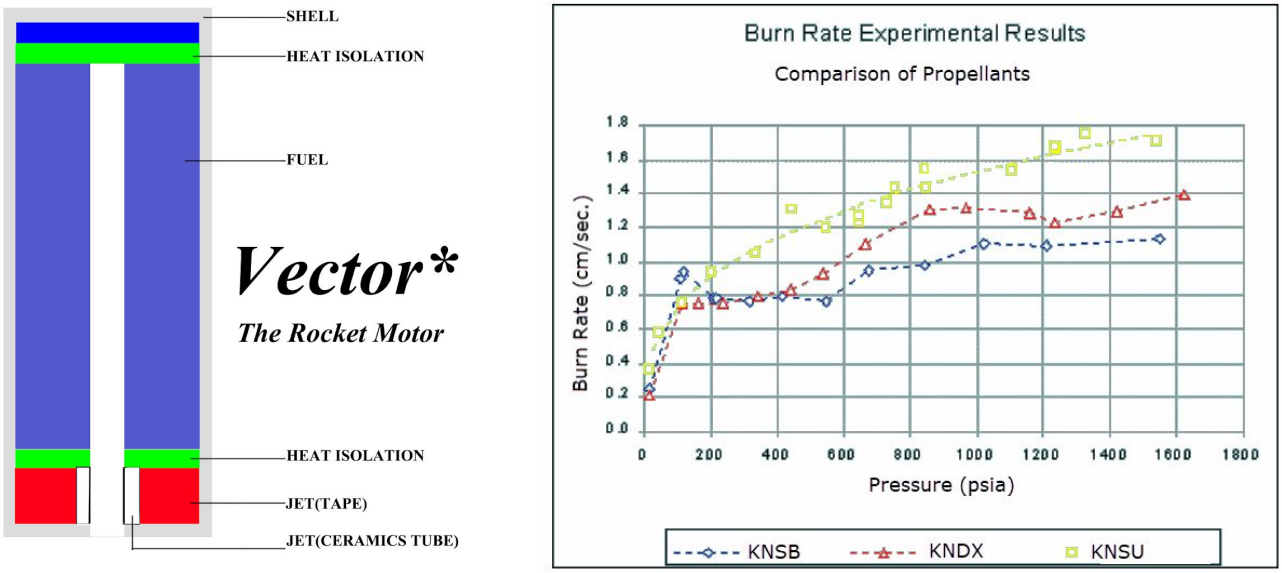

Fig. 3.12 (a) anatomy of the rocket motor (b) experimental results of burn rate for different fuels [9]

Motor shell (grey):	PPR tube with an outer diameter of 25mm and 3.5mm thickness
Jet (white + red):	ceramics tube with a diameter of 3mm, strong glue and paper are also used
Fuel (purple):	KNSB (60% KNO3 + 40% C6H14O6)
Heat isolation (green):	tinfoil
Protection (blue):	paper and strong glue

Table 2: descriptions of each component of the rocket motor anatomy

Fig. 3.12 (a) shows an anatomy of the rocket motor, also dubbed “Vector” in accordance to the flight computer. The shell encapsulates fuels insulated by heat wadding, and special attention is given to the jet exhaust (nozzle), which is made of ceramics, as it needs to withstand high temperature. More details of the components are given in Table 2.

The fuel choice bears multiple considerations. Fig. 3.12 (b) shows the burn rate of three popular fuels at different pressures. Roughly speaking, a low burn rate corresponds to less thrust as the fuel is burnt slower. All three fuels are mixtures of Potassium Nitrate and some type of “sugar”. Since access to the rocket fuel is none in mainland China, a major consideration is how easy to compose the fuel formulation. From hands-on experiments, it is found that KNSU (Potassium Nitrate + Sucrose) is prone to caramelization during recrystallization, yielding poor quality product. KNDX (Potassium Nitrate + Dextrose) turns out to demand a delicate set of production techniques, making it difficult to obtain

products with consistent quality. In comparison, KNSB (Potassium Nitrate + Sorbitol) has the best chemical properties at high temperatures: it is hard to get caramelized upon heating, but meanwhile mixes easily with Potassium Nitrate. Furthermore, casting cooled KNSB powders into grain (i.e. a cylindrical solid bar to be filled inside the rocket motor) is easy. Given all these superior advantages, KNSB is chosen as the fuel for this project, even though it gives the least thrust among the three. Some pictures of the actual fuel burning during tests are shown in Fig. 3.13.

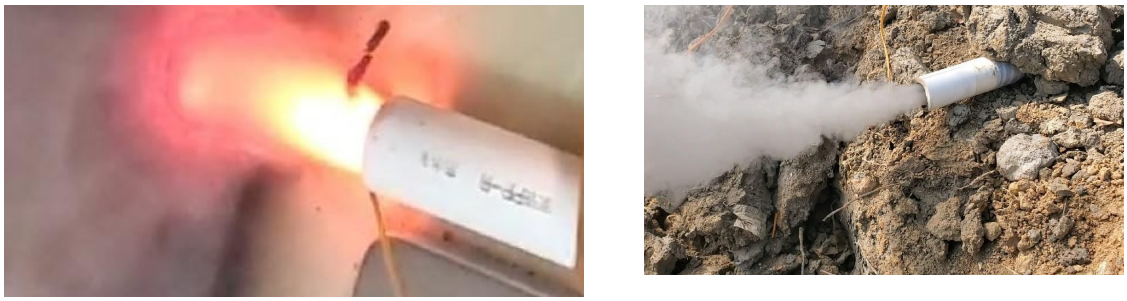

Fig. 3.13 snapshot of the burning of (a) KNSU (b) KNSB

Some calculations are also performed to estimate the thrust force needed to lift up the rocket. The maximum thrust F is given by the equation below [10]:

$$F = A^* P_o \sqrt{\frac{2k^2}{k-1}\left(\frac{2}{k+1}\right)^{\frac{k+1}{k-1}}\left[1-\left(\frac{P_e}{P_o}\right)^{\frac{k-1}{k}}\right]} + (P_e - P_a)A_e$$

where P_o is the stagnation pressure, P_e is the exit pressure, A^* is the throat cross-section area and k is the ratio of specific heats. For a model rocket, the last term can be neglected and we are left with:

$$F = A^* P_o \sqrt{\frac{2k^2}{k-1}\left(\frac{2}{k+1}\right)^{\frac{k+1}{k-1}}\left[1-\left(\frac{P_e}{P_o}\right)^{\frac{k-1}{k}}\right]}$$

From the online literature [11], KNSB has $k = 1.04$ and we estimate $P_o = 20\ atm$ and $P_e = 1.5 atm$. Also plug in the jet exhaust diameter of $3\ mm$, we finally arrive at:

$$F = 20N$$

From the burning tests, KNSB gives a surprisingly long 25s thrust time.

3.4.3 construction

The first step to construct the rocket motor is to prepare the shell, by cutting the PPR tube to the correct size (Fig. 3.14 (a)). It has been found advantageous to heat up the tube by a hot air gun to about 300 degree Celsius to make the cutting easier as the material softens. The bottom of the cut tube is to be filled by the nozzle. First prepare some ceramic tubes (Fig. 3.14 (b)), and then wrap around it with inflammable

tapes until the diameter fits the PPR tube (Fig. 3.14 (c)). Apply glue around the tape and fit the structure to the PPR tube.

Fig 3.14 (a) PPR tube (b) ceramics tubes (c) a nozzle made of ceramic core, tape and glue

Fig. 3.15 (a) the making of the rocket fuel (b) the completed rocket motor (c) remote firing control

Once the PPR tube has the bottom fixed with a nozzle, the rocket fuel can be injected inside the tube. To make the KNSB fuel [12], first grind Potassium Nitrate and get Sorbitol ready. Heat the pan to 230 degree Celsius and melt the Sorbitol first. As Sorbitol is completely melted, add Potassium Nitrate and mix them well with a wooden rod. Upon complete mixing, pour the fuel mixture into the tube (with the nozzle temporarily blocked). As the mixture solidifies, compress the fuel with rod to increase energy density. Finally dig out a cylindrical space with the same diameter as the nozzle to leave enough burning surface. Seal the top of the tube with glue and inflammable paper and the rocket motor is constructed (Fig. 3.15(b)). The rocket motor is to be mounted inside the motor rotor holder shown in the TVC system. Since the chemicals are explosive and pose a potential hazard, testing of the rocket motor requires much

caution. A firing system capable of remote control is also developed (Fig. 3.15 (c)) so that testers can stay clear the test zone.

4 Conclusions and recommendations

In this project, we present an exploratory study of adding new technologies in the design and fabrication of amateur rockets, and demonstrate that capabilities of traditional amateur rockets can be greatly extended with the new technology injection. In particular, four pieces of technologies have been investigated, using a combination of theoretical and experimental methods. 3D printing is shown to produce lightweight, custom-made parts that yield slim designs in a cost-effective and time-saving fashion. We used 3D printing technology to fabricate parts constituting the thrust vector control (TVC) system. By orienting the thrust to a direction that restores the posture of the amateur rocket, TVC system provides an alternative to stability control in the ascending phase compared to traditional amateur rocket fins. Moreover, TVC system acts as the sole mechanism that allows autonomous landing, as implemented by SpaceX based on similar principles. In order for the TVC system to work autonomously, the flight computer serves as the brain, processing signals from gyroscope sensors and instructing the TVC system to respond accordingly. New requirement of landing calls for a rocket fuel that delivers smoother and persistent thrust, especially at the later stages of the mission. Yet due to restricted access to out-of-the-factory rocket fuel products in mainland China, manual formulation of the rocket fuel has been attempted.

The “rocket science”, despite in an amateur sense, is still challenging for an independent researcher like me. It is useful to share reflections and recommendations for future enthusiasts, because throughout this project I personally have benefited much from knowledge shared from various individuals online and in person. After all, it is through countless iterations of passing the knowledge to the posterity that technology keeps evolving and contributes to humanity. The first reflection is to never underestimate the difficulties of working with new technologies, and a team can help the project go farther and faster. For example, instead of sequentially fixing the nozzle jam in 3D printers, and then debugging/tuning the TVC system, a small team of enthusiasts can parallelize independent tasks, and brainstorm novel solutions to tricky problems. Second, a more systematic approach can be taken in the design phase. For example, virtual motions of the various parts in the TVC system can be simulated beforehand using software such as Matlab Simulink. Also more sophisticated analysis can be performed to the rocket structures and aerodynamic shapes, using software in Finite-Element Analysis [13] and Computational Fluid Dynamics [14] respectively. Finally, as amateur rocket is still a potentially dangerous subject, always be safe and make sure laws and rules are strictly followed. For example, further testing of the rocket fuels has to be

aborted as police officers tracked my chemical orders and held interviews with me. Similarly, in the United States, launching of amateur rocket requires due process applications and approvals too. It is important to realize that personal enthusiasm and interest should never precede public safety.

In summary, through an exploratory study of new technologies including 3D printing, thrust vector control, onboard flight computer and re-engineered rocket motor, I am convinced that technology injection greatly expands the capability boundaries of traditional amateur rockets. It is expected that norms adopted in the design and fabrication of DIY rocket will be challenged and updated, and the smarter and more capable amateur rocket will become the bright future.

5 Acknowledgments

The generous financial assistance from my parents is acknowledged, and I also thank them for their support and patience throughout this project. I am grateful for the discussions with Xu YUAN and Ye CHENG, whose comments and suggestions proved very helpful. Thanks also go to the local police officers as they pardoned my rocket fuel tests, which could otherwise result in a one-week detention if I were an adult. Last but not the least, I appreciate the kind and generous knowledge sharing from everyone in the online amateur rocket community.

6 References

[1] Sutton, George P., and Oscar Biblarz. *Rocket propulsion elements*. John Wiley & Sons, 2016.

[2] Lipson, Hod, and Melba Kurman. *Fabricated: The new world of 3D printing*. John Wiley & Sons, 2013.

[3] Blackmore, Lars. "Autonomous precision landing of space rockets." *Frontiers of Engineering: Reports on Leading-Edge Engineering from the 2016 Symposium*. National Academies Press, 2017.

[4] https://en.wikipedia.org/wiki/List_of_private_spaceflight_companies

[5] https://www.nar.org/high-power-rocketry-info/high-power-resources/high-power-technical-reports/

[6] Guerrero, Valeria Avila, Angel Barranco, and Daniel Conde. "*Active Control Stabilization of High Power Rocket.*" (2018).

[7] Bjelde, Brian, Peter Capozzoli, and Gwynne Shotwell. "The SpaceX Falcon 1 Launch Vehicle Flight 3 Results, Future Developments, and Falcon 9 Evolution." *Space Exploration Technologies* (2008).

[8] McRoberts, Michael. *Beginning Arduino*. Apress, 2013.

[9] https://www.nakka-rocketry.net/sorb.html

[10] http://www.nakka-rocketry.net/articles/thr_example5.pdf

[11] http://www.nakka-rocketry.net/techs2.html

[12] http://www.doranaerospace.com/Sorbitol.html

[13] Szabó, Barna, and Ivo Babuška. *Finite element analysis*. John Wiley & Sons, 1991.

[14] Anderson, John David, and J. Wendt. *Computational fluid dynamics*. Vol. 206. New York: McGraw-Hill, 1995.

[15] https://github.com/Deemocean/TVC/

Appendix: code for thrust vector control

The code for thrust vector control is pasted below.

```
#include "Wire.h"       // allows communication to i2c devices connected to arduino
#include "I2Cdev.h"     // I2Connection library (communication to serial port)
#include "MPU6050.h"    // IMU library
#include "Servo.h"      // servo control library

MPU6050 mpu; //define the chip as an MPU so that it can be called in the future

int16_t ax, ay, az;  // x y z orientation values from accelerometer
int16_t gx, gy, gz;  // x y z orientation values from gyroscope
////////////////////////////////////////////////////////////////////////////////////

Servo outer;
Servo inner;
////////////////////////////////////////////////////////////////////////////////////

int valo;     // outer val
int prevValo; // outer prev val
////////////////////////////////////////////////////////////////////////////////////

int vali;  //inner val
int prevVali; //outer prev val
////////////////////////////////////////////////////////////////////////////////////

//initializes the IMU

void setup()
{
    Wire.begin();
    Serial.begin(38400);

    Serial.println("Initialize MPU");
    mpu.initialize();
    Serial.println(mpu.testConnection() ? "Connected" : "Connection failed");
    outer.attach(11); //servo on pin 11 for large ring y
    inner.attach(12);//servo on pin 12 for small ring x

}
////////////////////////////////////////////////////////////////////////////////////

void loop()
{

    mpu.getMotion6(&ax, &ay, &az, &gx, &gy, &gz);

    valo = map(ax, -17000, 17000, 0, 179);
    if (valo != prevValo)
    {
        outer.write(valo);
        prevValo = valo;
    }

     vali = map(ay, -17000, 17000, 179, 0);
    if (vali != prevVali)
    {
        inner.write(vali);
        prevVali = vali;
    }
}
////////////////////////////////////////////////////////////////////////////////////
```