Team B: The Monkey Bots

Trevor Decker, Stephanie Chen, Ian Hartwig, Ian Rosado
Final Report for Mechatronics Spring 2015

1. Abstract

At the beginning of the semester, we were given the challenging task of building an autonomous window washing robot that could remove dirt from windows of two different sizes. We took the task one step further by attempting to design and create a mechanism that would be able to easily move between window panes on a skyscraper. The project took 17 weeks to complete and cost $1032.10. This report includes detailed information regarding the design and fabrication of the robot as well as an analysis of system performance.

Table of Contents

2. Project description

In the window cleaning profession, most injuries on the job are a result of what the Occupational Safety and Health Administration call “falls (from elevation)” [1]. In addition to the slipping from soap or water, incorrect harnessing have contributed to the hazardous work environment associated with window washing. For the Window Washer project, our task was to design and prototype a robot that can automatically clean one side of a plexiglass window without scratching or damaging the surface. This hopefully would be able to allow window washer to spend less time lofted up in the air and decrease the number of casualties per year.

3. Design requirements

The goal of the Window Washer is to automatically clean one side of a glass or plexiglass window without scratching or damaging the surface. The robot has to accommodate any window size within the range of 3’x4’ to 5’x6’ with a solid frame. The robot has no required start position but the robot must be able to clear randomly applied dirt or streaking. Some of the other mandatory requirement include the following:

- 100% coverage with speed greater than or equal to 10 ft^2/min
- 2 ft^2 footprint when retracted, if applicable, with any aspect ratio
- cannot be supported from the ground
- no external structure can be installed other than a pneumatic compressor

Success will be based on visual inspection 3 feet away with no streaking or residual moisture 30 seconds after washing is complete. As a team, it was also decided that the robot should be able to move from window to window in order to be able to clean multiple adjacent windows. To do this, the robot must be able to pass over the window frame and continue to clean. Ideally, we want our robot to be practical to use on tall buildings.

4. Functional architecture

To accomplish the tasks illustrated above, our design focused on tackling window washing using three subsystems that allowed for localization, locomotion and cleaning on the window frame. Two gripping mechanisms hook onto the window frame and allow the robot to climb up and down the windows of a “skyscraper”. An extendable arm allows for flexibility in adjusting for window size. We planned to have a moving cleaning head which could move along the superstructure of the robot's arm as well as move its cleaning head in and out of contact with the window. An Arduino was used to control motion as well as sensing. Below is a block diagram that shows these main functions (Figure 1).

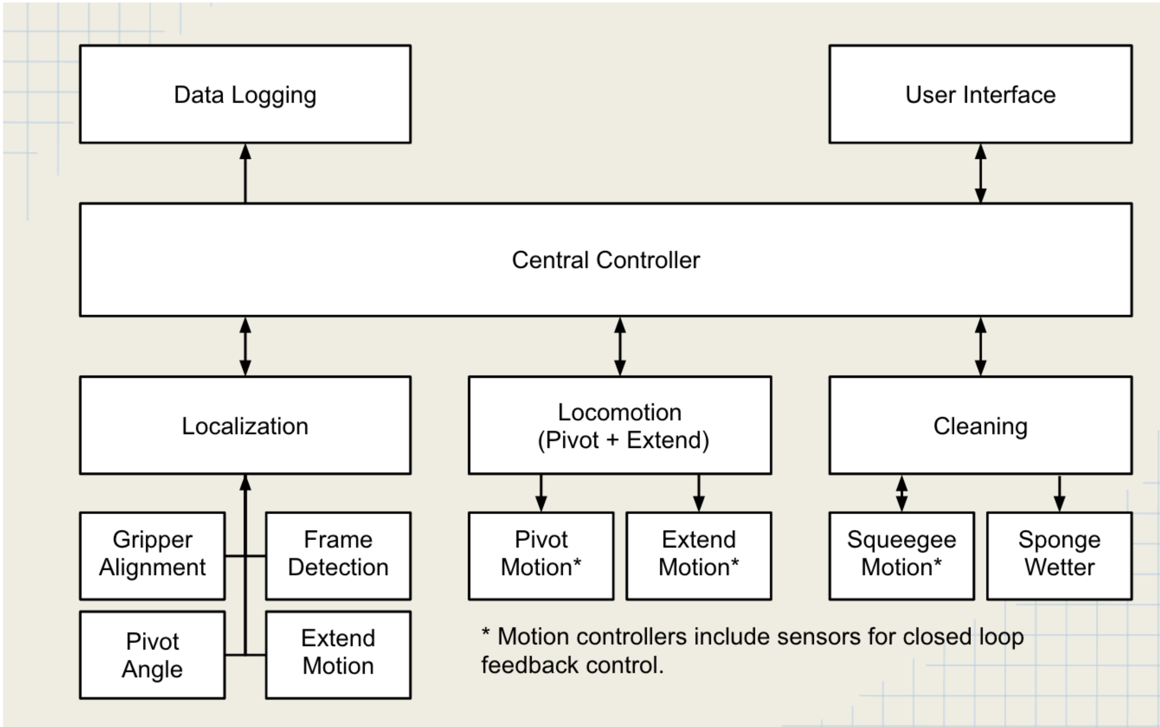

Figure 1. Primary functions of the Monkey Bot.

5. Design concepts

In the following section, the design choices made for each subsystem of the functional architecture are explained.

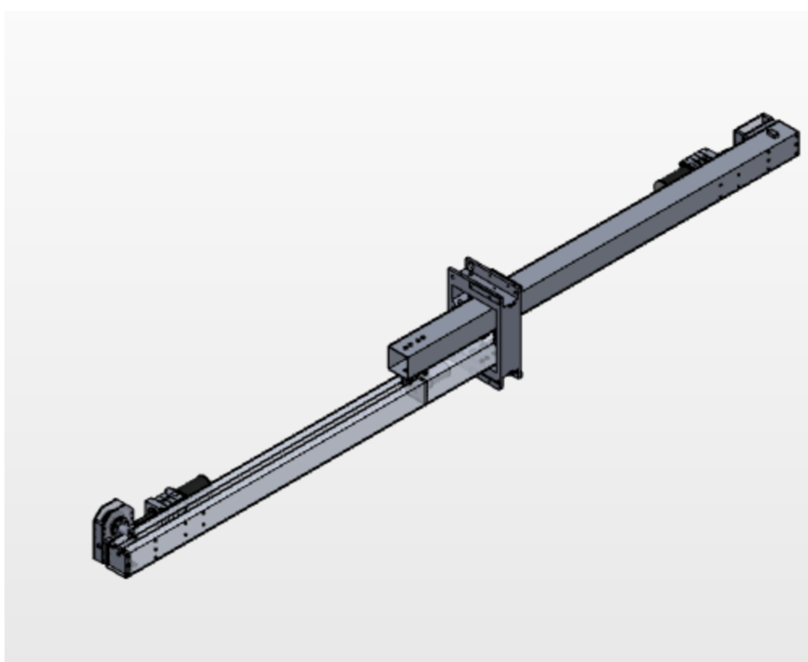

Figure 2. An isometric view of the cad model we developed of our robot.

Locomotion

Originally, we considered many methods of locomotion for the whole robot. We considered using suction to attach ourselves to the window. This would have allow us to move freely to any part of the window. The base of the Winbot 730 by Ecovacs [2] has an inner ring that is connected to a vacuum pump which holds the robot to the window. One of the disadvantages of the this method include streaking caused by the suction mechanism and loss of suction due to dirtying of the surface. Also, the area under the suction mechanism will be left uncleaned in the end. Therefore, we decided to explore other attachment options. We briefly discussed using magnets to secure the robot and clean both sides of the window. However, since having a magnetic attachment on the inside would be unrealistic in the cleaning of skyscrapers, we decided to pursue other avenues. The possibility of using a quadrotor or another type of flying mechanism as a means of locomotion was considered. This would allow for the robot to be very versatile and mobile, but in the end, issues with weight, controls, precision, and varying expected air conditions (outside of a skyscraper) also rendered this idea unfeasible.

We decided to proceed with moving up and down the window by gripping on to the frame because it was the most realistic considering the structure of skyscraper windows and our design goal of moving between window panes. To be able to move between window frames, our robot needed to be able to hold onto the frame from just one side to flip. Here is a general equation we used to estimate the size of tube, assuming a 5 lb mass at the end:

$$\delta = \frac{PL^3}{3EI}$$

$$\delta \geq \frac{5\ lb * (6\ ft * 12\ in/ft)^3}{3(10000000\ psi) * \pi * (d_o^4 - d_i^4)}$$

$$d_o^4 - d_i^4 \geq \frac{1.2673}{\delta}$$

By choosing an acceptable deflection, δ, for our application, we could calculate the outer and inner diameter of the beam that we needed. We chose to build the body out of thin-walled extruded aluminum for strength and chose a 2 inch x 2 inch cross section because it would prevent deflection.

As for the pivoting motion, we knew that we needed a high gear ratio, to allow for lifting from just one end. At the beginning, the motion of the two pivoting gearboxes were coupled with shafts, allowing us to drive both grippers with one motor. We eventually had to use two motors due to a change in our gripper design. More details and technical justification can be found in the subsystem section.

Localization

In order to ensure that all parts of the window are being cleaned adequately, we first brainstormed ways to determine location on the window. At first, we considered using vision. This involved us attaching cameras to both sides of the robot and using the images of the frame to position the gripper correctly for climbing. More specifically, current camera images could be compared to an image with the ideal orientation of the frame relative to the gripper.

This would allow for adjustment of the gripper until the two images matched. This idea was discarded because it would require that we had a camera near the center of our grippers. After the first prototype of this software was working we decided to not continue its development due to the added complexity of including it in our software stack. Our final design used the feedback mechanisms built in to each subsystem (potentiometers in the grippers, limit switches on the cleaner) to keep track of which parts of the window we have already gone over.

Cleaning

The first iteration of the cleaning unit included a squeegee and sponge pair that would be actuated back and forth. The sponge also had a gravity-fed water tank. We thought this was the optimal combination since the sponge could scrub and soak up dirt while the squeegee pushed the dirt to the bottom of the window as the cleaning unit moved.

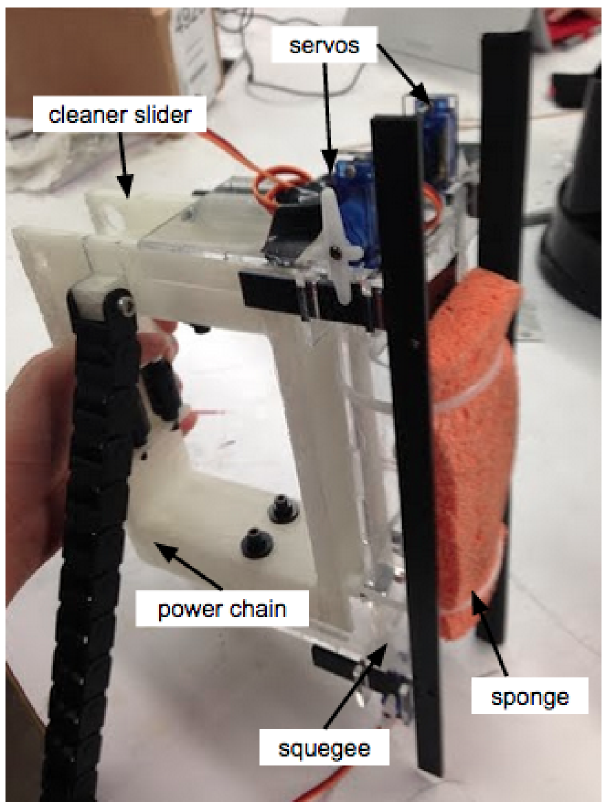

Figure 3. First iteration of sliding cleaner unit.

However, from brief testing with a sponge, we realized that a single sponge would get dirty quickly and thus, not clean the window effectively. To mitigate this issue, we designed a cleaning unit with a rotating component to allow for the cleaning surface to be automatically switched out once it gets dirty.

Electrical/Control Architecture

At the beginning of the semester we decided to attempt and use ST-Discovery 32F429 board as our microcontroller. This embedded computer would have allowed for us to have a Real Time Operating System (RTOS) running on the robot, making the control work much easier. Additionally it would have allowed for us to display debug information to the boards lcd. Moreover the discovery board does not have the same quarks and undocumented behavior that arduinos can show for larger more complex projects which use multiple libraries. Unfortunately, we were not able to fully bring the ST Discovery board up to the operational level that was needed for our robot so we had to move our project to Arduino C++ code in the week before final demos. We were able to have the st discovery board produce pwm signals, digital signals, and read both analog and digital signals. However when we tried to have the board read analog signals using its built in adc we were unable to consistently produce valid pwm signals. It is believed that this error was caused by the use of a single pin for both functionalities but we ran out of time to fully debug the issue.

While rushed, the move to Arduino was successful, we were able to avoid the problems which were discussed previously by reimplementing all arduino function ourselves so we knew exactly which pins and timers were in use at any time. The Arduino based control system handled the same number of inputs and outputs for motors and sensors. However, it lacked the user interface and feedback through a touch screen. This was not a core feature of our embedded software, so the tradeoff was worth it.

6. Cyberphysical architecture

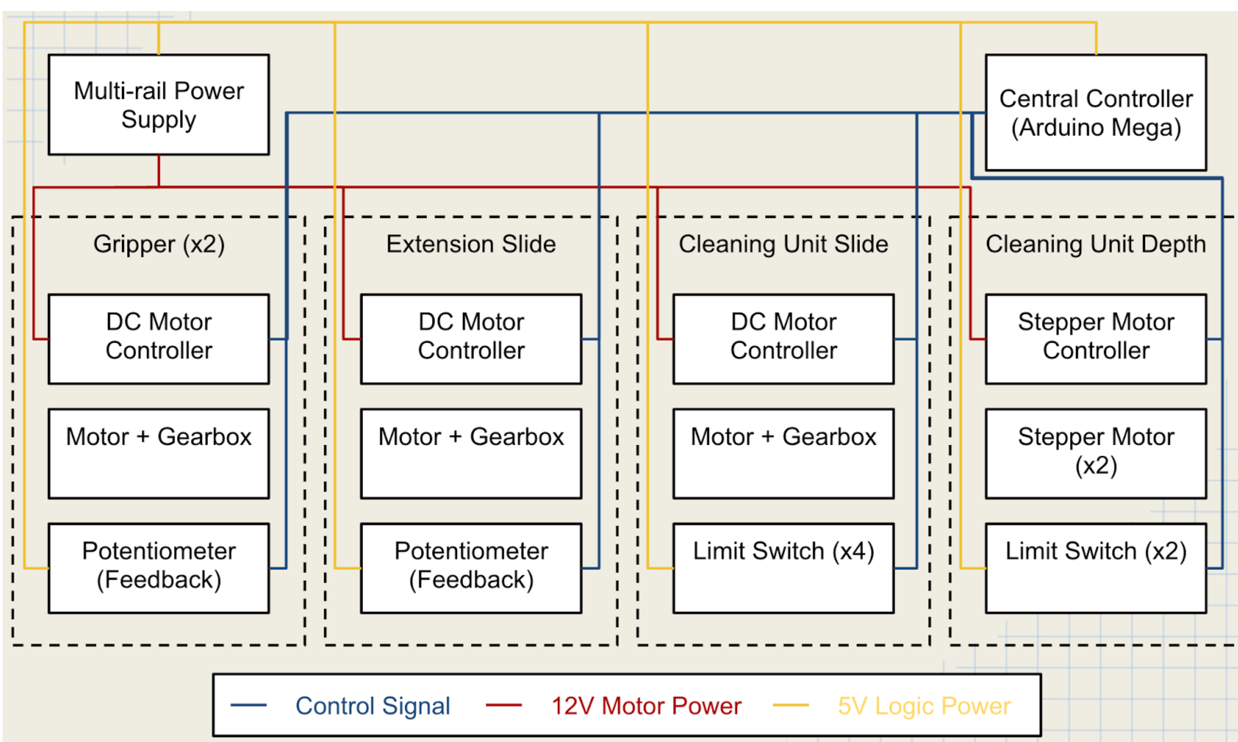

Figure 4. Our final cyberphysical architecture.

Our cyberphysical architecture is directly related to the tasks we set out to perform in our design and functional architecture. It is arranged as a central controller and power regulation system that feeds several closed-loop controlled systems. Each subsystem has the options of 5 and 12V power for motor and logic, respectively, and is connected to both logic inputs and outputs.

7. System Descriptions and Evaluations

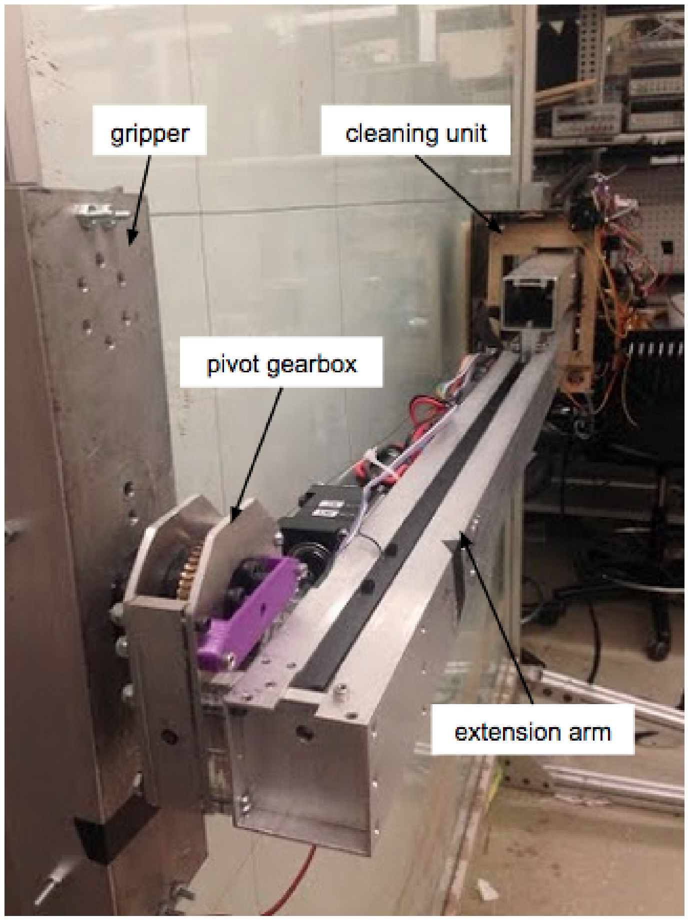

Figure 5. Monkey bot hanging on the window frame.

Our final system has 5 degrees of freedom: 2 for the rotation of the gripper wrist, 1 for the approach of the cleaning unit to the window, 1 for the horizontal motion of the cleaning unit along the arm, and 1 for extending and retracting the arm.

Pivot GearBox

The pivot gearbox functions to change the orientation of the gripper system relative to the extension arm. Initially this motion was meant to allow for the robot to tilt itself a few degrees down or up the window, while a single gripper supports all of the weight of the robot. The Pivot GearBox assembly consists of a 40:1 vex versaplanetary gearbox which is attached to a custom designed and built worm gearbox with a 48:1 reduction. The custom gearbox provides us with a total reduction of 1920:1 at 90 degrees from the original motor output shaft. The pivot gearbox is driven by a BAG motor. The output shaft of the pivot gearbox has a theoretical output torque of 384 nm and can rotate 6.25 times per minute. After construction we noticed that the worm gear caused for us to have a nontrivial amount of slop in the grippers orientation.

The Pivot Gearbox had a 270º turn potentiometer built into it. The potentiometer allowed for the pivot system to use closed loop control and for an automated attachment process to be developed.

Grippers

Being able to localize the robot on the window is dependent on its ability to reliably grip onto the window without slipping. We considered the idea of using a rolling gripper because friction from the wheel would not be adequate to hold up a robot of the assumed weight of 15 lbs. Even if we could creating a firm grip with wheels, the wheels would be too difficult to drive. We originally designed and manufactured a gripper that opened and closed using a lead screw coupled to a motor, as shown below. We were able to hold approximately 5 lbs with this design but determined that it was insufficient for the weight of our robot so we turned to a solid C-Channel gripper in the end.

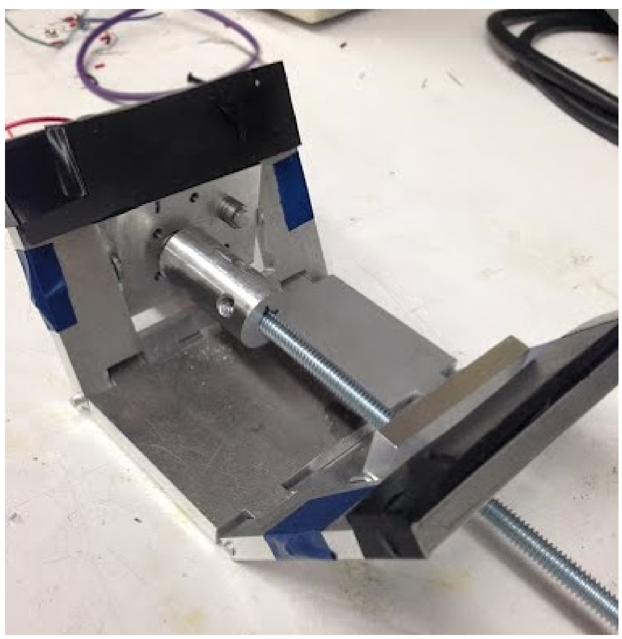

Figure 6. First gripper iteration.

The final iteration, however, consisted of two large pieces of aluminum c-channel with rubber padding. The c-channel fits over the frame of the window, and when the full weight of the robot is applied to the gripper, the large resulting moment creates a friction lock as the top and bottom of the c-channel bind against the window frame. The rubber padding increases the coefficient of friction to prevent slipping. One large advantage to this mechanism is that the weight of the robot actually helps the grippers stick to the window frame by applying more normal force and thus creating more friction.

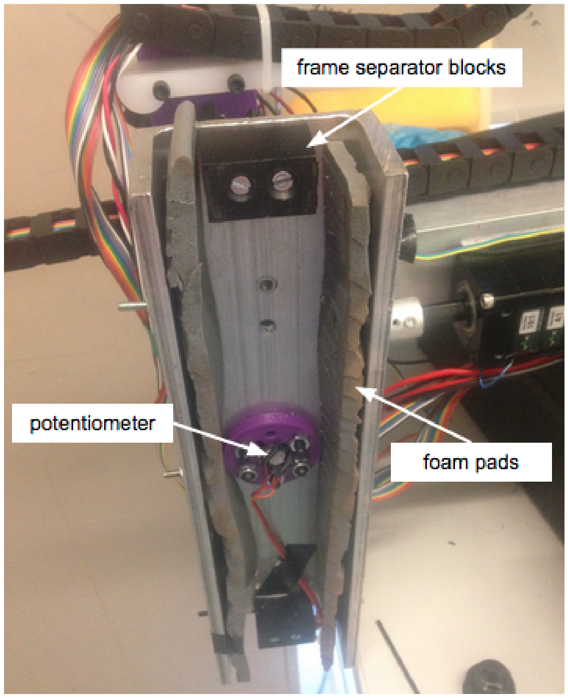

Figure 7. Gripper unit consisting of a C-Channel fitted with 3 main components.

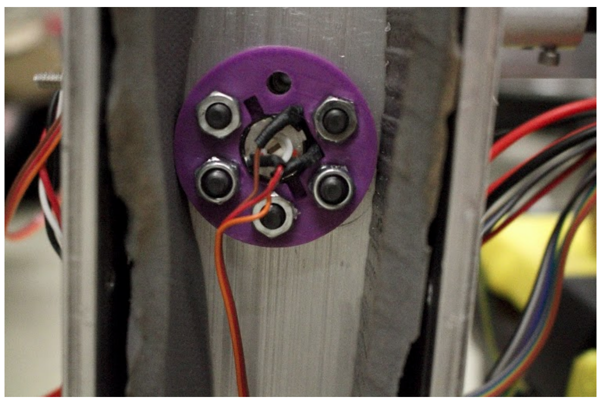

Figure 8. Potentiometer interfacing with C-Channel and pivot gearbox (behind C-Channel and not shown).

Extension System

The arm of the robot itself consists of two 2”x2” square aluminum beams that are able to move relative to each other in order to extend or contract the robot. Each beam has a .5” slot cut into one side, into which a small cart can fit and slide along. By rigidly attaching a cart to each beam and fitting the two together, the motion can be constrained and controlled relatively easily. To actuate the extension mechanism, there is a dc motor mounted in one of the beams which is attached to a lead screw, which is fed through a threaded nut that is held in place on one of the carts. Using a lead screw allows for smooth linear motion and also provides enough mechanical advantage to move the arm consistently. The cart position is controlled in closed-loop using feedback from a 10-turn potentiometer. In the arm tube opposite the one with the threaded rod a long timing belt links the potentiometer and the cart, giving us an accurate position measurement for this system.

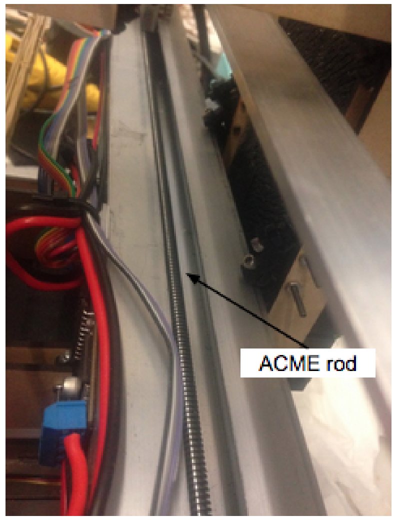

Figure 9. ACME rod that drives extension motion.

Cleaning System

The cleaning system that we developed is a box which would rolls along the main arm of our robot. It is able to transfer from a single tube to rolling along both tube, which is important when our arm is extended to more than 3 feet. This motion is possible due to the unique way that our cleaner was designed. Rollers are placed in the top left, bottom right, and middle right of the cleaner so that the cleaner will always have at least three rollers in contact with the arm superstructure at any time. After initial issues dealing with the transition form 1 tube to 2, we changed the design to utilize diagonal rollers that would be able to deal with any unexpected slop or play in the system.

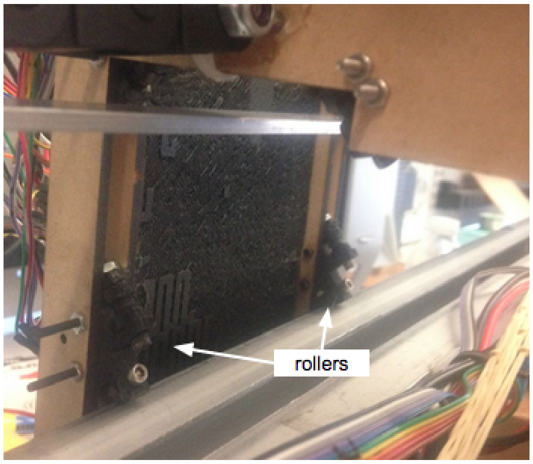

Figure 10. Cleaner rolling along extension arm.

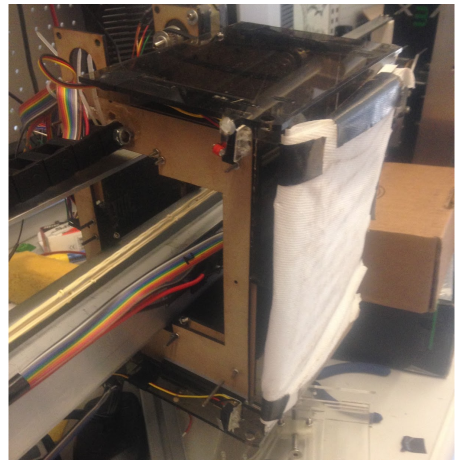

Figure 11. Cleaning pad made of paper towels.

The cleaning unit was moved back and forth along the arm using “drag chain”. Drag chain is usually used to house wires and cables in 3D printers and CNC machines, but has an interesting property that it is only able to bend in one direction. By orienting the chain so that it would not be able to bend until it is clear of the arm, it was possible to both push and pull the cleaning unit from one side while dealing with a variable length arm (which would be more difficult with a timing belt or pulley).

To actuate the chain, we used a dc motor with a custom sprocket that fit into the gaps in the power chain to drive it in both directions. The control logic is very simple since the only information we need is when the cleaning unit is at either end of the slide. When it is time to clean, the controller applies a constant motor signal until one of a pair of limit switches tells us the switch directions. The controller keeps track of how many passes we’ve made in this location.

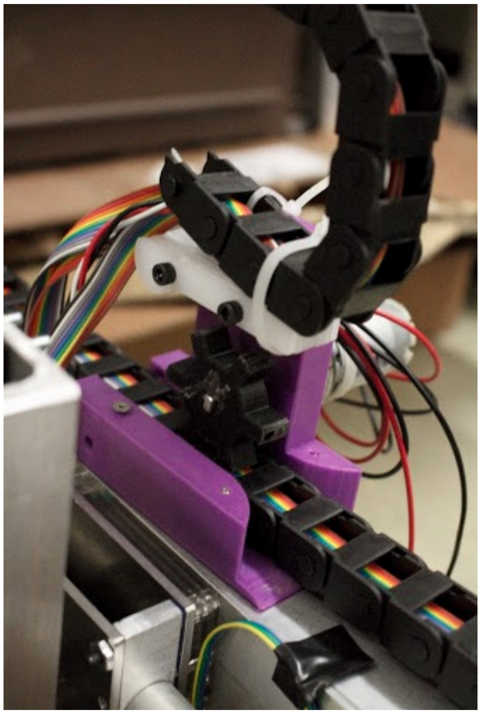

Figure 12. Power chain with sprocket.

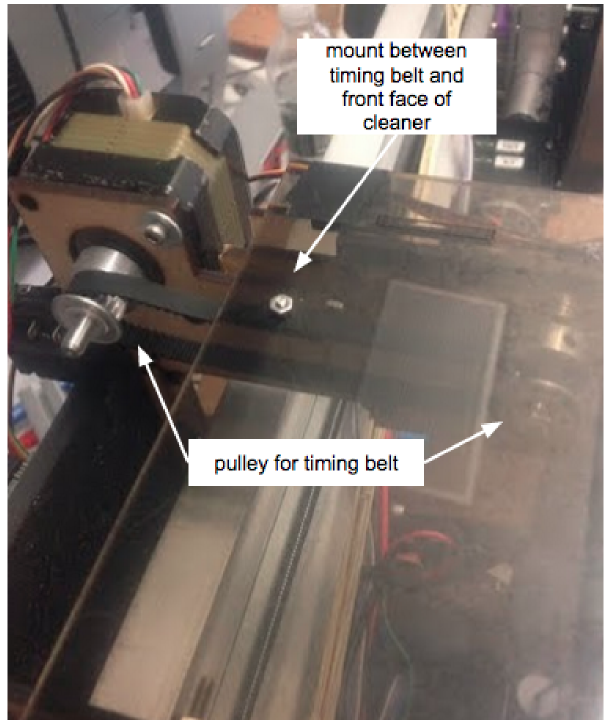

Figure 13. Stepper motors apply the cleaning surface to the window pane.

We can also control the distance between the cleaning surface and the window with the servo and timing belt mechanism depicted below. The turning of the servo motor pulls the timing belt back and forth, thus pulling the cleaning unit back and forth.

8. Project management

Schedule

The original schedule that we we planned to follow is shown below:

Table 1. Beginning of semester schedule

	Week															
Activity								0	1	2	3	4	5	6	7	8
Website Design																
Mechanical Mockup																
Mechanism Design																
Electrical Design																
Mechanism Construction																
Sensor Bringup																
Spring Break																
Software Construction																
Mechanism Test																
Integration Testing																
Emergency Redesign																
Final Report																
Lab Clean Up																

We did not completely follow the schedule that was originally proposed. We ended up spending weeks 3 to 9 in mechanism design and mockup. Since we only started to really build our robot until after spring break, the robot was not fully built until week 17. At which point we found that several of our initial design decisions made our climbing down approach impossible without significant system redesign. Since we only had 1 week until the public demo, we attempted several small redesigns to get the system operational but were not able to reach our initial goals. We underestimated the amount of time for subsystem building and assembly and should have dedicated time to simplifying our design when we found ourselves deviating from our timeline.

Budget

Table 2. Final budget list of purchased items

Description	Quantity	Unit Price	Total Price (Shipping Included)
Camera for testing positioning with microcontroller	1	$10.99	$10.99
Pololu DC Motor Controller	2	$24.95	$58.35
Rubber U-channel	1	$14.00	$19.35
ACME 1/4"-16 Rod	1	$5.14	$12.20
ACME 1/4"-16 Nuts	4	$2.08	$8.32
M5x8mm low profile cap screw	1	$5.39	$5.39
.125"x1.25" bar stock	1	$5.56	$39.54
2" square tube	1	$20.52	$20.52
.5" square bar stock	1	$8.90	$8.90
3/8" hex stock	**1**	$9.22	$9.22
1.5"x1.5" c channel	1	$3.92	$3.92
Tiny bearings for slide	2	$19.49	
Gripper Motors	1	$16.95	$21.93
Worm Gear	1	$35.98	$41.22
Worm Worm	1	$24.60	$24.60
Rubber Duck	1	$2.79	$2.79
Delrin Sheet 1/8"	1	$9.93	$9.93
Rotary Spring	1	$5.99	$5.99
String Pot Parts	2	$12.00	$32.84
Cleaning unit bearing bolts	1	$8.07	$9.74
Cleaning unit bearing bolts	1	$3.61	$3.61
Pivot gearbox rod	1	$12.66	$12.66
Pivot gearbox bolt	1	$10.77	$10.77
Cleaning unit bearings	20	$0.99	$28.75
1/8" Delrin Sheet	1	$25.60	$25.60
Versa Gearbox	1	109.96	$122.69
Worm Gear	1	$35.98	$41.22
Worm Worm	2	$30.60	$61.20
RoboClub 3D Printing	8	$0.80	$6.40
Microcontroller	1	$24.00	$32.15
Cleaning unit bearings	10	$0.99	
6-32 x 1.75" for stage 3 mount	1	$7.17	$7.17
6-32 x 3/8" for couplers	1	$8.50	$8.50
6-32 x 1/2" for couplers	1	$8.33	$8.33
6-32 nylock nuts	1	$2.67	$2.67
2-56 nuts	1	$0.90	$0.90
M3x10 for extend motor	1	$6.88	$6.88
Steppers for cleaning unit	2	$17.95	$45.35
Replacement linear slide motor	1	$15.00	$15.00

L6470 dSPIN Motor Driver	4	$8.50	$42.15
Pololu DC Motor Controller	2	$24.95	$49.90
Timing belt for cleaning unit	1	$16.00	$16.00
Cable Carrier for moving cleaner	2	$7.90	$15.80
Versa Gearbox	1	$109.96	$122.66
		TOTAL:	**$1032.10**

We were quite conservative with our purchases and only purchased things after we had settled on a design, we were able to stay close to the allotted budget of $1000. The main contributor to us going over budget is due to the extra motor and gearbox for the pivot unit. Since our system was complex, we had to purchase many specialty items and this contributed significantly to the overall cost.

9. Conclusions

At the end of the semester, we accomplished the tasks of gripping onto the frame and moving the cleaner along the window. However, since we had difficulties climbing up and down the window, we could not clean more than one section of dirt. Our coolness factor of flipping between window panes led us to solve a bigger problem than the one assigned and thus, caused us to fall short on our original goals of cleaning the whole window pane. Below is a list of issues that the robot faced at the end of the semester.

Issues and Modifications

1. The power chain needs a track since it falls off the extending arm and buckles when pushing the cleaning unit. The cleaner would need to be redesigned to fit around it.
2. The spacing of our cleaner and grippers needs to be modified to allow the cleaning unit to move towards and away from the window.
3. The cleaner material created friction with the window, causing the power chain motor to not be able to move the cleaning unit. However, reducing normal force on the window was less effective at cleaning.
4. The robot would fall off of the robot when stepping down on the frame due to deflection in the aluminum, slop in our robot and the weight of the system. Could be fixed by shifting the center of gravity of the robot to be toward the window, building actuated grippers for finer control or including inserts that hooked on to the inside of the 80/20 window frame.
5. The power supply we were using had a current cut off that would trigger when the grippers would rotate and the contact point between the robot and the frame was metal instead of rubber.

Incomplete Tasks

1. Rotating the cleaning head material so that it does not get saturated, designed but not fully implemented.
2. Stepper driver motion of the cleaning head in and out of the window.
3. Motion down the window (deemed impossible given the current robot mechanics) .

Since there were various mechanical and software challenges due to the complexity of our design, our first step would be to redesign the robot to ensure that it could clean one pane before trying to move between different panes. More specifically, we could decrease the weight by using one single tube for the telescoping arm and have one stationary and one moving gripper. With modification we could use a timing belt instead of a power chain for the motion of the cleaner. Also, moving forward we would focus on perfecting control using the Arduino instead of dedicating time to figuring out the ST Discovery Board. Although we should have built a simpler robot to complete the challenge, we had a good approach for a real-world setting and with more time we could have accomplished the course goals as well as the reach goals.

10. References

[1]http://www.newyorker.com/magazine/2013/02/04/life-at-the-top
[2]http://www.nytimes.com/interactive/2013/11/21/technology/how-a-winbot-730-robotic-window-cleaner-works.html?_r=1&