Extendable Pantograph Arms

Rick Goldstein
Robotics Institute
Carnegie Mellon University
rgoldste@cs.cmu.edu

Manuela Veloso
School of Computer Science
Carnegie Mellon University
mmv@cs.cmu.edu

Abstract

When designing a robot to interact with people, the decision to incorporate a robot arm may arise. In this paper, we investigate adding an inexpensive, functional arm to our mobile CoBot service robots. Specifically, we examine two-dimensional extendable pantograph arms for CoBot. Pantograph arms have intuitive kinematics and inverse kinematics. Pantograph arms are modular and adding additional linkages corresponds to simple changes in the kinematic calculations. These arms have several advantages (and disadvantages) compared to traditional robot arms. A prototype pantograph arm is currently in development and our goal is to attach a modular pantograph arm to CoBot to perform simple needed tasks, such as knocking on doors and pressing elevator buttons.

Introduction

CoBots are mobile service robots which use real-time localization and navigation to move around office buildings, in particular at Carnegie Mellon, while performing tasks such as guiding visitors and delivering objects (Biswas and Veloso 2013; Veloso et al. 2015). CoBots use symbiotic autonomy, meaning they overcome their limitations and ask for help when they are unable to complete a task by themselves (Rosenthal, Biswas, and Veloso 2010). For example, CoBots do not currently have arms, so when they wish to use an elevator, they must ask humans to assist by pushing floor buttons. Symbiotic autonomy works well in many situations, but our elevator example would fail when there are no people around, so no one can assist CoBot in pressing elevator buttons. A very similar issue arises when CoBot wishes to deliver something to an office with a closed door. CoBot cannot knock at the door, so CoBot must announce itself through speech, which people may not hear when the door is closed. It could ask a nearby person to knock on its behalf; however, this again would fail if there is no one around. Most of CoBot's tasks consist of going to office locations; it would be useful for CoBot to have an arm to knock at doors.

We would like to add an arm to CoBot so that it can perform these tasks without human assistance. At first, we would like CoBot to be able to knock on doors, push buttons, and perform human-like gestures. We are not currently working towards general object manipulation.

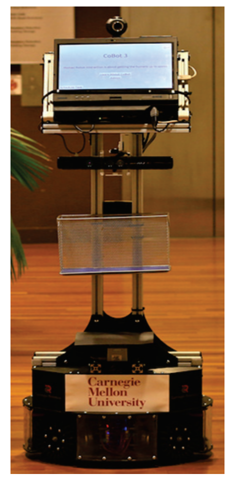

Figure 1: CoBot 3

We have three design principles in mind for our arm for CoBot - modular, low-cost, and minimal footprint when not in use. We want our arms to be modular so that we can outfit each CoBot with several similar arms. We want the arms to share similar kinematics. Yet we want the design to be flexible so that each arm can have a different, function specific use (this could take the form of function specific end-effectors). We want our arm to be low-cost so that we can outfit each CoBot with multiple arms. The cost of three or more arms, on each of the four CoBots, can quickly add up. CoBot is a general service robot and many of the services it provides are not reliant on arms. We want an arm that can remain out of the way when CoBot does not need to use the arm. Extendable pantograph arms satisfy these three design principles.

In this paper, we first describe pantographs in more detail and how they relate to the extendable arm we are building for CoBot. We briefly touch on related work. We then walkthrough a simple example deriving the kinematics and inverse kinematics calculations for a 2D, two-degree of freedom pantograph arm. We illustrate our prototype arms and discuss a knocking algorithm. We compare and contrast pantograph arms with traditional arms. We discuss options for modularity. We touch on several follow-up research topics, including safety and designing a 3D pantograph arm. We conclude with future goals.

Pantograph Background

Pantographs are a set of linkages where movement at one point determines the movement through the rest of the system. Figure 2, adapted from (Wikipedia 2016), is a pantograph used to scale images. A user manipulates the point circled in blue. The pantograph output is the point circled in orange. As the user manipulates the point in blue in the shape of a lowercase a, the point in orange traces a similar, yet larger, lowercase a. One more common example of a pantograph is a scissor lift, where the movement of one point, located at the base and circled in red in Figure 3, determines the height of the lift.

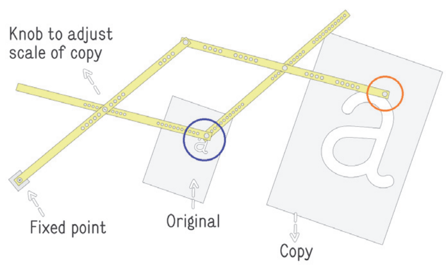

Figure 2: An image scaling pantograph, from (Wikipedia 2016)

Figure 3: A Scissor Lift, from GlobalIndustrial.com

For our pantograph arm, movements at the base will determine the movement through the rest of the system. This is advantageous in tranfering the majority of the weight of the system to the base. The first version of our extendable pantograph arm will have two degrees of freedom, both located at the base. Later sections will discuss how to add additional degrees of freedom and modularity to the arm.

Pantograph 1A, in Figure 4, is a mockup of a simple pantograph arm. This system has four rigid links (AB, BD, CD, and AC). In this simple example, all four links are equal length, $l_{AB} = 1$. It is not necessary for all four links to be the same length, but $l_{AB} = l_{CD}$ and $l_{AC} = l_{BD}$ are necessary so the mechanism is a parallelogram with easy to calculate kinematics.

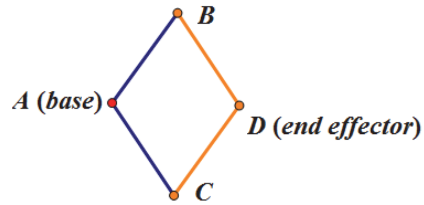

Figure 4: Pantograph 1A

Each of the blue links (AB and AC), can be rotated about A by a separate rotating servo motor. Because our rigid linkages form a closed loop, the location of the end-effector at D is forced. We derive the exact relation in a later section. Please note that in all images, red will mean base, the angles of blue linkages are directly controllable (by a motor), orange points and linkages are forced, and green lines will solely assist with measurements.

We can construct more complicated setups such as Pantograph 1B, where we have several linked parallelograms. For Pantograph 1B, E, H, K, and N all fall on the same line, simplifying our kinematics calculations.

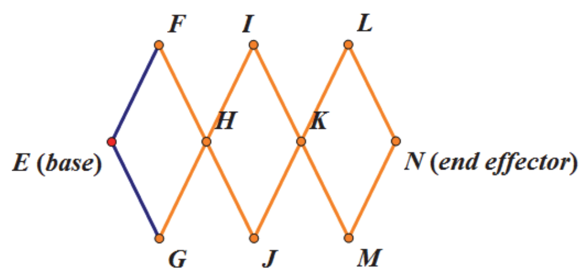

Figure 5: Pantograph 1B

Several other posible designs are presented below - 2A is a non-rhombus parallelogram, 2B and 2C are different ways to combine identical parallelograms, and 2D shows the combination of arbitrary parallelograms. These images demonstrate a fraction of the variety possible with pantograph arms.

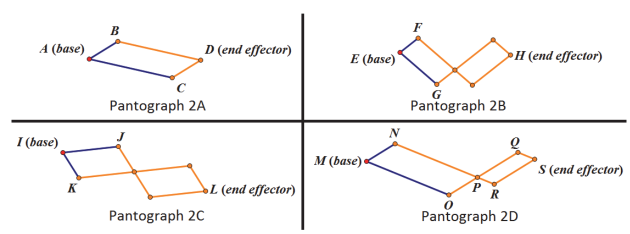

Figure 6: Pantographs 2A-2D

Related Work

There is a lot of work relating to robot arm hardware design and manipulation (Lu et al. 2012; Circuit Digest 2015).

Here, we focus on discussing hardware designs and how they relate to the arm we are developing. While thinking about an arm for CoBot, we looked at several commercial options. Universal Robots and Kinova Robotics both produce commercially available robot arms. Kinova arms have 4 or 6 degrees of freedom and the UR3 has 6 degrees of freedom (Kinova 2016; Universal Robots 2015). These arms are useful; yet $20,000+ arms did not seem reasonable for our intended applications. We are hoping to inexpensively outfit several robots with multiple arms which would add up if we were to purchase commercial arms.

Pantograph arms fall into a broader category known as parallel robots. Parallel robots are systems with interconnected linkages that determine the movement through the system (Mecademic 2015). Unlike traditional arms, actuators tend to be located at the base of the system and there are many linkages that are not directly actuated and instead controlled indirectly by the movements of other linkages (Mecademic 2015; Murray et al. 1994). Delta robots and Stuart platforms are the two most-well known examples of parallel robots.

The majority of pantograph arms have one parallelogram through the system. This work discusses how to easily add additional parallelograms, without over-complicating the dynamics. This work has a similar flavor to the work of Aad van der Geest, who built a multi-delta arm by aligning one delta arm beneathe another delta arm to increase range (van der Geest 2014).

Our pantograph arm has the property that it takes up minimal space when retracted. It extends when in use. Systems which change shape while in use are known as deployable structures. Several research groups focus on developing and understanding deployable structures (Pellegrino 2015; You 2015).

Kinematics

This section examines the forward and inverse kinematics for pantographs. We walkthrough the forward kinematics of Pantographs 1A and 1B; we then move on to the inverse kinematics for 1A, then the inverse kinematics for 1B. For all math below, our base is fixed at $(0, 0)$, $l_{AB} = 1$, Z is an arbitrary point on the $+x$ axis, the x-coordinate of the end effector is strictly positive, and angles are calculated the normal way with respect to the unit-circle.

Forward Kinematics

Pantograph 1A has three interesting angles, which are highlighted in Figure 7: $\angle ZAC$ - which is the angle of one motor with respect to the unit circle, $\angle ZAB$ - which is the angle of the second motor, and $\angle ZAD$ - which is the final angle of the end effector from the base. We denote $\theta_1 = \angle ZAC$ and $\theta_2 = \angle ZAB$ for shorthand as the motors which can be controlled. We denote $\theta_F = \angle ZAD$ as the final angle of the system. We denote the coordinates of a point as $P = (P_x, P_y)$. So $D = (D_x, D_y)$ is our end effector position for this pantograph. Applying trig geometry, $C = (C_x, C_y) = (\cos\theta_1, \sin\theta_1)$ and $B = (B_x, B_y) = (\cos\theta_2, \sin\theta_2)$. By the Parallelogram Law (one can treat

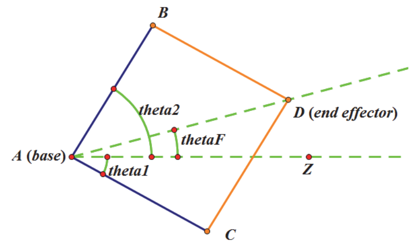

Figure 7: Pantograph 1A, with named angles

parallelogram edges as vectors and then add them) (Weisstein 2016):

$$D = (B_x + C_x, B_y + C_y) = (\cos\theta_1 + \cos\theta_2, \sin\theta_1 + \sin\theta_2)$$

We now examine the kinematics for Pantograph 1B. We can use a similar vector addition trick to get the positions of H, K, and N. From E to to H we are adding in the vectors $\vec{EF}, \vec{FH}$. From H to K, we are adding $\vec{HI}, \vec{IK}$. From K to N, we are adding $\vec{KL}, \vec{LN}$. Since $\vec{EF} = \vec{HI} = \vec{KL}$ and $\vec{FH} = \vec{IK} = \vec{LN}$ (because they are constructed to be parallel and equal length), we are adding in the same amount for each jump. Therefore, our final forward kinematics relation is as follows:

$$N = (N_x, N_y) = (3 * H_x, 3 * H_y) =$$
$$(3 * (\cos\theta_1 + \cos\theta_2), 3 * (\sin\theta_1 + \sin\theta_2))$$

Inverse Kinematics

This section derives expressions for the inverse kinematics of the systems. Please note the importance ensuring that angles are oriented in the proper direction: Suppose we are given a goal final location (D_x, D_y). We know that distance AD (not a link on our pantograph) is $\sqrt{D_x^2 + D_y^2}$. Similarly, we know that $\theta_F = \arctan(D_y/D_x)$. Given these values, we apply the law of cosines about $\angle CAD$ (Please refer to Figure 8):

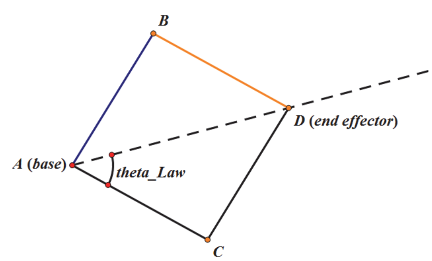

Figure 8: Pantograph 1A, Law of Cosines in black

$$\cos\angle CAD = (AC^2 + AD^2 - CD^2)/(2 * AC * AD)$$

Since $AC = CD = 1$, this simplifies to $\cos\angle CAD = AD/2$ or $\angle CAD = \arccos(AD/2)$. From θ_F, we can easily find θ_1 and θ_2. From Figure 9, $\angle CAD = \angle CAZ +$

$\angle ZAD = -\angle ZAC + \angle ZAD$, so

$$\angle ZAC = \theta_1 = \angle ZAD - \angle CAD =$$

$$\arctan(D_y/D_x) - \arccos(\sqrt{D_x^2 + D_y^2}/2)$$

.

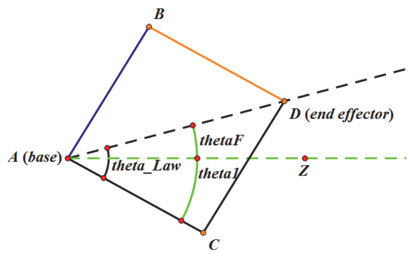

Figure 9: Pantograph 1A, Angle Sums

By similar triangles, $\angle BAD = \angle DAC$. Also, $-\theta_1 + \theta_2 = 2 * \angle CAD$, so

$$\theta_2 = 2 * \angle CAD + \theta_1 = \angle ZAD + \angle CAD =$$

$$\arctan(D_y/D_x) + \arccos(\sqrt{D_x^2 + D_y^2}/2)$$

Using the above inverse kinematics for Pantograph 1A, the inverse kinematics for Pantograph 1B is straightforward. For an end effector position for $M = (M_x, M_y)$, we first find $G = (M_x/3, M_y/3)$. We then treat our system as Pantograph 1A and solve for G as our goal location.

Pantograph Arm for CoBot

The ultimate goal of this work is to build a pantograph arm for CoBot. At first, this arm will perform simple functions, specifically knocking on doors and pushing buttons. CoBot performs many tasks that involve driving to an office (e.g. delivering something). When CoBot arrives at an office with a closed door, it does not have a good way to announce its arrival. Adding a knocker would resolve this issue. The button pusher would allow CoBot to push elevator buttons, so that it can ride the elevator with no human assistance. Future goals of this work may include gesturing, grasping simple objects, or writing.

These tasks are very different and require different flavors of end effectors; our current plan is to outfit CoBot with multiple low-cost modular arms, each with different end effectors. Specifically, by low-cost, we are currently shooting for below \$100 per arm. The second version cost \$50; the majority of which was the cost of the controller. We are in the process of writing Python (ROS) code to incorporate and test the arm with CoBot. The remainder of this section will discuss Versions 1 and 2 of the arm. Images are provided below.

Version 1 of the pantograph arm was built with NXT Mindstorms. It has three parallelograms (rhombi), with a door knocker as the end effector. It has two motors and therefore two degrees of freedom so it can span 2 dimensions (X and Y). It can compress to take up little space when not in use. Both motors are centered at the exact same (x,y) point

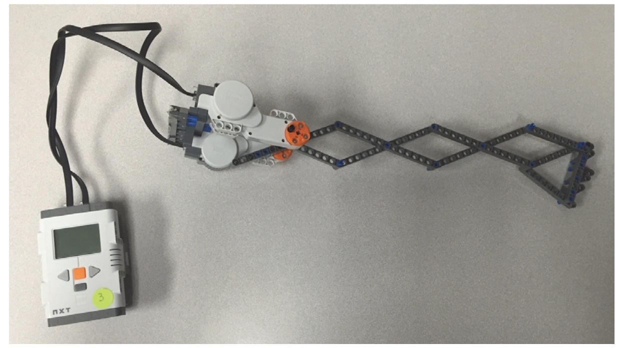

Figure 10: Version 1 - NXT Mindstorms

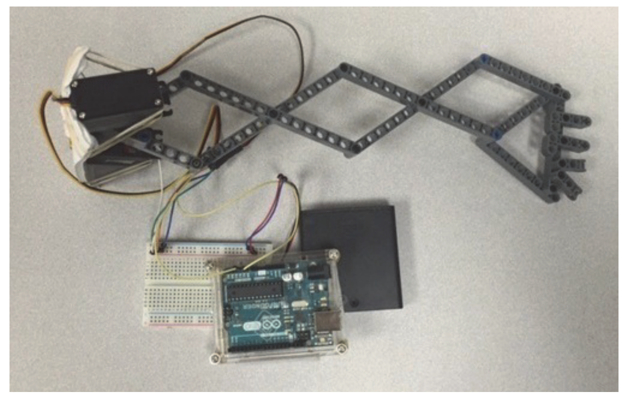

Figure 11: Version 2 - Arduino

(but are in different Z space), which we take to be the origin. This design choice is very important. It allows us to easily create parallelogram angles directly based on the motor rotations. We do not need to worry about aligning our movements because motor rotations already equal linkage rotations. This allows our pantograph to move smoothly and use the simple kinematics described above without any transformations or complications.

Control for the arm can be thought of in angle space as discussed in the above section. Control can also be thought of in extend/rotate space. Moving both motors in the same direction (at the same speed) only extends (or retracts) the arm. Moving one motor forward and the other backwards only rotates the arm.

We programmed the inverse kinematics in RobotC to implement movement to a desired point on this arm. The most important lesson from this was motor gliding. In the diagrams, when one motor rotates, only the end effector rotates. In practice, when one motor rotates, there is force at the end effector and while the end effector moves, it doesn't move the full amount and this is compensated by the undesired effect of the other motor rotating. With Mindstorms, this can be partially offset by always having both motors on, but there is still slip.

Another important lesson was that too many parallelograms in series made the arm slow to respond at times. We suspect this is due to friction throughout the system (additional parallelograms means additional linkage intersections and therefore friction) as well as limitations of motor torque. For this reason, we capped the number of parallelograms in our arm at 3. Future work involves designing and building custom linkages and connectors.

Version 2 is based on the same three parallelogram, 2 DoF design as Version 1. It is built with inexpensive hobby servos (HITEC HS-322HD) and an Arduino Uno as the controller. We continued to use Mindstorm pieces as the linkages. We made this switch because Arduino is easier to interface with ROS. This switch reduced the motor gliding; however, the design is not as robust as intended and sometimes the motors do not stay in place. There is much to improve the design. However, our next primary focus is incorporating with CoBot. We were able to program the inverse kinematics in Arduino to implement movement to a desired point on this arm. We are currently building Python code and hope to test on CoBot in the coming weeks.

Knocking Algorithm

As discussed in the above section, we were able to implement the inverse kinematic calculations and movements in both RobotC and Mindstorms. We used our motion primitives to create two simple states - extended and retracted - for our knocking algorithm. When a knock command is sent, the arm attempts to extend to a fixed location (by sending motor commands), waits roughly .2 seconds, retracts to a passive position, waits roughly .2 seconds, and then repeats this two more times. Note that since knocking requires making contact with a surface, while our motors attempt to extend to a specific location, they don't make it the full way, because a door or wall is in the way. This produces a knocking sound as intended. We do not model the collision and treat our motors as in a noisy position when the arm is in contact with the wall.

Below, are some stills taken from the knocking algorithm, every .25 seconds. Two things worth highlighting are that this entire algorithm is quick, taking less than 1.5 seconds and (you will have to take our word for it or view our video at https://youtu.be/jptSIcMNVWk) the knocks sound like knocks!

In testing, this works fairly well with both versions of the arm. For Version 1, we noticed some encoder errors when using the NXT motors and RobotC. One knock cycle (three extend/retracts) produced some error in the encoders from each knock, but each of the three knocks was roughly on-target. Running several more knocks without manual resets to the arm led to an accumulating error with the NXT motors. With the NXT motors, the bounces off of the wall were not fully captured by the motor encoders which is what caused this noise and accumulating error. For Version 2 of the arm, we did not have this issue. While there is noise in the motor positions after the bounce off of the wall, the encoders work fine and the retract step successfully pulls the motors back to the rest position.

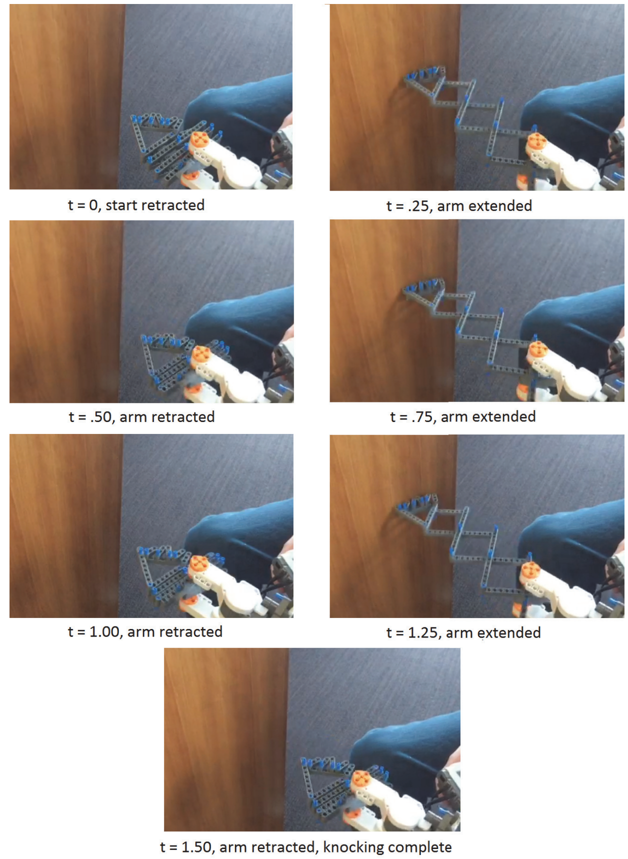

Figure 12: Stills from Knocking Algorithm. Video uploaded at https://youtu.be/jptSIcMNVWk

Pantograph vs Traditional Arms

There are many similarities and differences between the 2 DoF pantograph from 1A and a planar 2 DoF 2 link manipulator with revolute joints, which we refer to as a traditional arm. The traditional arm has a revolute joint at the origin, a link from this manipulator to a second manipulator, and then a link from the second manipulator to the end effector. To simplify this comparison, we will assume both links of the traditional arm are the same length, that this length is the same as in Pantograph 1A, and equal to 1.

Some basic similarities are that both manipulators have two degrees of freedom, and approximately span the same space. Theoretically, both manipulators can reach any point within a radius of $2 * l_{AB}$.

Another interesting comparision is the kinematic equations. The two arm linkage has additional θ_1 terms in its coordinates, but the general structure of these equations is the same:

$$Panto_x = \cos\theta_1 + \cos(0+\theta_2) \quad Panto_y = \sin\theta_1 + \sin(0+\theta_2)$$

$$Trad_x = \cos\theta_1 + \cos(\theta_1+\theta_2) \quad Trad_y = \sin\theta_1 + \sin(\theta_1+\theta_2)$$

There are several differences for implementation: one advantage of pantographs is that if motors are the majority of the weight of a system, then all of the weight can lie at the base of the system. One disadvantage is the need for additional linkages. A pantograph with n motors needs n^2 linkages, while the arm only needs n linkages. This can translate to a larger footprint during execution, which could prove impractical in constrained environments.

Modularity

This section discusses several ways in which these arms are modular. We discuss the following: number of linkages, end-effectors selection, and number of degrees of freedom, and a combination of linkages and degrees of freedom.

Understanding the kinematics for a variety of pantograph arms is built upon Pantograph 1A and the Parallelogram Law. We saw above that adding additional linkages for Pantograph 1B only added constants to the kinematic equations. For Pantograph 1B, we assumed that all lengths were the same. This is not a requirement and any number of links of any length (as long as they maintain the parallelogram structure) can be added. We can easily use the Parallelogram Law to calculate the kinematics for more complicated pantographs. The X-kinematic equations for Pantographs 2A - 2D are listed below. The equations for the Y coordinates are the same with sine replacing cosine. Note that both the base is located at $(0, 0)$ and Z is assumed to be a point on the $+X$ axis as before.

$$2A : D_X = l_{AB} * \cos \angle ZAB + l_{AC} * \cos \angle ZAC$$

$$2B : H_X = (l_{EF} + l_{EG}) * (\cos \angle ZEF + \cos \angle ZEG)$$

$$2C : L_X = 2 * l_{IJ} * \cos \angle ZIJ + 2 * l_{IK} * \cos \angle ZIK$$

$$2D : S_X = (l_{MN} + l_{PQ}) * \cos \angle ZMN + (l_{MO} + l_{PR}) * \cos \angle ZMO$$

These calculations follow from the Parallelogram Law discussed above.

The pantograph design is flexible and a variety of flavors of attachable/detachable end effectors can be added to the end of the parallelogram linkage. (We acknowledge that this isn't unique to pantographs). Currently, we have built a door knocker end-effector for our prototypes and are developing a pointer/button presser. In the future, we will investigate more complicated end effectors such as simple (object-specific?) graspers.

As shown in the above section, 2D pantographs can approximately replicate the movement of 2D traditional arms. We can add additional motors and linkages at the base as desired. Adding a third motor (please see Figure 13) can give us a third degree of freedom (to control, X,Y, and angle of end effector).

Again, this pantograph has fairly simple kinematics:

$$G_x = l_{AB} * \cos \angle ZAB + l_{AC} * \cos \angle ZAC + l_{AD} * \cos \angle ZAD$$

$$G_y = l_{AB} * \sin \angle ZAB + l_{AC} * \sin \angle ZAC + l_{AD} * \sin \angle ZAD$$

When adding motors, we need to add additional linkages. For the most basic setup with N motors, we have $\binom{N}{2}$ parallelograms. Each combination of angles appears exactly once (a combination means a parallelograms with angles $|\theta_A - \theta_B|$ and $(180 - |\theta_A - \theta_B|)$. For a given angle, there are N linkages at that angle, and one can trace a path from the outside of the pantograph going thru each linkage and then

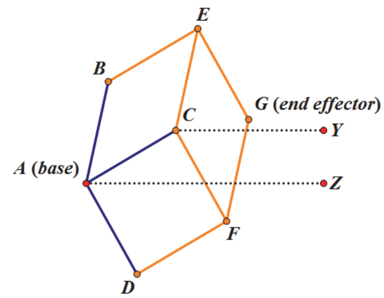

Figure 13: 3 Motor Pantograph

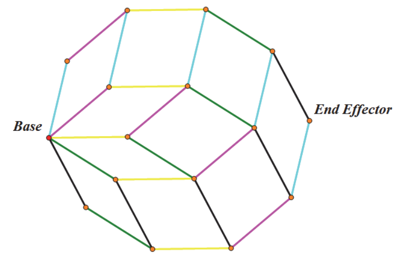

Figure 14: 5 Motor Linkage - Linkages at equal angles color-coded

exiting the pantograph. Please see the below figure with colored linkages.

There are ways to make the pantographs more complicated and to get the effect of an angle an additional time in the kinematic calculators. In the below image, where all non-red links are the same length and red is twice this length, the bottom-most angle has twice the effect as the other angles. The kinematics for this pantograph are below:

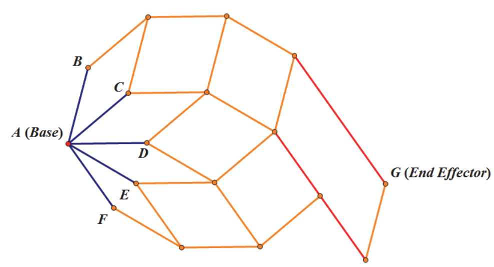

Figure 15: 5 Motor Pantograph - Bottom Link Doubled

$$G_x = l_{AB} * (\cos \angle ZAB + \cos \angle ZAC + \cos \angle ZAD + \cos \angle ZAE + 2 * \cos \angle ZAF)$$

$$G_y = l_{AB} * (\sin \angle ZAB + \sin \angle ZAC + \sin \angle ZAD + \sin \angle ZAE + 2 * \sin \angle ZAF)$$

Similarly, in the below image, $\angle ZAB$ and $\angle ZAD$ each have three times the effect as $\angle ZAC$ (note all linkages are equal length).

$$I_x = l_{AB} * (3 * \cos \angle ZAB + \cos \angle ZAC + 3 * \cos \angle ZAD)$$

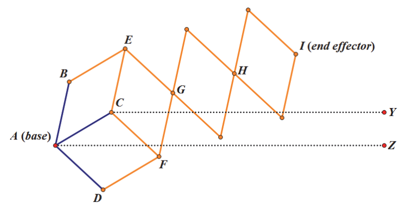

Figure 16: 3 Motor Pantograph - Top and Bottom Links Tripled

$I_y = l_{AB} * (3 * \sin \angle ZAB + \sin \angle ZAC + 3 * \sin \angle ZAD)$

The above design is interesting in that there are four points (C,G,H,I) that all fall on the same line. The position of C is decided solely by θ_2. The positions of the other points are a function of C, θ_1, and θ_3. Because of this property, we suspect there is a way to mount a camera at point C that continually focuses on the end-effector.

There are costs to the modularity. Below are three arms with similar kinematics. A single-parallelogram pantograph arm, a 4-parallelogram pantograph arm (each parallelogram side is one-quarter the length as the single-parallelogram arm) and a traditional arm (each side is the same length as the single-parallelogram arm). Extended, the 4-parallelogram pantograph takes up less space than the single-parallelogram pantograph (It requires roughly one quarter of the area). However, the additional linkages mean more complexity. In testing, additional joints mean movement was not as fluid. Overall, both pantographs take up more space than the traditional arm. However, the 4-parallelogram pantograph could be advantageous if working in an environment without much room to move in the Y-dimension. If our environment were constrained, the single-parallelogram pantograph and the traditional arm would not be able to move to the proper joint angles, while the 4-parallogram pantograph would. Additionally, the movement of the pantographs could be more intuitive to some users as it extends in a straight line from the base.

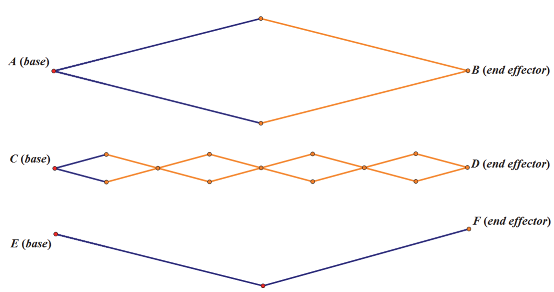

Figure 17: Comparing Different Arm Designs

Another downside of pantographs is that linkages cannot freely pass through each other. All motors are in different Z space. For the two DoF arm that we built, all linkages are in one of two different Z-spaces. We have a zig-zag pattern where links in Z=1 only connect to links in Z=2 and vise versa. Links cannot pass through each other, which roughly means the pantograph arm cannot extend fully straight. Practically speaking this limitation does not feel too bad. We could resolve this by adding additional Z-levels, though this can be cumbersome in practice. We need to add two additional Z-levels for the 2DoF pantograph. This number increases as we increase the number of motors.

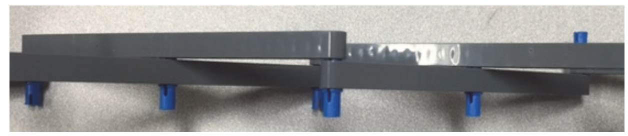

Figure 18: Current Design has 2 Z-levels

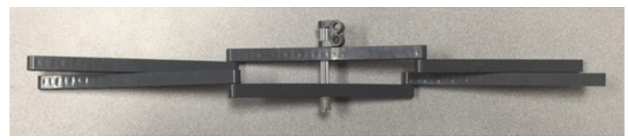

Figure 19: Proposed Passthrough Design with 4 Z-levels

Future Work

There are similarities in the kinematics equations between pantographs and traditional manipulators. One future research question which we intend to pursue is the following: Can the theoretical dynamics of any arm be replicated by a sufficiently complicated pantograph?

The 2 link pantograph has a very similar flavor to the traditional 2 link arm, specifically the kinematic equations. The IK calculations for the pantograph are similar to the IK calculations for the traditional arm, until you need to specify what exactly your angles are. This similarity continues when you add additional linkages in 2D space. Because you can stack motors on top of each other in the Z-dimension, we suspect these theoretical arms can actually be built in 2D space. We modelled a 5 motor 2D arm with both a pantograph and a traditional arm and similar areas can be covered. Intuitively, the bottom linkages of the pantograph are analogous to the traditional linkage.

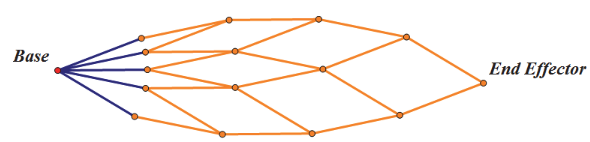

Figure 20: 5 Motor Pantograph Linkage

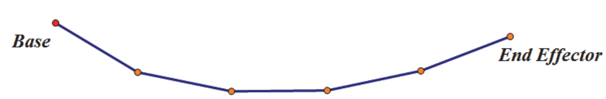

Figure 21: 5 Motor Traditional Linkage

There will be some practical limitations when expanding this comparison to XYZ space. For the arm we built, the motors are effectively at the same point in space. We are examining strategies to 'collocate' motors in 3D space. Additionally, in 3D space, linkages will need to pass through each other. We are looking forward to thinking about these challenges when expanding to 3D space.

We also intend to look at safety and compliance with pantograph arms. Pantographs have pinchpoints on both the inside and the outside of each parallelogram. We will examine materials to wrap the pantograph such that a flexible, protective cover prevents humans from trapping a finger in the mechanism. We will look at movement algorithms that do not require the pantograph to fully extend or retract, reducing the pinchpoint area. Additionally, we will program artificial intelligence safety checks that ensure CoBot knows it is about to knock a door and not move its arm into a human. We may also incorporate verbal warnings as an added precaution.

We will examine different arm placements on the robot. We hope to place many low-cost pantograph arms on the robot. For each arm, we have flexibility to choose the height and angle on the robot, the number of linkages, lengths of the linkages, number of motors, torque of motors, and types of end-effectors. We will experiment with different configuations of multiple arms so that CoBot can perform a variety of tasks without operator assistance.

Conclusion

This work examined extendable pantograph arms, one solution for an inexpensive, modular arm to add to a mobile service robot. Pantographs have many useful characteristics. Pantographs have simple forward kinematics and inverse kinematics, making them relatively easy to understand. Pantographs can compress while not in use. Pantographs have an intuitive footprint - extending out in effectively a line. One potential follow-up, outside of our scope, would be to evaluate whether users have an easier time interacting with pantograph robot arms compared to traditional arms because of this intuitive footprint. Additionally, we hypothesize that this intuitive footprint can be exploited so that cameras can be placed on the pantograph to automatically focus on the end effector. For constrained environments where the only path to reach an object is straight, pantographs can be advantageous; yet, we also acknowledge that in general, they have a larger footprint than traditional arms.

A user can add additional linkages and/or additional motors to pantograph arms to alter the dynamics for intended use. Additionally, a selection of end effectors can be added to pantographs depending on the use case. Pantographs also are advantageous because the majority of the system weight can be mounted on a mobile robot.

In addition to explaining pantograph arms and comparing with traditional arms, we also built a prototype pantograph arm that we intend to attach to CoBot in the next few weeks. We intend to further our understanding of pantographs by examining 3D pantographs. Two distinct 3D pantographs are parallelogram prisms, which we expect to have a flavor similar to the 2D pantographs discussed in this paper, and pantographs based around triangular prisms, which we expect to be similar to delta robots. We intend to design and build a 3D pantograph arm.

Acknowledgements

This research was sponsored by the Defense Advanced Research Projects Agency under grant number FA87501620042. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of any sponsoring institution, the U.S. government or any other entity.

References

Biswas, J., and Veloso, M. 2013. Localization and Navigation of the CoBots over Long-term Deployments. *International Journal of Robotics Research* 32(14):1679–1694.

Circuit Digest. 2015. What are Industrial Manipulators?. http://circuitdigest.com/article/what-are-industrial-manipulators.

Kinova. 2016. Robot arms. http://www.kinovarobotics.com/service-robotics/products/robot-arms/.

Lu, Z.; Chauhan, A.; Silva, F.; and Lopes, L. 2012. A brief survey of commercial robotic arms for research on manipulation. In *Robotics and Applications (ISRA), 2012 IEEE Symposium on*, 986–991.

Mecademic. 2015. What is a parallel robot?. http://www.mecademic.com/What-is-a-parallel-robot.html.

Murray, R. M.; Li, Z.; Sastry, S. S.; and Sastry, S. S. 1994. *A mathematical introduction to robotic manipulation*. CRC press.

Pellegrino, S. 2015. Space Structures Laboratory. http://www.pellegrino.caltech.edu/.

Rosenthal, S.; Biswas, J.; and Veloso, M. 2010. An Effective Personal Mobile Robot Agent Through Symbiotic Human-Robot Interaction. In *Proceedings of AAMAS'10, the Ninth International Joint Conference on Autonomous Agents and Multi-Agent Systems*.

Universal Robots. 2015. UR3 Automate tasks up to 3kgs desktop pick and place. http://www.universal-robots.com/products/ur3-robot/.

van der Geest, A. 2014. double delta triple delta. http://www.vouwbad.nl/dd_press/.

Veloso, M.; Biswas, J.; Coltin, B.; and Rosenthal, S. 2015. CoBots: Robust Symbiotic Autonomous Mobile Service Robots. In *Proceedings of IJCAI'15, the International Joint Conference on Artificial Intelligence*.

Weisstein, E. W. 2016. Parallelogram Law. http://mathworld.wolfram.com/ParallelogramLaw.html.

Wikipedia. 2016. Pantograph. https://en.wikipedia.org/w/index.php?title=pantograph &oldid=697699093. Page Version ID: 697699093.

You, Z. 2015. Research into Deployable Structures. http://www.eng.ox.ac.uk/deployable.