Desktop Manipulation using a Mobipulator

Ammar Husain
General Engineering
University of Illinois at Urbana Champaign

Justin Mann
Mechanical Engineering
University of Illinois at Urbana Champaign

***Abstract*—This paper describes the control policy of a mobipulator. The task is to move desktop objects by integrating manipulation and locomotion of the robot. We describe the engineering design of the robot and discuss the control laws governing its motion. The manipulation of the mobipulator is achieved by setting a dual-differential drive constraint. In this mode the front two wheels act as the "hands" and manipulation paper and the rear wheels are the "feet" that produces locomotion. We derive the equations governing the kinematics of the robot. Following that we discuss the policy implemented to control the robot on a discrete computable set of points. Lastly we describe the engineering build of the mobipulator and the experiments we successfully**

Keywords- Mobile Manipulation; Robotics; Desktop Manipulation; Differential Drive

I. Introduction

A. Mobipulator

This project entails the study of a combination of locomotion and manipulation in a nonholonomic system. A mobipulator is a small concept car, with four independently controlled wheels. The wheels of the car perform the locomotion as well as the manipulation of paper. Fig. 1 is a picture of the mobipulator we built. The car can be simplified to two differential drives, one for the front wheels and one corresponding to the rear wheels. The idea of a mobipulator was initiated at the Manipulation Lab at Carnegie Mellon University by Matt Mason et al [1][2][3]. In this project we built an inexpensive open-loop alternative of the robot.

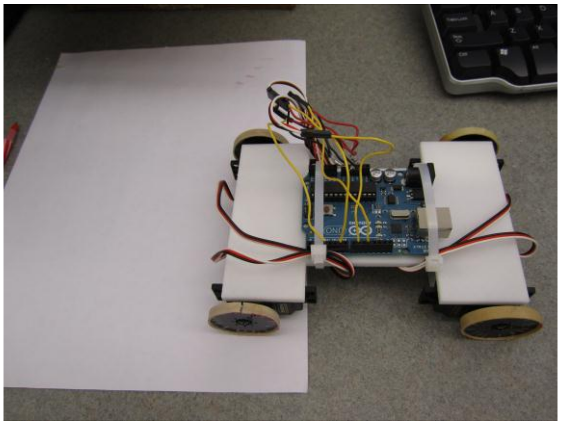

Fig 1- Mobipulator

B. Objective

The goal of this study is to successfully manipulate a sheet of paper on a Desktop from an arbitrary start position to a user specified goal. In formulating a control policy we provide the robot a discrete set of points (P) to goal. Additionally we impose a *dual-differential drive* constraint on the system. In order to satisfy this constraint the front wheels (*hands*) are on the paper and rear wheels (*feet*) are on the desktop at all times. Intuitively, the *hands* manipulate paper and the *feet* perform the locomotion.

II. Related Work

Extensive research has been carried out on mobile manipulators. In this section we discuss similar work that synchronizes locomotion with manipulation in robotic systems. In the mid-1960s Shakey [4], the robot, was a breakthrough in this field and achieved some degree of success in mobile manipulation. It rearranged objects by sensing its environment. This initiated exploration in robotic systems that consolidated locomotion and manipulation, rather than treating them as separate problems. One of the solutions proposed is to attach some sort of manipulating appendage to a moving platform. The JPL Cart [5] and the Stanford Assistant Mobile Manipulation platform [6] are examples of the same. Both these systems accomplish tasks by marrying a mobile platform with manipulation appendages. The distinction in our mobipulator approach is that we propose to integrate both functions into just one system using an elegant and simple robot. The Platonic Beast [7] is an example of this. The robot manipulates a variety of objects with the legs that support its posture and motion.

Nonprehensible manipulation is a similar field that studies manipulation through the functioning of one of the significant systems in the robot. Examples of this are the vibrating plate [8] or the use of a tilting tray for sensorless manipulation. One of the closest studies to the mobipulator is on the manipulation of laminar objects [9], such as cards, by pressing down on them from above. This approach is similar to the dual-diff drive mode of the mobipulator that we propose in the following sections.

Mason et al [1] designed a working prototype of the robot and demonstrated various modes of operation on it, such as the inchworm mode, dual-diff drive, scoot etc. These modes achieved a high degree of success in moving a sheet of desktop paper from one configuration to the other, by utilizing the nonholonomic constraints of the system [2]. Srinivasa et al [3], within the same research lab developed a robust control policy to manipulate paper in the dual-diff drive mode. Quantitative results have so far been established only for this mobipulator mode. Previous works in this section have demonstrated that a fundamental relationship exists between manipulation and locomotion. Integration between the two has produced exciting results. With this project we aim to further explore the possible manipulation and locomotion of paper using a mobipulator in the dual diff drive mode.

III. Kinematic Model

In this section we derive the motion equations governing the locomotion of the mobipulator and the manipulation of paper. The configuration of every point in the environment is described by three parameters as $Q = (x, y, \theta)$. The parameters are defined with reference to the frame mentioned in the superscript of Q .We divide the system into three frames of reference: Desktop, Robot and Paper. The origins of the frames are located at U, R and P respectively as shown in Fig 2. The three physical axes for the frames are represented by $\hat{u}_i$, $\hat{r}_i$ and $\hat{p}_i$ where i $\in$ {1, 2, 3}. The symbols a, b and c are the chassis half length, half width and wheel radius respectively. We treat the system as a set of two differential drive robots and solve for the motion of the mobipulator and paper accordingly. We conform to the following convention through the rest of this section:

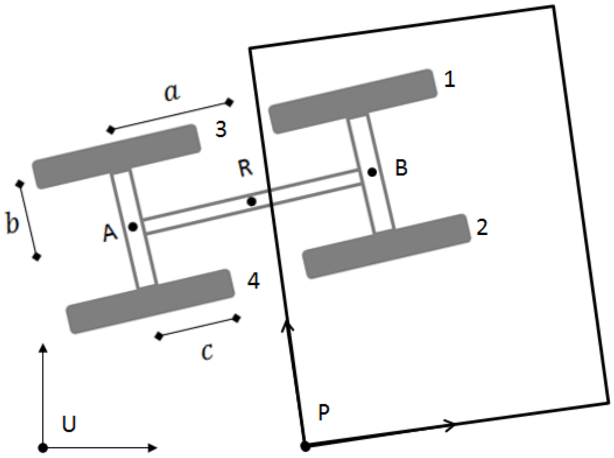

Fig 2- Mobipulator Reference Frames

A. Mobipulator Motion

Point A is located halfway between wheels 3 & 4 on the rear axle of the mobipulator. We can express the velocity of A as:

$$ {}^{U}\dot{Q}_A = v\hat{r}_1 + \omega_R \hat{r}_3 \qquad (1) $$

where U (superscript) = the frame of reference

A (subscript) = the point under consideration

$\dot{Q}$ = change in configuration with respect to time

$\hat{r}$ = expressed in coordinates of frame R

We will follow the above mentioned convention throughout this section. Using the rigid body equation, we then express the velocity of wheel 4 as:

$$ {}^{U}\dot{Q}_4 = {}^{U}\dot{Q}_A + (\omega_R \hat{r}_3 \times b\hat{r}_2) = (v - b\omega_R)\hat{r}_1 + \omega_R \hat{r}_3 \qquad (2) $$

Similarly velocity of wheel 3 is:

$$ {}^{U}\dot{Q}_3 = {}^{U}\dot{Q}_A + (\omega_R \hat{r}_3 \times -b\hat{r}_2) = (v + b\omega_R)\hat{r}_1 + \omega_R \hat{r}_3 \qquad (3) $$

The friction of our wheels guarantees negligible slip and therefore the velocities of the robot's wheels can be expressed as $c\omega_i$ where i $\in$ {1, 2, 3, 4}. Therefore for the rear axle we obtain:

$$ \begin{aligned} v &= \frac{c}{2}(\omega_4 + \omega_3) \\ \omega_R &= \frac{c}{2b}(\omega_4 - \omega_3) \end{aligned} \qquad (4) $$

Applying the rigid body conditions again, we obtain the velocities of the robot center R caused due to the rear axle.

$$ {}^{U}\dot{Q}_R = {}^{U}\dot{Q}_A + \omega_R \hat{r}_3 \times a\hat{r}_1 $$

$$ \Rightarrow {}^{U}\dot{Q}_R = \begin{pmatrix} \frac{c}{2}(\omega_4 + \omega_3) \\ \frac{ac}{2b}(\omega_4 - \omega_3) \\ \frac{c}{2b}(\omega_4 - \omega_3) \end{pmatrix} \begin{pmatrix} \hat{r}_1 \\ \hat{r}_2 \\ \hat{r}_3 \end{pmatrix} \qquad (5) $$

B. Paper Manipulation

In this section, we derive ${}^{U}\dot{Q}_P$, which is the motion of paper in the desktop frame. The motion of the paper is caused due to locomotion of the robot and due to manipulation of paper [14].

$$ {}^{U}\dot{Q}_P = {}^{U}\dot{Q}_R + {}^{R}\dot{Q}_P \qquad (6) $$

where, ${}^{R}\dot{Q}_P = {}^{P}\dot{Q}_R + \omega \times \vec{r}_{RP}$

The front axle of the mobipulator must always be on the paper, in order to satisfy the dual-differential drive constraint. Therefore we may derive ${}^{P}\dot{Q}_R$, which is the motion produced by the front axle, similar to the derivation in subsection *A.*

$$ {}^{P}\dot{Q}_R = -\begin{pmatrix} \frac{c}{2}(\omega_2 + \omega_1) \\ \frac{-ac}{2b}(\omega_2 - \omega_1) \\ \frac{c}{2b}(\omega_2 - \omega_1) \end{pmatrix} \begin{pmatrix} \hat{r}_1 \\ \hat{r}_2 \\ \hat{r}_3 \end{pmatrix} \qquad (7) $$

It must be noted that due to the difference in frictional force, the frame P is moving and therefore we obtain the outer negative sign in Equation (7). The vector $\vec{r}_{RP}$ is expressed in robot frame coordinates as ${}^R x_P \hat{r}_1 + {}^R y_P \hat{r}_2$. The rotation of $\vec{r}_{RP}$ is caused due to all four wheels and is therefore an extension of ω_R from Equation (4).

$$
{}^R\dot{Q}_P = -\begin{pmatrix} \frac{c}{2}(\omega_2+\omega_1) \\ \frac{-ac}{2b}(\omega_2-\omega_1) \\ \frac{c}{2b}(\omega_2-\omega_1) \end{pmatrix}\begin{pmatrix} \hat{r}_1 \\ \hat{r}_2 \\ \hat{r}_3 \end{pmatrix} \quad (8)
$$

$$
+\frac{c}{2b}((\omega_4+\omega_2)-(\omega_3+\omega_1))\hat{r}_3 \times ({}^R x_P \hat{r}_1 + {}^R y_P \hat{r}_2)
$$

Equations 5,6 and 3 may now be simply combined to obtain the motion of the paper. However in order to simplify the solution of wheel angular velocities, Mason et al [1] represent the paper motion as:

$$
{}^U\dot{Q}_P = \sum_{i=1}^{4} f_i \omega_i \text{ where}
$$

$$
f_1 = \begin{pmatrix} \frac{-c}{2} - \frac{c}{2b}\,{}^R y_P \\ \frac{c}{2b}\,{}^R x_P - \frac{ac}{2b} \\ \frac{c}{2b} \end{pmatrix},\quad f_2 = \begin{pmatrix} \frac{-c}{2} + \frac{c}{2b}\,{}^R y_P \\ \frac{-c}{2b}\,{}^R x_P + \frac{ac}{2b} \\ \frac{-c}{2b} \end{pmatrix},
$$

$$
f_3 = \begin{pmatrix} \frac{c}{2} + \frac{c}{2b}\,{}^R y_P \\ \frac{-c}{2b}\,{}^R x_P - \frac{ac}{2b} \\ \frac{-c}{2b} \end{pmatrix},\quad f_4 = \begin{pmatrix} \frac{c}{2} - \frac{c}{2b}\,{}^R y_P \\ \frac{c}{2b}\,{}^R x_P + \frac{ac}{2b} \\ \frac{c}{2b} \end{pmatrix} \quad (9)
$$

The vector f_i represents the direction of motion of paper corresponding to angular velocity in wheel i . The coordinate axes for the kinematic model in this section are $\hat{r}_1, \hat{r}_2 \,\&\, \hat{r}_3$. These may be converted to desktop frame coordinate using a rotation matrix.

$$
{}^U R_R = \begin{pmatrix} \cos\theta_R & -\sin\theta_R & 0 \\ \sin\theta_R & \cos\theta_R & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad (10)
$$

IV. Control Policy

A. *Translation Mode*

In formulating a control policy for the robot paper system, we assume that a planning algorithm would provide the controller a discrete set of points P from the start configuration to the user specified goal configuration. Therefore for a time interval dt we know ${}^U\dot{Q}_P$. We may use Equation 7 to solve for the angular velocities required to reach the next paper configuration. However, we have 3 equations and 4 unknowns. Therefore we simplify our system into pure manipulation and pure locomotion. A corollary of Equation 7, we observe that

$$
f_1(0)+f_2(0)+f_3(1)+f_4(1) = \begin{pmatrix} c \\ 0 \\ 0 \end{pmatrix} \quad \text{and}
$$

$$
f_1(0)+f_2(1)+f_3(0)+f_4(1) = \begin{pmatrix} 0 \\ \frac{ac}{b} \\ 0 \end{pmatrix} \quad (11)
$$

Therefore it is possible to obtain a motion of the paper in all three axes ($\hat{u}_1, \hat{u}_2 \,\&\, \hat{u}_3$) when $\omega_i \in W$, where W is a set of the following angular velocities:

$$
W = \{(0,0,\omega,\omega),(\omega,0,\omega,0),(0,\omega,0,\omega), \\ (0,0,0,\omega),(0,0,\omega,0),(\omega,0,0,0),(0,\omega,0,0)\} \quad (12)
$$

Thus we can establish that this system is Small Time Locally Controllable (STLC). In the simulator we input a finite ω and it picks a configuration from W and calculates the required dt to achieve the next step in the set of discrete points P . It must be noted that in the actual experiments we set $\omega = 0.1$ when $\omega_R \neq 0$ and $\omega = 1$ otherwise. This is because a rotation of the robot produces significant wheel slip and therefore to avoid skidding is run slower.

B. *Control Switch Mode*

Throughout the experiments the entire system is constrained to satisfy dual-differential drive mode. However at certain configurations of the robot and paper, further manipulation is not possible. At this point the robot escapes the Translation mode described in subsection *A* and enters a Control Switch Mode. We broadly classify the two possible scenarios and propose a control policy to escape them.

1) Trigger: A manipulating wheel is at the corner of the paper.

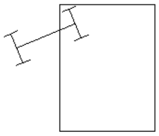

Fig 3- Control Switch # 1

It must be noted that the drift of the paper in *A* is restricted to the set of points P. In order to account for the control switch condition, we the motion of the paper as

$$^{U}\dot{Q}_P(P(t)) = \alpha \dot{P}(t) + \beta \qquad (13)$$

where β is a constant instead restrict drift in the paper when α is set to 0. This essentially means that we drift off the path temporarily and eventually come back, in order to parallel park the mobipulator with respect to the paper.

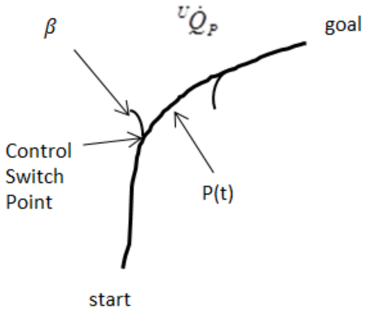

Fig 4- Paper Drift

Below is the pseudo code for the parallel park maneuver:

```
1) for n iterations
2)          set $\omega_i = (\omega, 0, \omega, 0)$
3) end for
4) set $\omega_i = (\omega, \omega, \omega, \omega)$ to center robot
5) for n iterations
6)          set $\omega_i = (0, \omega, 0, \omega)$
7) end for
8) set $\omega_i = (\omega, \omega, \omega, \omega)$ to center robot
```

- In the pseudo code 'n' is the number of iterations that is directly proportional to the desired shift in the robot position.
- Steps 4 and 8, center the robot until the point R is closest to the edge of the paper. This is desired to ensure equal clearance for the manipulating and locomotion wheels.

We successfully implemented this algorithm on the robot simulator as well as the actual robot, obtaining a significant robot shift and negligible overall drift in the paper. In the figure below, red indicates the final position of the robot and paper for a small 'n'.

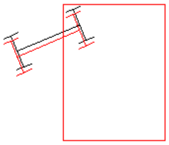

Fig 5- Parallel Parking Maneuver

2) Trigger: A manipulating wheel and a locomoting wheel are on the edge of the paper

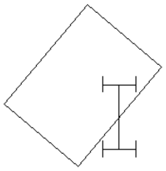

Fig 6- Control Switch # 2

When encountered with this category of switching points, we propose a Sweep maneuver. This motion aligns the angle of the robot to the current paper orientation. The following is the pseudo code that we implemented on the robot:

```
1) While thetaP != thetaR
2)          set $\omega_i = (\omega, 0, 0, 0)$
3)          counter += 1
4) end while
5) set $\omega_i = (\omega, \omega, \omega, \omega)$ to center robot
6) for iterations = 1 to counter
7)          set $\omega_i = (0, 0, \omega, 0)$
8) end for
9) set $\omega_i = (\omega, \omega, \omega, \omega)$ to center robot
```

- The pseudo code produces a clockwise rotation of the paper in step 2, as is required for the condition in Figure (x). An opposite rotation may be produced by setting ω for wheel 2.

The figure below illustrates the initial (black) and final (red) configurations of the robot and paper that we obtained in the simulator.

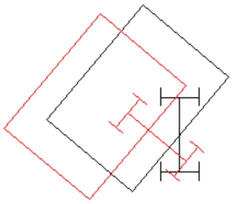

Fig 7- Sweep Maneuver

It can be seen that the final paper configuration is shifted along $\hat{r}_1$ & $\hat{r}_2$. This can be compensated by performing a backward translation followed by a parallel park of the paper. We argue that the Sweep maneuver is more efficient than performing a Parallel Park at this switching point for the following reasons:

i. The robot is now positioned perpendicular to the robot producing easier translation going forward

ii. Multiple parallel parking maneuvers might sometimes be required in order to clear the switching point.

iii. The robot generates significant paper drift in each parallel park motion as compared to a Sweep.

V. Engineering Design

We used the Futaba S3004 Standard servo to drive the four wheels. The servos had to be modified for continuous rotation. This is done by replacing the potentiometer with a pair of equal resistors. We also physically removed a knob on the gear that restricted 360 degree rotation.

The motors were controller using an Arduino Uno. This microcontroller is basically a wrapper around the ATmega328 8-bit microcontroller. The Arduino IDE has embedded libraries to send PWM outputs to drive servos. Since the potentiometer was removed, we manually zeroed out the stopping point for each servo.

A critical component for the mobipulator to function correctly was for the four servos to rotate at the same speed. This was made more difficult because the center value for each servo was different, and thus, a common angular velocity between the servos needed to be determined experimentally. We established a baseline speed for one of the servos, and found the amount of time it took for the servo to rotate 30°, 45°, 60°, 90°, 180° and 360° and stop for both the clockwise and counterclockwise directions. The baseline speed became the standard speed that each of the other three servos was calibrated against. For the other three servos, the PWM output that produces the same amount of rotation per time was found. It must be noted that given the inexpensive servos used, our calibration method was not robust. Although using brushless DC motors or pre-made continuous servo would help increase precision of the robot.

Attached to the end of each of the servos are four plastic 1" wheels, with rubber bands wrapped and glued around the edges to provide a sufficient amount of friction between the wheels and the paper and the table.

The four servos are connected together on an "I" shaped plastic chassis that was approximately 0.2" thick using glue. The chassis was made through rapid prototyping, and served as the base of the mobipulator. The rigidity of the plastic chassis allowed it to support the weight of all four servos evenly without splaying out or warping. The total dimensions of the mobipulator came out to approximately 4" x 6".

Part	Cost ($)
Futaba S3004 Standardservo x 4	15 x 4
1" Plastic Wheels x 4	NA
Plastic chassis	NA
Arduino Uno	30
Total	90

Table 1: Bill of Materials

VI. Experiments

A. MATLAB Simulator

The control policy was formulated and tested through simulations in MATLAB. The purpose of the simulator was also to detect switching points and keep track of current robot and paper configurations. This sensing capability inherent in the simulator was used as a compensation for a lack of visual servoing on the actual robot. Therefore the simulator would output to us the wheel angular velocities that were then incorporated for the physical motions. Currently there was not interface between MATLAB and the Arduino IDE; however that is certainly a possibility.

B. Mobipulator Physical Implementation

Following extensive simulations we implemented our proposed maneuvers on the robot. This section describes the various motions that we performed.

1) Traversal

The first motion that was tested was a simple traversal in the forward and reverse directions of the mobipulator, without any paper. This movement was accomplished by running all four wheels in the same direction, at the same speed, and then doing the same in reverse. The results of this motion corresponded fairly closely with the simulator results.

2) Translation

The second motion was to have the mobipulator pushing a piece of paper as it translates. This was done by having the front two wheels, the "hands" on the paper motionless and translating the rear two wheels. For the most part, this motion worked and the mobipulator was able to push the paper down the table and back. However, some areas of concern came from the excessive paper drift that was evident on certain trial runs. Because of non-uniform support friction forces under the paper and inconsistent servos, when the mobipulator pushes the paper with its front two wheels, the system tends to drift to one side or the other. The amount of drift is largely dependent on how fast the wheels are moving. The faster the wheels, the more drift is evident. We additionally created a rotation in the mobipulator by varying the rear two wheels of the mobipulator, like a differential drive. The motion of the robot and the paper was fairly consistent with our simulated results.

3) *Paper Parallel Park*

In this example, the “legs” of the mobipulator are stationary, and only the front two wheels on the paper move. This ensures that the mobipulator stays still despite all the movement it gives to the paper. The purpose of this movement is to shift the paper directly to the right, perpendicular to the direction of the wheels resulting in a shift approximately 1 cm from its initial position. This is accomplished by running one of the front wheels so that the paper is rotated and then the paper is slid directly forward by running both front wheels together. Finally, the reverse motion is done, running the same one wheel as earlier in reverse, and then sliding the paper directly backward. This accomplishes the small shift to the side, and when this motion is done consecutively, can move the paper from left to right and vice versa. This worked well, and for the most part experienced very little paper drift. This was mainly due to the fact that the mobipulator was not moving relative to the world reference frame. From our work, it seems that the mobipulator wheels slip much more when moving on the table than when moving on the paper.

4) *Robot Parallel Park*

We described the parallel park of the robot in Section IV B Control Switch #1. This maneuver is slightly more complicated because any motion of the robot will cause a motion in the paper. This was accomplished by running two of the wheels on the same side of the robot. This rotates the robot, but also causes a rotation of the paper. Then all four wheels were run in the same direction, translating the robot directly backwards. Finally, the two wheels that weren’t run in the first step are run at the same speeds and for the same duration as the other two. This causes a reverse rotation of the paper and the robot, moving the paper back into its original configuration, and moving the robot back to its original orientation, but shifted slightly over. This motion evidenced much more drift than the parallel parking of the paper due to both the robot and the paper moving at the same time, compounding the imprecisions that occur.

5) *Sweep*

The sweep maneuver is described in Section IV B Control Switch #2. It realigns the robot to the paper’s edge, and then moves the paper back into its original configuration. This is done by running the front-most wheel that is on the paper in reverse so that the two front wheels align with the paper edge. Then, one of the back wheels is driven to slowly turn both the mobipulator and paper so that the paper is on its original heading. For this maneuver, the area that displays the most drift occurs when moving the robot and paper back to its original heading.

From our experimentation and implementation, many observations can be drawn about the differences between the ideal and the real world. It seems that when both the robot and the paper are moved at the same time, there is a lot of drift. This may be due to the fact that when the wheels are on different surfaces, they stop exhibiting the same speed as they did when they were spun on the same surface. Also, there is more drift that is evident when two wheels are translating, like in the pushing paper, as opposed to all four wheels translating in the regular mode.

VII. Future Work

A. *Side-Drag Mode*

In this paper we impose the dual differential drive mode constraint on the control of the robot. Adding another control mode could significantly enhance the abilities of the mobipulator. Mason et al [1] briefly describe a few other modes. Though quantitative results have only been established for the dual differential drive mode. Extending our derivations of the kinematic model from Section III, we propose a model for the ‘Side-Drag Mode’ shown is Fig 8. This mode could be integrated into the existing control policy giving additional flexibility to the robot. For example at Control Switch #1, performing a parallel park maneuver generates significant drift. The robot could instead switch modes to perform a side-drag until other switching points are encountered. In this section we simply present the kinematics of this system.

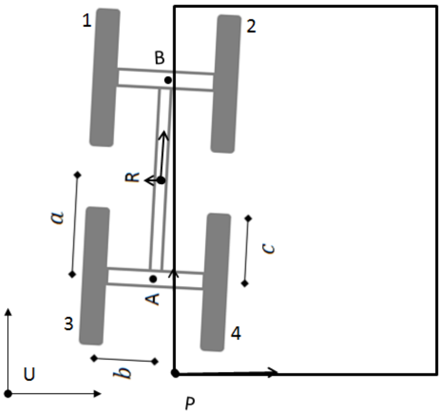

Fig 8- Side-Drag Mode

The wheels for locomotion in this mode are 1 & 3. The equations governing pure translation of the system can then be derived as [12]:

$$ {}^{U}\dot{Q}_R = \begin{pmatrix} \frac{c}{2}(\omega_1 + \omega_3) \\ \frac{ac}{2b}(\omega_1 - \omega_3) \\ \frac{-c}{2b}(\omega_1 - \omega_3) \end{pmatrix} \begin{pmatrix} \hat{r}_1 \\ \hat{r}_2 \\ \hat{r}_3 \end{pmatrix} \tag{13} $$

If we again represent the motion of paper as:

$$ {}^{U}\dot{Q}_P = \sum_{i=1}^{4} f_i \omega_i $$

For the side-drag mode we can derive the wheel vectors f_i as:

$$ f_1 = \begin{pmatrix} \frac{c}{2} - \frac{c}{2b}\,{}^{R}y_P \\ \frac{ac}{2b} + \frac{c}{2b}\,{}^{R}x_P \\ -\frac{c}{2b} \end{pmatrix}, \quad f_2 = \begin{pmatrix} \frac{-c}{2} + \frac{c}{2b}\,{}^{R}y_P \\ \frac{ac}{2b} - \frac{c}{2b}\,{}^{R}x_P \\ \frac{c}{2b} \end{pmatrix}, $$

$$ f_3 = \begin{pmatrix} \frac{c}{2} + \frac{c}{2b}\,{}^{R}y_P \\ \frac{-c}{2b}\,{}^{R}x_P - \frac{ac}{2b} \\ \frac{c}{2b} \end{pmatrix}, \quad f_4 = \begin{pmatrix} \frac{-c}{2} - \frac{c}{2b}\,{}^{R}y_P \\ \frac{c}{2b}\,{}^{R}x_P - \frac{ac}{2b} \\ -\frac{c}{2b} \end{pmatrix} \tag{14} $$

where $\vec{r}_{RP} = {}^{R}x_P\hat{r}_1 + {}^{R}y_P\hat{r}_2$.

B. Path Planning Algorithm

Currently we provide the controller with a discrete set of points from start to goal configurations that could be achieved for a certain time interval when $\omega_i \in W$. However this could be automatically generated given a start and goal position. One possibility of path planning could be the implementation of Reeds-Shepp curves. In Section IV A, we established that our system is STLC. We could consider the entire robot-paper system as a differential drive robot, and therefore treat it as a Reeds-Shepp car [10].

Reeds and Shepp prove that, in an obstacle free world, the optimal path is contained in the given set of path types[11]:

{C|C|C, CC|C, C|CC, CCa|CaC, C|CaCa|C,
C|$C_{pi/2}$SC, CS $C_{pi/2}$|C, C| $C_{pi/2}$S $C_{pi/2}$|C, CSC}

It must be noted that our environment is free of external obstacles, however, in order to maintain dual-differential drive mode the edge of the paper serves as an obstacle. Therefore this path might not be time optimized. Although, given our handling of control switches, we can guarantee to reach the goal condition. The set above will provide us a discrete path to goal. We could translate that into time scaled control inputs $\omega_i \in W$.

C. Visual Servoing

Currently we keep track of objects in our world through a simulator assuming certain initial configurations. Adding visual feedback will help reduce in the error caused in the real-world implementation. Additionally the Arduino does not support encoder feedback, and therefore the motors are running open loop. We could potentially use brushless DC motors that have higher stability.

Acknowledgment

We would like to thank Professor Timothy Bretl and Professor Seth Hutchinson for their invaluable help with the mobipulator. Many of the ideas presented here came from discussions and critiques with them. We would also like to thank Dan Block for his help in building the mobipulator, and Miles Johnson for his help with acquiring the materials that we needed to get started.

References

[1] Matthew T. Mason, Dinesh Pai, Daniela Rus, L. R. Taylor, and Michael Erdmann, "A Mobile Manipulator" in *IEEE International Conference on Robotics and Automation (ICRA '99)*

[2] Siddhartha Srinivasa, Michael Erdmann, and Matthew T. Mason, "Bilateral time-scaling for control of task freedoms of a constrained nonholonomic system", *2003 IEEE International Conference on Robotics and Automation (ICRA '03)*

[3] Siddhartha Srinivasa, Christopher Baker, Elisha Sacks, Grigoriy Reshko, Matthew T. Mason, and Michael Erdmann, "Experiments with nonholonomic manipulation", in *2002 IEEE International Conference on Robotics and Automation (ICRA '02)*

[4] N. Nilsson, "Shakey the robot," *Technical Report 323*, SRI International, 1984.

[5] A. Thompson, "The navigation system of the jpl robot," in *Fifth International Joint Conference on Artificial Intelligence*, pages 749-757, 1977.

[6] O. Khatib, K. Yokoi, K. Chang, D. Ruspini, R. Holmber, A. Casal, and A. Baader. "Force strategies for cooperative tasks in multiple mobile manipulation systems." In G. Giralt and G. Hirzinger, editors, *Robotics Research: The Seventh International Symposium*, pages 333-342, 1996.

[7] D. Pai, R. Barman, and S. Ralph. "Platonic beasts: a new family of multilimbed robots," in *Proceedings 1994 IEEE International Conference on Robotics and Automation*, pages 1019-1025, 1994.

[8] D. Reznik and P. deSantis. "A flat rigid plate is a universal planar manipulator," in *Proceedings of the 1998 IEEE International Conference on Robotics and Automation*, pages 1471-1477, 1998.

[9] A. Bicchi and R. Sorrentino, "Dexterous manipulation through rolling," in *IEEE International Conference on Robotics and Automation*, pages 452-457, 1995.

[10] H. Choset, K. M. Lynch, S. Hutchinson, G. Kantor, W. Burgard, L. E. Kavraki and S. Thrun, "Principles of Robot Motion: Theory, Algorithms, and Implementations", pages 450-453

[11] J.A. Reeds and L.A. Shepp, **"**Optimal paths for a car that goes both forwards and backwards". *Pacific Journal of Mathematics*

[14] Tim Bretl."A quick note on mobipulator kinematics"