CONTENTS

ROCO504
Catch-bot

Tom Queen
School of Computing,
Electronics and Mathematics
Plymouth University
Plymouth, Devon PL4 8AA
Email: xxxx

Daniel Gregory-Turner
School of Computing,
Electronics and Mathematics
Plymouth University
Plymouth, Devon PL4 8AA
Email: xxxx

Demetrius Zaibo
School of Computing,
Electronics and Mathematics
Plymouth University
Plymouth, Devon PL4 8AA
Email: xxxx

***Abstract*—The abstract goes here.**

I. Introduction

This article discusses the development and construction of a Translational Planar 4-cable Cable-Direct-Driven Robot (CDDR) and its applications in catching and throwing.

This demo file is intended to serve as a "starter file" for IEEE conference papers produced under LaTeX using IEEEtran.cls version 1.8b and later. I wish you the best of success.

mds

January 4, 2018

A. Subsection Heading Here

Subsection text here.

1) Subsubsection Heading Here: Subsubsection text here.

II. Design Process

A. Problem

1) Inverse Kinematics: Initially it was thought that the commands to each motor could be generated by looking directly at the returned X and Y coordinates of the tracked objects. To move the gripper upwards, the top two motors should rotate clockwise and the bottom two anticlockwise. To move the gripper to the left, the two left motors should rotate clockwise and the right two anticlockwise. This led to the following kinematic solution:

$$\begin{aligned} M1 &= Y - X \\ M2 &= Y + X \\ M3 &= -Y - X \\ M4 &= -Y + X \end{aligned} \tag{1}$$

For high torque motors, elastic cords and a closed loop between the tracked object and the gripper, this approximation may have been functional. However, it would not have been accurate, slack cords would be common and it would unnecessarily load the motors. When using stepper motors with low-current drivers, this solution caused the steppers to skip if the gripper was directed more than a few centimetres from the centre of the working area. This led to re-evaluation of the kinematic solution as can be seen in section II-D5.

2) Force storage:

3) Camera speed: In order to track the target quickly, a low latency high FPS camera was needed.

4) Motors: Fast motors were needed to allow the gripper to keep track of the target. The working area of the frame measured XXXX 90cm by 75cm. If the gripper was initialised to the centre of the frame and a ball was thrown from three meters away, then the robot would have 0.7 seconds to move from center to the corner of the working area. PROOF

B. Research

1) CDDR: CDDRs are a type of parallel manipulator wherein the end-effector link is supported in parallel by n cables with n tensioning motors [1]. CDDRs can be made lighter, stiffer, safer and more economical than traditional serial robots [2] since their primary structure consists of lightweight, high load-bearing cables. More complex Cable-driven parallel robots can be designed to move in six Degrees of Freedom (DoF) in three dimensional space, but these are beyond the scope of this article.

C. Requirements

D. Solutions

To achieve the required speed and torque, several motor solutions were designed.

1) Micro motors: The first was to use high torque ($300Ncm$) encoded DC motors (E192.24.125). The maximum speed of these motors was 33rpm. In order to achieve the required gripper speed of 1.5m/s a gearbox was needed.

$$\frac{s}{\pi \cdot d \cdot r} = \frac{150}{\pi \cdot 10 \cdot 0.55} = 1 : 8.68 \tag{2}$$

Where d is the spool diameter ($10cm$), s is required cord speed ($150cm/s$) and r is motor speed ($0.55\ RPM$). This ratio would result in a maximum cord velocity of $149.9cm/s$.

2) Stepper motors: Another solution was to use stepper motors. Four SST58D3820 stepper motors were acquired. These motors are specified to hold $7.3Kg.cm$, which drops to $6.5Kg.cm$ at a frequency of 1200 pulses per second (PPS).

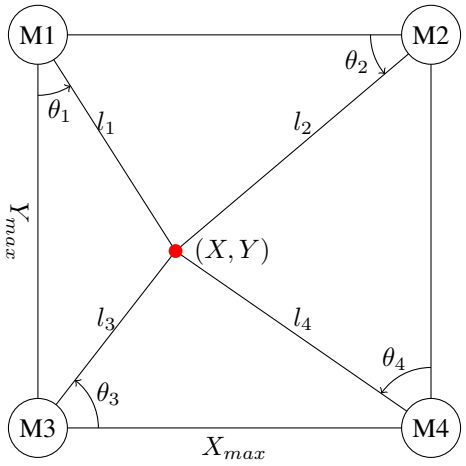


Fig. 1. Kinematic diagram

The motors step size was 1.8, giving a max speed of 6RPS and thus a maximum cord velocity of $188.5cm/s$.

$$\frac{1.8 \cdot 1200}{360} = 6RPS \tag{3}$$

$$6 \cdot \pi \cdot 10 = 188.49cm/s \tag{4}$$

3) Clamps: To address the problem discussed in II-A2, clamps were designed to lock each cord in place. Each clamp consisted of a high-friction surface suspended above a fixed plate via four springs. The cord to be clamped runs between the two surfaces. A dynamixel AX-12 servo is connected the suspended surface with a pulley. When actuated, the servo pulls the suspended surface towards the fixed plate, closing the gap and clamping the cord in place. When released, the four springs push the suspended surface away from the fixed plate, quickly releasing the clamped cord. A clamp was produced for each of the five cords, one for each corner of the frame and one for the throwing motor.

4) Camera: The Sony Playstation3 eye camera can be purchased second-hand for 50p. This camera can output 187 frames per second (FPS) at a resolution of 320x240, or 60 FPS at a resolution of 640x480. In addition, the camera allows for control of exposure, gain, white balance, saturation and hue shift.

5) Kinematic solution 1: To address the issues discussed in II-A1, the following kinematic model was produced. Since no rotational motions and no moment resistance are required at the end-effector, our kinematic model can be simplified to assume that all cables meet at a point [2] in the centre of our end-effector.

Inverse Kinematics:

$$\cos(\theta_3) = \frac{{X_{max}}^2 + {l_3}^2 - {l_4}^2}{2 * X_{max} * l_3} \tag{5}$$

$$X = l_3 \sin(\theta_3) \tag{6}$$

$$Y = l_3 \cos(\theta_3) \tag{7}$$

Or, without trigonometry:

$$\begin{aligned} X &= \left(\frac{{X_{max}}^2 + {l_3}^2 - {l_4}^2}{2 * X_{max} * l_3}\right) = \frac{X_{max}}{2} + \frac{{l_3}^2 - {l_4}^2}{2 * X_{max}} \\ Y &= \left(\frac{{Y_{max}}^2 + {l_4}^2 - {l_1}^2}{2 * Y_{max} * l_3}\right) = \frac{Y_{max}}{2} + \frac{{l_4}^2 - {l_1}^2}{2 * Y_{max}} \end{aligned} \tag{8}$$

Forward Kinematics:

$$\begin{aligned} l_1 &= \sqrt{(X)^2 + (Y_{max} - Y)^2} \\ l_2 &= \sqrt{(X_{max} - X)^2 + (Y_{max} - Y)^2} \\ l_3 &= \sqrt{(X)^2 + (Y)^2} \\ l_4 &= \sqrt{(X_{max} - X)^2 + (Y)^2} \end{aligned} \tag{9}$$

E. Prototype

1) Hardware: These solutions led to the development of the initial prototype. This robot used four stepper motors to drive the gripper. Each motor was driven by a HST-8325B stepper driver with control signals generated on a teensy3.2 running the arduino firmware.

2) Software: The teensy was using the rosserial library to receive the targets CofM from the computer. The teensy then offset the coordinate system so that pixel 0,0 was in the centre of the image. The teensy kept track of all motor positions, and when a new CofM arrived the current gripper position was calculated from the motor positions using the forward kinematics described by equation 9 in II-D5. The program generates motor commands designed to move the gripper so that the centre of the camera image contains the CofM of the target. This is known as visual servoing. The x and y error between the centre of the camera image and the target is calculated, generating a vector that points towards the target. This vector is translated into the change in length of each cord required to center the gripper over the tracked target using the inverse kinematics described by equation 8 in II-D5. The desired gripper location is checked to see if it has left the bounds of the working area. If it has, movement in the direction of the axis which has been breached is set to zero. Finally, motor speeds are calculated from the desired changes in lengths as is described in equation 12 in section II-G1.

F. Further problems

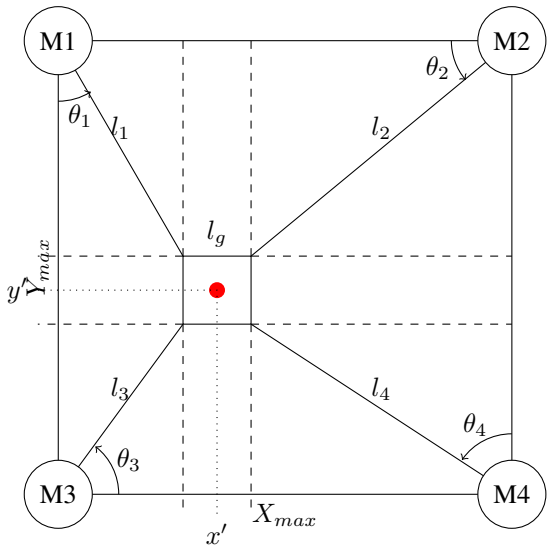
1) Gearbox: A gearbox ratio of 1:8.68 was chosen as can be seen in II-D1. If a 500g gripper were to be raised along either vertical wall of the workspace, the output of the gearbox raising it would experience $5Kg.cm$ of force. Transferring this force through the gearbox, a 1cm input gear would experience $43.4Kg.cm$ of force. By making the length of each gear inversely proportional to its accumulated speed multiplier, the stress felt by the input gear could be spread out over the rest of the gears. However, as this would have more than doubled the size of our original gearbox (SIZESXXXXXXXXXXX), the decision was made to use stepper motors instead.

2) Stepper Drivers: Four HST-8325B were purchased to drive the SST58D3820 stepper motors. The HST-8325B is designed to deliver up to 3.5A RMS. The drivers current can be limited down to 0.8A RMS in eight steps. The performance in the SST58D3820 datasheet is rated for when the motors are being supplied with 2.0A per phase. From this we can see that the stepper motor requires 250mA per phase more than the driver can provide. An issue arose

drivers were connected to the stepper motors and the current to the driver was measured, the stepper driver

3) Skipping steps: The prototype was assembled with SST58D3820 stepper motors and HST-8325B drivers. When the robot was switched on the motors could barely hold the weight of the gripper when stationary. An experiment was performed where commands were sent to the robot instructing the gripper to move to the top corner and then return to the centre, with cord lengths measured before and after. It was found that requesting any movement of the gripper resulted in the stepper motors skipping steps, often resulting in the gripper dropping tens of centimetres over short movements. Further investigation showed that the stepper drivers were drawing minimal current, and would overheat and shutdown if their current limit was set to more than half of what they were rated for. Even with a current limit set at 2.0A RMS, each stepper never drew more than 1.0A. This issue could have been partly mitigated if the motors were encoded.

4) Kinematics: Initially the kinematic solution considered the gripper as a point. This worked for preliminary testing, but soon proved to be a problem when the limited torque of the stepper motors required equal tension on all cords at all times. This led to the development of the following kinematic model, which considered the gripper as a square.



Inverse Kinematics:

$$X = \frac{X_{max}}{2} + \frac{{l_3}^2 - {l_4}^2}{2\left(X_{max} - l_g\right)}$$
$$Y = \frac{Y_{max}}{2} + \frac{{l_3}^2 - {l_4}^2}{2\left(Y_{max} - l_g\right)} \tag{10}$$

Forward Kinematics:

$$\begin{aligned} l_1 &= \sqrt{\left(X - \frac{l_g}{2}\right)^2 + \left(Y_{max} - Y + \frac{l_g}{2}\right)^2} \\ l_2 &= \sqrt{\left(X_{max} - X + \frac{l_g}{2}\right)^2 + \left(Y_{max} - Y + \frac{l_g}{2}\right)^2} \\ l_3 &= \sqrt{\left(X - \frac{l_g}{2}\right)^2 + \left(Y - \frac{l_g}{2}\right)^2} \\ l_4 &= \sqrt{\left(X_{max} - X + \frac{l_g}{2}\right)^2 + \left(Y - \frac{l_g}{2}\right)^2} \end{aligned} \tag{11}$$

5) Motor velocities: For the tension to remain equal on all cords during motion of the gripper, the motors must turn at different rates. For example: if the gripper starts in the centre of the frame and moves upwards, the top two lengths will shorten and the bottom two cords will get longer. To maintain a uniform velocity of the gripper, the top two motors must slow down and the bottom two must speed up.

6) Motors:

7) Arduino:

G. Redesign

1) Motor velocities: To address the issue raised in II-F5 a simple speed scaling system was devised. This algorithm was called once per main loop after the required changes in lengths are calculated. The algorithm takes in the requested changes in length of each cord and divides each change in length by the largest. The relative speeds of each motor are then scaled by the global speed scalar.

$$\vec{l} = \begin{bmatrix} l_1 \\ l_2 \\ l_3 \\ l_4 \end{bmatrix} \qquad x = \max\left(\vec{l}\right)$$
$$x = \max \vec{l}$$
$$\vec{s} = \frac{g}{x} \cdot \vec{l} \tag{12}$$

Where g is the global speed scalar and s contains the speed of each motor.

2) Servos: Due to the points mentioned in II-F3, the decision was made to change the motors to Dynamixel MX-64 servos. The MX-64 can supply $0.6Kg.cm$ at 62 RPM, giving a top gripper speed of $0.32m/s$ which is considerably slower than our requirement. The MX-64 servos communicate

over either a RS-485 or a TTL bus and can be daisy-chained if required. A single dynamixel servo on a bus allowed us to send 60 control packets per second. Due to the nature of the communication protocols involved in the dynamixel bus (clarify), the rate at which commands could be sent to each servo were reduced with each successive servo added to the daisy chain. Initially, the four servos responsible for translating the end-effector were daisy-chained to a single dynamixel bus, resulting in a command rate of 7Hz. This was increased to 15Hz by increasing the baud rate of the servos from 57.6kb/s to 1Mb/s. The reduced control rate led to the decision to use separate busses for each time-critical motor, returning the command rate to 60Hz. A separate bus was used for all five clamps, and a final sixth bus was used to actuate the gripper as the uncertainty of the command latency to this servo was required to be as low as possible in order to successfully pre-empt and act upon the target landing in the gripper.

3) Processor: As servos were to be used instead of stepper motors, a microcontroller was no longer required for the control of the end-effector. Software previously running on the Teensy (forward/inverse kinematics, boundary checking etc...) was ported to a C++ program on a laptop using ROS on Ubuntu.

H. Final solution

III. Implementation

A. Software

Frames enter the object tracking node at a rate of 60FPS. The object tracker performs a series of filters in different colour spaces on the image before calculating its centre of mass. This coordinate is published to the kinematic controller 17.5ms after the frame enters the object tracker. Upon entering the kinematic controller, the coordinate frame is offset so that the center of the camera image is now at pixel 0,0. The required change in length of each cord is then calculated and set (via the method described in II-F4). From the desired changes in length, the required speed of each motor is calculated and set. The kinematic controller node then calculates the current gripper position in order to calculate the changes in length for the next loop. Figure 3 shows the high-level software flow diagram of the robot.

B. Frame

C. Kinematics

D. Grippers

E. Clamps

IV. Experiments

A. Method

B. Results

C. Discussion

V. Overall Discussion

VI. Conclusion

The conclusion goes here.

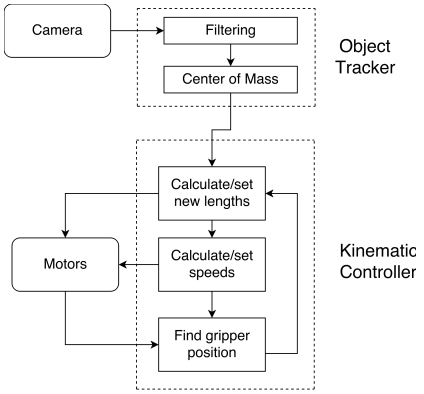


Fig. 2. High-level software flow diagram.

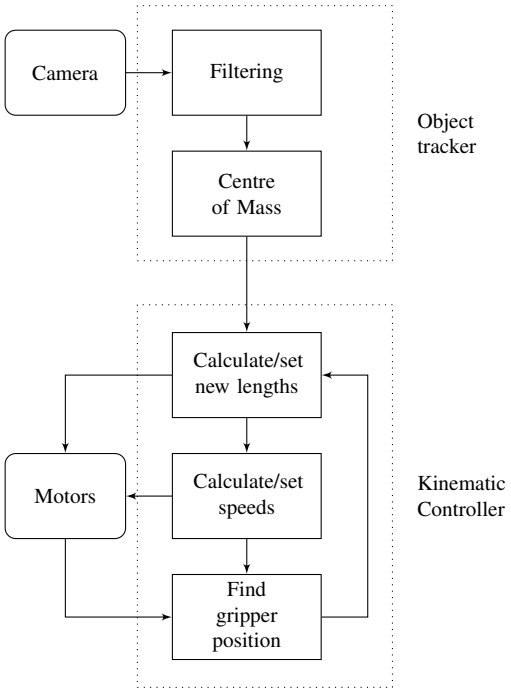


Fig. 3. High-level software flow diagram.

Acknowledgment

The authors would like to thank...

References

[1] S.-R. Oh and S. K. Agrawal, "Cable suspended planar robots with redundant cables: Controllers with positive tensions," *IEEE Transactions on Robotics*, vol. 21, no. 3, pp. 457–465, 2005.

[2] R. L. Williams II and P. Gallina, "Translational planar cable-direct-driven robots," *Journal of Intelligent and Robotic Systems*, vol. 37, no. 1, pp. 69–96, May 2003. [Online]. Available: https://doi.org/10.1023/A:1023975507009

References

[1] Homer J. Simpson. *Mmmmm...donuts.* Evergreen Terrace Printing Co., Springfield, SomewhereUSA, 1998

[2] ROSserial Wiki *Mmmmm...donuts.* Evergreen Terrace Printing Co., Springfield, SomewhereUSA, 1998
@onlinerosserial, author = Misc, title = Rosserial wiki, date = 30/12/17, url = http://wiki.ros.org/rosserial,

[3] H. Kopka and P. W. Daly, *A Guide to LaTeX*, 3rd ed. Harlow, England: Addison-Wesley, 1999.
[4] G. Eason, B. Noble, and I. N. Sneddon, "On certain integrals of Lipschitz-Hankel type involving products of Bessel functions," Phil. Trans. Roy. Soc. London, vol. A247, pp. 529–551, April 1955.
[5] J. Clerk Maxwell, A Treatise on Electricity and Magnetism, 3rd ed., vol. 2. Oxford: Clarendon, 1892, pp.68–73.
[6] I. S. Jacobs and C. P. Bean, "Fine particles, thin films and exchange anisotropy," in Magnetism, vol. III, G. T. Rado and H. Suhl, Eds. New York: Academic, 1963, pp. 271–350.
[7] K. Elissa, "Title of paper if known," unpublished.
[8] R. Nicole, "Title of paper with only first word capitalized," J. Name Stand. Abbrev., in press.
[9] Y. Yorozu, M. Hirano, K. Oka, and Y. Tagawa, "Electron spectroscopy studies on magneto-optical media and plastic substrate interface," IEEE Transl. J. Magn. Japan, vol. 2, pp. 740–741, August 1987 [Digests 9th Annual Conf. Magnetics Japan, p. 301, 1982].
[10] M. Young, The Technical Writer's Handbook. Mill Valley, CA: University Science, 1989.