CARBON DANCE THEATRE PRESENTS

SCIENCE PER FORMS

OCTOBER 25 (7:30PM)
OCTOBER 27 (7:30PM)
OCTOBER 28 (2:30PM)

CHRIST CHURCH NEIGHBORHOOD HOUSE
20 NORTH AMERICAN STREET, PHILADELPHIA
[OFF OF MARKET STREET AND 2ND AVENUE]

GENERAL, $20 SENIOR, $15 STUDENT & DANCEPASS HOLDERS
CARBON DANCE THEATRE PRESENTS

SCIENCE PER FORMS

OCTOBER 25 (7:30PM)
OCTOBER 27 (7:30PM)
OCTOBER 28 (2:30PM)

CHRIST CHURCH NEIGHBORHOOD HOUSE
20 NORTH AMERICAN STREET, PHILADELPHIA
(2ND AVENUE)

$25 GENERAL, $20 SENIOR, $15 STUDENT & DANCEPASS HOLDERS

SYMPOSIUM AT BRYN MAWR COLLEGE: OCTOBER 26 (2:30PM)

FOR MORE INFORMATION VISIT: WWW.CARBONDANCETHEATRE.ORG

BRYN MAWR
MASCHER SPACE
PennDesign
THE HACKTORY

Dance UP
GRASP LABORATORY
IMMERSIVE KINEMATICS
Science per Forms

October 25, 27 & 28, 2012
Neighborhood House Theater, Christ Church
Philadelphia, PA
$25 general admission $20 seniors $15 students and
dancepass holders

Showtimes:
Thursday 10/25 @ 7:30PM
Saturday 10/27 @ 7:30PM
Sunday 10/28 @ 2:30PM

for tickets click here

More information on Carbon Dance Theatre's

CARBON DANCE THEATRE creates projects that empower performers and entertain audiences by creating work that is rooted in classical ballet and infused with the collaborative process of theatre.

meredithcarbondancetheatre.com  |  2920 Cambridge Street Philadelphia, PA 19130  |  graphic design: tori lawrence | www.torilawrence.com
<table>
<thead>
<tr>
<th>From</th>
<th>Subject</th>
<th>Date Received</th>
<th>Time</th>
<th>Date</th>
</tr>
</thead>
<tbody>
<tr>
<td><a href="mailto:qiong@seas.upenn.edu">qiong@seas.upenn.edu</a></td>
<td>ik</td>
<td>Oct 16, 2012</td>
<td>5:01 PM</td>
<td></td>
</tr>
<tr>
<td><a href="mailto:shengj@seas.upenn.edu">shengj@seas.upenn.edu</a></td>
<td>Project 1: Team04</td>
<td>Oct 16, 2012</td>
<td>1:14 AM</td>
<td></td>
</tr>
<tr>
<td><a href="mailto:shengj@seas.upenn.edu">shengj@seas.upenn.edu</a></td>
<td>Project 1: Team04 (Resubmit)</td>
<td>Oct 16, 2012</td>
<td>4:44 AM</td>
<td></td>
</tr>
<tr>
<td><a href="mailto:jonghak@seas.upenn.edu">jonghak@seas.upenn.edu</a></td>
<td>PUMA IK : team19</td>
<td>Oct 16, 2012</td>
<td>4:51 AM</td>
<td></td>
</tr>
<tr>
<td>Dalton Banks</td>
<td>PUMA IK : Team 00</td>
<td>Oct 16, 2012</td>
<td>5:00 PM</td>
<td></td>
</tr>
<tr>
<td><a href="mailto:yulu@seas.upenn.edu">yulu@seas.upenn.edu</a></td>
<td>PUMA IK : Team 01</td>
<td>Oct 16, 2012</td>
<td>5:02 PM</td>
<td></td>
</tr>
<tr>
<td>Thomas Koletschka</td>
<td>PUMA IK : Team 02</td>
<td>Oct 16, 2012</td>
<td>5:19 AM</td>
<td></td>
</tr>
<tr>
<td>Jonathan Balloch</td>
<td>PUMA IK : Team 05</td>
<td>Oct 16, 2012</td>
<td>5:23 PM</td>
<td></td>
</tr>
<tr>
<td>chao qu</td>
<td>PUMA IK : Team 06</td>
<td>Oct 16, 2012</td>
<td>5:30 PM</td>
<td></td>
</tr>
<tr>
<td>chao qu</td>
<td>PUMA IK : Team 06</td>
<td>Oct 16, 2012</td>
<td>5:50 PM</td>
<td></td>
</tr>
<tr>
<td><a href="mailto:amirand@seas.upenn.edu">amirand@seas.upenn.edu</a></td>
<td>PUMA IK : Team 07</td>
<td>Oct 16, 2012</td>
<td>5:59 PM</td>
<td></td>
</tr>
<tr>
<td>Alex Sher</td>
<td>PUMA IK : Team 08</td>
<td>Oct 16, 2012</td>
<td>4:25 PM</td>
<td></td>
</tr>
<tr>
<td>Zhongqi Yue</td>
<td>PUMA IK : Team 08 Resubmission</td>
<td>Oct 16, 2012</td>
<td>5:21 PM</td>
<td></td>
</tr>
<tr>
<td>Vivian Chu</td>
<td>PUMA IK : Team 09</td>
<td>Oct 16, 2012</td>
<td>4:58 PM</td>
<td></td>
</tr>
<tr>
<td>Michael Lo</td>
<td>PUMA IK : Team 10</td>
<td>Oct 16, 2012</td>
<td>3:14 PM</td>
<td></td>
</tr>
<tr>
<td>James Yang</td>
<td>PUMA IK : Team 12</td>
<td>Oct 16, 2012</td>
<td>3:24 PM</td>
<td></td>
</tr>
<tr>
<td>Shaqjun ZHU</td>
<td>PUMA IK : Team 13</td>
<td>Oct 16, 2012</td>
<td>5:07 PM</td>
<td></td>
</tr>
<tr>
<td>Jesse E. Morzel</td>
<td>PUMA IK : Team 14</td>
<td>Oct 16, 2012</td>
<td>4:58 PM</td>
<td></td>
</tr>
<tr>
<td>Collin Boots</td>
<td>PUMA IK : Team 15</td>
<td>Oct 16, 2012</td>
<td>4:03 PM</td>
<td></td>
</tr>
<tr>
<td>Dieter Neckermann</td>
<td>PUMA IK : Team 16</td>
<td>Oct 16, 2012</td>
<td>4:10 PM</td>
<td></td>
</tr>
<tr>
<td>Sawyer Brooks</td>
<td>PUMA IK : Team 17</td>
<td>Oct 16, 2012</td>
<td>5:03 PM</td>
<td></td>
</tr>
<tr>
<td><a href="mailto:siyuanz@seas.upenn.edu">siyuanz@seas.upenn.edu</a></td>
<td>PUMA IK : Team 20</td>
<td>Oct 16, 2012</td>
<td>12:23 PM</td>
<td></td>
</tr>
<tr>
<td>Romaine Waite</td>
<td>PUMA IK : Team 21</td>
<td>Oct 16, 2012</td>
<td>4:58 PM</td>
<td></td>
</tr>
<tr>
<td>Tyler Barkin</td>
<td>PUMA IK : Team 21</td>
<td>Oct 16, 2012</td>
<td>6:50 PM</td>
<td></td>
</tr>
<tr>
<td>Hardik Gupta</td>
<td>PUMA IK : Team 22</td>
<td>Oct 16, 2012</td>
<td>4:39 PM</td>
<td></td>
</tr>
<tr>
<td>Alex Jose</td>
<td>PUMA IK : Team 23</td>
<td>Oct 16, 2012</td>
<td>4:21 PM</td>
<td></td>
</tr>
<tr>
<td>Hang</td>
<td>PUMA IK : Team 24</td>
<td>Oct 16, 2012</td>
<td>4:41 PM</td>
<td></td>
</tr>
<tr>
<td>Takashi Furuya</td>
<td>PUMA IK : Team 26</td>
<td>Oct 16, 2012</td>
<td>3:45 PM</td>
<td></td>
</tr>
<tr>
<td>吴迪</td>
<td>PUMA IK : Team 27</td>
<td>Oct 16, 2012</td>
<td>2:29 PM</td>
<td></td>
</tr>
<tr>
<td>Charlie Xu</td>
<td>PUMA IK : Team 28</td>
<td>Oct 16, 2012</td>
<td>12:11 AM</td>
<td></td>
</tr>
<tr>
<td><a href="mailto:qiangeng@seas.upenn.edu">qiangeng@seas.upenn.edu</a></td>
<td>PUMA IK : Team 28</td>
<td>Oct 16, 2012</td>
<td>3:23 PM</td>
<td></td>
</tr>
<tr>
<td>Brian C. Lee</td>
<td>PUMA IK : Team 28</td>
<td>Oct 16, 2012</td>
<td>4:44 PM</td>
<td></td>
</tr>
<tr>
<td>Brian C. Lee</td>
<td>PUMA IK : Team 29</td>
<td>Oct 16, 2012</td>
<td>4:52 PM</td>
<td></td>
</tr>
<tr>
<td><a href="mailto:golas@seas.upenn.edu">golas@seas.upenn.edu</a></td>
<td>PUMA IK : Team 30</td>
<td>Oct 16, 2012</td>
<td>4:28 PM</td>
<td></td>
</tr>
<tr>
<td>Robert Parajon</td>
<td>PUMA IK : Team 31</td>
<td>Oct 16, 2012</td>
<td>10:01 AM</td>
<td></td>
</tr>
<tr>
<td>Gabrielle Merritt</td>
<td>PUMA IK : Team 32</td>
<td>Oct 16, 2012</td>
<td>4:52 PM</td>
<td></td>
</tr>
<tr>
<td>Fiona Strain</td>
<td>PUMA IK : Team 33</td>
<td>Oct 16, 2012</td>
<td>2:42 PM</td>
<td></td>
</tr>
<tr>
<td>Leslie Callaghan</td>
<td>PUMA IK : Team 34</td>
<td>Oct 16, 2012</td>
<td>3:53 PM</td>
<td></td>
</tr>
<tr>
<td><a href="mailto:tianyud@seas.upenn.edu">tianyud@seas.upenn.edu</a></td>
<td>Fwd: test_puma_ik_team11 (Latest code)</td>
<td>Oct 14, 2012</td>
<td>8:24 PM</td>
<td></td>
</tr>
</tbody>
</table>
Project 1: PUMA Light Painting
Now that you did your inverse kinematics solution, it's time to do light painting. This assignment is due by 5:00 p.m. on Thursday, October 25. Your team must submit this assignment and get it to work correctly before you will be allowed to do the next part of the project (working with the robot). Submissions after the deadline will be penalized, but not as harshly as for individual homework assignments.

Your task is to write a MATLAB program that moves the PUMA's LED around in space to create a lovely light painting (long exposure image).

You should use our PUMA simulator (v1) to test your light painting code. As shown at right, it creates an animation of the PUMA and leaves colored markers in the air so you can see how your creation looks. After you download the simulator, run demo.m to see how it works. Read pumasim_manual_v1.pdf to learn more about the simulator's interface. Please post on Piazza if you are confused about any aspect of the simulator or if you find any bugs.

Submission

1. Start an email to meam520@seas.upenn.edu
2. Make the subject PUMA Simulation: Team 00, replacing 00 with your team number.
3. Attach all of your correctly named MATLAB files to the email. It should be puma_light_painting_teamXX.m, where XX is your team number, plus any additional files you may have created, also named according to this convention.
4. In the body of the email, explain the status of your submission. If you are submitting a new version of your IK with this assignment, state that in the email.
5. Send the email.
6. Wait for a response from the teaching team about whether your code is ready to run on the robot.
RUNNING THE PUMA

The basic work flow when using the PUMA 260 arm is

1. Make sure the Emergency Stop (E-stop) button is engaged (pressed down).

2. Call pumaStart('Hardware', 'on', 'Delay', 10), where the number following delay is the minimum allowable time, in milliseconds, between calls to pumaServo. This may be set to any value above 0.5ms. This will display a warning that the PUMA will return to the home position. Ensure that the workspace is clear and manually move the PUMA closer to the home position if you think it may hit the table or an object.

3. Type 'y' or 'yes' then hit Enter to continue.

4. Release the E-stop by pulling up on the button, at which time the PUMA will return to the home position.

5. In a separate MATLAB process, call startFrameBuffer to begin capturing video from the webcam.

6. Make a light painting, using pumaServo to command the robot to move. Remember to call pumaLEDOn to enable the LED and use pumaLEDSet to select the color.

7. Call stopFrameBuffer to finish capturing video.

8. Return the PUMA to the home position, if possible.

9. Call pumaStop to disable the controller.

10. Engage the E-stop by pressing the button down.

11. Use makeVideoAndImage to create long-exposure picture and video. Optionally, you may wish to save the image files to a different location using saveImagesToFolders before starting a new video.

There are several other things to keep in mind:

- Do NOT use the clear all command once the PUMA has been initialized before calling pumaStop, otherwise MATLAB will crash.

- Test the video capture before running your whole light painting.
%% Puma Simulator Demo

% Initialization
% Close all figure windows.
close all

% Clear all global variables. You should do this before calling any PUMA
% functions.
clear all

% Move the cursor to the top of the command window so new text is easily
% seen.
home

%% Demo 1

% Open figure 1 and clear it.
figure(1); clf

% Initialize the PUMA simulation
pumaStart;
pumaStart;

% turn on the LED
pumaLEDOn;

disp('Demo 1: Use pumaMove to make the arm jump to any desired location')
disp('Note, this works only in simulation')
disp('Press ENTER to start')
title('Press ENTER to start')
pause;
pumaLEDSet(1,0,0); % set the LED to red
pumaMove(0, 0, 0, 0, 0, 0);
disp('Press ENTER to move the robot')
title('Press ENTER to move the robot');
Confirmed Midterm Date
Thursday, November 8, in class

~ email KJK if you have a severe conflict ~
Velocity
Kinematics

Slides created by
Jonathan Fiene
How do the velocities of the joints affect the linear and angular velocity of the end-effector?

These quantities are related by the **Jacobian**, a matrix that generalizes the notion of an ordinary derivative of a scalar function.

**Jacobians** are useful for planning and executing smooth trajectories, determining singular configurations, executing coordinated anthropomorphic motion, deriving dynamic equations of motion, and transforming forces and torques from the end-effector to the manipulator joints.
explore how changes in joint values affect the end-effector movement

could have \textbf{\textit{N}} joints, but only \textbf{\textit{six}} end-effector velocity terms (xyzpts)

The \textbf{\textit{Jacobian}} matrix lets us calculate how joint velocities translate into end-effector velocities (depends on configuration)

look at it in two parts - position and orientation

\[
\begin{align*}
v^0_n &= J_v \dot{q} \\
\omega^0_n &= J_\omega \dot{q}
\end{align*}
\]

How do we calculate the position Jacobian?
\[ \dot{p} = J_p(q) \dot{q} \]

**Jacobian matrix**

\[ J_p = \begin{bmatrix} \frac{\delta x}{\delta q_1} & \frac{\delta x}{\delta q_2} & \cdots & \frac{\delta x}{\delta q_n} \\ \frac{\delta y}{\delta q_1} & \frac{\delta y}{\delta q_2} & \cdots & \frac{\delta y}{\delta q_n} \\ \frac{\delta z}{\delta q_1} & \frac{\delta z}{\delta q_2} & \cdots & \frac{\delta z}{\delta q_n} \end{bmatrix} \]

**Prismatic**

\[ J_{v_i} = z_{i-1} \]

**Revolute**

\[ J_{v_i} = z_{i-1} \times (o_n - o_{i-1}) \]
A Use for the Position Jacobian

\[ v_n^0 = J_v \dot{q} \]

What joint velocities should I choose to cause a desired end-effector velocity? (inverse velocity kinematics)

\[ \dot{q} = J_v^{-1} v_n^0 \]

This works only when the Jacobian is square and invertible (non-singular).

SHV 4.11 explains what to do when the Jacobian is not square: rank test (v is in range of J)
use \( J^+ \) (right pseudoinverse of J)
when the robot has extra joints, there are many solutions
Singularities

Singularities are points in the configuration space where infinitesimal motion in a certain direction is not possible and the manipulator loses one or more degrees of freedom.

Mathematically, singularities exist at any point in the workspace where the Jacobian matrix loses rank.

A matrix is singular if and only if its determinant is zero:

$$\det(J) = 0$$
Position Singularities : Planar RR

\[
\begin{align*}
\mathbf{d}_2^0 &= \begin{bmatrix} a_2 c_{12} + a_1 c_1 \\ a_2 s_{12} + a_1 s_1 \\ 0 \end{bmatrix} \\
(x, y) &= \mathbf{d}_2^0
\end{align*}
\]

\[
\mathbf{J} = \begin{bmatrix} -a_1 s_1 - a_2 s_{12} & -a_2 s_{12} \\
 a_1 c_1 + a_2 c_{12} & a_2 c_{12} \end{bmatrix}
\]

\[
\text{det}(\mathbf{J}) = \, ?
\]

\[
\text{det}(\mathbf{J}) = a_1 a_2 (c_1 s_{12} - s_1 c_{12})
\]

When does \( \text{det}(\mathbf{J}) = 0 \)?
\[
\text{det}(\mathbf{J}) = 0 \text{ when } \theta_2 = 0
\]

Is that the only time?

\[\text{No} \ldots \text{ det}(\mathbf{J}) = 0 \text{ when } \theta_2 = \ldots, -2\pi, -\pi, 0, \pi, 2\pi, \ldots\]

Any other times? \( \text{det}(\mathbf{J}) = 0 \text{ when } a_1 = 0 \text{ or } a_2 = 0 \)
For $\theta_2 = 0$

The Jacobian collapses to have linearly dependent rows

$$J_{\theta_2=0} = \begin{bmatrix}
-a_1 s_1 - a_2 s_1 & -a_2 s_1 \\
a_1 c_1 + a_2 c_1 & a_2 c_1
\end{bmatrix}$$

This means that actuating either joint causes motion in the same direction.
Questions ?
explore how changes in joint values affect the end-effector movement

could have \textbf{N joints}, but only \textbf{six} end-effector velocity terms (xyzpts)

The \textbf{Jacobian} matrix lets us calculate how joint velocities translate into end-effector velocities (depends on configuration)

look at it in two parts - position and orientation

\[ v_n^0 = Jv \dot{q} \quad \quad \quad \omega_n^0 = J_\omega \dot{q} \]

How do we calculate the orientation Jacobian?
And now, angular velocity

\[ \omega = J_\omega(q) \dot{q} \]
A Note about Notation

\( \omega_{i,j}^{k} \)

this is the angular velocity of frame \( j \)
with respect to frame \( i \),
expressed in frame \( k \)

SHV 4.1 gives a good explanation of angular velocity for fixed-axis rotation. SHV 4.2-4.5 go into greater detail.
The Angular Velocity of Connected Rigid Bodies

\[\omega_{0,1} = 0 \hat{x}_0 + 0 \hat{y}_0 + \dot{\theta}_1 \hat{z}_0\]
\[\omega_{1,2} = 0 \hat{x}_1 + 0 \hat{y}_1 + \dot{\theta}_2 \hat{z}_1\]
\[\omega_{1,2} = R_1^0 \omega_{1,2}^1\]
\[\omega_{0,2} = \omega_{0,1} + R_1^0 \omega_{1,2}^1\]
\[= 0 \hat{x}_0 + 0 \hat{y}_0 + (\dot{\theta}_1 + \dot{\theta}_2) \hat{z}_0\]

\[\omega_{0,n} = \sum_{i=1}^{n} R_{i-1}^0 \omega_{i-1,i}^{i-1}\]
\[\omega_{0,n} = \sum_{i=1}^{n} (R_{i-1}^0 \hat{z}) \dot{\theta}_i\]

Note: this holds for revolute joints only (by definition, a prismatic joint cannot create angular velocity)
And now, for that Jacobian!

$$\omega_{0,n}^0 = \sum_{i=1}^{n} \rho_i (R_i^0) \dot{\theta}_i$$

$$\rho_i = \begin{cases} 0 & \text{for prismatic} \\ 1 & \text{for revolute} \end{cases}$$

$$\omega_{0,n}^0 = \begin{bmatrix} \rho_1 \hat{z} & \rho_2 R_1^0 \hat{z} & \rho_2 R_2^0 \hat{z} & \cdots & \rho_n R_{n-1}^0 \hat{z} \end{bmatrix} \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \vdots \\ \dot{\theta}_n \end{bmatrix}$$

$$\omega = J_\omega(q) \dot{q}$$
The Jacobian is easily constructed from the manipulator’s forward kinematics.

What do you need from the forward kinematics?
4.6.3 Combining the Linear and Angular Velocity Jacobians

As we have seen in the preceding section, the upper half of the Jacobian $J_v$ is given as

$$ J_v = [J_{v_1} \ldots J_{v_n}] $$ \hspace{1cm} (4.56)

in which the $i^{th}$ column $J_{v_i}$ is

$$ J_{v_i} = \begin{cases} z_{i-1} \times (o_n - o_{i-1}) & \text{for revolute joint i} \\ z_{i-1} & \text{for prismatic joint i} \end{cases} $$ \hspace{1cm} (4.57)

The lower half of the Jacobian is given as

$$ J_\omega = [J_{\omega_1} \ldots J_{\omega_n}] $$ \hspace{1cm} (4.58)

in which the $i^{th}$ column $J_{\omega_i}$ is

$$ J_{\omega_i} = \begin{cases} z_{i-1} & \text{for revolute joint i} \\ 0 & \text{for prismatic joint i} \end{cases} $$ \hspace{1cm} (4.59)
Singularities are points in the configuration space where infinitesimal motion in a certain direction is not possible and the manipulator loses one or more degrees of freedom.

Mathematically, singularities exist at any point in the workspace where the Jacobian matrix loses rank.

A matrix is singular if and only if its determinant is zero:

$$\det(J) = 0$$
For a 6-DOF manipulator with a spherical wrist, we can decouple the determination of singular configurations into two simpler problems.
$$J = [J_{\text{arm}} \mid J_{\text{wrist}}]$$

(the book calls this $J = [J_P \mid J_O]$)

$$J = [J_{\text{arm}} \mid J_{\text{wrist}}] = \begin{bmatrix}
\frac{J_{11}}{J_{21}} & \frac{J_{12}}{J_{22}}
\end{bmatrix}$$

$$J_{\text{wrist}} = \begin{bmatrix}
z_3 \times (o_6 - o_3) & z_4 \times (o_6 - o_4) & z_5 \times (o_6 - o_5)
z_3 & z_4 & z_5
\end{bmatrix}$$

if we choose $o_4 = o_5 = o_6$

$$J_{\text{wrist}} = \begin{bmatrix}
0 & 0 & 0
z_3 & z_4 & z_5
\end{bmatrix}$$

$$J = \begin{bmatrix}
\frac{J_{11}}{J_{21}} & 0
\end{bmatrix} \quad \det(J) = \det(J_{11}) \det(J_{22})$$
\[ \det(J) = \det(J_{11}) \det(J_{22}) \]

\[ J_{22} = \begin{bmatrix} z_3 & z_4 & z_5 \end{bmatrix} \]

Singular when any two wrist axes align

\[ z_3 \perp z_4 \quad z_4 \perp z_5 \]

\[ z_3 \text{ can become } \parallel z_5 \]

\[ \theta_5 = 0, \pi \text{ are singular configurations} \]
For a specific configuration, the Jacobian scales the input (joint velocities) to the output (body velocity)

\[ \xi = J(q)\dot{q} \]

If you put in a joint velocity vector with unit norm, you can calculate in which direction and how fast the robot will translate and rotate.

If the Jacobian is full rank, you can calculate the manipulability ellipsoid.

\[
\frac{\dot{q}}{||\dot{q}||} = J^+\xi \\
||\dot{q}||^2 = \xi^T (JJ^T)^{-1}\xi
\]

If not redundant, manipulability

\[ \mu = |\det(J)| \]
What does the manipulability ellipsoid look like for the planar RR robot?
\[ \mu = |\det(J)| = a_1 a_2 |\sin(\theta_2)| \]

Can be used to tell you where to perform certain tasks.

Also useful for deciding how to design a manipulator.
Soon I will release Homework 4, an individual assignment on Jacobians.

Not sure when it will be due...
Homework 2 and 3 graded