ASSESSMENT 4 - FINAL REPORT

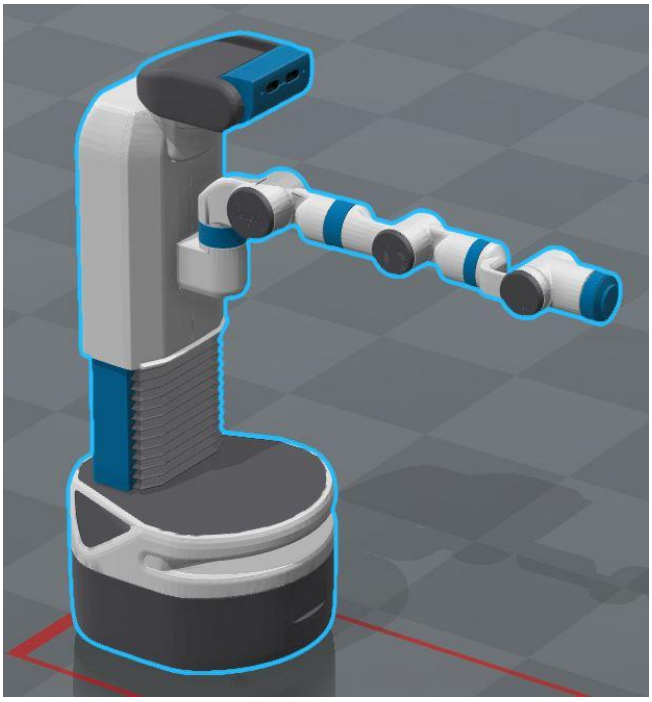
Robotics 41013 Spring 2017

Denis Draca 11710326
Denis.draca@student.uts.edu.au

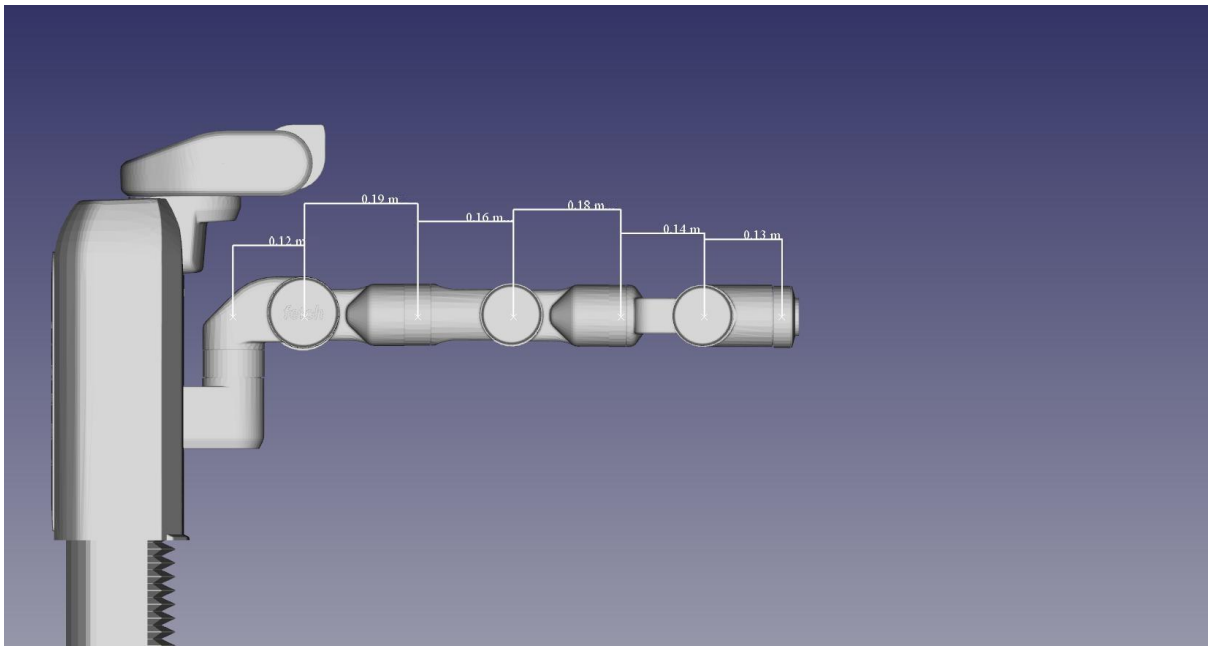
Contents

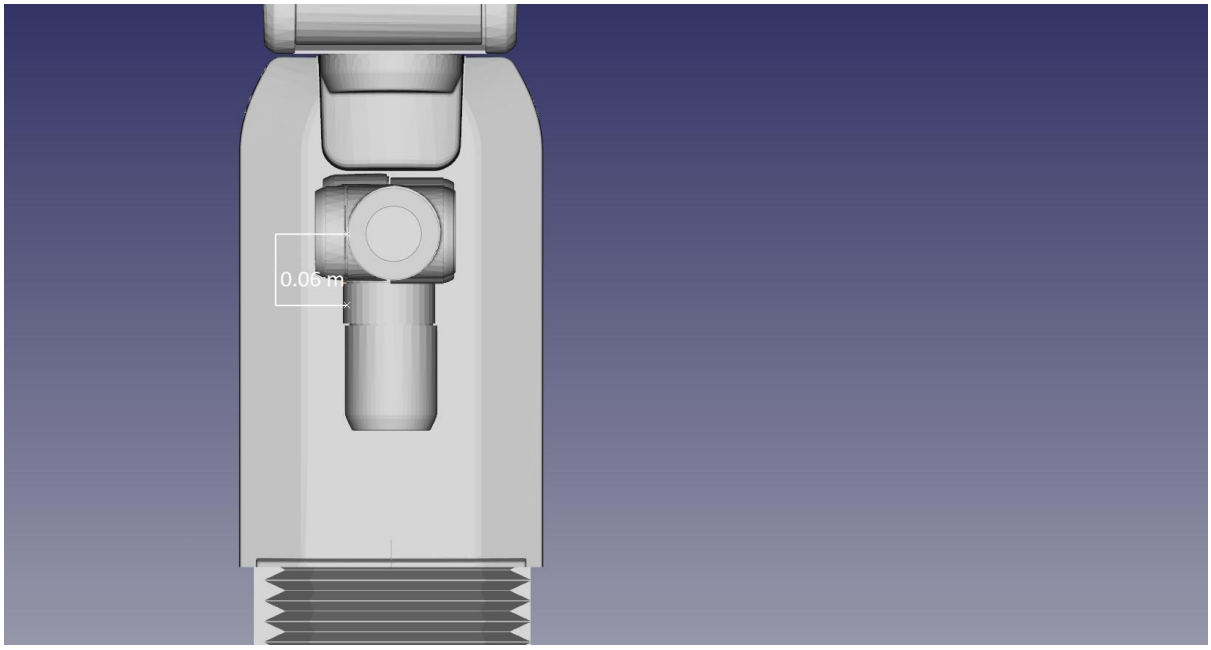
Table of Figures

Blender Assembled Fetch



DH Dimensions





Dh Parameters

Θ_j	d_j	a_j	α_j
Q_1	0.06	0.1170	π/2
Q_2	0	0	-π/2
Q_3	0.352	0	π/2
Q_4	0	0	-π/2
Q_5	0.3215	0	π/2
Q_6	0	0	-π/2
Q_7	0.3049	0	0

End effector not pictured, all length dimension in meters, all angles in radians.

Kinematic List

The transform of a joint, j, relative to the previous can be given by the following transformation matrix:

$$A_j = \begin{bmatrix} \cos(\Theta_j) & -\cos(\alpha_j)*\sin(\Theta_j) & -\sin(\alpha_j)*\cos(\Theta_j) & a_j*\cos(\Theta_j) \\ \sin(\Theta_j) & \cos(\alpha_j)*\cos(\Theta_j) & -\sin(\alpha_j)*\cos(\Theta_j) & a_j*\sin(\Theta_j) \\ 0 & \sin(\alpha_j) & \cos(\alpha_j) & d \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

A_j follows the DH transformation rule:

$$A_j = R_z(\Theta_j) * T_z(d_j) * T_x(a_j) * R_z(\alpha_j)$$

Therefore, the transform of a joint, j, relative to the base can be given by the following transformation:

$$T_j = \left(\prod_{n=1}^{j-1} A_n \right) * A_j$$

Also, it can be expressed more simply as:

$$T_j = \left(\prod_{n=1}^{j} A_n \right)$$

In words, the transform of joint, j, can be expressed by the product of all the previous transformation matrixes.

For Fetch, which has a 7DOF arm, the transform of the end effector can be expressed simply with the following:

$$T_E = \left(\prod_{n=1}^{7} A_n \right)$$

Also, given an initial an initial XYZ location and:

$$XR = roll$$

$$YP = Pitch$$

$$ZY = Yaw$$

The base rotation can be expressed by:

$$R_x = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(XR) & -\sin(XR) \\ 0 & \sin(XR) & \cos(XR) \end{bmatrix}$$

$$R_y = \begin{bmatrix} \cos(YP) & 0 & \sin(YP) \\ 0 & 1 & 0 \\ -\sin(YP) & 0 & \cos(YP) \end{bmatrix}$$

$$R_z = \begin{bmatrix} \cos(ZY) & -\sin(ZY) & 0 \\ \sin(ZY) & \cos(ZY) & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$R = R_x * R_y * R_z$$

The base position can then be expressed as:

$$base = \begin{bmatrix} R & T \\ Z & 1 \end{bmatrix}$$

Where:

$$T = \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}$$

$$Z = \begin{bmatrix} 0 & 0 & 0 \end{bmatrix}$$

Finally, we can say that the position of the end effector relative to the world frame can be given by:

$$T_E = base * \left(\prod_{n=1}^{7} A_n \right)$$

Fetch Model Generation

Joint limits defined based on limits provided on the fetch robotics website.

```
jointLim1 = deg2rad([-92 92]);
jointLim2 = deg2rad([-70 87]);
jointLim3 = [-2*pi 2*pi];
jointLim4 = deg2rad([-129 129]);
jointLim5 = [-2*pi 2*pi];
jointLim6 = deg2rad([-125 125]);
jointLim7 = [-2*pi 2*pi];
```

By importing the fetch urdf using MATLAB's inbuilt capabilities and setting the fetch to be in its home position (arm outstretched). The transform of each joint in the world frame can be requested. Using this information, the transform of each joint relative to each other can easily be extracted. Since in the home position all the joints translate along only one axis in the world frame, the values can easily be converted to DH.

```
l1 = Link('d',0.06,'a',0.1170,'alpha',pi/2, ...
    'offset', 0, 'qlim', jointLim1);
l1.I = body1.Inertia;
l1.r = body1.CenterOfMass;
l1.m = body1.Mass;

l2 = Link('d',0,'a',0,'alpha',-pi/2,...
    'offset',-pi/2, 'qlim', jointLim2);
l2.I = body2.Inertia;
l2.r = body2.CenterOfMass + body1.CenterOfMass;
l2.m = body2.Mass;

l3 = Link('d',0.3520,'a',0,'alpha',pi/2,'offset',0, ...
    'offset', 0, 'qlim', jointLim3);
l3.I = body3.Inertia;
l3.r = body3.CenterOfMass;
l3.m = body3.Mass;

l4 = Link('d',0,'a',0,'alpha',-pi/2,'offset',0, 'qlim', jointLim4);
l4.I = body4.Inertia;
l4.r = body3.CenterOfMass + body4.CenterOfMass;
l4.m = body4.Mass;

l5 = Link('d',0.3215,'a',0,'alpha',pi/2,'offset',0,...
    'offset', 0, 'qlim', jointLim5);
l5.I = body5.Inertia;
l5.r = body5.CenterOfMass;
l5.m = body5.Mass;

l6 = Link('d',0,'a',0,'alpha',-pi/2,'offset',0, 'qlim', jointLim6);
l6.I = body6.Inertia;
l6.r = body6.CenterOfMass + body5.CenterOfMass;
l6.m = body6.Mass;

l7 = Link('d',0.3049,'a',0,'alpha',0,'offset',0, 'qlim', jointLim7);
l7.I = body7.Inertia;
l7.r = body7.CenterOfMass;
l7.m = body7.Mass;
links = [l1 l2 l3 l4 l5 l6 l7];
robot = SerialLink(links, 'name', 'test');
robot.base = robot.base * transl(0,0,0.7260);
```

Where:

```
robot = importrobot('fetch.urdf');
body1 = getBody(robot,'shoulder_pan_link');
body2 = getBody(robot,'shoulder_lift_link');
body3 = getBody(robot,'upperarm_roll_link');
body4 = getBody(robot,'elbow_flex_link');
body5 = getBody(robot,'forearm_roll_link');
body6 = getBody(robot,'wrist_flex_link');
body7 = getBody(robot,'wrist_roll_link');
```

	Body1	Body2	Body3	Body4	Body5	Body6	Body7 $*10^{-3}$
Intertia	0.0125	0.0029	0.0019	0.0025	0.0028	0.0018	0.1000
	0.0388	0.0657	0.0361	0.0430	0.0229	0.0176	0.1122
	0.0308	0.0659	0.0363	0.0434	0.0246	0.0176	0.1122
	0.0007	0.0000	0	0	0	0.0000	0.0000
	-0.0124	0.0000	0	0	0	0.0000	0.0003
	0.0012	-0.0048	-0.0005	-0.0036	0.0045	-0.0002	-0.0005
CenterOfMass	0.0927	0.1432	0.1165	0.1279	0.1097	0.0882	9.5000
	-0.0056	0.0072	0.0014	0.0073	-0.0266	0.0009	0.4000
	0.0564	-0.0001	0	0	0	-0.0001	-0.2000
Mass	2.5587	2.6615	2.3311	2.1299	1.6563	1.7250	135.4000

All Values are in SI units.

Task 1 – Visual Servoing

Simulated Camera on Gripper

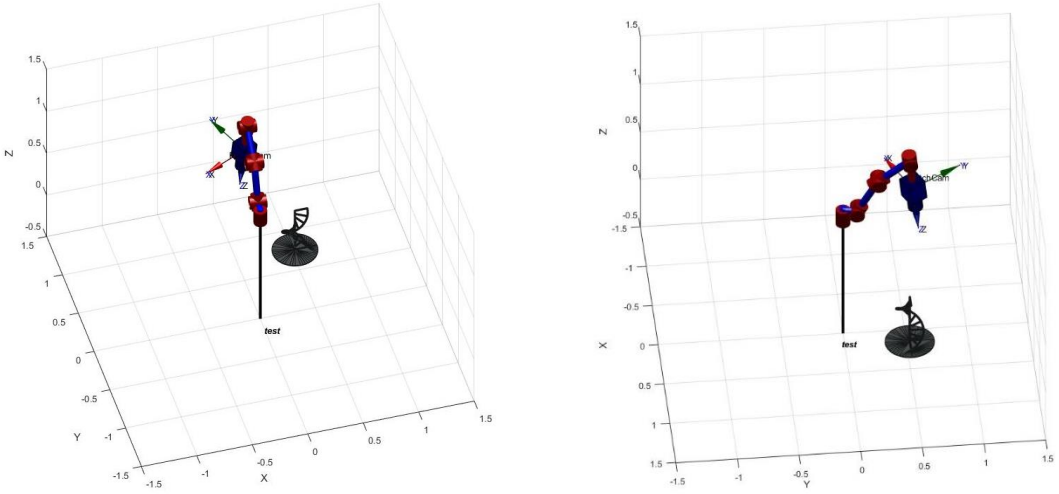


Figure 1:Fetch in Initial Pose with Simulated Camera

We can see above the fetch carrying a simulated camera at its endeffector.

Projection View (Initial View)

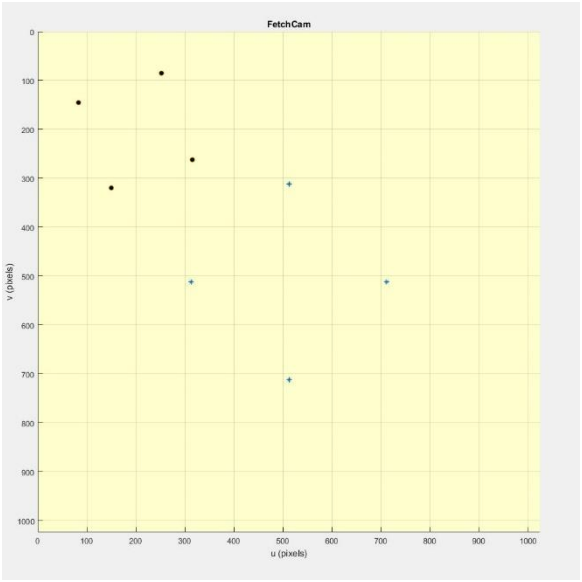


Figure 2: Fetch Initial View

Visual Servoing Results

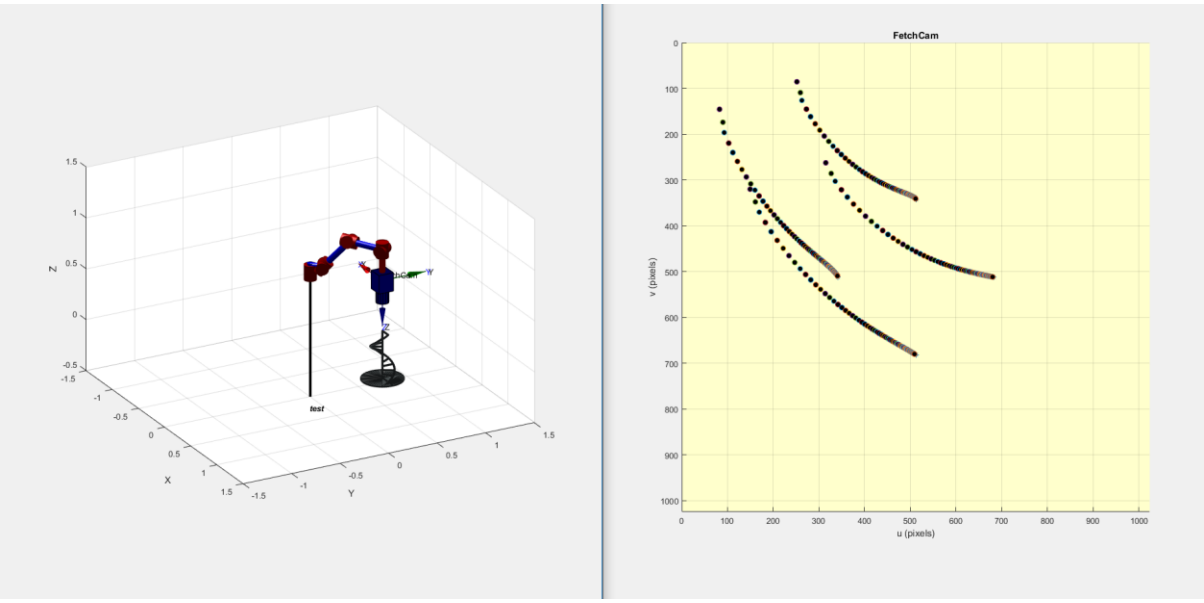


Figure 3: Visual Servoing, Just before Completion

Four points were selected that represent the outer edge of the circular base and these were used to perform the visual servoing task. Using one point would make the solution simpler to implement however it would have a final rotational error as well. As looking at a single point can be done from a variety of poses. However, looking at four can only be done from a single pose, which in this case is straight down.

Process

Refer to Appendix for full code solution. This is simplified inline code.

Initialize Fetch and helix model:

```
bot = drawFetch(startPos);
modelLocation = transl(0.45,  0.5, 0);
hold on;
helixModel = PartLoader('helix3.ply', modelLocation);
W = eye(7);
c = [1 1 1 1 1 1 1];
```

Pick 4 points on the other extremes of the circle base:

```
mod1 = modelLocation * transl(0.15,0,0);
mod2 = modelLocation * transl(-0.15,0,0);
mod3 = modelLocation * transl(0,0.15,0);
mod4 = modelLocation * transl(0,-0.15,0);
```

Select 4 points in the centre of the image view that will represent the goal of the visual servoing task:

```
pS1 = [(512 + 170);512];
pS2 = [(512 - 170);512];
pS3 = [512;(512 - 170)];
pS4 = [512;(512 + 170)];
```

Setup camera

```
cam = CentralCamera('focal', 0.08, 'pixel', 10e-5, ...
'resolution', [1024 1024], 'centre', [512 512],'name', 'FetchCam');
fps = 25;
lambda = 1;
depth = mean (P(1,:));
```

Draw initial State:

```
Tc0= bot.fkine(bot.getpos);
cam.T = Tc0;
cam.plot_camera(P, 'label','scale',0.15);
```

Project views

```
p = cam.plot(P, 'Tcam', Tc0);

%camera view and plotting
cam.clf()
cam.plot(pStar, '*'); % create the camera view
cam.hold(true);
cam.plot(P, 'Tcam', Tc0, 'o'); % create the camera view
pause(2)
cam.hold(true);
cam.plot(P);    % show initial view
```

Iterate until error is less than 3 pixels across all 4 points.

Update Camera View

```
uv = cam.plot(P);

% compute image plane error as a column
e = pStar-uv;   % feature error
e = e(:);
Zest = [];
```

Calculate Image Jacobian

```
J = cam.visjac_p(uv, depth );
```

Calculate Velocity in image frame

```
v = lambda * pinv(J) * e;
```

Due to initial camera starting view and orientation. Some of the velocities didn't line up. This is fixed with the following velocity assignment.

```
v(1) = -v(1);
v(2) = v(2);
v(3) = -v(3);
v(4) = -v(4);
v(5) = v(5);
v(6) = -v(6);
```

Calculate Joint velocities for over actuated arm:

```
J2 = bot.jacob0(bot.getpos);
jV = (inv(W)*J2')*inv(J2*inv(W)*J2')*v;
```

Limit velocities to maximums. To maintain correct trajectory, each joint will be reduced by the same ratio that the over speed joint was reduced by.

```
    for z = 1:length(jV)
        if(jV(z) > maxSpeed(z))
            ratio = maxSpeed(z)/jV(z);
            jV = jV*ratio;
        elseif (jV(z) < -maxSpeed(z))
            ratio = -maxSpeed(z)/jV(z);
            jV = jV*ratio;
        end
    end
```

Update W (refer to calc 'W' function in appendix)

```
W = calcW(W, bot,bot.getpos,c);
```

Update Joints

```
q = q0' + (1/fps)*qp;
bot.animate(q');
```

Update Camera Location

```
Tc = bot.fkine(q);
cam.T = Tc;
drawnow
```

Test Error against threshold:

```
test = (abs(e) < 3);
if(test)
    e
    test
    break
end
```

Task 2, Dynamic torque

Orange = Joint Limit

Blue = Joint Torque

2kg Load

Motion 1

Move up along Z 1m and along X -0.4m from where the object was grasped.

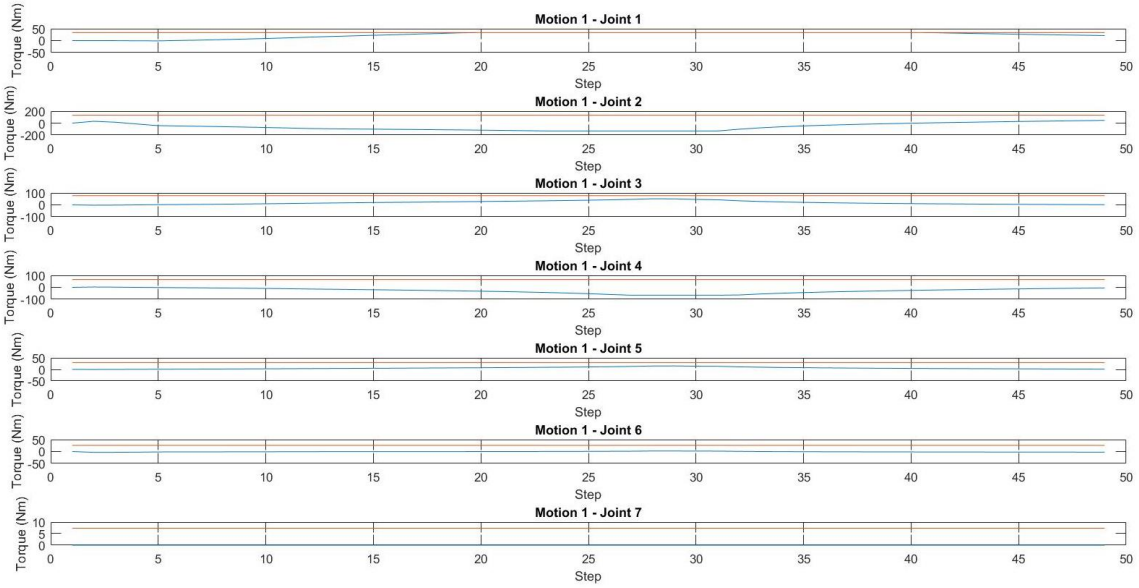


Figure 4: 2kg Load Motion 1

Motion 2

Move along Y -3m from motion 1

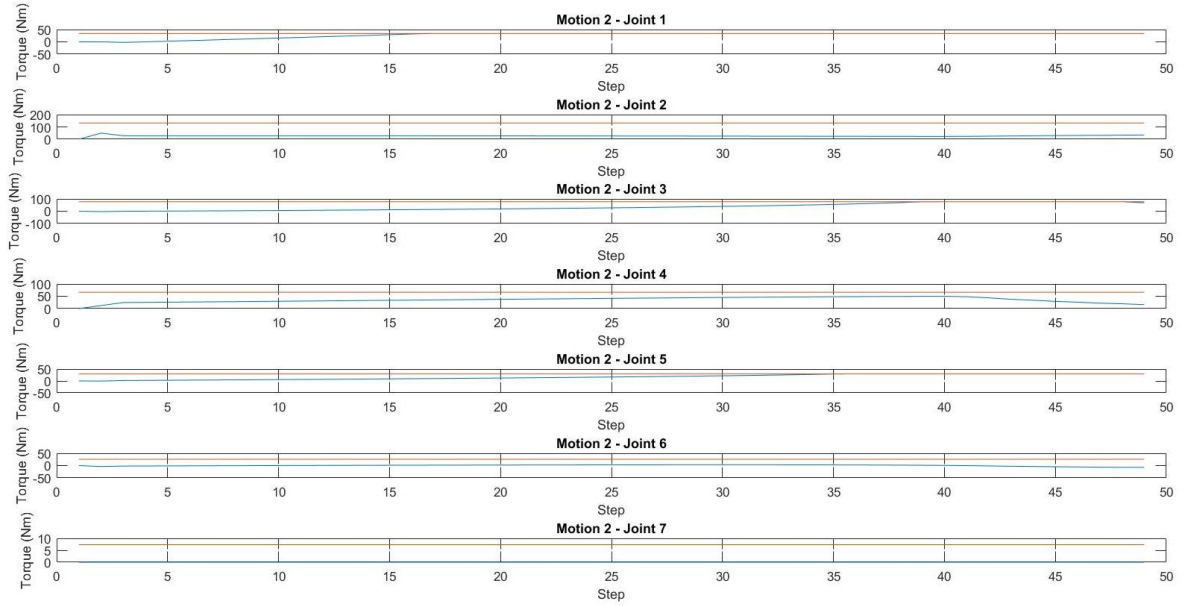


Figure 5: 2kg Load Motion 2

Motion 3

Move down along Z 1m and along X 0.4m to place the object back down.

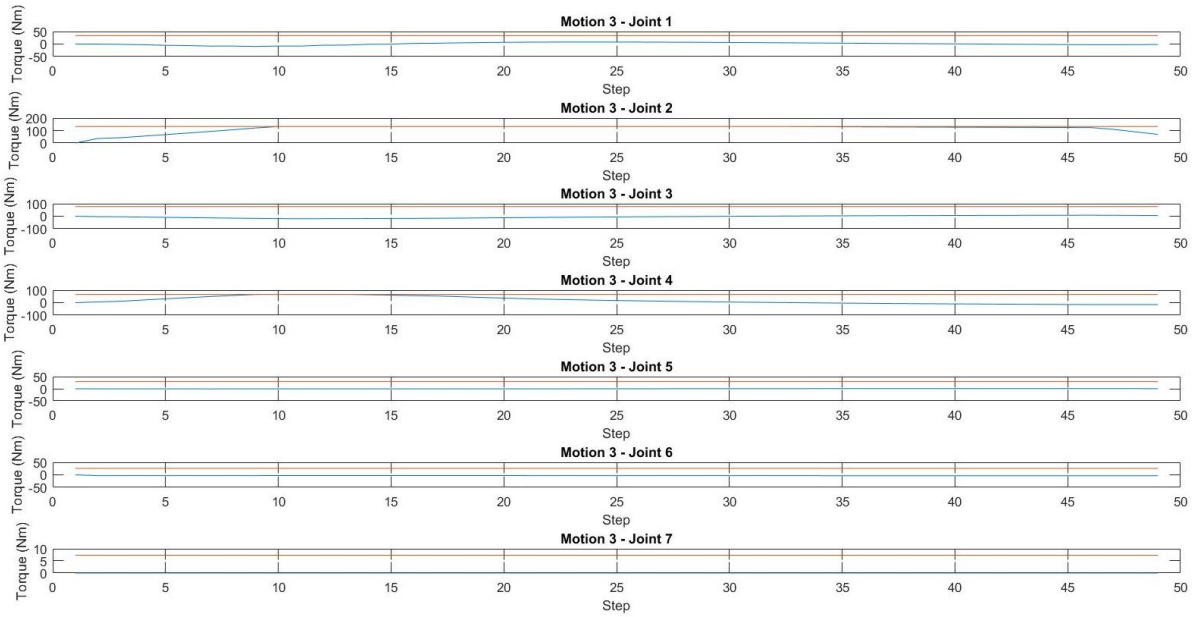


Figure 6: 2kg Load Motion 3

Motion 4

Move up along Z 0.5m from where the object was released. The payload on the endeffector is now 0 aswell.

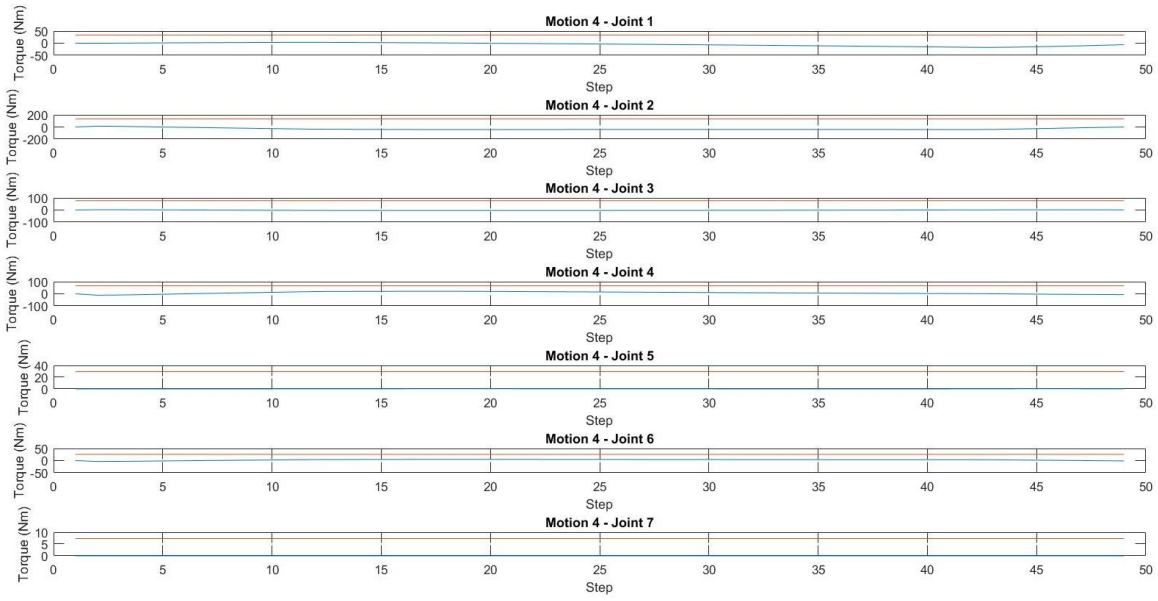


Figure 7: 2kg Load Motion 4

We can see that throughout the movement all the torques remained below the maximum joint torque. We however do see that at times it got fairly close. Motion 2, joint 1 was very close to the joint limit.

5kg Load

Motion 1(Joint Failure Here)

Move up along Z 1m and along X -0.4m from where the object was grasped.

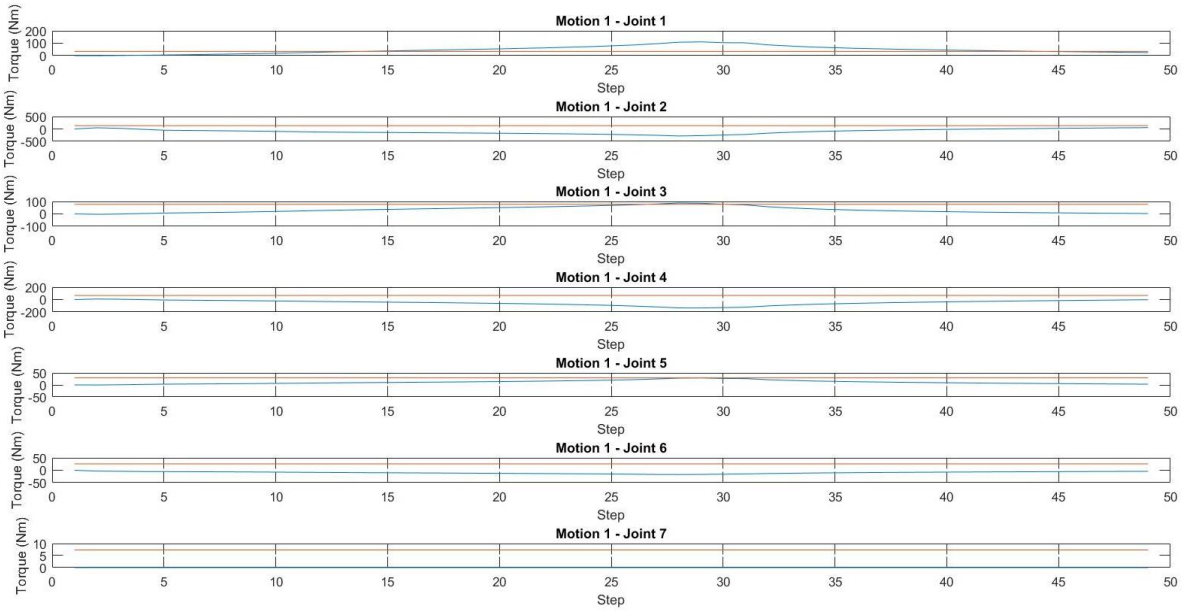


Figure 8: 5kg Load Motion 1

We can see above in motion 1, under joint 1, the arm torque exceeds the maximum joint torque, as such the fetch has failed to perform its task and all the motions have not been completed.

Process

Refer to Appendix for full code solution. This is simplified inline code.

Firstly, the maximum torques are defined:

```
tau_max = [33.82 131.76 76.94 66.18 29.35 25.70 7.36]';
```

Then we initialise the time derivative and the initial state:

```
dt = 1/50;
steps = time/dt;
q1 = bot.getpos;

q(1,:) = q1;
```

Create a trapezoidal velocity profile:

```
x1 = zeros(6,1);
x2 = zeros(6,1);

x1(1:3,1) = T1(1:3,4);
x2(1:3,1) = T2(1:3,4);

x = zeros(length(x1),steps);
s = lspb(0,1,steps);                                % Create interpolation
scalar
for i = 1:steps
    x(:,i) = x1*(1-s(i)) + s(i)*x2;
end
```

Initialise Arrays and set payload on end effector

```
qd = zeros(steps,7);
% Array of joint velocities
qdd = nan(steps,7);
% Array of joint accelerations
tau = nan(steps,7);
% Array of joint torques
% Payload mass (kg)
bot.payload(mass, massPos); % Mass and masspos are passed in arguments
```

Initialize 7DOF RMRC states

```
W = eye(7);
c = [1 1 1 1 1 1 1];
```

Loop through all states

Calculate joint velocities (For over actuated arm):

```
xdot = (x(:,i) - x(:,i-1))/dt;
J = bot.jacob0(q(i - 1,:));
jV = (inv(W)*J')*inv(J*inv(W)*J')*xdot;
```

Limit velocities to maximums. To maintain correct trajectory, each joint will be reduced by the same ratio that the over speed joint was reduced by.

```
    for z = 1:length(jV)
```

```
        if(jV(z) > maxSpeed(z))
            ratio = maxSpeed(z)/jV(z);
            jV = jV*ratio;
        elseif (jV(z) < -maxSpeed(z))
            ratio = -maxSpeed(z)/jV(z);
            jV = jV*ratio;
        end
    end
```

Calculate joint position

```
q(i,:) = q(i-1,:) + (jV*dt)';
```

Update W (Refer to calc 'W' function in Appendix)

```
W = calcW(W, bot,q(i-1,:),c);
```

Calculate joint acceleration

```
qdd(i,:) = (1/dt)^2 * (q(i,:) - q(i,:) - dt*qd(i-1,:));
```

Calculate all affecting factors on joints

```
M = bot.inertia(q(i,:));
C = bot.coriolis(q(i,:),qd(i,:));
g = bot.gravload(q(i,:));
```

Calculate Torque on joints

```
tau(i,:) = (M*qdd(i,:)' + C*qd(i,:)' + g')';
```

If safe torque argument is passed in, limit torques to safe working limits

```
for j = 1:7
      if abs(tau(i,j)) > tau_max(j)
            tau(i,j) = sign(tau(i,j))*tau_max(j);
      end
end
```

Robots Affecting Our World

Why Automation is Different this time

Video

https://www.youtube.com/watch?v=WSKi8HfcxEk

A nice video by Kurzgesagt demonstrating how automation is different this time around. Innovation always got rid of jobs but at the same time innovation always brought better jobs and helped improve the quality of life overall. Now we find that job creation has been stifled. Even with a rising population the total hours worked have remained the same. Had more jobs been created we would have expected this hours to go up with the increasing population. That is why I find this video interesting, it shows how this time around its different.

Article

https://economics.mit.edu/files/12763

Robots and Jobs: Evidence from US Labor Markets

This nice paper agrees with the above video, it shows that the influx of robotics has had a large and negative impact on employment and wages on the average US worker. Considering that here in Australia we have a large mining industry we can expect those jobs to be slowly taken over by robots. We can see the trend in the US markets as one of stagnation and decline in terms of employment and wages. As mentioned in the video the total hours worked has remained the same even with the

Summary

Overall through this video and journal article have shown me that even with a rise in innovation which we used to associate with a rise in the standard of living and the creation of new jobs. With the increasing in automation capabilities with have seen a negative trend in terms of employment and wages. We see that jobs that we thought were once safe such as the creative arts and office jobs have also been slowly taken over by robots. Before it was always the manual labour jobs that were taken, now it's the creative jobs as well. We see in the video that project management software exists (Something I'd never heard of before) that essentially removes the need for management, it also learns from the freelancers it employs and is able automate their jobs as well. This is a scary prospect and shows that no one's jobs is really safe from being automated.

AI and Robotics

Reply to Cameron Knox AI and Robotics Thread

The idea of creating artificial neurons and emulating an organic brain using electronics could lead to a new field of computing and automation. I can see this leading to considerably more intelligent systems ones that are able to perform creatively and think abstractly instead of just crunching numbers. This could lead to human workers being completely pushed out of the workforce. Another possibility for this type of technology is the complete mapping of the human brain. Emulating a functional human brain could lead to completely simulating a person. This could work as a pseudo immortality and could lead to complex ethical questions. If we can completely simulate someone, to be exactly like you, does it have same rights as you? Is it still you or has it become a whole different person?

The article does paint a much more promising future, integrating the younger generation with the future of technology could lead to less people being ignorant of the technology around them then fearing that technology. It could lead to a whole new generation of robotic designers who have had

the moral and ethical questions from a young age, a generation that has thought for a long time about AI, a very well-informed generation. This could bring around the current views of automation

"Robots are increasingly impacting industry, our job and our daily lives in both positive and negative ways"

Unfortunately, when we look at the direction that robotics is taking we see a very bleak picture, particularly when we look at the effects that robotics has already had on society. We can see, particularly when looking at the US labour markets, a decline in employments and a reduction in total wages as more and more repetitive tasks are being automated (Acemoglu and Restrepo, 2017). As this is a continual trend we can expect that this amount of reduction of jobs will not only continue but increase in rate as technology grows exponentially.

However, the future doesn't need to be that bleak. We currently from a world which didn't have this level of technology, we come from a time of great change where we went from no TV's to a world completely connected with the internet. This could be causing the fears and expectations of AI that we currently have. The new generation will grow up with all the wonders of modern technology. By Raising them to become AI literate we can expect them to better tackle all the issues that we are facing today in terms of AI. They will fear the technology less and be able to tackle it with a clear mind and come better equipped to deal with issues that arise with AI (Kandlhofer et al., 2016).

One aspect that the current and future generations will have to deal with is morality within technology. Something recent that will need to be dealt with is morality in autonomous vehicles. "If motor vehicles are to be truly autonomous and able to operate responsibly on our roads, they will need to replicate—or do better than—the human decision-making process" (Lin, 2016). Vehicles will need to make human decision based in morality, if someone must die, how do you decide? This will see an increase in morality considerations made by programmers and engineers. Moving from purely quantitative decision make to qualitative coupled with a deeper understanding of morality to allow systems to make life or death decisions.

Even with these grim considerations, that if dealt with properly will lead to much higher standards of living. We can see in the field of medicine an increase in surgical robots. These do not take over their human counterparts but add to their skills and capabilities. They have been able to improve operation success rates, reduce recovery time and reduce mistakes (Merrifield and Taylor, 2017). Overall, they have improved surgery and paint a bright future where robots work alongside humans improving their lives.

We have seen that the impact of robotics can be grim and there will be many hurdles that future designers will need to consider. But considering them early and not shying away from the issues will allow us to solve these problems. Humans have always had innovation at their door, we have always seen things relatively and have always striven for more. We have always wanted technology to improve and we have adapted to that technology. The same will happen with robotics and AI, we will adapt (Cueni, 2017). When we do, we can see a new era of prosperity and increased standard of living, leading to very positive changes because of robotics.

"how my job may require me to create/integrate/install robots which may result in other people no longer being required to work due to no fault of their own"

Robotics is a fast-growing field that has the potential to do great, but it is already responsible and will continue being responsible for the loss of jobs. Robotics and AI are inherently skilled at doing a singular task very well, as such creative work has been untouched by robots. With the advances in machine learning we can expect robots to take over even those fields. We can see even in manufacturing, the

use of robots has already taken jobs, but the act of programming a robot to do a task has been still left to a human worker. Gúerin et al show us the advances in manufacturing techniques that lead to robots learning and developing on their own. Essentially kicking out the human worker. As my field centres around robotics I can expect to use techniques as the one expressed by Gúerin et al to develop systems that will not require a human touch down the line.

Particularly in the field that I will be moving into, military, we see robotics integrating into the military as well. There has been a shift in the capabilities of robotics and as such has already reduced the operator to robot ratio (Jacoby and Chang, 2008). This is a welcome change as it will reduce the number of war fatalities, at least that of our soldiers. It will however displace a lot of the combat personnel, this is something that I will have to consider down the line. Developing robots that will take these jobs will save lives, but I will be taking jobs from people, can I justify it for the greater good. This is further reinforced by Sharkey (Sharkey, 2008), where particularly America is further expanding its robot combat force. With new and improved systems that will allow them to make kill decisions themselves. Other than the terrifying thought of killer robots, this further shows that even the military isn't safe from automation.

Automation is taking over all aspects of the market, they are starting to infiltrate all aspects of manufacturing, design and creative works (Piva et al., 2017). No matter where you turn robotics and automation will play hand. It is inevitable that every job will be one day automated. This is where our roles play a big part. We are the primary developers of automated products. My field will see me taking jobs in whatever field I end up working in.

Frey and Osborne (Frey and Osborne, 2017) show us the potential different jobs have of being computerised, aka automated. The list is extensive and to no one' surprise, telemarketers are ranked as the most likely to be automated. However, this list has a frightening amount of jobs that have more than 90 % chance of being automated (around 170 fall in this range) and the total amount that are more than 50% likely, 403. This is a staggering amount of potential computerised jobs, one of which being hospitality. Most developed nations move towards service type jobs, which is where we sit. Given the high probability of the hospitality industry being computerised, I can expect to play a role in the reduction of hospitality jobs and the eventually replacement of those already hired.

Given the large potential for automation in almost all field, we can further expect this list to grow and to have even more jobs fall under potential and actual computerisation. No job is safe and they will all fall to automation sooner or later. With that, it is inevitable that one day I will be responsible for the loss of someone's job, through no fault on their own, due to something that I had created. It is inevitable that one day I will take jobs with my designs regardless of what field I work under.

Self-Assessment

Class	Task	SubTask	Mark	Comment
Manipulate the World	Fetch Robot Model	Robot Drawing	1/2	Professionally Drawn and Correct DH parameters but Dimensions Difficult to read
		DH Parameters	2/2	Correct DH Parameters
		Model Creation	3/3	Robot Drawn correctly, with link masses, inertia's and centre of Masses added
		Forward Kinematics	3/3	Transform Relative to Base and Relative to world provided using transformation Matrix
	Visual Servoing	Mount Camera	3/3	Camera Mounted
		Initial Pose	2/3	Camera can see features in top left-hand corner of screen but not all the way in the corner
		Visual Servo	8/10	Task is performed correctly, joint speeds are observed, however, joint limits are not directly observed. Lambda is a bit high (at 1) but the task is performed well.
		Pick Up part	2/4	RMRC is used to move the gripper vertically down (only z velocity component) but joint limits are not observed.
	Dynamic Torque	2kg Scenario	8/10	Completes task, torques saturate and do not allow the joint to over torque. Joint speeds are limited to max speed aswell. Performed at 50 Hz. Joint torques are plotted at the end. Joint limits are not observed however.
		5kg Scenario	8/10	Joint will attempt to lift the part then fail. Joint speeds are limited to max speed aswell. Performed at 50 Hz. Joint torques are plotted at the end. Joint limits are not observed however.
		Section Total	40/50	
Robot Affecting Our World	Link and Article		3/3	Relevant Link and Article provided. Discussed the content
	Critical Comment		2/2	Provided a comment on Cameron Knox's thread
	Integrating Robots		20/20	Content Discussed using 5 relevant articles.
	Causing Job Loss		18/20	Content Discussed using 5 relevant articles. The flow of the discussion was not the best and could use improvement. Seemed to jump a bit
		Section Total	43/45	
		TOTAL	83/95	

Bibliography

Acemoglu and Restrepo, 2017 Acemoglu, D. and Restrepo, P. 2017. Robotsand jobs: Evidence from us labor markets.

Kandlhofer et al., 2016] Kandlhofer, M., Steinbauer, G., Hirschmugl-Gaisch,S., and Huber, P. 2016. Artificial intelligence and computer science in edu-cation: From kindergarten to university. InFrontiers in Education Conference(FIE), 2016 IEEE, pages 1–9. IEEE.

Lin, 2016 Lin, P. 2016. Why Ethics Matters for Autonomous Cars, pages 69–85. Springer Berlin Heidelberg, Berlin, Heidelberg

Merrifield and Taylor, 2017 Merrifield, R. and Taylor, R. (2017). The surgi-cal robot challenge (from the guest editors). IEEE Robotics & AutomationMagazine, 24(2):22–23. Cueni, 2017 Cueni, R. (2017).

Robots Will Take All Our Jobs, pages 35–36.Springer International Publishing, Cham

Gu ́erin et al., 2016] Gu ́erin, J., Gibaru, O., Nyiri, E., and Thiery, S. (2016).Learning local trajectories for high precision robotic tasks: Application tokuka lbr iiwa cartesian positioning. InIECON 2016 - 42nd Annual Conference of the IEEE Industrial Electronics Society, pages 5316–5321.

Jacoby and Chang, 2008 Jacoby, G. A. and Chang, D. J. (2008). Towards command and control networking of cooperative autonomous robotics for military applications (carma). InElectrical and Computer Engineering, 2008.CCECE 2008. Canadian Conference on, pages 000815–000820. IEEE.

Piva et al., 2017 Piva, M., Vivarelli, M., et al. (2017). Is innovation destroyingjobs? firm-level evidence from the eu. Technical report, United Nations University-Maastricht Economic and Social Research Institute on Innovationand Technology (MERIT).

Frey and Osborne, 2017 Frey, C. B. and Osborne, M. A. (2017). The future of employment: how susceptible are jobs to computerisation? Technological Forecasting and Social Change, 114:254–280.

Sharkey, 2008 Sharkey, N. (2008). Cassandra or false prophet of doom: Airobots and war.IEEE Intelligent Systems, 23(4).

Appendix

TorqueFun Function

```
function [ torques, qTotal ] = torqueFun( bot, T1, T2, time, mass, massPos,
torqueSafe, maxSpeed )


tau_max = [33.82 131.76 76.94 66.18 29.35 25.70 7.36]';

dt = 1/50;
steps = time/dt;
q1 = bot.getpos;

q(1,:) = q1;

x1 = zeros(6,1);
x2 = zeros(6,1);

x1(1:3,1) = T1(1:3,4);
x2(1:3,1) = T2(1:3,4);

x = zeros(length(x1),steps);
s = lspb(0,1,steps);                 % Create interpolation scalar
for i = 1:steps
    x(:,i) = x1*(1-s(i)) + s(i)*x2;  % Create trajectory in x-y plane
end

qd = zeros(steps,7);            % Array of joint velocities
qdd = nan(steps,7);             % Array of joint accelerations
tau = nan(steps,7);             % Array of joint torques

bot.payload(mass, massPos);                                                 %
Set payload mass in Puma 560 model: offset 0.1m in x-direction

W = eye(7);
c = [1 1 1 1 1 1 1];
for i = 2:steps-1

    xdot = (x(:,i) - x(:,i-1))/dt;

    J = bot.jacob0(q(i - 1,:));

    jV = (inv(W)*J')*inv(J*inv(W)*J')*xdot;

    for z = 1:length(jV)
        if(jV(z) > maxSpeed(z))
            ratio = maxSpeed(z)/jV(z);
            jV = jV*ratio;
        elseif (jV(z) < -maxSpeed(z))
            ratio = -maxSpeed(z)/jV(z);
            jV = jV*ratio;
        end
    end

    qd(i,:) = jV;
    q(i,:) = q(i-1,:) + (jV*dt)';
    W = calcW(W, bot,q(i-1,:),c);
```

```
    qdd(i,:) = (1/dt)^2 * (q(i,:) - q(i,:) - dt*qd(i-1,:)); % Calculate
joint acceleration to get to next set of joint angles
    M = bot.inertia(q(i,:)); % Calculate inertia matrix at this pose
    C = bot.coriolis(q(i,:),qd(i,:)); % Calculate coriolis matrix at this
pose
    g = bot.gravload(q(i,:)); % Calculate gravity vector at this pose
    tau(i,:) = (M*qdd(i,:)' + C*qd(i,:)' + g')'; % Calculate the joint
torque needed
    torques(i,:) = tau(i,:);
    for j = 1:7
        if abs(tau(i,j)) > tau_max(j) % Check if torque exceeds limits
            tau(i,j) = sign(tau(i,j))*tau_max(j); % Cap joint torque if
above limits
        end
    end

    if(torqueSafe)
        torques(i,:) = tau(i,:);
    end

    qdd(i,:) = (inv(M)*(tau(i,:)' - C*qd(i,:)' - g'))';
end
qTotal = q;
end
```

Plot Torques Function

```
function [] = plotTorques( overallTorque, tauMax )
for i = 1:length(overallTorque)

    figure('Name', ['Motion ', num2str(i)]);

    motionTorques = overallTorque{i};

    for T = 1:length(motionTorques(1,:))
        hold on;
        subplot(length(motionTorques(1,:)),1,T);

        plot(motionTorques(:,T));

        hold on;
        plot([1 length(motionTorques(:,1))], [tauMax(T), tauMax(T)]);

        test = (motionTorques(:,T) > tauMax(T));

        xlabel("Step");
        ylabel("Torque (Nm)");

        title(['Motion ', num2str(i), ' - Joint ', num2str(T)]);

    end
end

end
```

Dynamic Torque Function

```
function [ overallTorque ] = dynamicTorque( bot, helixModel, safeSteps,
mass, time,tau_max,maxSpeed)
    masPos = [0 0 0.1];

    % Move piece up
    T1 = bot.fkine(bot.getpos);
    T2 = T1 * transl(-0.4,0,-1);

    [torqueList, qMatrix] = torqueFun(bot, T1, T2, time, mass, masPos,
safeSteps, maxSpeed);

    overallTorque{1} = torqueList;

    for q = 1:length(qMatrix(:,1))

        Q = qMatrix(q,:);
        T = torqueList(q,:);


        for jT = 1:length(T)

            if(T(jT) > tau_max(jT))
               disp("Joint Failed")
               disp(jT);
               disp(T(jT));
               return;
            end
        end

        bot.animate(Q);
        helixModel.MovePart(bot.fkine(Q)* troty(pi));
        drawnow;
    end

     % Move along y
    T1 = bot.fkine(bot.getpos);
    T2 = T1 * transl(0,-3,0);


    [torqueList, qMatrix] = torqueFun(bot, T1, T2, time, mass, masPos,
safeSteps,maxSpeed);
    overallTorque{2} = torqueList;

    for q = 1:length(qMatrix(:,1))

        Q = qMatrix(q,:);
        T = torqueList(q,:);


        for jT = 1:length(T)

            if(T(jT) > tau_max(jT))
               disp("Joint Failed")
               disp(jT);
               disp(T(jT));
               return;
```

```
            end
        end

        bot.animate(Q);
        helixModel.MovePart(bot.fkine(Q)* troty(pi));
        drawnow;
    end

     % place piece down
    T1 = bot.fkine(bot.getpos);
    T2 = T1 * transl(0.4,0,1);

    [torqueList, qMatrix] = torqueFun(bot, T1, T2, time, mass, masPos,
safeSteps,maxSpeed);
    overallTorque{3} = torqueList;

    for q = 1:length(qMatrix(:,1))

        Q = qMatrix(q,:);
        T = torqueList(q,:);


        for jT = 1:length(T)

            if(T(jT) > tau_max(jT))
               disp("Joint Failed")
               disp(jT);
               disp(T(jT));
               return;
            end
        end

        bot.animate(Q);
        helixModel.MovePart(bot.fkine(Q)* troty(pi));
        drawnow;
    end

     %Move Away
    T1 = bot.fkine(bot.getpos);
    T2 = T1 * transl(0,0,-0.5);


    [torqueList, qMatrix] = torqueFun(bot, T1, T2, time, 0, masPos,
safeSteps,maxSpeed);
    overallTorque{4} = torqueList;

    for q = 1:length(qMatrix(:,1))

        Q = qMatrix(q,:);
        T = torqueList(q,:);


        for jT = 1:length(T)

            if(T(jT) > tau_max(jT))
               disp("Joint Failed")
               disp(jT);
               disp(T(jT));
```

```
                return;
            end
        end

        bot.animate(Q);
        drawnow;
    end

end
```

Draw Fetch Function

```
function [ robot ] = drawFetch( startPos )
robot = importrobot('fetch.urdf');

homeConfig = homeConfiguration(robot);

transform1 = getTransform(robot,homeConfig,'shoulder_pan_link');
transform2 = getTransform(robot,homeConfig,'shoulder_lift_link');
transform3 = getTransform(robot,homeConfig,'upperarm_roll_link');
transform4 = getTransform(robot,homeConfig,'elbow_flex_link');
transform5 = getTransform(robot,homeConfig,'forearm_roll_link');
transform6 = getTransform(robot,homeConfig,'wrist_flex_link');
transform7 = getTransform(robot,homeConfig,'wrist_roll_link');
transform8 = getTransform(robot,homeConfig,'gripper_link');

transformList = [];

transformList(:,:,1) = transform1;
transformList(:,:,2) = transform2;
transformList(:,:,3) = transform3;
transformList(:,:,4) = transform4;
transformList(:,:,5) = transform5;
transformList(:,:,6) = transform6;
transformList(:,:,7) = transform7;
transformList(:,:,8) = transform8;

distanceList = [];

for i = 1:(length(transformList) - 1)
    differenceTransform = transformList(:,:,i+1) - transformList(:,:,i);

    distanceList(1,i) = differenceTransform(1,4);
end

body1 = getBody(robot,'shoulder_pan_link');
body2 = getBody(robot,'shoulder_lift_link');
body3 = getBody(robot,'upperarm_roll_link');
body4 = getBody(robot,'elbow_flex_link');
body5 = getBody(robot,'forearm_roll_link');
body6 = getBody(robot,'wrist_flex_link');
body7 = getBody(robot,'wrist_roll_link');


close all;

jointLim1 = deg2rad([-92 92]);
jointLim2 = deg2rad([-70 87]);
jointLim3 = [-2*pi 2*pi];
jointLim4 = deg2rad([-129 129]);
jointLim5 = [-2*pi 2*pi];
jointLim6 = deg2rad([-125 125]);
jointLim7 = [-2*pi 2*pi];


l1 = Link('d',0.06,'a',0.1170,'alpha',pi/2, ...
    'offset', 0, 'qlim', jointLim1);
l1.I = body1.Inertia;
l1.r = body1.CenterOfMass;
```

```
l1.m = body1.Mass;

l2 = Link('d',0,'a',0,'alpha',-pi/2,...
    'offset',-pi/2, 'qlim', jointLim2);
l2.I = body2.Inertia;
l2.r = body2.CenterOfMass + body1.CenterOfMass;
l2.m = body2.Mass;

l3 = Link('d',0.3520,'a',0,'alpha',pi/2,'offset',0, ...
    'offset', 0, 'qlim', jointLim3);
l3.I = body3.Inertia;
l3.r = body3.CenterOfMass;
l3.m = body3.Mass;

l4 = Link('d',0,'a',0,'alpha',-pi/2,'offset',0, 'qlim', jointLim4);
l4.I = body4.Inertia;
l4.r = body3.CenterOfMass + body4.CenterOfMass;
l4.m = body4.Mass;

l5 = Link('d',0.3215,'a',0,'alpha',pi/2,'offset',0,...
    'offset', 0, 'qlim', jointLim5);
l5.I = body5.Inertia;
l5.r = body5.CenterOfMass;
l5.m = body5.Mass;

l6 = Link('d',0,'a',0,'alpha',-pi/2,'offset',0, 'qlim', jointLim6);
l6.I = body6.Inertia;
l6.r = body6.CenterOfMass + body5.CenterOfMass;
l6.m = body6.Mass;


l7 = Link('d',0.3049,'a',0,'alpha',0,'offset',0, 'qlim', jointLim7);
l7.I = body7.Inertia;
l7.r = body7.CenterOfMass;
l7.m = body7.Mass;


links = [l1 l2 l3 l4 l5 l6 l7];

robot = SerialLink(links, 'name', 'test');

robot.base = robot.base * transl(0,0,0.7260);

robot.plot(startPos);

end
```

MAKE SURE THE FETCH.URDF IS INCLUDED IN YOUR PATH. IT WILL EXTRACT DETAILS FROM IT.

PerfromVS function

```
function [cam] =  performVS(status)
%% Draw Fetch and staircase model
close all;
clc;
% clear;

tau_max = [33.82 131.76 76.94 66.18 29.35 25.70 7.36]';
maxSpeed = [1.25 1.45 1.57 1.52  1.57 2.26 2.26];
startPos = [1.4772    1.0687    0.1256   -0.4053   -0.0001   -2.1380
1.1257];
bot = drawFetch(startPos);

modelLocation = transl(0.45,  0.5, 0);
hold on;
helixModel = PartLoader('helix3.ply', modelLocation);

mod1 = modelLocation * transl(0.15,0,0);
mod2 = modelLocation * transl(-0.15,0,0);
mod3 = modelLocation * transl(0,0.15,0);
mod4 = modelLocation * transl(0,-0.15,0);

pS1 = [(512 + 170);512];
pS2 = [(512 - 170);512];
pS3 = [512;(512 - 170)];
pS4 = [512;(512 + 170)];

pStar = [pS2 pS1 pS4  pS3];

P = [mod1(1:3,4) mod2(1:3,4) mod3(1:3,4) mod4(1:3,4)];

depth = mean (P(1,:));
T1 = bot.fkine(bot.getpos);
depth = norm(T1(1:3,4) - modelLocation(1:3,4));

axis([-1.5 1.5 -1.5 1.5 -0.5 1.5])
%% Setup Cam

cam = CentralCamera('focal', 0.08, 'pixel', 10e-5, ...
'resolution', [1024 1024], 'centre', [512 512],'name', 'FetchCam');

% frame rate
fps = 25;
%Define values
%gain of the controler
lambda = 1;

%% Draw inital State

Tc0= bot.fkine(bot.getpos);
cam.T = Tc0;
cam.plot_camera(P, 'label','scale',0.15);

%% Projection
p = cam.plot(P, 'Tcam', Tc0);

%camera view and plotting
cam.clf()
```

```
cam.plot(pStar, '*'); % create the camera view
cam.hold(true);
cam.plot(P, 'Tcam', Tc0, 'o'); % create the camera view
pause(2)
cam.hold(true);
cam.plot(P);    % show initial view

%% Loop

ksteps = 0;
W = eye(7);
c = [1 1 1 1 1 1 1];

while true
    ksteps = ksteps + 1;
    % compute the view of the camera
    uv = cam.plot(P);

    % compute image plane error as a column
    e = pStar-uv;   % feature error
    e = e(:);
    Zest = [];

    % compute the Jacobian
    if isempty(depth)
        % exact depth from simulation (not possible in practice)
        pt = homtrans(inv(cam.T), P);
        J = cam.visjac_p(uv, pt(3,:) );
    elseif ~isempty(Zest)
        J = cam.visjac_p(uv, Zest);
    else
        J = cam.visjac_p(uv, depth );
    end

    % compute the velocity of camera in camera frame
    try
        v = lambda * pinv(J) * e;
    catch
        status = -1;
        return
    end
%     fprintf('v: %.3f %.3f %.3f %.3f %.3f %.3f\n', v);

%     The bottom weirdness (changing velocity directions) comes from
testing and the initial camera pose. Only the values under else are used
the top was left over.
    if(length(pStar(1,:)) == 1)
        v(1) = -v(1);
        v(2) = v(2);
        v(3) = v(3);
        v(4) = -v(4);
        v(5) = v(5);
        v(6) = v(6);
    else

        v(1) = -v(1);
        v(2) = v(2);
        v(3) = -v(3);
        v(4) = -v(4);
        v(5) = v(5);
```

```
        v(6) = -v(6);
    end

    J2 = bot.jacob0(bot.getpos);

    jV = (inv(W)*J2')*inv(J2*inv(W)*J2')*v;

    slowDown = 0;

    for x = 1:length(jV)
        if(jV(x) > 20)
            slowDown = 1;
        end
    end

    if(slowDown)
        jV = jV.*0.01;
    end

    qp = jV;
    q0 = bot.getpos;
%     qp = bot.getPos + (jV*deltaT)';

    W = calcW(W, bot,bot.getpos,c);

    %Update joints
    q = q0' + (1/fps)*qp;
    bot.animate(q');


    %Get camera location
    Tc = bot.fkine(q);
    cam.T = Tc;
    drawnow

    pause(1/fps)

    test = (abs(e) < 3);
    if(test)
        e
        test
        break
    end

end %loop finishese

%% Move along Z
%Camera is no longer needed, lets just move it away
cam.T = transl(0,0,10);
drawnow;

height = bot.fkine(bot.getpos);
height = (height(3,4));
W = eye(7);
c = [1 1 1 1 1 1 1];
fps = 25;
while(height ~= 0)
    distance = bot.fkine(bot.getpos) - modelLocation;
```

```
    height = bot.fkine(bot.getpos);
    height = (height(3,4));

    test = norm(distance(1:3,4));
    if(test < 0.09)
        break;
    end

    velocity = [0 0 -1 0 0 0]';

     J2 = bot.jacob0(bot.getpos);

    jV = (inv(W)*J2')*inv(J2*inv(W)*J2')*velocity;


    for z = 1:length(jV)
        if(jV(z) > maxSpeed(z))
            ratio = maxSpeed(z)/jV(z);
            jV = jV*ratio;
        elseif (jV(z) < -maxSpeed(z))
            ratio = -maxSpeed(z)/jV(z);
            jV = jV*ratio;
        end
    end



    qp = jV;
    q0 = bot.getpos;
%     qp = bot.getPos + (jV*deltaT)';

    W = calcW(W, bot,bot.getpos,c);

    %Update joints
    q = q0' + (1/fps)*qp;
    bot.animate(q');

    pause(1/fps);
end
%% Move piece to random spot

if(status == 1)
    qMatrix = jtraj(bot.getpos, [0.9313    0.9591    0.2513   -0.4053   -
0.1257    0    1],40);

    for i = 1:length(qMatrix(:,1))

        bot.animate(qMatrix(i,:));

        helixModel.MovePart(bot.fkine(qMatrix(i,:)) * troty(pi));
    end
end

if(status == 2)
    disp("Mass = 2kg");
    mass = 2;
    time = 1;
```

```
    overallTorque = dynamicTorque(bot,helixModel,1, mass, time,tau_max,
maxSpeed);
    disp("Sucessfully Completed Drop");

end


if(status == 3)
    disp("Mass = 5kg");
    mass = 5;
    time = 1;
    overallTorque = dynamicTorque(bot,helixModel,0, mass, time,tau_max,
maxSpeed);

end
try
    plotTorques(overallTorque,tau_max);
end

end
```

Calculate 'W' Function

```
function [ W ] = calcW( WPrev, robot, pose, c)

W = eye(length(pose));
Links = robot.links;

for i = 1:length(pose)
    try
        qMin = Links(i).qlim(1);
        qMax = Links(i).qlim(2);
        q = pose(i);

        value = ((qMax - qMin)*(2*q - qMax - qMin))/(c(i)*((qMax -
q)^2)*((q - qMin)^2));


        if(isnan(value))
            c(i)
            q
            qMax
            qMin
        end

        if(value - WPrev(i,i) > 0)
            W(i,i) = 1 + abs(value);
        else

            W(i,i) = 1;
        end
    catch
        W(i,i) = 1;
    end

end

end
```

RunMe Script

```
%% Only VisualServoing
%Will visual servo to the object, move down, pick the object up and move
%using jtraj to a joint position I randomly selected. No torque or torque
%plot will be checked here
clear all;
camObj = performVS(1);

%% Mass = 2kg
% Visual Servo to object, move down, pick up the object, then maintain a
% safe torque (and speed) throughout the process while trying to complete
% the task at the same time as the 5kg scenario
%Three motions once the object is grasped
%   Move up 1 and along x -0.4
%   move along y -3
%   move down 1 and along x 0.4
% Torques will be graphed at the completion of all motions. Seperate figure
% for each motion. Each joint is on seperate subplot. Blue line indicates
% joint torque. Orange Line indicates max joint torque.
 clear all;
camObj = performVS(2);

%% Mass = 5kg
% Visual Servo to object, move down, pick up the object, a safe speed will
% be maintained but not a safe torque while trying to complete the task at
%the same time as the 2kg scenario. It will fail while moving along y
%Three motions once the object is grasped
%   Move up 1 and along x -0.4
%   move along y -3
%   move down 1 and along x 0.4
% Torques will be graphed at the completion of all motions. Seperate figure
% for each motion. Each joint is on seperate subplot. Blue line indicates
% joint torque. Orange Line indicates max joint torque.
clear all;
camObj = performVS(3);
```

GitHub Clone Link

https://github.com/denis-draca/roboticsA4.git

Start with the RunMe Script