Table of Contents

[1 Table of Figures 1](#_Toc515884390)

[2 Table of Tables 1](#_Toc515884391)

[3 Table of Equations 1](#_Toc515884392)

[4 Subsystem Three: Controls and Decision Making 3](#_Toc515884393)

[4.1 Requirements and Functional Decomposition 3](#_Toc515884394)

[4.2 Background and Prior Art 4](#_Toc515884395)

[5 Bibliography 9](#_Toc515884396)

# Table of Figures

[Figure 31: SS3 Breakdown 3](#_Toc515884397)

# Table of Tables

No table of figures entries found.

# Table of Equations

[Equation 2: Jacobian 4](#_Toc515884398)

[Equation 3: Relationship between q and x 4](#_Toc515884399)

[Equation 4: Euler–Lagrange equations (Khatib, 2008) 5](#_Toc515884400)

[Equation 5: The Lagrangian (Khatib, 2008) 5](#_Toc515884401)

[Equation 6: Kinetic Energy (Khatib, 2008) 5](#_Toc515884402)

[Equation 7: Kinetic Energy Partial Derivative 5](#_Toc515884403)

[Equation 8: Kinetic Energy Time Derivative 5](#_Toc515884404)

[Equation 9: Inertial Forces 5](#_Toc515884405)

[Equation 10: Vector of centrifugal and Coriolis forces 5](#_Toc515884406)

[Equation 11: Explicit form of EOM 6](#_Toc515884407)

[Equation 12: Kinetic Energy of Link i 6](#_Toc515884408)

[Equation 13: Kinetic energy of System 6](#_Toc515884409)

[Equation 14: Kinetic Energy of Total System 6](#_Toc515884410)

[Equation 15: Explicit form of Manipulator Mass Matrix 6](#_Toc515884411)

[Equation 16: mijk 7](#_Toc515884412)

[Equation 17: Christoffel Symbols 7](#_Toc515884413)

[Equation 18: Coefficients associated with centrifugal forces 7](#_Toc515884414)

[Equation 19: Coefficients associated with Coriolis force 7](#_Toc515884415)

[Equation 20: Potential Energy of the System 7](#_Toc515884416)

[Equation 21: Gravitational Potential Energy of Each Link 8](#_Toc515884417)

[Equation 22: Vector of Gravity Force 8](#_Toc515884418)

# 

# Subsystem Three: Controls and Decision Making

This section details the analysis, design, implementation, and results of the subsystem responsible for determining the actions required by the actuation system.

## Requirements and Functional Decomposition

The overarching purpose of subsystem three (SS3) was to determine the action that should be taken by the actuator system to minimise the error in the system.



Figure 31: SS3 Breakdown

As seen in Figure 29: SS3 Breakdown, to determine the action required in a state (given by the values determined by SS1, SS2, and SS4) the action required in any given state must be known.

To determine the general solution of what actions should be taken at any given state the controls parameters and method for the system should be derived, and then this model should be refined by practically tuning the solution.

The methodology for tuning the controls parameters of the system is discussed in kt, but the initial values to be refined are best source directly from the theory.

To determine the control parameters for the system the following method is employed:

1. Derive the equations of motion (EOM) for the system;
2. From the EOM derive the transfer function (TF) of the system (torque () with respect to angle ());
3. Transform the TF into the Laplace Domain; and,
4. Derive the PID parameters from the Laplace Domain TF.

Ultimately, five sets of controls were derived during the project: three for the 3 Degree of Freedom (DOF) lower extremity system (each joint had its own parameters), one for the continuous servomotor (which was abandoned as discussed in kt), and the positional servomotor.

The 3 DOF system was to control the lower extremity exoskeleton being constructed by the actuators and structural side of the project. However, as actual values for system parameters (masses, dimensions, moments of inertia, etc…) were never confirmed the solution had to be found algebraically.

The two systems used in testing featured their own embedded control systems, and their torque, angle, velocity, and acceleration could not be directly controlled. As such precise controls could not be derive from first principles. Instead the controls systems would need to be tuned empirically to achieve the desired system response.

## Background and Prior Art

The goal of SS3 is to model the dynamics expected of the system, establish the manipulator equations of motion, and derive the appropriate controls structure to create the behaviour required, in a stable fashion.

### Jacobian

For a system, in this case a manipulator, in the configuration given by the vector there is corresponding psotion for the end-effector given by the vector . The Jacobian matrix, , describes the relationship between the time derivatives of and ( and respectively). The Jacobian matrix, or simply the Jacobian, given by Equation 2, allows use to describe the system by Equation 3.

Equation 2: Jacobian

Equation 3: Relationship between q and x

Where .

### Dynamics

#### Explicit Form of the Equations of Motion

We begin with the Euler–Lagrange equations, or Lagrange's equations of the second kind, Equation 4.

Equation 4: Euler–Lagrange equations (Khatib, 2008)

Where is the vector of applied generalised torques. The Lagrangian, L, is given by Equation 5.

Equation 5: The Lagrangian (Khatib, 2008)

Where V is the potential energy of the system, and K is the kinetic energy of the system. As seen in Equation 6, K may be given in terms of the generalised velocities, (as seen in Equation 4: Euler–Lagrange equations) and the manipulator mass matrix M.

Equation 6: Kinetic Energy (Khatib, 2008)

Through Equation 4 we may say that

Equation 7and Equation 8 hold.

Equation 7: Kinetic Energy Partial Derivative

Equation 8: Kinetic Energy Time Derivative

Thus the inertial forces of Equation 5 may be expressed as Equation 9, where is the vector of centrifugal and Coriolis forces, Equation 10.

Equation 9: Inertial Forces

Equation 10: Vector of centrifugal and Coriolis forces

Through Equation 9 we may yield the explicit form of the equations of motion (EOM), see Equation 11. Where is the vector of gravity force and is the vector of centrifugal and Coriolis forces. Equation 11, once found, may be used to map the relationship between the torque applied by the systems actuators and the resulting system configuration.

Equation 11: Explicit form of EOM

#### Explicit form of Manipulator Mass Matrix

Kinetic energy is subject to the adaptive property (Siciliano & Khatib, 2016), and thus the total kinetic energy of a system is the summation of the kinetic energy of its links. Links here refers to the actuated limb segments of the exoskeleton correlating with the thigh, shin, and foot.

The kinetic energy of each link is comprised of a rotational and linear motion component. For a link with linear motion of , an angular motion of , and an inertia tensor of , the kinetic energy of the link , , is given by Equation 12Equation 12. Where refers to the centre of mass of the link.

Equation 12: Kinetic Energy of Link i

Given Equation 12 and the additive property it may be said that the kinetic of the system in total is given by Equation 13

Equation 13: Kinetic energy of System

Using Equation 12 and Equation 13, factoring out , we develop Equation 14

Equation 14: Kinetic Energy of Total System

Equating Equation 14 to Equation 6 we find the Explicit form of Manipulator Mass Matrix, Equation 15.

Equation 15: Explicit form of Manipulator Mass Matrix

#### Vector of centrifugal and Coriolis forces

We begin with Equation 10: Vector of centrifugal and Coriolis forces.

Sparing the derivation, we can say that givem Equation 16 and Equation 17, that Equation 18 and Equation 19 hold true.

Equation 16: mijk

Equation 17: Christoffel Symbols

Equation 18: Coefficients associated with centrifugal forces

Equation 19: Coefficients associated with Coriolis force

Where

#### Vector of gravity force

The vector of gravity force, , represents the gravitational potential energy of the system. The gravitational potential energy of the system is given by the gravitational potential energy of every link in the system, see Equation 20.

Equation 20: Potential Energy of the System

The gravitational potential energy of each link is given by Equation 21, where is the height of the centre of mass of the link relative to the origin (pelvis).

Equation 21: Gravitational Potential Energy of Each Link

Thus, we may say (using the Jacobian to map the location) the vector of gravity force, , is given by Equation 22.

Equation 22: Vector of Gravity Force

## Approach and Execution

Before the kinematics of the system could be found, the exoskeleton needed to be abstracted into a model. Consider the following:

1. The exoskeleton is to be affixed to the lower torso of the pilot;
2. The pilot is presumed to maintain the balance of the system using their body;
3. The pilot should be able to manipulate the legs of the system independently;
4. The legs, while part of the same exoskeletons, are essentially fixed at the pelvis and operate independently; and,
5. We may therefore consider each leg as an independent manipulator with a fixed reference frame at the pelvis.

As noted in kt, each joint of the exoskeleton shall be constrained to 1 DOF. Therefore, we may abstract the exoskeleton as two 3 DOF RRR manipulators, as seen in Figure 30: Exoskeleton Abstraction.



Figure 32: Exoskeleton Abstraction

For modelling the system, the parameters seen in Figure 31: 3 DOF RRR Parameterisation shall be used. Note angle shall be measured relative to the previous link with a clockwise positive convention.



Figure 33: 3 DOF RRR Parameterisation

As noted in kt (requirements section) the actual values for the exoskeleton were never confirmed and the controls had to be completed symbolically. On one hand, this resulted in a general solution that can be applied to any 3 DOF RRR serial manipulator. On the other hand, the equations become cumbersome and large. As a result, many of the equations shall be taken to their general form, as explicit solutions shall be left as an exercise to the reader.

### Jacobian

To find the Jacobian of a 3 DOF Revolute Manipulator:

Or, for a system at low velocity (i.e. standing, squatting, sitting, stairs, walking):

In matrix form

The angular velocities are simply additive:

From which we obtain the Jacobian of a 3 DOF Revolute Manipulator, as seen in Equation 23.

Equation 23: 3 DOF Revolute Manipulator Jacobian

### Dynamics

For a 3 DOF Revolute Manipulator where the inertia tensors of the links are , , and (Equation 24).

Equation 24: Inertia Tensors

#### Explicit form of Manipulator Mass Matrix

The mass matrix for the 3 DOF Revolute Manipulator is given by Equation 25. This process was completed with symbolic variables in MATLAB R2017b, as detailed in the attached files (get\_EOM.m) kt.

Equation 25: Mass Matrix for the 3 DOF Revolute Manipulator

#### Vector of centrifugal and Coriolis forces

We begin with Equation 18.

Equation 26: Vector of Centrifugal Forces

Next Equation 19.

Equation 27: Vector of Coriolis Force

From Equation 17

Where

This process was completed with symbolic variables in MATLAB R2017b, as detailed in the attached files (get\_EOM.m) kt.

#### Vector of gravity force

Continuing from Equation 22: Vector of Gravity Force, we find

)

Equation 28.

Equation 28: Gravity Vector

Given , see Equation 23, we may find Equation 29.

Equation 29: Vector of Gravity Force

This process was completed with symbolic variables in MATLAB R2017b, as detailed in the attached files (get\_EOM.m) kt.

### Explicit Form of the Equations of Motion

Base on Equation 25Equation 25: Mass Matrix for the 3 DOF Revolute Manipulator, Equation 26: Vector of Centrifugal Forces, Equation 27: Vector of Coriolis Force, and Equation 29: Vector of Gravity Force we may find the solution to Equation 11, as given by Equation 30

Equation 30: Equations of Motion

This process was completed with symbolic variables in MATLAB R2017b, as detailed in the attached files (get\_EOM.m) kt. The output of these equations, and the equations of motion for the system are as seen in Table 3: Equations of Motion.

Table 3: Equations of Motion

|  |  |
| --- | --- |
| T1 | dda2\*(Izz2 + Izz3 + m3\*(L2^2 + 2\*cos(a3)\*L2\*l3 + L1\*cos(a2)\*L2 + l3^2 + L1\*cos(a2 + a3)\*l3) + l2\*m2\*(l2 + L1\*cos(a2))) + dda3\*(Izz3 + l3\*m3\*(l3 + L1\*cos(a2 + a3) + L2\*cos(a3))) - da1\*(L1\*da2\*(l2\*m2\*sin(a2) + l3\*m3\*sin(a2 + a3) + L2\*m3\*sin(a2)) + da3\*l3\*m3\*(L1\*sin(a2 + a3) + L2\*sin(a3))) - da2\*(L1\*(da1 + da2)\*(l2\*m2\*sin(a2) + l3\*m3\*sin(a2 + a3) + L2\*m3\*sin(a2)) + da3\*l3\*m3\*(L1\*sin(a2 + a3) + L2\*sin(a3))) + dda1\*(Izz1 + Izz2 + Izz3 + L1^2\*m2 + L1^2\*m3 + L2^2\*m3 + l1^2\*m1 + l2^2\*m2 + l3^2\*m3 + 2\*L1\*l3\*m3\*cos(a2 + a3) + 2\*L1\*L2\*m3\*cos(a2) + 2\*L1\*l2\*m2\*cos(a2) + 2\*L2\*l3\*m3\*cos(a3)) + g\*m2\*(l2\*cos(a1 + a2) + L1\*cos(a1)) + g\*m3\*(L2\*cos(a1 + a2) + L1\*cos(a1) + l3\*cos(a1 + a2 + a3)) + g\*l1\*m1\*cos(a1) - da3\*l3\*m3\*(L1\*sin(a2 + a3) + L2\*sin(a3))\*(da1 + da2 + da3) |
| T2 | dda1\*(Izz2 + Izz3 + m3\*(L2^2 + 2\*cos(a3)\*L2\*l3 + L1\*cos(a2)\*L2 + l3^2 + L1\*cos(a2 + a3)\*l3) + l2\*m2\*(l2 + L1\*cos(a2))) + da1\*(L1\*da1\*(l2\*m2\*sin(a2) + l3\*m3\*sin(a2 + a3) + L2\*m3\*sin(a2)) - L2\*da3\*l3\*m3\*sin(a3)) + dda2\*(Izz2 + Izz3 + m3\*(L2^2 + 2\*cos(a3)\*L2\*l3 + l3^2) + l2^2\*m2) + dda3\*(Izz3 + l3\*m3\*(l3 + L2\*cos(a3))) + g\*m3\*(L2\*cos(a1 + a2) + l3\*cos(a1 + a2 + a3)) + g\*l2\*m2\*cos(a1 + a2) - L2\*da2\*da3\*l3\*m3\*sin(a3) - L2\*da3\*l3\*m3\*sin(a3)\*(da1 + da2 + da3) |
| T3 | dda3\*(m3\*l3^2 + Izz3) + dda1\*(Izz3 + l3\*m3\*(l3 + L1\*cos(a2 + a3) + L2\*cos(a3))) + da1\*(da1\*l3\*m3\*(L1\*sin(a2 + a3) + L2\*sin(a3)) + L2\*da2\*l3\*m3\*sin(a3)) + dda2\*(Izz3 + l3\*m3\*(l3 + L2\*cos(a3))) + g\*l3\*m3\*cos(a1 + a2 + a3) + L2\*da2\*l3\*m3\*sin(a3)\*(da1 + da2) |

Where

### Laplace Transform

Once the equations of motion (EOM) had been found via MATLAB, the Laplace transform was performed on the EOM. Alas, MATLAB doesn’t recognise the relationship between , , and for *syms*; they had to be treated as separate variables. Thus, the Laplace transform could not be applied outright. Instead, it would be performed manually by transforming each term individually through string manipulation (how MATLAB stores symbolic equations). This can be seen in detail in kt (get\_EOM). But essentially boiled down to finding all instances of a variable (e.g. cos(a3)) and transforming it (e.g. 'A3\*(s/(s^2 + 1))'). This of course had to happen from greatest term to smallest to avoid confusions in replacements (i.e. a3 includes dda3, da3, and a3, so it should be completed after all terms including dda3 and da3 are transformed.

While 3D laplace transforms are possible (Dawkins, 2018) it rather messy. Instead it will be assumed the behaviour of the systems in relation to a joint can be approximated to equal the behaviour of the system if the instantaneous state of the rest of the joints is constant. This is to say, if we update sufficiently quickly/frequently we may treat changes in and as negligible.

The resulting equations may be found in Table 4: Laplace EOM.

Table 4: Laplace EOM

|  |  |
| --- | --- |
| T1 | dda2\*(m3\*L2^2 + 2\*m3\*cos(a3)\*L2\*l3 + L1\*m3\*cos(a2)\*L2 + m2\*l2^2 + L1\*m2\*cos(a2)\*l2 + m3\*l3^2 + L1\*m3\*cos(a2 + a3)\*l3 + Izz2 + Izz3) + dda3\*(Izz3 + l3^2\*m3 + L1\*l3\*m3\*cos(a2 + a3) + L2\*l3\*m3\*cos(a3)) - da2\*(L1\*(da2 + A1\*s)\*(l2\*m2\*sin(a2) + l3\*m3\*sin(a2 + a3) + L2\*m3\*sin(a2)) + da3\*l3\*m3\*(L1\*sin(a2 + a3) + L2\*sin(a3))) - A1\*s\*(L1\*da2\*(l2\*m2\*sin(a2) + l3\*m3\*sin(a2 + a3) + L2\*m3\*sin(a2)) + da3\*l3\*m3\*(L1\*sin(a2 + a3) + L2\*sin(a3))) + A1\*s^2\*(Izz1 + Izz2 + Izz3 + L1^2\*m2 + L1^2\*m3 + L2^2\*m3 + l1^2\*m1 + l2^2\*m2 + l3^2\*m3 + 2\*L1\*l3\*m3\*cos(a2 + a3) + 2\*L1\*L2\*m3\*cos(a2) + 2\*L1\*l2\*m2\*cos(a2) + 2\*L2\*l3\*m3\*cos(a3)) + (A1\*g\*m3\*(L1\*s - l3\*sin(a2 + a3) - L2\*sin(a2) + l3\*s\*cos(a2 + a3) + L2\*s\*cos(a2)))/(s^2 + 1) + (A1\*g\*m2\*(L1\*s - l2\*sin(a2) + l2\*s\*cos(a2)))/(s^2 + 1) - da3\*l3\*m3\*(L1\*sin(a2 + a3) + L2\*sin(a3))\*(da2 + da3 + A1\*s) + (A1\*g\*l1\*m1\*s)/(s^2 + 1) |
| T2 | dda1\*(Izz2 + Izz3 + m3\*(L2^2 + l3^2 + 2\*L2\*l3\*cos(a3) + (A2\*L1\*L2\*s)/(s^2 + 1) - (A2\*L1\*l3\*(sin(a3) - s\*cos(a3)))/(s^2 + 1)) + l2\*m2\*(l2 + (A2\*L1\*s)/(s^2 + 1))) + da1\*(L1\*da1\*((A2\*l2\*m2)/(s^2 + 1) + (A2\*L2\*m3)/(s^2 + 1) - (A2\*l3\*m3\*(cos(a3) - s\*sin(a3)))/(s^2 + 1)) - L2\*da3\*l3\*m3\*sin(a3)) + dda3\*(Izz3 + l3\*m3\*(l3 + L2\*cos(a3))) + A2\*s^2\*(Izz2 + Izz3 + m3\*(L2^2 + 2\*cos(a3)\*L2\*l3 + l3^2) + l2^2\*m2) - g\*m3\*((A2\*L2\*(sin(a1) - s\*cos(a1)))/(s^2 + 1) + (A2\*l3\*(sin(a1 + a3) - s\*cos(a1 + a3)))/(s^2 + 1)) - (A2\*g\*l2\*m2\*(sin(a1) - s\*cos(a1)))/(s^2 + 1) - L2\*da3\*l3\*m3\*sin(a3)\*(da1 + da3 + A2\*s) - A2\*L2\*da3\*l3\*m3\*s\*sin(a3) |
| T3 | dda2\*(Izz3 + l3\*m3\*(l3 + (A3\*L2\*s)/(s^2 + 1))) + da1\*(da1\*l3\*m3\*((A3\*L2)/(s^2 + 1) - (A3\*L1\*(cos(a2) - s\*sin(a2)))/(s^2 + 1)) + (A3\*L2\*da2\*l3\*m3)/(s^2 + 1)) + dda1\*(Izz3 + l3\*m3\*(l3 - (A3\*L1\*(sin(a2) - s\*cos(a2)))/(s^2 + 1) + (A3\*L2\*s)/(s^2 + 1))) + A3\*s^2\*(m3\*l3^2 + Izz3) - (A3\*g\*l3\*m3\*(sin(a1 + a2) - s\*cos(a1 + a2)))/(s^2 + 1) + (A3\*L2\*da2\*l3\*m3\*(da1 + da2))/(s^2 + 1) |

### Transfer Function

Given the solution found in Table 4: Laplace EOM, we may rearrange the equations found so the solution is given in terms of the transfer function . The transfer functions for the system may be found in

Table 5: Transfer Functions

|  |  |
| --- | --- |
|  | -1/(((da2\*(L1\*da2\*(l2\*m2\*sin(a2) + l3\*m3\*sin(a2 + a3) + L2\*m3\*sin(a2)) + da3\*l3\*m3\*(L1\*sin(a2 + a3) + L2\*sin(a3))) - dda2\*(m3\*L2^2 + 2\*m3\*cos(a3)\*L2\*l3 + L1\*m3\*cos(a2)\*L2 + m2\*l2^2 + L1\*m2\*cos(a2)\*l2 + m3\*l3^2 + L1\*m3\*cos(a2 + a3)\*l3 + Izz2 + Izz3) - dda3\*(Izz3 + l3^2\*m3 + L1\*l3\*m3\*cos(a2 + a3) + L2\*l3\*m3\*cos(a3)) + da3\*l3\*m3\*(L1\*sin(a2 + a3) + L2\*sin(a3))\*(da2 + da3))/(s^2\*(Izz1 + Izz2 + Izz3 + L1^2\*m2 + L1^2\*m3 + L2^2\*m3 + l1^2\*m1 + l2^2\*m2 + l3^2\*m3 + 2\*L1\*l3\*m3\*cos(a2 + a3) + 2\*L1\*L2\*m3\*cos(a2) + 2\*L1\*l2\*m2\*cos(a2) + 2\*L2\*l3\*m3\*cos(a3)) - s\*(L1\*da2\*(l2\*m2\*sin(a2) + l3\*m3\*sin(a2 + a3) + L2\*m3\*sin(a2)) + da3\*l3\*m3\*(L1\*sin(a2 + a3) + L2\*sin(a3))) - L1\*da2\*s\*(l2\*m2\*sin(a2) + l3\*m3\*sin(a2 + a3) + L2\*m3\*sin(a2)) + (g\*m3\*(L1\*s - l3\*sin(a2 + a3) - L2\*sin(a2) + l3\*s\*cos(a2 + a3) + L2\*s\*cos(a2)))/(s^2 + 1) + (g\*m2\*(L1\*s - l2\*sin(a2) + l2\*s\*cos(a2)))/(s^2 + 1) - da3\*l3\*m3\*s\*(L1\*sin(a2 + a3) + L2\*sin(a3)) + (g\*l1\*m1\*s)/(s^2 + 1)) - 1)\*(s^2\*(Izz1 + Izz2 + Izz3 + L1^2\*m2 + L1^2\*m3 + L2^2\*m3 + l1^2\*m1 + l2^2\*m2 + l3^2\*m3 + 2\*L1\*l3\*m3\*cos(a2 + a3) + 2\*L1\*L2\*m3\*cos(a2) + 2\*L1\*l2\*m2\*cos(a2) + 2\*L2\*l3\*m3\*cos(a3)) - s\*(L1\*da2\*(l2\*m2\*sin(a2) + l3\*m3\*sin(a2 + a3) + L2\*m3\*sin(a2)) + da3\*l3\*m3\*(L1\*sin(a2 + a3) + L2\*sin(a3))) - L1\*da2\*s\*(l2\*m2\*sin(a2) + l3\*m3\*sin(a2 + a3) + L2\*m3\*sin(a2)) + (g\*m3\*(L1\*s - l3\*sin(a2 + a3) - L2\*sin(a2) + l3\*s\*cos(a2 + a3) + L2\*s\*cos(a2)))/(s^2 + 1) + (g\*m2\*(L1\*s - l2\*sin(a2) + l2\*s\*cos(a2)))/(s^2 + 1) - da3\*l3\*m3\*s\*(L1\*sin(a2 + a3) + L2\*sin(a3)) + (g\*l1\*m1\*s)/(s^2 + 1))) |
|  | 1/(((dda1\*(Izz2 + Izz3 + m3\*(L2^2 + 2\*cos(a3)\*L2\*l3 + l3^2) + l2^2\*m2) + dda3\*(Izz3 + l3\*m3\*(l3 + L2\*cos(a3))) - L2\*da1\*da3\*l3\*m3\*sin(a3) - L2\*da3\*l3\*m3\*sin(a3)\*(da1 + da3))/(dda1\*(m3\*((L1\*l3\*(sin(a3) - s\*cos(a3)))/(s^2 + 1) - (L1\*L2\*s)/(s^2 + 1)) - (L1\*l2\*m2\*s)/(s^2 + 1)) - s^2\*(Izz2 + Izz3 + m3\*(L2^2 + 2\*cos(a3)\*L2\*l3 + l3^2) + l2^2\*m2) + g\*m3\*((L2\*(sin(a1) - s\*cos(a1)))/(s^2 + 1) + (l3\*(sin(a1 + a3) - s\*cos(a1 + a3)))/(s^2 + 1)) - L1\*da1^2\*((L2\*m3)/(s^2 + 1) + (l2\*m2)/(s^2 + 1) - (l3\*m3\*(cos(a3) - s\*sin(a3)))/(s^2 + 1)) + (g\*l2\*m2\*(sin(a1) - s\*cos(a1)))/(s^2 + 1) + 2\*L2\*da3\*l3\*m3\*s\*sin(a3)) - 1)\*(dda1\*(m3\*((L1\*l3\*(sin(a3) - s\*cos(a3)))/(s^2 + 1) - (L1\*L2\*s)/(s^2 + 1)) - (L1\*l2\*m2\*s)/(s^2 + 1)) - s^2\*(Izz2 + Izz3 + m3\*(L2^2 + 2\*cos(a3)\*L2\*l3 + l3^2) + l2^2\*m2) + g\*m3\*((L2\*(sin(a1) - s\*cos(a1)))/(s^2 + 1) + (l3\*(sin(a1 + a3) - s\*cos(a1 + a3)))/(s^2 + 1)) - L1\*da1^2\*((L2\*m3)/(s^2 + 1) + (l2\*m2)/(s^2 + 1) - (l3\*m3\*(cos(a3) - s\*sin(a3)))/(s^2 + 1)) + (g\*l2\*m2\*(sin(a1) - s\*cos(a1)))/(s^2 + 1) + 2\*L2\*da3\*l3\*m3\*s\*sin(a3))) |
|  | 1/(((dda1\*(m3\*l3^2 + Izz3) + dda2\*(m3\*l3^2 + Izz3))/(s^2\*(m3\*l3^2 + Izz3) + da1\*(da1\*l3\*m3\*(L2/(s^2 + 1) - (L1\*(cos(a2) - s\*sin(a2)))/(s^2 + 1)) + (L2\*da2\*l3\*m3)/(s^2 + 1)) - dda1\*l3\*m3\*((L1\*(sin(a2) - s\*cos(a2)))/(s^2 + 1) - (L2\*s)/(s^2 + 1)) - (g\*l3\*m3\*(sin(a1 + a2) - s\*cos(a1 + a2)))/(s^2 + 1) + (L2\*da2\*l3\*m3\*(da1 + da2))/(s^2 + 1) + (L2\*dda2\*l3\*m3\*s)/(s^2 + 1)) + 1)\*(s^2\*(m3\*l3^2 + Izz3) + da1\*(da1\*l3\*m3\*(L2/(s^2 + 1) - (L1\*(cos(a2) - s\*sin(a2)))/(s^2 + 1)) + (L2\*da2\*l3\*m3)/(s^2 + 1)) - dda1\*l3\*m3\*((L1\*(sin(a2) - s\*cos(a2)))/(s^2 + 1) - (L2\*s)/(s^2 + 1)) - (g\*l3\*m3\*(sin(a1 + a2) - s\*cos(a1 + a2)))/(s^2 + 1) + (L2\*da2\*l3\*m3\*(da1 + da2))/(s^2 + 1) + (L2\*dda2\*l3\*m3\*s)/(s^2 + 1))) |

### PID Development

At this point we in many ways reach the limits of symbolic variables in MATLAB. PIDs can be created with tuneable parameters, so it is possible to treat the system’s variables (mass, length, etc…) as tuneable values preserving our purely algebraic solution. However, this would exist as a purely intellectual exercise of no practical merit. Instead it is more prudent to develop the software to find the PID parameters for the system once basic fundamentals regarding the system can be confirmed.

See (imma\_real\_boy) kt for MATLAB code which shall substitute all systems variables for their actual variables (when confirmed). This shall allow the reader to solve for the “actual” PID values for their system[[1]](#footnote-1).

At one point this section was to be used to show the process by which one could create the PID parameters to control the system:

* A general solution sis provided to the EOM for a 3 DOF manipulator (just sub in the desired values for p, doesn’t even need to be RRR);
* The laplace transform is performed on these values; and,
* A transfer function is created from these equations.

It is trivial from this point to plug the transfer functions into MATLAB’s pidTuner and generate a PID. But that’s the point, its trivial, it merely shows the author is capable of using a MATLAB suite. It does not provide actual values for control (these can not be found until there is an exoskeleton to control), and the documentation for pidTuner is sufficient that if you got this far you’ll be fine.

Time could be spent discussing how the root locus method could be used to develop the controls for the system. However, that would just be a recount of Ogata and as such I refer to *Modern Control Engineering*[[2]](#footnote-2) (Ogata, 2010), rather than repeating it.

### PID Tuning

For the reasons discussed in kt to control the equipment used in the demonstration rig (a 1 DOF exoskeleton) PID control was selected.

## Results and Discussion

# Bibliography

Agarwal, A. (2005). *Foundations of analog & digital electronic circuits* (1 ed.). Massachusetts: Massachusetts Institute of Technology. Retrieved June 1, 2018, from https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=1&ved=0ahUKEwiBu7yQl7LbAhUEoZQKHfYlBzkQFggpMAA&url=http%3A%2F%2Fsiva.bgk.uni-obuda.hu%2Fjegyzetek%2FMechatronikai\_alapismeretek%2FEnglish\_Mechatr%2FElectr\_Eng-1%2FLiterature%2FFoundations%2520o

American Technologies Network Corporation. (2018, May 30). *How Does Night Vision Work*. Retrieved from atncorp.com: https://www.atncorp.com/hownightvisionworks

Arrow. (2018). *Magnetoresistive Sensor.* Retrieved May 30, 2018, from arrow.com: https://www.arrow.com/en/categories/sensors/magnetoresistive-sensors

Axe, D. (2012, May 23). *Combat Exoskeleton Marches Toward Afghanistan Deployment*. Retrieved May 30, 2018, from Wired: https://www.wired.com/2012/05/combat-exoskeleton-afghanistan/

Bulgrin, M. (2017, May 11). *The History of the Hose Clamp.* Retrieved from normagroup.com: https://blog.normagroup.com/en/the-history-of-the-hose-clamp/

Bunnings. (2018, May 31). *Kinetic 21 - 44mm 304 Stainless Steel Hose Clamp.* Retrieved May 31, 2018, from Bunnings.com: https://www.bunnings.com.au/kinetic-21-44mm-304-stainless-steel-hose-clamp\_p4920194

Charara, S. (2015, July 9). *This robotic exoskeleton helps paralysed patients to walk and it's getting smarter*. Retrieved August 23, 2017, from Wearable: https://www.wareable.com/wearable-tech/exoskeleton-paralysed-patients-ekso-bionics-gt-sarah-thomas

Computer Cable Store. (2018, May 31). *11 7/8 Inch Black Standard Nylon Cable Tie - 100 Pack.* Retrieved May 31, 2018, from computercablestore.com: https://www.computercablestore.com/11-78-inch-black-standard-nylon-cable-tie-100-pack

Cornwall, W. (2015, October 15). *Feature: Can we build an ‘Iron Man’ suit that gives soldiers a robotic boost?* Retrieved August 20, 2017, from sciencemag.org: http://www.sciencemag.org/news/2015/10/feature-can-we-build-iron-man-suit-gives-soldiers-robotic-boost

Cracknell, A. P., & Hayes, L. (2007). *Introduction to Remote Sensing* (2 ed.). London: Taylor and Francis. Retrieved May 30, 2018

Cutnell, J. D., & Johnson, K. W. (1998). *Physics* (4th ed.). New York: Wiley.

Cyberdyne. (2015, August 1). *CYBERDYNE Inc. has begun seeking approval from the U. S. Food and Drug Administration (FDA)*. Retrieved August 23, 2017, from cyberdyne.jp: https://www.cyberdyne.jp/english/company/PressReleases\_detail.html?id=1075

Cyberdyne. (2016). *What’s HAL?* Retrieved August 19, 2017, from cyberdyne.jp: https://www.cyberdyne.jp/english/products/HAL/

Cybernetic Zoo. (2010, October 14). *1890 – Assisted-walking Device – Nicholas Yagn (Russian)*. Retrieved May 30, 2018, from Cyberneticzoo.com: http://cyberneticzoo.com/tag/nicholas-yagn/

Cybernetic Zoo. (2010, April 10). *1965-71 – G.E. Hardiman I Exoskeleton – Ralph Mosher (American)*. Retrieved May 30, 2018, from cyberneticzoo.com: http://cyberneticzoo.com/man-amplifiers/1966-69-g-e-hardiman-i-ralph-mosher-american/

Dawkins, P. (2018). *Differential Equations - Notes - Laplace’s Equation*. Retrieved June 3, 2018, from Paul's Online Math Notes: http://tutorial.math.lamar.edu/Classes/DE/LaplacesEqn.aspx

Dunietz, J. (2017, July 27). *Robotic Exoskeleton Adapts While It’s Worn*. Retrieved August 20, 2017, from scientificamerican.com: https://www.scientificamerican.com/article/robotic-exoskeleton-ldquo-evolves-rdquo-while-its-worn/

Future Electronics. (2018, May 30). *What is Optoelectronics?* Retrieved from Future Electronics: http://www.futureelectronics.com/en/optoelectronics/infrared-receivers.aspx

Garbett, I. (2001, Janurary 1). *Light attenuation and exponential laws*. Retrieved May 30, 2018, from plus.maths.org: https://plus.maths.org/content/light-attenuation-and-exponential-laws

Golnaraghi, F., & Kuo, B. C. (2010). *Automatic Control Systems* (9 ed.). Hoboken: John Wiley & Sons, Inc. Retrieved June 4, 2018

Gross, K. (2018, Feburary 19). *Ultrasonic Sensors: Advantages and Limitations*. Retrieved May 30, 2018, from MaxBotix: https://www.maxbotix.com/articles/advantages-limitations-ultrasonic-sensors.htm/

Jackson, R., Green, K. R., & Eisenbeis, R. (2017). *Achieve greater precision, reliability with integrated magnetic sensing technology.* Retrieved May 30, 2018, from ti.com: http://www.ti.com/general/docs/lit/getliterature.tsp?baseLiteratureNumber=sszy030&fileType=pdf

Karlin, S. (2011, July 29). *Raytheon Sarcos’s Exoskeleton Nears Production*. Retrieved August 11, 2017, from spectrum.ieee.org: http://spectrum.ieee.org/at-work/innovation/raytheon-sarcoss-exoskeleton-nears-produc

Keller, M. (2016, August 25). *Exoskeleton - Do You Even Lift, Bro? Hardiman Was GE’s Muscular Take On The Human-Machine Interface*. (General Electric) Retrieved May 30, 2018, from GE Reports: https://www.ge.com/reports/do-you-even-lift-bro-hardiman-and-the-human-machine-interface/

Keyence Corporation. (2018). *What is a Inductive Proximity Sensor?* Retrieved May 30, 2018, from keyence.com: https://www.keyence.com/ss/products/sensor/sensorbasics/proximity/info/

Khatib, O. (2008). Chapter 5 - Dynamics. In O. Khatib, *Introduction to Robotics* (pp. 125-150). Stanford: Stanford University.

Liew, S. C. (2018, May 30). *Electromagnetic Waves*. Retrieved from Centre for Remote Imaging, Sensing and Processing.: https://crisp.nus.edu.sg/~research/tutorial/em.htm

Lynch, D. K., & Livingston, W. C. (2001). *Color and Light in Nature* (2nd ed.). Cambridge, United Kingdom: Cambridge University Press. Retrieved May 30, 2018, from https://books.google.com.au/books?id=4Abp5FdhskAC&pg=PA231&redir\_esc=y#v=onepage&q&f=false

Merriam-Webster Dictionary. (2018, May 18). *noise*. Retrieved May 30, 2018, from merriam-webster.com: https://www.merriam-webster.com/dictionary/noise

National Instruments. (2018). *PID Theory Explained.* Retrieved June 4, 2018, from NationalInstruments.com: http://www.ni.com/white-paper/3782/en/

Ogata, K. (2010). *Modern Control Engineering* (2 ed.). New Jersey, United States of America: Prentice Hall. Retrieved August 25, 2017

Otaga, K. (2004). *System Dynamics* (4 ed.). Upper Saddle River: Pearson. Retrieved June 4, 2018

Robomart. (2015, November 9). *Advantages and Disadvantages of ultrasonic distance sensor.* Retrieved May 30, 2018, from Robomart: http://roboticsensors.blogspot.com/2015/11/advantages-and-disadvantages-of.html

Siciliano, B., & Khatib, O. (2016). *Springer Handbook of Robotics* (2 ed.). (B. Siciliano, & O. Khatib, Eds.) Berlin: Springer Nature. doi:10.1007/978-3-319-32552-1

Texas Instruments Incorporated. (2017). *Hall effect sensors*. Retrieved May 30, 2018, from ti.com: http://www.ti.com/sensing-products/magnetic-sensors/hall-effect/overview.html

Thomas Publishing Company. (2018). *Capacitive Proximity Sensors*. Retrieved May 30, 2018, from Thomas: https://www.thomasnet.com/articles/instruments-controls/proximity-sensors

Vishay Semiconductors. (2017, February 8). TCRT5000 - Reflective Optical Sensor with Transistor Output.

Yagin, N. (1890, February 11). *United States of America Patent No. 440684.*

Yuhas, D. (2012, May 24). *Speedy Science: How Fast Can You React?* Retrieved from scientificamerican.com: https://www.scientificamerican.com/article/bring-science-home-reaction-time/

ZJIA. (2018, June 1). *Generic YZC-161B 50kg Body Scale Sensor Human Scale Weighing Load Cell Sensor (Pack of 4)* . Retrieved June 1, 2018, from Amazon.com: https://www.amazon.com/Generic-YZC-161B-Scale-Sensor-Weighing/dp/B00MTJ6WZ2

1. Angles should be entered as tuneable parameters, this can then be used to find the variation in PID parameters for the full range of movement. My own analysis has found the variation sufficiently small for the range of human leg movement that PID parameters may be approximated as constant. This would allow for the creation of three independent SISO PID loops, rather than the unholy endeavour of trying to control the exoskeleton with a MIMO system. [↑](#footnote-ref-1)
2. Chapter 6 directly address design by the Root-Locus Method and Chapter 10 speaks directly regarding servo system design (a perhaps more apt framing of the ‘actuators side’ of the project. [↑](#footnote-ref-2)