A Robotic Hand that Utilizes Ergonomic Evaluation as Feedback to Improve Human Robot Collaboration in Soldering Applications

by

Moses Teddy Ort

Submitted to the Department of Mechanical Engineering
in Partial Fulfillment of the Requirements for the Degree of

BACHELOR OF SCIENCE IN MECHANICAL ENGINEERING

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 2016

Signature redacted

Author: _____
Department of Mechanical Engineering
May 6, 2016

Signature redacted

Certified By: _____
H. Harry Asada
Ford Professor of Engineering
Thesis Supervisor

Signature redacted

Accepted By: _____
Anette Hosoi
Professor of Mechanical Engineering
Undergraduate Officer

A Robotic Hand that Utilizes Ergonomic Evaluation as Feedback to Improve Human Robot Collaboration in Soldering Applications

by

Moses Teddy Ort

Submitted to the Department of Mechanical Engineering
on May 6, 2016 in Partial Fulfillment of the
Requirements for the Degree of

Bachelor of Science in Mechanical Engineering

Abstract

People never seem to have enough hands. There are many tools that aim to address this challenge, ranging from the ubiquitous benchtop vise to the "helping hands" commonly used for soldering. However, these tools do not measure up to their human counterparts. They cannot adjust the position or orientation of the workpiece to suit a particular task which can cause workers to maintain unhealthy postures that are detrimental to their long-term health. This thesis addresses this shortcoming with a robotic arm that utilizes a gripper to grasp and hold a workpiece during a soldering task. The robot uses a Microsoft Kinect sensor to continuously analyze the posture of the human worker and calculate a score based on the RULA (Rapid Upper Limb Assessment), an objective measure used in the ergonomics field to evaluate ergonomic working postures. The robot adjusts the workpiece in order to optimize the RULA score using an adaptive simulated annealing algorithm to balance the exploration and exploitation phases of the optimization process. Initial testing indicates that the robot can consistently find positions which improve the RULA ranking by 24.6% of the measured range. This project demonstrates that human robot collaboration can be improved by utilizing sensors to evaluate the needs of a human partner and adjust the robot behavior accordingly.

Thesis Supervisor: H. Harry Asada
Title: Ford Professor of Engineering

Acknowledgements

I would like to first thank my mother, and all of my siblings for always believing in me and encouraging me, first to attend college, and then to transfer to MIT. Without all of your love and support, critiquing of countless resumes, and emergency editing of application essays, I would never be where I am today.

I would also like to express my appreciation to Professor Asada for his mentorship and guidance over the past two years during which I was a part of his research group. Professor Asada sparked my first interest in robotics when I was a student in his class, and then provided me the opportunity to conduct the research for this project, which ultimately led to my decision to continue in this field in graduate school. I would also like to thank Faye Wu who first introduced me to the nuts and bolts of robotics research during our work on her Supernumerary Robotic Fingers, and then continued to provide advice and direction throughout this thesis project. Our weekly meetings were invaluable not only for this work, but also for providing me a framework for conducting research upon which I will surely rely in the future.

I am grateful to Aaron Ross whose expertise and assistance was critical for the ergonomics aspect of this work. Thank you to Ayelet Lobel for being at my side throughout my years at MIT for always being there to help and support me, even when my work seemed to leave little time for anyone else, and for reading this thesis (in its entirety!) and providing insightful and valuable comments.

Finally, I'd like to extend my heartfelt gratitude to all of the amazing faculty, staff, administration, and students, of the Massachusetts Institute of Technology. Thank you for welcoming me with open arms when I transferred in my sophomore year, for providing me with an environment of innovation and invention beyond my wildest expectations, and for allowing me to join a community of some of the kindest and most supportive people that I have ever met. I Have Truly Found Paradise.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background and Prior Work

Robots have become popular in the manufacturing field for their ability to perform repetitive or hazardous tasks with precision and reliability. However, these robots are typically unable to collaborate with human workers in close range as they lack the human ability to infer the intentions of others. Figure 1.1 shows a line of industrial robots at work in a BMW production plant. The conspicuous gates prevent human workers from wandering into the robot workspace and suffering injury. More recently, innovations in controls using series elastic actuators has produced a number of industrial robots that are intended to be safe to work near humans [2]. One example is the Baxter robot made by Rethink Robotics (Figure 1.2). It aims to provide manufacturing facilities more flexibility by allowing Baxter to work near humans [3]. However, Baxter is intended to be trained through demonstration to complete repetitive tasks. It does not actively collaborate with human workers during the task execution.

This thesis examines soldering as a case study for human-robot collaboration. Humans are not well suited to soldering alone, which usually involves holding multiple

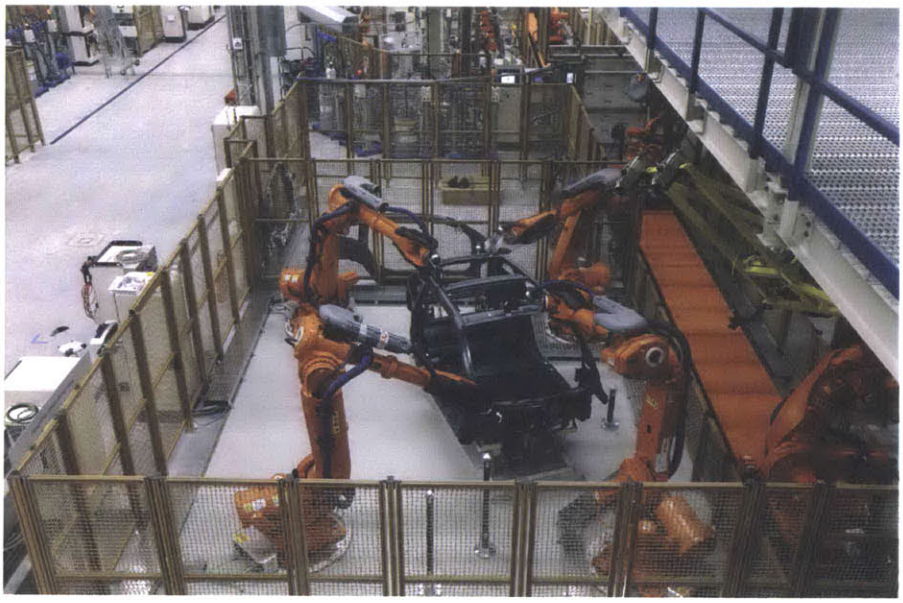

Figure 1.1: An automotive industrial robot at work in a BMW production plant. The gates are intended to prevent humans from wandering into the robot's workspace and potentially suffering an injury. [4]

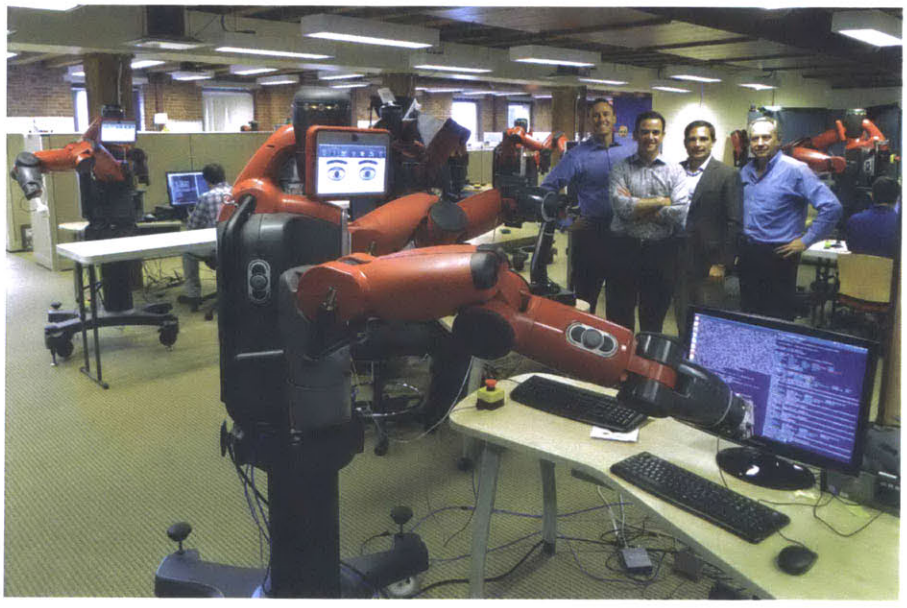

Figure 1.2: The Baxter robot created by Rethink Robotics can safely operate near human workers[3]. Its series elastic actuators and torque control allow it to stop upon making unexpected contact with human without causing injury. [5]

components in precise relative positions while handling a dangerously hot soldering iron. On the other hand, while machines for soldering components do exist, they typically require a substantial amount of time for programming and set up. Consequently, humans are still required to perform soldering tasks when the number of parts needed is not large enough to justify setting up an autonomous system. In this thesis, a method is proposed to allow a robot to collaborate with a worker to complete a soldering task. The robot monitors the worker's posture in order to choose a suitable position for the workpiece without any explicit direction from the worker.

The field of designing better methods of communication between humans and robots is known as Human Robot Interaction [6]. While some techniques make use of natural communication such as gestures and language to understand human intentions, these techniques often suffer from a lack of sufficient accuracy in interpretation of human language or must use a limited vocabulary [7]. Other techniques employ sensors which monitor the human to detect natural poses which indicate task state transitions [8]. In this work, these techniques are expanded to include monitoring and evaluation the human posture to infer from these data an estimate of how the robot should best perform its part of the task. This feedback is then used to adjust the robot's actions in order to optimize the measured parameters. Prior work has investigated the feasibility and accuracy of utilizing a Microsoft Kinect sensor to evaluate a human posture [9]. This technique has been used to automatically measure the ergonomic health of a posture using the Rapid Upper Limb Assessment (RULA) a common ergonomic standard [9, 10]. This project utilizes these techniques to quantify the quality of the human posture in order to provide a feedback metric which the robot uses to evaluate and adjust its own performance. While this particular application seeks specifically to maximize the ergonomic quality of a worker's posture, this technique could be generalized to allow a robot to use feedback to optimize other

performance metrics such as speed at which the human is working or the quality of of the finished parts. Allowing a robot to infer the quality of its job performance in real-time and utilize this feedback to adjust its actions makes for a more collaborative and successful partnership between the human and the robot. This could combine the benefits of the precision, reliability, and safety of a robot with the flexibility of a human operator allowing humans and robots to collaborate on a variety of tasks and thus improve both the productivity and safety of human workers.

1.2 Thesis Overview

The remainder of this thesis describes the project using a bottom-up approach. The individual components described first, followed by the system level architecture, and finally the results and the conclusions that were reached.

In Chapter 2 the design and construction of robotic device is described along with the analysis and motivations behind the various design choices. In Chapter 3 the method by which the robot quantifies and evaluates the human posture is discussed along with the setup of the Kinect sensor which provides the data. In Chapter 4 the simulated annealing optimization algorithm is described. The reason this algorithm was chosen is discussed and the various design choices used in this particular implementation are explained. Finally, in Chapter 5 we transition to the system level architecture. The hardware communication setup including how the various computational components, sensors, and actuators communicate is described. Additonally, the inputs, outputs, and functionality of the various software modules are presented. Chapter 6 gives the results of some tests that were performed to evaluate the functionality of the system and we wrap up with Chapter 7 which provides some suggestions for future work and areas for improvement.

Chapter 2

Device Design and Analysis

2.1 Serial Linkage Design

The robotic arm was designed to be as simple as possible while still achieving the minimum required functionality. The arm contains five degrees of freedom (DOF). Four are used to choose the position and orientation of the end-effector, while the last actuates the gripper. Figure 2.1 shows how the serial linkage was constructed in a manner that is similar to that of a human arm: two DOF for a universal joint at the shoulder, one DOF for the elbow and the last allowing the wrist to rotate. While future iterations could certainly increase the flexibility of the robotic arm by adding two more revolute joints at or near the wrist, which would increase the DOF to six and allow for an arbitrary position and orientation of the end-effector, for this project, it was determined that four kinematic DOF would suffice. These allow the end-effector to be placed at any position in the workspace using the first three DOF while the wrist rotation is primarily used to choose which surface of the workpiece is facing up.

Each of the first three DOF was actuated using a Robotis Dynamixel servo. The

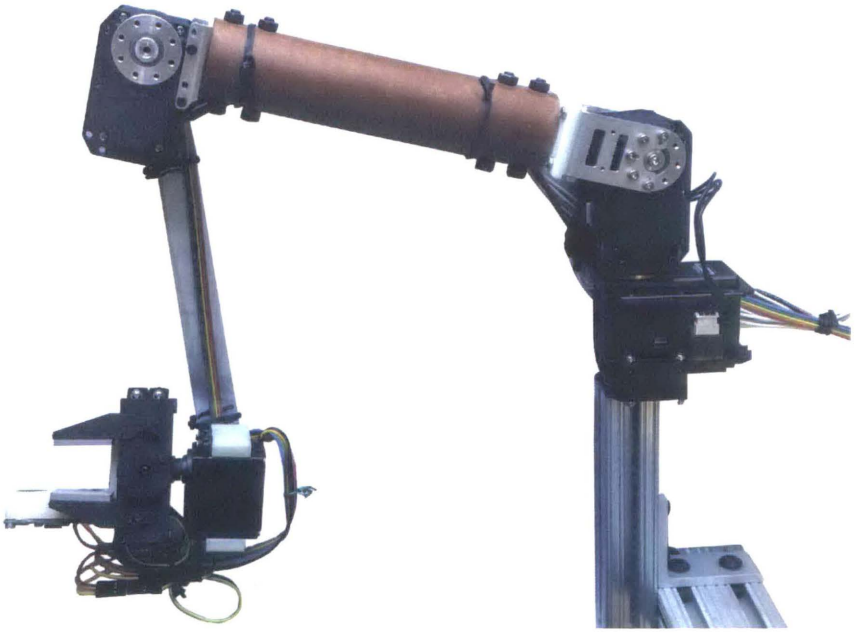

Figure 2.1: The robotic arm. A serial linkage composed of a uiversal joint at the shoulder, revolute joints at the elbow and wrist, and a gripper at the end-effector.

first two actuators in the shoulder joint were the MX-106 model which can supply a maximum torque of 10 N·m. These are mounted so that their axes of rotation crossed perpendicularly at a single point providing a universal joint. The center of rotation of this joint is also on the longitudinal axis of the upper arm link which simplifies the kinematics. Since these actuators are fully supported by the structure, their weight was not a concern. For the elbow joint, the weight of the actuator must be borne by the shoulder joint with a moment arm equal to the length of the upper-arm link. Therefore, a smaller actuator, the MX-64 model weighing 153g, was used in order to decrease the load. The final two actuators, one at the wrist, and one on the gripper were both Hitech HS-422 servos. These servos are much smaller, weighing only 45g. Although they can only provide 0.3 N·m of torque, the torque requirements of these joints are small because the load at the end-effector is transmitted by the structure to the elbow and shoulder joints.

2.2 Kinematics

2.2.1 Forward Kinematics

The robotic arm described above can be parameterized with two parameters, L_1 and L_2, the lengths of the upper-arm and forearm respectively. Figure 2.2 shows a schematic view of the kinematics of the robotic arm. The forward kinematics solution was calculated to in order to determine the position of the end effector $\mathbf{p} = (x,\, y,\, z)^T$ measured with respect to a fixed frame whose origin is at the axis of rotation of the shoulder joint as a function of the three joint angles $\mathbf{q} = (\theta_1,\, \theta_2,\, \theta_3)^T$.

The forward kinematics solution was found to be:

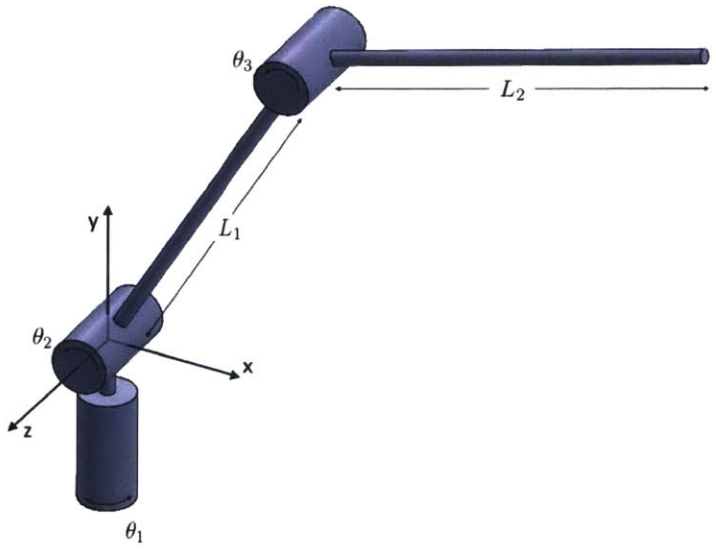

Figure 2.2: A schematic of the kinematics of the robotic arm. The forward kinematics solution gives the coordinates $\mathbf{p} = (x, y, z)^T$ in the coordinate frame shown as a function of the joint angles $\mathbf{q} = (\theta_1, \theta_2, \theta_3)^T$.

$$\mathbf{p} = \begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{bmatrix} \cos\theta_2 \cos\theta_1, & \cos(\theta_2 + \theta_3)\cos\theta_1 \\ \sin\theta_2, & \sin(\theta_2 + \theta_3) \\ \cos\theta_2 \sin\theta_1, & \cos(\theta_2 + \theta_3)\sin\theta_1 \end{bmatrix} \begin{pmatrix} L_1 \\ L_2 \end{pmatrix}$$

2.2.2 The Jacobian

The jacobian was calculated in order to find the workspace of the robot parameterized by the link lengths. This aided in designing the links to achieve the desired workspace. The Jacobian was found from the previous forward kinematics relationship. For brevity, let

$$C_i = \cos(\theta_i), \; C_{ij} = \cos(\theta_i + \theta_j)$$

$$S_i = \sin(\theta_i), \; S_{ij} = \sin(\theta_i + \theta_j)$$

then

$$\mathbf{J} = \frac{d\mathbf{p}}{d\mathbf{q}} = \begin{bmatrix} -L_1C_2S_1 - L_2C_{23}S_1, & -L_1S_2C_1 - L_2S_{23}C_1, & -L_2S_{23}C_1 \\ 0, & L_1C_2 + L_2C_{23}, & L_2C_{23} \\ L_1C_2C_1 + L_2C_{23}C_1, & -L_1S_2S_1 - L_2S_{23}S_1, & -L_2S_{23}S_1 \end{bmatrix}$$

$$\det(\mathbf{J}) = L_1L_2S_3(L_1C_2 + L_2C_{23})$$

Solving for $\det(\mathbf{J}) = 0$ yields:

$$\theta_3 = n\pi, \; n \in \mathbb{Z}$$

Thus the reachable workspace for the robot is independent of θ_1 and θ_2 (it is spherically symmetric) and is singular when θ_3 is either fully extended or fully folded

back which encloses the space between two spherical shells with radii $L_1 + L_2$ and $L_1 - L_2$ respectively.

2.2.3 Workspace

As seen from the jacobian the robot has a workspace enclosed between two spherical shells. However, for the purpose of collaborating with a human worker, the required interaction space was chosen as a rectangular region centered in front of the person. Figure 2.3 shows the bounds of the interaction space which ensured that the robotic arm would not invade the personal space of the worker, collide with the work table, or move too far to the sides to be within human reach. The link lengths were chosen to be $L_1 = L_2 = 23$ cm in order to ensure that the required interaction space was fully contained within the workspace of the arm. These lengths are 2 cm longer than strictly required for reach in order to keep the interaction space free from singularities.

2.2.4 Inverse Kinematics

The inverse kinematics solution was required in order to control the position of the gripper in cartesian space. It is expect that the worker would react to changes in this space, for example to prefer certain regions of cartesian space for a particular task. Therefore, all of the control and optimization of the robot positioning was done in cartesian coordinates and the inverse kinematics was used as the last step to transform a desired destination point into a joint configuration command. The inverse kinematics was calculated from the geometry by first calculating the following intermediate values:

$$L = \sqrt{x^2 + y^2 + z^2}$$

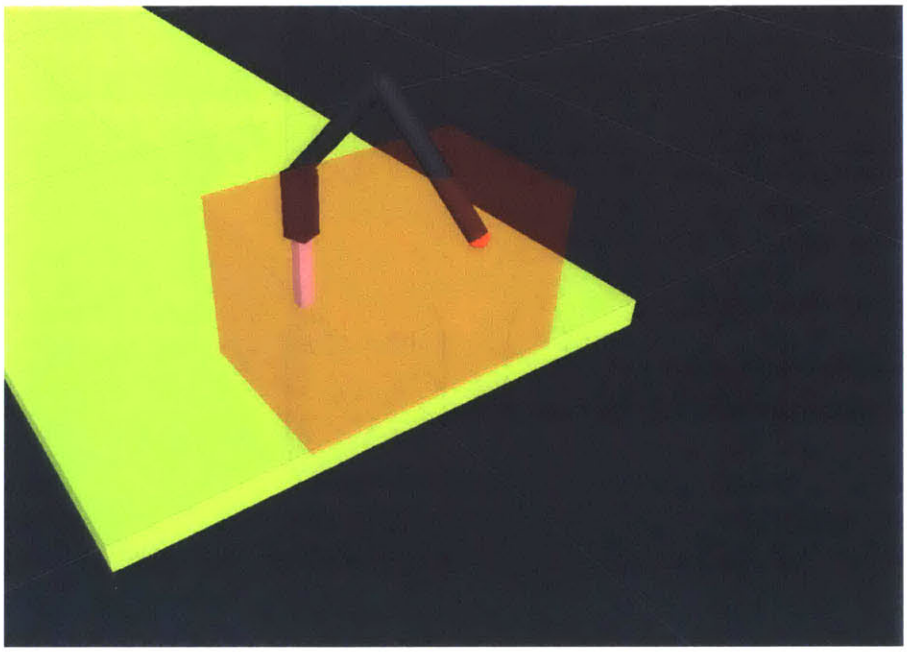

Figure 2.3: Designing the robot workspace. The red-shaded rectangular region is the space in which the human will interact with the robot when seated at the work table. The robot links were designed to ensure that this space is fully enclosed within the spherical workspace of the robotic arm.

$$a = \tan^{-1}\left(\frac{z}{L}\right)$$

$$b = \cos^{-1}\left(\frac{L_1^2 - L_2^2 + L^2}{2L_1L}\right)$$

$$c = \pi - \cos^{-1}\left(\frac{L_1^2 + L_2^2 - L^2}{2L_1L_2}\right)$$

Then the inverse kinematics has two physically unique solutions. The elbow up solution is:

$$\mathbf{q} = \begin{pmatrix} \theta_1 \\ \theta_2 \\ \theta_3 \end{pmatrix} = \begin{pmatrix} \tan^{-1}\left(\frac{y}{x}\right) \\ a + b \\ -c \end{pmatrix}$$

while the elbow down solution is:

$$\mathbf{q} = \begin{pmatrix} \theta_1 \\ \theta_2 \\ \theta_3 \end{pmatrix} = \begin{pmatrix} \tan^{-1}\left(\frac{y}{x}\right) \\ a - b \\ c \end{pmatrix}$$

The entire interaction space is accessible with the elbow-up solution alone, whereas the elbow-down solution could collide with the worktable. Therefore, the elbow-up solution was used exclusively throughout this project. Furthermore, the solutions which involve, for example, rotating θ_1 by 180° are physically equivalent to the previous solutions because the links are symmetric so they were not considered.

2.2.5 Gripper

A Lynxmotion parallel gripper as shown in Figure 2.4 was attached at the end-effector of the robot arm. This gripper was chosen because the parallel motion of the fingers is ideal for grasping the flat circuit boards used for soldering. Furthermore, the rack-and-pinion mechanism reduces the back-drive load on the servo. Finally, the gripper

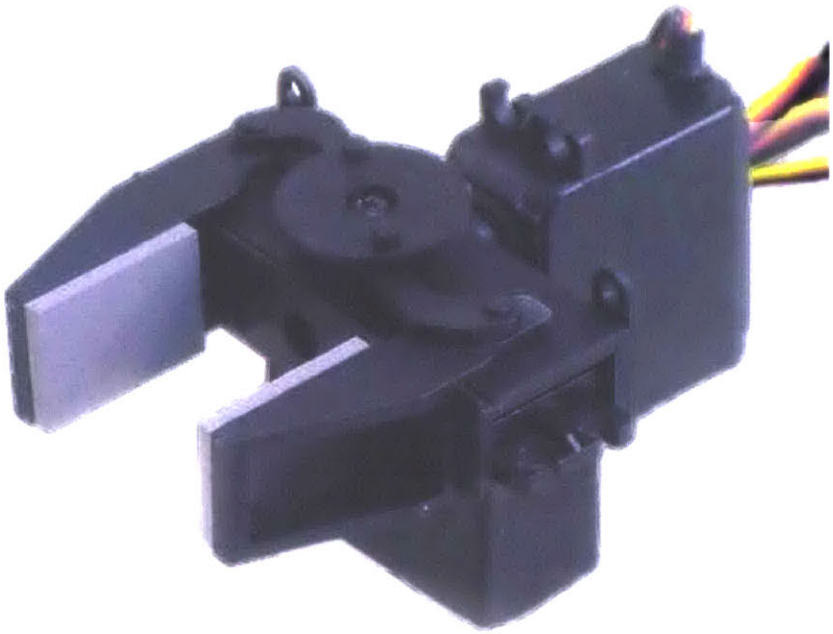

Figure 2.4: A Lynxmotion parallel gripper was attached at the end-effector of the robotic arm for grasping the workpiece.

also includes an attachment for the rotating wrist servo which allows mounting both the wrist rotation and gripper actuators at a single gripper attachment point.

2.3 Sensor Selection

2.3.1 Proximity Sensor

A Parallax Ultrasonic Distance sensor was mounted on the gripper in order to sense the presence of a workpiece. This sensor provides non-contact measurement from 2 cm to 3 m by measuring time between pings. The distance is calculated using $d = 2vt$ where the d is the distance to the reflecting object, $v = 343\,\frac{\mathrm{m}}{\mathrm{s}}$ is the speed of sound in

air, and t is the duration between the sending of a ping, and its detection. In order to convert this measurement into a binary value indicating the presence of a workpiece, a threshold was used. Since the distance to the gripper was 3 cm from the sensor, a threshold of 5 cm was found to be large enough to guarantee detection, while still small enough to avoid false positives.

2.3.2 Solder Stand Sensor

The solder stand sensor is a binary sensor for determining if the soldering iron is currently in the soldering stand. In order to detect this information, the soldering stand was connected to an Arduino microcontroller board through a 10 kΩ resistor pulled up to 5.0 V. In order to meet Electrostatic Discharge safety requirements, soldering irons typically have the tip of the iron connected to ground. This provided a convenient method for detecting the location of the iron without requiring the attachment of any additional sensors. When the soldering iron contacted the conductive aluminum stand, it grounds the stand and thus pulls the Arduino pin down. This information was then transmitted to the main robot computer over the serial connection. Since the wire brush used to clean the solder tip was also conductive, cleaning the tip could also be detected using this sensor. This provided the robot with a way of detecting that the worker was taking a break to clean the soldering iron between joints, which was used by the robot as a signal that it was a good opportunity to adjust the position if necessary. The only caveat was that the soldering stand used had a protective coating which was not conductive and thus had to be removed at the junction between parts in order to ensure a reliable electrical connection throughout.

2.3.3 Temperature Sensor

One of the additional applications explored was the ability for the robot to assist with soldering by feeding solder wire through a tube directly onto the tip of the hot iron. Since it would be undesirable for the robot to attempt to feed solder onto a cold iron, a temperature sensor was used to allow the robot to know when the iron was at a hot enough temperature to begin tinning the tip in preparation for soldering a joint. In order to determine the temperature of the soldering iron tip, the voltage across a K-type thermocouple was measured since the soldering iron already had such a thermocouple embedded in the iron tip in order to keep the temperature at the desired setting. The terminals for the K-type thermocouple were simply connected in parallel to the main soldering unit control board, and a separate amplifier was used to increase this voltage from the millivolt range at which it operates to the Arduino range of $0-5\,\mathrm{V}$. Figure 2.5 shows the soldering station with the added BNC connector circled in red. This connector allowed the robot to determine the soldering iron temperature in parallel with the soldering unit without the need for an additional temperature sensor.

Figure 2.5: The BNC connector circled in red was added to the soldering station unit. It was connected in parallel to the existing K-type thermocouple embedded in the tip of the soldering iron. This allowed the temperature of the soldering iron to be measured without the need for an additional sensor.

Chapter 3

Posture Measurement and Evaluation

3.1 Rapid Upper Limb Assessment

In order to evaluate the posture of the worker to detect positions that are healthy and comfortable, the robot continuously performs a Rapid Upper Limb Assessment (RULA), a standard assessment in the ergonomics industry[1]. The RULA assigns a score between 1 and 7 based on the positions of the upper limbs of a worker during a task. While there exists a more general form of assessment known as the Rapid Entire Body Assessment (REBA) [11], which includes more joints, the RULA was used in this case because the worker is presumed to be sitting at a workbench and thus only the upper limbs are affected.

The RULA assesses two primary areas of upper limb ergonomic health: Group A includes the arms and wrists while Group B includes the neck and trunk. The scores in these two groups are first computed individually based on the relative joint angles. Figure 3.1 shows, for example, how the subscore for the upper arm is calculated. The lowest score of 1 is assigned to joint angles which involve minimal ergonomic risk with increasing scores indidcating higher risk positions. Additional criteria are also taken

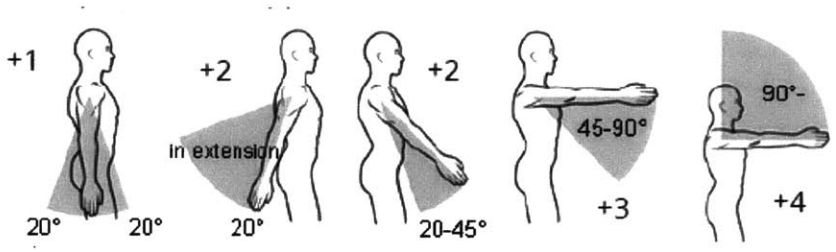

Figure 3.1: Assessing a RULA subscore for the upper arm [12]. The subscore for the upper arm is calculated by measuring the joint angle in the sagittal plane. The lowest score of 1 is assigned to the joint position with minimal ergonomic risk.

into account such as a heavy load or repeated motions. Once the score in the two groups is computed, they are combined to form a single holistic score in the range 1-7. Table 3.1 gives the meaning for the scores as described in [1].

For this project, the complete RULA score was not measured because some joint angles such as the wrists proved difficult to obtain visually do to occlusion by the soldering tool, Additionally, some of the other measurements were not applicable such as the loading force which we assume is small for soldering applications. The parameters that were measured included: neck angle, trunk angle, right and left upper arm angles, right and left elbow angles, as well as right and left shoulder positions with respect to the base of the neck. These measurements were used to compute a subset of the RULA score which ranged from 1-4. Since this subset has a different range than the complete RULA score it is used only for comparing observed postures on a relative basis and cannot be used to infer the the true total score. Therefore, while we cannot determine based on the subset measured into which range in Table 3.1 a given posture would fall, we can compare two postures and determine which one has a lower overall score and by improving the components of the score which the robot could observe, infer that this would increase the total score as well. Furthermore, subjective testing showed that more comfortable postures tended to get better scores

RULA Score Range	Required Action
1-2	Posture is acceptable if it is not maintained or repeated for long periods.
3-4	Further investigation is needed and changes may be required.
5-6	Investigation and changes are required soon.
7	Investigation and changes are required immediately.

Table 3.1: Interpreting a RULA score. The required action to take as recommended in [1].

as measured in this subset, while less comfortable postures that required excessive leaning or twisting also received higher (worse) scores in the measured subset of the RULA.

3.2 Kinect Sensor

In order to measure the required joint angles to compute the RULA score, a Microsoft Kinect sensor was used to monitor the human worker. The viability of using a Kinect sensor to estimate ergonomic information was investigated in [9, 10] they find that the ideal location for the Kinect is at an angle to the subject between $20° - 45°$. As shown in Figure 3.2 the Kinect was placed at a distance of 1.5 m in front of the worker's position and at an angle of 20°. The angle also helped ensure the sensor obtained an unobstructed view of the worker's upper body despite the robotic arm mounted directly in front of the worker. The transform between the camera frame and the base frame for the robotic arm ${}^{base}T_{kinect}$ was measured in order to transform all joint measurements into the robot reference frame.

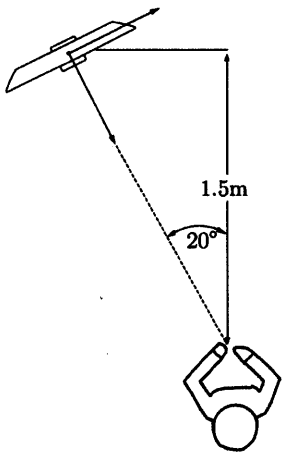

Figure 3.2: Position of the Kinect Sensor with respect to the worker.

Chapter 4

Closing the Loop: Optimization of Workpiece Position

4.1 Formulation of the Optimization Problem

Using the RULA score for worker's posture obtained in real time as described in the previous chapter, the problem of finding the optimal position for the workpiece can be reduced to a constrained optimization problem where the robot must find the position that minimizes the objective function $f(\mathbf{x})$ which gives the RULA score over the parameter $\mathbf{x} = (x, y, z)^T$ which represents the position of the end-effector in Cartesian coordinates. Formally:

$$\arg\min_{\mathbf{x}} f(\mathbf{x})$$

$$\text{subject to: } \mathbf{x} \in \mathbb{R}^3 : \underline{\mathbf{x}}_i \leq x_i \leq \overline{\mathbf{x}}_i,\ i = 1..3$$

where l and u are vectors that define the lower and upper limits of the interaction space in which the robot and human operate. The values were $\underline{\mathbf{x}} = (-0.2, -0.2, 0.05)^T$ m

and $\overline{\mathbf{x}} = (0.05, 0.2, 0.3)^T$ m as shown by the red shaed region in Figure 2.3.

There are many choices of algorithms for finding solutions to optimization problems such as this and a detailed treatment of the subject is beyond the scope of this thesis. However, there are certain features unique to this problem which motivated the particular solution used here. One key point is the expense of sampling the objective function. While many algorithms assume hundreds or thousands of points can be sampled in order to find a very accurate solution, in this problem, obtaining each sample requires that the human worker complete a task at a given position and that the RULA score for the posture during that time be calculated. Thus, it is infeasible to require a large number of samples before converging to an optimal solution. This also highlights another concern that is especially important for this application: the tradeoff between exploration and exploitation. Once some information is known, the benefit of exploiting that knowledge to increase the likelihood of choosing a good position for the next task must be balanced with the possibility that exploring new areas of the parameter space might lead to the discovery of an even better solution. One final consideration is our prior knowledge of the qualities of the objective function. Since the human has many more degrees of freedom than the three dimensional space of the parameter, we expect there to be many possible human postures for a given workpiece position. Furthermore, the posture could switch between solutions at discrete points such as changing from an elbow-down to elbow-out posture once the joint limit is reached. Therefore, there is no expectation that the function will be smooth, or even continuous which makes it more difficult to use gradient based methods.

4.2 The Simulated Annealing Algorithm

To solve the optimization problem, an adaptive simulated annealing (ASA) algorithm was used. This algorithm is particularly suitable for this problem based on the requirements described above. In particular, the temperature and cooling schedule provide a built-in way of balancing the exploration vs. exploitation requirements. Additionally, ASA does not assume the function is convex thus with suitable parameters it can find the global optimum even in the presence of local minima. Finally, it does not require knowledge of the gradient or continuity of the objective function.

4.3 Designing the Alogrithm Functions and Parameters

4.3.1 The Acceptance Function

For the simulated annealing algorithm to work for this application a number of choices and parameters had to be chosen. The acceptance function used is a commonly used function known as the Metropolis acceptance function which is defined as follows:

$$F(f, f_{i+1}, T) = \begin{cases} \text{True}, & \text{if: } f_{i+1} \leq f_i \\ \text{True}, & \text{if: rand}() < e^{-\frac{f_{i+1}-f_i}{T}} \\ \text{False}, & \text{othewise} \end{cases}$$

where f is the previous value of the objective function. f_{i+1} is the candidate value at the new sample point, T is the current temperature, and rand () is a random variable drawn from a uniform disribution on the interval [0, 1].

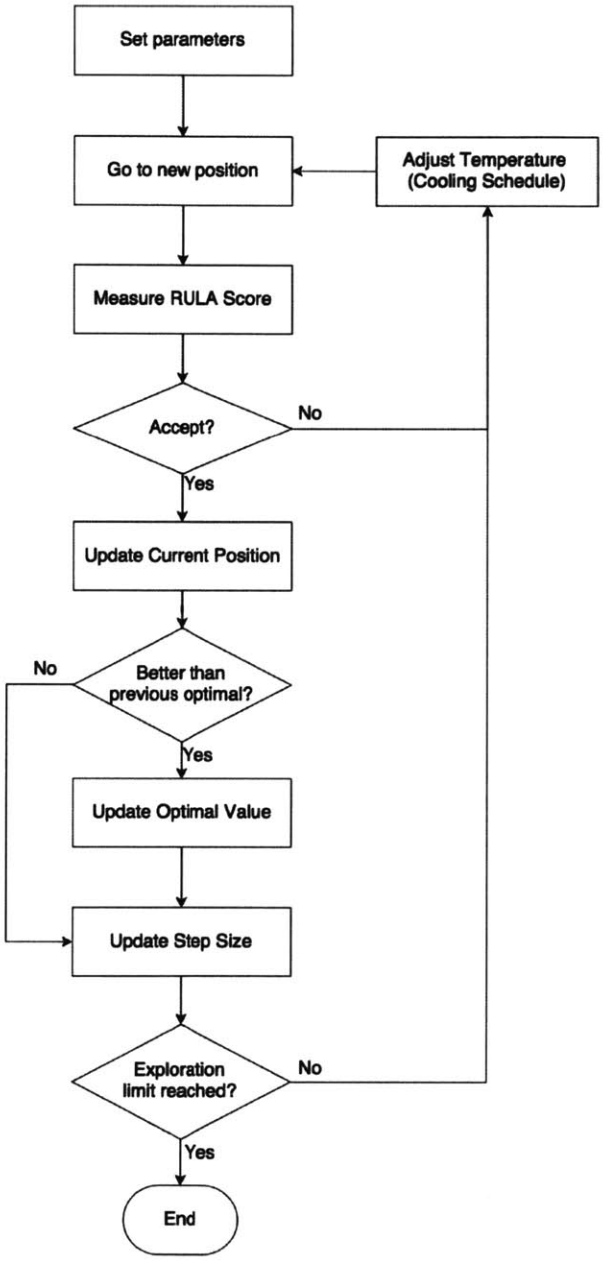

Figure 4.1: Adaptive Simulated Annealing optimization algorithm.
This algorithm is used to find the position which optimizes the time average of the RULA score during each sample period.

4.3.2 The Adaptive Step Function

For the adaptive step function, the step size was applied to each of the three coordinate axes independently allowing the step to shrink faster along axes which were moving quickly and more slowly along axes that did not have many accepted samples [13]. The step size is as a diagonal matrix $\mathbf{D}$ with with the current step sizes for each axis along the diagonal entries. A new sample point is generated as follows:

$$\mathbf{x}_{i+1} = \mathbf{x}_i + \mathbf{D}\mathbf{v}$$

where $\mathbf{v}$ is a vector of random variables drawn from a uniform distribution on the interval $[-1, 1]$. Each time a sample is accepted, the step size is updated such that:

$$\mathbf{D}_{i+1} = (1 - \alpha)\,\mathbf{D}_i + \alpha\omega\mathbf{R}$$

where α is a parameter that determines how strongly the successful sample should affect the step size, $\mathbf{R}$ is a diagonal matrix where the diagonal elements are taken from the vector $\mathbf{x}_{i+1} - \mathbf{x}_i$, and ω is a scaling factor appropriate for $\mathbf{R}$. The initial step size $\mathbf{D}_0$ was set with the diagonal values taken from the vector $\beta\,(\overline{\mathbf{x}} - \underline{\mathbf{x}})$ such that β is a constant that gives the proportion of the initial step size to the limits of the interaction space defined previously in Section 4.1.

4.3.3 The Cooling Schedule

The cooling schedule determines the rate at which the algorithm settles on an optimal value, if it cools too quickly, the algorithm could get stuck and converge to a local minimum, while if it cools too slowly, it effectively becomes a random search which

Parameter	Symbol	Value
Initial Temperature	T_0	1
Step Size Rate	α	0.1
Step Size Adjustment Ratio	ω	1.5
Cooling Schedule Ratio	r	0.9
Initial Step Size Ratio	β	0.75

Table 4.1: Tuned parameter values for the adaptive simulated annealing optimization algorithm.

doesn't converge at all. The cooling was calculated as follows:

$$T_{i+1} = rT_i$$

where the initial temperature T_0 is a parameter.

4.3.4 Tuning the Parameters

In order to choose approprate values for the parameters described above, simulation was used to test the algorithm on various functions. There is always a tradeoff between the quality of the result and the number of iterations for which the algorithm runs. Since each iteration required the completion of a soldering task, twenty was chosen as the maxium allowable number of iterations for the alogrithm to converge. With this goal, the parameters were tuned in sumlation to find the values that would give the appropriate tradeoff between exploration and exploitation in that range. An initial testing phase consisiting of five runs with twenty samples in each run was then used to further fine tune the parameters. The resulting tuned values for these parameters are listed in Table 4.1.

Chapter 5

System Architecture

5.1 Putting it all Together

The hardware schematic in Figure 5.1 shows how the various system components communicated. There are three computational units: 1) The windows PC used exclusively for managing the Microsoft Kinect sensor 2) The Arduino microcontroller which performed low-level management of the various sensors and the wrist actuators, and 3) The main Ubuntu PC running the Robot Operating System (ROS). The functionality of each of these is described in the following sections.

5.2 The Windows PC

The Windows PC serves as the processing unit for the Microsoft Kinect. The Kinect sensor is compatible with the Natural User Interface (NUI) library provided by Microsoft which is only compatible with a PC running Microsoft Windows. Since ROS is only compatible with UNIX based operating systems, this was the primary motivation for using two computers. However, since the Kinect sensor collects a very large

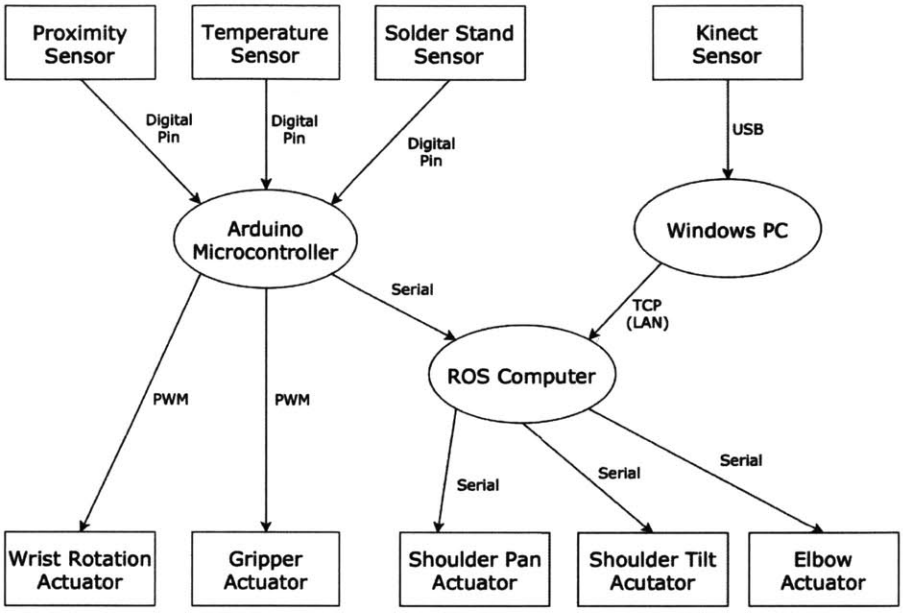

Figure 5.1: A hardware schematic showing how the various sensors and actuators communicated. The three computational units are shown as ellipses while the sensors and actuators are rectangular. The directed arrows denote communication and are labelled with the protocol used.

amount of data at a high rate, and processing this data to segment and detect human joints in the images can be very processor intensive, there is a benefit to using a dedicated PC for this task. The resulting joint angles are streamed over the network as a simple array of numbers which has very little overhead in comparison to the image segmentation task.

5.3 The Arduino Microcontroller

The Arduino microcontroller is primarily used for low-level sensor management. In particular, it monitors the voltage of the solder holder to detect the location of the soldering iron, and sends a status update to the ROS computer over a USB serial connection. The temperature of the soldering iron is also monitored by measuring the analog voltage of the output of the K-Type thermocouple amplifier. The voltage is

read using the built-in analog-to-digital converter of the Arduino Nano, converted to degrees Celsius, and sent to the ROS computer at a rate of 10 Hz. Next, the proximity sensor discussed above is also managed by the Arduino, which sends out pings lasting $5\,\mu s$, times the delay, calculates the distance measurement, and then transmitts it to the ROS computer at a rate of 10 Hz. Finally, the Arduino is also responsible for driving the two small HS-422 servos in the robot wrist. Upon receiving a joint command from the ROS computer, the Arduino generates a Pulse Width Modulation (PWM) signal to drive the servo to the desired position.

5.4 The ROS Computer

The ROS computer serves as the main controller for the robot. Figure 5.2 shows how the various modules (*ROS nodes)* interact in order to achieve the desired behavior. The sensors attached to the arduino provide their respective measurements directly to the main Controller node. The Robot Joint State Publisher node polls the Dynamixel motors at a rate of 20 Hz to obtain joint angle measurements and makes that information available to the Controller node as well. Next, the Human Skeleton State Publisher listens over a TCP Socket for human sekeleton updates from the Windows PC. Once an update arrives, it publishes a vector of Human Skeleton Joint Positions. The RULA Score Evaluation node takes these skeleton joint positions as input and evaluates a RULA score from these as described in Section 3.1. The RULA Time Average Calculation node maintains an average RULA score over the course of a task evaluation. Each time a task is begun, the average is reset in order to independently measure the time average for a given robot position. Finally, this average score is provided to the controller which performs the adaptive simulated annealing optimization in order to select the optimal choice for the robot position.

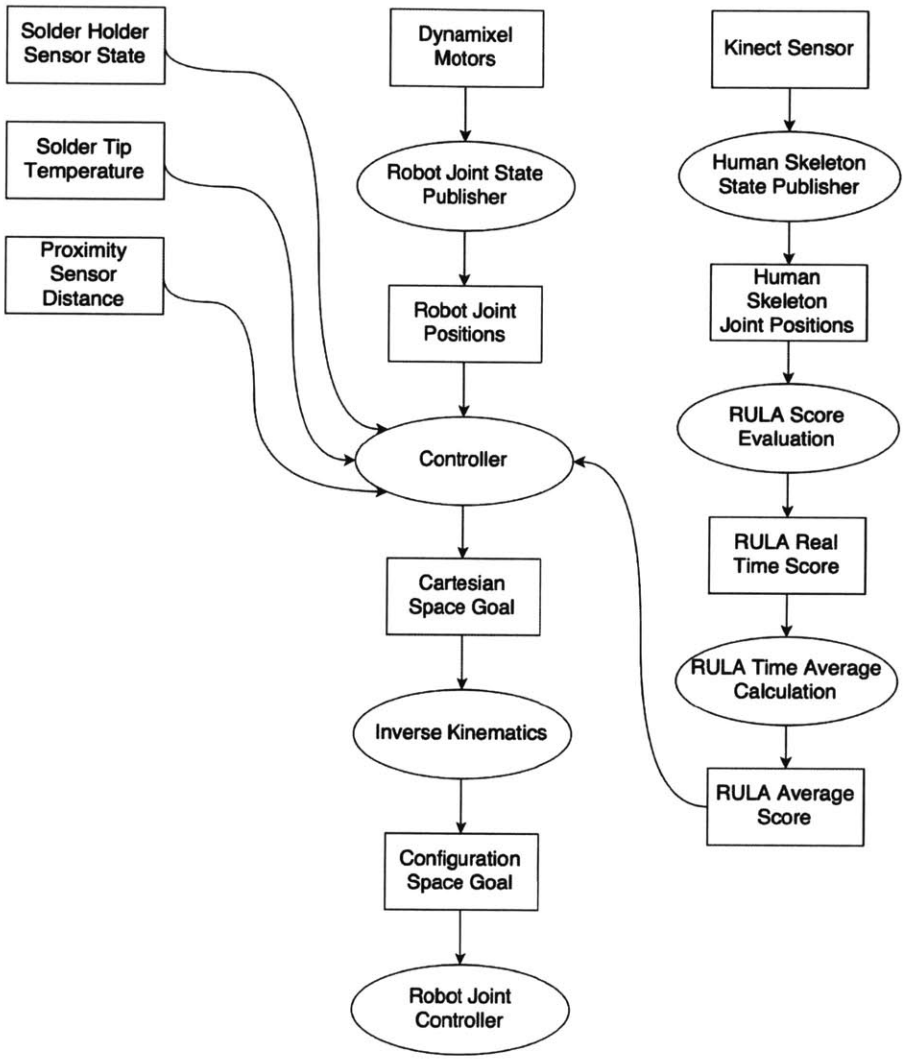

Figure 5.2: ROS node graph. Each ellipse represents a ROS node or module while the rectangles show the inputs and outputs.

Chapter 6

Testing and Results

6.1 Testing Setup

In order to test the system performance using a typical application, the task of soldering a 20 pin header to a circuit board was performed for a total of 11 runs. Each run was independent in that no information was saved between runs. However, within each run, the state was saved after each joint was completed allowing the system to optimize the position over the course of twenty samples. Twenty samples was chosen as a quantity small enough that it would not be overly cumbersome to require a worker to perform twenty joints as the robot attempted to find the optimal position while still being large enough to reasonably expect the optimization algorithm to converge. For each run, the following steps were taken:

1. The robot chooses a random position in the workspace and waits with an open gripper to receive the circuit board.

2. The worker hands the circuit board to the robot which grasps it with the gripper.

3. The worker places a 20 pin header into the top of the circuit board.

4. The worker removes the solder from the holder which signals the robot to rotate the circuit board to have the solder side face up.

5. The worker completes a joint by placing the iron on the pin and dispensing molten solder.

6. The worker cleans the tip of the solder or places it back into the solder holder, which signals the robot that it is an opportune time to consider a change of position.

7. The robot measures the human's posture during the course of the previous task, as well as its current position in the workspace and adds this sample point to the optimization algorithm.

8. The optimization algorithm gives a new point to sample, if the algorithm is in the early exploration stage, this could be anywhere in the workspace, if the algorithm has already converged this would be the optimal location.

9. The robot moves to the new location if necessary and resets the posture score to begin a new measurement.

10. The process is repeated until the header is complete at which point the data for the posture score at every sample point is saved and the run is complete.

6.2 Results

Figure 6.1 shows the results of the 11 test runs as each of the colored lines. The thick black line is the average RULA score at each sample number. As expected, the individual runs behave differently with some finding optimal scores relatively quickly while others explore for a longer number of samples. Furthermore, the average RULA

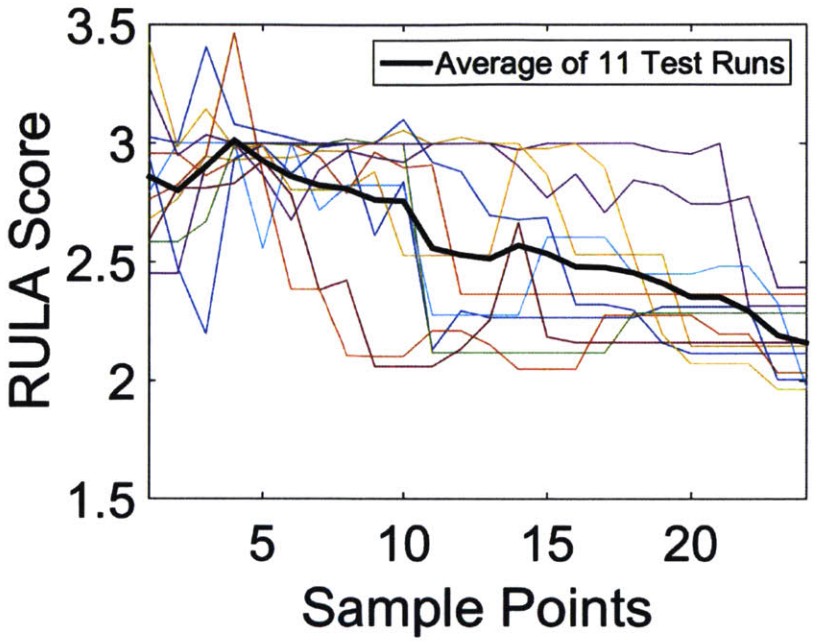

Figure 6.1: Results of eleven test runs. The RULA score is shown on the vertical axis while the horizontal axis denotes sample number. The thick black line shows the average of the all eleven runs at each sample point. This average decreases over the course of each run even though the individual runs take different paths toward the lower RULA scores. This is expected due to the random nature of the adaptive simulated annealing optimization algorithm used.

score continues to decline as the number of samples increases. Notice that during the first four samples, the average does increase a small amount. However at the beginning the algorithm is essentially a random sampling of the space and so it is unsurprising that it isn't strictly decreasing in the earliest stages. The total change of the average, amongst the 11 test runs was 0.74 RULA points. Since the subset of the RULA score measured had a range of 3 RULA points (from $1-4$), this yields a decrease of 24.7% of the measured range.

6.3 Discussion

The goal of this project was to build a robotic gripper that could better assist a human partner through real-time feedback by evaluating the posture of the human and assimilating that information into its decision making for the future. The previous results demonstrate that such a strategy can sucessfully lead to an increase in the quality of the assistance the robot provides without any explicit direction from the human. It is anticipated that this concept could be applied to additional tasks where, as long as a method is devised to allow the robot to evaluate a quality score of its performance thus far, it could learn to increase that score using this method.

Chapter 7

Future Work and Improvements

One of the key components of this work was measuring the quality of a human worker's posture and using it to evaluate the RULA score. This was accomplished solely using the Microsoft Kinect sensor. While limiting the posture evaluation to this method had the benefit of minimal interference with the worker, there were several drawbacks. Firstly, only a subset of the full RULA score could be measured reliably due to the visual occlusion that occurs at the wrist joints when a tool is used. Furthermore, even those joints which are not blocked by the tool, could sometimes become occluded by other limbs. Future work could incorporate additional sensors to solve the occlusion problem. These could include additional cameras that view from another angle or even other types of sensors such as Inertial Measurement Units or Strain sensors that could be used in conjuction with, or in place of, the visual method. Secondly, while the RULA is a standard score for ergonomic health, it could also be valuable to measure more subjective metrics of the robot's performance by asking the human for feedback periodically. While it is certainly important to maintain a healthy posture from an ergonomics perspective, it would also be beneficial for the robot to favor actions that engender a positive feeling for its performance in the mind of the human

worker.

Additionally, while this project tested the specific task of soldering a printed circuit board, future work could test this method in a variety of other scenarios including:

- Assembly and Disassembly
- Sorting and Counting
- Loading and Unloading

Theses are just some of the examples where a robotic arm with a standard gripper could assist a human worker. If the robot is equipped with specialty tools, the possibilities are truly endless. For each task, the quantity measured would also differ. For example, rather than measuring worker posture, the time taken for completion or the quality of a finished workpiece could be measured. The robot would then use this feedback to choose optimal strategies that maximize the score in those areas.

Finally, more testing is needed especially on untrained workers to quantify the robot's performance amongst users who do not have the same level of experience with robots as that of robotics researchers. A full series of experiments in which users with comparable skill levels to a factory worker are compared when working with the robot to a control group completing the same tasks alone is necessary in order to quantify how much such a robot could assist human workers in the completion of real-world tasks.

Bibliography

[1] L. McAtamney and E. N. Corlett, "RULA: A survey method for the investigation of work related upper limb disorders," *Applied Ergonomics*, vol. 24, no. 2, pp. 91–99, 1993.

[2] G. A. Pratt and M. M. Williamson, "Series elastic actuators," *1995 IEEE/RSJ International Conference on Intelligent Robots and Systems. 'Human Robot Interaction and Cooperative Robots'*, vol. 1, no. 1524, pp. 399–406, 1995.

[3] C. Fitzgerald, "Developing baxter," in *IEEE Conference on Technologies for Practical Robot Applications, TePRA*, 2013.

[4] BMW PressClub, "Bmw i3 production bmw plant leipzig: Cfrp body shop," 2013. https://www.press.bmwgroup.com/global/photo/detail/P90127374/bmw-i3-production-bmw-plant-leipzig-cfrp-body-shop-09-2013.

[5] Wikimedia Commons, "Bmw i3 production plant," 2013. https://commons.wikimedia.org/wiki/File:Caught_Coding_(9690512888).jpg.

[6] M. a. Goodrich and A. C. Schultz, "Human-Robot Interaction: A Survey," *Foundations and Trends® in Human-Computer Interaction*, vol. 1, no. 3, pp. 203–275, 2007.

[7] R. Boboc, "Natural Interaction with an Assistive Humanoid Robot.," *Applied Mechanics & ...*, 2015.

[8] B. Llorens-Bonilla and H. H. Asada, "Control and Coordination of Supernumerary Robotic Limbs based on Human motion Detection and Task Petri Net Model," in *Dynamic Systems and Control Conference*, pp. 1–7, 2013.

[9] P. Plantard, E. Auvinet, A.-S. Pierres, and F. Multon, "Pose Estimation with a Kinect for Ergonomic Studies: Evaluation of the Accuracy Using a Virtual Mannequin," *Sensors*, vol. 15, no. 1, pp. 1785–1803, 2015.

[10] L. G. Wiedemann, R. Planinc, and M. Kampel, "Ergonomic-Monitoring of Office Workplaces Using Kinect," *Ambient Assisted Living and Daily Activities*, vol. 8868, pp. 275–278, 2014.

[11] L. McAtamney, "REBA: A Survey Method for the Investigation of Work-Related Upper Limb Disorders," *Applied Ergonomics*, vol. 24, no. 2, pp. 91–99, 1993.

[12] A. Hedge, "RULA Employee Assessment Worksheet," 2001.

[13] F. Busetti, "Simulated annealing overview," *JP Morgan, Italy*, no. 1, pp. 1–10, 2003.