Towards Image-based Control of an Industrial Potato Peeler

by

Zachary Knopp

Bachelor of Science in Mechanical Engineering, UNB, 2014

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF

Master’s of Science in Mechanical Engineering

In the Graduate Academic Unit of Mechanical Engineering

Supervisor(s): Rickey Dubay, PhD. Mechanical Engineering
Dhirendra Shukla, PhD. Entrepreneurial Finance

Examining Board: Amirkianoosh Kiani, PhD. Mechanical Engineering, Chair
Phil Garland, PhD. Mechanical Engineering
Monica Wachowicz, PhD. Geography

This thesis is accepted

Dean of Graduate Studies

THE UNIVERSITY OF NEW BRUNSWICK

October, 2016

©Zachary Knopp, 2017
Abstract

Difficult multivariate industrial control problems can be solved by combining standard control theory, adaptive computer vision algorithms and intelligent modeling into an overarching generalized control system. Creating the foundations for such a system, to be implemented on an industrial potato peeler, was the scope of this thesis. Computer vision algorithms that provided quantifiable metrics from the peeling process were developed using data gathered at a potato research center. Experiments were performed controlling the steamtime and pressure of the peeler and the size and seasonality of the potatoes. Thermal signatures and optical videos of peeled potatoes were recorded throughout testing. It was found that smaller potatoes are more difficult to peel and changes in pressure do not correlate with changes in peel efficiency. Recommendations were made for the next steps towards intelligent control of industrial potato peeling processes.
Acknowledgements

First and foremost, I would like to thank my supervisor, Dr. Rickey Dubay. He gave me the freedom to pursue my interests and the support I needed to excel. I feel extremely fortunate to have had the opportunity to work with him on so many different projects. I also have to thank the other lab members for their help, from editing things I wrote to helping me build robots. Furthermore, I would like to thank the team at Eigen Innovations and the team at McCain Foods for their help on the project. Lastly, I would like to thank the Intelligent Controls Lab, MITACS, NBIF and NSERC for their generous funding.
# Table of Contents

**Abstract**  

**Acknowledgments**  

**Table of Contents**  

**List of Figures**  

## 1 Introduction and Background Theory  

1.1 Introduction  

1.2 Research Objectives  

1.3 Significance and Contributions  

1.4 Basic Control Theory  

1.5 Model Predictive Control Theory  

1.6 Computer Vision Theory  

1.6.1 Overview and Background  

1.6.2 Convolution  

1.6.3 Preprocessing Techniques
1.6.4 Recognition .................................................. 18
1.6.5 Specifics For This Project ............................... 19
1.7 Validating the Approach on a Robotic Manipulator .... 26
  1.7.1 Experimental Setup ..................................... 28
  1.7.2 Computer Vision Feedback ............................. 29
  1.7.3 Control ..................................................... 29
  1.7.4 Results ................................................... 33
  1.7.5 Conclusion ............................................... 40

2 Application Study: Industrial Potato Peeling Process 43
  2.1 Introduction and Background ............................. 43
  2.2 Description of the Process .................................. 45
  2.3 Experimental Setup, Data Collection and Methods ........ 46
  2.4 Developing the Segmentation Algorithm .................. 49
    2.4.1 Background Subtraction ............................... 51
    2.4.2 Seam Detection ......................................... 53
    2.4.3 Hole Filling ............................................ 54
    2.4.4 Find Contours and Perimeter Checking ................. 55
    2.4.5 Contour Splitting ...................................... 56
    2.4.6 Concavity Segmentation ................................ 56
    2.4.7 Segmentation of poorly peeled potatoes ............... 58
  2.5 Methods For Creating a Database ......................... 60
    2.5.1 Classification Methods .................. 60
List of Figures

1.1 Structure for a basic control loop ......................... 6
1.2 Model Predictive Controller structure .................... 8
1.3 General block diagram for a computer vision algorithm .... 12
1.4 Illustration of a binary raster image ...................... 12
1.5 The effects of histogram equalization on a grayscale image . 15
1.6 The effects of a Gaussian blur on an image ................ 16
1.7 The effects of thresholding on a grayscale image, range [120,255] 17
1.8 Chain assignment diagram and an example of a boundary classified by Freeman chain ......................... 22
1.9 Determination of overlap locations using arcs and chords along boundary edges ......................... 22
1.10 One example of potato image segmentation[6] .............. 24
1.11 2DOF planar RR serial robotic manipulator ................ 28
1.12 Plots of $\theta_1$ and $\theta_2$ under CTC control without PID for model comparison ......................... 32
1.13 Schematic block diagram of CTC-E control ................ 33
3.14  Recommended future setup for peeling potatoes  . . . . . . . . 78

4.1  Overall intelligent factory approach for industrial potato peelers  81
Chapter 1

Introduction and Background

Theory

1.1 Introduction

A common theme in industry over the past few decades is the desire to optimize processes through research and development in order to increase profitability. An overarching way in which process optimization occurs is through the development of better automatic control schemes. A survey on various control schemes in Japanese industry highlights that the optimal controller increases system efficiency while also reducing labor and maintenance costs[33]. Model predictive control (MPC) was evaluated with higher labor costs than advanced PID control schemes but it also had a higher score in process efficiency. Hence, integrating process models into advanced controllers
has its benefits. However, if the process is time-varying, multivariable, complex, or non-linear then the formulation of a process model becomes difficult and initial implementation costs can increase significantly.

The formulation of a process model for systems that have limited sensory feedback is particularly difficult, especially for compound complex industrial processes that have a range of unusual output parameters. Compound refers to multiple smaller processes in series as part of the overall process. Complex refers to processes that have high variability and cannot be modeled analytically. For example, imagine a robotic manipulator performing MIG welding. Automated inputs parameters could include things like the feedrate, tip velocity, approach angle and current setting, which are all easily quantifiable. The output, however, is not. The output defined as the quality of a weld has many parameters, such as penetration depth, temperature profile and bead width. No sensors exist that can measure the parameters independently in order to relate changes in inputs to changes in the weld quality. Hence, industrial welding does not have real-time adjustments, instead they are set up once and checked periodically by technicians to make sure the welds are adequate. However, imaging sensors, along with computer vision algorithms, can be used to quantify these key process metrics to make real-time control and quality inspection feasible[4]. This approach has wide applications in industry as shown in [21].

Closed loop control of an industrial potato peeling process is similar to the welding control problem. A peeler has various input parameters such as
pressure, steam time, steam temperature, potato size, seasonality and potato variety, all of which can significantly affect how the potatoes are peeled. Certain settings result in better peeling, but more waste. Other settings result in worse peeling, but less waste. The end goal is to reduce waste while also completely peeling the potatoes. The implementation of batch feedback control on this process can reduce output variability and decrease waste. The first step of which, and the topic of this thesis, is to create quantifiable feedback metrics through the use of imaging sensors in order to create a process model for improved control. The next section highlights specific research objectives that must be met in order to accomplish this goal.

1.2 Research Objectives

The preliminary step towards image-based control of industrial processes is validating the benefits of image-based control. This is covered in section 1.7, where visual control is applied to a robotic manipulator in order to overcome process non-linearities in a non-invasive way. This served as a smaller application study prior to beginning work on the potato peeling investigation and deemed important for setting the stage on the future of image-based control for this.

The next two research objectives are specific to the industrial potato peeler application. The first objective is to develop a method for quantifying the output of the potato peeler in order to use it as feedback for control in future
work. The second objective is to run experiments and analyze the output with respect to the input parameters of the peeler to gain a deeper understanding of the peeling process, which will then shape the future direction of this project and others. Note that the scope of this thesis is limited to validating image-based control, developing quantitative feedback methods for an industrial peeling process and then investigating the process using the new feedback methods. Creating a robust process model that can be updated using a learning algorithm is reserved for future work, hence it is not included in this thesis.

1.3 Significance and Contributions

The significance of this project falls into two different categories: general, to industry as a whole, and specific, to the industrial potato peeling application. In general, the outcome of this work is an additional adaptive segmentation tool for round objects passing on a conveyor belt. This can be applied to the produce grading and packing industry as well as the food processing industry. It will also demonstrate the applicability of a generalized approach to industrial control problems where computer vision algorithms quantify process metrics that are used to train models for better control.

A specific application in the food processing industry, which is also the main application of this thesis, is potato peeling at an industrial scale. The major contribution to this problem is data visualization. It is the idea that a deeper
understanding of the process can come from developing algorithms that pull quality information from potato imaging data. This new information is valuable as it can be used to adjust the input parameters of the peeler based on the expected output of the peeler. At the moment, these parameters are adjusted based on personal experience of plant operators without a foundation of quantitative batch data to back them up.

The contributions of similar work has been seen where imaging sensors and computer vision algorithms replace more traditional feedback mechanisms, or lack thereof, in order to improve, or implement, closed loop control. There are many existing situations where this type of implementation has had positive results. In 1997, temperature feedback from an infrared camera was used to modulate the power of a plasma-arc torch in laminated object manufacturing through closed loop control\[9\]. In 2012, a vision system was adapted to a micro-scale haptic manipulator in order to overcome the lack of positional sensor capabilities. At such a small scale conventional sensors, such as encoders, are not practical options. Instead, the team used a combination of a conventional frame-based camera with a silicone retina in order to track the position of the micro-gripper at a frequency of 4 kHz. They were able to track and control the gripper for pick and place operations at the micro-scale using a haptic interface for tele-operation\[25\].

The remainder of this chapter discusses the control and computer vision theory relevant to this thesis, as well as a preliminary application study on a robotic manipulator.
1.4 Basic Control Theory

The goal of automatic control is to create self-regulating systems without the need for direct intervention when the desired reference input, the set point, changes. The basic structure for a closed-loop system is given in Fig. 1.1.

There are many variations of this basic control loop, all of which attempt to increase system performance. Open loop control occurs without sensor feedback and consists of simply sending commands to the system based on the reference input. An example of this type of control can be seen in remote controlled cars where the voltage supplied to the motors changes based on the position of a joy stick on the remote. This type of control is ideal for simple systems that can operate effectively without feedback. On the other hand, closed loop control measures the system output using sensors and creates an error signal based on the difference between the desired and the measured output. Advantages of closed loop control include better reference tracking, disturbance rejection and the ability to stabilize otherwise unstable systems.
Another common type of control is feed-forward control. This type of control can be implemented either in an open loop or closed loop structure. In the open-loop case, feed-forward control uses known plant dynamics to estimate the necessary control input based on the reference signal. Feed-forward control without feedback can be thought of as a more intelligent type of open loop control. Feed-forward control with feedback takes it one step further by overcoming variations between the system math model and the actual system, which leads to better control. Computed torque control on a robotic arm is a good example of this, see section 1.7 for details. An overview of automatic control along with standard theory and practices can be found in [26].

As stated in section 1.1, there are many benefits associated with model based controllers. When the system is non-linear, multiple-input-multiple-output (MIMO), or complex, then model based controllers are preferable. However, they require formulation of a model, which can be difficult. A widely used model based controller, discussed in the next section, is Model Predictive Control (MPC). The theory is explained to give an understanding of a typical model based controller used in industry.

### 1.5 Model Predictive Control Theory

Model Predictive Control, also known as receding horizon control, has become widely used in industry [30]. See Fig. 1.2 for the standard MPC block
The motivation behind MPC was the need to efficiently control multivariate processes with constraints, much like the potato processing application discussed in section 1.3. The MPC algorithm uses an explicit process model to predict future system states. It then minimizes the error between the prediction and the reference by manipulating the present and future control actions using an optimization algorithm. The present control action is sent to the system and the future control actions are sent to the model.

![Diagram showing Model Predictive Controller structure]

**Fig. 1.2: Model Predictive Controller structure**

Dynamic Matrix Control (DMC) is a common type of MPC that uses a system model in the form of a matrix, denoted as $A$ and called the dynamic matrix. The general form of $A$ is shown in Eq. 1.1 A unit step response vector, given by $a$, is used to populate the dynamic matrix. The notation $a_i$ refers to the value of the plant at the $i^{th}$ time-step. The unit step response includes every value to the $N^{th}$ time-step where $N$ denotes the prediction.
horizon, typically specified as the time-step where the plant reaches 95% of its steady state value. The control horizon, $n_u$, represents how many future time steps the controller takes into consideration while optimizing the control action. The value of this parameter varies depending on the plant.

$$A = \begin{bmatrix}
    a_1 & 0 & \ldots & 0 \\
    a_2 & a_1 & \ldots & 0 \\
    \vdots & \vdots & \ddots & \vdots \\
    a_N & a_{N-1} & \ldots & a_{N-n_u}
\end{bmatrix}$$  \hspace{1cm} (1.1)

The cost function, $J$, is given in Eq. 1.2 where $\hat{y}$ is the predicted response, $r$ is the set point and $\Delta u$ is the change in control action. The indices represent the time instant at which the value is taken, $t$ is the time instant now and anything added on to $t$ represents future time instants. A tuning parameter, $\lambda$, is used to improve the conditionality of $A$. The goal of the controller is to manipulate $\Delta u$ to eliminate the difference between a setpoint and the current system state by driving the cost function to zero. Minimizing the cost function yields the equation for a vector of current and future control actions $\Delta u$ in Eq. 1.3 where $r$ refers to a setpoint vector and $\hat{y}$ refers to the a vector of predicted outputs. Every time-step both the prediction and control action vectors must be updated using equations page 10 and Eq. 1.4.

$$J = \sum_{j=1}^{N} (\hat{y}_{t+j} - r_{t+j})^2 + \sum_{j=1}^{n_u} \lambda(\Delta u_{t+j-1})^2$$  \hspace{1cm} (1.2)
\[
\Delta u = (A^T A + \lambda I)^{-1} A^T (r - \hat{y}) \quad (1.3)
\]

\[
\hat{y}_i = \hat{y}_{i-1} + A \Delta u_{i-1} \quad (1.4)
\]

\[
\Delta u_i = \Delta u_{i-1} + \Delta u \quad (1.5)
\]

DMC is only one example of MPC, other variations are described in detail in [3], each has their own benefits and disadvantages.

### 1.6 Computer Vision Theory

Correct and sufficient feedback are key for maintaining adequate control of a process, regardless of whether the controller is model-based, like MPC, or not. Recent advances in computational power have increased the feasibility of image-based control schemes, particularly at lower frequency control bands, to the point that video analysis is relevant to controls engineers. Computer vision uses two-dimensional signal processing techniques to analyze video frames in order to derive useful information. The choice of which computer vision tool to use generally depends on the application and the desired result. An overview of image processing and computer vision will be presented as well as specifics relating to this thesis.
1.6.1 Overview and Background

Image processing and computer vision are related in that they both seek to manipulate images. However, computer vision tends to go one step further than image processing through the desire to learn information that can be used in decision making processes. The basic structure for a computer vision algorithm is given in Fig. 1.3. It shows typical steps to extracting useful information from a camera. The first step, image acquisition, is self-explanatory. The second step, preprocessing, includes any tools that are used to enhance the overall effectiveness of subsequent operations. The third step is to extract the desired information from the image. This step varies significantly from case to case depending on the application but it usually contains at least one of the following: segmentation, learning, feature detection and feature extraction. Prior to discussing these in detail, it is first necessary to understand the structure of imaging data.

The majority of digital images are raster, which means they consist of a discrete number of picture elements, or pixels, that lay in a grid, see Fig. 1.4 for an example of a binary raster image. Each of its pixels has a value associated with it. In this way raster images are equivalent to matrices and can be manipulated as discrete spatial two-dimensional (2D) signals. Spatial refers to the fact that the value is a function of the pixel location on the grid, as opposed to being a function of time. Discrete refers to the fact that these locations are not continuous, pixel locations are non-negative integers. Since images are essentially 2D discrete functions, it is possible to use conventional
Fig. 1.3: General block diagram for a computer vision algorithm

Techniques for manipulating functions on images as well. One of the more common algorithms, convolution, is used to apply spatial filters on raster images. Convolution is discussed further in the next section.

Fig. 1.4: Illustration of a binary raster image
1.6.2 Convolution

Convolution is essentially weighted filtering. One function, the weight function, is multiplied by the input function to give the filtered function. An expression for discrete convolution is given in Eq. 1.6 where \( f \) is the function being filtered, \( g \) is the weight function and \( h \) is the filtered output. The weight function has a length \( 2N \). The center value of the weight function is lined up with \( f(x) \) so that \( g \) spans from \( f(x - N) \) to \( f(x + N) \). Each value of the weight function and the original function are multiplied and then summed to give the filtered output \( h(x) \).

\[
h(x) = \sum_{j=-N}^{N} f(x - j)g(j)
\]  

(1.6)

For image processing applications, convolution is expanded to two dimensions by defining a 2D weight function, called a kernel or a mask, see Eq. 1.7. \( H(i, j) \) is the filtered pixel value which is calculated by masking the kernel matrix over the input matrix and then summing element-wise products. Convolution can be used for a wide variety of purposes such as blurring, sharpening and edge detection, all of which are examples of preprocessing techniques discussed in the next section.

\[
H(i, j) = \sum_{m=0}^{a} \sum_{n=0}^{b} K(m, n)I(i + m, j + n)
\]  

(1.7)
1.6.3 Preprocessing Techniques

The objective of preprocessing is to regulate the variance between images in a set, reduce noise in each image and remove any irrelevant information. There are a number of tools available for preprocessing that are useful in different applications of computer vision, such as histogram equalization, Gaussian blurring and thresholding.

Histogram equalization is used to regulate the variance between images in a set as well as improving the contrast of an image, making differentiating parts of that image easier. A histogram of an image is a plot of the number of occurrences of every possible pixel value. Histogram equalization works by mapping the pixels of an input image to the pixels of an output image using the cumulative distribution function of the input image. The cumulative distribution function is a plot of probability versus pixel value where the probability is the chance that a pixel value in the image is equal to or lesser than the value on the x-axis. The output pixel value is calculated using Eq. 1.8 where \( dst_{i,j} \) is the destination pixel value, \( src_{i,j} \) is the source pixel value, \( L \) is the number of possible intensity values, normally 256 and \( p_n \) is the probability that any given pixel in the image has intensity \( n \). The effects of histogram equalization is shown in Fig. 1.5.

\[
    dst_{i,j} = (L - 1) \sum_{n=0}^{src_{i,j}} p_n \tag{1.8}
\]

---

1Source of original Lena photo: USC-SIPI Image Database, Volume 3: Miscellaneous
Fig. 1.5: The effects of histogram equalization on a grayscale image
A Gaussian blur is used to smooth an image and reduce noise by convolving an image with a Gaussian kernel. A Gaussian kernel is a two dimensional matrix modeled after the 2D Gaussian function shown in Eq. 1.9, where $\sigma$ is the standard deviation. An approximate Gaussian kernel is given by $K$ in Eq. 1.10 where $\sigma \approx 0.8$. Fig. 1.6 shows this technique suppressing high frequency noise of the input image.

$$g(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}$$

$$K = \frac{1}{16} \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$$

Thresholding is used in order to remove irrelevant information. For instance,
if a robot is tracking the movement of a part on a conveyor belt, then thresholding can be used to remove everything in the image that does not correspond to the color of the part. This works by setting all of the pixel values in an image to 1 if they fall within a specified range and 0 if they fall outside of the range. The effect of thresholding on a grayscale image is shown in Fig. 1.7.

Histogram equalization, Gaussian blurs and thresholding are three methods for enhancing the overall effectiveness of subsequent operations. Since the goal of computer vision is to make informed decisions based on image data, these techniques can increase the confidence of decisions. However, computer vision algorithms also rely on recognition to make effective decisions, which is discussed in the next section.
1.6.4 Recognition

Recognition is characterized by learning information from images for use in decision making processes. This area of research is extensive but three items that are prevalent throughout different applications are segmentation, feature detection and feature extraction. This section is meant as an introduction to these techniques.

Segmentation involves dividing an image into different areas based on characteristics such as color, texture, intensity, saturation, size and shapes. The goal of segmentation is to highlight regions of interest within the image and to simplify any subsequent analysis. Three common types of image segmentation include thresholding, clustering and histogram-based methods although there are many more\cite{29}. The most basic form of segmentation is thresholding, which was described in subsection 1.6.3. Thresholding works on individual pixel values. This means that every pixel is evaluated individually. Images can also be segmented by evaluating groups of pixels, texture segmentation is performed in this way. Clustering segmentation is slightly more complex and involves a comparison between pixels and cluster centers. Pixels are assigned to the cluster that is most like them in terms of any preset factors such as color, intensity or texture based on an iterative approach. Histogram-based methods rely on segmenting the image based on the peaks and troughs in its histogram with either color or intensity values. Thresholding, clustering and histogram segmentation were used in various algorithms in this thesis, see section 2.4 for details.
In order to discuss feature detection or extraction it is necessary first to understand what a feature is. There is no formal definition for an image feature but they may be described as an aspect of an image that is interesting, significant or relevant to the task at hand. This includes predefined characteristics such as corners, edges, blobs, lines, circles, and ridges. However, unique features can be defined as well, using parts of another images as a template. The power of feature detection comes from the ability to find these key areas in an image and exploit them in decision making processes. Feature detection is preformed using a variety of statistical algorithms, each one tailored to find a certain type of feature. Corners, edges, blobs and ridges can be found using their respective detectors. Circles and lines can be found using Hough transforms. User-defined features can be detected using feature description algorithms. Feature descriptors operate by defining interest points in an image, typically places with high contrast, and then they search for a similar contrast point profile to determine if and where the feature is in the new image. Once features are detected, they can be extracted, where the image is partitioned based on the detected features. For a more in-depth look at feature detection, description and extraction see [27].

1.6.5 Specifics For This Project

Now that general computer vision theory has been discussed, it is necessary to explain specifics that relate to this project. As section 1.1 stated, the main goal of this thesis is to create new methods for analyzing the output of an
industrial potato peeler. At the moment, the peeler parameters are set by the plant operators based on their experience, with little to no quantitative data for support. The only vision system on the market gives batch average peel efficiency. Peel efficiency is the percent of the potato area that is peeled, when viewed by an overhead camera. This means that they are missing valuable information.

An example will illustrate why having more in-depth data analytics is useful for this particular problem. Imagine two different batches of peeled potatoes. They both have a batch peeling efficiency of 90%, but for the first batch the peel is distributed across all the potatoes in little specks. The second batch has the majority of its peel on one potato. In the first case, it would make sense to increase the steam time a little bit to get rid of the specks of peel from all the potatoes. In the second one, where only one potato has peel left on, it is not worth the peel loss to try to peel one potato in the batch. This is the difference between peel quality and batch average peeling efficiency. It can lead to a deeper understanding of the relationship between the peeled potatoes and the input parameters of the peeler.

The first step towards in-depth quantitative metrics for the peeling process is segmentation. Segmentation makes it possible to analyze peel quality on an individual potato basis as opposed to the batch average analysis that is in place at the moment. Generalized methods for segmentation are briefly discussed in the computer vision theory section: [subsection 1.6.4](#). Attempts have been made for discriminating individual potatoes in images before, with
mixed results. They mainly dealt with separating individual potatoes from clusters of potatoes. Methods already exist that can adequately analyze individual potatoes for size, shape and irregularities if they are separated onto a single-file conveyor[10]. This process of separating objects from a bulk conveyor onto a single file conveyor is called singulation. This process occupies a large amount of floor space in pack-houses because of the series of conveyors necessary to perform the operation[1]. As such, computer vision systems that can recognize individual objects from clusters are desirable, not only for potatoes but for any farm product that is graded by hand or using singulation.

The earliest method for discriminating potatoes in clusters is described in [22]. This method relies on a conveyor which causes the potatoes to lay in rows between conveyor rollers with their longest axis parallel to the rollers. This simplified the computer vision process while also removing the need for singulation conveyors systems. The image processing apparatus was triggered using encoders attached to the conveyor shaft. The algorithm itself worked by compressing the image into boundary information using Freeman chain code, Fig. 1.8 shows how boundary information is compressed into data. A Freeman chain algorithm starts at a boundary pixel of a blob and travels around the boundary while recording direction of travel. It records the pixel location that it started from and once the algorithm reaches this pixel location again then the boundary is complete.

A binary image compressed into boundary information allows for quicker
Fig. 1.8: Chain assignment diagram and an example of a boundary classified by Freeman chain

subsequent processing. After compressing the image, another algorithm was used to identify the individual potatoes. This algorithm worked by iterating around the boundary while recording the y-axis distance between the midpoint of an arc and the midpoint of a chord, illustrated in Fig. 1.9 where $a$ represents a specified arc length, and $y$ is the desired variable. Hence $y$ becomes a function of the distance along the boundary as the algorithm runs. Local maxima of this function can then be used to determine where the potatoes are touching.

Fig. 1.9: Determination of overlap locations using arcs and chords along boundary edges
A more recent example builds on this method by making it work with clods as well. The method in [1] uses an iterative contour bounding algorithm to determine the touching points of potatoes instead of using an arc-chord method. This method works by scanning the image for a cluster (one or more potatoes or clods together form a cluster). If a cluster is found then the algorithm iterates around the boundary of that cluster until it returns to the pixel that it first encountered the cluster. It will then measure the horizontal diameter of the cluster and delete all of the periphery pixels. This is repeated until there is a drastic change between one diameter reading and the next. At this point deleting the periphery pixels results in separating the big cluster into two smaller ones, hence the the diameter will appear much smaller. This point is recorded and used to segment the image appropriately.

Another method for segmenting potatoes in an image was developed by [6]. Their method relies on projecting the picture values vertically and rotating the image until there is a local minimum projected value that is past a certain threshold, see Fig. 1.10. This method works well for two potatoes, but results suffer as the number of clustered potatoes rises. This limitation is easily shown by an image with no ”best cut”, as would be the case with three potatoes that form a ring. The second limitation to this method is the requirement to rotate the image at increments before calculating the projection. This adds unnecessary computational time.

Once the potatoes in the image are segmented, it may be necessary to perform video tracking as each individual potato passes on the conveyor so that
the total number of potatoes can be counted. The goal of video tracking is to identify an object in a video frame and then locate the object in subsequent video frames. Tracking refers to knowing the pixel locations of each object as well as knowing which object is which. An adequate video tracking algorithm requires a robust tagging system that does not misidentify objects. Complications occur under instances of merging, splitting and occlusion.

The design of a vision tracking algorithm depends on the intended application. A tool that is often used for control, the Kalman filter, is also widely used for video tracking\cite{34}. It works by treating the objects in the video feed as dynamic systems and recursively estimating the state of the moving

Fig. 1.10: One example of potato image segmentation\cite{6}
objects and comparing this estimate to the observed state. This makes the tracker robust to occlusion, because if the observed state is empty, meaning the object is hidden from view, the Kalman filter will use the predicted state in order to track where it thinks the object would be.

Applying video tracking to potatoes on a conveyor belt requires an algorithm that can track multiple objects simultaneously in real time, even with overlapping objects. Kalman filter based tracking algorithms have proven to be proficient at this task using cost functions, probabilistic association techniques and elastic matching in order to distinguish tracked objects that merge and split [13, 23, 20]. Additionally, [23] shows a Kalman filter tracking algorithm that is more computationally efficient than particle filter tracking, which is a widely used alternative to the Kalman filter for video tracking. Furthermore, using multiple cameras for stereoscopic color video tracking or color-thermal fusion tracking have shown improvements over more basic tracking methods [35, 18]. This shows that integrating multiple imaging sensors together has advantages.

Lastly, if the segmentation process results in overlapping potatoes that are missing significant surface area in the image then curve reconstruction may be required. Curve reconstruction involves curve extrapolation through estimation based on given perimeter points. An algorithm that guarantees a correct polynomial curve from a given set of data points is given in [8]. This could be adapted to work with potatoes.
1.7 Validating the Approach on a Robotic Manipulator

A preliminary study was conducted to validate the approach of this thesis. The goal of this study was to illustrate how computer vision, automatic control and modeling theory can coalesce to create an intelligent controller that overcomes process limitations caused by a lack of reliable feedback. In this particular case, backlash in the gearboxes of a robotic manipulator limits the precision of the end-effector, since the encoders, which are attached to the motor shafts, cannot recognize the backlash.

Since the goal is to compensate for system flaws by incorporating computer vision into the control structure, it is desirable to improve system control by changing the software, not the hardware, note that installing anti-backlash gears is an appropriate solution to this problem. A collection of control algorithms that have been designed to overcome specific challenges in position control of robotic manipulators is given in [16] and a survey of different approaches to position control in the presence of backlash can be found in [28, 2]. The simplest approach for compensating for backlash is to approach the desired position from a given direction without introducing overshoot to the controlled response. Alternatively, incorporating an inverse backlash model into the control structure to negate adverse effects works as well [31]. Ideal methods would not require a priori knowledge, since backlash may be difficult to measure and time varying.
Another option is to use two separate feedback mechanisms to compensate for backlash. This has been accomplished by using a PID dual loop controller and two encoders per joint[19]. Other methods include using multiple process models and a switching rule, where the backlash model is uses within the backlash zone and the normal model is used within the normal zone[14] [15]. Visual feedback is another option, preferable for simple manipulators since it is non-invasive. The position of the end-effector with respect to the robot frame is determined by the locating the end-effector in the image and then performing a frame transformation. A disadvantage of this method is that the feedback accuracy depends on the transformation variables used, which may be difficult to measure. However, methods exist that handle unknown transformation variables[32]. This is less of a concern if the manipulator is for picking and placing parts, where the parts are also located in the image, since both the end-effector locating and the part location will use the same transformation variables.

A PD computed torque controller with an integral switching law is used in this investigation, as described in section 1.7.3. The steady state error, settling time and stability were analyzed with respect to a variable integral gain and switch depth. This was tested on a 2DOF planar serial robotic manipulator with backlash. A camera was fixed above and facing the planar robot for visual feedback.
1.7.1 Experimental Setup

The controllers were implemented on the planar RR 2DOF robotic manipulator in Fig. 1.11. The manipulator has two geared revolute joints with approximately one degree of backlash. Position is measured using 100 PPR encoders fixed to the motor shafts. Each motor has a 38.3:1 gearbox which leads to 3830 PPR of the output shaft. Signals were sent to the motors using an NI PCIe-6321 DAQ board with Lab Windows CVI software. A Microsoft VX-800 camera was fixed above and facing the working plane of the manipulator, the resolution of which was 640x480. Table 1.1 gives modeling parameters of the robot.
1.7.2 Computer Vision Feedback

The location of the end-effector was found using color thresholding, noise reduction and Hough circle transforms. Once the pixel location of the end-effector was known, the joint angles, denoted by $\theta_1$ and $\theta_2$, were calculated using a standard frame transformation and RR serial manipulator inverse kinematics, this can be found in [5].

1.7.3 Control

The computed torque control methodologies implemented in this investigation are discussed here. Both visual and encoder measurements were recorded throughout testing.

Computed-Torque Control with Encoder Feedback -

Computed-torque control estimates the required torque for a given desired acceleration, this torque is then applied to the joint. Differences between the model and the manipulator are compensated for by proportional, integral and derivative terms that are added to the desired acceleration, see Fig. 1.13. The computed torque control law is given by (1.11), adapted from [24].

$$\tau = M(q)\left[\ddot{q}_d + K_pe + K_i\varepsilon + K_d\dot{e}\right] + C(q, \dot{q}) + B(\dot{q}) \quad (1.11)$$

$$\dot{e} = e \quad (1.12)$$
Where $\tau$ is the calculated torque vector, $q$ is the position vector, $M(q)$ is the mass matrix, $C(q, \dot{q})$ is the coupling matrix and $B(\dot{q})$ is the friction matrix of the system, given by (1.13-1.21).

$$\tau = \begin{bmatrix} \tau_1 \\ \tau_2 \end{bmatrix} \quad (1.13)$$

$$q = \begin{bmatrix} \theta_1 \\ \theta_2 \end{bmatrix} \quad (1.14)$$

$$M(q) = \begin{bmatrix} m_{11} & m_{12} \\ m_{21} & m_{22} \end{bmatrix} \quad (1.15)$$

$$m_{11} = m_1 a_1^2 + m_2 (a_1^2 + a_2^2 + 2a_1 a_2 \cos \theta_2) + I_1 + I_2 \quad (1.16)$$

$$m_{12} = m_2 (a_2^2 + a_1 a_2 \cos \theta_2) + I_2 \quad (1.17)$$

$$m_{21} = K_m m_2 (a_2^2 + a_1 a_2 \cos \theta_2) + I_2 \quad (1.18)$$

$$m_{22} = m_2 a_2^2 + I_2 \quad (1.19)$$
\[
C(q, \dot{q}) = \begin{bmatrix}
-K_c m_2 a_1 a_2 \sin(\theta_2)(2\dot{\theta}_1 \dot{\theta}_2 + \dot{\theta}_2^2) \\
m_2 a_1 a_2 \sin(\theta_2) \dot{\theta}_1^2
\end{bmatrix}
\] (1.20)

\[
B(\dot{q}) = \begin{bmatrix}
 b_1 \text{sgn}(\dot{\theta}_1) |\dot{\theta}_1|^{0.648} \\
 b_2 \text{sgn}(\dot{\theta}_2) |\dot{\theta}_2|^{0.65}
\end{bmatrix}
\] (1.21)

The dynamic model of the manipulator was adapted from [5], see Table 1.1 for parameter values. Moments of inertia and torque gains were estimated using acceleration tests on individual links and mass and length properties were measured directly. This model has two correctional gains for nonlinearities, $K_c$ and $K_m$, which were used to more accurately model the feed-forward effects of link 1 and link 2. Also, note the use of a non-linear friction term. Uncertain parameters were adjusted iteratively using a computed torque controller without PID compensation to validate the choice of values, see Fig. 1.12.

**Computed-Torque Control with Dual Feedback -**

A dual-loop computed-torque controller with PID compensation was developed, incorporating both encoder and visual measurements, in order to capitalize on advantages specific to each individual type of feedback, based on [19]. The encoders provide a rapid response with a PD controller component and the steady state visual error is driven to zero by an integral controller component, the block diagram is shown in Fig. 1.14. A switching adapta-
Table 1.1: Summary table of system model parameters

<table>
<thead>
<tr>
<th>Property</th>
<th>Symbol</th>
<th>Units</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mass of link 1</td>
<td>$m_1$</td>
<td>kg</td>
<td>2.616</td>
</tr>
<tr>
<td>Mass of link 2</td>
<td>$m_2$</td>
<td>kg</td>
<td>0.763</td>
</tr>
<tr>
<td>Length of link 1</td>
<td>$a_1$</td>
<td>m</td>
<td>0.508</td>
</tr>
<tr>
<td>Length of link 2</td>
<td>$a_2$</td>
<td>m</td>
<td>0.4064</td>
</tr>
<tr>
<td>Distance from joint to centroid, link 1</td>
<td>$a_{c1}$</td>
<td>m</td>
<td>0.2540</td>
</tr>
<tr>
<td>Distance from joint to centroid, link 2</td>
<td>$a_{c2}$</td>
<td>m</td>
<td>0.2032</td>
</tr>
<tr>
<td>Mass moment of inertia of link 1</td>
<td>$I_1$</td>
<td>$kgm^2$</td>
<td>0.220</td>
</tr>
<tr>
<td>Mass moment of inertia of link 2</td>
<td>$I_2$</td>
<td>$kgm^2$</td>
<td>0.155</td>
</tr>
<tr>
<td>Friction term of link 1</td>
<td>$b_1$</td>
<td>Nms*</td>
<td>0.500</td>
</tr>
<tr>
<td>Friction term of link 2</td>
<td>$b_2$</td>
<td>Nms*</td>
<td>0.235</td>
</tr>
<tr>
<td>Coupling compensation gain</td>
<td>$K_c$</td>
<td></td>
<td>0.17</td>
</tr>
<tr>
<td>Mass gain compensation</td>
<td>$K_m$</td>
<td></td>
<td>0.24</td>
</tr>
<tr>
<td>Voltage-torque ratio for link 1</td>
<td>$K_{v1}$</td>
<td>V/Nm</td>
<td>3.0</td>
</tr>
<tr>
<td>Voltage-torque ratio for link 2</td>
<td>$K_{v2}$</td>
<td>V/Nm</td>
<td>3.6</td>
</tr>
</tbody>
</table>

tion was implemented on this controller in an aim to improve the transient response, which suffers in a dual-loop controller as a result of competing dynamics between the different feedback errors attempting to correct the

![Model Comparison (CTC without PID)](image)

Fig. 1.12: Plots of $\theta_1$ and $\theta_2$ under CTC control without PID for model comparison
trajectory at the same time. The switching controller avoids this problem by turning on the integral gain when steady state precision is desired. The switch turns on a certain number of degrees from the final resting set point, this number is denoted by $sw$ and is referred to as the switch width. The PD gains used in this experiment for link 1 and link 2 had values of 1.2, 0.005, 2.4 and 0.005 respectively. The integral gain was variable.

1.7.4 Results

Visual Position Feedback Results

Fig. 1.15 shows the detected circle superimposed on the unprocessed image. The circle was located with an accuracy of ±1 pixel, approximately ±2.6mm.
The end-effector position was updated at 30Hz, corresponding to the camera frame rate.

**Inverse Displacement Solution** -

To illustrate the results of the inverse displacement solution, two tests were performed. In the first test the manipulator was passively moved through typical joint movements while recording both visual and encoder joint angles. These tests are given in Fig. 1.16. The difference between the encoder measurement and the visual measurement are a result of backlash, calibration constants, feedback lag and transform parameter errors. The effect of transform parameter error is also shown in Fig. 1.17 and Fig. 1.18, where one link is moved while the other is fixed. It shows that when one link is moved, the measurement of the other link is affected, even though the physical link was fixed. Note, however, that if a part is located in the image plane and
the end-effector can be controlled accurately within the image plane, then conversion between the visual frame and the Cartesian frame does not have to be as precise as it would if the part was located in Cartesian frame.

Position Control Results

The angular set point profile followed a fifth order polynomial path trajectory. In the figures the cyan dot marks where a set point starts to change and

Fig. 1.16: Visual and encoder joint angles for typical movements

Fig. 1.17: Plots showing $\theta_2$ when $\theta_1$ is kept constant
finishes changing and the black dot marks when the integral switch is turned on. The complete set point change had a duration of 1.4 seconds and 2.5 seconds for the fast and slow tests respectively. Fig. 1.19 illustrates the backlash present in the system.

The integral gain and switch width were changed throughout testing and each test composed of three separate trials. A switch width of 48 degrees means the integral term is always turned on since the set point change was 45 degrees, resulting in a standard computed torque controller with PID compensation. The $k_i$ value for both links was kept the same to limit the different number of treatments required, however results indicate that the integral gain for link 2 can be greater than that of link 1. This is supported by Fig. 1.20 in which link 1 is marginally stable but link 2 is stable, a consequence of the dynamic behavior of link 2 compared to link 1.

The results showed a reduced steady state error while maintaining a smooth transient response. The dynamic response had a seamless transition when

![Fig. 1.18: Plots showing $\theta_1$ when $\theta_2$ is kept constant](image)

36
turning on the integral term with a certain switch width, shown in Fig. 1.21. A switch width of 10 degrees also resulted in an ideal test for the slow set point, see Fig. 1.22. This indicates that the switch should be turned on while the is link is in its transient stage.

The dynamic response for a standard PD computed torque controller for a fast set point change is shown in Fig. 1.23. This response has a non-zero steady state error for the visual measurements but good trajectory following of the encoder position. The steady state error decreases at an unacceptably slow rate if the value for $k_i$ is too low, see Fig. 1.24.

**Steady State Error**

Visual steady state error for each trial was measured, averaged and plotted with respect to the switch width and integral gain, see Fig. 1.25 for the slow set point change and Fig. 1.26 for the fast set point change. The steady

![Graph](image)

Fig. 1.19: Demonstration of backlash at the joints
state error of unstable or marginally stable trials was set to three degrees to illustrate them graphically, shown by the high flat spots in the two figures. There is a clear valley in the steady state error maps where $k_i$ is equal to 0.015 for joint 1 and 0.02 for joint 2, meaning that steady state error is lowest when the integral gains are equal to these values.

Fig. 1.20: Plot showing link 1 is marginally stable while link 2 remains stable

Fig. 1.21: Plot showing an ideal response for link 1 and link 2 with a fast set point change
Settling Time -

Plots of average settling times for the experiment are given in Fig. 1.27 and Fig. 1.28. These plots support that an ideal switch width is between five degrees and ten degrees with a $k_i$ of approximately 0.015, shown by the lower settling times in this region.

Fig. 1.22: Plot showing an ideal response for link 1 and link 2 with a slow set point change

Fig. 1.23: Plot showing PD computed torque control for a fast set point change
1.7.5 Conclusion

Controlling the position of a serial robotic manipulator in the presence of measurement errors, such as backlash, presents difficulties. Introducing redundancy into the system by taking feedback measurements before and after the backlash can be used to overcome position inaccuracies. In this investigation, a low cost USB camera was used to locate the end-effector.

A dual-loop switching computed torque controller was implemented on an RR serial manipulator to mitigate the adverse effects of backlash. An analysis of steady state error and settling times was performed for different switch widths and integral gains. The steady state error maps, Fig. 1.25 and Fig. 1.26, indicate that certain values of $k_i$ result in smaller steady state errors. The settling time maps, Fig. 1.27 and Fig. 1.28, show that a switch width between five degrees and ten degrees is desirable. In general, this approach improved the dynamic response and could be extended to other controllers and implemented on other types of plants.

The next chapter discusses the main thesis topic, foundations for intelligent control of an industrial potato peeler, in detail.
Fast Setpoint Change: \( k_i = 0.010, \ sw = 1, \ trial = 3 \)

**Fig. 1.24:** Plot showing integral buildup with a low integral gain

Steady State Error Map, slow setpoint, joint 1

**Fig. 1.25:** Steady state error maps for a slow set point change

Steady State Error Map, fast setpoint, joint 1

**Fig. 1.26:** Steady state error maps for a fast set point change
Fig. 1.27: Settling time maps for a slow setpoint change

Fig. 1.28: Settling time maps for a fast set point change
Chapter 2

Application Study: Industrial Potato Peeling Process

2.1 Introduction and Background

The objective of process optimization is to increase the efficiency of a process. Ideally the process would have a higher yield per unit input, whether the input is raw material, capital expenses or labor. Recent advancements in computational power have led to the development of intelligent software approaches that manages processes with the goal of achieving optimal control. The food processing industry capitalizes on this technology by reducing waste and labor through computer vision quality inspection\[7\]. However, they have yet to take full advantage of the union of computer vision, data modeling and automatic control theory. A general solution to optimizing
food processing control can be seen in Fig. 2.1. The idea is that computer vision can be used to provide quantitative metrics that are controllable using model-based controllers.

The first step to this approach is to run a preliminary investigation in order to determine how significant different metrics are to the task as well as the model. This involves gathering a starter set of data that encompasses any and all metrics of interest. This set of data can then be analyzed in order to determine the significance of each metric to the process and the desired result. These two steps are repeated until a satisfactory set of metrics is acquired. Fig. 2.1 shows only one key metric for simplicity.

Afterwards, a method is designed to measure and quantify any key metrics that are not directly measurable. For instance, a process might need a quality measurement for an orange, for which there are no “orange quality” sensors. Hence, a computer vision algorithm, or a similar solution, would have to be developed in order to quantify the quality of an orange for the particular process in question. This is illustrated by the Computer Vision Feedback
block in Fig. 2.1. Ideally this solution would be robust and adaptive.

Now that the key metrics of the process are measurable, it is possible to gather large amounts of data and to start creating a model of the process. This data will include any relevant input/output parameters of the system as well as the key metrics calculated by the computer vision program. Gathering the data will be non-invasive since the process model will need to be developed before the process can be controlled. Thus, it is possible to record the data, create the model and test the controller in simulation. The model will also adapt to any time-varying properties of the process, since it learns from the data. The last step is to verify the controller works on the actual process and make any necessary adjustments to the algorithms so that closed loop control remains robust.

The rest of this chapter discusses the application study of this method on an industrial potato peeling process.

2.2 Description of the Process

The potato peeling process has a simple goal that can be difficult to achieve. Ideally, the skin and only the skin would be removed from the potato. However, geometric factors such as eyes, depressions and defects influence how easy the potato is to peel. Since each potato will have a different geometry, it is ideal to peel each potato individually. However, this is not practical on an industrial scale. The only time this happens is when potatoes are peeled
by hand.

On an industrial scale, there are three typical ways that potatoes are peeled. Abrasive peelers peel potatoes by rubbing the skin off using something similar to sandpaper. Caustic peelers peel potatoes by submerging them in a caustic solution to loosen the skin and then washing it off with water. Steam peelers peel potatoes by exposing them to high pressure steam and then dropping the pressure suddenly. When the pressure is released, the moisture trapped beneath the potato skin flashes, pushing the peel off of the potato. The tests for this thesis were performed on a steam peeler.

A schematic of the overall peeling process is shown in Fig. 2.2. First, the potatoes are loaded into the hopper. Then the conveyor moves the potatoes from the hopper and drops them into the peeler. Once the potatoes are peeled, they are released underneath the peeler into an auger that moves the potatoes into the deskinner. The deskinner is made up of several rotating cylindrical brushes that brush any loose peel off of the potatoes while the potatoes are washed with water.

2.3 Experimental Setup, Data Collection and Methods

Three separate rounds of tests were completed in which batches of potatoes were peeled with varying steamtime, steam pressure, and potato size. Steamtime is the length of time that the potatoes are exposed to steam before
Fig. 2.2: Schematic overview of the potato peeling process

pressure is released. Steam pressure is the pressure of that steam. Potato size is the small diameter of the potato, sorted by passing potatoes through a template. Recorded measurements include the batch peel loss as well as optical and thermal data. Peel loss is the percentage of mass lost during peeling. Peel loss was calculated by weighing a sample of potatoes before peeling and after peeling, then taking the difference and dividing it by the weight before peeling. Peel loss was measured in air and in water. Weight in air was measured using a standard scale. Weight in water was measured using a standard scale with a hook attached to a submerged basket of potatoes. The optical and thermal data were recorded using the peel vision scanner (PVS). The PVS is a light box containing three cameras overlooking a blue conveyor belt. One camera is an optical camera, one is a thermal camera and the third is a proprietary optical camera. In this investigation only data from the optical camera and the thermal camera were used because of technical
difficulties with the proprietary camera. The potatoes were placed on the conveyor belt in order to have their optical and thermal signatures recorded. The three separate rounds of tests, design of experiments one (DOE1), DOE2 and DOE3, had three different seasonalities new crop, mid crop and late crop respectively. Seasonality refers to how long the potatoes have been in storage. New crop means that they were just harvested, whereas late crop means they have been in storage for several months. All of the potatoes peeled during this investigation were russet Burbank potatoes. The overall number of trials was constrained by equipment malfunctions and the availability of the test facility.

DOE1 and DOE3 consisted of 16 treatments and 3 trials per treatment. For DOE2, technical issues with the PVS limited the video recordings to 18 trials, leading to 114 trials for DOE1, DOE2 and DOE3 combined. Potato size varied from 2.0 in, 2.5 in and 3.0 in, steamtime varied from two seconds to fifteen seconds, and pressure varied from 14 bar to 20 bar, depending on the trial.

At the beginning of a trial, a 27 kg batch of unpeeled potatoes was scanned using the PVS. Afterwards, a 5 kg sample was taken from the batch. A shallow “x” was cut onto each potato in the sample using a knife dipped in red dye. This allowed the potatoes to be identified as belonging to the sample, even after being peeled. The number of potatoes in the sample was recorded and the potatoes were weighed in air and in water. Once these measurements were complete, the potatoes were loaded into the hopper and
dropped into the peeler.
After leaving the peeler, the potatoes went through the deskinner with the gate open and room temperature water running over them. They were then sprayed with a hose, also with room temperature water, in order to rinse any detached peel from the flesh. The potatoes were then scanned through the PVS again, and the potatoes belonging to the 5 kg sample were found. The sample was then weighed after peeling, in air and in water. Analysis of this data is given in chapter 3.

2.4 Developing the Segmentation Algorithm

The development of a segmentation algorithm for discriminating individual potatoes on a conveyor belt was the first step towards gathering quantitative data on individual potatoes. Originally, it was thought that the thermal data would be used for segmentation, since the contrast between peel and white flesh was not present, meaning standard edge detection algorithms would be able to separate the individual potatoes. However, the thermal data lacked distinct borders between potatoes. This was likely a result of the experimental process where the potatoes went through the deskinner before the PVS. The raw image and corresponding edges are given in Fig. 2.3 to illustrate this point. Hence, the optical data was used for segmentation.

The optical segmentation algorithm follows the schematic given in Fig. 2.4. The frame is captured from a video file and the sure background is classified
in an adaptive way. Afterwards, seam detection finds pixels that correspond to touching points between separate potatoes. These pixels are added to the background and then any remaining holes in whole potatoes are filled. Contours are then found in the image and their perimeters are checked if they are single potatoes or clumps of potatoes. Any contour that passes this check is added to the final (single potato) contour list. Any contour that fails this check is added to a list of contours that undergo subsequent operations, contour splitting and concavity segmentation, until only single potatoes remain. The remaining single potato contours are then added to the final contour list. In this way, a bounding area for each potato is found, with coordinates corresponding to each potato contour. An example of the segmentation algorithm applied to an optical frame is given in Fig. 2.5. The next few sections discuss the segmentation steps in more detail.

![Fig. 2.3: Thermal image and corresponding canny edge detection](image)

2.4.1 Background Subtraction

Background subtraction is needed to reliably remove anything that is not part of a potato from the image. The background subtraction in this segmentation algorithm is adaptive and works by creating an HSV histogram of the input image. An example input image and the processed image are shown in [Fig. 2.6](#). The HSV histogram is bimodal when potatoes are present in the image. The two main peaks in the histogram are identified and the corresponding hue values are used for thresholding. [Fig. 2.7](#) shows the bimodal histogram and corresponding peaks, denoted by the black markers.
Fig. 2.5: Example of segmentation algorithm on medium sized potatoes

Fig. 2.6: Optical image and processed background subtraction image
2.4.2 Seam Detection

Seam detection identifies the pixels that belong to the seams between potatoes, so that these pixels can be removed. One of the major problems with finding the seams, and the main reason standard edge detection algorithms cannot be used, is that the peel and white flesh boundaries create hard edges. Meaning, if standard edge detection algorithms are used then a single potato may be misidentified as multiple potatoes, see Fig. 2.8 for an example of a canny edge detector applied to a frame. Instead, blackhat morphology and subsequent thresholding was used to identify the seams. An example of the morphed image and the thresholded image is given in Fig. 2.9.
Morphological filtering operates based on shape principles. Blackhat morphology is defined as the difference between the closing of an input image and the input image itself. The closing of an input image is determined by a dilation operation followed by an erosion operation. In the most basic terms, dilation makes shapes in the image larger, and erosion makes shapes in the image smaller, any other morphological operations are simply sequences of dilation and erosion.

Fig. 2.8: Canny edge detector applied on a frame after background subtraction

2.4.3 Hole Filling

Hole filling is performed in order to clean up the new background. Shown in Fig. 2.10 the process uses connected components analysis to find the holes in the potatoes and fills them. This eliminates noise created by seam detection.
2.4.4 Find Contours and Perimeter Checking

Once the background is subtracted, the seams are detected and the holes are filled, then the contours can be found and sorted by single potatoes and clumps. Fig. 2.11 shows a frame of contours and the corresponding sorted contours, where green denotes single potatoes and red denotes clumps. A histogram of contour perimeter sizes can be plotted to determine the perimeter range for single potatoes and clumps of potatoes. Fig. 2.12 shows that most
individual potatoes have a perimeter size between 200 and 400 pixels in this case. This varies based on the size distribution of potatoes in a batch. Any contours that do not pass this check are then evaluated further using contour splitting.

![Fig. 2.11: Contours found before and after perimeter checking](image)

### 2.4.5 Contour Splitting

Contour splitting draws a black line around each clump and then checks to see if it splits, as shown in [Fig. 2.13](image) where two of the clumps split, but one does not. If the blobs are attached by a thin section, then this usually suffices. If not, then concavity segmentation is performed.

### 2.4.6 Concavity Segmentation

Concavity segmentation identifies points where it looks like potatoes are touching, based on the distance between the convex hull and the contour
Fig. 2.12: Example histogram of contour perimeter sizes for medium potatoes

Fig. 2.13: Contours before and split image mask afterwards

line. Black lines are then drawn between the points in order to segment the clump, see Fig. 2.14 for an example.
2.4.7 Segmentation of poorly peeled potatoes

The seam detection step in the segmentation algorithm relies on blackhat morphology in order to identify where separate potatoes are touching. Because of this, the algorithm fails to detect seams when potatoes are poorly peeled. This is shown in Fig. 2.15, where the image on the left is the input frame and the image on the right is the result of a blackhat morphological operation. When thresholded, it gives page 59, where the seams should be identified in red. Instead, the algorithm identifies false positives for seam pixel locations. In practice, this is not critical since potatoes are relatively well peeled on the actual production line. However, for the data gathered at the testing facility, where many of the trials were very poorly peeled, a minimum distance classification algorithm was used to determine batch peel efficiencies. Minimum distance classification assigns each pixel the label of its closest cluster center.
Fig. 2.15: Image results showing the thresholded frame and blackhat morphed frame

Fig. 2.16: Image showing thresholded blackhat morphology of poorly peeled potatoes
2.5 Methods For Creating a Database

Since segmentation was limited to well-peeled potatoes, discussed in subsection 2.4.7, it was necessary to create methods for analyzing test data without the segmentation algorithm. The first set of analytics involved classification methods that calculated batch average peel efficiency. The second set of analytics involved segmenting potatoes by hand and then analyzing how the peel quality of individual potatoes changed with respect to the input parameters. The first is discussed in subsection 2.5.1 and the second is discussed in subsection 2.5.2.

2.5.1 Classification Methods

Two different classification algorithms were investigated: HSV thresholding and minimum distance classification. It was thought that the potato area would have a bimodal histogram, since it would be either peeled or unpeeled, and so the same method that was used to subtract the background could be used to classify peel and whiteflesh. However, the histogram was only bimodal for an 8 second steam time, see Fig. 2.17, Fig. 2.18 and Fig. 2.19. HSV thresholding was performed by setting a range for hue, saturation and value limits that correspond to each class. These limits were found by hand, using a thresholding program with a sliding bar. Any pixel that fell within a class range was labeled as belonging to that class. The ranges for HSV classification for DOE1 are given in Table 2.1. An example of an unpro-
cessed image, overlaid by a classified image is given in Fig. 2.20. Background subtraction was performed prior to classification to avoid wrongly classifying the background, see Fig. 2.22.
Table 2.1: Pixel ranges for HSV classification

<table>
<thead>
<tr>
<th>Category</th>
<th>H</th>
<th>S</th>
<th>V</th>
</tr>
</thead>
<tbody>
<tr>
<td>White flesh</td>
<td>35-46</td>
<td>170-255</td>
<td>65-255</td>
</tr>
<tr>
<td>Green flesh</td>
<td>20-49</td>
<td>100-255</td>
<td>80-255</td>
</tr>
<tr>
<td>Peel</td>
<td>0-31</td>
<td>90-255</td>
<td>0-120</td>
</tr>
</tbody>
</table>

Fig. 2.19: Histogram of small potatoes, steamtime = 15 s

Fig. 2.20: Overlaid HSV classification example with green flesh
HSV thresholding classifies pixels accurately, but it is not a robust method. Small changes in lighting conditions and camera settings can negatively affect performance. Because of this, a combination of HSV thresholding and minimum distance classification was ultimately used to calculate batch average peel efficiencies, see Fig. 2.22 for a flowchart showing the calculation and Fig. 2.21 for example classification results. HSV thresholding was used to classify green flesh in the image and minimum distance algorithms were used to classify white flesh and peel. The class pixel values were determined using a color picker multiple times and taking an average of the returned values. This process was applied to all 114 optical video files to calculate batch average peel efficiencies.

![Fig. 2.21: Overlaid minimum distance classification example with green flesh](image)

![Fig. 2.22: Process for finding peel efficiency](image)
Rot Identification

Another feature of these methods is the ability to differentiate between peel and rot. A thermal image and an optical image can be aligned in a way that checks can be made on peel patches to make sure that they are, in fact, peel and not rot. This is possible because rot retains moisture and moisture retains heat, giving rot and peel different thermal signatures, as shown in Fig. 2.23 through Fig. 2.26. Once the rot areas are identified, they can be removed from the peel data to more accurately reflect current peeler performance metrics. Note that this is a new method that is not practiced in industry.

2.5.2 Individual Potato Analytics

Since the segmentation algorithm had difficulty distinguishing borders when the potatoes were poorly peeled, it became necessary to segment potatoes by
hand in order to analyze the peel quality of individual potatoes in the test data set. This approach used human interaction in order to draw contours around 12 potatoes in each trial of all three DOEs, an example of which is shown in Fig. 2.27. The potatoes that were segmented were chosen randomly. Once the contours for every trial were saved, a program was written to loop through the contours, in both the optical images as well as the thermal images, and save the data to a file. This data contained the trial parameters and measurements: steamtime, pressure, potato size, peel loss in air, peel loss in water and average peel quality. The data also contained measurements for 12 individual potatoes from each trial, which were: peel quality, thermal matrix, rgb histogram, Hu moments, contour area, major diameter, minor diameter, and a list of contour areas of peel pieces left on each potato. This database was investigated and significant results are discussed in the next section.

Fig. 2.27: Example of potato images segmented by hand
Chapter 3

Results and Discussion

Discoveries from the database discussed in subsection 2.5.2 are described in detail in this section. This includes batch average data as well as individual potato data. Fig. 3.1 presents a plot of peeling efficiency across a range of steam pressures, from 14 bar to 20 bar, using a variety of potato sizes. Based upon these results, there does not appear to be a strong correlation between steam pressure and peeling efficiency. This is reinforced by Fig. 3.11, a histogram of peel distribution, where there the trend is the same regardless of the pressure. On the other hand, a histogram of peel distributions, with steamtime indicators, shows that peel distribution varies with steamtime, see Fig. 3.12. Even so, there are likely interactions between pressure and other variables, which cannot be evaluated based upon the limited data in these experiments. For this reason, pressure will be included in future research. Continuing, Fig. 3.5 presents a plot of peeling efficiencies, across a range of
steam times, versus peel loss measured in air. In this figure, potato size is represented by the size of the individual circles. Based upon these results, there is a strong, positive correlation between peeling efficiency and steam time. Eventually, further increased steam time results in increased peel loss with negligible improvement in peeling efficiency, implying that there will be an optimum steam time, for any given set of other potato parameters. Furthermore, smaller potatoes are more difficult to peel and will require longer steam times to achieve comparable peeling efficiency results, shown by lower peeling efficiency at the same peel loss.

Fig. 3.2 through to Fig. 3.4 reinforces that smaller potatoes are more difficult to peel, figures are plots of individual potato peeling efficiency vs steamtime, with pressure indicators, for small, medium and large potatoes respectively. Small potatoes have a noticeably larger amount of variation between individual potato peeling efficiencies than medium or large potatoes.

Furthermore, not only are small potatoes more difficult to peel, but, while in storage, they become more difficult to peel at a faster rate than medium or large potatoes. Fig. 3.6 illustrates this point with a plot of peeling efficiency versus peel loss measured in air, across a range of potato sizes and crop ages. This is logical since the ratio of surface area to volume of a sphere is inversely proportional to its radius, so even under an optimal peeling strategy, losses will be higher for smaller potatoes.

Lastly, the above results were replicated with plots versus peel loss measured in water. However, it is important to note that peel loss measured in water
has higher variability, shown in Fig. 3.8. This is most likely a cause of measurement error. While measuring in air the mass of a sample is around 5 kg. Buoyancy effects in water cause this to drop to about 0.4 kg. A scale with the same precision was used for both measurements. Hence peel loss in water, because it is a percent measurement, will have a larger error than peel loss measured in air. That being said, the results discussed were supported by both measurements.

Fig. 3.1: Peeling efficiency vs actual pressure with potato size indicators
Fig. 3.2: 2.0 inch tubers
Fig. 3.3: 2.5 inch tubers
Fig. 3.4: 3.0 inch tubers
Fig. 3.5: Peeling efficiency vs peel in air with steamtime indicators

Fig. 3.6: Peeling efficiency vs peel loss in air with seasonality indicators
Fig. 3.7: Peeling efficiency vs peel loss in air with pressure indicators

Fig. 3.8: Peeling efficiency vs peel in water with steamtime indicators
Fig. 3.9: Peeling efficiency vs peel loss in water with seasonality indicators

Fig. 3.10: Peeling efficiency vs peel loss in water with pressure indicators
Fig. 3.11: Histogram of peel distributions for all data sorted by pressure

Fig. 3.12: Histogram of peel distributions for all data sorted by steamtime
3.1 Modeling the Process

This section summarizes preliminary work on modeling the process from the relatively small data set and presents an illustrative framework for acquiring more online data.

The required size of a dataset for modeling a MIMO system increases exponentially with respect to each variable belonging to the model. As such, a complete model of the peeling process cannot be formulated from the limited experiments on a potato processing test line. Although the dataset is limited, polynomial regressions can be performed with one or two variables on subsets of data in order to analyze trends. Fig. 3.13 shows a surface fit illustrating the effects of steamtime and pressure on peeling efficiency. The surface has the polynomial equation:

\[ \eta = -183.1 + 18.7P - 0.51P^2 + 13.5S - 0.47S^2 \] (3.1)

where P is pressure in bar, S is steamtime in seconds and \( \eta \) is peeling efficiency. This is a simple example of a static model trained on a subset of data.

For creating a learning and adaptive predictive model for \( \eta \), the approach would be to obtain data from a high throughput production line, so that a close to complete model can be created. This future strategy is illustrated in Fig. 3.14 which shows an intelligent potato processing where high volume data can be acquired for deep learning and model formulation. At first the
Fig. 3.13: 2D polynomial regression surface of peeling efficiency as a function of steamtime and pressure

plant operators can control the production line themselves while the system passively records data. Once an accurate and robust model is formulated, using online learning, then automatic control can be performed on the various parameters.

Automatic control will work in the following way: As a batch of potatoes enters the peeler a laser scanner calculates the size distribution of the batch which is then relayed to a model based controller. The controller sets the peeling parameters based on this information, as well as seasonality and potato variety. The output of the peeler is then scanned with the PVS and analyzed to determine peeling efficiency, peel quality and peel distribution. This information is then used to update the model and control parameters.
Offline peel loss measurements will be performed and cataloged by plant operators, so that peel loss is included in the model as well.

Future models can leverage neural networks in order to capture complex interactions between variables. Neural networks can approximate any function and have the advantage of not requiring a priori knowledge to make a model. Ideally, a neural network could incorporate every parameter in the form $\eta = f(\mathbf{x})$ where $f()$ represents a neural network and $\mathbf{x}$ is a parameter vector containing items such as: size distribution, variety, seasonality, steam time, pressure, growing conditions, pretreatment, peeler model etc.
Chapter 4

Conclusions and Recommendations

In summary, potatoes were peeled in 27 kg batches at a potato testing facility and peel loss (measured by percent weight loss both in water and in air) was measured on 5 kg samples. Videos captured by thermal and optical cameras were recorded as the peeled potatoes passed beneath the cameras on a conveyor. The optical camera data was used to evaluate a variety of peeling metrics, including peeling efficiency, the percent of potato area with no peel, and the size/number of peel spots on individual potatoes. An investigation was performed to determine the effects of various peeling parameters on the distribution of peel left on the potatoes after peeling. These variables included the crop age, potato size, steamtime and steam pressure. Lastly, the thermal camera data was used to validate that thermal images can distinguish
between potato peel and rot, which is not possible with the optical camera. Several significant discoveries were made throughout this investigation. One discovery was that smaller potatoes are more difficult to peel and they become even more difficult to peel at a faster rate than medium or large potatoes. Consequently, it is recommended that the size distribution of a batch be recorded prior to peeling, in any future work. This way, steamtime can be adjusted for optimal peeling based on the size distribution of any batch. Furthermore, the age of the crop should be taken into consideration, along with the batch size distribution, when setting a steam time. Lastly, peel loss and peeling efficiency showed no correlation with changes in steam pressure. However, interactions between pressure and other parameters may not be negligible. Hence, it is suggested that pressure be recorded during any future work. Moreover, a larger dataset is required to validate findings from this investigation and to create a model of the process. Thus, it is recommended that further tests be conducted where data is passively recorded in a main processing plant to generate a larger dataset.

In conclusion, the ultimate outcome of future research would be the creation of an online, dynamic learning model that could be used to provide real-time, closed-loop control of the peeling system. It would predict the optimal operating parameters, using the input variables (as noted previously) and execute closed-loop control. Optimization of the peeling process could offer significant savings in costs for peel waste disposal and maximize the net production throughput under a wide variety of operating conditions. This
thesis served as the first steps towards realizing this goal. Fig. 4.1 summarizes the methodology for an intelligent factory approach to controlling industrial potato peelers.

Fig. 4.1: Overall intelligent factory approach for industrial potato peelers
Bibliography


[13] Jong-Min Jeong, Tae-Sung Yoon, and Jin-Bae Park, Kalman filter based multiple objects detection-tracking algorithm robust to occlusion, SICE


Curriculum Vitae

Candidate’s full name:

- Zachary Chase Knopp

University attended:

- Bachelor of Science in Mechanical Engineering, UNB, 2014

Conference Presentations:

