

Analysis of the Industrial Automation of a Food Processing Quality Assurance Workcell

by

Boyd Allin

B.Sc.E., Queen's University, 1995

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF

THE REQUIREMENTS FOR THE DEGREE OF

MASTER OF APPLIED SCIENCE

in

THE FACULTY OF GRADUATE STUDIES

(Department of Mechanical Engineering)

We accept this thesis as conforming

to the ~~required~~ standard

THE UNIVERSITY OF BRITISH COLUMBIA

April 1998

©Boyd Drew Allin, 1998

In presenting this thesis in partial fulfilment of the requirements for an advanced degree at the University of British Columbia, I agree that the Library shall make it freely available for reference and study. I further agree that permission for extensive copying of this thesis for scholarly purposes may be granted by the head of my department or by his or her representatives. It is understood that copying or publication of this thesis for financial gain shall not be allowed without my written permission.

Department of Mechanical Engineering
The University of British Columbia
Vancouver, Canada

Date April 30th / 98

Abstract

This thesis details a research project into the viability of increasing the level of automation employed in the canning of salmon. Specifically the post-filling quality control patching table has been studied. Data on this system was compiled from survey, interviews, and video tape of quality control personnel, and from a series of experiments done in an industry plant during a canning operation. The patching table workcell has been modelled and simulated using a computer simulation package. Several automation implementations are explored for their effectiveness and physical realizability. The most promising of these options have been simulated and evaluated for their economic profitability. Recommendations to the industrial producers are then presented.

Contents

Abstract	ii
List of Figures	viii
List of Tables	xi
Acknowledgements	xiii
Glossary	xiv
1 Introduction	1
1.1 Motivation	2
1.2 Canning Process Details	3
1.3 The Patching Table	4
1.4 Research Objectives	6
2 Literature Review	9
2.1 Modelling Manufacturing Systems	9
2.2 Food Processing Simulations	12
2.3 Human Factors	13

2.3.1	Inspection Accuracy	13
2.3.2	Factors Affecting Inspection Accuracy	14
2.4	Semi-Automated Manufacturing Systems	20
2.5	Economics	22
3	Data Acquisition and Modelling	25
3.1	Quality Control Questionnaire and Interviews	25
3.1.1	Safety Defects:	28
3.1.2	Regulatory Defects:	29
3.1.3	Appearance Defects:	29
3.2	On-site Experimentation	30
3.2.1	Layout	31
3.2.2	System Setup and Product Input	32
3.3	Experimental Results and Data Modelling	32
3.3.1	Worker Tasks and Table Operation	32
3.3.2	Flange Defects	34
3.3.3	Cross-packs	37
3.3.4	Weight	39
3.3.5	Poor-fills	42
3.3.6	Meat Preparation	44
3.3.7	General Observations	45
4	Basecase Simulation	46
4.1	Basecase	46
4.1.1	Logic and Layout	47

4.1.2	Tuning and Evaluation	52
4.1.3	Results	54
5	Comparative Methodology	56
5.1	Operational Parameters	56
5.2	Monetary Savings	59
5.2.1	Direct Labour Savings (DLS)	59
5.2.2	Indirect Labour Savings (ILS)	59
5.2.3	Raw Material Savings	60
5.3	Economic Parameters	62
5.4	Non-Monetary Effects	63
6	Automation Technologies and Prototypes	64
6.1	Automated Can Inspection	65
6.2	Automated Weight Correction	66
6.2.1	Slurry Injection System	66
6.2.2	Chunk Injection Systems	67
6.3	Automated Flange Cleaning	70
6.3.1	Prototype Trimmer	70
6.3.2	Existing Trimmer Design	72
7	Automation Option Simulation and Evaluation	74
7.1	Automated Inspection (V)	74
7.1.1	System Assumptions	74
7.1.2	Implementations	75

7.1.3	Physical Layout	77
7.1.4	Results	78
7.1.5	Sensitivity	79
7.2	Automated Weight Correction (W)	81
7.2.1	System Assumptions	81
7.2.2	Implementations	81
7.2.3	Physical Layout	83
7.2.4	Results	83
7.2.5	Sensitivity	83
7.3	Hybrid System	84
7.3.1	System Assumptions	84
7.3.2	Physical Layout	84
7.3.3	Results	86
7.3.4	Sensitivity	87
7.4	Summary	88
8	Conclusions and Recommendations	89
8.1	Conclusions	89
8.2	Recommendations	90
A	Distribution Acceptance	93
	Appendices	96
B	Quality Control Survey	96
C	Awesim Networks	117

D Awesim User Written C Routines	121
E Screening Program Details	149

List of Figures

1.1	The salmon canning process.	4
1.2	Schematic of the patching station.	4
1.3	Patching table during operation.	5
2.1	Generalized relationship between inspection time and accuracy of performance. . . .	19
2.2	Effect of defect rate on inspection accuracy.	20
2.3	Cost system supporting analysis of advanced manufacturing systems.	23
3.1	Various filled salmon cans.	27
3.2	Experimental setup.	31
3.3	Patching station.	32
3.4	Worker interaction and responsibilities.	33
3.5	Off-line flange patch times.	36
3.6	Cross-pack patch times.	39
3.7	Type (b) cross-pack patch time distribution fit.	39
3.8	Checkweigher totals.	40
3.9	Post-patching weight control.	41
3.10	Checkweights, 565 can sample.	41

3.11	Weight patch time distribution fit.	43
3.12	Poor-fill patch time distribution fit.	44
4.1	Selected Awesim network nodes.	47
4.2	Simplified simulation network for basecase, 4 workers.	51
5.1	Canadian salmon canning statistics	58
5.2	Canadian wholesale salmon prices	58
5.3	Probable savings per can.	62
6.1	Prototype Machine Vision Can Inspection System.	65
6.2	Schematic representation of the slurry meat injection concept.	67
6.3	Conceptual automated salmon meat addition mechanisms.	69
6.4	Cutting flange defect prototype.	70
6.5	Assembly diagram of the stuffing concept automatic flange defect correction unit. . .	72
6.6	Automatic flange cleaning unit employed at B.C. Packers.	73
7.1	Simplified vision network	77
7.2	Current table layout.	78
7.3	Modified table layout for vision system.	78
7.4	Sensitivity to type 1 errors.	80
7.5	Sensitivity to defect recognition accuracy.	80
7.6	Sensitivity to non-defect handling time.	80
7.7	Sensitivity to on-the-fly flange defect patching time.	80
7.8	Sensitivity to worker time utilization.	80
7.9	Recommended physical layout for the combined vision/trimmer automation option. .	85

7.10 Simplified vision/trimmer network	86
C.1 Complete network employed in the basecase simulations.	118
C.2 Complete network employed in the vision simulations.	119
C.3 Complete network employed in the vision/trimmer simulations.	120

List of Tables

2.1	Mackworth's results.	17
3.1	Can filling defect descriptions.	26
3.2	Survey results.	27
3.3	Statistical data on can filling defects.	28
3.4	Flange patch time distribution fit.	35
3.5	Cross-pack patching data.	37
3.6	Incoming cross-pack population, 599 can sample.	38
3.7	Outgoing cross-pack population, 597 can sample.	38
3.8	Cross-pack patch time distribution fit.	39
3.9	Pre-patching weight distribution.	40
4.1	Function of the major network nodes employed in simulations.	48
4.2	Simulation input.	49
4.3	Tuned efficiency parameters.	53
4.4	Simulation output.	54
4.5	Output comparison.	54
5.1	Operational parameters.	57

7.1	Task times for workers processing certain cans.	75
7.2	Changes in the worker utilizations and activities resultant from automated inspection.	76
7.3	Effect of implementing automated inspection systems.	78
7.4	Effect of adding an automated weight addition systems to the patching table system.	83
7.5	Automation option comparison.	87
A-1	Patch time data for off-line patches.	94
A-2	Patch time data for weight.	94
A-3	Patch time data for poor-fill patches.	94
A-4	Patch time data for cross-pack.	94

Acknowledgements

I would like to thank a number of people for making this work possible. Mr. Fred Nolte, Mr. Tom Coleman, Mr. Rick Heroux, Mr. Roger Gibb, Mr. Richard Gregory and Mr. Harold French of B.C. Packers all provided me with indispensable information. I would like to thank Dr. Farrokh Sassani for his assistance with the simulation and Awesim. Also I would like to Acknowledge Dr. Clarence de Silva for the use of his facilities and the infrastructure of the lab. Thanks to my lab-mate Matthew O'Dor who answered all my stupid questions. To my supervisor, Dr. Elizabeth Croft, who always had time for me and somehow managed to transform my ambiguous writing into lucid text, thank you deeply. Lastly I would like to thank my mates for putting up with me during the stressful periods when no-one else would want to, I appreciate it greatly.

Glossary

- SI, Safety Index
- QI, Quality Index
- W1 - W4, Workers one through four
- DLS, Direct Labour Savings
- ILS, Indirect Labour Savings
- QC, Quality Control

Chapter 1

Introduction

In comparison with other manufacturing industries, the food processing industry has traditionally lagged in technological innovation and implementation. Economic pressures to trim labour costs, increase product recovery and boost quality has pushed concepts such as expert systems, flexible manufacturing and computer simulation into most manufacturing industries. This trend is paralleled in the food industry but again with an appreciable time lag. Many food processors still employ a high ratio of human labour to automation. In these mixed automation processes there is a need to balance the advantages offered by automation; namely, standardization, repeatability continuous operation and, decreased errors with the benefits of human labour; namely, flexibility, dexterity and the ability to deal with incomplete information.

This thesis concerns the analysis of a mixed automation system in a fish processing plant. The general objectives of the work are to analyze one of the existing quality control stations, and to consider a number of automation options for improving the performance of this system.

1.1 Motivation

The pacific salmon stocks on the coast of British Columbia and Washington State appear to be in decline. Even prior to the beginning of 1996, when the Federal government banned harvesting of Sockeye salmon on the Fraser River, and then implemented the Mifflin Plan to reduce the west coast fishing fleet, there was a great deal of concern about the future of the B.C. salmon industry. An example of this tension is the Prince Rupert blockade that the Canadian fishermen set up in the early portion of the 1997 salmon season. Whether temporary or permanent, this decline in salmon stocks has increased pressure on the industry to reduce costs and increase product recovery efficiencies in order to remain competitive.

Whole pacific salmon and canned salmon products have traditionally been a high value B.C. export. Worldwide, B.C. salmon exports average 412 million dollars yearly, or 2.2% of B.C. exports. Processed salmon (canned, smoked, etc.) represents 28% of the salmon exports and whole salmon the remainder [1]. The mean wholesale value of the resource stands at \$578 million per annum [2]. Much work has been done in order to improve and expand the worldwide markets for these products. However, the process for preparing the salmon and packing it in cans has not changed significantly over the past 60-70 years. Thus, the current salmon canning industry, as was the case in the twenties and thirties, requires a great deal of manual labour for preparing, sorting, packing and inspecting the canned salmon products.

In order to reduce costs and increase product recovery efficiencies, the UBC Industrial Automation Laboratory, in collaboration with B.C. Packers Ltd., a major B.C. based fish processor, has undertaken to evaluate, improve, and in some cases re-design the canning process. Product inspection prior to sealing, for quality control and assurance, the focus of this thesis, has been identified as one high potential starting point for the implementation of automation.

In all parts of the salmon canning process, a great deal of repetitive, and relatively low-skill manual labour is employed. A large percentage of this labour is concentrated around the patching tables located at the exit of each filling machine. At these patching tables the filling defects are repaired before the cans are lidded and sealed.

At the patching table, the fill quality of (ideally) every can is assessed, and if necessary improved, based on weight and fill characteristics, including product appearance. The patchers are required to perform both visual product assessment and patching tasks on items that are moving past them at rates of about 4 per second. This work is seasonal, monotonous, and low skill. The workers stand at the patching table throughout their shift in a confined, noisy, and hazardous environment. These adverse working conditions coupled with fatigue may cause product quality to drop during the shift. By replacing the current manned patching tables with minimally-manned automated workcells, both economic and product quality benefits could be realized.

1.2 Canning Process Details

In the present day salmon canning process the ‘cannery dressed’ (gutted, tailed, headed and finned) salmon are sliced width-wise into steaks which are routed to the filling machine. The salmon steaks are volumetrically metered and pressed into the cans within the filling machine. The filled cans exiting the machine are inspected, and fill defects, including weight deviation and product appearance, are repaired. The inspection and ‘patching’ or repair duties are performed manually at a patching station located at the exit of the filler and prior to vacuum sealing of the cans. Figure 1.1 shows a schematic representation of the salmon canning process currently employed at B.C. Packers Ltd. Three different sizes of cans are processed; 1/4 lb., 1/2 lb. and 1 lb or ‘talls’. In addition to this variation six different grades or qualities of salmon are processed. Factors affecting

the grade of the fill include; species, whether the fish is fresh or frozen, filler characteristics and the postmortem age of the fish.

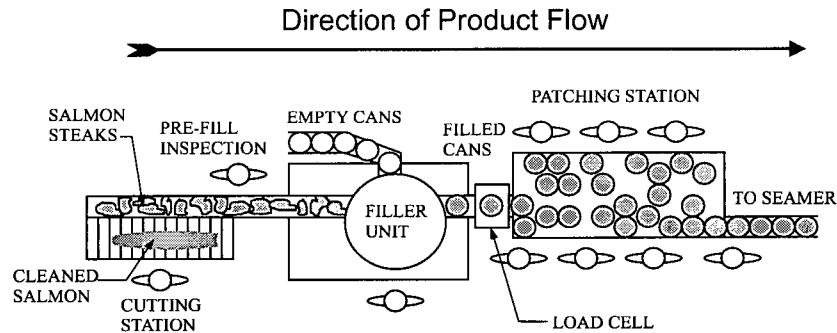


Figure 1.1: The salmon canning process.

1.3 The Patching Table

The patching station is composed of a rectangular table bisected by a number of conveyors. The table is fed by a single line of cans exiting the filler at up to 250 cans per minute as shown schematically in Figure 1.2.

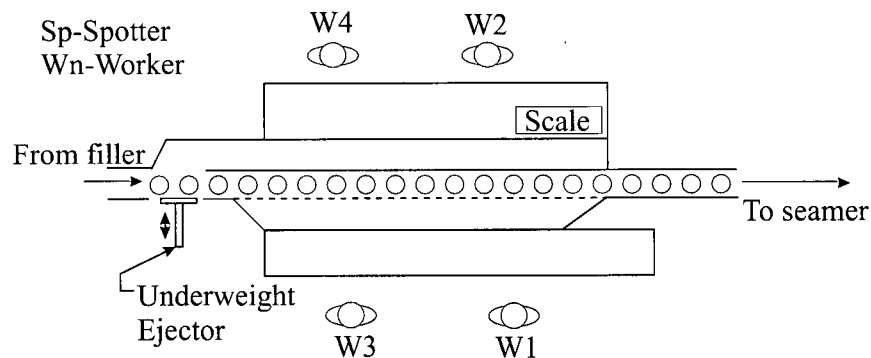


Figure 1.2: Schematic of the patching station.

Figure 1.3 shows a patching table during normal operation at the B.C. Packers plant in Prince Rupert, B.C., in September 1996. There are from 4 to 7 workers per patching table, depending

mainly on the quality of the fish being processed, but also on other factors such as can size, the condition of the fill, and the intended product market. On most half-pound and one-pound lines an automated high-speed weighing system (a checkweigher) ejects the underweight cans to a parallel conveyor for patching. On most quarter-pound lines only those cans which visually appear to have low weight are pulled from the line, manually weighed and, if necessary, corrected.

The last worker on each patching line is specifically responsible for quality control inspection. This worker is called the 'spotter' and generally performs no patching operations. The remainder of the workers 'pull' and 'patch' (repair) offending cans from the line in addition to patching those cans the spotter has pulled for repair. These patching tasks include safety, appearance and regulatory defects as detailed in Section 3.1.



Figure 1.3: Patching table during operation.

1.4 Research Objectives

As described above, the patching table is a manually operated quality assurance workcell for the can filling operation. The operations which take place at this workcell fall into two fundamental categories, *defect identification* and *defect correction*.

Defect identification is done primarily through visual inspection. The exception is the underweight detection, which is automated on certain lines, through the use of a checkweigher. *Defect correction* is done almost exclusively by the patching table workers (i.e., manually) although some automated equipment (e.g., sprayers which automatically wash bits of meat off the rim of each can and trimmers which attempt remove skin and bone over the flange) may be employed on some patching lines.

The objectives of this research then, are to:

1. examine the patching table process
2. identify high potential candidate operations for automation as well as those manual processes which could be streamlined
3. develop and analyze a model of the workcell, introduce automation into this model, and finally optimize this model, through simulation and experimental prototype development
4. select the optimal system configuration based on this analysis.

The result of this work will be the analysis, design and cost effectiveness evaluation of automating the patching table. As in the manual workcell, the automated workcell will need to perform two fundamental tasks, namely, defect identification and defect patching or repair. The underlying problems that need to be addressed in this work include defect classification, can sorting, queueing and processes diagnostics.

In order to accomplish these objectives given the current state of the workcell, the following steps will be required:

1. In order to properly judge the benefits and detriments of modifying the existing patching system, a clear and thorough understanding of the current workcell is required in order to facilitate the modelling and simulation of the present process. Each task undertaken in the workcell must be identified and modelled. The information used by patching workers in identifying problems and the flow of information concerning those problems is essential to both model and then modify the process. As well, processing times required by the human workers in the areas of inspection and repair of defects, and success rates for the workcell must be compiled.
2. For each identified task in the workcell, the current level of automation, and the amenability of the task to further automation must be considered. Task frequency, importance and difficulty are factors which must be considered in a preliminary assessment of each task.
3. As outlined above tasks in the patching workcell can be broadly classified into a) inspection/sorting tasks and b) repair tasks.
 - a) In considering the inspection/sorting tasks, the visual cues and techniques used by the human inspectors should be determined in order to assist in identifying candidate machine vision techniques, the proposed automated inspection technique.
 - b) In considering the repair tasks, identifying the repair techniques employed by the workers will help to direct the selection of automated repair techniques. Other factors, such as the costs and availability of current or proposed equipment to perform each task, are also important.

Finally, the techniques and strategies used by the workers as a group to deal with the large numbers of cans, and the inherently large amounts of associated information, require identification.

4. Past experience with industry indicates that the current feasibility of fully automating the entire patching table is low. Hence, those tasks for which automation is least suitable will remain as manual tasks. However, improvements through the use of modified information flow, information tagging and pre-sorting must be identified.
5. Implement the modifications to the workcell in a simulated workcell in order to gauge their effectiveness and consequences on the system.
6. Perform economic evaluations of the plausible system configurations.

The result of this work will be a number of workcell design concepts or scenarios incorporating various levels of automation. Coupled with these will be economic, production and quality investigations. It is the intent of this investigation that sufficient information be portrayed such that management personnel may select a scenario which meets their needs and constraints.

Chapter 2

Literature Review

This chapter examines some of the more common methods for modelling manufacturing systems, including simulations used for modelling of food production systems. As well, the factors affecting manual inspection accuracy are considered. A review of semi-autonomous production systems is presented and, finally, methods for evaluating automation options are discussed as presented in the literature.

2.1 Modelling Manufacturing Systems

The patching table is a manufacturing quality assurance (inspection and repair) workcell employing a mix of automation and manual labour. The incoming components to the workcell are filled cans of unknown nature. The workcell product is an inspected filled can of salmon which is ideally defect free. The physical can size is constant over a shift. The contents of the can (the 'fill') can change in grade and, therefore, quality during a shift. However, such a change would be identified by the production manager, and would be treated as a separate batch. Due to the stochastic nature of the incoming components the workcell operations are also flexible and stochastic in nature.

The patching table operations do not fit well in the typical classes for flexible manufacturing, namely, volume flexibility, routing flexibility or product flexibility. While there are some similarities to product flexibility, this generally pertains to the production of a varied component mix, rather than the ability to cope with a variety of incoming component states. However, there are certainly manufacturing modelling methods in the literature which can be applied to this problem.

Methods for modelling manufacturing systems are distinguished by model characteristics [3]:

1. static, where changes over time are not significant;
2. dynamic, where changes over time are considered;
3. deterministic, where the random behaviour of the system is not significant;
4. stochastic, where randomness is considered;

Modelling techniques include Petri nets [4, 5, 6], Markovian modelling [7, 8], queueing network theory [9], analytic discrete event systems (DES) [10, 11], and discrete event simulations [12, 13, 14, 15, 16]. Each of these techniques is reviewed briefly here.

Recently, Petri nets have been advanced as a high level modelling technique ideal for manufacturing systems; however, Petri nets still require substantial development because of the huge processing times required to model all but the simplest systems [17]. More recent advances, including modularity and stochastic nets [4, 6], indicate that Petri nets may be a viable modelling tool since the various ‘breeds’ of net may be dynamic or static, stochastic or deterministic. However, even with these improvements to Petri net modelling, Petri models are considered to be useful only in the first, high-level stage of the design and modelling process [6].

Markovian modelling addresses the processes that each part undergoes. These models are static and deterministic and they can easily be implemented in matrix form, for example, on a

spreadsheet. Each element in a Markov transition matrix represents the relative frequency of a specific transition of the product or machine from one state to another. Generally, Markov models come in two forms; first order and multi-order. First order models require a single step transition between states whereas multi-order models facilitate the description of several steps in the state to state transition.

Queueing network theory is used to describe the processing and waiting times for manufactured products. This model is static and stochastic. It can be implemented using software such as MANUPLAN II [3] and MUTIQ [9]. In this modelling method, parts are transported between machines by a material transport system, for example, AGVs or conveyors. When a part reaches an occupied machine it is placed in a queue or buffer to await processing.

DES are dynamic systems, similar in nature to Petri nets, which evolve with the discrete occurrence of events. DES are generally asynchronous and non-deterministic. As a result they ideally model the behaviour of systems such as production and manufacturing workcells.

Discrete event simulators such as STARCELL [3] and AWESIM [18] are used to describe processes undergoing discrete and continuous changes in state and system variables. These simulations are regarded as the best method by which to evaluate FMS systems, [9]. These simulators enable the dynamic alteration of parameters during a simulation. These might include changing task times, breakdowns and maintenance.

The major drawback in using petri-nets, DES and queueing network models is that they all become extremely complex when modelling a system of realistic size. For this reason these methods work quite well with modest problems but encounter difficulties when attempting to model complex systems. Furthermore, queueing networks are static and, therefore, do not allow the alteration of parameters with time. Markov methods, on the other hand, present difficulties when attempting to implement changing transition probabilities due to their deterministic nature.

In general, mixed manual/automata workcells have successfully been dealt with through the use of discrete event simulations [15, 14]. Hence in considering the modelling and simulation of the patching workcell it appears, upon examination of the methods available and of the processes modelled using them, that a discrete event simulation approach is most suitable and reputable.

2.2 Food Processing Simulations

The literature on modelling and simulation of food processing is mainly concerned with the material handling requirements, e.g., [19, 20] and mathematical modelling of specific processes [21], for example, extrusion, cooking, cooling and dehydration. These treatments draw on theory from fields such as fluid mechanics and heat transfer, e.g., [22]. However, there are a few worker based simulations considered in the literature and these generally employ a discrete event simulation as stated previously.

In one more relevant example, Pizza Hut Restaurant Inc. simulated the operation of their restaurant using the SIMAN simulation package to predict bottlenecks, peak performance equipment capacity, and customer service times as well as their sensitivity to input parameters (e.g., staffing levels) [15]. In order to properly model the situation, time and motion studies were employed to determine task times, such as cooking, seating customers, order taking, delivery times, etc. Internal data from the restaurant chain was used for modelling events such as arrival distributions and order breakdown.

In another example, Wells [14] discusses a SLAM II simulation of the complete processing of Craw-fish including automated washing, grading and cooking with manual picking and packing operations. In this simulation, time and motion studies again determined task completion times where manual labour was involved. In this example a discrete event simulation was appropriate

since continuous factors such as temperatures etc. were not integral to the investigation.

2.3 Human Factors

In the examination of mixed automation systems it is important to consider and model, where possible, the behavior of the personnel involved in the system. This is an extremely difficult task, as indicated in the following review; however, it is possible to identify the major factors which affect the human interaction with the system. These factors must be carefully considered when adjustments to the system are made for automation purposes.

2.3.1 Inspection Accuracy

In the review that follows, terminology, which is used to discuss the performance of workers, can be confusing. Thus, a brief description of the types of inspection errors which are used to measure worker performance is given here.

Two types of errors are defined for the inspection of an item, Type 1 and Type 2. When a signal (namely a defect) is present the item is a positive, (P). When no signal (defect) is present the item is negative, (N). Type 1 error, which is also referred to as a 'false positive', (FP), refers to the identification of a signal when no signal is present, (N). In the specific case of the patching table this would mean the classification of a can as defective when it is non-defective, (N). The FP rate is defined as the number of false positives divided by the total number of positives present in the batch, i.e., (FP/P) . Type 2 error, or false negatives, (FN), are the identification of a non-signal when in fact one is present, (P). This would be the identification of a can as non-defective, when in fact it is defective. The FN rate is defined as the number of false negatives divided by the total number of negatives in the batch, i.e., (FN/N) .

Two types of correct decisions are also defined. True positives, (TP), are the identification of a signal, when one is present, (P). This is the identification of a defect when one is present. The TP rate is the number of true positives divided by the total number of positives in the batch, (TP/P). True negatives, (TN), are the identification of a non-signal when no signal is present, (N). In our case this is the identification of a defect-free can as non-defective. True negative rate is the number of true negatives divided by the number of negatives in the batch, (TN/N).

By these definitions, the following identities hold:

1. $TP + FN = P$
2. $TN + FP = N$

2.3.2 Factors Affecting Inspection Accuracy

The use of manual labour in manufacturing has both advantages and disadvantages. Advantages include the human ability to quickly and accurately identify and classify defects, to deal with variation and unforeseen factors, and to reason and solve problems with incomplete data using heuristics, a priori knowledge and intuition. Disadvantages include relatively slow working speeds, forgetfulness, high cost of compensation, required training, absenteeism, and inconsistencies in work quality. Megaw [23] lists four groups of factors which affect these inconsistencies, specifically with regard to industrial inspection accuracy:

1. Variables that the inspector brings to the task: e.g., visual ability, performance, age, experience, personality, intelligence, etc.
2. Physical and environmental factors: lighting, the workplace layout, background noise and visual aids.

3. Organizational factors and characteristics of the job situation: e.g., the number of inspectors, the type of training and feedback received, the number and duration of rest breaks, the work schedule (shift-work vs. 9-5), social aspects, etc.
4. Factors pertaining specifically to the task: e.g., paced or unpaced inspection, the probability of defects, the complexity of the inspected product, the density of the products, etc.

Many of the factors considered by Megaw [24], e.g., noise, visual acuity, training, etc. do not change over the period of a shift, and are unlikely to be affected by the introduction of automation modifications to the workcell. The effect of workcell automation on sociological factors which contribute to inspector effectiveness is not predictable, and may, or may not be negligible. However, those factors which can be expected to be influenced by modifying the workcell are treated in the following sections.

Vigilance

The process of industrial inspection has been strongly linked to the branch of industrial organizational psychology known as *vigilance*. This link is especially strong for paced inspection tasks, such as the patching table inspection task considered herein. One of the pioneers in this area of research, N.H. Mackworth, defined vigilance as: "A state of readiness to detect and respond to certain specified small changes occurring at random time intervals in the environment" [25]. Czaja and Drury [26] generated a taxonomy of inspection tasks from which they formulated the following categories:

- Monitoring: observing a continuous process and reporting any deviations.
- Examining: searching items, or arrays of items, for defects. Judgements are made without numerical measurement.

- Measuring: using instruments and measuring tools to provide numerical measures on which decisions are based.
- Patrolling: checking and organizing the work of other inspectors.

Quality inspection, such as that done at a patching table falls into the second category, namely examining. Osborne [27] and Murrell [28] state that, based on reports from inspectors, the complex task of inspection can be accurately and easily modelled. The proposed model is essentially mental template matching. The inspector has a mental picture of an good item or product which they then compare with the products requiring inspection. Deviations from this mental image are grounds for rejection. This is similar to the method of template matching used in machine vision inspection systems. In many cases the initial segregation of fault/no-fault is followed by other operations such as classification, logging and/or repair.

Vigilance Decrement

‘Vigilance decrement’ was first reported by Mackworth in the 1940’s. Vigilance decrement refers to the human phenomenon of declining efficiency with time while engaging in vigilance activities. At the time of Mackworth’s research the Royal Air Force (RAF) was concerned with the efficiency of their radar operators on anti-submarine patrols. Mackworth conducted a number of tests and found that after 30 minutes a significant deterioration of efficiency occurred followed by a steady but small decline afterwards (see Table 2.1). In all tests a vigilance decrement was detected.

Vigilance is concerned with three main factors: detection rate, detection latency (time), and false positives. Mackworth’s test did not explore changes in the last of these, namely, false positives. This is an important factor that has been addressed by Broadbent and Gregory [29] using a signal detection theory (SDT) of vigilance. This work “...suggests that the decline in detection rate can

Task	Subjects	Percentage of correct detections in half-hour periods of task			
		1	2	3	4
Clock test I	25 RAF cadets	84.3	74.2	73.2	72.0
	25 Naval ratings	82.3	72.7	72.7	72.7
Synthetic radar task					
Bright echo*	17 Naval ratings, 5 RAF cadets, 3 wireless operators	94.0	91.8	84.3	88.5
Dim echo	23 RAF cadets	72.0	62.2	61.2	60.1
Main listening task	22 RAF aircrews, 3 Naval ratings	80.4	71.6	70.6	64.0

* Average of 2 runs about 1 week apart.

Table 2.1: Mackworth's results.

be viewed in terms of either an increase in the criterion or as a reduction in sensitivity" [25]. Essentially, the inspector suffers 'drift' with regards to their mental picture of a signal (a positive, i.e., a defect), almost always increasing the criteria for the signal. This leads to a decrease in the number of correct detections and also decreases the number of false positives. As a result the inspectors sensitivity remains constant but their decision criterion is shifted to a more conservative value.

From another perspective, in vigilance experiments on multi-featured inspection tasks Craig and Colquhoun [30] found that, although performance decrements were observed (i.e., the percent of correct detections declined), the decrements could be attributable to the ratio of signal detections, positives (P), dropping to unity. This means that the subjects probably reduced their responses in order to match the occurrences (i.e., probability matching) and not due to the monotony of the task. This vigilance decrement, however, was not corroborated by their second experiment. In this test the same decrement was not observed to any significance for either true positives or false positives. Deese [31] points out that "detection rate does not always simply decline but may show some improvement or fluctuate". Many tasks show no decrement, especially complex tasks [32].

A number of authors are of the opinion that vigilance decrements are not found in real world monitoring, [33, 34, 35, 36, 37]. However, the vigilance decrement has not been conclusively ruled out for real industrial inspection tasks. Many 'real world' experiments reveal vigilance decrements, [38]. When a vigilance decrement is observed it is usually completed by 20-30 minutes and half of the decrement occurs in the first 15 minutes [39]. This conflicting data is indicative of the need for more real world experimental studies.

From this review it is apparent that the formulation of a vigilance decrement in the present work would be problematic without extensive on-site experimental data regarding worker behaviour. However, such a study is outside of the scope of this thesis. Furthermore, specific studies of worker behaviors beyond those related to the physical manipulation of the cans is undesirable from the point of view of union/management relations, and were specifically excluded by management in this study. To compensate, in this work, all experimental data was collected after half an hour of working and thus should represent the 'degraded' or steady state inspection efficiencies.

Speed Decrement

Many experimenters have confirmed that there is an exponential vigilance degradation associated with increasing event rate, namely, the rate at which a human must examine individual items. This degradation is noted in two general situations: (i) event rate is increased but fault signals per period are held constant (i.e., defects/time = const); [40, 41, 42, 43, 44, 45, 46] and (ii) the event rate is increased and the fault probability is held constant (i.e., defects/lot = const): [47, 43, 48, 49, 46]. In addition, the false positive rate has been shown to decrease linearly with event rate [24, 50, 51]. The generalized trends representing these findings are shown in Figure 2.1 and are adapted from Megaw [24]. Megaw states that in order to estimate these two curves, performance data is required for two average inspection times (or inspection rates), one of which should be near the point where

the true positive curve starts to break away. The work of Williams and Borrow [52], shows that detection rate remains constant regardless of event rate until the event rate is about 8° per second (at the eye) at which point the exponential decay is observed. They conducted a moving visual search experiment in which individuals were shown a moving visual search scenario which was continuously scrolling on a screen to keep it in constant view. The speed of this display was altered and the time taken to complete the search recorded.

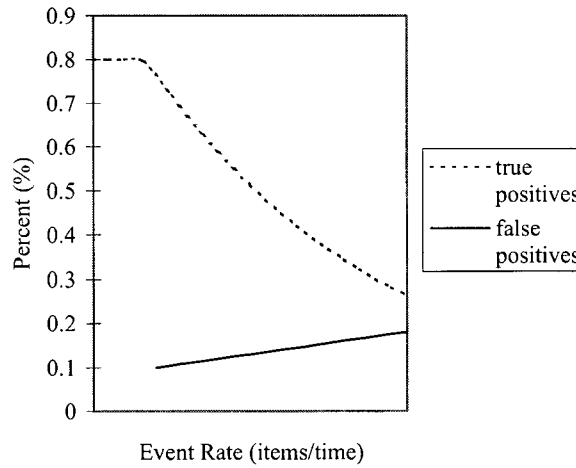


Figure 2.1: Generalized relationship between inspection time and accuracy of performance.

The decay of the true positive rate curve is given by Drury [51] and many others as the standard equation for inspection and search tasks as a function of inspection time, namely:

$$p(\text{correct detection}) = 1 - \exp\left(-\frac{t}{t_0}\right) \quad (2.1)$$

where:

t = time to the local the target,

t_0 = scale factor dependent upon the size of the field, difficulty of recognizing the target, etc.

Defect Density Factors

It has been shown by many authors that as the defect rate or defect density is increased, from near zero the true positive rate increases rapidly and then levels off. On the other hand, the false

positive rate decreases rapidly and then also levels off. These trends are shown in Figure 2.2 taken from Harris' paper [53]. These findings are task specific, however they can be generalized.

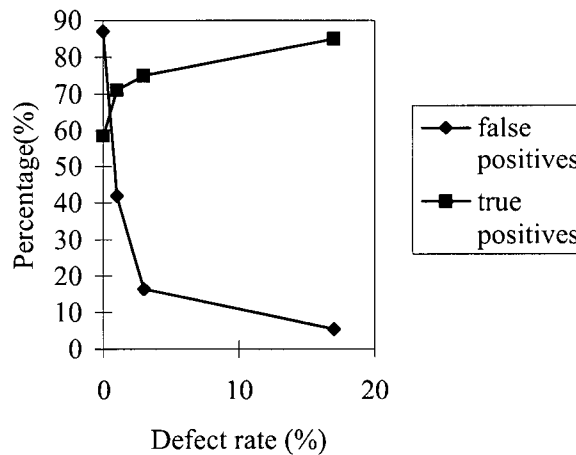


Figure 2.2: Effect of defect rate on inspection accuracy.

In summary, the major factors affecting inspection accuracy are line speed, defect density and time. Any modifications to the workcell which significantly affect these factors, may lead to alterations in inspection accuracy.

2.4 Semi-Automated Manufacturing Systems

The patching workcell considered in this work can be classified as part of a manufacturing workcell. Automation of this labour intensive workcell may lead to cost savings and quality enhancements. A definition for a Flexible Manufacturing System (FMS), adapted by Mansfield from the International Institution for Production Engineering Research, is: "...a production unit capable of producing a range of discrete products with a minimum of manual intervention; it consists of production-equipment workstations (machine tools or other equipment for fabrication, assembly, or treatment) linked by a materials-handling system to move parts from one workstation to another, and it operates as an integrated system under full programmable control." [54].

In the patching workcell considered in this work, the materials handling system is relatively simple. There are a small number of components and the product flow is unidirectional. However, in the present system the product attributes (namely, safety, appearance, and regulatory attributes) are not deterministic. Thus, the tasks undertaken in the workcell reflect a much more complex set of processing requirements than for example, executing typical machine-tool operations in a standard FMC.

Robotic systems capable of performing complex, dextrous tasks such as repair or disassembly are generally very costly [55]. The cost involved in such tasks rises exponentially with the task difficulty due to the costs incurred through hardware and software acquisition and development. For these, and other labour related issues, it is not considered reasonable or feasible to consider the full automation of the workcell. Instead, a mixed automation approach is considered to have the best potential for success.

An example of a semi-automated workcell is the U.S. military's printed circuit board repair workcell [56]. In this work, it was determined that automating 63% of the repair tasks and leaving the remainder to human workers was economically superior to full automation. System 'learning' or updating was facilitated with the aid of the workers who continually updated an expert system data base.

In another example, Hattersley Newman Hender utilized a mixed human-machine manufacturing system in order to fabricate valve bodies. The cell employs automated machining and material handling equipment and is linked to a manual work station which handles tasks requiring a high level of dexterity. Video display units are employed at the manually operated o-ring fitting station and other human/machine interfaces to facilitate the integration of the human workers and automated equipment.

In summary, mixed automation is well suited to situations involving a mix of both complex and

simple, repetitive tasks. The use of a video interface with human workers can provide an effective bridge between the human worker and the automated system.

2.5 Economics

In order for automation options to be evaluated it is required that the economic benefits of the various options be considered. Monetary savings or profit increases are the main factors by which a company will accept or decline a proposed investment in automation. Factors such as quality improvement and work place safety as well as employee moral also need to be considered although they generally play a lesser role in the decision making and are usually evaluated by non analytic means, hence, their contribution to the decision is based on the judgment of management.

A typical cost breakdown for the analysis of an advanced manufacturing system is shown in Figure 2.3, [57]. Numerous authors stress the need for 'relatively ill-structured' costs (RISC), such as quality and flexibility costs to be addressed in the economic analysis of advanced manufacturing systems; [57, 58, 59]. Relatively well-structured costs (RWSC) include items such as labour and capital costs. Son [57] stresses the need to consider the shift in labour from 'direct labour' to 'indirect labour' when implementing automation. For example, the shift from manual labourers to computer programmers or technicians. Furthermore, many economic evaluations do not deal with the savings obtained through improved quality and flexibility. Some exceptions are the Taguchi Method [59], the Analytic Hierarchy Process (AHP) [60, 61] and Son's methodology [57].

Generally there are three investment appraisal methods employed in the literature for the evaluation of investment options based on RWSC factors.

1. Degree of necessity
2. Payout time

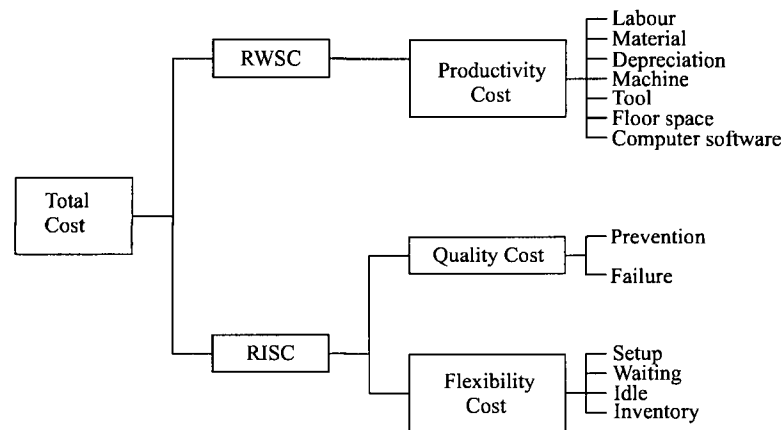


Figure 2.3: Cost system supporting analysis of advanced manufacturing systems.

3. Rate of return

The degree of necessity indicates an investor requires the improvement. This is usually the overriding argument in decisions concerning replacement of broken equipment, recovering after disaster, government regulated improvements, or equipment upgrades that are relatively unprofitable but required in order to meet competition. This type of decision making is generally qualitative and hence comparisons based on this methodology are ill-defined.

Payout time analysis is by far the most widely used method for investment comparison. In this method the number of years required for cash earnings or savings generated by the investment to payoff the capital expenditure is calculated, [62, 63, 64]. Drawbacks of this methodology include an overemphasization of the importance of early cash returns, it ignores projects economic life and fails to consider project earnings after the initial investment has been recovered.

Rate of return analysis of investments relates the project's anticipated earnings to the amount of capital tied up during the projects estimated life. Various methods exist for the calculation of the return including; the accounting approach, the operating return approach, the present worth approach and the discounted cash flow (DCF) approach. The DCF approach eliminates some of the

disadvantages associated with the other approaches and has thus been adopted in this study. What the DCF does is that it sets all the present worths of cash flows (in and out) to zero. Whatever discount rate applied to the cash flows to make their discounted values zero is the DCF rate of return, [62, 63, 64]. This method assumes continuous compounding and in addition it assumes that the earnings generated are reinvested in the project or in some other investment in which it will earn at the same rate as the project in question.

In summary, several mixed automation systems were examined including two in the food production industry. From the modelling of these systems and the subsequent simulation it was determined that time motion studies are an ideal way to gather information about manual processes. From this literature review it has been determined that the most reputable simulation method for mixed automata systems is a discrete event simulation. Factors affecting inspection accuracy have been examined and the methodology by which to economically evaluate the automation options has been determined. A discounted cash flow approach will be implemented along with the payout period method.

Chapter 3

Data Acquisition and Modelling

3.1 Quality Control Questionnaire and Interviews

Many quality evaluations made in the food industry are subjective, because they depend on the judgment of individuals, e.g., plant quality control (Q.C.) inspectors, government inspectors, consumer panels, and of course, consumers. To automate quality evaluations on the can-filling line, the features which affect the quality of the filled can must be well understood. That is, fill features and fill defects must be identified in order to form a basis on which filled cans may be classified as either defective or acceptable.

In order to gain an understanding of those defects which require patching, a questionnaire was prepared for B.C. Packers Q.C. personnel. The survey was composed of a series of short answer questions and a series of 36 colour photographs of various filled salmon cans. The cans were hand packed in the Industrial Automation Laboratory. The appearance of the filled cans was designed to represent possible fill defects as well as acceptable fills based on initial information gained from

interviews with B.C. Packers Q.C. experts.¹

The participants were asked to identify on the photographs any Q.C. problems and to indicate their severity. A copy of the questionnaire is included in Appendix B. Through the results of this survey and further consultation with industry experts, the filling problems listed in Table 3.1 were identified.

<i>Filling Problem</i>	<i>Description</i>
Cross-pack	Steaks inserted sideways with skin showing.
Flange obstructions	Meat, skin or bones over the flange.
Underweight	Can does not satisfy government weight regulations.
Overweight	Can is overfilled and may not seal properly.
Poor colouration	Meat is not a uniform colour (bruises, blood-spots, mixed colours).
Flange damage	Can is scratched or dented during the filling process
Poor-fill	Too many pieces, ragged cuts, fins, or other misc. defects.

Table 3.1: Can filling defect descriptions.

The most severe defects identified in the survey were flange obstructions and flange damage. Flange obstructions include skin, flesh and bones on the flange of the can, Figure 3.1(a). These problems can result in a seam defect when the lid is attached to the can. This could lead to an improper (failed) seal, and in the worst possible case, spoilage of the can contents. Cross-packs, Figure 3.1(b) and flange obstructions were identified as the most frequent defect types. More detailed results are listed in Table 3.2. The poor-fill category was not included in these results since it is a catch-all category for a number of less common, ill-defined miscellaneous defects.

Using the expert comments the following descriptions of a well filled and poorly filled can were developed. A well filled can, Figure 3.1(c) consists of one to three clean cut transverse salmon steaks with a uniform bright colour. A poorly filled can may consist of any number of the following problems: ragged cuts, too many small pieces, Figure 3.1(d), bones on the surface, bruises, blood

¹One major concern in creating this survey was that the pictures of hand-packed cans would not resemble typical cans found on the canning line. However, one of the Q.C. experts surveyed used the survey pictures to create a training document for new patching table workers. This was taken as an indication that the pictures were reasonably realistic.

spots or mixed colours. These defects are in addition to flange defects, cross-pack and incorrect weight.

<i>Problem</i>	<i>Severity*</i>	<i>Frequency*</i>
Cross-pack	4.0	3.75
Flange obstructions	5.0	3.25
Underweight	4.0	2.25
Overweight	3.75	2.00
Poor colouration	3.75	2.00
Flange damage	4.50	2.00

*5-high, 1-low

Table 3.2: Survey results.



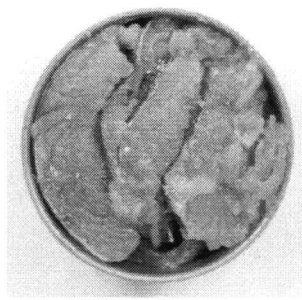
(a) Flange defect
(Bones over the
flange).



(b) Cross-pack.



(c) Well filled can.



(d) Too many pieces.

Figure 3.1: Various filled salmon cans.

The identification of can-filling problems on the pictures in the third section of the survey were somewhat inconsistent. The experts disagreed on the filling problems in 8 of the 36 images used in the survey. This indicates that the characteristics of a well filled can are not crisply defined. The inconsistent answers could also be due to the open ended nature of this question.

For the purposes of this project, specific data on filling defects were generated by Q.C. personnel under the direction of Mr. Roger Gibb, of B.C. Packers, during a previous canning season (August 1995). Analysis of this raw data produced the statistical information listed in Table 3.3 concerning filling defects encountered at the patching table. One can note that, the observed frequency of the

filling defects is consistent with the perceptions of the Q.C. experts. That is, cross-pack and flange defects occur more frequently than other defects.

<i>Defect</i>	<i>Mean (defects/min. *)</i>	<i>Std. Dev.</i>	<i>% Total number</i>
Cross-Pack	12.08	4.88	30.9
Flange obstructions	15.45	7.25	39.5
Underweight	6.27	5.9	16.0
Overweight	1.38	1.17	3.5
Other	3.96	4.29	10.1
Total	39.15	12.11	100.0

*Average line speed was 249 cans/minute.

Table 3.3: Statistical data on can filling defects.

The overall rate of defects arriving at the patching table was about 15.7%, or about 39 defective cans per minute. One can note that, these are only the number of defects observed while the line was running and therefore, may not represent the actual number of defects which could occur (i.e., some defects may have been missed).

A classification system has been developed for the grouping of defects based on defect severity and criticality. Three categories have been generated: safety, regulatory and appearance in order of descending criticality.

3.1.1 Safety Defects:

This is the most important category since it represents those defects that may compromise the can seal, possibly resulting in spoilage. Flange obstructions constitute most of this category. Bones or the tough salmon skin are the most common causes of flange obstructions. In addition to spoilage problems, missed flange obstructions which are later detected through “dud detection”² after a specified storage time, represent a loss of finished product for the producer. For the same reasons, bent or damaged can flanges constitute a safety hazard. In this case, when detected the

²The process of measuring the deflection of the can lid as a representation of the can’s seal. This process is done on all salmon cans prior to labelling. Three percent of the most deviant cans are ejected and hand inspected.

can must be discarded and the meat repacked in a new can to ensure a proper seal. Foreign object contamination also represents a safety hazard; however, this is a relatively rare occurrence. At present, the patching table workers do not specifically look for foreign material, but when found, it is of course, removed. Inadequate vacuum is a safety failure that cannot be directly verified at the patching table but is influenced by the way the can is filled. The amount of head space, (vertical height from steak surface to the can flange) and the void fraction, (ratio of volume displaced to available volume), influences the level of vacuum achievable when the can is lidded and sealed. Since inadequate vacuum may lead to spoilage; B.C. Packers requires a vacuum of not less than 2 inches of H_2O ; however, they have not determined an optimal head-space or void fraction required to meet this requirement.

3.1.2 Regulatory Defects:

This category has a single member, namely, weight violation. B.C. Packers has indicated the need to control the net fill weight such that the standard deviation is not greater than 1.5g for 1/4 lb., 2g for 1/2 lb. and 3g for a 1 lb. can. On 1/2 lb. lines B.C. Packers attempts to patch all cans 9 grams or more below the label weight of 213g. Statistical quality control is done by means of sampling throughout the shift. The filler is deemed in control if not more than 2.5% of cans fall below one tolerance (-9g) and not more than 0.5% fall below two tolerances.

3.1.3 Appearance Defects:

These are aesthetic defects and hence they are less critical since they do not pose a health or safety concern. Instead, they reduce the appeal of the product for the consumer. Large voids in the can are undesirable since they give the impression that the can is underfilled, regardless of the weight content of the can. Cross-packed steaks are steaks which, during filling have been flipped onto their

side exposing the skin to the top surface. The appearance defects are listed below:

- skin flaps.
- cross-pack.
- blood spots.
- fins.
- too many pieces.
- ragged cuts.
- tail pieces.
- colour non-uniformity.
- large voids.

3.2 On-site Experimentation

A number of on-site experiments were performed to acquire data on each defect category as well as to obtain overall process data for modelling purposes. The B.C. Packers cannery in Prince Rupert B.C. was selected as the site for these experiments. The experiments were conducted during the 1996 summer canning season. The objectives for the experiments at the cannery were:

- To measure the defect production rate and distribution exiting the filling machine.
- To measure the failure rate of the patching table.
- To obtain images to test can feature extraction algorithms.
- To observe the operation of the patching table.
- To collect data from the high speed weighing system.

The defect production rate of the filling machine and the failure rate of the patching table were estimated by taking a series of digital pictures of cans exiting both the filling machine and the patching table (i.e., cans entering and leaving the patching table). In order to obtain an equal sample from the six-pocket filler, over an extended sampling time an image of every seventh can entering or exiting the patching table was acquired. (The images of entering and exiting cans were obtained over two different time periods due to equipment limitations.) The pictures were then

reviewed to provide statistical data on the can filling defects entering the patching table, as well as defects leaving the table (i.e., defects missed by the patching table staff). The digital pictures were acquired directly into a computer using a electro-optical trigger, a digital camera and a frame grabber, Figure 3.2.

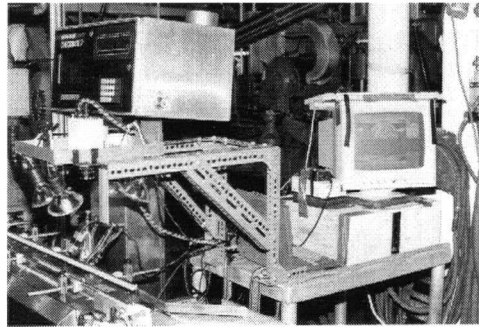


Figure 3.2: Experimental setup.

The operation of the patching table was videotaped and observed to obtain data on worker interaction and task identification. The video of the patching table was also used to identify the manual patching motions used to repair different fill defects. The end of day totals of underweight cans were collected from the high speed weighing machine located just upstream of the entrance to the patching table.

3.2.1 Layout

Experiments were performed on canning line 4, running 1/2 lb. tapered cans. The line was equipped with a Mettler Toledo HI-SPEED CM9400 can and bottle checkweigher which would divert underweight cans to an auxiliary conveyor on the patching table. The table employed four patching workers. The locations of the workers at the table are indicated in Figure 3.3.

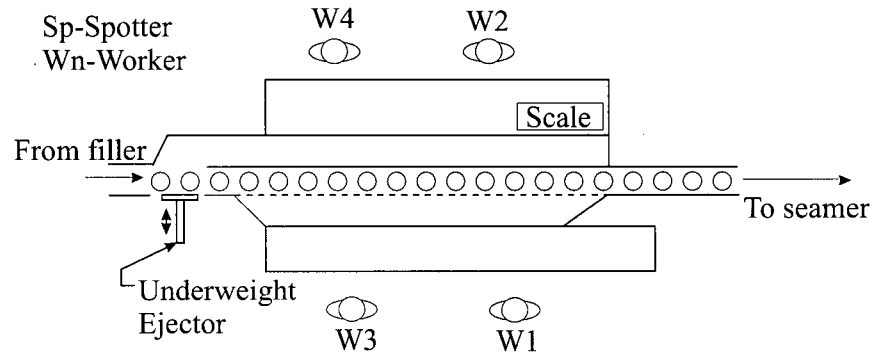


Figure 3.3: Patching station.

3.2.2 System Setup and Product Input

The line was run with previously frozen Pink salmon for most of the experimental work, followed by a small run of Chum salmon, also previously frozen. All data is with regard to Pink salmon unless otherwise indicated. The checkweigher was set to eject those cans weighing less than 204 grams, (i.e., 1 tolerance). The filler was set to a fill weight of 214g, 1g over the desired weight of 213g and the line was operated at a rate of 232 cans/min. The product being produced was grade four salmon. The grade of the product affects operations carried out at the patching table. The lower grades, (e.g., 4-6), have a lower priority for patching cross-packed cans. This lower priority means that the severity of the cross-pack which is acceptable increases as the grade drops. This is a function of the current state of the table, if the table is idle, during a low grade run the patchers will patch minor cross-packs.

3.3 Experimental Results and Data Modelling

3.3.1 Worker Tasks and Table Operation

One of the fundamental requirements in the modelling of the patching table is to clearly understand the responsibilities of each worker and their interaction with the cans and the other workers. From

the video footage the primary tasks of the workers have been identified and, in addition, any worker interactions have been identified. A schematic representation of the work flow and worker responsibilities is presented in Figure 3.4.

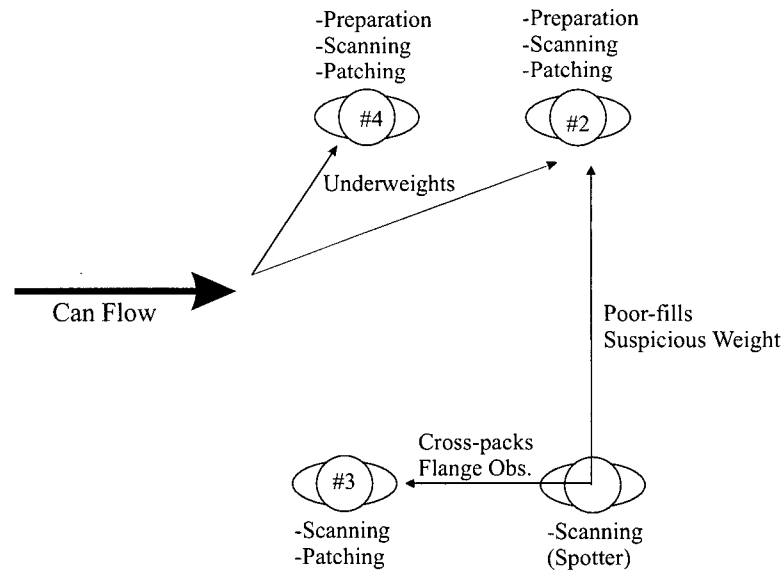


Figure 3.4: Worker interaction and responsibilities.

Worker #1

Worker #1, 'the spotter' has the responsibility of continuously scanning the cans which are leaving the patching table heading for the lidding and seaming machines. This worker is the last point in the quality control chain at which a recovery can be made of the product. The spotter pulls any defective cans which have traveled across the table and passes them to other workers for repair. Those cans with flange obstructions or those that are cross-packed are passed to worker #3 for repair. The spotter also pulls off the line those cans deemed to be 'poor-fill' (discussed in section 3.3.5) and directs them to worker #2 for repair. In addition the spotter repairs certain cans without taking them off line if the operation is quick and easy like tucking in some meat with a finger. The spotter also presses many of the steaks flat with their hands, and thus facilitates easier

lidding.

Worker #2

Worker #2 patches those poorly filled cans directed to her by the spotter and repairs some of those underweight cans ejected by the checkweigher (i.e., those which are not pulled by worker #4 from the auxiliary conveyor). When not occupied worker #2 will pull and patch defectives from the line.

Worker #3

Worker #3 patches those cans directed to her by the spotter. When not occupied by patching tasks, this worker scans the line for flange and cross-pack defects and patches those defects as found.

Worker #4

Worker #4 patches underweight cans ejected by the checkweigher. When not occupied by patching tasks, this worker scans the line for flange and cross-pack defects and patches those defects as found.

Additional Workers

Additional workers may be added as required depending on product quality (grade) and fish quality. Poor quality salmon may require additional workers to maintain acceptable quality (grade). These workers alleviate extra load on the other workers.

3.3.2 Flange Defects

Patching Method

It was found that the patching of flange defects occurs in two modes; 'on-the-fly' and 'off-line'.

On-the-fly patches are very quick patches of flange obstructions which do not involve the removal

of the can from the line. Usually the worker obstructs the flow of the can with one hand while simultaneously tucking in the obstruction. These patches are, of course, only carried out when the flange obstruction is the only defect. Off-line patching involves the removal of the can from the line and is usually a longer patch operation. Generally a stuffing or tucking method of patching is employed using fingers or scissors.

Distribution and Patch Times

The percentage of flange obstructions observed in a 599 can sample was 14.69%. As mentioned above, the patching of flange defects occurs in two modes, on-the-fly and off-line. The two modes were separated by analysis of patch times required for on-the-fly and off-line flange patching. In order to determine the split between those flange defects which can be patched by the two methods, worker #4 (the first worker to look at the cans) was analysed to determine the number of off-line vs. on-the-fly patches that she performed. This analysis showed that about 50% of the defects were patched by each method (Table 3.4). The on-the-fly type patch requires no dedicated patching time in the simulation of the current workcell, since this patch does not interfere with the workers scanning the line (i.e., they perform both tasks simultaneously). On the other hand, a patch time distribution is necessary for the off-line type patching mode.

<i>Mode</i>	μ	σ	<i>Distribution</i>	<i>Type</i>
On-the-Fly	N/A	N/A	51.1%	None
Off-Line	0.6767	0.6137	48.9%	lognormal

Table 3.4: Flange patch time distribution fit.

A log transform of the sample data generated a normal distribution indicating a lognormal distribution was probably the underlying general distribution for off-line patching. “The lognormal distribution is the distribution of a random variable whose natural logarithm follows the normal distribution” [18]. Hence, a lognormal distribution:

$$f(x) = \frac{1}{\sigma x \sqrt{2\pi}} \exp \left[-\frac{1}{2} \left(\frac{\ln(x) - \mu}{\sigma} \right)^2 \right] \quad (3.1)$$

was fit to the off-line patching time data by minimizing the square of the error, see Figure 3.5. Table 3.4 gives the optimized lognormal distribution fit. It should be mentioned that other distributions including gamma and Weibull were fit to the data but with less satisfactory results and hence the lognormal distribution was selected. This and all distributions were tested according to the Chi-Squared goodness of fit test with a significance level of 0.05 except in a few special cases where the amount of available data rendered such testing impossible. In all cases considered it was not possible to reject the hypothesis of a distribution fit at this level of significance. These tests are compiled in Appendix A. Since the distributions are lognormal (in most cases) the patch times require taking the exponential of the indicated patch time in order to generate the real patching time.

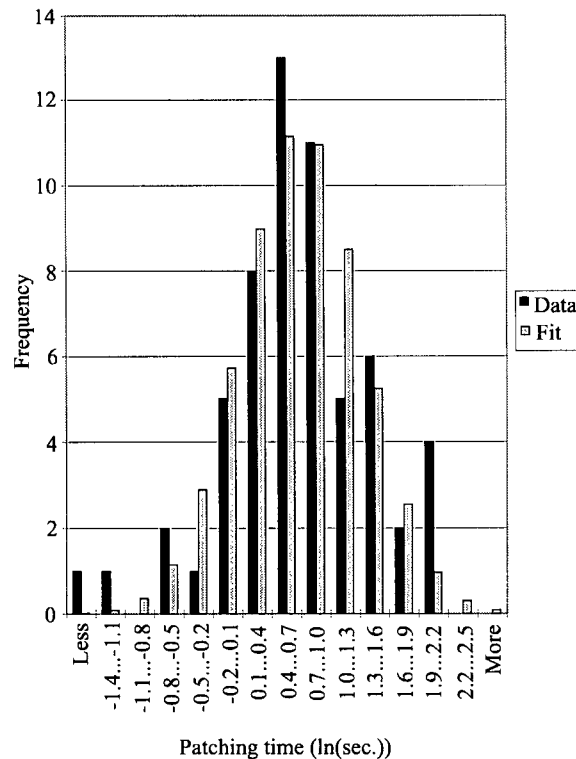


Figure 3.5: Off-line flange patch times.

3.3.3 Cross-packs

Patching Method

The patching of cross-packs is generally longer than flange patching. It almost always involves the use of scissors and, in some cases, removal of the steak and subsequent re-packing. This is a high dexterity task since significant handling of the steak is usually required. Thus, this defect would be costly to automate, requiring some high level intelligence and dextrous manipulation.

Cross-pack Qualification

Due to the large variation in the patching of cross-packs, even within a specific grade and the variability in what 'experts' consider to be a cross-pack defect requiring patching, the up-stream and down-stream data sets were used to determine the allowable level of cross-pack. The data sets were examined for cross-pack and offending cans were placed into groups based on the surface area of exposed skin. Table 3.5 shows the distribution of the cross-packs entering and exiting the system. Examining the tabulated data shows a significant differential between entering and exiting cross-packed cans only for cross-packs of one eighth the surface area or greater.³ Utilizing this as a hard threshold for cross-pack qualification the percentage of cans entering the table 'cross-packed' is 7.8%.

<i>cross-pack area</i>	$< 1/12$	$< 1/8$	$< 1/6$	$< 1/4$	$< 1/3$	$> 1/3$
<i>Entering (%)</i>	14.86	5.68	4.34	2.67	0.835	0.0
<i>Exiting (%)</i>	13.40	5.36	1.84	1.17	0.168	0.0

Table 3.5: Cross-pack patching data.

Another factor which affects appearance of the cross-pack is distribution of the problem (i.e., if just over a sixth of the can surface has skin visible but it is due to two or three pieces it might not

³This cross-pack area would increase, or decrease, depending on the class of salmon being processed. The data herein is for grade 4 salmon.

be considered average but only minor). It would be valuable upon implementation of automated inspection to incorporate this feature into the problem. An additional factor is the importance of colour. It has been indicated by B.C. Packers that dorsal skin (black) is considered a more severe defect and hence an attempt is made to patch all black skin cross-packs. This however, is not backed up by the data sets shown in Tables 3.6 and 3.7. These factors are considered in [65].

<i>cross-pack area</i>	$< 1/12$	$< 1/8$	$< 1/6$	$< 1/4$	$< 1/3$	$> 1/3$
<i>black skin (#)</i>	36	18	15	6	3	0
<i>white skin (#)</i>	53	16	11	10	2	0

Table 3.6: Incoming cross-pack population, 599 can sample.

<i>cross-pack area</i>	$< 1/12$	$< 1/8$	$< 1/6$	$< 1/4$	$< 1/3$	$> 1/3$
<i>black skin (#)</i>	35	16	7	5	0	0
<i>white skin (#)</i>	45	16	4	2	1	0

Table 3.7: Outgoing cross-pack population, 597 can sample.

Patching Time

Upon examining the distribution of patching times required for fixing cross-packed cans, Figure 3.6, it was hypothesized that again the process was multi-modal due to the obscure occurrence of several very long patch times shown in the “more” bin. Re-examination of video footage confirmed two distinctly different modes of patching. These modes were: a) those requiring cutting and b) those requiring only poking and tucking of the cross-packed skin. The most common patch is type b), whose patching time is distributed about the 2-3 second patch bins. Type (a) patches, appearing in the “more” bin of Figure 3.6 are much longer in duration and less common. In order to model these aspects the two cases have been separately addressed in a similar fashion to the flange defects.

Figure 3.7 shows the lognormal distribution fit to the type (b) cross-pack patch utilizing a mean of 1.56 and a standard deviation of 0.34. It is not possible to accurately model the type (a) cross-pack patch time distribution as only a small sample of this type of patching was available. Hence

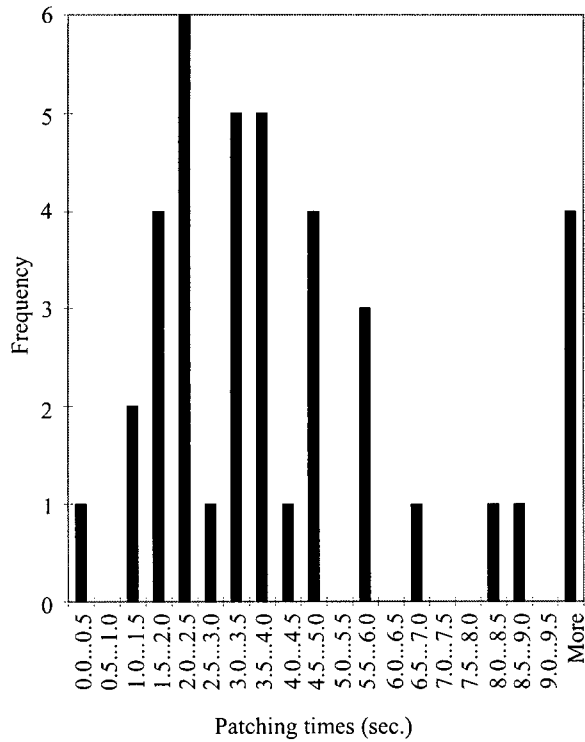


Figure 3.6: Cross-pack patch times.

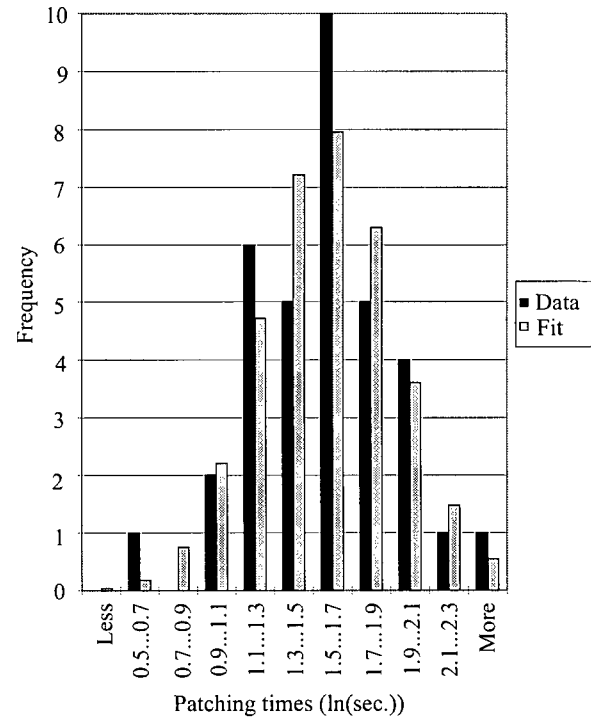


Figure 3.7: Type (b) cross-pack patch time distribution fit.

a normal distribution has been assumed and placed at the mean value of this type of patch, 24.0 seconds with a standard deviation of 7.17. This type of patching represents 10.2% of the cross-pack repairs. The data is tabulated in Table 3.8.

<i>Mode</i>	μ	σ	<i>Distribution</i>	<i>Type</i>
cutting	24.0	7.17	10.2%	normal
no cutting	1.56	0.344	89.8%	lognormal

Table 3.8: Cross-pack patch time distribution fit.

3.3.4 Weight

Patching Method

In general, it was observed that underweight fills were corrected, while overweights were not, representing a loss to the producer. The patching of the underweight cans is usually done by the

worker inserting their finger down the side of the can between the can and the steak and pulling radially towards the center thereby creating a pocket for meat addition. While the pocket is being formed the worker picks up a prepared meat chunk and inserts it into the can and then presses the steak flat once more.

Pre-Patching Weight Distribution

The end of the day weight totals registered by the checkweigher are recorded in Table 3.9. Over fifteen tones of product were processed with a mean weight of 218.6 grams and a standard deviation of 4.95 grams. Figure 3.8 illustrates the distribution graphically. Unfortunately the histogram distribution chosen by B.C. Packers does not represent the upper distribution effectively. Furthermore, the checkweigher distributions are not consistent with the weight distributions computed by QC personnel, hand weighing samples of cans leaving the table, Figure 3.9.

<i>Range</i>	<i>Count</i>	<i>Weight(g)</i>
0-204	963	146623.0
204-210	2851	592440.8
210-214	7479	1587688.0
214-218	17551	3793823.1
218- ∞	40500	8994371.5
		15114946.4

Table 3.9: Pre-patching weight distribution.

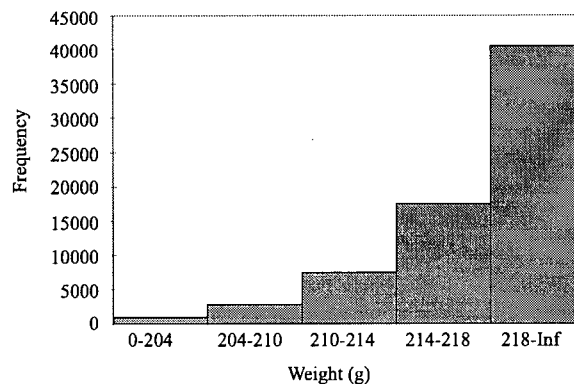


Figure 3.8: Checkweigher totals.

An independent 565 can sample was compiled and the results are shown in Figure 3.10. The mean weight was found to be 216.9 grams with a standard deviation of 5.07 grams. Geary's test for normality indicates a high degree of confidence in the assumption of normality for this distribution (see AppendixA). To check this result, a second sample was generated by observing the number of ejected cans on the live video feed. Six samples were taken over the course of the experiments, 14523 cans. For each sample the mean of the distribution was fit such that the tail area below 204 grams would generate the observed number of ejections for the sample time. The mean weight was 216.9 grams (the same as from the 565 can sample recorded from the checkweigher) and a sample standard deviation of the mean between the six samples was 0.733. Hence, a normal distribution utilizing the sample mean and standard deviation have been adopted for use in the simulations. The data used in the both the 565 can sample and the video sample were collected during controlled times in the shift while Pink salmon was being processed with normal operating conditions (i.e., no filler anomalies). Recalling that B.C. Packers desires the weight of 1/2 lb. cans to be controlled to within a 2 gram standard deviation, the present filling machine produces a product with 2.5 times the desired standard deviation.

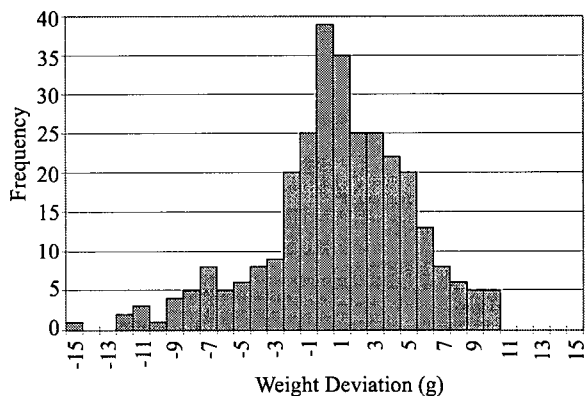


Figure 3.9: Post-patching weight control.

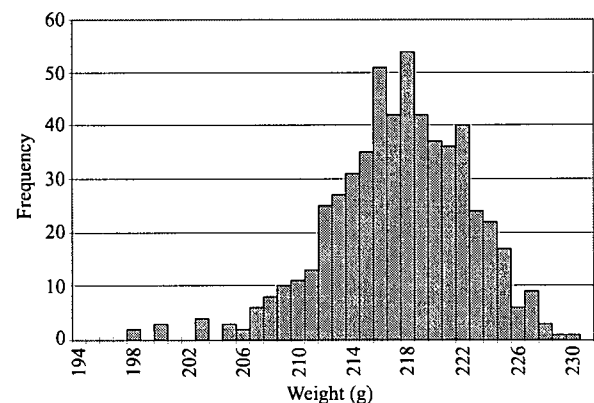


Figure 3.10: Checkweights, 565 can sample.

Patching Time

The distribution of the weight patching times was also found to be multi-modal. In this case the two modes stem from the fact that the task of weight addition requires preparation of the salmon for adding to the cans. This preparation allows some weight patches to be rapidly completed when there is prepared salmon ready, while other patches take much longer when no prepared salmon is ready. Two possible ways of addressing this situation were considered.

- I. the use of a Monte Carlo type simulation of the distribution.
- II. a distribution fit to the 'prepared' weight patch data plus an additional action or function required by weight patchers of 'preparation'.

The second approach was selected since the patchers do some preparation even when not in the act of patching. Therefore the weight patching task requires modelling of this function in both cases. The preparation task is a function of the number of weight patches that the worker has processed. Figure 3.11 shows the lognormal distribution fit to the weight patch times not involving preparation of meat. The mean and standard deviation used in the fit are 1.97 and 0.732 seconds respectively.

3.3.5 Poor-fills

Patching Method

It was observed that worker 2 repaired defects which were not easily classifiable. This category of defect includes very poor colour pieces, pieces requiring spine or rib cage removal and other difficult patches. The patching of the poor-fill cans is complex in nature and varies from case to case. The repair will usually involve the removal of 'poor' pieces and subsequent addition of better pieces

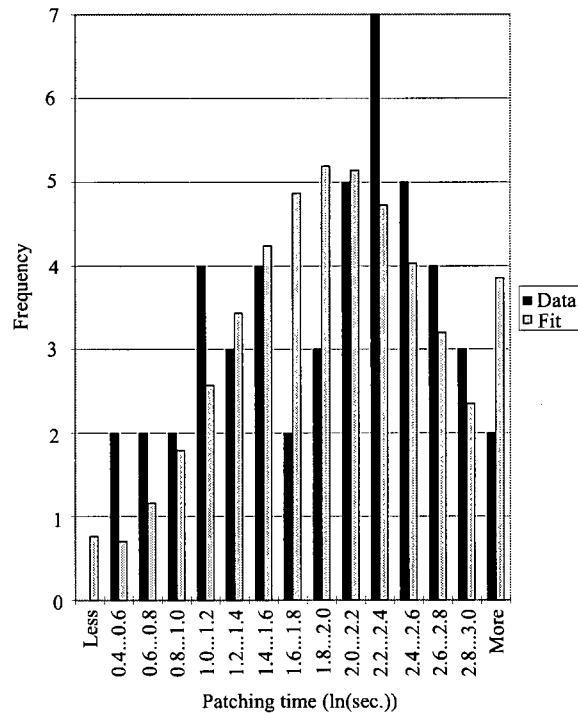


Figure 3.11: Weight patch time distribution fit.

on hand. Hence both worker 2 and 4 require significant meat preparation activities; worker 2 for poor-fill and weight patches and worker 4 for weight patches.

Distribution

The 'poor-fill' category is a highly subjective classification in that expert knowledge would be required to classify fills into this category. Since expert graders were not available for this task, the 599 can sample was graded based on criteria described in the quality control survey. It was found that 7.85% of the incoming cans were poor-fills.

Patching Times

The distribution of the patching times for poor-fill repairs along with the lognormal distribution fit is shown in figure 3.12. The best fit resulted in a mean of 3.49 and a standard deviation of 0.300.

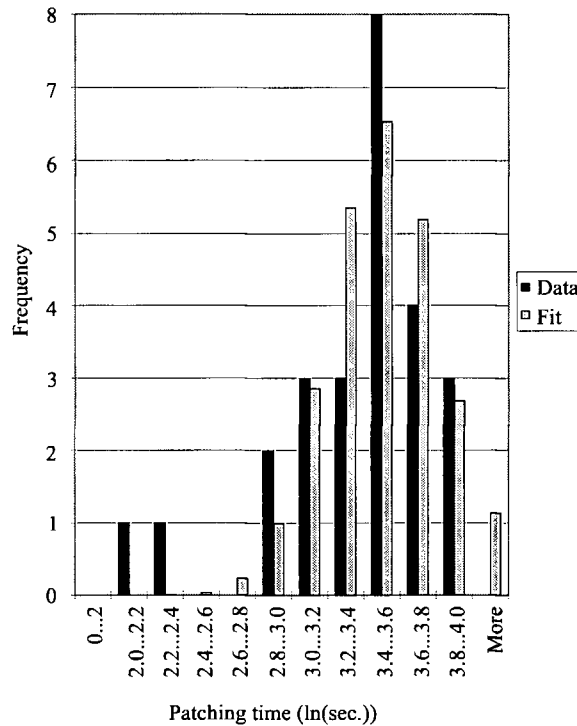


Figure 3.12: Poor-fill patch time distribution fit.

3.3.6 Meat Preparation

In the modelling of the meat preparation task it would be ideal to model the actual level of meat that the patch worker has on hand and its status; 'prepared' or 'unprepared'. This level of modelling however is not feasible since there are many processes which occur to this meat stock including can dumping and throw away. Hence to model the preparation task, video data was used to compile the total time spent on preparation activities for workers 2 and 4. Worker #2 was observed to spend 5.6% of the time engaged in preparation work and worker #4 was observed to spend 13% of the time engaged in preparation work. The task times for the meat preparation activity did not follow any standard distribution and hence the task times have been simulated using Monte Carlo simulations. The number of occurrences of meat preparation was assumed to be a function of the number of underweight and poor-fill patches. Hence the triggering of a preparation activity was tuned such that the simulation provided a similar percentage of preparation time as the experimental data.

The results of this tuning are discussed in Section 4.1.2.

3.3.7 General Observations

A previously unknown type of patch was recognized to take place at the table. Tail pieces are often flipped if their tapered end is facing up. This patch can be thought of as cross-pack in that skin showing is the primary concern. In addition many tail pieces are cut open axially. This operation was done by worker 2 and is included in the poor-fill defect category.

One extra function the workers (mainly the spotter) provide is the general tamping down of the steak using their fingers. This tamping presses the surface of the steak below the flange to allow proper lidding. This process is done on-line: therefore, no task time is necessary. However, any automation introduced for defect patching such as flange defects should include some mechanism for tamping of the steak surface. This simple addition would remove an additional worker activity and thus reduce work load and improve quality.

Chapter 4

Basecase Simulation

This chapter details the simulation of the patching table workcell based on the experimental data collected and described in Chapter 3. This simulation is referred to as the ‘basecase’. The simulated workcell utilizes a high speed checkweigher and four patching workers including the spotter. The basecase simulation is used in the chapters which follow to analyze the effects of the introduction of various automation aspects into the workcell.

4.1 Basecase

The Awesim simulation package [18] was chosen as the simulation platform for the basecase and the subsequent comparative cases. This package offers a network based modelling environment with discrete and continuous modelling capabilities. It allows the incorporation of user written Visual C++ or Visual Basic functions when appropriate modelling nodes are not available. Figure 4.1 shows the major nodes used in the simulation networks, and Table 4.1 details their functionality.

The basecase simulation mimics the operation of the patching table observed during the data collection experiments in Prince Rupert. Assumptions include 100% accuracy of the checkweigher

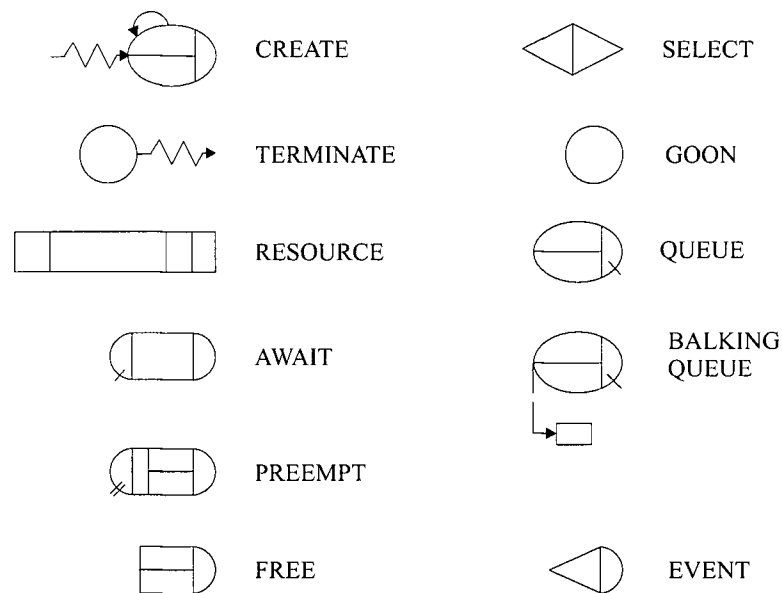


Figure 4.1: Selected Awesim network nodes.

and 100% patching accuracy by the patching workers (i.e., when a worker does recognize a defect, it is correctly patched). For grade 4 salmon fill (the grade of fill running during the data collection at Prince Rupert) the input defect distribution is shown in Table 4.2, based on the analysis of a 599 can sample spread over two 10 minute intervals. Two separate samples being used in order to account for, through averaging, the defect density variation with time.

4.1.1 Logic and Layout

Figure 4.2 shows a simplified version of the basecase network modelled in Awesim, reduced for clarity. The full network employed in the Awesim simulation is shown in Appendix C.

The underlying flow logic for the simulation network in Figure 4.2 is described below. Each node described below is numbered, e.g., ① in Figure 4.2 and paths are lettered, e.g., (x). The specially developed C routines called by the network are referenced in the form (codename.c,call #) and are listed in Appendix D.

Entities (filled salmon cans), enter the network via the CREATE node, ①, at a rate dictated

<i>Node</i>	<i>Function</i>
CREATE	-used to introduce entities (cans) into the system at specified intervals.
TERMINATE	-used to remove entities from the system.
RESOURCE*	-specifies the number of resource units initially available and from which files (AWAIT and PREEMPT nodes) it is to delegate units of resource.
AWAIT	-node is tied to a specific resource and stores entities until resource units are freed such that the entity may be processed by the resource.
PREEMPT	-node is tied to a specific resource. When an entity arrives at this node it immediately seizes a specified number of units of a resource when they become available regardless of whether or not other entities elsewhere have been waiting for that resource.
FREE	-used to release a specified number of units of a resource so that they can again be utilized.
SELECT	-determines which of multiple possible queue nodes to select entities from and which of multiple activities or paths to send them on to.
GOON	-the goon node or the go on node simply passes entities through according to any outgoing conditions emanating from it.
QUEUE	-stores entities until the outgoing activity (generally) has been completed.
BALKING QUEUE	-node is identical to the standard QUEUE node except that when the queue reaches capacity subsequent arriving entities are routed to the node specified by the balk command.
EVENT	-used to call a user written function when an entity arrives.

*The resource block is not a 'true' node since it is not directly attached to the network.

Table 4.1: Function of the major network nodes employed in simulations.

by the desired line speed. Unless otherwise stated the line speed for all scenarios was set to the filling speed observed at Prince Rupert, 232 cans/min.

The EVENT node (2), (BEVENT.c,1) assigns defects to the cans based on the defect distributions formulated in Chapter 3.

Those entities which are below one weight tolerance, (i.e., 9 grams under label weight), are routed down the path labeled "Underweight", (X). This represents the action of the checkweighing machine employed on the patching table during the experiments. It was observed that 90% of

<i>Defect</i>	<i>(%)</i>
Underweight	0.55*
Flange Obstruction	14.69
Cross-Pack	7.845
Poor-Fill	7.85

*determined from checkweigher data

Table 4.2: Simulation input.

the time worker #4 will patch these defects (and any other defect present in the fill) leaving the remaining 10% to be patched by worker #2. This split is represented by the GOON node, (3), following the underweight path which routes 90% to worker 4 and the remainder to worker 2. Cans that are routed to worker 4 are sent via GOON node, (4), to worker 4's patching queue, AWAIT node, (5). At the same time 'dummy entities' are created at GOON node, (4), to initiate the worker meat preparation tasks. These entities are created at a rate of 0.628 times the arrival rate of underweight cans, to allow for a sufficient amount of meat preparation to occur.

The split indicated for generation of a meat preparation task was determined using the simulation and is discussed in section 4.1.2. When one of these dummy entities enters the PREEMPT node, (6), the worker immediately performs a meat preparation task. This activity is preemptive since patching activities cannot continue until prepared meat is available. The dummy cans are subsequently destroyed after initiating the task. The meat preparation times are generated from Monte Carlo simulations of the observed preparation times for workers 2 and 4. These are called by (BUSERF.c,2) for worker 2 and (BUSERF.c,3) for worker 4.

Cans which are not underweight enter GOON node, (7). If worker 4 is not occupied with either meat preparation or patching the can enters the scanning EVENT node, (8). If however, worker 4 is occupied the can goes by un-inspected to the next available worker.

The scanning EVENT node, (8), calls (BEVENT.c,2) which emulates the scanning of the line by the worker for defective cans. The code gives probabilities for that worker correctly identifying

a defective can.

When a defective can is identified it is routed to the patching queue, AWAIT node, (5). The duration of the patching time is assigned by (BUSERF.c,1) and is governed by the statistical distributions outlined in Chapter 3.

The worker has been modelled as a single unit resource which is utilized by various activities. Hence, when the worker is engaged in patching or meat preparation the unit resource is occupied for the duration of the task. After the activity is completed the resource (worker) is 'freed' by FREE node, (9), (for worker 4) and automatically takes on the default activity of scanning for defects. Thus, when a resource (worker) is occupied by non-scanning tasks the entities (cans) are not inspected by the worker. Instead, they travel to the next worker, as is the case in the actual workcell.

The other workers are modelled in much the same manner as worker 4 with the exception of worker 1, the spotter. The spotter is charged with the task of observing every can exiting the system and thus is never required to perform 'complicated' repair tasks. However, the spotter can perform simple 'tamping' operations on the cans as they pass by to correct any minor flange defects. (The nodes which represent this operation are not shown in Figure 4.2, but are shown in the full system diagram given in Appendix C). The spotter routes any detected poor-fills to worker 2 and all other detected defects to worker 3 for patching. This is shown in the network by paths (Y) and (Z) leading from the spotter directly into the AWAIT nodes, (10) and (11), of worker 3 and 2, respectively.

4.1.2 Tuning and Evaluation

In order to tune the system so that its behaviour reproduced the observed operation of the patching table, parameters were selected to indicate how closely the simulation represented the observed data.

The parameters selected were:

- the time utilization (time working/total working time) of each worker
- the patching rate (patches/unit time) of each worker
- the outgoing defect distribution (relative levels of the various defects)
- the table failure rate

In addition the rate at which the spotter passed cans to workers 2 and 3 was matched with the observed experimental data. Successful matching of these parameters would indicate that the models of the worker behaviors, specifically the worker task breakdowns, task efficiencies and task times were appropriate.

The variables which were adjusted to tune the system were: the efficiencies of workers in spotting each of the three defect categories; flange, poor-fill and cross-pack. The values of the tuned parameters appear in Table 4.3. In addition, the experimental data indicated how much time workers 2 and 4 spent engaged in meat preparation. Therefore, tuning was done in order to determine the ratio of meat preparation task-time to underweight task-time and to poor-fill task-time. For example, it was determined from the experimental data that worker 4 spends about 13.0% of their time engaged in meat preparation. Based on the task times for worker 4, for every 1.59 underweight patches that worker 4 completes, one meat preparation task should be initiated. The spotter was assigned an effectiveness level for patching 'on-the-fly' type flange defects. This was tuned to 54%, i.e., the spotter fixes 54% of the 'on-the-fly' flange defects.

	<i>Worker</i>			
	4	3	2	S
η_{xp}	0.405	0.02	0.105	0.107
η_{fd}	0.275	0.13	0.0	0.46
η_{pf}	N/A	N/A	0.0	0.068

- η =inspection efficiency for task

- xp =cross-pack

- fd =flange defect

- pf =poor-fill

-S=spotter

Table 4.3: Tuned efficiency parameters.

Tuning produced the results shown in Tables 4.4 and 4.5. Satisfactory results were achieved through tuning with the exception of the patching rate of the table. Table 4.5 shows the outgoing defect rate of the simulation and the data differing by about 4 percent. This appears to indicate a problem in the simulation. However, closer examination of the data indicated the discrepancy is related to the experimental data. The measured overall patching rate of the table is the sum of the patching rates of the individual workers. From Table 4.4 this is $(6.26 + 8.94 + 2.0) = 17.2$ cans/min or 7.41% for 232 cans per minute. However, the change in the number of defects in the upstream and downstream sample data indicates a patch level of $(30.94\% \text{ in} - 13.23\% \text{ out}) = 17.71\%$. The ‘true’ patching rate is likely somewhere between the two numbers. The outgoing and incoming defect data were each from two samples representing only 597 cans. In both cases, these sample sets differed by more than a percent. Furthermore, due to hardware limitations the upstream and downstream data sets were collected at different times in the shift. On the other hand, the data concerning the individual worker patching rates and spotter activities were compiled from a much richer data set and hence a higher level of confidence. However, computing the overall patching rate from this data is problematic since ‘on-the-fly’ patching done by the spotter are not included. For these reasons the patching rate was not used in tuning. Instead, individual patching rates and the other parameters were used.

In addition other variables were examined for their deviation from the experimental data such as queue sizes and worker task breakdowns. In all cases the match was very close to the data.

	<i>Data</i>	<i>Simulation</i>
<i>Patching Utilization, W4</i>	53.3%	53.2%
<i>Patching Utilization, W3</i>	86.2%	86.1%
<i>Patching Utilization, W2</i>	77.4%	77.1%
<i>Preparation Utilization, W4</i>	13.0%	13.2%
<i>Preparation Utilization, W2</i>	5.8%	5.9%
<i>Patching Rate (cans/min.), W4</i>	6.26	6.20
<i>Patching Rate (cans/min.), W3</i>	8.94	8.89
<i>Patching Rate (cans/min.), W2</i>	2.0	1.62
<i>Spotter passes to W3*</i>	3.6%	3.6%
<i>Spotter passes to W2*</i>	0.53%	0.53%

* percentage of all cans

Table 4.4: Simulation output.

	<i>Data</i>	<i>Simulation</i>
<i>Defectives</i>	13.23%	17.63%
<i>Single Defects</i>		
<i>Flange</i>	4.36%	5.50 %
<i>Cross-pack</i>	3.18%	4.81%
<i>Poor-fill</i>	5.19%	5.97%
<i>Multiple Defects</i>		
<i>pf + xp</i>	0.0%	0.452%
<i>xp + fd</i>	0.17%	0.365%
<i>fd + pf</i>	0.17%	0.495%
<i>fd + pf + xp</i>	0.0%	0.0351%

pf = poor-fill

xp = cross-pack

fd = flange defect

Table 4.5: Output comparison.

4.1.3 Results

The simulation time was increased until the deviation of the tuning parameters in successive runs was less than 0.05%. It was found that a simulated time of 277.8 hours or one million seconds provided the desired level of consistency. Tables 4.4 and 4.5 show the parameters matched and the

outgoing defect distributions. This basecase output will be used in evaluating various automation options based on a comparison of their results to this output. Although the basecase is based on a rather small data set (one shift on one patching table in one plant) the automation option analysis which is carried forward from this basecase is relative in nature. In other words, if improvements can be shown over this basecase, it can be expected that similar improvements could be made for more general patching table scenarios.

Chapter 5

Comparative Methodology

In this chapter, the ways in which automation of the patching table could result in economic and or quality benefits are discussed. The methods by which these benefits are quantized and evaluated are detailed. The quantization of these benefits is then used to evaluate and compare a number of options for automation of the patching table, based on the results from the basecase patching table simulation.

5.1 Operational Parameters

Several important operational parameters of the canning line are tabulated in Table 5.1. It is assumed that a typical canning line will process a percentage of each of the various salmon species according to the data in Figure 5.1 (provided by the B.C. Salmon Marketing Council database)¹. Thus, the average price per pound of raw material is determined from the data appearing in Figures 5.1 and 5.2 (also from the same database).

¹www.bcsalmon.ca

The remainder of the operational parameters are based on data provided by various personnel at B.C. Packers Ltd.

Fish (raw materials)	\$1.13/Lb
Operating hours	640/season
Production hours	560/season
hours/shift	8 hrs
Line-rate	240 can/min.
Patching table labour Cost	\$21/hr/worker
Inspection labour cost	\$25/hr/inspector
Hand inspection rate	3 cans/min.
Batch size	19020 cans
Weight tolerance	9 grams (1/2lb)
Label weight	213 grams (1/2lb)
Filler Operational Time	95%

Table 5.1: Operational parameters.

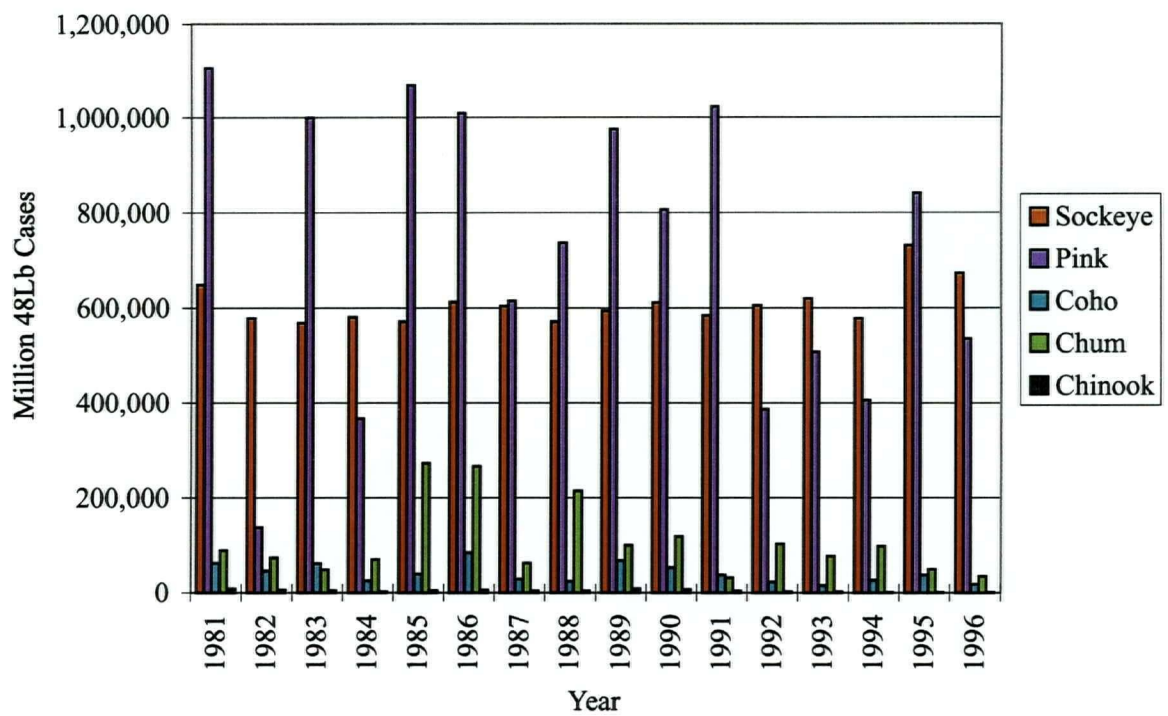


Figure 5.1: Canadian salmon canning statistics

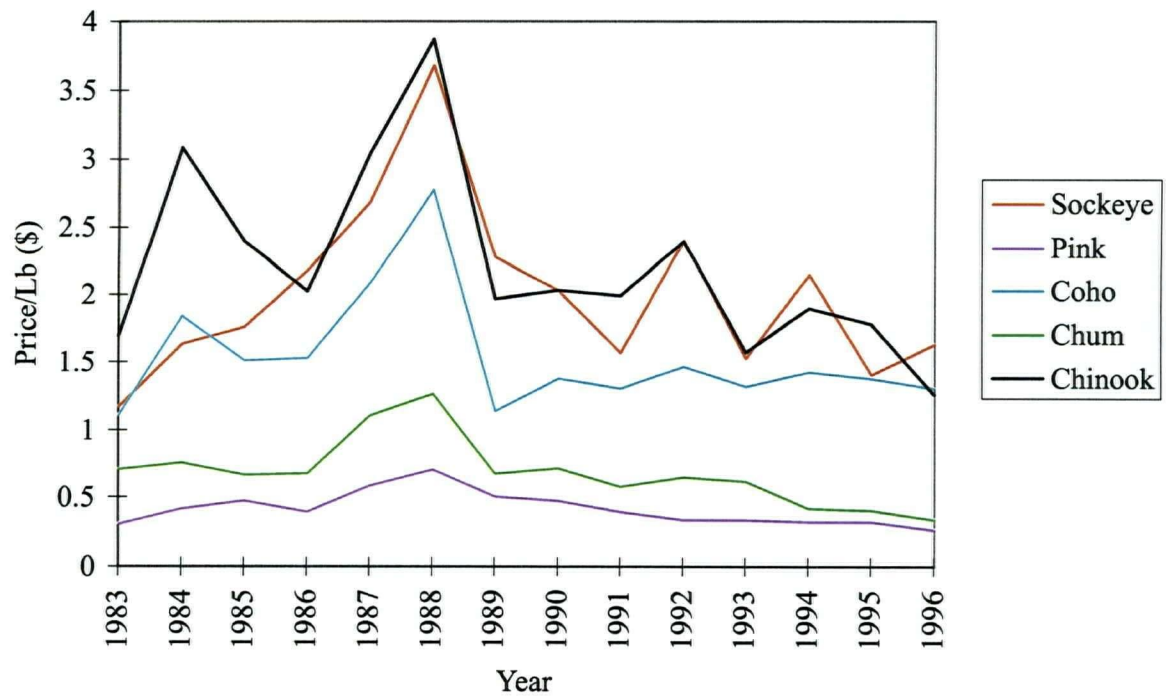


Figure 5.2: Canadian wholesale salmon prices

5.2 Monetary Savings

5.2.1 Direct Labour Savings (DLS)

Direct labour savings are realized when the utilization of the patching table workers is reduced such that one or more workers can be eliminated from the table. Based on the worker utilization observed in the basecase, the mean worker utilization is 80.6%. That is, the average worker is physically handling the product about eighty percent of the time. If simulation shows that through the implementation of automation 80.6% or more worker utilization is 'freed' then a worker can be saved. In this manner, on average, no extra work will be required of the remaining workers. Based on the data in Table 5.1 the cost (wages plus benefits) of a single patching table worker working the entire season at regular time pay rate is about \$13,440, yielding the DLS for the replacement of one worker.

5.2.2 Indirect Labour Savings (ILS)

Indirect labour savings are generated when the percentage of flange defects leaving the patching table is reduced. In short, a percentage of flange defects which leave the patching table can result in seam defects when the cans are lidded and sealed. The sealed cans are automatically inspected using checkweighers and double dud detectors on the labeling line, prior to labeling. (Further details about this inspection process are given in Appendix E). If the number of serious seam defects in a batch of cans (cans produced in a single/shift by a single line) is above a government specified level, namely, 25/100,000, then the cans must be hand inspected (culled). Hand culling is very expensive since it is labour intensive. Therefore, a reduction in flange defects, leads to a reduction in seam defects, and thus to a reduction in the probability of a hand cull being required. This reduction in hand culls results in the Indirect Labour Savings.

In order to determine the ILS derived from reductions in flange defects an estimate of the mean industry cull rate is required. Based on industry data 2.47% is used in this analysis. Of the serious seam defects, 47.3% can be attributed to flange defects [66]. Thus a cull rate of 1.17% attributable to flange defects is estimated. A linear relation between outgoing flange defects and cull rate is assumed.

The busiest canning lines, where automation would most likely be applied operate about 60 days a season. Twenty days of the season are double shift days. Thus, the filler at 95% capacity yields 7,660,800 cans or 402.8 batches per line per season based on the data in Table 5.1. Consultation with industry personnel indicates that the time required for a worker to hand inspect a can ranges from 1-5 cans per minute and a mean of 3 cans per minute has been assumed. Based on an average of 19020 cans per batch and an inspection labour cost of \$25/hr the cost of a hand cull is \$2,642. Thus the expected cost to the producer for hand culls per season is the number of batches, times the cost per cull, times the probability of a cull, $(402.8 \times 2,642 \times 0.0117) = \$12,451.1/\text{season/table}$. A new probable cost can be calculated for a reduced value of outgoing flange defects. The reduction in this expected cost of culling is the ILS.

5.2.3 Raw Material Savings

In order to comply with government regulations regarding average and minimum fill weights, industrial producers overfill the cans resulting in a mean fill weight greater than the label weight. Hence, by shifting the mean fill weight downwards, closer to the label weight, raw material savings can be realized. It is required that the mean fill weight of the cans be equal to or greater than the label weight. Hence the best improvement possible through automated weight correction would be to set the mean fill weight to the label weight and correct those which fall one tolerance below that weight. A cost function of the form in Equation 5.1 can be defined. The distribution of the

fill weight is normal, Equation 5.2. Hence the expected savings per can is given by Equation 5.3. However, the distribution is cut at one tolerance below the label weight, since cans falling below this cut-off are patched. It is assumed that these cans are repaired to the label weight and hence do not contribute to savings or losses. Figure 5.3 plots Equation 5.3 using the mean and standard deviation obtained from the experimental data and the average price per gram based on the data in Table 5.1. Total savings per can are determined by integrating this curve, Equation 5.4. The total savings in the basecase are negative since the filler was filling at 216.9 grams (i.e., losing money due to the use of extra raw material). The total raw material savings are found by multiplying the absolute value of Equation 5.4 by the total number of cans processed over the course of the season.

$$f(x) = (x_{lb} - x)C_g \quad (5.1)$$

$$C_g = \text{cost per gram of product(\$)}.$$

$$x_{lb} = \text{label weight}$$

$$g(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \quad (5.2)$$

$$\text{probable savings per can(\$)} = f(x)g(x) = h(x) \quad (5.3)$$

$$\text{savings per can(\$)} = \int_{x_{lb}-tol}^{\infty} h(x)dx^* \quad (5.4)$$

$$(\text{tol} = \text{weight tolerance})$$

$$*\forall x < tol, x = x_{lb} \Rightarrow f(x) = 0$$

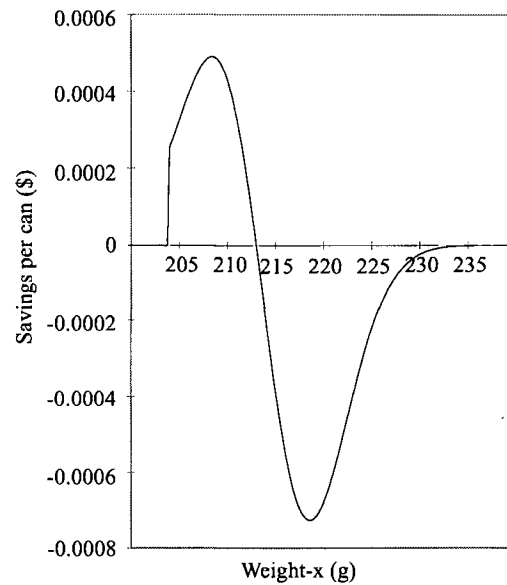


Figure 5.3: Probable savings per can.

5.3 Economic Parameters

The Discounted Cash Flow, (DCF), rate of return is used to economically evaluate possible automation options for the patching table. In this methodology the present worth of all cash flows in and out are set equal to zero. The discount rate which achieves this state is the DCF rate of return. Generally the solution to this problem requires an iterative solution in order to solve for the rate of return. If however, the investment is such that a single capital expenditure (cash outflow) is required at the start of the project followed by a number of years of ensuing cash inflows a closed form solution exists. This is the case in the present study, and the ROI (return on investment) can be computed using Equation 5.5, where R is the ROI and the initial investment is made in year zero and the cash inflows take place uniformly over years 1 to n . In the analyses herein an economic planning horizon of 5 years is assumed, therefore $n=5$.

$$\text{capital recovery factor} = \frac{R(1+R)^n}{(1+R)^n - 1} = \frac{\text{annual cash inflow}}{\text{initial investment}} \quad (5.5)$$

Additionally, the pay-out period for each of the automation options is calculated. The pay-out

period is the capital expenditure divided by the annual savings (or profits) incurred due to the implementation of the automation option.

5.4 Non-Monetary Effects

In addition to economic benefits, improvements in product quality in terms of the reduction of poor-fills and cross-packs leaving the table should be considered, although they do not directly lead to cost savings (i.e., labour or raw materials). These defect levels are, however, important, given their relationship to reputation and customer satisfaction. In addition, it may be possible to produce higher grade product from poorer quality fish if the level of these aesthetic defects is decreased. However, the relative importance of these quality effects is not clearly defined by the industry. Thus, in order to present this data in a compact format two indices have been defined.

A quality index (QI) was defined as the level of poor-fill and cross-pack defects occurring in the basecase divided by those occurring in the automation option being investigated. Thus values below unity indicate a reduction in quality on the basis of aesthetic characteristics while values above unity indicate an improvement in the product quality. Similarly a safety index (SI) has been defined as the percentage of flange defects occurring in the basecase divided by those occurring in the automation option being investigated. Again values below unity suggest a reduction in the safety of the product while values above unity indicate an improvement in the product safety.

Chapter 6

Automation Technologies and Prototypes

Three main functions of the patching table have been examined for automation

- Can inspection, (identifying defective cans and removing them from the line for patching.)
- Weight correction, and
- Flange cleaning/trimming.

Other tasks such as patching cross-pack, or poor-fills, were identified as being too manipulation intensive to be suitable for cost effective automation. In addition, the success rate of the repair is not expected to be sufficiently high in such a complex task. Finally, until well-defined costs or benefits can be associated with changes in the aesthetic appearance of the pack, further analysis of the automation of such tasks is not well motivated.

In each case, research into existing, new (under development) and/or feasible technology was done to determine suitable automation for the task. In this chapter, the results of this research is reported. Then, based on the results, various automation options are considered, simulated, and evaluated in Chapter 7.

6.1 Automated Can Inspection

The task of accessing the fill quality of the pack has been addressed through the use of a machine vision inspection system. A prototype machine vision system is currently under development in the Industrial Automation Laboratory at the University of British Columbia [67, 65]. Figure 6.1 shows the prototype machine vision inspection system under development. The inspection system captures a digital image of each can entering the patching table and through image processing determines if the fill is cross-packed or has a flange defect. Those cans which the system determines to be defective are ejected to waiting queues for manual patching or to automated correction systems. It is desired to evaluate the effect introducing the machine vision inspection system into the patching table workcell, and to investigate performance variations in the vision system.

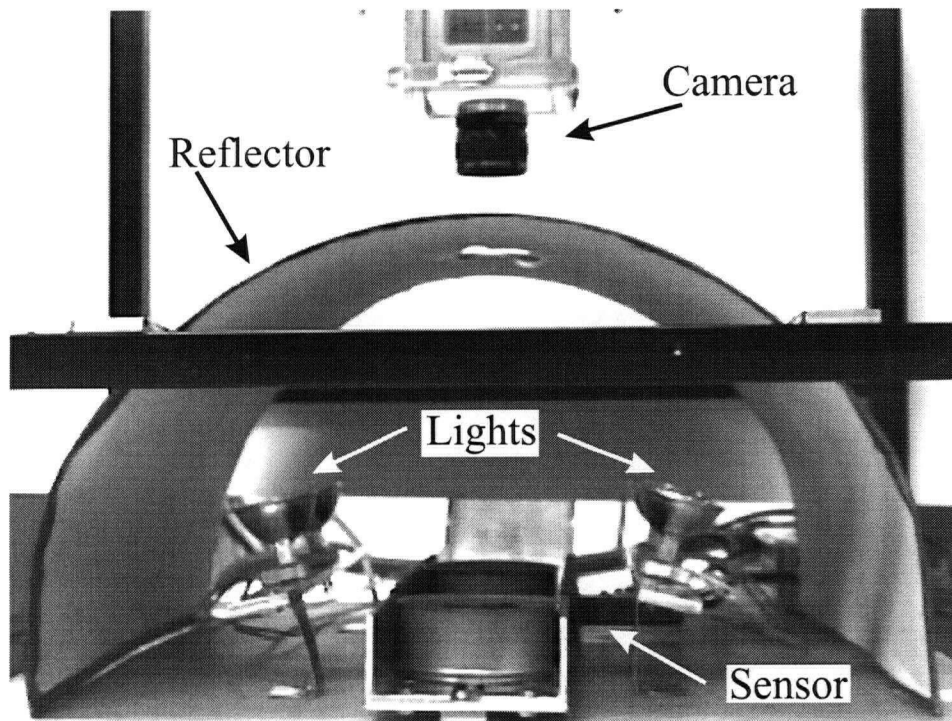


Figure 6.1: Prototype Machine Vision Can Inspection System.

6.2 Automated Weight Correction

6.2.1 Slurry Injection System

The object of the weight correction task is mainly to add additional product to underweight cans in order to bring them up to label weight. A much less frequent task is to reduce the fill in overfilled (overweight) cans. Furthermore, fill accuracy is important since the high raw material cost coupled with high production indicates the possibility of significant savings being realized if the overfilling is reduced. In the early stages of design, a concept for the automation of weight correction was formulated which utilized ground salmon or "slurry". The idea was to correct the weight of the fill by injecting a precisely controlled amount of ground salmon into the fill.

Prior industry experience indicated that deposition of ground product on the top surface of the fill was not acceptable to the consumer[68]. Thus, the idea in the proposed system was to inject the slurry underneath the steak. The method was evaluated experimentally in the Industrial Automation Laboratory using the apparatus shown in Figure 6.2. Ground salmon was loaded into a hypodermic syringe attached to a short section of thin plastic tubing. The results of testing were positive. The delicate salmon meat was easily penetrated by the plastic tubing, and the slurry was deposited evenly on the underside of the steak.

The added weight could be precisely controlled and a prototype could be assembled from relatively inexpensive equipment. In addition, the mechanism could be loaded with raw material which is currently scrap (e.g., tail pieces) increasing savings for the producer.

However, on further review, personnel at B.C. Packers decided that this solution would probably also be unacceptable to the consumer.

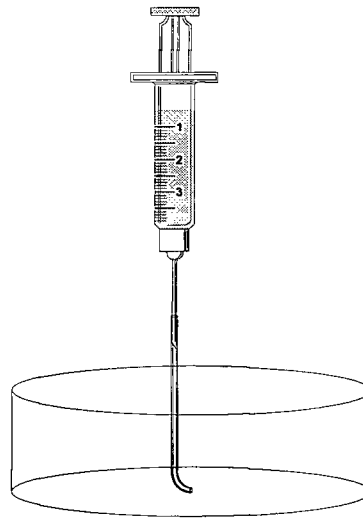


Figure 6.2: Schematic representation of the slurry meat injection concept.

6.2.2 Chunk Injection Systems

A revised design concept utilizes a method of inserting chunks of salmon, much like the patching table workers do, was proposed. This solution is, however, significantly more mechanically complex. A more versatile insertion system would be required. Furthermore, the weight correction would be less precise than for a slurry system. When correcting the weight of the fill the patching workers typically insert their finger between the wall of the can and the side of the steak and then pull radially inward towards the center of the can. Then they insert a piece of prepared salmon of the appropriate weight/size into the pocket created at the side of the can. Figure 6.3 shows two conceptual designs employing the same approach. The first, (A), utilizes a fixed hopper which is loaded with scrap material (salmon tails etc.) and a pocket forming mechanism. The pocket former rotates from a vertical position such that its edge is inserted between the can wall and the steak. The pocket former then translates a couple of centimeters towards the can center. The hopper could be pneumatically driven or through the use of a plunger or possibly even a positive displacement pump. The can would require the use of a fixturing pallet but the accuracy would

not be very high, (e.g. 2mm).

A second concept, (B), utilizes a two axis (planar) manipulator which is fitted with a hollow cylinder. The outside housing of the cylinder is fitted with a spring loaded "finger" which performs the pocket forming operation in a similar manner as the patching workers do. Initially the manipulator cores several salmon steaks using the cylindrical cutter thereby filling the "load chamber". The manipulator then positions the finger at the inside edge of the can, and translates down between the steak side and the can. The manipulator is then moved radially inwards forming the required pocket. At this point air pressure is used to deposit one or more of the cored pieces.

A filler operating with a mean fill of 213 grams with the observed standard deviation (i.e., it is assumed that the standard deviation remains constant regardless of mean fill weight) produces an average of 8.82 cans/min. below 204 grams. Thus a complete task cycle time of less than 6.8 seconds would be required of the system.

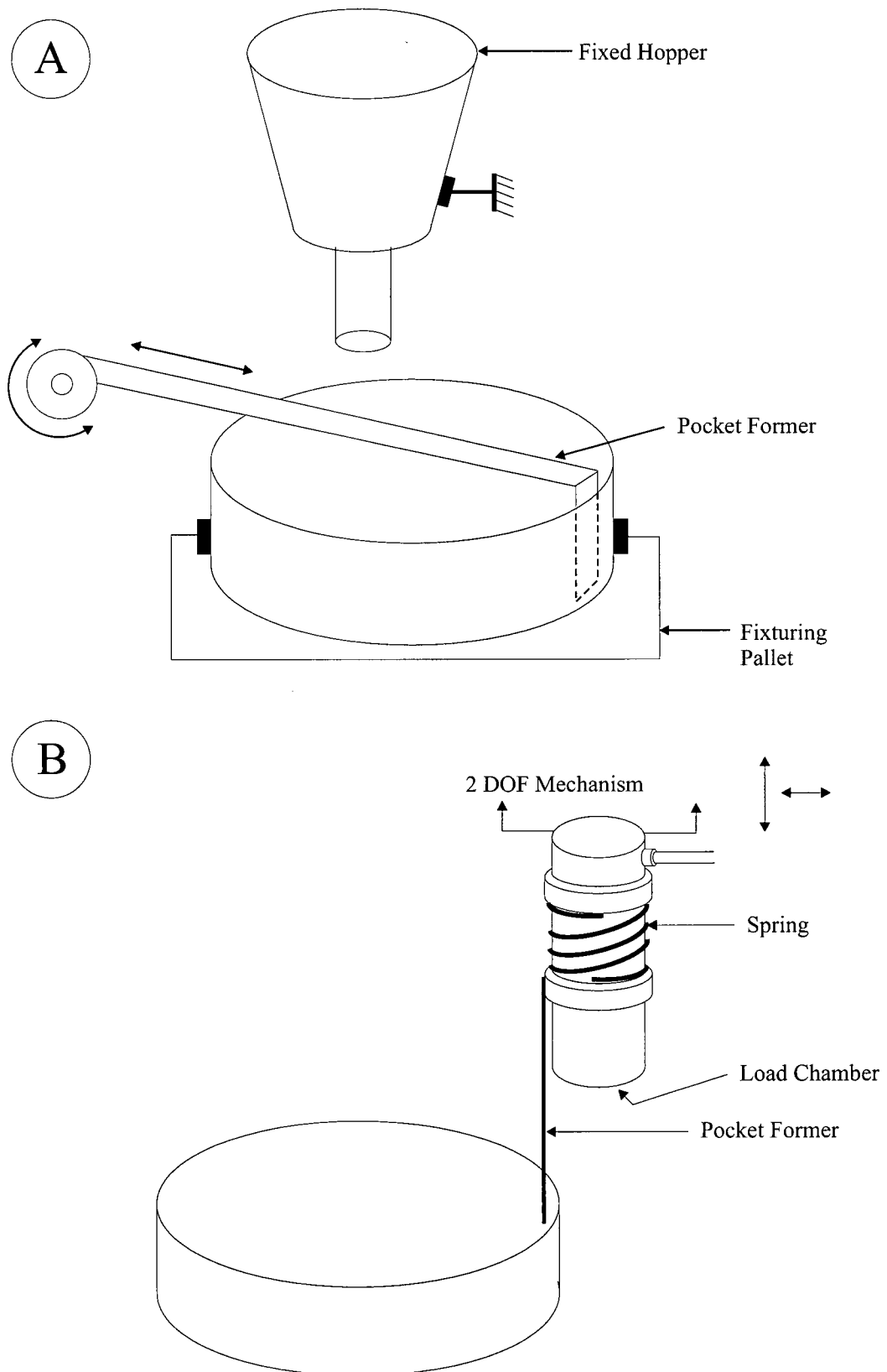


Figure 6.3: Conceptual automated salmon meat addition mechanisms.

6.3 Automated Flange Cleaning

6.3.1 Prototype Trimmer

It was not known that B.C. Packers employs a trimming machine until a full year into the project and hence preliminary designs for a flange cleaning mechanism had been conceptualized, prototyped and tested in the Industrial Automation Laboratory. The first design to be prototyped is shown in Figure 6.4. The design employs an outer holding housing to secure flange defects and a translating rotating cutter. This design was tested in the laboratory on hand packed flange defective cans with encouraging results. However this prototype was abandoned with the advice from B.C. Packers that close metal-can contact could prove potentially dangerous. It was indicated that this close metal-can interface might introduce metal shavings into the salmon steak.

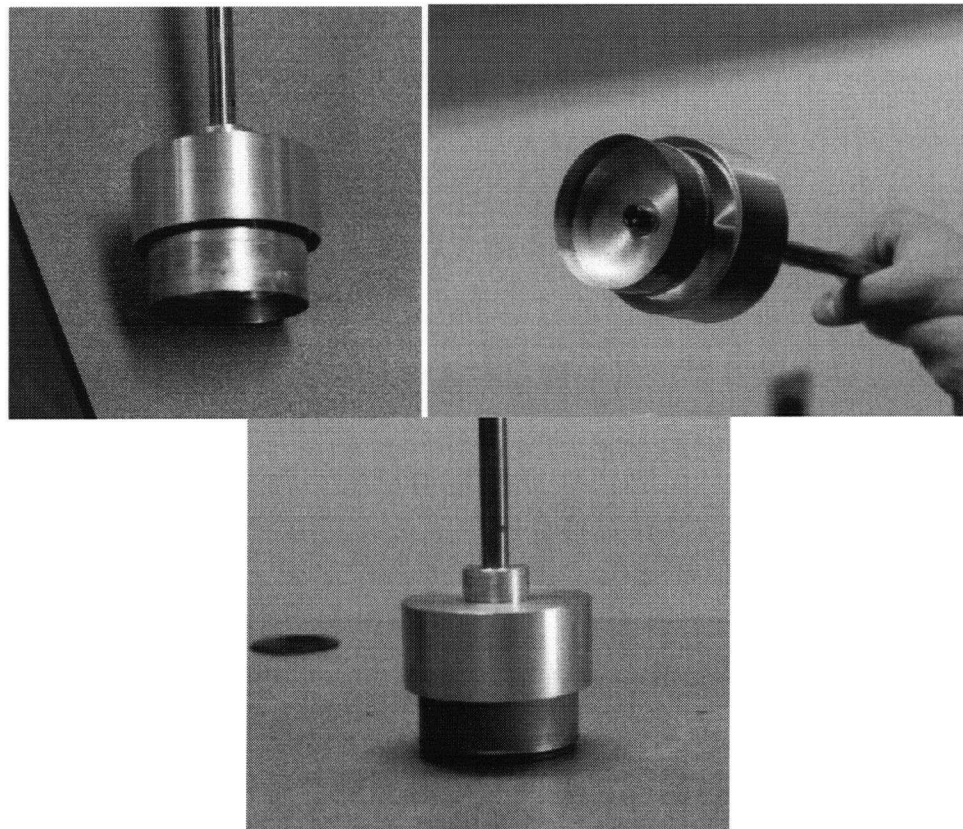


Figure 6.4: Cutting flange defect prototype.

A second design concept was derived from observation of how the patching workers themselves patch these types of defects. It was observed that the patching table workers often do not cut the offending skin or bone but simply stuff it down between the side of the steak and the can wall. Thus a stuffing technique was incorporated into an automatic flange defect correction design. The prototype design is shown in Figure 6.5. The prototype utilizes a PVC housing with the end machined to form a thin wall which fits snugly between the inside can wall and the side of the steak. The holding mechanism is spring loaded to apply pressure to the top surface of the steak thereby holding the steak in the can upon withdrawal. The mechanism pushes offending flange defects down the inside wall of the can in much the same manner as the patching workers do. In addition the holding mechanism serves to provide the “tamping” effect referred to earlier, which is normally done by the spotter on cans leaving the table facilitating proper lidding. The design was tested in the Industrial Automation Laboratory on hand packed flange defective cans and performed extremely well. Upon notification of a new commercial “trimmer” system employed at Prince Rupert further design activities in formalizing the mechanical implementation of this concept were abandoned.

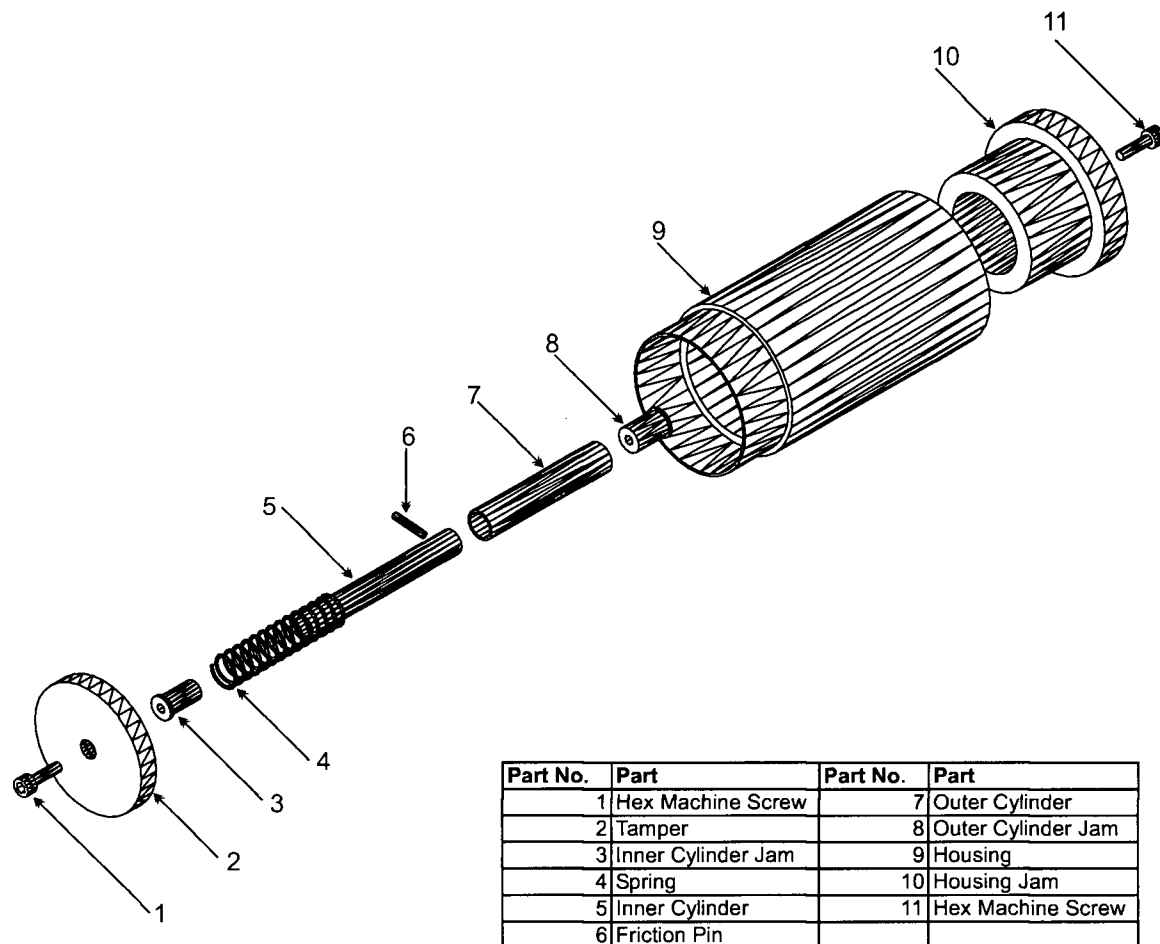


Figure 6.5: Assembly diagram of the stuffing concept automatic flange defect correction unit.

6.3.2 Existing Trimmer Design

The development process of the currently employed trimmer was quite lengthy covering several years and multiple prototypes. It was indicated that a large source of trouble in the design was that cans were being chewed by the cutting blades due to the necessarily high can positioning accuracy required. If in the future B.C. Packers would like to avoid the present arrangement of a lease on an automatic flange cleaning system it is felt that the "stuffer" concept would provide a rapid development schedule with minimal prototyping due to the absence of a cutter assembly. The use of a stuffing mechanism reduces the number of moving parts, eliminates the need for sharpening

and as mentioned has the added benefit of tamping the surface of the steak.

The automated flange trimmer B.C. Packers employs at their cannery in Price Rupert manufactured by Marco Marine is shown in Figure 6.6. This device has been reported to operate satisfactorily however, the cost of the lease has made the cost effectiveness of the device less than satisfactory. Due to proprietary concerns a detailed analysis of the savings generated from the use of the trimmer is not possible, however, as will be demonstrated in the subsequent chapter the installation of the automated inspection system could facilitate a far more cost effective use of the trimmer.

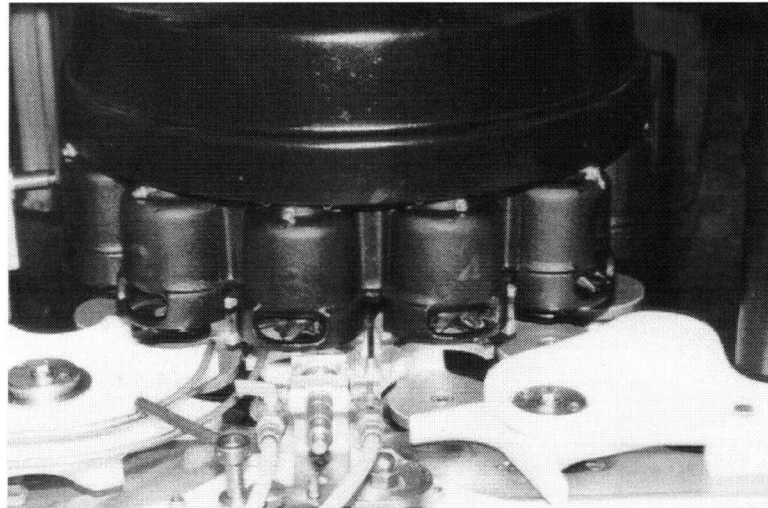


Figure 6.6: Automatic flange cleaning unit employed at B.C. Packers.

Chapter 7

Automation Option Simulation and Evaluation

As described in Chapter 6 three patching table tasks were identified as feasible for automation:

- Can inspection.
- Weight correction.
- Flange cleaning/trimming

This chapter details the simulation and evaluation of a number of automation scenarios incorporating these automatons.

7.1 Automated Inspection (V)

7.1.1 System Assumptions

The vision system is assumed to have three efficiency factors, two for the correct identification of the defects (cross-pack and flange defects) and one for correct identification of non-defects. For the purposes of simulation, an operating point was assumed about which sensitivity analysis is

conducted. The operating point was assumed to be 97% efficiency in recognizing flange and cross-pack defects and a 99% efficiency in recognizing non-defects. To model worker behaviour, results from the basecase simulation were used. In light of the fact that no human can work 100% of the time, the average utilization of the workers was limited to not exceed 95% utilization. The estimated distribution time for a worker to process the various defect types remain unchanged from the basecase with the exception of those in Table 7.1. The sensitivity of all of these parameters were investigated and reported in Section 7.1.5.

<i>Defect</i>	Processing Time (sec.)
<i>on-the-fly flange</i>	2.0
<i>non-defect</i>	1.0

Table 7.1: Task times for workers processing certain cans.

7.1.2 Implementations

Two implementations are considered, vision 1 (V1) and vision 2 (V2) in lieu of the fact that these implementations employ automated product inspection with no other new automation. V1 assumes that the producer would attempt to patch all detected flange defects. The second implementation, V2, assumes that the producer is willing to let some of the minor flange defects through the system under the assumption that these minor defects would likely not result in a seam defect. In case V1, the simulated system routes all detected flange defects, in addition to all of the underweight cans from the checkweigher, to the patch workers for patching. The workers patch all flange and weight defects. The spotter task is eliminated. Any excess worker capacity is then directed to patching of those cross-packs which the vision system identifies, and to scanning the line for poor-fills. Simulation shows that it requires more than three workers to patch all of the weight defects and flange defects.

Intuitively, the sole addition of an automatic inspection system would seem to remove the need

for one worker, namely, the spotter. This is not the case since the spotter performs significant patching of the “on-the-fly” flange defects and the remainder of the workers cannot make up this work. Hence, direct labour savings are not possible for V1. However, the level of worker utilization is, increased since the workers are now engaged in patching duties for a significantly increased period of time, since less scanning is required. The maximum level of utilization is assumed to be 95%. The additional utilization over the basecase utilization is shown in Table 7.2. Hence the total additional utilization is 38.3%.

	basecase task	new task	basecase utilization	new utilization
<i>spotter/worker #1</i>	spotting/tamping	scanning/patching	100	95
<i>worker #2</i>	scanning/patching	scanning/patching	83	95
<i>worker #3</i>	scanning/patching	scanning/patching	86.1	95
<i>worker #4</i>	scanning/patching	scanning/patching	72.6	95

Table 7.2: Changes in the worker utilizations and activities resultant from automated inspection.

In order to optimize the worker utilizations the automated inspection system has been modelled to direct defective fills to defect specific waiting queues, namely: underweight, flange defective, cross-packed, poor-fills and multiple defects. Those fills which are checkweighed as being underfilled are always routed to the underweight queue. Workers repairing underweights repair any additional defects simultaneously. In order to allow the cross-pack patching worker to divide their time evenly between cross-packs and poor-fills proximity sensors are used to monitor the size of the cross-pack patching queue. Then, the vision system adjusts the cross-pack ejection criteria such that this worker is kept busy but not at over capacity allowing sufficient time to scan for and patch some poor-fills. In this way, the worst of the cross-packs are always patched. Tuning of this system would be required in order to achieve a balance between idle time for the worker to engage in scanning of the line for poor-fills and patching of those cross-packs that the vision system ejects.

The network model, Figure 7.1, represents the workers drawing from the waiting queues. Selection priority is first given to the underweight queue and then to the flange queue. When these

two queues are empty the other three queues are selected from in a cyclic manner. In this simulation, tuning has been assumed to result in equal weighting of cross-packs and poor-fill defects. In implementation it would be up to the producer to decide the actual weighting for patching of these defects.

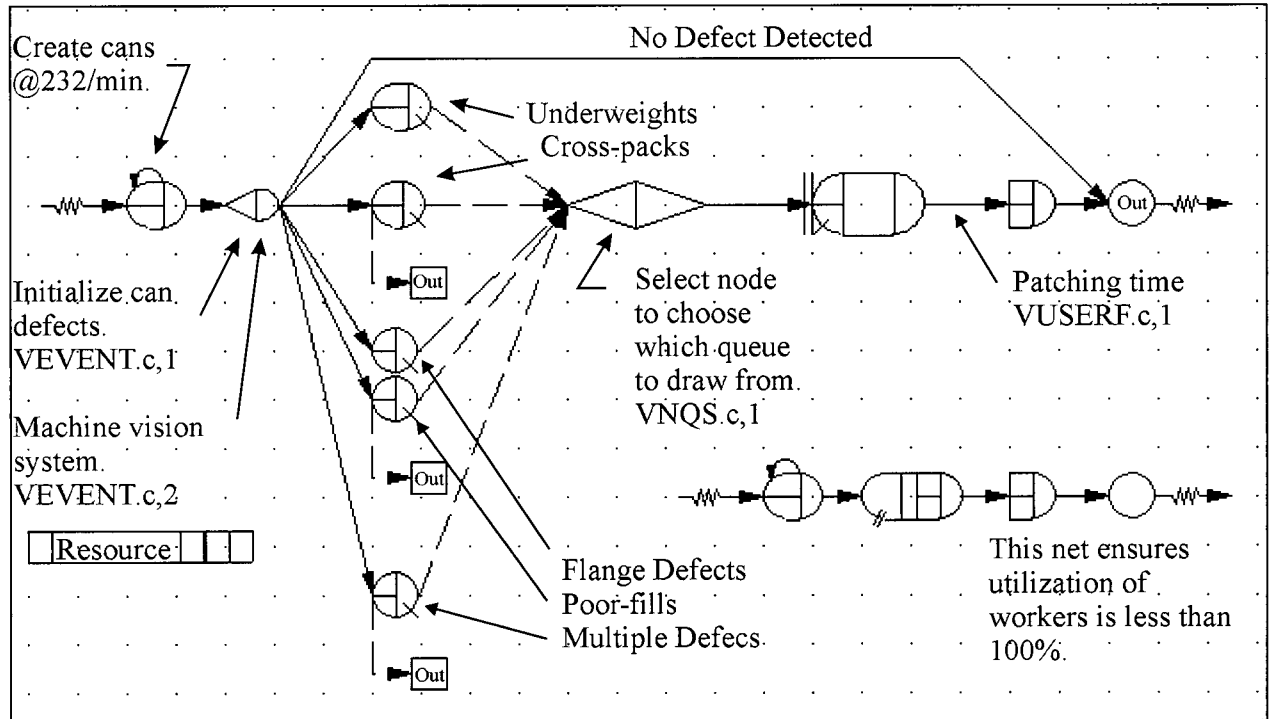


Figure 7.1: Simplified vision network

7.1.3 Physical Layout

The proposed physical layout of the modified patching table for cases V1 and V2 is detailed in Figure 7.3. The physical changes from the current patching table setup are the addition of the vision system, with multiple ejection devices (kickers), and multiple queues (tracks) on the table. The new waiting queues, parallel to the “through track” would hold cross-packs and flange defects respectively while the weight queue function is the same as in the current system. Estimated costs and part sourcing for the system are given in [69].

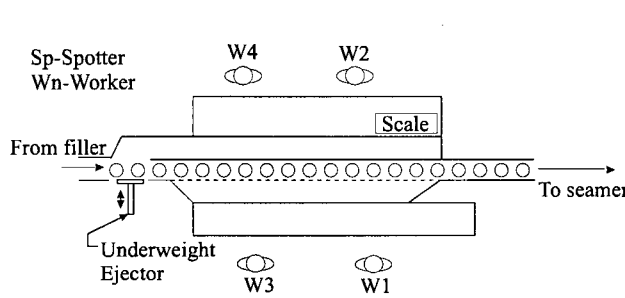


Figure 7.2: Current table layout.

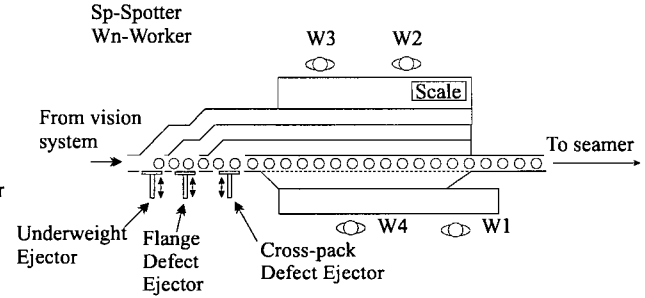


Figure 7.3: Modified table layout for vision system.

7.1.4 Results

In the V1 simulation all flange defects are patched; as a result no direct labour savings were realized, Table 7.3. However, significant indirect labour savings are realized due to the reduction in flange defects leaving the table. The quality index is essentially unchanged from the basecase and the safety index is, very high due to the patching of all of the identified flange defects.

The V2 implementation assumes that some of the flange defects are allowed to pass un-patched through the table, reducing the number of required workers to three. Thus the direct labour savings is one worker. The number of flange defects leaving the patching workcell is still reduced significantly relative to the basecase as indicated by a safety index of nearly two. The quality index, however, is reduced because less time is spent patching poor-fill and cross-pack defects than in either V1 or the basecase. In the V2 case the only time these types of defects would receive attention is when a localized reduction in the flange defect rate occurred enabling scanning of the line and patching of cross-packs and poor-fills.

<i>Automation Option</i>	<i>DLS</i>	<i>ILS</i>	<i>RMS</i>	<i>SI</i>	<i>QI</i>	<i>POP(years)</i>	<i>ROI</i>
V1	\$0.00	\$11,255	\$0.00	10.15	0.923	1.97	42.0%
V2	\$13,440	\$6,079.4	\$0.00	1.95	1.07	1.13	84.0%

Table 7.3: Effect of implementing automated inspection systems.

7.1.5 Sensitivity

The sensitivity of the results have been examined for perturbations of the following parameters:

- reliability of the vision system in classifying non-defects as non-defective (true negatives), Figure 7.4
- reliability of the vision system in classifying defects as defective (TP, true positives), Figure 7.5
- the handling time required for type 1 errors (i.e., the handling time required to recognize a non-defect and return it to the line), Figure 7.6
- the patching time required for on-the-fly type flange defects, Figure 7.7
- the time utilization of the workers, Figure 7.8

Figure 7.4 shows a 17.7% variation in expected savings for a four percent variation in susceptibility to making type 1 errors. The vertical divisions in the figures mark the operational point of each parameter. The quality index varies about 0.5% while the safety index varies by 90%. Figure 7.5 shows the variation caused by a six percent change in the systems ability to correctly identify defects. A 0.85% variation in estimated savings, a 0.1% variation in QI and a 5% variation in SI result from this perturbation. Figure 7.6 shows the variability due to a 150% change in the handling time of non-defects. The estimated savings change by 1.36%, QI changes by 0.1% and SI varies by 8%. These small effects are in part due to the fact that the operational point for the susceptibility of type 1 errors was only one percent. If this value was increased the effects of varying the handling time of these mis-classifications would be more noticeable (i.e., the effects of this parameter are dependent not only on the task time but also the frequency of this task occurring). A one hundred percent variation in the required patching time for on-the-fly type flange defects results in a 15% variation in the expected savings. The safety index varies by 95% while the quality index ranges varies by 2.1%, Figure 7.7. A ten percent variation in worker utilization generates a change in the expected savings of 12.8%. The safety and quality indices vary by 81% and 1.7% respectively. Based on this analysis the most important factor to be controlled is the susceptibility of the system to making type 1 errors.

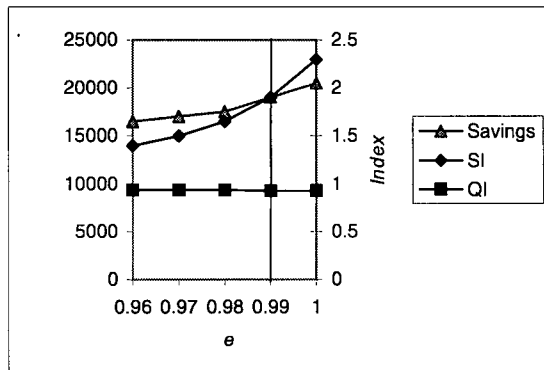


Figure 7.4: Sensitivity to type 1 errors.

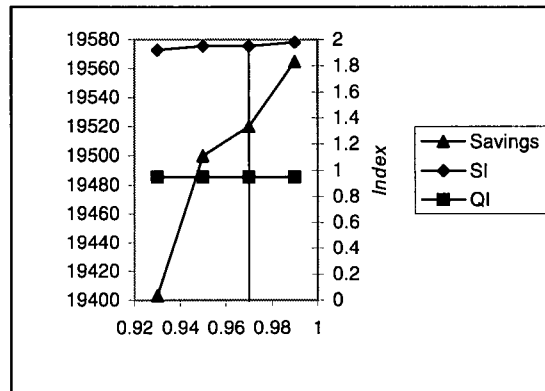


Figure 7.5: Sensitivity to defect recognition accuracy.

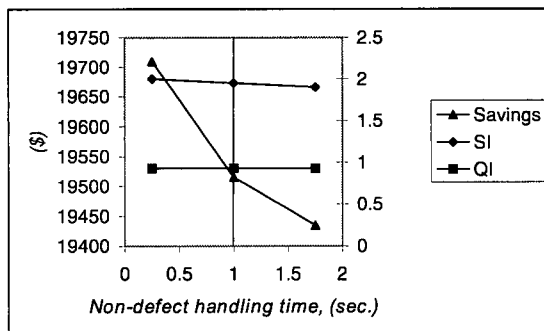


Figure 7.6: Sensitivity to non-defect handling time.

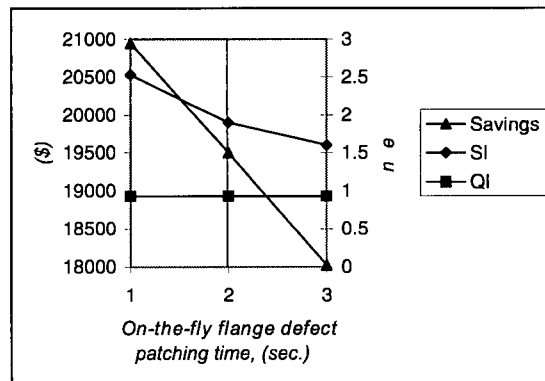


Figure 7.7: Sensitivity to on-the-fly flange defect patching time.

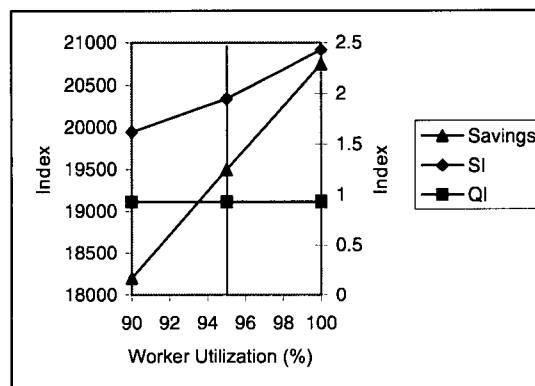


Figure 7.8: Sensitivity to worker time utilization.

7.2 Automated Weight Correction (W)

7.2.1 System Assumptions

The assumed meat addition system is one of the chunk style meat addition systems, (Section 6.2.2). A cost estimate for this system has been made and is included in [69]. The weight addition system is feed by the checkweigher and is assumed to patch underweight cans exactly to the label weight. All other patching activities and task times remain the same as in the vision, (V), implementations.

7.2.2 Implementations

A checkweighing system operating with 100% weighing accuracy has been assumed for the present analysis. Two implementations of the automated weight patching system have been considered in the analysis, W1 and W2. W1 assumes that the mean fill weight is the same as observed on the day of the experimental data collection in Prince Rupert, namely 216.9 grams. This represents a 3.9 gram average overfill and thus the implementation of automated weight patching results in tremendous raw material savings. At this high fill weight, however, the basecase simulation requires only 37% utilization of one worker to patch all the weight defects. Thus the elimination of this patching task does not translate into the reduction of the required patching table staff. Instead, in the W1 simulation, this spare utilization created by the implementation of an automated weight patching system is directed into the patching of other defects. For an optimistic evaluation we can assume that all of this utilization is directed into patching of flange defects. This assumption provides the best case scenario, since in reality some of the utilization would probably be used to patch other types of defects. Simulation of a worker patching flange defects for an extra 37% of the time results in a flange defect reduction of 19.5%. Applying this to the 6.4% output flange defect rate observed in the basecase results in a new output flange defect rate of 5.15%. This

translates to indirect labour savings through a reduction in hand culls, Table 7.5. The quality index is unchanged and the safety index is greater than one. This represents the best case as mentioned and upon implementation we would see somewhat of a shift in these indices, SI would drop and QI would increase. Raw material savings are very high in that the mean fill weight is almost four grams over the label weight.

Consultation with B.C. Packers management personnel indicated that the mean fill weight recorded during the data collection day was atypical. The mean fill weight is normally set to 1 gram over the label weight (i.e., 214 grams).¹ Thus, a second economic analysis was done using a mean fill weight of 214 grams, and designated W2, Table 7.5. The material savings are reduced significantly although direct labour savings are now realizable. Simulation shows that 163.3% utilization is required in order to patch all of the underweights. Since the utilization rate required to replace a single worker is 80.6%, just over two workers are needed to keep pace with the weight defects. Two possibilities are considered. W2a, assumes that upon implementation of the automated system the producer eliminates two workers from the patching table. W2b assumes that the producer eliminates one worker and leaves one in place to boost quality and/or reduce flange defects. In this case a “best case scenario” is again employed where the extra worker patches only flange defects. A worker employed 80.6% of the time (i.e., fully utilized) reduces flange defects by 42.6%. In this simulation, flange defects are reduced by 42.6% to 3.7% from the basecase, 6.4%.

In all of these simulations spare labour capacity was directed into flange defects. This can only be achieved if a vision system is implemented to detect defects. However, even if a vision system was not implemented the savings from automated weight addition alone would not be greatly affected since most savings are from direct labour saving (reduced weight patching) and

¹The actual mean fill weight is difficult to establish. Management guidelines are 1 gram above label weight. However, this results in a large number of underweight cans and a high workload for the patching table workers. Therefore, the workers (who control the filler) may tend to increase the average fill weight in order to reduce the workload at the patching table.

raw material savings.

7.2.3 Physical Layout

The physical layout for the weight addition automation options is not well defined since the actual mechanisms which would be employed are not known. The automated station would have a waiting queue feed by the checkweigher and would employ a fixturing mechanism with which to secure the can for patching. The station would need to precede the patching table since the patching of weight problems would not necessarily repair other defects present in the fill. Section 6.2.2 details two chunk meat addition concepts.

7.2.4 Results

From Table 7.4 it is apparent that even with a costly weight addition system assumed, automated meat addition is an economically viable option. If, in the future consumer surveys shows that the less expensive weight addition method using a meat slurry (Section 6.2.1) is acceptable, this option would only increase in viability. Furthermore, given the current trade off between low mean fill weight and worker utilization, the introduction of automation for this process has multiple benefits (savings of raw materials, and direct/indirect labour savings).

<i>Automation Option</i>	<i>DLS</i>	<i>ILS</i>	<i>RMS</i>	<i>SI</i>	<i>QI</i>	<i>POP(years)</i>	<i>ROI</i>
W1	\$0.00	\$2,432.9	\$64,945	1.24	1.0	1.86	38.8%
W2a	\$13,440	\$5,303.4	\$15,568	1.74	1.0	3.64	22.4%
W2b	\$26,880	\$0.00	\$15,568	1.0	1.0	2.94	26.8%

Table 7.4: Effect of adding an automated weight addition systems to the patching table system.

7.2.5 Sensitivity

No additional parameters outside of those examined in the vision sensitivity section, Section 7.1.5, are introduced and hence a sensitivity analysis is not indicated here.

7.3 Hybrid System

Using the vision system it is possible to split the utilization of the costly, but effective, trimmer currently used by B.C. Packers over multiple lines. At present the trimmer is utilized by a single line and trims every can regardless of whether or not it has any flange defects. With a vision system, only those cans which are identified as defective need to be routed to the trimmer.

7.3.1 System Assumptions

The data from Prince Rupert showed a flange defect rate of about 15% and hence the trimmer could theoretically handle the load from over six separate lines. It is, however, somewhat unlikely that the producer would elect to install six vision systems, at least initially. In addition routing problems become very complex as more lines are fed to the trimmer, for two reasons. Firstly the product from the various lines cannot be mixed, and secondly overhead routing becomes necessary. For these reasons a workcell employing two vision system equipped lines in conjunction with a single trimmer has been analysed. The results of this implementation, denoted VT, are shown in Table 7.5. The modifications required to the line are shown in Figure 7.9.

It was required for simulation purposes to estimate the efficiency of the trimmer in correcting flange defects. Information from personnel at B.C. Packers indicates that the trimmer correctly trims flange defects better than 95% of the time.

7.3.2 Physical Layout

In this implementation the cans which the vision system classes as having flange defects are sent to the trimmer and after trimming are returned to the vision system for reclassification. The action of trimming is a simple routine called by the event node code, 3, (VTEVENT.c,3) which repairs the flange defect 95% of the time. The workers are assumed to patch flange defects which go unnoticed

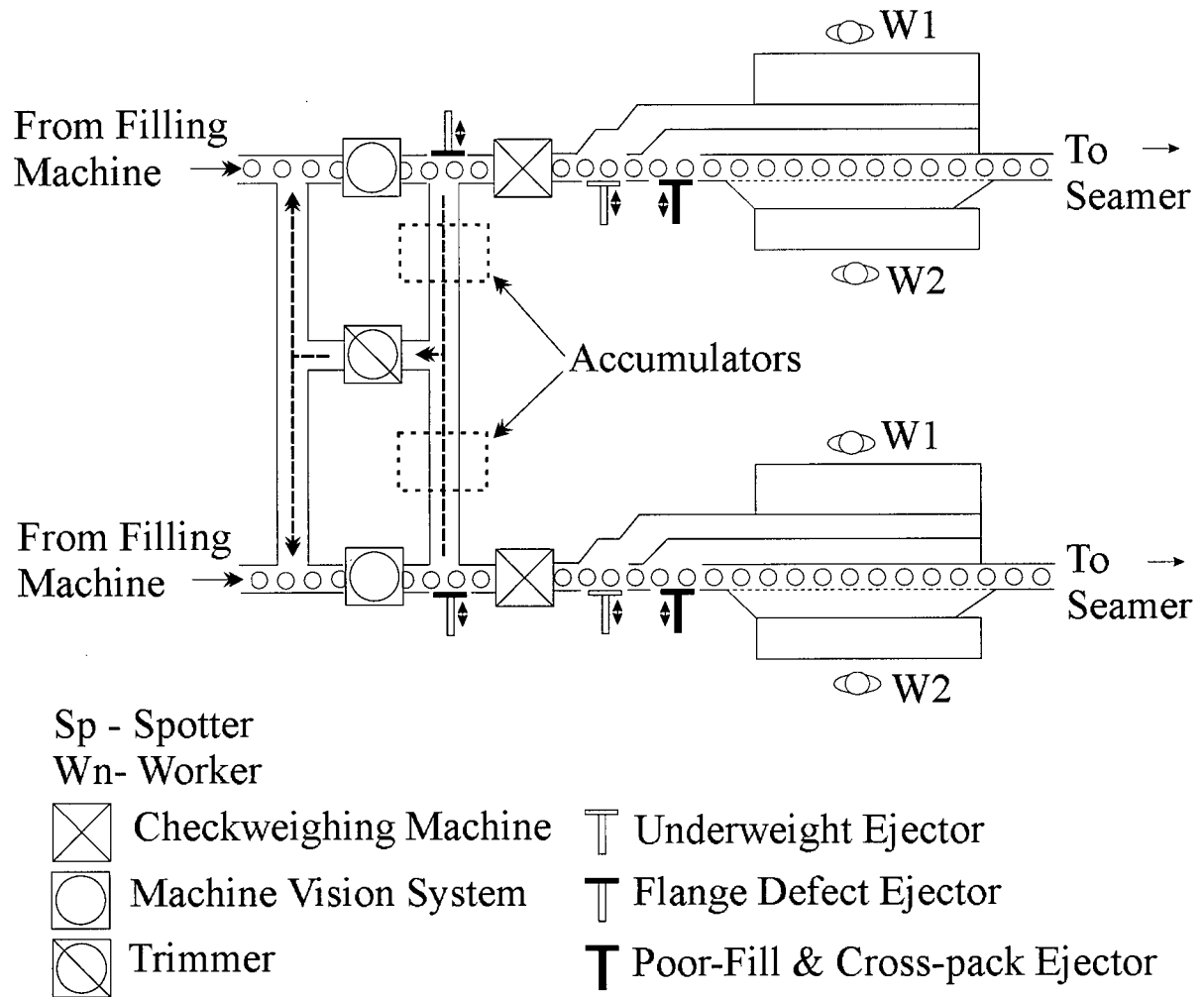


Figure 7.9: Recommended physical layout for the combined vision/trimmer automation option.

by the vision system but which are classed as having other defects and hence enter other defect queues for patching. The network layout of the system is shown in Figure 7.10. The physical setup required differs little from the vision only system. Cost estimates for the required modifications and the costs associated with the trimmer are found in [69].

In order to install the trimmer and the vision system in the physical plant a number of criteria must be met. The physical size of the cell must be constrained to be no larger than the present workcell. Additionally, it is essential that the required modifications be kept to a minimum in

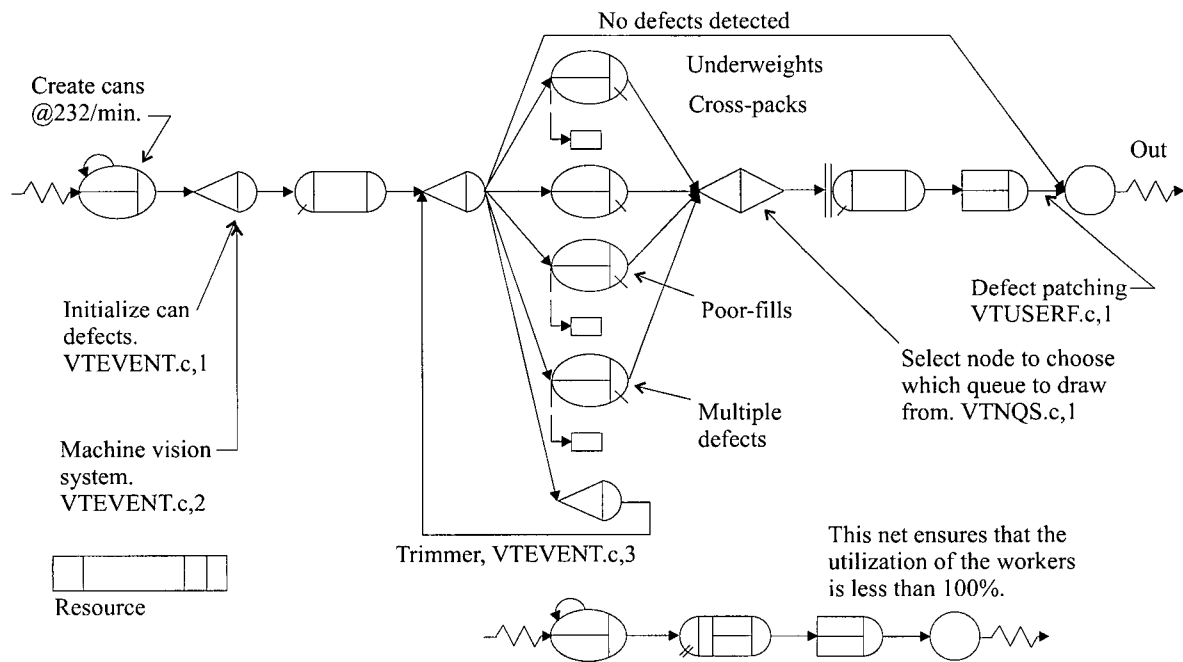


Figure 7.10: Simplified vision/trimmer network

order to minimize capital costs and to disrupt the present working conditions as little as possible. Reducing the deviation from the present workcell will mean less stress for the patching workers and a minimization of training on the new workcell. The additional waiting queue required for cross-packs needs to be controlled such that it does not overflow. To do this a number of sensors are placed along the queue length such that the capacity of the queue may be monitored.

7.3.3 Results

Implementation of the vision system in conjunction with the trimmer serves to relieve two patching workers and in addition generates significant savings due to the reduction in flange defects. This set-up increases the safety factor markedly and in addition improves the quality index somewhat, Table 7.5. Consultation with B.C. Packers personnel indicates that the trimmer mechanism alone results in significant worker savings not indicated with the present analysis and hence the benefits of this dual configuration represents a highly conservative estimate.

<i>Automation Option</i>	<i>DLS</i>	<i>ILS</i>	<i>RMS</i>	<i>SI</i>	<i>QI</i>	<i>POP(years)</i>	ROI
V1	\$0.00	\$11,255	\$0.00	10.15	0.923	1.97	42.0%
V2	\$13,440	\$6,079.4	\$0.00	1.95	1.07	1.13	84.0%
W1	\$0.00	\$2,432.9	\$64,945	1.24	1.0	1.86	38.8%
W2a	\$13,440	\$5,303.4	\$15,568	1.74	1.0	3.64	22.4%
W2b	\$26,880	\$0.00	\$15,568	1.0	1.0	2.94	26.8%
VT	\$26,880	\$11,572.6	\$0.00	14.2	0.897	1.09	88.0%

Table 7.5: Automation option comparison.

7.3.4 Sensitivity

No additional parameters were introduced in the VT implementation which warrant a sensitivity analysis.

7.4 Summary

The results of this analysis indicate that the industrial producers could save a considerable amount of money through the implementation of various automation options. The savings for each proposed option are tabulated in Table 7.5. The option with the highest payback for the least investment is case V2. This case employs the vision system as a stand alone add on to the workcell. The V2 implementation would pay for itself in a little over one year with an 84% ROI while providing a doubling of the safety factor and leaving the quality index essentially undisturbed.

Chapter 8

Conclusions and Recommendations

8.1 Conclusions

Discrete event simulations have been identified as the most suitable technique for use in modelling mixed automation processes. The use of time motion studies to predict task times required by manual workers is well documented. Relevant factors affecting manual inspection accuracy at the patching table include; event rate, vigilance duration and signal density.

The task times required by the patching workers to repair can fill defects have been shown to follow lognormal distributions. The filling machine fills 213 gram cans with a standard deviation of about 5 grams. This represents a 250% overshoot of the desired variation indicated by B.C. Packers.

A simulation of the current patching table (i.e., the basecase) was completed and provided good results and a reference by which to judge table modifications. Simulation of various table automation options was successfully completed.

The major savings incurred through automation of the patching table are due to three factors: direct labour savings incurred through the reduction in required patching workers, indirect labour

savings incurred through the reduction in the number of required hand culls leading to fewer required inspections and lastly savings generated through the reduction in raw material wastage.

Analyzing the simulation results of a vision system implementation indicate lucrative savings would be realized. An 84% ROI and just over one year payout period are indicated for the second of the two inspection system implementations considered. The costs associated with the economic analysis for this implementation are considered to have a high confidence.

The implementation of an automated weight addition system indicates less attractive economic savings, 39% ROI in the best case. However, one can note that the cost estimate for this system was based on a conceptual design and hence could vary considerably. A higher confidence cost estimate would provide a better estimate on the benefits which could be achieved.

The coupling of the automated trimming machine used by the industrial producer with the automated inspection system allows for the sharing of one trimmer between two lines. The modifications are such that different product (i.e., species) can be run on the two lines without mixing. This system indicates a 69.6% ROI which would be paid off in about a season and a third. Additionally this configuration nearly eliminates the problem of flange defects as indicated by a safety index of 14.2.

8.2 Recommendations

- A proposal for implementation is that the producers first invest in the vision system alone, V1, and evaluate its effectiveness. When confidence is established in the system a worker may be removed from the line, V2.
 - Following this implementation the coupling of two lines to a single trimmer can be facilitated with the purchase of an additional vision system. The effect of coupling two

lines to a single trimmer is that the lease is divided over the production of the two lines. This implementation, which ensures no mixing of the products from the two lines has herein been referred to as VT. This implementation would pay for itself in one and a third seasons if it was implemented from the start, (i.e., prior to purchase of the vision system).

- It should be noted that the quoted savings for the VT implementation are for one line and hence the savings are actually double when considering the two lines involved.
- It is suggested that the producer carefully examine the mean fill weight of the pack utilizing an independent audit. A screening of the weight for instance after the pack would suffice.
 - If indeed the mean fill weight is tightly controlled at only one gram over the fill weight then the pursuit of automated weight addition is not highly economically viable.
 - Due to the high cost of the raw material, if this mean is higher than one gram over the label weight the viability of an automated weight correction system becomes significant.
 - Due to the highly labour intensive nature of weight patching, if the cost of labour was to significantly increase, the viability of automation would again become viable.
 - If the producer conducts customer surveys which find the use of ground salmon slurry deposited underneath the salmon steak to be acceptable, the automation of weight addition will be very viable.
 - The high cost of the proposed meat addition system is the major detriment to the justification of automating this process. In addition if the producer is able to secure an acceptable cheaper automation option for this procedure the viability would again be high.

- The industrial producer should privately examine the benefits of coupling the trimmer to two lines with the use of a vision system.
 - Consultation with B.C. Packers personnel indicates conflicting information on the cost effectiveness of the trimmer as it now stands, and hence the estimate made herein is subject to these uncertainties. A savings analysis based on the methods employed in this study place the savings incurred through the use of the trimmer alone at the low end of those indicated by B.C. Packers personnel. Hence, the VT analysis may be significantly more lucrative than already indicated. This analysis has not been included within this study in order to preserve the proprietary concerns of the industrial producer.
- Analysis of the experimental data indicates that a single day sample of the patching data is not sufficient for analysis.
 - Studies which are not differential in nature, should endeavour to secure a least 5 shifts of patching table data.
 - The traits of cross-packed, flange defective and underweight cans are well defined, however, those of poor-fill require more attention.
 - More data should be acquired and used to expand the model to better deal with poor-fills and to reinforce the task time distributions.
 - A more diverse data set should be acquired for patching tables with other configurations such as those employing upto 7 workers and those with the trimmer operating as well as for various fish grades and species.

Appendix A

Distribution Acceptance

The Chi-squared Goodness of Fit Test

The Chi-squared goodness of fit test is a hypothesis type test used to determine whether or not a population (i.e. the sample) has a particular underlying theoretical distribution. This test checks the correlation between the frequency of occurrence of the sample data and the expected frequencies generated by the hypothesized distribution. This test of goodness of fit is based on the Chi-squared quantity [70]:

$$\chi^2 = \sum_{i=1}^k \frac{(o_i - e_i)^2}{e_i} \quad (\text{A-1})$$

χ^2 is a random variable whose distribution is closely matched by the Chi-squared distribution with degrees of freedom (ν), $\nu = k-1$. The o_i and e_i represent the observed and expected frequencies for the i^{th} cell. If χ^2 falls below the rejection area of the Chi-squared distribution for the chosen level of significance at the appropriate number of degrees of freedom then there is insufficient grounds upon which to reject the null hypothesis. In our case the null hypothesis being a specified distribution type, mean and standard deviation.

Flange Patch Times

Off-line

Computing χ^2 for the data in table A-1 generated by a lognormal distribution with mean, 0.094649, and standard deviation, 1.255953, results in a value of 1.648. The number of degrees of freedom is $k-1$, thus $\nu=5-1 = 4$. For significance level $\alpha=0.05$ the rejection criterion starts at 9.488, thus we cannot reject the hypothesis of a lognormal distribution with $\mu=0.0946$ and $\sigma=1.256$.

Poor-fill Patch Times

Computing χ^2 for the data in table A-3 generated by a lognormal distribution with mean, 3.025157, and standard deviation, 0.382421, results in a value of 0.214. The number of degrees of freedom is $k-1$, thus $\nu=3-1 = 2$. For significance level $\alpha=0.05$ the rejection criterion starts at 5.991, thus we cannot reject the hypothesis of a lognormal distribution with $\mu=3.025$ seconds and $\sigma=0.382$.

$bin(\ln(sec.))$	o_i		e_i	
$-\infty \dots -2.4$	0	7	0.916	5.901
$-2.4 \dots -2.0$	2		0.943	
$-2.0 \dots -1.6$	3		1.597	
$-1.6 \dots -1.2$	2		2.445	
$-1.2 \dots -0.8$	4	7	3.386	7.626
$-0.8 \dots -0.4$	3		4.240	
$-0.4 \dots 0.0$	6	8	4.802	9.719
$0.0 \dots 0.4$	2		4.917	
$0.4 \dots 0.8$	7	11	4.554	8.367
$0.8 \dots 1.2$	4		3.814	
$1.2 \dots 1.6$	2	6	2.888	7.387
$1.6 \dots 2.0$	4		1.978	
$2.0 \dots 2.4$	0		1.225	
$2.4 \dots 2.8$	0		0.686	
$2.8 \dots \infty$	0		0.609	

Table A-1: Patch time data for off-line patches.

$bin(\ln(sec.))$	o_i		e_i	
$\infty \dots 0.4$	5	8	4.227	6.130
$0.4 \dots 0.6$	3		1.903	
$0.6 \dots 0.8$	3	6	2.422	8.769
$0.8 \dots 1.0$	1		2.941	
$1.0 \dots 1.2$	2		3.405	
$1.2 \dots 1.4$	2	7	3.761	11.700
$1.4 \dots 1.6$	2		3.961	
$1.6 \dots 1.8$	3		3.979	
$1.8 \dots 2.0$	4	11	3.812	7.294
$2.0 \dots 2.2$	7		3.483	
$2.2 \dots 2.4$	3	14	3.035	12.106
$2.4 \dots 2.6$	3		2.522	
$2.6 \dots 2.8$	4		2.000	
$2.8 \dots 3.0$	4		1.511	
$3.0 \dots \infty$	0		3.038	

Table A-2: Patch time data for weight.

$bin(sec.)$	o_i		e_i	
$-\infty \dots 1.8$	2	12	0.016	10.897
$1.8 \dots 2.1$	0		0.163	
$2.1 \dots 2.4$	1		0.995	
$2.4 \dots 2.7$	2		3.352	
$2.7 \dots 3.0$	7		6.352	
$3.0 \dots 3.3$	6	6	6.671	6.671
$3.3 \dots 3.6$	3	5	3.905	5.432
$3.6 \dots 3.9$	2		1.272	
$3.9 \dots \infty$	0		0.255	

Table A-3: Patch time data for poor-fill patches.

$bin(\ln(sec.))$	o_i		e_i	
$-\infty \dots 0.2$	3	9	1.178	8.212
$0.2 \dots 0.4$	0		1.342	
$0.4 \dots 0.6$	2		2.287	
$0.6 \dots 0.8$	4		3.405	
$0.8 \dots 1.0$	4	7	4.433	9.479
$1.0 \dots 1.2$	3		5.046	
$1.2 \dots 1.4$	8	8	5.021	5.021
$1.4 \dots 1.6$	3	11	4.367	12.288
$1.6 \dots 1.8$	5		3.321	
$1.8 \dots 2.0$	1		2.208	
$2.0 \dots 2.2$	2		1.283	
$2.2 \dots \infty$	0		1.108	

Table A-4: Patch time data for cross-pack.

Cross-pack Patch Times

Type (a) Patches, (cutting required)

The amount of available data for this type of patching task was extremely limited and hence the distribution was not discernible. For this reason a normal distribution was placed at the mean of the sample data with the sample standard deviation. Since only four data points for this type of patch were observed there is little point in performing goodness of fit analysis on the distribution fit. The sample data: 16.8, 27, 30.2 and 16 seconds gives a mean of 22.5 seconds with a standard deviation of 7.17 seconds.

Type (b) Patches, (no-cutting)

Computing χ^2 for the data in table A-4 generated by a lognormal distribution with mean, 1.192569, and standard deviation, 0.542464, results in a value of 2.627. The number of degrees of freedom is $k-1$, thus $\nu=5-1 = 4$. For significance level $\alpha=0.05$ the rejection criterion starts at 9.488, thus we cannot reject the hypothesis of a lognormal distribution with $\mu=1.19$ seconds and $\sigma=0.54$.

Weight

Defect Distribution

It was assumed in the analysis that the incoming distribution of weights was normally distributed around the mean. In order to test this, Geary's test [71] is used to test the normality assumption.

$$u = \frac{\sqrt{\pi/2} \sum_{i=1}^n |X_i - \bar{X}| / n}{\sqrt{\sum_{i=1}^n (X_i - \bar{X})^2 / n}} \quad (\text{A-2})$$

This test is based on the ratio of two estimators or the sample standard deviation. The numerator is a good estimator of σ if the distribution is normal and the denominator is a good estimator of σ regardless of the distribution. Hence, departures from u values near 1.0 are indicative of departures from normality. Using the data of the 565 can sample (should this go into an appendix ????) with a mean of 216.9 grams the Geary test gives a u value of 0.977 indicating a high confidence in assuming a normally distributed weight distribution.

Weight Patch Times

Computing χ^2 for the data in table A-2 generated from a lognormal distribution with mean, 1.618938, and standard deviation, 0.917078, results in a value of 5.512. The number of degrees of freedom is $k-1$, thus $\nu=5-1 = 4$. For significance level $\alpha=0.05$ the rejection criterion starts at 9.488, thus we cannot reject the hypothesis of a lognormal distribution with $\mu=1.62$ seconds and $\sigma=0.92$.

Appendix B

Quality Control Survey

University of British Columbia
Industrial Automation Laboratory
Salmon Can Filling Quality Control Survey

In order to help us gain a better understanding of the salmon can filling process and the factors that effect the quality of a can of salmon, the following survey has been developed. In particular, the criteria and methods for identifying and correcting patching problems before the can is sealed is being studied. In conjunction with BC Packers personnel we have developed a number of questions, contained herein. In addition, we have packed and photographed a number of cans of salmon with possible filling problems. These may not be representative of the problems that are actually produced by a real salmon canning machine. If the packed cans are not representative of the problems actually seen on the patching line, please describe the differences as best as you can. We greatly appreciate your time and input in this project.

1. A number of quality control problems are indicated below. Please rate them on their severity and frequency from 1 to 5: 5 meaning "critical failure" or "most common" - 1 meaning "minor problem" or "least common". Please fill in any other quality control problems which can occur on the canning line.

	Severity					Frequency				
- Cross-pack	5	4	3	2	1	5	4	3	2	1
- Bones on the top of the steak	5	4	3	2	1	5	4	3	2	1
- Too many backbones	5	4	3	2	1	5	4	3	2	1
- Skin and/or bone obstructing the edge .	5	4	3	2	1	5	4	3	2	1
- Under-weight	5	4	3	2	1	5	4	3	2	1
- Over-weight	5	4	3	2	1	5	4	3	2	1
- Poor meat colouration	5	4	3	2	1	5	4	3	2	1
- Damage to can edge	5	4	3	2	1	5	4	3	2	1
- Too much air space	5	4	3	2	1	5	4	3	2	1
- Other _____	5	4	3	2	1	5	4	3	2	1

- Other _____	5	4	3	2	1	5	4	3	2	1
_____						5	4	3	2	1
- Other _____	5	4	3	2	1	5	4	3	2	1

2. Please briefly list, in order of importance, the features which are associated with "good appearance" and "poor appearance."

Good Appearance

1. _____

Poor Appearance

1. _____

3. Can the presence of meat on the edge of the can cause a sealing problem?
4. Is there a limit as to the number of small steaks with backbones which would be allowed to be in one can?

- Yes () If yes when? _____
- Sometimes () _____
- No () _____
- Yes (), Max. # ()
- No ()
- Comment: _____
- _____

5. How much head space is considered too little and too much for each of the following can sizes?

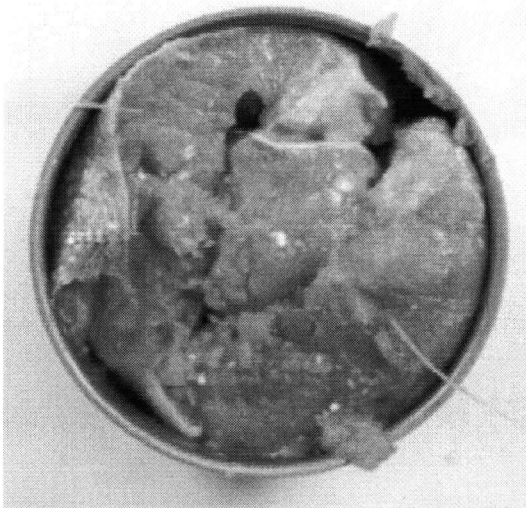
- | | | |
|---------|-------|-------|
| 1/4 lb. | Low: | _____ |
| | High: | _____ |
| 1/2 lb. | Low: | _____ |
| | High: | _____ |
| 1 lb. | Low: | _____ |
| | High: | _____ |

6. The following are pictures of packed salmon cans. Please indicate which cans have problems that warrant patching by circling the problem and commenting beside each picture what steps would be required to patch the can. If the problem is extremely rare or very common please make a note of this. In particular, we are interested in determining the following:

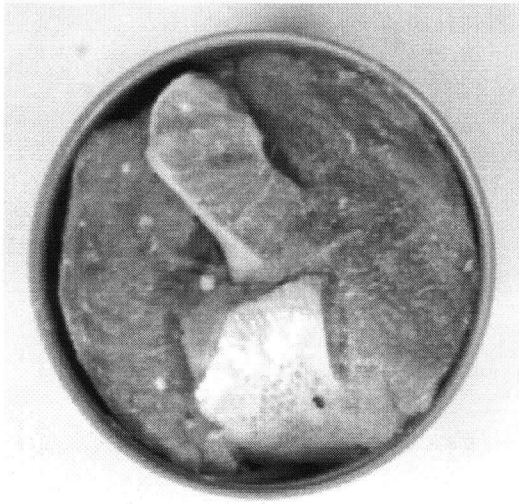
- Which cans have enough meat, skin, or bone on the edge of the can to require cleaning.
- Which cans have enough skin showing to be considered a cross packed can or simply unacceptable.



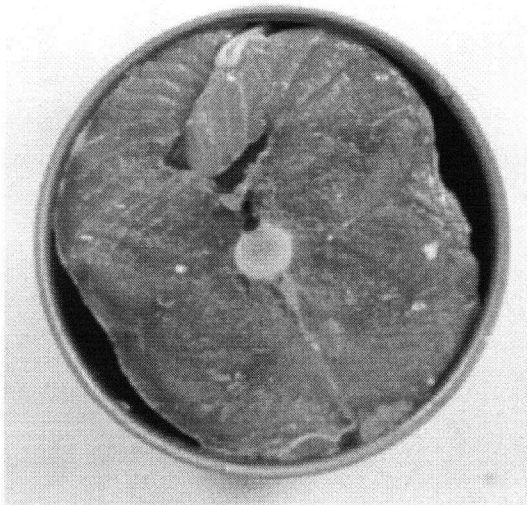
Can 1 Problems:



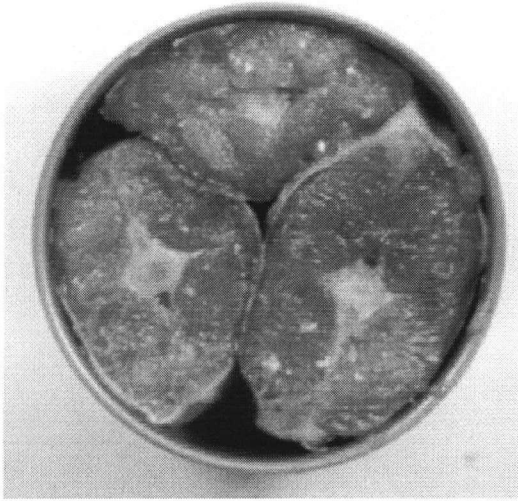
Can 2 Problems:



Can 3 Problems:



Can 4 Problems:



Can 5 Problems:



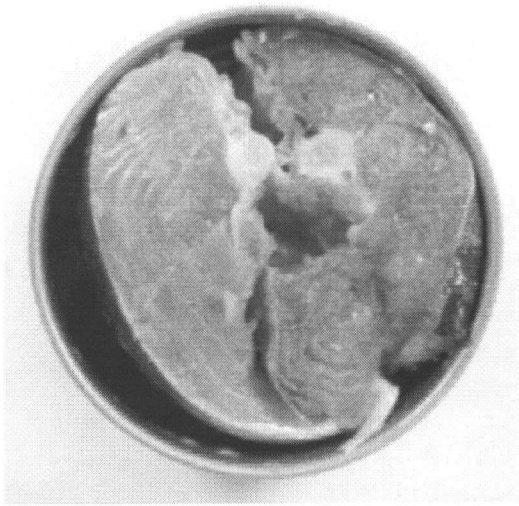
Can 6 Problems:



Can 7 Problems:



Can 8 Problems:



Can 9 Problems:



Can 10 Problems:



Can 11 Problems:



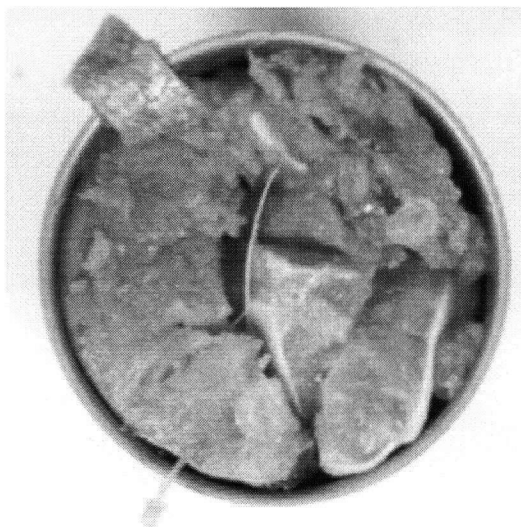
Can 12 Problems:



Can 13 Problems:



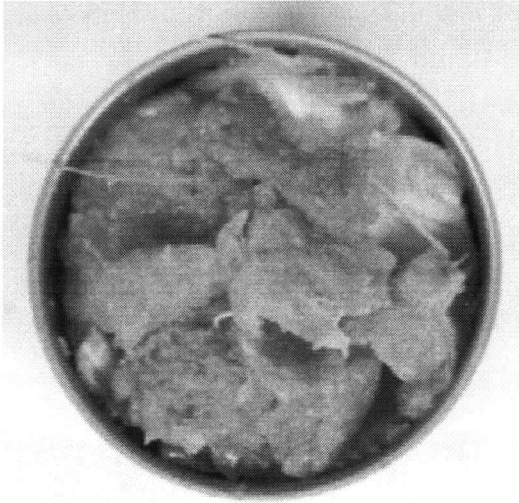
Can 14 Problems:



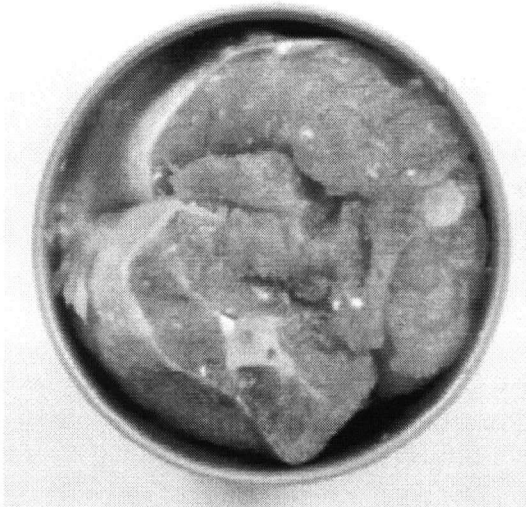
Can 15 Problems:



Can 16 Problems:



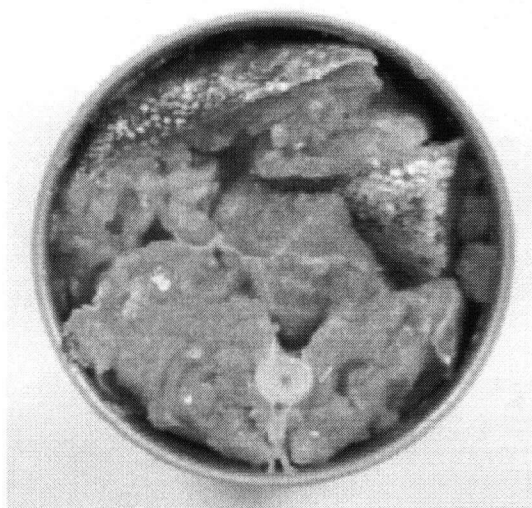
Can 17 Problems:



Can 18 Problems:



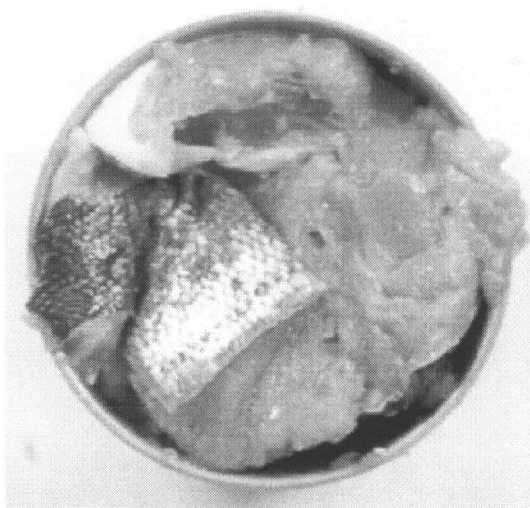
Can 19 Problems:



Can 20 Problems:



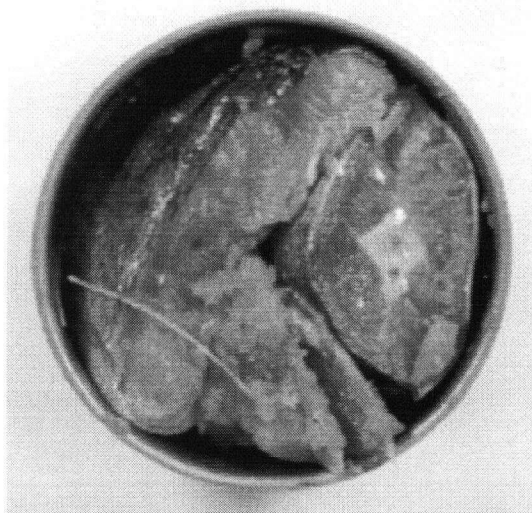
Can 21 Problems:



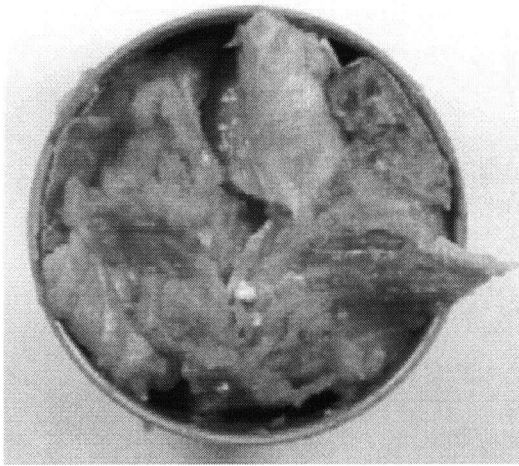
Can 22 Problems:



Can 23 Problems:



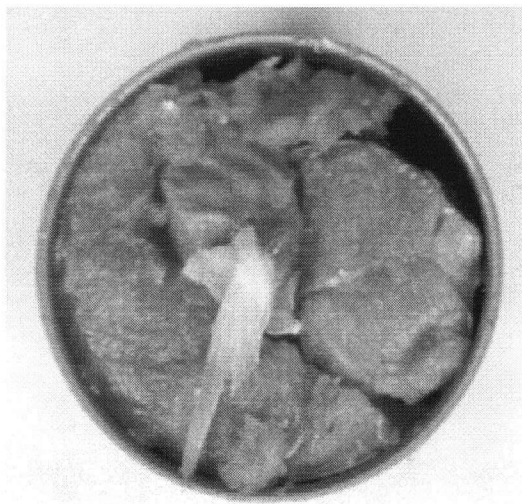
Can 24 Problems:



Can 25 Problems:



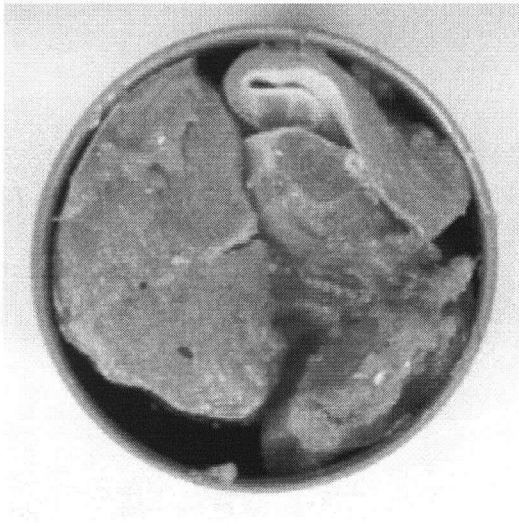
Can 26 Problems:



Can 27 Problems:



Can 28 Problems:



Can 29 Problems:



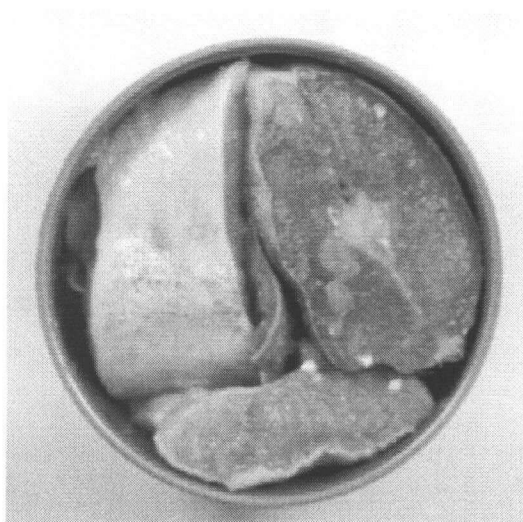
Can 30 Problems:



Can 31 Problems:



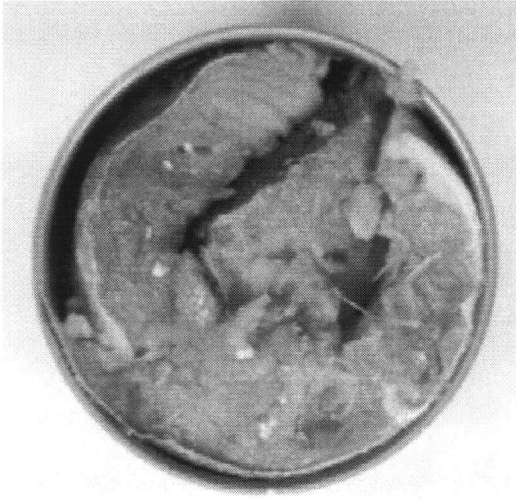
Can 32 Problems:



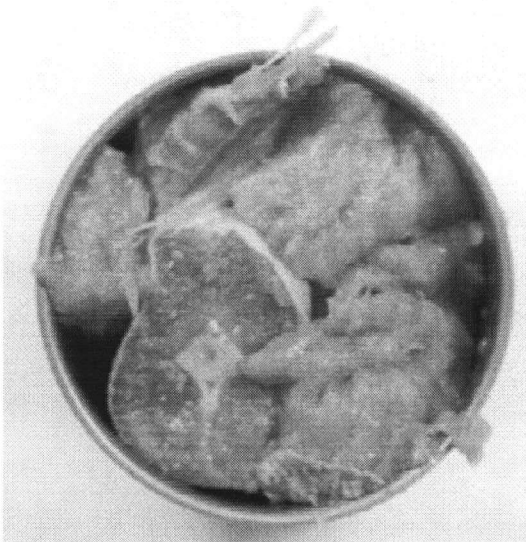
Can 33 Problems:



Can 34 Problems:



Can 35 Problems:



Can 36 Problems:

Appendix C

Awesim Networks

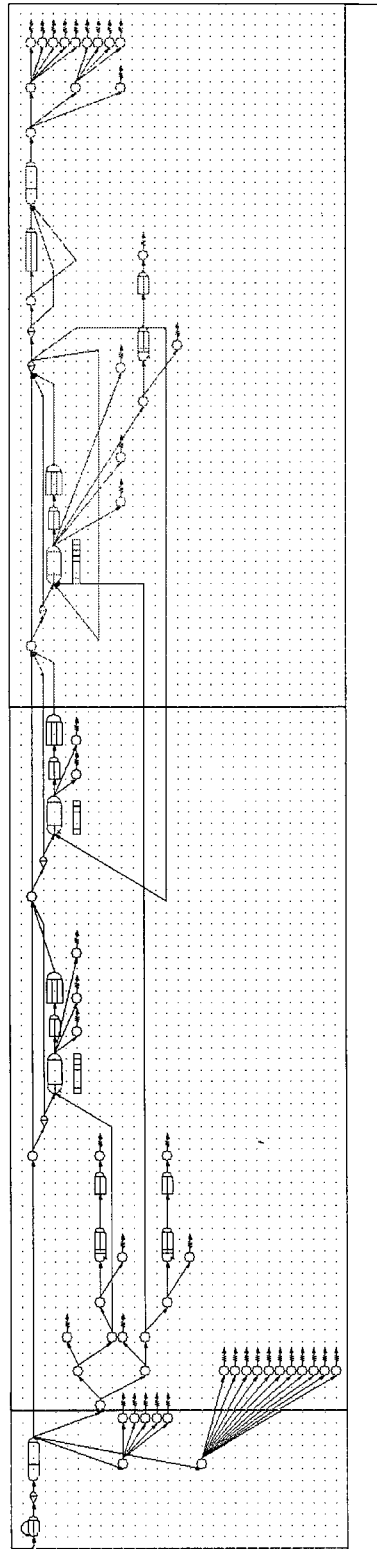


Figure C.1: Complete network employed in the basecase simulations.

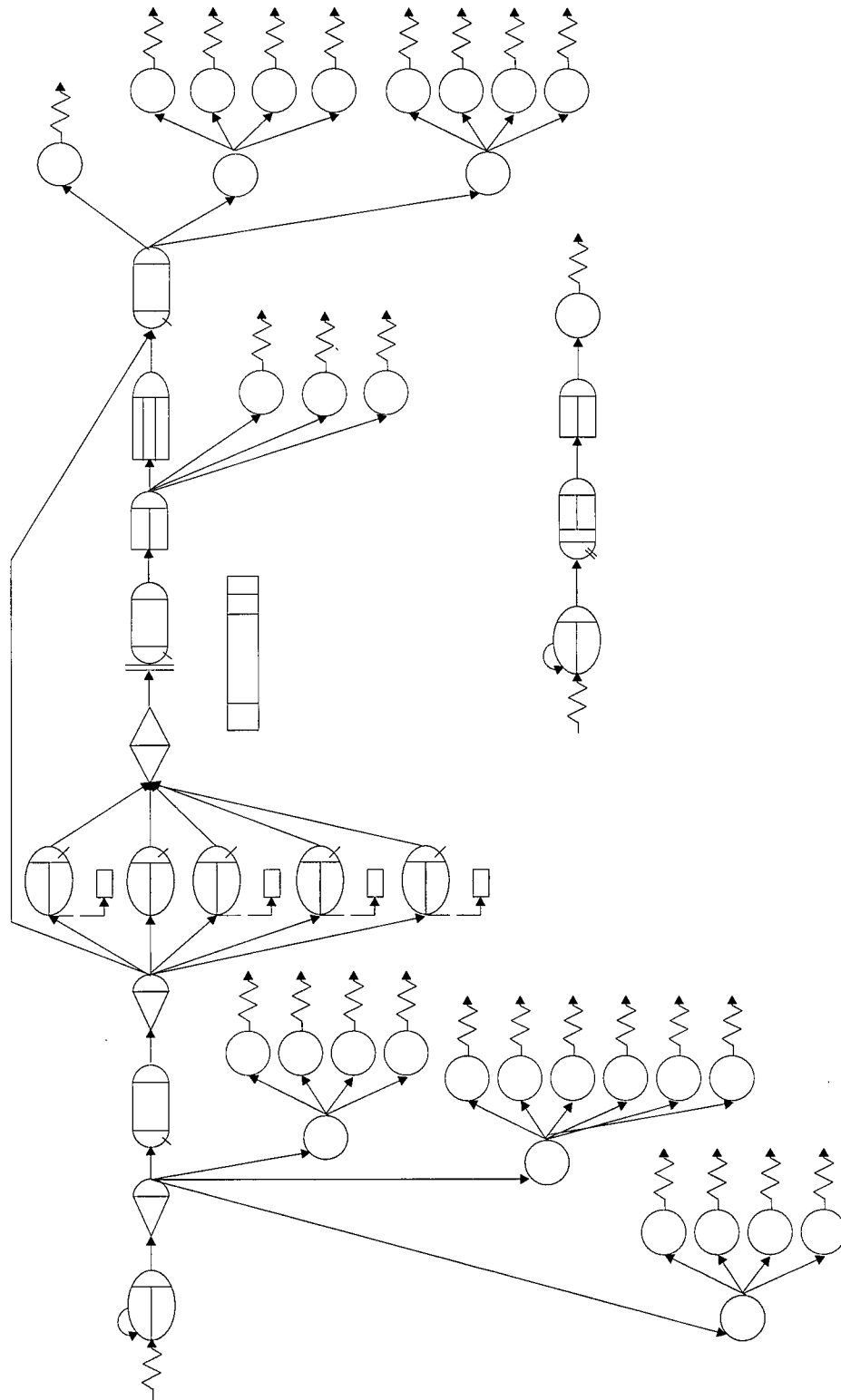


Figure C.2: Complete network employed in the vision simulations.

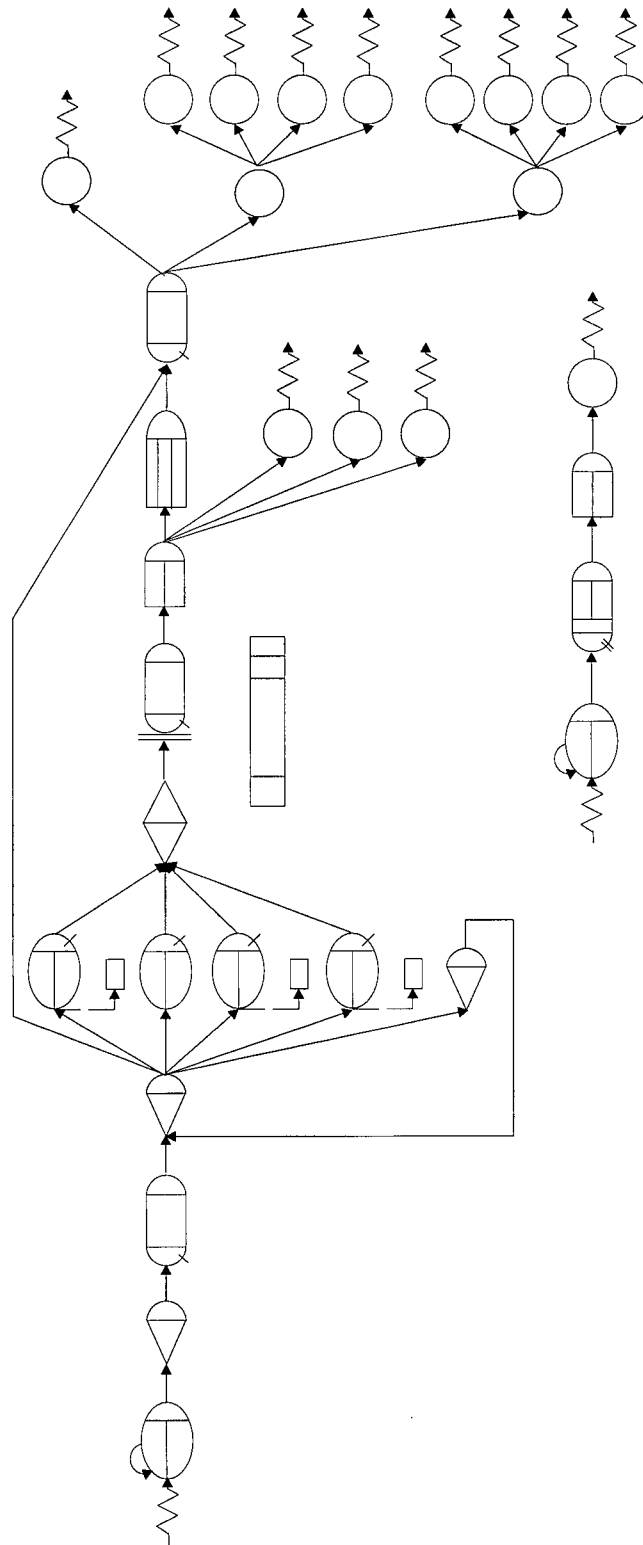


Figure C.3: Complete network employed in the vision/trimmer simulations.

Appendix D

Awesim User Written C Routines

Basecase Coding

BEVENT.c

```

/*****
  This code is called by the event nodes in the basecase scenario.
*****/
#include "vslam.h"
#define rimdef 14.69      /* flange defect \% */
#define split 48.9       /* \% serious flange defects */
#define labelwt 213.0    /* the label weight (g) */
#define weighttol 9.0    /* tolerance on the checkweigher (g)*/
#define weightmean 216.9 /* mean weight of fill (g) */
#define weightdev 5.07   /* std dev of fill weight (g) */
#define xpck 7.845       /* cross-pack \% */
#define poorfill 7.85    /* % poor-fill */
#define numdefects LTRIB[1]
#define underweight LTRIB[2]
#define sflangedefect LTRIB[3] /* serious flange defect, off-line */
#define poorfilldefect LTRIB[4]
#define crosspackdefect LTRIB[5]
#define mflangedefect LTRIB[6] /* minor flange defect, on-the-fly */

void SWFUNC EVENT(int JEVNT, ENTITY * peUser)
{
  double weight, temp;
  double effflange4 = 0.275;
  double effxpack4 = 0.405;
  double effflange3 = 0.13;
  double effxpack3 = 0.02;
  double efffl2 = 0.0;
  double effxp2 = 0.105;
  double effpf2 = 0.0;
  double effpoorfill1 = 0.068;

```

```

    double effxpack1 = 0.107;
    double effflange1 = 0.46;
switch(JEVNT)
{
    case 1:
/*****
    This case is designed to assign the defects to each entity (can)
    entering the system based on the statistical data.
*****/

    peUser->numdefects = 0;      /* number of defects in can */
    peUser->underweight = 0;     /* weight defect */
    peUser->sflangedefect = 0;   /* serious flange defect */
    peUser->mflangedefect = 0;   /* minor flange defect */
    peUser->poorfilldefect = 0; /* poor-fill */
    peUser->crosspackdefect = 0; /* cross-pack */

    weight = RNORM(weightmean,weightdev,1);
    if (weight < (labelwt - weighttol)) {
        peUser->underweight = 1;
    }

    temp = DRAND(2);
    if (temp < (xpck/100.0) ) {
        peUser->crosspackdefect = 1;
    }

    temp = DRAND(3);
    if (temp < (rimdef/100.0)) {
        temp = DRAND(4);
        if (temp > (split/100.0)) {
            peUser->mflangedefect = 1;
        }
        else
            peUser->sflangedefect = 1;
    }

    temp = DRAND(5);
    if (temp < (poorfill/100.0)) {
        peUser->poorfilldefect = 1;
    }

    peUser->numdefects = (peUser->underweight + peUser->mflangedefect
+ peUser->sflangedefect + peUser->poorfilldefect
+ peUser->crosspackdefect);
    break;
    case 2:

```

```

/*****
This case is designed to simulate the scanning of the can line by
worker 4 for flange and cross-pack defects. This worker will not
select a can with a poor-fill defect regardless of whether it has a
cross-pack or a flange defect.
*****/

peUser->STRIB[1] = ("unrecognized");

    if ((peUser->sflangedefect == 1) || (peUser->mflangedefect == 1)) {
        temp = DRAND(6);
        if (temp < effflange4)
            peUser->STRIB[1] = ("recognized");
    }
    if (peUser->crosspackdefect == 1) {
        temp = DRAND(7);
        if (temp < effxpack4)
            peUser->STRIB[1] = ("recognized");
    }
    if (peUser->poorfilldefect == 1) {
        peUser->STRIB[1] = ("unrecognized");
    }

break;
case 3:
/*****
This case is designed to simulate the scanning of the can line by
worker 3 for flange and cross-pack defects. Again poor-fills are not
selected by this worker.
*****/

peUser->STRIB[1] = ("unrecognized");
    if ((peUser->sflangedefect == 1) || (peUser->mflangedefect == 1)){
        temp = DRAND(8);
        if (temp < effflange3)
            peUser->STRIB[1] = ("recognized");
    }
    if (peUser->crosspackdefect == 1) {
        temp = DRAND(9);
        if (temp < effxpack3)
            peUser->STRIB[1] = ("recognized");
    }
    if (peUser->poorfilldefect == 1) {
        peUser->STRIB[1] = ("unrecognized");
    }

break;

```

```

case 4:
/*****
This case is designed to simulate the scanning of the can line by
worker 2 for poor-fills, flange defects, and cross-packs.
*****/

    peUser->STRIB[1] = ("unrecognized");

    if ((peUser->mflangedefect == 1) || (peUser->sflangedefect == 1)) {
        temp = DRAND(10);
        if (temp < efffl2)
            peUser->STRIB[1] = ("recognized");
    }
    if (peUser->poorfilldefect == 1) {
        temp = DRAND(11);
        if (temp < effpf2)
            peUser->STRIB[1] = ("recognized");
    }
    if (peUser->crosspackdefect == 1) {
        temp = DRAND(12);
        if (temp < effxp2)
            peUser->STRIB[1] = ("recognized");
    }

break;
case 5:
/*****
This case simulates the spotter scanning the line for defects and
routes poor-fill defects to worker 2, flange and cross-packs are
routed to worker 3.
*****/

    peUser->STRIB[1] = ("unrecognized");

    if (peUser->sflangedefect == 1) {
        temp = DRAND(13);
        if (temp < effflange1){
            peUser->STRIB[1] = ("recog3");
        }
    }
    if (peUser->crosspackdefect == 1) {
        temp = DRAND(14);
        if (temp < effxpack1){
            peUser->STRIB[1] = ("recog3");
        }
    }
    if (peUser->poorfilldefect == 1) {

```

```

        temp = DRAND(15);
        if (temp < effpoorfill1){
            peUser->STRIB[1] = ("recog2");
        }
    }

break;
case 6:
/*****
this case ensures the spotter patches all minor flange defects.  this
is done to reflect the tamping action carried out by the spotter on
all cans exiting the system resulting in those on-the-fly flange
patches (51.1% of all flange defects) which make it through the table
are fixed by the spotter but with no task time as the spotter does not
interrupt their scanning to do so.
*****/

        if (peUser->mflangedefect == 1)
            peUser->STRIB[1] = ("onthe-fly");
break;
} /* close switch */
}

```

BUSERF.c

```

/*****
This code is called by the USERF calls in the basecase scenario.
*****/
#include "vslam.h"
#include <stdio.h>
#include <math.h>

/* WEIGHT STATISTICS */
#define wghtmean 1.972985 /* LOGNORM mean for weight patch */
#define wghtdev 0.731795

/* CROSS-PACK STATISTICS */
#define axp 10.2 /* this is the % of type a cross-packs */
#define bxp 89.8 /* this is the % of type b cross-packs and
must be changed if a is */
#define axpmean 24.0 /* this is the mean value for the NORMAL dist
used in modelling type (a) patch operation */
#define axpdev 7.171239 /* this is the std dev for the NORMAL dist */
#define bxpmean 1.559038 /* this is the mean value for the LOGNORM dist
used in modelling type (b) patch operation */
#define bxpdev 0.343788 /* this is the std dev for the LOGNORM dist */

```



```

/* FLANGE STATISTICS */

#define olinemean 0.67672    /* LOGNORM mean */
#define olinedev 0.613708   /* LOGNORM std deviation */

/* POOR-FILL STATISTICS */

#define poormean 3.492833    /* mean of the LOGNORMAL dist fit to
                             poor-fill patch times */
#define poordev 0.299851    /* std deviation */

double SWFUNC USERF (int iCode, ENTITY*peUser)
{
    double ptctime = 0.0;
    double temp1,temp2,temp3,temp4,temp5,temp6,a,b,c,maximum;
    double temp;
    switch(iCode)
    {
    case 1:
        /*****
        Function USERF(1) used to calculate the patch time for
        workers patching defects.
        *****/

        temp1 = peUser->LTRIB[2];
        temp2 = peUser->LTRIB[3];
        temp3 = peUser->LTRIB[4];
        temp4 = peUser->LTRIB[5];
        temp5 = DRAND(16);
        temp6 = peUser->LTRIB[6];

        if (temp1 == 1) {
            temp1 = RNORM(wghtmean,wghtdev,17);
            temp1 = exp(temp1);
        }

        if (temp6 == 1) {
            temp6 = 0;
        }

        if (temp2 == 1) {
            temp2 = RNORM(olinemean,olinedev,19);
            temp2 = exp(temp2);
        }

        if (temp3 == 1) {
            temp3 = RNORM(poormean,poordev,20);

```

```

    temp3 =exp(temp3);
}

if (temp4 == 1) {
    if (temp5 > (bxp/100.0)) {          /* then its type a */
        temp5 = RNORM(axpmean,axpdev,21);
    }
    else {
        temp5 = RNORM(bxpmean,bxpdev,22);
        temp5 = exp(temp5);
    }
}
else
    temp5 = 0;

a = (temp1 > temp6) ? temp1 : temp6;
b = (a > temp2) ? a : temp2;
c = (b > temp3) ? b : temp3;
maximum = (c > temp5) ? c : temp5;

ptchtime = maximum;

if (ptchtime < 0.0) {
    ptchtime = 0.0;
}
break;
case 2:
/*****
this case generates the task time for the meat preparation task based
on a monte carlo simulation of the data for worker #2.
*****/

temp = DRAND(23);

if (temp <= 0.083333)
    ptchtime = 2.0;
else if (temp <= 0.166666)
    ptchtime = 4.0;
else if (temp <= 0.25)
    ptchtime = 11.0;
else if (temp <= 0.416666)
    ptchtime = 13.0;
else if (temp <= 0.583333)
    ptchtime = 14.0;
else if (temp <= 0.666666)
    ptchtime = 16.0;
else if (temp <= 0.75)

```

```
    ptchtime = 17.0;
else if (temp <= 0.833333)
    ptchtime = 19.0;
else if (temp <= 0.916666)
    ptchtime = 25.0;
else
    ptchtime = 32.0;
break;

case 3:
/*****
    this case does the same as case 2 only for worker #4.
*****/

temp = DRAND(24);

if (temp <= 0.125)
    ptchtime = 2.0;
else if (temp <= 0.1875)
    ptchtime = 4.0;
else if (temp <= 0.3125)
    ptchtime = 5.0;
else if (temp <= 0.375)
    ptchtime = 6.0;
else if (temp <= 0.4375)
    ptchtime = 8.0;
else if (temp <= 0.5)
    ptchtime = 11.0;
else if (temp <= 0.5625)
    ptchtime = 16.0;
else if (temp <= 0.6875)
    ptchtime = 17.0;
else if (temp <= 0.75)
    ptchtime = 18.0;
else if (temp <= 0.8125)
    ptchtime = 19.0;
else if (temp <= 0.9375)
    ptchtime = 22.0;
else
    ptchtime = 25.0;
break;
}
return (ptchtime);
}
```

Vision Coding

VNQS.c

```

/*****
    this code is called from the select node in the vision scenario.
    the code selects which queue to select from for patching.
*****/
#include "vslam.h"
UINT SWFUNC NQS(int iCode, ENTITY *peUser, SN_NODE *pNode)

{ double a,b,c,largest,temp;
  UINT iReturn;

  switch(iCode)
  {
  case 1: iReturn = 0;
    if ((NNQ(1)==0) && (NNQ(2)==0) && (NNQ(3)==0) && (NNQ(4)==0) &&
        (NNQ(5)==0)) break;

    if (NNQ(2) > 0)
      iReturn = 2;
    else if (NNQ(1) > 0)
      iReturn = 1;
    else {
      if ((NNQ(3) > 0) && (XX[7] == 0)) {
        iReturn = 3;
        XX[7] = 1;
      }
      else if ((NNQ(4) > 0) && (XX[7] == 1)) {
        iReturn = 4;
        XX[7] = 2;
      }
      else if ((NNQ(5) > 0) && (XX[7] == 2)) {
        iReturn = 5;
        XX[7] = 0;
      }
    }
    else{
      a = NNQ(3);
      b = NNQ(4);
      c = NNQ(5);
      temp = (NNQ(3) > NNQ(4)) ? NNQ(3) : NNQ(4);
      largest = (temp > NNQ(5)) ? temp : NNQ(5);
      if (largest == a){
        iReturn = 3;
        XX[7] = 1;
      }
    }
  }
}

```

```

        else if (largest = b){
            iReturn = 4;
            XX[7] = 2;
        }
        else{
            iReturn = 5;
            XX[7] = 0;
        }
    }
}
break;
}
return iReturn;
}

```

VEVENT.c

```

/*****
This code is for the event nodes in the vision scenario
*****/
#include "vslam.h"

#define rimdef 14.69      /* flange defect % */
#define split 48.9       /* % serious flange defects */
#define labelwt 213.0    /* the label weight (g) */
#define weighttol 9.0    /* weight tolerance (g) */
#define weightmean 216.9 /* mean weight of fill (g) */
#define weightdev 5.07   /* std dev of fill weight (g) */
#define xpck 7.845       /* cross-pack % */
#define poorfill 7.85    /* % poor-fill */
#define effflg XX[1]     /* vision system performance */
#define effpfl XX[2]
#define effxpk XX[3]
#define efftype1 XX[4]
#define numdefects LTRIB[1]
#define underweight LTRIB[2]
#define sflangedefect LTRIB[3] /* serious flange defect, off-line */
#define poorfilldefect LTRIB[4]
#define crosspackdefect LTRIB[5]
#define mflangedefect LTRIB[6] /* minor flange defect, on-the-fly */

void SWFUNC EVENT(int JEVNT, ENTITY * peUser)
{
    double weight, temp;
    int vector[3];

    switch(JEVNT)

```

```

{
  case 1:
/*****
  This case is designed to assign the defects to each new can
  entering the system based on the distribution fits.
*****/

  peUser->numdefects = 0;      /* number of defects in can */
  peUser->underweight = 0;     /* weight defect */
  peUser->sflangedefect = 0;   /* serious flange defect */
  peUser->mflangedefect = 0;   /* minor flange defect */
  peUser->poorfilldefect = 0; /* poor-fill */
  peUser->crosspackdefect = 0; /* cross-pack */

/*****
  assigns an underweight defect to the can if the sampling
  distribution indicates a weight of 9 grams or more under the
  can label weight.  LTRIB[2] = 1
*****/
  weight = RNORM(weightmean,weightdev,1);
  if (weight < (labelwt - weighttol)) {
    peUser->underweight = 1;
  }

/*****
  assigns a cross-pack defect "xpck" percent of the time.
  LTRIB[5] = 1
*****/
  temp = DRAND(2);
  if (temp < (xpck/100.0)) {
    peUser->crosspackdefect = 1;
  }

/*****
  assigns a flange defect "rimdef" percent of the time, of which
  "split" percent are serious defects requiring off-line patching
  the remainder are on-the-fly patches.
  LTRIB[3] = 1 or LTRIB[6] = 1
*****/
  temp = DRAND(3);
  if (temp < (rimdef/100.0)) {
    temp = DRAND(4);
    if (temp > (split/100.0)) {
      peUser->mflangedefect = 1;
    }
    else
      peUser->sflangedefect = 1;
  }

```

```

    }
/*****
    assigns a poor-fill defect "poorfill" percent of the time.
    LTRIB[4] = 1
*****/
    temp = DRAND(5);
    if (temp < (poorfill/100.0)) {
        peUser->poorfilldefect = 1;
    }
/*****
    assigns LTRIB[1] as the total number of defects.
*****/
    peUser->numdefects = (peUser->underweight + peUser->mflangeddefect
+ peUser->sflangeddefect + peUser->poorfilldefect
+ peUser->crosspackdefect);
break;

case 2:
/*****
    This code is designed to simulate the operation of the robotic
    vision system and the associated routing of the defects.
*****/
    vector[0] = (peUser->sflangeddefect + peUser->mflangeddefect);
    vector[1] = peUser->poorfilldefect; /* poor-fill */
    vector[2] = peUser->crosspackdefect; /* x-pack */
/*****
    the following section simulates the vision system where eff***
    is the probability of correctly recognizing a *** type defect
    efftype1 is the prob of correctly recognizing a non-defect.
*****/
    temp = DRAND(6);
    if (vector[0] == 1) {
        if (temp > effflg)
            vector[0] = 0; // missclassification type(2) 1->0 //
        else
            vector[0] = 1;
    }
    else {
        if (temp > efftype1)
            vector[0] = 1; // missclassification type(1) 0->1 //
        else
            vector[0] = 0;
    }

    temp = DRAND(7);
    if (vector[1] == 1) {
        if (temp > effpfl)

```

```

        vector[1] = 0; // missclassification type(2) 1->0 //
    else
        vector[1] = 1;
}
else {
    if (temp > efftype1)
        vector[1] = 1; // missclassification type(1) 0->1 //
    else
        vector[1] = 0;
}

temp = DRAND(8);
if (vector[2] == 1) {
    if (temp > effxpk)
        vector[2] = 0; // missclassification type(2) 1->0 //
    else
        vector[2] = 1;
}
else {
    if (temp > efftype1)
        vector[2] = 1; // missclassification type(1) 0->1 //
    else
        vector[2] = 0;
}
/*****
this is the routing section of the code. The logic is such
that the "multi" queue contains those cans with poor-fill and
cross-pack. If the can is underweight or has a flange defect
it goes to the associated queue regardless if it has multiple
defects. In the case of the can having flange defects and
underweight it is sent to the weight queue.
*****/
if (vector[1] == 1)
    peUser->STRIB[1] = ("poor-fill");

if (vector[2] == 1)
    peUser->STRIB[1] = ("xpack");

if ( (vector[1] + vector[2]) == 2 )
    peUser->STRIB[1] = ("multi");

if ((vector[0] + vector[1] + vector[2]) == 0)
    peUser->STRIB[1] = ("noprobdetected");

if (vector[0] == 1)
    peUser->STRIB[1] = ("flange");

```



```

    if (peUser->LTRIB[2] == 1)
        peUser->STRIB[1] = ("underweight");
break;

}    /* close switch */
}

```

VUSERF.c

```

/*****
    This code is called by the USERF calls in the vision scenario.
*****/
#include "vslam.h"
#include <stdio.h>
#include <math.h>

/* Underweight statistics */

#define wghtmean 1.972985    /* LOGNORMAL mean for weight patch */
#define wghtdev 0.731795    /* standard deviation */

/* Cross-pack statistics */

#define axp 10.2            /* this is the % of type a cross-packs */
#define bxp 89.8            /* this is the % of type b cross-packs and
                             must be changed if a is */
#define axpmean 24.0        /* this is the mean value for the NORMAL
                             dist used in modelling type (a) patch
                             operation */
#define axpdev 7.171239     /* this is the std dev for the NORMAL dist
                             */
#define bxpmean 1.559038    /* this is the mean value for the LOGNORMAL
                             dist used in modelling type (b) patch
                             operation */
#define bxpdev 0.343788     /* this is the std dev for the LOGNORMAL
                             dist */

/* Flange statistics */

#define olinemean 0.67672    /* LOGNORMAL mean for off-line patches */
#define olinedev 0.613708    /* LOGNORMAL std dev */

/* POOR-FILL STUFF */

#define poormean 3.492833    /* mean of the LOGNORMAL dist fit to
                             poor-fill patch times */
#define poordev 0.299851     /* std dev */

```

```

/* Other data */

#define onthefly XX[5]      /* patch time for on the fly flange */
#define nodefect XX[6]     /* handling time for type1 errors */

double SWFUNC USERF (int iCode, ENTITY*peUser)
{ double pfpreptime = 0.0;
  double wtpreptime = 0.0;
  double ptchtime = 0.0;
  double temp1, temp2, temp3, temp6;
  double temp, temp5, a, b, c, maximum;

switch(iCode)
{
case 1:
/*****
Function USERF(1) used to calculate the patch time for workers
patching defects. When the can has multiple defects of any kind the
patch time is the maximum of the separate patch times for the
problems. Meat preparation task in this scenario is hard coded and is
based on a percentage of the poor-fills and underweights. The
percentages are the same as in the basecase and use the same monte
carlo simulations for the preparation times.
*****/

  temp5 = DRAND(9);

  if (peUser->LTRIB[2] == 1) {
    temp1 = RNORM(wghtmean,wghtdev,10);
    temp1 = exp(temp1);
  }

  if (peUser->LTRIB[6] == 1) {
    temp6 = onthefly;
  }

  if (peUser->LTRIB[3] == 1) {
    temp2 = RNORM(olinemean,olinedev,11);
    temp2 = exp(temp2);
  }

  if (peUser->LTRIB[4] == 1) {
    temp3 = RNORM(poormean,poordev,12);
    temp3 = exp(temp3);
  }

```

```

if (peUser->LTRIB[5] == 1) {
    if (temp5 > (bxp/100.0)) {          /* then its type a cross-pack*/
        temp5 = RNORM(axpmean,axpdev,13);
    }
    else {
        temp5 = RNORM(bxpmean,bxpdev,14);
        temp5 = exp(temp5);
    }
}
else
    temp5 = 0;

a = (temp1 > temp6) ? temp1 : temp6;
b = (a > temp2) ? a : temp2;
c = (b > temp3) ? b : temp3;
maximum = (c > temp5) ? c : temp5;

ptchtime = maximum;

/*****
this section is the code for the meat preparation tasks and is based
on the data that of the underweights that are patched 81.8% are done
by worker number 4 and of these 62.8% generate a preparation
activity. Worker 2 patches the remaining 18.2% of those
underweights which are patched.
*****/
if (peUser->LTRIB[2] == 1){          /* underweight */
    temp1 = DRAND(15);
    temp2 = DRAND(16);
    if (temp1 <= 0.818){              /* worker 4 */
        if (temp2 <= 0.628){
            temp = DRAND(17);

            if (temp <= 0.083333)
                wtpreptime = 2.0;
            else if (temp <= 0.166666)
                wtpreptime = 4.0;
            else if (temp <= 0.25)
                wtpreptime = 11.0;
            else if (temp <= 0.416666)
                wtpreptime = 13.0;
            else if (temp <= 0.583333)
                wtpreptime = 14.0;
            else if (temp <= 0.666666)
                wtpreptime = 16.0;
            else if (temp <= 0.75)
                wtpreptime = 17.0;

```

```

        else if (temp <= 0.833333)
            wtpreptime = 19.0;
        else if (temp <= 0.916666)
            wtpreptime = 25.0;
        else
            wtpreptime = 32.0;
    }
}

else{                                     /* worker 2 */
    temp2 = DRAND(18);
    temp = DRAND(19);
    if (temp2 <= 0.628){
        if (temp <= 0.125)
            wtpreptime = 2.0;
        else if (temp <= 0.1875)
            wtpreptime = 4.0;
        else if (temp <= 0.3125)
            wtpreptime = 5.0;
        else if (temp <= 0.375)
            wtpreptime = 6.0;
        else if (temp <= 0.4375)
            wtpreptime = 8.0;
        else if (temp <= 0.5)
            wtpreptime = 11.0;
        else if (temp <= 0.5625)
            wtpreptime = 16.0;
        else if (temp <= 0.6875)
            wtpreptime = 17.0;
        else if (temp <= 0.75)
            wtpreptime = 18.0;
        else if (temp <= 0.8125)
            wtpreptime = 19.0;
        else if (temp <= 0.9375)
            wtpreptime = 22.0;
        else
            wtpreptime = 25.0;
    }
}

}

if (peUser->LTRIB[4] == 1){               /* meat preps generated by poor-fills*/
    temp1 = DRAND(20);
    if (temp1 <= 0.125){
        temp = DRAND(21);
        if (temp <= 0.125)
            pfpreptime = 2.0;
    }
}

```

```

        else if (temp <= 0.1875)
            pfpreptime = 4.0;
        else if (temp <= 0.3125)
            pfpreptime = 5.0;
        else if (temp <= 0.375)
            pfpreptime = 6.0;
        else if (temp <= 0.4375)
            pfpreptime = 8.0;
        else if (temp <= 0.5)
            pfpreptime = 11.0;
        else if (temp <= 0.5625)
            pfpreptime = 16.0;
        else if (temp <= 0.6875)
            pfpreptime = 17.0;
        else if (temp <= 0.75)
            pfpreptime = 18.0;
        else if (temp <= 0.8125)
            pfpreptime = 19.0;
        else if (temp <= 0.9375)
            pfpreptime = 22.0;
        else
            pfpreptime = 25.0;
    }
}
if ((pfpreptime != 0.0) && (wtpreptime != 0.0))
    wtpreptime = 0.0;

ptchtime = (ptchtime + pfpreptime + wtpreptime);

if (ptchtime < 0.0) {
    ptchtime = 0.0;
}

if (peUser->LTRIB[1] == 0){
    ptchtime = nodefect;                /* time for a non-defective */
}

break;
} /* close switch */
return (ptchtime);
}

```

Vision/Trimmer Coding

VTEVENT.c

```

/*****

```

```

This code is for the event nodes in the vision/trimmer scenario
*****/
#include "vslam.h"

#define rimdef 14.69      /* flange defect % */
#define split 48.9       /* % serious flange defects */
#define labelwt 213.0    /* the label weight (g) */
#define weighttol 9.0    /* weight tolerance (g)*/
#define weightmean 216.9 /* mean weight of fill (g) */
#define weightdev 5.07   /* std dev of fill weight (g) */
#define xpck 7.845       /* cross-pack % */
#define poorfill 7.85    /* % poor-fill */
#define effflg XX[1]     /* vision system performance */
#define effpfl XX[2]
#define effxpk XX[3]
#define efftype1 XX[4]
#define numdefects LTRIB[1]
#define underweight LTRIB[2]
#define sflangedefect LTRIB[3] /* serious flange defect, off-line */
#define poorfilldefect LTRIB[4]
#define crosspackdefect LTRIB[5]
#define mflangedefect LTRIB[6] /* minor flange defect, on-the-fly */
#define trimeff 0.95        /* trimmer efficiency */

void SWFUNC EVENT(int JEVNT, ENTITY * peUser)
{
    double weight, temp;
    int vector[3];

    switch(JEVNT)
    {
        case 1:
            /*****
            This case is designed to assign the defects to each new can
            entering the system based on the distribution fits.
            *****/

            peUser->numdefects = 0;      /* number of defects in can */
            peUser->underweight = 0;    /* weight defect */
            peUser->sflangedefect = 0;  /* serious flange defect */
            peUser->mflangedefect = 0;  /* minor flange defect */
            peUser->poorfilldefect = 0; /* poor-fill */
            peUser->crosspackdefect = 0; /* cross-pack */

            /*****
            assigns an underweight defect to the can if the sampling
            distribution indicates a weight of 9 grams or more under the

```

```

    can label weight.  LTRIB[2] = 1
    *****/
    weight = RNORM(weightmean,weightdev,1);
    if (weight < (labelwt - weighttol)) {
        peUser->underweight = 1;
    }

    /*****
    assigns a cross-pack defect "xpck" percent of the time.
    LTRIB[5] = 1
    *****/
    temp = DRAND(2);
    if (temp < (xpck/100.0) ) {
        peUser->crosspackdefect = 1;
    }

    /*****
    assigns a flange defect "rimdef" percent of the time, of which
    "split" percent are serious defects requiring off-line patching
    the remainder are on-the-fly patches.
    LTRIB[3] = 1 or LTRIB[6] =1
    *****/
    temp = DRAND(3);
    if (temp < (rimdef/100.0)) {
        temp = DRAND(4);
        if (temp > (split/100.0)) {
            peUser->mflangeddefect = 1;
        }
        else
            peUser->sflangeddefect = 1;
    }

    /*****
    assigns a poor-fill defect "poorfill" percent of the time.
    LTRIB[4] = 1
    *****/
    temp = DRAND(5);
    if (temp < (poorfill/100.0)) {
        peUser->poorfilldefect = 1;
    }

    /*****
    assigns LTRIB[1] as the total number of defects.
    *****/
    peUser->numdefects = (peUser->underweight + peUser->mflangeddefect
    + peUser->sflangeddefect + peUser->poorfilldefect
    + peUser->crosspackdefect);
    break;

```

```

case 2:
/*****
This code is designed to simulate the operation of the robotic
vision system and the associated routing of the defects.
*****/
vector[0] = (peuser->stlangedefect + peuser->mtlangedefect);
vector[1] = peuser->poorfilldefect; /* poor-fill */
vector[2] = peuser->crosspackdefect; /* x-pack */
/*****
the following section simulates the vision system where eff
is the probability of correctly recognizing a ** type defect
efftype1 is the prob of correctly recognizing a non-defect.
*****/
temp = DRAND(6);
if (vector[0] == 1) {
    if (temp > efft1g)
        vector[0] = 0; /* missclassification type(2) 1->0 //
    else
        vector[0] = 1;
}
if (temp > efftype1)
    vector[0] = 1; /* missclassification type(1) 0->1 //
else
    vector[0] = 0;
}
temp = DRAND(7);
if (vector[1] == 1) {
    if (temp > effpfl)
        vector[1] = 0; /* missclassification type(2) 1->0 //
    else
        vector[1] = 1;
}
if (temp > efftype1)
    vector[1] = 1; /* missclassification type(1) 0->1 //
else
    vector[1] = 0;
}
temp = DRAND(8);
if (vector[2] == 1) {
    if (temp > effxpk)
        vector[2] = 0; /* missclassification type(2) 1->0 //
    else
        vector[2] = 1;
}

```



```

    }
    else {
        if (temp > efftype1)
            vector[2] = 1; // missclassification type(1) 0->1 //
        else
            vector[2] = 0;
    }
}
/*****
    this is the routing section of the code. The logic is such
    that the "multi" queue contains those cans with poor-fill and
    cross-pack. If the can is underweight or has a flange defect
    it goes to the associated queue regardless if it has multiple
    defects. In the case of the can having flange defects and
    underweight it is sent to the weight queue.
*****/
if (vector[1] == 1)
    peUser->STRIB[1] = ("poor-fill");

if (vector[2] == 1)
    peUser->STRIB[1] = ("xpack");

if ( (vector[1] + vector[2]) == 2 )
    peUser->STRIB[1] = ("multi");

if ((vector[0] + vector[1] + vector[2]) == 0)
    peUser->STRIB[1] = ("noprobdetected");

if (vector[0] == 1)
    peUser->STRIB[1] = ("flange");

if (peUser->LTRIB[2] == 1)
    peUser->STRIB[1] = ("underweight");
break;

case 3:
/*****
    This case represents the action of the trimmer in repairing the
    flange defects.
*****/
if ((peUser->LTRIB[3] == 1) || (peUser->LTRIB[6] == 1)){
    temp = DRAND(22);
    if (temp < trimeff){
        peUser->LTRIB[3] = 0;
        peUser->LTRIB[6] = 0;
        peUser->LTRIB[1] = peUser->LTRIB[1] - 1;
    }
}
}

```

```

break;
} /* close switch */
}

```

VTNQS.c

```

/*****
    this code is called from the select node in the vision scenario.
    the code selects which queue to select from for patching.
*****/
#include "vslam.h"
UINT SWFUNC NQS(int iCode, ENTITY *peUser, SN_NODE *pNode)

{
    double a,b,c,largest,temp;
    UINT iReturn;

    switch(iCode)
    {
    case 1: iReturn = 0;
        if ((NNQ(1)==0) && (NNQ(2)==0) && (NNQ(3)==0) && (NNQ(4)==0))
            break;

        if (NNQ(1) > 0)
            iReturn = 1;
        else {
            if ((NNQ(2) > 0) && (XX[7] == 0)) {
                iReturn = 2;
                XX[7] = 1;
            }
            else if ((NNQ(3) > 0) && (XX[7] == 1)) {
                iReturn = 3;
                XX[7] = 2;
            }
            else if ((NNQ(4) > 0) && (XX[7] == 2)) {
                iReturn = 4;
                XX[7] = 0;
            }
            else{
                a = NNQ(2);
                b = NNQ(3);
                c = NNQ(4);
                temp = (NNQ(2) > NNQ(3)) ? NNQ(2) : NNQ(3);
                largest = (temp > NNQ(4)) ? temp : NNQ(4);
                if (largest == a){
                    iReturn = 2;
                    XX[7] = 1;
                }
            }
        }
    }
}

```

```

        else if (largest = b){
            iReturn = 3;
            XX[7] = 2;
        }
        else{
            iReturn = 4;
            XX[7] = 0;
        }
    }
}
break;
}
return iReturn;
}

```

VTUSERF.c

```

/*****
This code is called by the USERF calls in the vision/trimmer scenario.
*****/
#include "vslam.h"
#include <stdio.h>
#include <math.h>

/* Underweight statistics */

#define wghtmean 1.972985 /* LOGNORMAL mean for weight patch */
#define wghtdev 0.731795 /* standard deviation */

/* Cross-pack statistics */

#define axp 10.2 /* this is the % of type a cross-packs */
#define bxp 89.8 /* this is the % of type b cross-packs and
must be changed if a is */
#define axpmean 24.0 /* this is the mean value for the NORMAL
dist used in modelling type (a) patch
operation */
#define axpdev 7.171239 /* this is the std dev for the NORMAL dist
*/
#define bxpmean 1.559038 /* this is the mean value for the LOGNORMAL
dist used in modelling type (b) patch
operation */
#define bxpdev 0.343788 /* this is the std dev for the LOGNORMAL
dist */

/* Flange statistics */

```

```

#define olinemean 0.67672    /* LOGNORMAL mean for off-line patches */
#define olinedev 0.613708   /* LOGNORMAL std dev */

/* POOR-FILL STUFF */

#define poormean 3.492833    /* mean of the LOGNORMAL dist fit to
                             poor-fill patch times */
#define poordev 0.299851    /* std dev */

/* Other data */

#define onthefly XX[5]      /* patch time for on the fly flange */
#define nodefect XX[6]     /* handling time for type1 errors */

double SWFUNC USERF (int iCode, ENTITY*peUser)
{ double pfpreptime = 0.0;
  double wtpreptime = 0.0;
  double ptchtime = 0.0;
  double temp1, temp2, temp3, temp6;
  double temp, temp5, a, b, c, maximum;

switch(iCode)
{
case 1:
/*****
Function USERF(1) used to calculate the patch time for workers
patching defects. When the can has multiple defects of any kind the
patch time is the maximum of the separate patch times for the
problems. Meat preparation task in this scenario is hard coded and is
based on a percentage of the poor-fills and underweights. The
percentages are the same as in the basecase and use the same monte
carlo simulations for the preparation times.
*****/

  temp5 = DRAND(9);

  if (peUser->LTRIB[2] == 1) {
    temp1 = RNORM(wghtmean,wghtdev,10);
    temp1 = exp(temp1);
  }

  if (peUser->LTRIB[6] == 1) {
    temp6 = onthefly;
  }

  if (peUser->LTRIB[3] == 1) {
    temp2 = RNORM(olinemean,olinedev,11);

```

```

        temp2 = exp(temp2);
    }

    if (peUser->LTRIB[4] == 1) {
        temp3 = RNORM(poormean,poordev,12);
        temp3 = exp(temp3);
    }

    if (peUser->LTRIB[5] == 1) {
        if (temp5 > (bxp/100.0)) { /* then its type a cross-pack*/
            temp5 = RNORM(axpmean,axpdev,13);
        }
        else {
            temp5 = RNORM(bxpmean,bxpdev,14);
            temp5 = exp(temp5);
        }
    }
    else
        temp5 = 0;

    a = (temp1 > temp6) ? temp1 : temp6;
    b = (a > temp2) ? a : temp2;
    c = (b > temp3) ? b : temp3;
    maximum = (c > temp5) ? c : temp5;

    ptchtime = maximum;

/*****
    this section is the code for the meat preparation tasks and is based
    on the data that of the underweights that are patched 81.8% are done
    by worker number 4 and of these 62.8% generate a preparation
    activity. Worker 2 patches the remaining 18.2% of those
    underweights which are patched.
*****/
if (peUser->LTRIB[2] == 1){ /* underweight */
    temp1 = DRAND(15);
    temp2 = DRAND(16);
    if (temp1 <= 0.818){ /* worker 4 */
        if (temp2 <= 0.628){
            temp = DRAND(17);

            if (temp <= 0.083333)
                wtpreptime = 2.0;
            else if (temp <= 0.166666)
                wtpreptime = 4.0;
            else if (temp <= 0.25)
                wtpreptime = 11.0;
        }
    }
}

```

```
        else if (temp <= 0.416666)
            wtpreptime = 13.0;
        else if (temp <= 0.583333)
            wtpreptime = 14.0;
        else if (temp <= 0.666666)
            wtpreptime = 16.0;
        else if (temp <= 0.75)
            wtpreptime = 17.0;
        else if (temp <= 0.833333)
            wtpreptime = 19.0;
        else if (temp <= 0.916666)
            wtpreptime = 25.0;
        else
            wtpreptime = 32.0;
    }
}

else{                                     /* worker 2 */
    temp2 = DRAND(18);
    temp = DRAND(19);
    if (temp2 <= 0.628){
        if (temp <= 0.125)
            wtpreptime = 2.0;
        else if (temp <= 0.1875)
            wtpreptime = 4.0;
        else if (temp <= 0.3125)
            wtpreptime = 5.0;
        else if (temp <= 0.375)
            wtpreptime = 6.0;
        else if (temp <= 0.4375)
            wtpreptime = 8.0;
        else if (temp <= 0.5)
            wtpreptime = 11.0;
        else if (temp <= 0.5625)
            wtpreptime = 16.0;
        else if (temp <= 0.6875)
            wtpreptime = 17.0;
        else if (temp <= 0.75)
            wtpreptime = 18.0;
        else if (temp <= 0.8125)
            wtpreptime = 19.0;
        else if (temp <= 0.9375)
            wtpreptime = 22.0;
        else
            wtpreptime = 25.0;
    }
}
```

```

}
if (peUser->LTRIB[4] == 1){      /* meat preps generated by poor-fills*/
    temp1 = DRAND(20);
    if (temp1 <= 0.125){
        temp = DRAND(21);
        if (temp <= 0.125)
            pfpreptime = 2.0;
        else if (temp <= 0.1875)
            pfpreptime = 4.0;
        else if (temp <= 0.3125)
            pfpreptime = 5.0;
        else if (temp <= 0.375)
            pfpreptime = 6.0;
        else if (temp <= 0.4375)
            pfpreptime = 8.0;
        else if (temp <= 0.5)
            pfpreptime = 11.0;
        else if (temp <= 0.5625)
            pfpreptime = 16.0;
        else if (temp <= 0.6875)
            pfpreptime = 17.0;
        else if (temp <= 0.75)
            pfpreptime = 18.0;
        else if (temp <= 0.8125)
            pfpreptime = 19.0;
        else if (temp <= 0.9375)
            pfpreptime = 22.0;
        else
            pfpreptime = 25.0;
    }
}
if ((pfpreptime != 0.0) && (wtpreptime != 0.0))
    wtpreptime = 0.0;

ptchtime = (ptchtime + pfpreptime + wtpreptime);

if (ptchtime < 0.0) {
    ptchtime = 0.0;
}

if (peUser->LTRIB[1] == 0){
    ptchtime = nodefect;          /* time for a non-defective */
}
break;
} /* close switch */
return (ptchtime);
}

```

Appendix E

Screening Program Details

The canning industry suffered a tremendous setback in July of 1978. Four people from Birmingham England contracted botulism poisoning after eating canned Alaskan red salmon. Two of these people subsequently died. In February of 1982 two people in Belgium were stricken by botulism poisoning attributed to consumption of Alaskan canned salmon, and one person died. Both of these incidents proved exceptionally costly to the industry both through necessary recalls, enforced hand inspection and loss of sales and reputation. In both instances a failure in the hermetical seal was the cause of the poisoning. In order to deal with this issue, the industry adopted a quality assurance program known as, 'the can screening program'. The purpose of this program is to reduce the possibility of a faulty seal reaching market.

The screening program employs a checkweighing machine in-line with a double-dud-detector, (DDD). The DDD is a device which measures the deflection of the lid of the salmon can. This deflection is an indication of the level of vacuum in the can which is a good indicator of seal integrity. The purpose of the checkweigher preceding the DDD is to eject underweight and overweight cans. Underweight cans are ejected since during retorting or cooking of the cans when they are under high internal pressure occasionally a leak will occur through which material is leaked. Subsequently at some point the leak can be plugged by the cooking of a protein into the gap and upon cooling the can will re-vacuum. Thus although the can would appear to the DDD as properly sealed, it may still be defective. Overweight cans are ejected since these cans might be ejected by the DDD not due to inadequate vacuum but simply because the can is too full to allow significant lid deflection. The DDD is then tuned such that those cans ejected make up the two percent of the population with the least lid deflection. This 2% is then hand inspected for seam defects. If more than 25/100,000 are found to have serious seam defects the entire batch must be hand inspected or "culled".

Bibliography

- [1] British Columbia export statistics, 1989–1995.
- [2] British Columbia Seafood Industry year in review., 1990–1994.
- [3] Cary M. Huettnner and Harold J. Steudel. Analysis of a manufacturing system via spreadsheet analysis, rapid modelling, and manufacturing simulation. *International Journal of Production Research*, 30(7):1699–1714, 1992.
- [4] G. Righini. Modular petri nets for simulation of flexible production systems. *International Journal of Production Research*, 31(10):2463–2477, 1993.
- [5] Sheng-Hsien (Gary) Teng and Jie Zhang. A petri net-based decomposition approach in modeling of manufacturing systems. *International Journal of Production Research*, 31(6):1423–1439, 1993.
- [6] Chu-Chai Chan and Hsu-Pin (Ben) Wang. Design and development of a stoichastic high-level petri net system for FMS performance evaluation. *International Journal of Production Research*, 31(10):2415–2439, 1993.
- [7] Anil Lahoti and William J. Kennedy. Analyzing the effect of installing diagnostic equipment in a manufacturing cell. In *Proceedings Annual Reliability and Maintainability Symposium*, pages 10–14, 1991.

- [8] Robert P. Davis and W. J. Kennedy. Markovian modeling of manufacturing systems. *International Journal of Production Research*, 25(3):337–351, 1987.
- [9] Marino Widmer and Philippe Solot. Do not forget the breakdowns and the maintenance operations in FMS design problems! *International Journal of Production Research*, 28(2):421–430, 1990.
- [10] Bertil A. Brandin. The real-time supervisory control of an experimental manufacturing cell. *IEEE Transactions on Robotics and Automation*, 12(1):1–14, February 1996.
- [11] W. M. Wonham B. A. Brandin and B. Benhabib. Manufacturing cell supervisory control - a timed discrete event system approach. In *Proceedings of the 1992 IEEE International Conference on Robotics and Automation*, pages 931–936, May 1992.
- [12] Charles R. Harrell. Modeling beverage processing using discrete event simulation. In *Proceedings of the 1993 Winter Simulation Conference*, pages 845–850, 1993.
- [13] Enrique Rotstein José A. Bandoni and José A. Romagnoli. Simulation modelling and optimal operation of an apple juice concentrate plant. *Journal of Food Engineering*, 7:159–175, 1988.
- [14] J. H. Wells F. A. Escobar and A. M. Waikar. Simulation based performance analysis of crawfish processing operations. *Journal of Food Process Engineering*, 14:147–162, 1991.
- [15] A. K. Kharwat. Computer simulation: An important tool in the fast-food industry. In *Proceedings of the Winter Simulation Conference*, pages 811–815, 1991.
- [16] L. Guyon A. Bofill, J. A. Diaz and R. Martinez. Application of mathematical methods to the management of sugar cane enterprises. *Computer Applications in food production and agricultural engineering.*, pages 175–190, 1982.

- [17] D. DuBois and K. Stecke. Using petri nets to represent production processes. In *Proceedings of the 22nd IEEE Conference on Decision & Control*, December 1983.
- [18] Jean J. O'Reilly A. Alan B. Pritsker and David K. LaVal. *Simulation with Visual SLAM and AweSim*. John Wiley & Sons, New York, 1997.
- [19] R. D. Phillips. Material handling requirements for a distribution center. In *Proceedings of the Winter Simulation Conference*, pages 396–401, 1991.
- [20] Kripa Shanker and Anil Kumar Agrawal. Loading problem and resource considerations in FMS: a review. *International Journal of Production Economics*, 25:111–119, 1991.
- [21] Peter Fryer. Mathematical models in food processing. *Chemistry & Industry, (Great Britain)*, pages 515–518, July 1994.
- [22] R. P. Singh and J. R. Banga. Recent advances in food process optimization. *Chemistry and Industry, Great Britain*, pages 511–514, July 1994.
- [23] E. D. Megaw. Some factors affecting inspection accuracy. Paper presented to Symposium on Ergonomics and Visual Inspection, Birmingham.
- [24] E. D. Megaw. Factors affecting visual inspection accuracy. *Applied Ergonomics*, March 1979.
- [25] D. R. Davies and R. Parasurman. *The Psychology of Vigilance*. Academic Press, London, 1982.
- [26] S. J. Czaja and C. G. Drury. Training programs for inspection. *Human Factors*, 23:473–484, 1981.
- [27] David J. Osborne. *Ergonomics at Work: Human Factors in Design and Development*. John Wiley & Sons, Chichester, 3rd edition, 1995.

-
- [28] K. F. H. Murrell. *Human Performance in Industry*. Reinhold Publishing Corp., Great Britain, 1965.
- [29] D. E. Broadbent and M. Gregory. Vigilance considered as a statistical decision. *British Journal of Psychology*, 54:309-323, 1963.
- [30] A. Craig and W. P. Colquhoun. Vigilance effects in complex inspection. In Robert R. Mackie, editor, *Vigilance II: Relationships Among Theory, Physiological Correlates, and Operational Performance.*, pages 239-262, New York, 1977. Plenum.
- [31] J. Deese. Some problems in the theory of vigilance. *Psychological Review*, 62:359-368, 1955.
- [32] W. H. Teichner. The detection of a simple visual signal as a function of time on watch. *Human Factors*, 16:339-353, 1974.
- [33] Joel S. Warm, editor. *Sustained Attention in Human Performance: Studies in Human Performance*. John Wiley & Sons, Chichester, 1984.
- [34] A. W. Kibler. The relevance of vigilance research to aerospace monitoring tasks. *Human Factors*, 7:93-99, 1965.
- [35] A. Chapanis. The relevance of laboratory studies to practical situations. *Ergonomics*, 10:557-577, 1967.
- [36] R. L. Smith and L. F. Lucaccini. Vigilance research: its application to industrial problems. *Human Factors*, 11:149-156, 1969.
- [37] Glynn D. Coates Earl A. Alluisi and Ben B. Morgan, Jr. Effects of temporal stressors on vigilance and information processing. In Robert R. Mackie, editor, *Vigilance II: Relationships*

- Among Theory, Physiological Correlates, and Operational Performance.*, pages 361–421, New York, 1977. Plenum.
- [38] D. E. Chapman and M. A. Sinclair. Ergonomics in inspection tasks in the food industry. In C. G. Drury and J. G. Fox, editors, *Human Reliability in Quality Control*. Taylor & Francis, London, 1974.
- [39] Joel S. Warm. Psychological processes in sustained attention. In *Vigilance II: Relationships Among Theory, Physiological Correlates, and Operational Performance.*, pages 623–644, 1977.
- [40] R. M. Pickett H. J. Jerison and H. H. Stenson. The elicited observing rate and decision processes in vigilance. *Human Factors*, 7:107–128, 1965.
- [41] J. F. Mackworth. Deterioration of signal detectability during a vigilance task as a function of background event rate. *Psychonomic Science*, 3:421–422, 1965.
- [42] W. C. Howell W. A. Johnston and I. L. Goldstein. Human vigilance as a function of signal frequency and stimulus density. *Journal of Experimental Psychology*, 72:736–743, 1966.
- [43] H. J. Jerison. Activation and long term performance. *Acta Psychologica*, 27:373–389, 1967.
- [44] M. J. Guralnick. Effects of event rate and signal difficulty on observing responses and decision measures in vigilance. *Journal of Experimental Psychology*, 99:261–265, 1973.
- [45] R. Parasurman and D. R. Davies. Decision theory analysis of response latencies in vigilance. *Journal of Experimental Psychology: Human Perception and Performance*, 2:578–590, 1976.
- [46] R. Parasurman. Memory load and event rate control sensitivity decrements in sustained attention. *Science*, 205:924–927, 1979.
- [47] H. J. Jerison. Human and animal vigilance. *Perceptual and Motor Skills*, 21:580–582, 1965.

- [48] M. Loeb and J. R. Binford. Variation in performance on auditory and visual monitoring tasks as a function of signal and stimulus frequencies. *Perception and Psychophysics*, 4:361–367, 1968.
- [49] H. A. Taub and F. H. Osborne. Effects of signal and stimulus rates on vigilance performance. *Journal of Applied Psychology*, 52:133–138, 1968.
- [50] Leo A. Smith and James W. Barany. An elementary model of human performance on paced visual inspection tasks. *AIIE Transactions*, 2(4), December 1970.
- [51] C. G. Drury. The effect of speed of working on industrial inspection accuracy. *Applied Ergonomics*, 4(1):2–7, 1973.
- [52] L. G. Williams and M. S. Borrow. The effect of rate and direction of display movement upon visual search. *Human Factors*, 5:139–146, 1963.
- [53] Douglas H. Harris. The nature of industrial inspection. *Human Factors*, 11(2):139–148, 1969.
- [54] Edwin Mansfield. Technical management notes: New evidence on the economic effects and diffusion of FMS. *IEEE Transactions on Engineering Management*, 40(1):76–78, February 1993.
- [55] Alexandra Weigl. Requirements for robot assisted disassembly of not appropriately designed electronic products: Lessons from first studies. In *Proceedings of the 1994 IEEE International Symposium on Electronics and the Environment*, pages 337–342, may 1994.
- [56] David W. Bennett and Mark S. Evans. Development of a semi-automated workcell for repair of printed circuit boards. In *IEEE International Conference on Systems Engineering*, pages 446–448, 1990.

-
- [57] Young K. Son. A cost estimation model for advanced manufacturing systems. *International Journal of Production Research*, 29(3):441–452, 1991.
- [58] Michael Corbey. Measurable economic consequences of investments in flexible capacity. *International Journal of Production Economics*, 23:47–57, 1991.
- [59] Robert L. Horst. Factory automation economic justification by Taguchi method: Automotive example. In *Conference Record of the 1991 IEEE Industrial Applications Society Annual Meeting*, volume 2, pages 1367–1373, 1991.
- [60] Roger N. Wabalickis. Justification of FMS with the analytic hierarchy process. *Journal of Manufacturing Systems*, 7(3), 1988.
- [61] Zone-Ching Lin and C.B. Yang. Economic evaluation of the planning of a flexible manufacturing system. *International Journal of Computer Applications in Technology*, 8(1/2):21–29, 1995.
- [62] Ira H. Kleinfeld. *Engineering Economics - Analysis for Evaluation of Alternatives*. Van Nostrand Reinhold, New York, New York, 1993.
- [63] Alfred L. Kahl James L. Riggs, William F. Rentz and Thomas M. West. *Engineering Economics*. McGraw-Hill Ryerson Limited, Toronto, ON, first canadian edition, 1986.
- [64] Donald G. Newman. *Engineering Economic Analysis*. Engineering Press, Inc., San Jose, California, third edition, 1988.
- [65] E.A. Croft and M.A. O'Dor. Identifying salmon can-filling defects using machine vision. To be presented at the World Automation Conference in Anchorage Alaska, May 1988.
- [66] Can salmon screening line: Theory and operation workshop., 1997.

-
- [67] M.A.O Dor B.D.Allin and E.A.Croft. Industrial automation for a fish processing quality assurance work-cell. In *International Conference on Advanced Intelligent Mechatronics 97*, Tokyo, Japan, June 1997.
- [68] 1997. Personal communication with H. Frech of B.C. Packers.
- [69] B.D. Allin. Bc packers proprietary documentation., 1998. Proprietary information not available to the public.
- [70] Richard L. Scheaffer and James T. McClave. *Probability and Statistics for Engineers*. PWS-KENT, Boston, Massachusettes, third edition, 1990.
- [71] Ronald E. Walpole and Raymond H. Myers. *Probability and Statistics for Engineers and Scientists*. Macmilan, New York, fifth edition, 1993.