MODELLING AND OPTIMISATION OF TURKISH ARMY 5TH LEVEL RENOVATION MAINTENANCE SYSTEM VIA SIMULATION

A THESIS
SUBMITTED TO THE DEPARTMENT OF INDUSTRIAL ENGINEERING
AND THE INSTITUTE OF ENGINEERING AND SCIENCES
OF BILKENT UNIVERSITY
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
MASTER OF SCIENCE

By
Reşat Ali TÜTÜNCÜOĞLU

JULY, 2000

MODELLING AND OPTIMISATION OF TURKISH ARMY 5^{TH} LEVEL RENOVATION MAINTENANCE SYSTEM VIA SIMULATION

A THESIS
SUBMITTED TO THE DEPARTMENT OF INDUSTRIAL ENGINEERING AND THE INSTITUTE OF ENGINEERING AND SCIENCE OF BILKENT UNIVERSITY
IN PARTIAL FULFILMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
MASTER OF SCIENCE

By

Reşat Ali TÜTÜNCÜOĞLU

JULY, 2000

I certify that I have read this thesis and in my opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

Assoc. Prof. İhsan Sabuncuoğlu (Principal Advisor)

I certify that I have read this thesis and in my opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

Assoc. Prof. Selim Aktürk

I certify that I have read this thesis and in my opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

Assoc. Prof. Erdal Erel

Approved for the Institute of Engineering and Science

Prof. Mehmet Baray

Director of Institute Of Engineering and Science

ABSTRACT

MODELLING AND OPTIMISATION OF TURKISH ARMY 5TH LEVEL RENOVATION MAINTENANCE SYSTEM VIA SIMULATION

Reşat Ali Tütüncüoğlu
M.S. in Industrial Engineering
Advisor: Assoc. Prof. İhsan Sabuncuoğlu

Logistics is the application of time and space factors to war. It is the economics of warfare, and it comprises, in the broadest sense, the three big M's of warfare; material, movement, and maintenance. This thesis employing the simulation tool as an effective vehicle for defining the path from competitive concepts to real word solutions, modelling Turkish Army's 5th Level Renovation System and bringing up ways of optimisation. Steady state performances of the renovation unit are measured. Different types of configurations are tested and their advantages and disadvantages are discussed.

Keywords: Simulation, Optimisation, and Throughput.

ABSTRACT

MODELLING AND OPTIMISATION OF TURKISH ARMY 5TH LEVEL RENOVATION MAINTENANCE SYSTEM VIA SIMULATION

Reşat Ali Tütüncüoğlu
M.S. in Industrial Engineering
Advisor: Assoc. Prof. İhsan Sabuncuoğlu

Logistics is the application of time and space factors to war. It is the economics of warfare, and it comprises, in the broadest sense, the three big M`s of warfare; material, movement, and maintenance. This thesis employing the simulation tool as an effective vehicle for defining the path from competitive concepts to real word solutions, modelling Turkish Army`s 5th Level Renovation System and bringing up ways of optimisation. Steady state performances of the renovation unit are measured. Different types of configurations are tested and their advantages and disadvantages are discussed.

Keywords: Simulation, Optimisation, and Throughput.

ÖZET

SİMULASYON KULLANARAK TÜRK KARA KUVVETLERİ 5NCİ KADEME YENİLEŞTİRME BAKIM SİSTEMİNİN MODELLENMESİ VE OPTİMİZASYONU

Reşat Ali Tütüncüoğlu
Endüstri Mühendisliği Bölümü Yüksek Lisans
Danışman: Doç. İhsan Sabuncuoğlu

Logistik zaman ve alan faktörlerinin savaşa uygulanma şekli, savaşın ekonomisi ve tamamlayıcısıdır. Daha geniş bir ifadeyle, savaşta üç önemli faktör olan; ikmal malzemesi, hareket imkan kabiliyeti ve bakımın bütünüdür. Bu tez calışması simulasyonu etkili bir araç olarak kullanarak rekabet sağlayıcı konseptleri gerçek hayata taşımak için Türk Kara Kuvvetleri 5nci kademe yenileştirme sistemlerinin modellenmesini ve optimizasyon yollarının gösterilmesini ifa etmektedir. Bu calışmada yenileştirme ünitesinin sabit dönem performansları belirlenmiş ve değişik tipteki konfigürasyonlar test edilerek avantaj ve dezavantajları tartışılmıştır.

Anahtar Sözcükler: Simulasyon, Optimizasyon, ve Çıktı.

To my wife and parents

ACKNOWLEDGEMENTS

I would like to express my deep gratitude to Dr. İhsan Sabuncuoğlu for his guidance, attention, understanding, and patience throughout all this work.

I am indebted to the readers Selim Aktürk and Erdal Erer for their effort, kindness, and time.

I cannot fully express my gratitude and thanks to my uncle, aunt, and friends for their care, support and encouragement.

Resat Ali TUTUNCUOGLU

CONTENTS

List of Figures

List of Tables

Glossary

MAINTENANCE: It is the function of sustaining material and facilities in an operational status, restoring them to a serviceable condition or upgrading their functional utility through modification.

These facilities are cleaning, control, testing, lubrication, readiness for duty, adjusting, squeezing, repair, rectification, and renovation.

REPAIR: The workmanship to change the condition of the breakdown army materials to a healthy state.

CHANGING: Exchanging the breakdown army material with a new or renewed material.

RENOVATION: The process for testing and defining the performance of the army goods and by repairing or renewing, the army goods became as a new good.

RECTIFICATION: To add some new properties.

MILITARY WORDS' TURKISH MEANINGS

Army: Ordu, involves approximately 9 brigades. Its Commander is full-general.

Corps: Kolordu, involves approximately 3 brigades. Its commander is lieutenant general.

Brigade: Tugay, involves approximately 3 battalion task forces and 6000 soldiers. Its commander is brigadier general.

Battalion: Tabur, involves approximately 3 company teams. Its commander is lieutenant colonel.

Company: Bölük, involves approximately 4 platoons. Its commander is captain.

Platoon: Takım, involves approximately 50 persons. Its commander is first lieutenant or second lieutenant.

Ordnance Company: Ordudonatım Bölüğü, Its commander is a captain.

Chapter 1

INTRODUCTION

In this chapter, first a brief information is given about the army logistics and the problem undertaken in this thesis.

1.1. The Army Logistics

1.1.1. General

Logistics is the art and science of creating and maintaining a military capability. Its purpose is to create weapons and forces and then provide sustained support to these weapons and forces in combat.

A wide range of multi layered, external forces influence army logistics. Within the realm of military activities, logistics is the bound to strategy and tactics. Military activities function in an environment, which is driven by national objectives and policies and shaped primarily by socioeconomic and political factors [9].

Logistics contributions encompass the means to equip and sustain the army in its role to support Republic of Turkey's national policy and military strategy.

1.1.2. The Turkish Army Logistics Mission

The basic mission of the Army Logistics is to support the soldier in the field with what is needed, where, when, and in the condition and quantity required at minimum expenditure of resources [7].

1.1.3. The Logistics Processes

Requirement Determination: The process is the statement of need, together with the definition of the resources necessary to accomplish the stated need.

Acquisition: The translation of the need from requirement to terms suitable for acquisition. The obtaining of what is needed by leasing, buying, recruiting, and constructing. Acceptance and compensation for value received.

Distribution: This process involves all logistical aspects of moving, receiving, storing, handling, and issuing material into the Army supply system.

Maintenance: It is the function of sustaining material and facilities in an operational status, restoring them to a serviceable condition: or upgrading their functional utility through modification. In Section 1.5, a detailed information is given.

1. Direct Maintenance Support. Maintenance performed on material while it remains under the custody of the using military commands. Upon restoration to serviceable condition, the material is normally returned directly to service.

2. Indirect Maintenance Support. Maintenance performed on material after its withdrawal from the custody of the using military command. Upon restoration to serviceable condition, the material is returned to stock for reissue or returned

directly to the user under conditions authorised by the military department concerned.

Disposal: This process involves the purging of excess, obsolete, or surplus material, supplies and real property; making such items available to other prospective users; and effecting maximum possible recovery of value of items [8].

1.1.4. Logistics principles

Logistics principles can be listed as follows:

-Promote combat efficiency of the armed services as a whole by prevention of unnecessary duplication of facilities, services, supplies, and equipment.

- Design logistics systems for expansion to meet peak loads they will face in an emergency.

- Be responsive to operational and technical requirements of commanders.

- Avoid depriving operational units of essential support.

- Provide for administrative control by one service where facilities are used jointly.

- Provide for operational control of personnel [16].

1.1.5. Maintenance System in Turkish Army

In the Turkish Army, maintenance system can be classified into four general levels.

1. Unit Maintenance: Unit maintenance is the maintenance for which the using organisation is responsible and it is performed on assigned equipment. The phases normally consist of inspecting, servicing, lubricating, adjusting, and replacing parts, minor assemblies, and sub-assemblies [15]. This level of maintenance is done by two sub-levels:

-1st Level Maintenance Unit: The user of the army good, operator of vehicle, or crew of the gun system and vehicle, do it. It is called also as preventive maintenance. Everybody is responsible from the 1st level maintenance.

-2nd Level Maintenance Unit: Special trained technical personnel do this level. In this level, there are special equipment, additional parts, and measurement devices for doing jobs that could not be done at 1st level. In every battalion, there is 2nd level 2nd section maintenance service, and every company has 2nd level 1st section maintenance group.

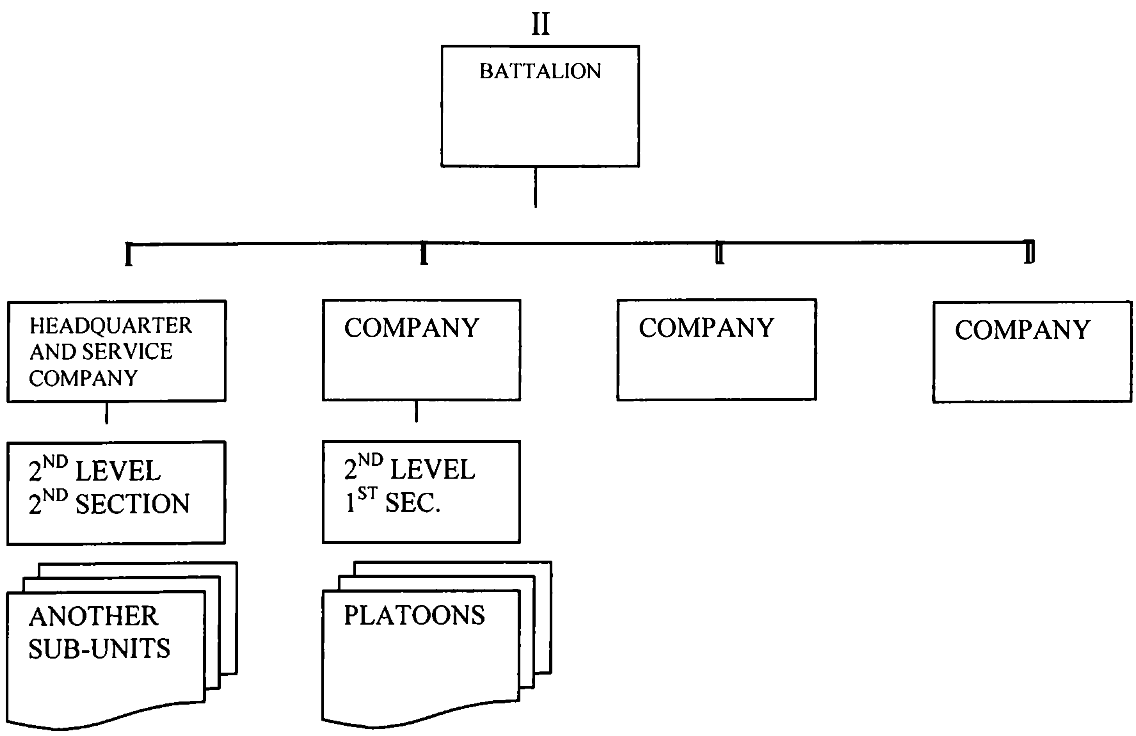


Figure 1-1. 2nd Level Maintenance Sections

2. Direct Support Maintenance (DS): Direct Support (DS) maintenance is performed in support of the user. It is characterised by forward orientation, repair by replacement, and provides mobile, responsive “one-stop" maintenance support. Direct support maintenance is done by ,

-3rd Level Maintenance Units: With trained personnel, direct support union does it. This level of maintenance unions has equipment and sets for Union Maintenance, measurement devices. Moreover, they make more detailed maintenance than 2nd level maintenance. These units have also 100% mobile capacity. In every brigade, there is an Ordnance company and it is called the 3rd level [15].

3. General Support Maintenance (GS): General Support maintenance is performed in support of the theater (battlefield area) supply system. It is characterised by semi-fixed facilities with job or production line operations. General support maintenance is done by:

-4th Level Maintenance Units: These units are located at echelons above corps, and repair of class VII and class IX items. Generally every division and army corps has 4th level Maintenance Union. In total, there are twelve 4th level maintenance unions in the Army [18].

4. Depot Maintenance: This level of maintenance is the responsibility of the Army Material Command (AMC). It is performed by organic Army depots and commercial contractors. Depot maintenance augments depot stocks of serviceable material and supports unit and intermediate maintenance activities by using more extensive shop facilities, and personnel of higher technical skill than are available at lower levels of maintenance. Tasks in this level normally consist of the following: inspection and testing; modification; analytical; calibration; overhaul; and fabrication of items not supported by the supply system in support of national maintenance point (NMP) requirements. This level of maintenance is oriented toward support of the supply system at both theater and national levels. Organisations are fixed or semi-fixed. Maintenance at this category will be primarily production line oriented and will be performed by selected commodity oriented organisations. Depot maintenance is done by:

-5th Level Maintenance Units: This level includes overhaul, rebuild, modification, calibration, analytical, special and non-destructive testing/inspection cannibalisation, and fabrication of items not supported by the supply system. Normally, this level increases the stock rates by manufacturing additional and renewed parts. In addition, this level makes some special technical calibrations. [17]

1.2. Simulating of the 5th Level Maintenance System Renovation Units in Turkish Army

Computer simulation has been widely used tool for studying the dynamics of the real world systems to see its behaviour in response to the changes in the environment. The application areas cover a wide range, especially the cases where the system to be analysed is too complex to be modelled and studied analytically. Production and logistics simulations include those applications that assist in determinations of logistics requirements, system productivity assessments, and industrial base appraisals. These simulations support the Army's procurement, transportation, and maintenance of personnel, material, and facilities.

In the Turkish Army, there are five depots at 5th Level Maintenance for Ordonnance Goods. These are:

1009 Main Repair Depot in KAYSERI for Tanks (Tracked Vehicles)
1010 Main Repair Depot in ARIFIYE for Tanks.
1011 Main Repair Depot in ANKARA for gasoline operated Wheeled V.
1012 Main Repair Depot in BALIKESIR for diesel operated Wheeled V.

1013 Main Repair Depot in TUZLA for small type Vehicles.

1011 Main Repair Depot is one of the most important depots of the Turkish Army that supports all of the Army Units. 1011 Main repair depot has five subdivisions (See *Figure 1-2*).

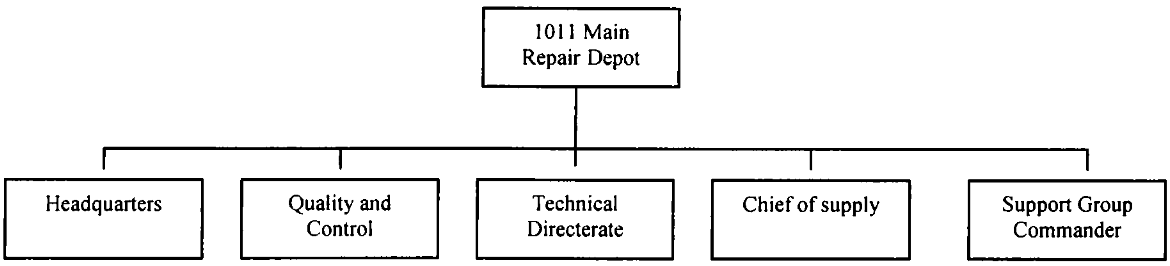


Figure 1-2. 1011 Main Repair Depot.

According to the orders of the Turkish Army Headquarters, the production and production planning for the facilities are done at the Technical Directorship. There are eight main units. These units are mostly production line oriented.

These units are motor renovation unit (Wheeled vehicles), drive-train renovation unit, arms renovation unit, 1st spare part manufacturing unit (Hard materials), 2nd Spare part manufacturing unit (Plastics and tire spare parts), battery manufacturing unit, tire renovation unit, paint manufacture unit.

As seen in the above, the depots (military factory) have many departments and it is difficult to collect all data and constitute the simulation model. In this study, computer simulation is used to analyse the largest department, called renovation unit. In addition, this department has the same characteristics with other depots. By simulating and analysing the renovation unit, we will try to understand the general problem areas and the possible solutions.

The renovation unit can be classified as a flow shop (Fm). There are m machines in series. Each job has to be processed on each one of the m machines. All jobs have the same routing, that is, they have to be processed on machine 1, then machine 2 and so on. After completion on one machine, a job joins the queue at the next machine. All queues operate under the first in first out (FIFO) discipline; that is, a job can not "pass" another while waiting in queue (see *Figure 1-3*) [26].

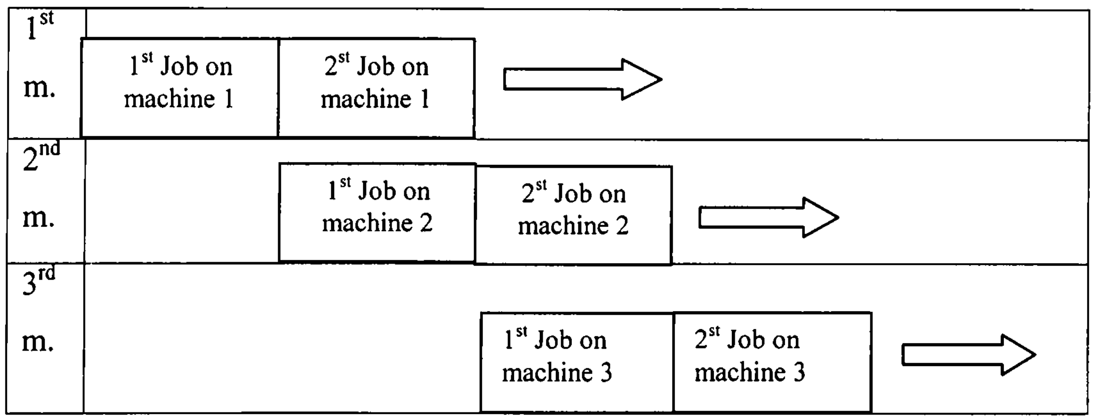


Figure 1-3. Flow Shop.

The renovation processes are carried out by using various machines in the sections. There are mainly three products and sub-units for each type of motor. These products are renovated large, small and unimog type motors. There are about 200 workers in the renovation unit for dismantling, renovating, assembling and testing operations. Most of the workers are equally qualified.

Renovation lead times for the products are known approximately by the past experiences. Production planning is done very roughly based on these data, the state of machines and workers.

The rest of the thesis is organised as follows: In Chapter 2, a brief review of the literature is given. The renovation department is described in Chapter 3. First, a general view of the system is given and the simulation model is explained in detail. The elements of the simulation model, their relationships, and the flow of the entities that represent the products are explained. The data requirement is also discussed. Finally the model is verified and validated. Output data analysis is performed in Chapter 4. Determination of the warm-up period and the run lengths, sample size and steady state performance measures are explained. In this chapter, genetic algorithm is used to optimise the system's utilisation and throughput. In Chapter 5, an additional unit is applied to the simulation model and simulation experiments performed to see the effect of the alternative configurations. In Chapter 6, the existing system is modified from flow shop to flexible flow shop to discuss the effects on the simulation model. In Chapter 7, the effects of the ready spare part usage on the simulation model are investigated. In Chapter 8, the results of the simulation study are discussed and further research topics are stated.

Chapter 2:

LITERATURE SURVEY

Simulation has been applied extensively and successively to a wide range of military problems, including wargaming, acquisition, logistics, and communication. The use of modelling and simulation is most prevalent in the areas of engineering and manufacturing. Many commercial simulation languages (e.g. Awesim, Arena, and Automod) are used in weapon system design, production, and maintenance (Kang and Roland, [19]).

Manufacturing is one of the earliest simulation application areas (Naylor et al. [22]) Simulation provides a method for finding answers to questions about the behaviour of manufacturing system. Savolainen et al. [30] indicate that simulation models are really formal descriptions of real systems to understand conditions as they exist in the system today and to achieve a better system design through performing what-if analysis. Also, Law and McComas, [21] have given the steps of the simulation of the manufacturing systems.

The use of modelling and simulation in manufacturing is aiming toward a future "virtual manufacturing" environment. In this approach the operational requirements identified in the synthetic battlefield environment are translated into design concepts. These designs are passed along to a network of distributed manufacturing processes, facilities, and tooling requirements. This vendor base is the closest to the manufacturing processes and is in the best position to develop cost and schedule estimates. These estimates may then be fed back to provide better estimates of costs and schedules to support trade-offs and system-level alternative evaluation in cost and operational effectiveness analysis (Piplani et al. [27]).

There have been many trends in manufacturing methods. Types of manufacturing systems are defined by Harrell and Tumay [11]. These are project shop, job shop, cellular manufacturing, flexible manufacturing systems, batch flow shop, and line flow systems.

An overwiew of how simulation modelling techniques can be employed in the design and analysis of advanced manufacturing systems are presented Evans and Biles [5].

While doing literature review, it is noted that there are too many studies in manufacturing and logistics area, but there are limited studies, involving both military and manufacturing (maintenance) systems. There are some studies in the USA Armed Forces, but they are usually classified and hence, not accessible. For this reason, we will present a few studies in the maintenance of military systems.

Parsons and Krause [24] studied about the tactical logistics and distribution systems simulation to response to changing technology. New supply and distribution techniques employing a wide variety of equipment combinations both existing and proposed systems are tested.

John D. Ianni [13] studied maintenance simulation in the US. Air Force to decrease the cost for requirements of the missions. In his study, he determined life cycle cost and maintenance problems. The research addresses how the usage of the human figure models can be used to simulate maintenance.

Larry Jenkins [14] developed a simulation model to schedule the inspection of machine breakdowns that can be used for field maintenance systems. The aim of the study, if a machine (vehicle) breaks down because of failure of a compenent, cost of lost production and repair will be greater than if the part is replaced earlier in a routine inspection (maintenance). Simulation was conducted using FORTRAN and BASIC. The program tabulates for each of compenent the number of breakdown replacements, the number of replacements on inspection, and inspection frequency that minimise total cost.

Harvey et al. [12] studied and developed a SLAM II model to simulate the C-141 Depot maintenance for defining resource requirements. There are about 275 C-141 aircraft in the US. Air Force. Approximately every six monts each aircraft is flown to depot and undergo to the programmed periodic depot maintenance (PDM). PDM is a process that inspects and repairs as a mechanical, electrical, hydraulic, and structural compenents of the aircraft. Simulation was chosen as the evaluation tool for this project due to its ability to handle complex requirements for resources, as well as the stochastic processing times. First the initial model built up and was used to determine the achievability of present throughput goals to identify bottlenecks within the system then proposed model built and compared with the existing system. The study is resembling our study with respect to military depot maintenance and the structural design of the maintenance unit.

There are three main approaches in discrete event simulation models (see Pritsker, pages: 54-58 [25]): Event orientation, activity scanning orientation, and process orientation.

Garzia, and Zeigler [10] explain the structure and development of discrete event simulation models together with simulation languages. They emphasize the importance of event list by saying that " the heart of a simulation program is the event list, an ordered list of everything that happens during the simulation"

Ronald D. Painter [23] indicates that the immediate need facing the military simulation community is to agree on and build a framework for object-oriented simulations due to requirement of rapid definition of the simulation objects and standardization of the systems.

There are also three main procedures for gathering observations in simulation: The replication method, the subintervals (batch means) method, and the cycles method. In all methods, initial warm-up period is allowed for the system to reach steady state. In the replications method, observations are gathered from separate runs having the same initial conditions but different sequence of random numbers. In our study, an event oriented discrete simulation model is developed. Observations for the experiment are gathered by using the replications method.

Jerry Banks [2] explain the importance of selecting software for simulation and selection includes: Input, Processing, Output, Environment, Vendor, Cost. The most popular event oriented discrete simulations are GPSS/H, SLX, SIMSCRIPT.II.5, AweSim, SIMPLE++, and EXTEND. In recent years, many manufacturing-oriented software laguages have been developed. Some of them are ProModel, AutoMod, Taylor II, WITNESS, FACTOR/AIM, and ARENA. In this study, AutoMod simulation software is used. Because it is very powerful in its description of material handling systems. The range of definition is extensive. Numerous control statements and also a separate utility option (AutoStad) is available.

The decision-makers concerned with whether a model and its results are correct. This concern is adressed through model verification and validation.

Robert G. Sargent [28] recommended a procedure for the verification and validation.

Simulation models are built with the intent of studying the behaviour of the real system represented by the model. However, a simulation model generates random outputs. These outputs should be analysed with certain tecniques and concepts to interpret some conclusions about the model, Centeno and Reyes, [3].

The benefits of the planning and proper design can often increase the precision of estimates and strengthen confidence in conclusion in drawn. Farrigton and Swain [6] are described a methodology for manufacturing systems.

There are a number of techniques to find the optimal values of controllable variables through a responce surface generated by simulation of the particular system (Tekin, Sabuncuoglu, [31]). The classification sheme is:

1) Local optimisation
   - Discrete Decision space
   - Continuous Decision space
2) Global optimisation
   - Genetic Algorithms
   - Tabu Search
   - Simulated Annealing
   - Bayesian/Sampling Algorithm
   - Gradient surface Method

The genetic algorithm procedure is a useful procedure when the system has stochastic variables (Stuckman, Evans and Mollaghasemi, [29]). In our study, we used genetic algorithms to optimise certain performance measures of the existing system.

Farhad Azadivar [1] presented the use of simulation in optimisation of maintenance policies and selecting an optimum maintenance policy. In the research, response surface topology is investigated with using genetic algorithms and best decision on the type of maintenance policy and the other characteristics of the system are presented.

Chapter 3

SIMULATION OF THE RENOVATION UNIT IN 1011 MAIN REPAIR DEPOT.

In this study, simulation is used to analyse the behaviour of the renovation unit. Stochastic flow shops can be analysed with both queuing network models and simulation models. For the simple systems, performance measures can be computed mathematically at great savings in time and expense compared to use of simulation model. But for realistic models of complex systems, simulation is usually required. Because queuing models required many simplifying assumptions in the realistic systems. The renovation unit has 27 sections and their storage capacities are bounded. Also the system has stochastic interruptions which can not be modelled by queuing models such as breakdowns. This stochastic and dynamic nature requires simulation.

3.1. Formulation of problem and plan study

The objectives and the scope of the project is to examine the behaviour of the system, to evaluate the existing system and to estimate the performance measures such as, utilisation of resources, queues and their lengths, average number of renovated motors in system and average breakdown rates in system.

In this study, we will investigate the capacity of the existing and proposed systems, the relationships between sub-units, and the effect of hierachical laws on the systems.

Data requirements: The required data for the modelling of the renovation unit is determined and presented in the Appendix A.1.

The study will be used for understanding the existing system (the way of working of the whole system for finding problematic areas and re-optimising the system). In addition, the end user will be all Turkish Army Maintenance System. They can redescribe their maintenance system and maintenance plans.

We made the following assumptions at the existing system that has no priority for renovation orders, no set-up times, and no back orders. Also distribution system and its requirements are not included in the simulation model.

3.2. Model Development

The model is developed under the structure of the Figure 3.2-1 [20].

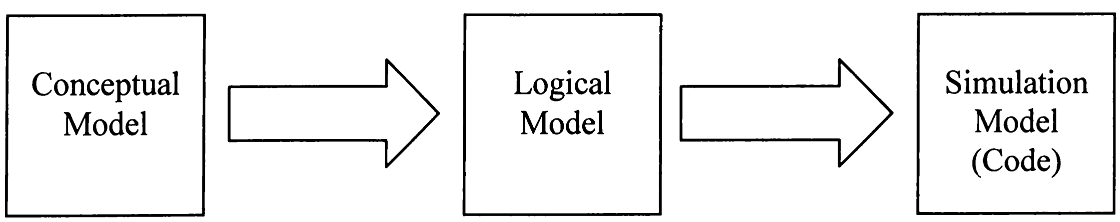


Figure 3-1. Structure of development.

3.2.1. Conceptual model

Conceptual model contains elements of the real system, which we believe should be included in our model. These include events, entities, attributes, activities, exogenous variables, endogenous variables, operational rules, initial conditions, and assumptions of the existing system.

3.2.1.1. Events

In this model the events are preparation of production plans, arrival and departure of motors to the disassembling and washing process, renovation processes, repair processes, assembling processes, testing process, and packing process. A complete list of the events is presented in the Appendix A.2 details.

3.2.1.2. Entities and Attributes

The entities are large size motors, small size motors, unimog size motors, electrical parts, and fuel oil system parts (carburation parts). The attributes are the type of motors, timestamps for every entity, and part availability.

3.2.1.3. Activities

The activities are the disassembling, washing, block renovation, crank renovation, cylinder bed preparation, piston renovation, 1st and 2nd repair, mounting, testing and packing sections. The detailed activities are also presented in the Appendix A.2.

3.2.1.4. Exogeneous Variables

Exogeneous variables are type of motors and their specifications, number of workers & resources and their capacities, flow processes, operation times, arrival patterns, work-times: A shift of 7.5 hours in a day, operation policies, number of breakdowns and their specifications.

3.2.1.5. Endogeneous Variables:

Endogenous variables are the number of motors, number of the motors waiting for renovation, examination time of processes, sections utilisations and queue lengths.

(1). State Variables: State of motors, state of renovation orders, status of sections, rate of disposals, state of queues and queue lengths, state of spare parts availability are the state variables of the system.

(2). Performance Measures: Waiting times, average time in system, queue lengths for every section, average renovated motor in system, utilisation of sections, number of disposed motors are the performance measures of the system.

3.2.1.6. Assumptions of the Model

In this study our main goal is to model the renovation unit. Therefore we included only renovation sub-sections and assumed that no beginning set up times for the sections, some data sets and processing times defined by technicians since the difficulty for obtaining data from the processes, no back order is designed in the system and no priority is assumed between the same type of motors.

3.2.1.7. Initial Conditions and Operational Rules

There is no beginning breakdown in the system and spare part levels are known at the beginning in the system. Renovation plans are prepared on one-year basis. The flow process must be applied for each type of motor and the renovation unit works 1 shift per day, each shift taking 7.5 hours and 5 days in a week.

3.2.2. Logical model

The logical relationships among the elements of the system as well as exogenous variables that effect the system. In Figure 3-2, the general lay-out of the renovation unit is given and then the following flow-charts are presented to describe the logical relation ship of the model.

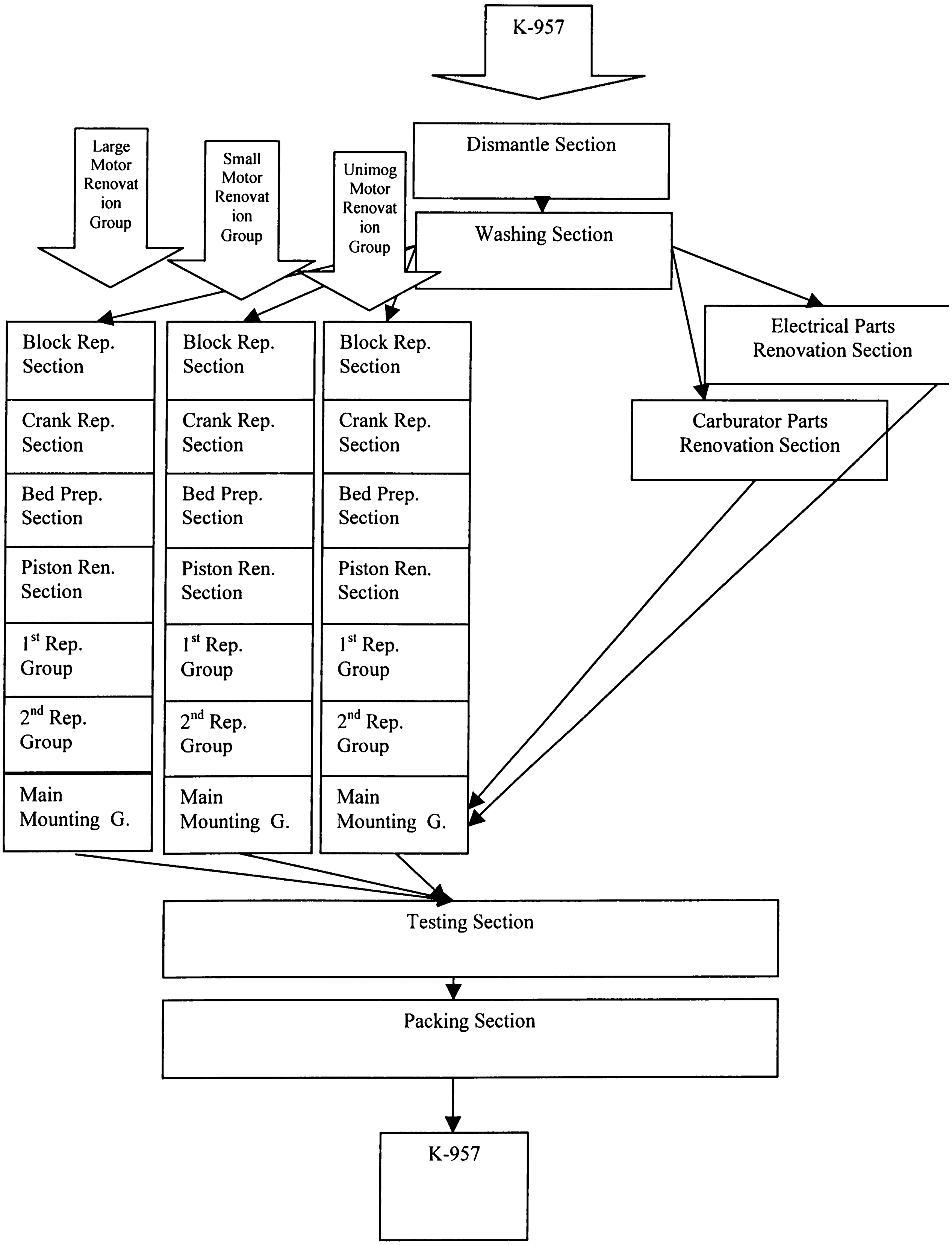


Figure 3-2. Renovation Unit System

Figure 3-3. 5th Level:

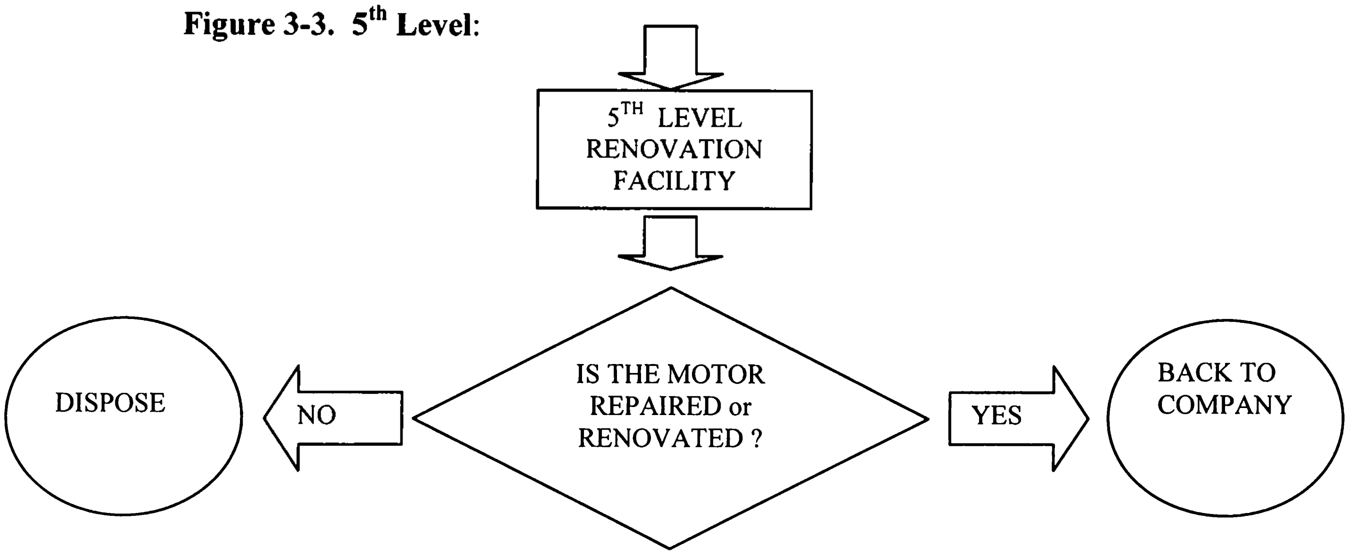


Figure 3-4. Motor Renovation Unit

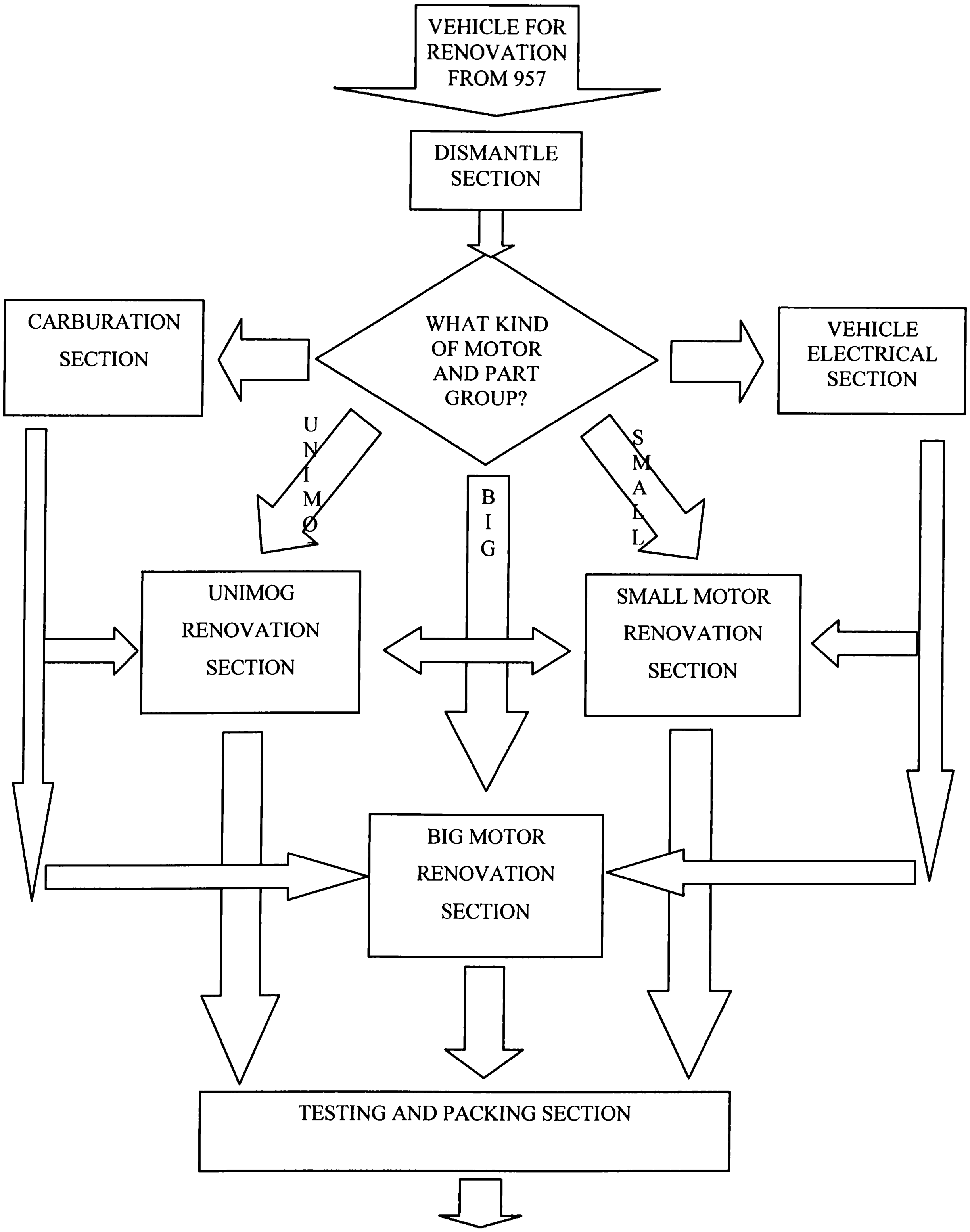
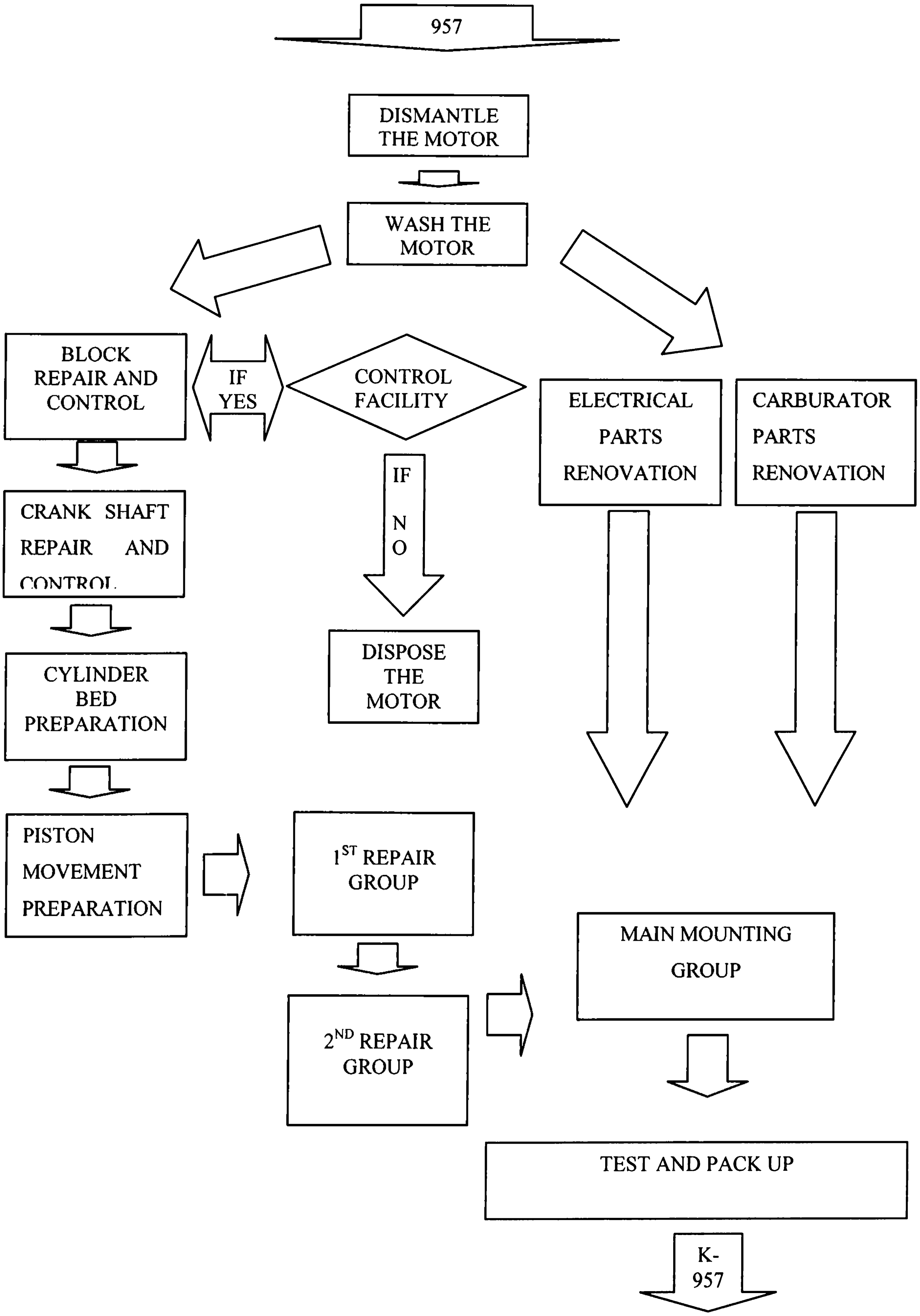


Figure 3-5. Motor Renovation Unit.



3.2.3. Simulation Model (Code)

The model, which executes logic, contained in the logical model. The simulation code of the existing system is developed in AutoMod 9.0 (1999). AutoMod has the ability to define a sequence for moving entities through the system and it enables the modelling system especially manufacturing systems. Autostat which is output data analyse processor of AutoMod that assists to obtain confidence intervals, graphics and so on, it is portable to all types of personnel computers. The source file for renovation unit is about 550 lines and also additional user defined functions, standard library functions, time-specific functions, model communications functions and multi-model synchronisation functions are used in coding processes. The code for this section is at the Appendix A.3.

3.3. Verification and Validation of the Model

3.3.1. Verification of the Existing Model

In this section, the computer program representing the existing system is verified by using certain techniques [2].

Technique 1 (Debugging): In developing the existing system's simulation model, a computer program is written in form of modules and sub-programs. First, the main part is developed and tested. Then, additional sub-programs and levels of detail are added and debugged successively, until the model is matured to satisfactorily represent the existing system.

Technique 2 (Input and Output Control): The simulation code is run under a variety of settings of the input parameters and checked to see that the output is reasonable.

Technique 3 (Animation): An animation of the simulation model is performed and it is observed that the animation of the simulation output imitates of the existing system.

Technique 4 (Proper Software Selection): With using a simulation package (Automod 9.0) the required number of lines of code are reduced.

Technique 5 (Checking): The computerised representation is checked by 1st Lieutenant Hakan UTKU and Captain Özgür NUHUT.

3.3.2. Validation of the Existing System

Simulation model of a system is only an approximation of the actual system and embodies set of required performance measures. In validating the existing system, the most desired performance measures are used in the validation process and they are repeated whenever the model is improved or changed.

3.3.2.1. Face Validity

As explained below, the model is developed with high degree face validity.

Extensive conversations are made with the experts of the actual system. In modelling the renovation unit, information from such sources as machine operators, manufacturing and industrial engineers, managers and their reports are also referred to the knowledge of the system substantially contributed to the actual validation of the model. Numerous observations are done on the actual system. Data obtained from historical records and sorted during a time study.

3.3.2.2. Statistical Validity

This method is the most definitive test for the validation of the simulation model. The output data obtained from the simulation model is tested for close resemblances to the output data of the actual system [32].

The results of ten replications are presented in *Table 3.3.2-1*. In this table, each row represents a different replication result. The cumulative sums, averages, standard deviations and the confidence intervals (α=0.05) are shown at the end of the each column.

REPLICATION	LARGE MOTOR	SMALL MOTOR	UNIMOG MOTOR
1	2310	3175	1821
2	2332	3204	1822
3	2346	3212	1825
4	2350	3168	1841
5	2356	3204	1815
6	2333	3216	1880
7	2347	3176	1813
8	2341	3185	1833
9	2313	3184	1861
10	2331	3218	1849
CUM. SUM	23359	31942	18360
AVERAGE	2335.9	3194.2	1836
STAN. DEV	15.26	18.63	21.79
C. I. for 0.05	9.46	11.55	13.50

Table 3.3.2-1. Throughput of the existing system.

REPLICATION	LARGE MOTOR	SMALL MOTOR	UNIMOG MOTOR
1	2260	3150	1840
2	2290	3200	1820
CUM. SUM	4550	6350	3660
AVERAGE	2275	3175	1820
STAN. DEV.	21,21	35,35	28,28
CI .for 0.05	34,89	58,15	46,52

Table 3.3.2-2. Historical data about the actual system.

When the results (given in *Table 3.3.2-1* and *3.3.2-2*) are compared using the statistical tests (t-test), it is observed that there are no significant changes, in terms of the averages of the number of the renovated vehicle motors. As a technique, Welch approach [32] is used to validate the existing system's validation process since the historical data and the existing system simulation results are independent and no correlation between each other.

a. Comparison for the large motors.

The Welch approach is applied to see if there is a difference, if any, between the actual and the simulation model. Even though there is 2.67% difference in the number of renovated large motors but the test results shows that the simulation model is not different from the actual system, (See *Table 3.3.2-3*) because the average difference plus and minus confidence interval (-60.9±191) includes the zero.

ACTUAL SYSTEM	LARGE MOTOR
YEAR 1998	2260
YEAR 1999	2290
AVERAGE	2275
STAND. DEV.	21.21
CI. for 0.05	34.89
X-Y Difference	-60.9
CHANGE	-0.026
DEGREES OF FREEDOM	1.216
WELCH APPROACH	
CI. for 0,05	191

Table 3.3.2-3. Comparison for the large motors.

b. Comparison for the small motors.

When the same procedure is repeated for the small type of motors, we observed that there is average 0.6% difference in the simulation model. The Welch test again does not detect any statistically significant difference between the simulation model and real system, since the average difference plus and minus confidence interval (-19.2 ± 321) includes zero (See *Table 3.3.2-4*).

REAL SYSTEM	SMALL MOTOR
YEAR 1998	3150
YEAR1999	3200
AVERAGE	3175
STAND. DEV.	35.35
CI for 0.05	58.15
X-Y Difference	-19.2
CHANGE	-0.006
DEGREES OF FREEDOM	1.114
WELCH APPROACH	
CI. for 0,05	321.09

Table 3.3.2-4. Comparison for the small motors.

c. Comparison for the unimog motors.

When the same procedure is repeated for the unimog, we observed that there is average 0.879% difference in the simulation model. The Welch test result shows that the simulation model is not different from the actual system, (See *Table 3.3.2-5*) since the average difference plus and minus confidence interval (-16± 241) includes the zero.

REAL SYSTEM	UNIMOG MOTOR
YEAR 1998	1800
YEAR1999	1840
AVERAGE	1820
STAND. DEV.	28.28
CI. for 0.05	46.52
X-Y Difference	-16
CHANGE	-0.008
DEGREES OF FREEDOM	1.250
WELCH APPROACH:	
CI for 0,05	241.020

Table 3.3.2-5. Comparison for the unimog motors.

d. Comparison for the breakdowns.

When the simulation model is compared the real system in terms of breakdowns, we see no significant difference, (given in the Table 3.3.2-6 and 3.3.2-7) since the average difference plus and minus confidence interval (-0,8± 4,155) includes the zero.

REPLICATION	EXISTING SYSTEM
1	9
2	13
3	16
4	18
5	11
6	12
7	6
8	11
9	22
10	15
SUM	133
AVERAGE	13,3
STAND. DEV.	4,62000481
CI for 0.05	0,854199709

Table 3.3.2-6. The average breakdowns in the simulation model.

REAL SYSTEM	BREAKDOWNS
YEAR 1998	12
YEAR1999	13
AVERAGE	12,5
STAND.DEV	0,70
CONF.INT	1,16
X-Y= Difference	-0,8
DEGREES OF FREEDOM	5,968
WELCH APPROACH	
CONF.INT(0,05)	4,155382

Table 3.3.2-7 Comparison for the actual and the simulation model.

Chapter 4:

THE EXPERIMENTATION AND OUTPUT DATA ANALYSIS

This study is performed to evaluate the performance of the existing system. Having the simulation model developed, verified and validated with all the necessary data collected, the initial transient period of the system and the steady state performances are analysed since the system under analysis is non-terminating. Recall that a non-terminating simulation is one which there is no natural event E to specify the length of run and a measure of performance for such a simulation is said to be steady state distribution of some output stochastic process Y_1, Y_2, . . . If the random variable Y has the steady state distribution then we may be interested in estimating the steady state mean v=E(Y) [20].

4.1. Determination of the Warm-up Period

Statistics gathered during the warm-up period that may not truly reflect the steady state of the system [2]. Thus, a warm-up period analysis needs to be first carried out to determine the length of this initial transient state.

The system does not have fixed starting condition and a natural event specifying the end of a run can not be defined. Although the renovation unit stops renovation between two consecutive shifts, the renovation starts at a state that is the same as the end of the previous shift. Hence, we are interested in the steady state performance of the renovation unit. We started the simulation with an empty system except the electrical and fuel oil parts renovation sections and made ten replications (40 daylong) for the existing system. These initial runs were used for determination of the length of the warm-up period after which the system can be said to be in the steady state. In calculating statistics, we deleted the observations collected during warm-up period.

We decided to use the time an entity spends in the resources and queues, and utilisation of the resources. After the system reaches the steady state these measures should not change much although random fluctuations are possible. We used the Welch's procedure to identify the transient period. Recall that in this method, we execute the following procedure:

1. Make n replications of the simulation ($n \geq 5$), each length of m (where m is large).

2. Let Y_{ji} be the *i*th observation from the *j*th replication ($j=1,2,\ldots,n$; $i=1,2,\ldots,m$) then let $\overline{Y}_I=\sum^{n}_{j=1} Y_{ji}/n$ for i=1,2,m. The averaged process $\overline{Y}_1$, $\overline{Y}_2\ldots$ has means $E(\overline{Y}_i)=E(Y_i)$ and variances $Var(\overline{Y}_i)=Var(Y_i)/n$. Thus, the averaged process has the same transient mean curve as the original process, but its plot has only (1/n) th the variance.

3. To smooth out the high-frequency oscillations in $\overline{Y}_1$, $\overline{Y}_2,\ldots$, we further define the moving average $\overline{Y}_i(w)$ as follows:

$$\overline{Y}i(w) = \begin{cases} \dfrac{\sum_{s=-w}^{w} \overline{Y}_{i+s}}{2w+1} \; if\ .i = w+1,\ldots,\ m-w \\ \dfrac{\sum_{s=-(i-1)}^{i-1} \overline{Y}_{i+s}}{2i-1} \; \ldots\ldots\ldots \quad \ldots if\ .i = 1,\ldots,\ w \end{cases}$$

Thus, if I is not too close to the beginning of the replications, then $Y_i(w)$ is just the simple average of 2w+1 observations of the averaged process centred at observation i. It is called a moving average since i moves through time.

4. Plot Y_i (w) for i=1,2, ... m-w and choose i to be that value of i beyond which $Y_1(w)$, $Y_2(w)$... appears to have converged (Welch, [32]).

Graphical analyses of these measures show that the system rapidly reaches the steady state. For this analysis, we run the system for 960 hours. Moving average of these values are taken (w=6). The graph of the throughput (per hour) versus time of large, small and unimog motors is given in Figure 4-1. In the graph, three of the responses level off after about 66 hours. When we have multiple responses in the existing system (See Appendix D.1 Figure 1,2,3,4), we decided to take first 72 hours as the warm-up period for our simulation study.

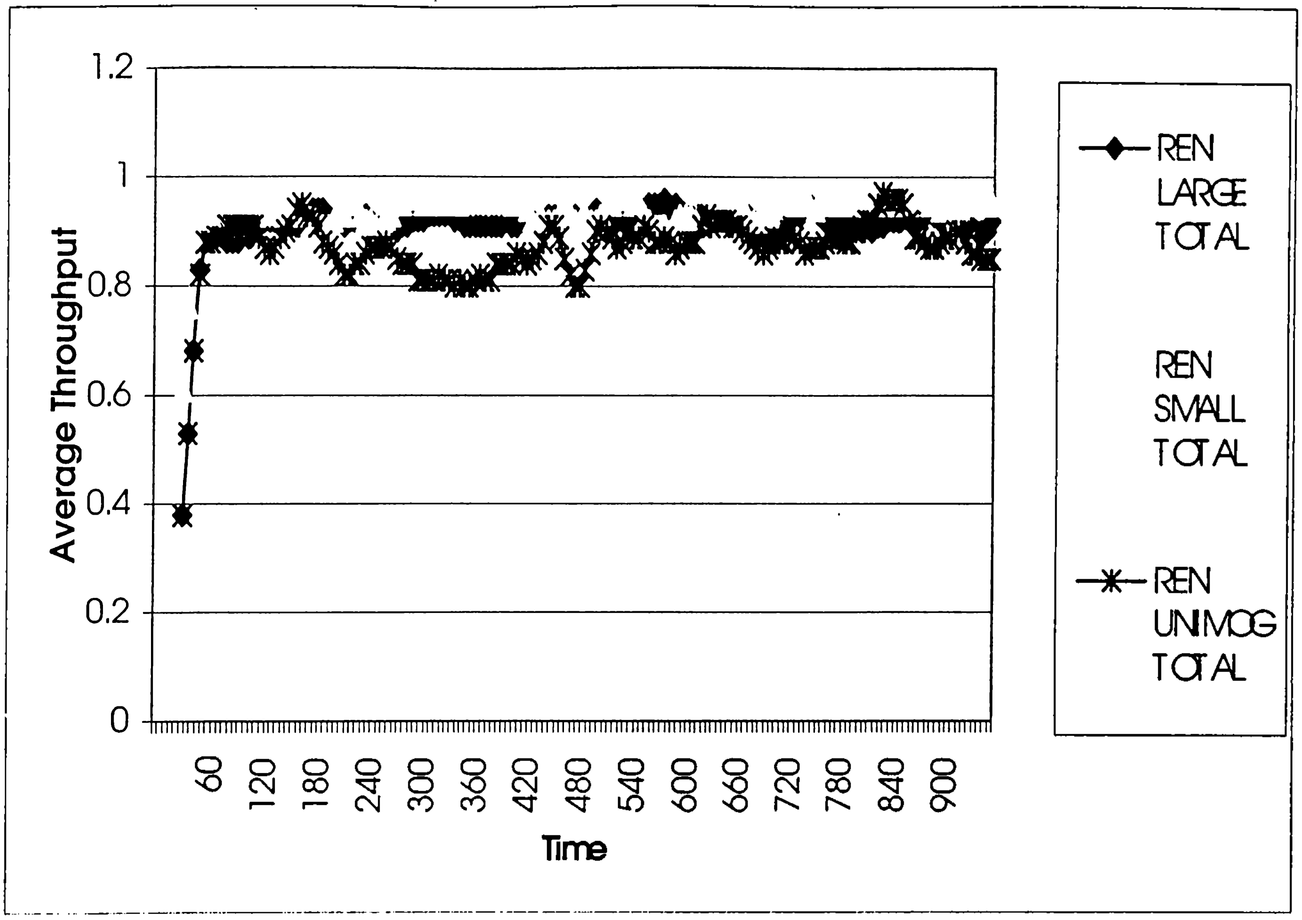


Figure 4-1 Sub-sections throughput (per hour) versus time .

4.2. Determination of the Run Length

We have decide that a run length of one year would be sufficient for simulating the renovation unit since we have historical throughput data on the yearly basis [33]. This corresponds to 1820 hours [(365-104(weekend)-18 (religious, governmental and yearly holidays)*7.5 (hour/shift)]. The total run length including the warm-up period becomes 1820 + 72 = 1892 hours.

4.3. Determination of the Total Sample Size Required

We use the following inequality to determine the number of replications required achieving the desired accuracy. Recall that initially ten replications were taken to validation process. The inequality given below assures that we obtain the results within the desired level of accuracy. The accuracy is defined as 20 motors per year averagely at the simulation model for every type motor. The absolute error a β (half-length) is defined according to production control and management section's thoughts in face validity.

$$i \geq S^2(n)\left[\frac{z_{1-\alpha/2}}{\beta}\right]^2$$

We applied the inequality to find required number of replications with respect to average renovated motors according to their types.

We get $n_a^*(\beta)$ by using the following iterative procedure [20]:

$$n_a^*(\beta) = \min\left\{ i \geq n : t_{i-1,1-\alpha/2}\sqrt{\frac{S^2(n)}{i}} \leq \beta \right\}$$

And we calculated the below results.

Type	*I*	*t-test statistic*	*≤β*
Large	3	2.980303	3
Small	4	4.442384	5
Unimog	5	6.077436	7

. n=10 replications, $\beta = 20$, $\alpha = 0.05$

For large motors,

$n_a^*(\beta) = n_a^*(20) = 3$

Therefore, 3-10 = -7 additional replications is not needed.

For small motors,

$n_a^*(\beta) = n_a^*(20) = 5$

Therefore, 5-10 = -5 additional replications is not needed.

For unimog motors,

$n_a^*(\beta) = n_a^*(20) = 7$

Therefore, 7-10 = -3 additional replications is not needed.

From the above calculations, we obtain the following results.

Throughput Types	**Sample Size**	**Additional need for Replication**
Large motors	3	-
Small motors	5	-
Unimog motors	7	-

Table 4.3-1. Required Sample Sizes.

4.4. Output Analysis of the Existing System

Replication/Deletion method is used to remove initial bias by using data obtained after a warm-up period in each replication. The replication/deletion method strived to use steady-state data in the formation of point estimates and confidence intervals for the various responses, which is accomplished by obtaining the average level of the response for each replication after the warm-up period. These averages can be shown to be independent and approximately normal random variables (see for the normality check in Appendix B). Thus, based on independence and normality assumptions we construct a confidence interval for the steady-state mean value of the responses (See for *Table B.4-1* in appendix for the average responses, variances, means, medians, number of replications, and the confidence intervals (1-α=0.90,0.95,0.99)).

Table B.4-1 contains a great amount of information that it is difficult to interpret the results. Therefore, we converted them to graphs. As seen in the *Figure 4.4-1.(a,b,c,d),* large motors electrical renovation section (Q1CARB), large motors electrical renovation section (Q1ELECT), small motors electrical renovation section (Q2ELECT), large motors block renovation section (QBLOCK), large motors crank renovation section (QCRANKB), small motors crank renovation section (QCRANKS) have relatively high average waiting time in queues.

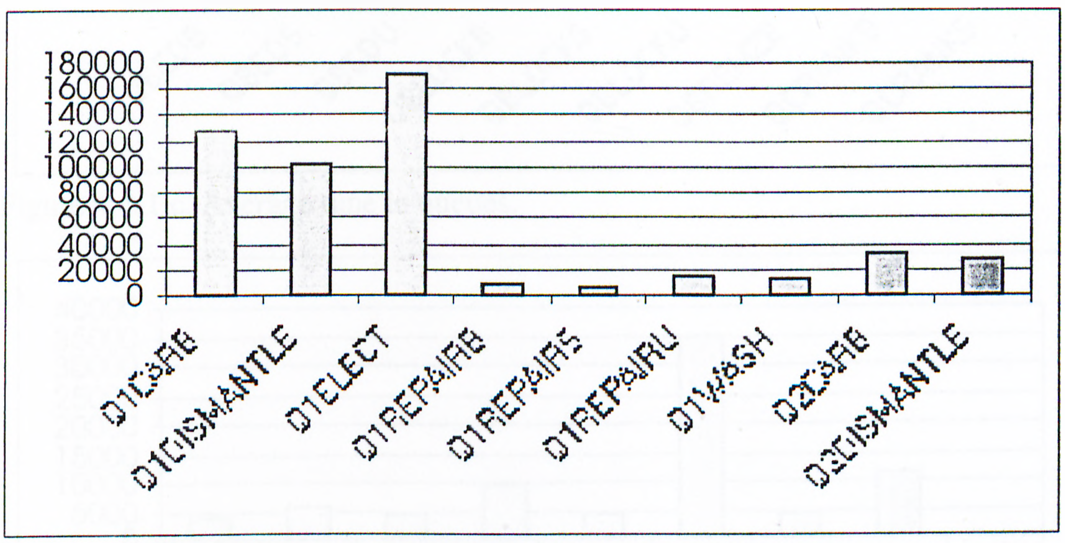


Figure 4.4-1.a. Average time in queues.

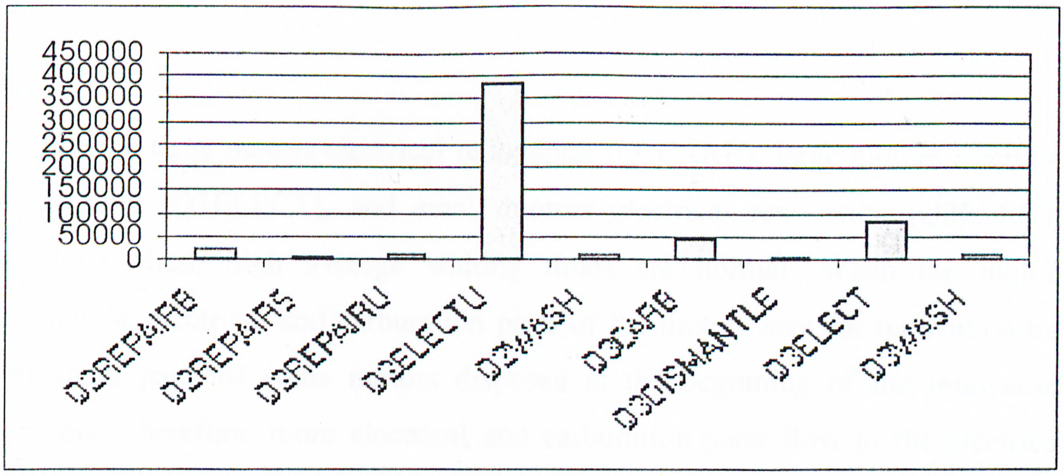


Figure 4.4-1.b. Average time in queues.

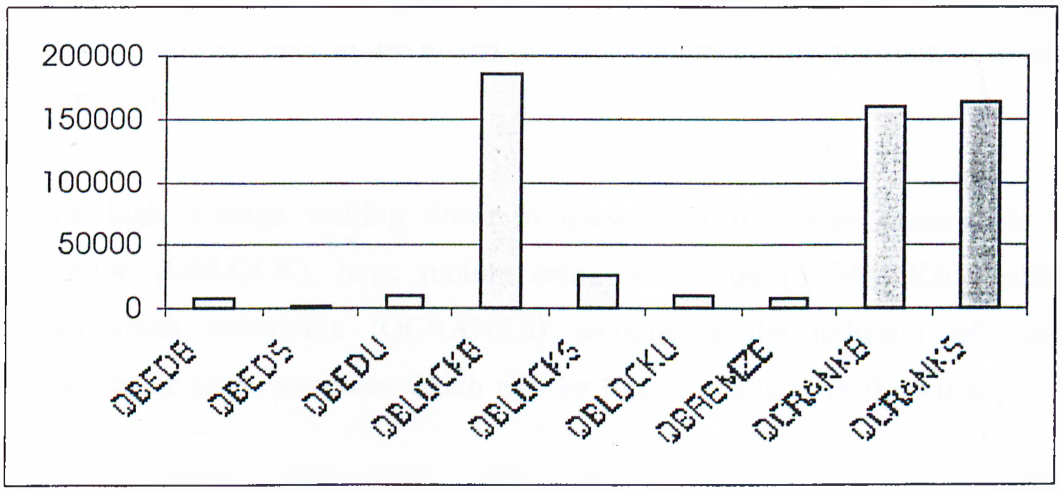


Figure 4.4-1.c. Average time in queues.

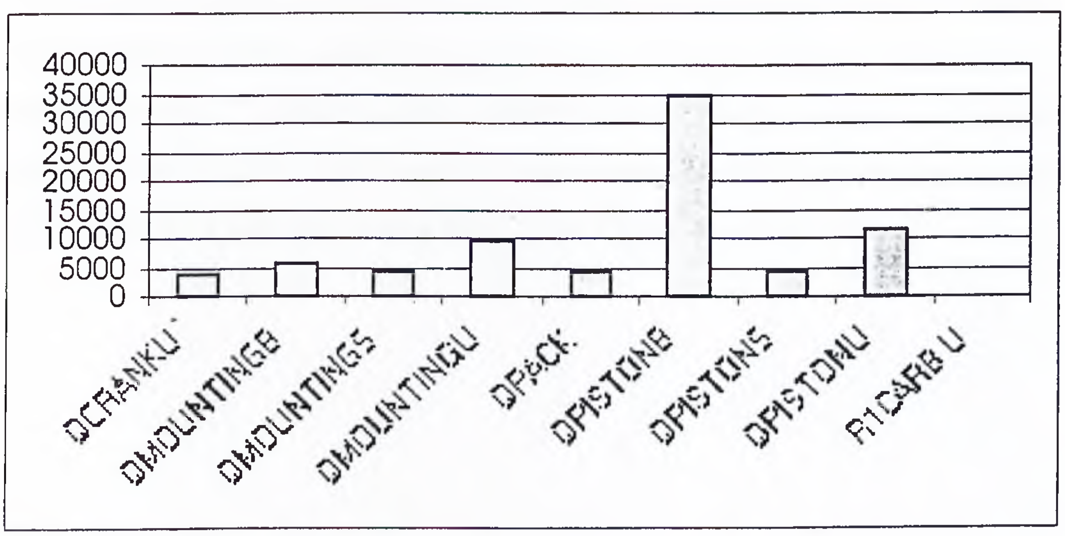


Figure 4.4-1.d. Average time in queues.

For the large motors electrical renovation (Q1CARB), large motors electrical renovation (Q1ELECT), and small motors electrical renovation (Q2ELECT) sections, these high average waiting times are normal. When the motors dismantle, electrical and carburation parts of the motors sent for renovation but the main parts of some motors disposed at the beginning of the renovation process. Therefore, more electrical and carburation parts flow to the electrical and carburation renovation sections are renovated for the mounting process but not all the parts can not be processed due to over utilisation. These excess renovated parts are sent to the mixed goods accountancy for distribution to the military units.

The high average waiting times in queues for the large motors block renovation (QBLOCK), large motors crank renovation (QCRANKB), small motors crank renovation (QCRANKS) sections is the indicator of over utilisations of some resources which will be further analysed in the subsequent sections.

From the simulation results, we also observed that large motors 2nd repair section (R2REPAIRB), small motors electrical renovation section (R2ELECT), unimog motors 1st repair section (R1REPAIRU), large motors carburation renovation section (R1CARB) and the large motors electrical renovation section (R1ELECT) have high processing times (See *Figure 4.4-2.a,b,c,d*).

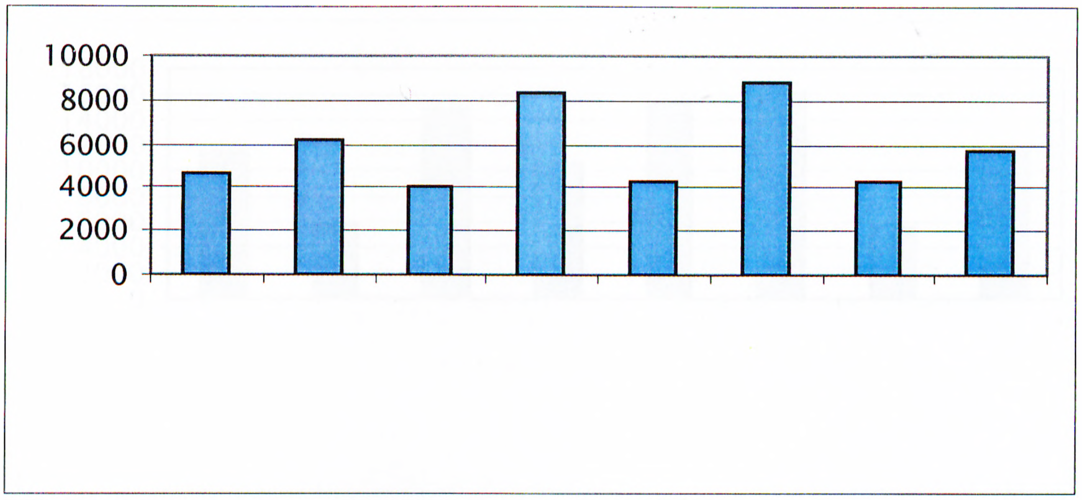


Figure 4.4-2.a. Average time in resources.

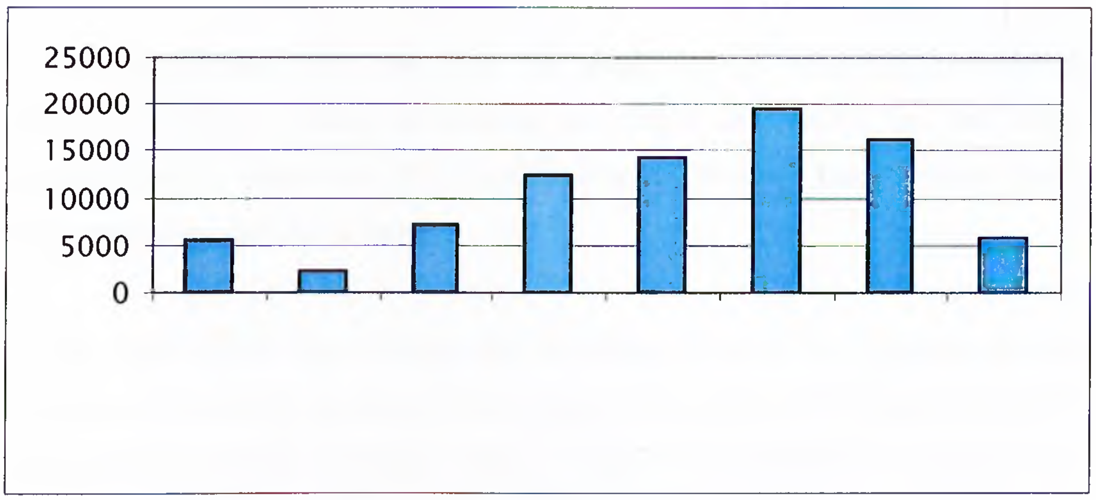


Figure 4.4-2.b. Average time in resources.

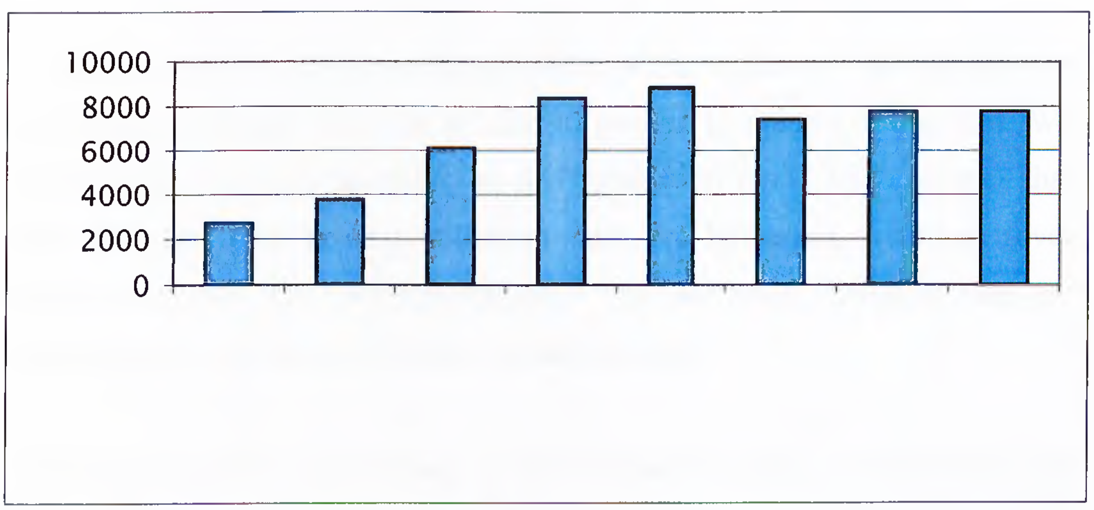


Figure 4.4-2.c. Average time in resources.

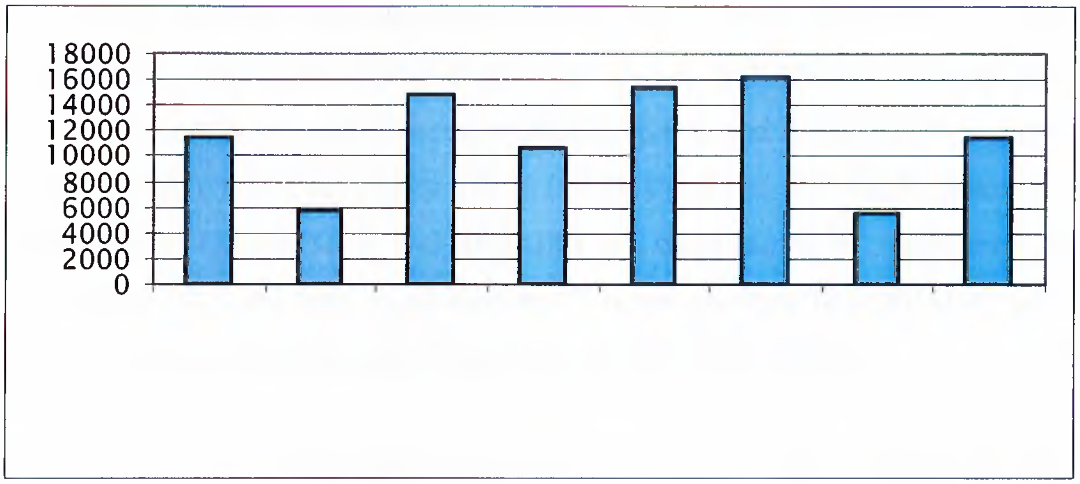


Figure 4.4-2.d. Average time in resources.

The processing times are high for small motors electrical renovation (R2ELECT), large motors carburation renovation (R1CARB) and the large motors electrical renovation (R1ELECT) sections. Because these sections have many processes steps to be done.

The main effects that increase the processing time of the sections are the processing steps at the sections and the type of the motors. The large motors 2nd repair (R2REPAIRB), unimog motors 1st repair (R1REPAIRU) sections have several processing steps that increase the processing times.

Figure 4.4-3.(a,b,c,d) shows the utilisation of the resources. The utilisation of the resources changes between 40 and 90 percent in the renovation unit. We classified the resources according to their utilisation rates. The resources that have 90% and over percent utilisation rates are bottleneck resources (over utilised sections). The resources that have 70% and lower utilisation rates are called capacity lost resources (lower utilised sections).

In the renovation unit looking at their utilisation levels, we identified the following bottleneck sections: the small electrical renovation section (R2ELECT

U), unimog electrical and carburation renovation sections (R2ELECT U and R2CARB U), large motors block renovation section (RBLOCKB U), large and small motors crank renovation sections (RCRANKB U and RCRANKS U), large motors piston renovation section (RPISTONB U), large and small motor 1st repair sections (REPAIRL U and REPAIRS U), small motor 2nd repair section (REPAIR2S U), large and small motors 1st repair sections (R1REPAIRL and R1REPAIRS) and small motors 2nd repair section (R2REPAIRS).

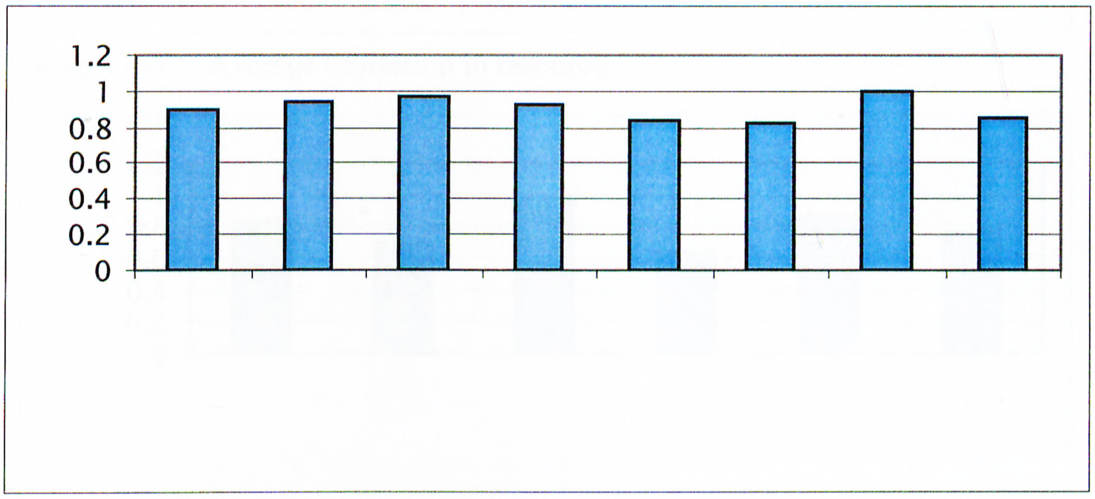


Figure 4.4-3.a. Average utilisation in resources.

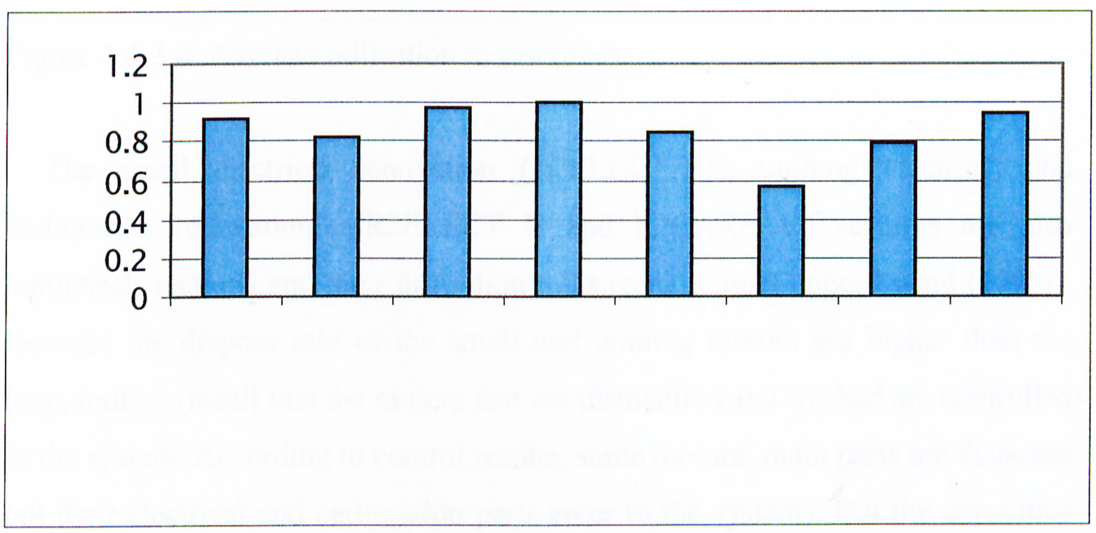


Figure 4.4-3.b. Average utilisation in resources.

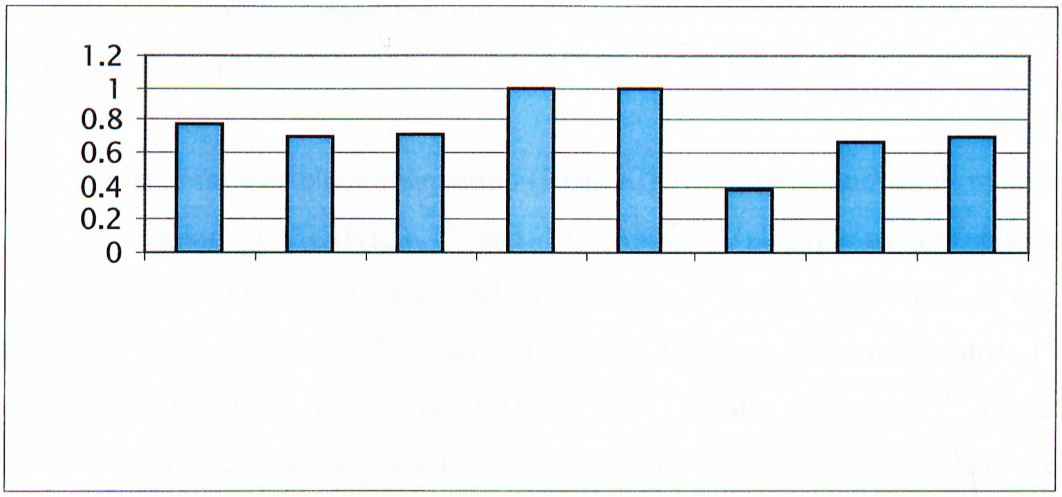


Figure 4.4-3.c. Average utilisation in resources.

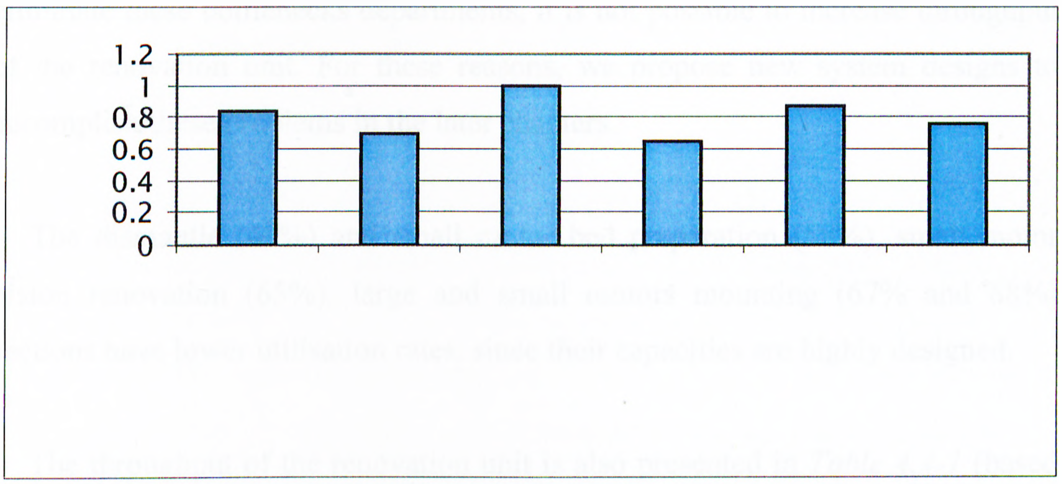


Figure 4.4-3.d. Average utilisation in resources.

The small electrical renovation (R2ELECT U), unimog electrical and carburation renovation (R2ELECT U and R2CARB U) sections are also bottleneck sections and their utilisation rates reach almost upper bound (100%). Because the dispose rate of the small and unimog motors are higher than the large motors (recall that the motors that are dismantled and washed are controlled in the system. According to control results, some motors` main parts are disposed but their electrical and carburation parts enter to the system). But the capacities of these sections are designed to needs for main renovation sections. Therefore, disposes of motors increase the utilisations of the electrical and carburation renovation sections.

At the large motors block renovation (RBLOCKB U), large and small motors crank renovation (RCRANKB U and RCRANKS U), large motors piston renovation (RPISTONB U), large and small motor 1st repair (REPAIRL U and REPAIRS U), small motor 2nd repair (REPAIR2S U), large and small motors 1st repair (R1REPAIRL and R1REPAIRS) and small motors 2nd repair (R2REPAIRS) sections` utilisation rates are over 90%. This means that, there are over utilisations and bottlenecks due to their capacity limitations. Unless we eliminate these bottlenecks departments, it is not possible to increase throughput of the renovation unit. For these reasons, we propose new system designs to accomplish these problems in the later chapters.

The dismantle (40%) and small motor bed preparation (58%), small motor piston renovation (65%), large and small motors mounting (67% and 68%) sections have lower utilisation rates, since their capacities are highly designed.

The throughput of the renovation unit is also presented in *Table 4.4-1* (based on ten replication results). These results are also very close to the historical data. Therefore, the production control and management department may use the simulation data as a real data.

Throughput	Averages (in steady state)	Half-length (0.05)
Large motors	2331	2321.5-2340.4
Small motors	3189,8	3177.9-3201.6
Unimog motors	1852.9	1836-1869

Table 4.4-1. Throughput of the system.

4.5. Improved system with GA (Genetic Algorithms)

The real-word systems are so complex that computing values of the performance measures and finding optimal decision variables analytically are sometimes very hard or impossible [29].

Recall that a stochastic process is a collection of random variables ordered over time, which is all defined on common sample size. The simulation model developed for the renovation unit use random variables as input and it has many dynamic procedures such as breakdowns. This stochastic and dynamic nature of the renovation unit requires computer simulation to improve of the performances of the system.

In our problem, we want to increase throughput of the system with lowest additional resource requirements. Therefore, we defined controllable and quantitative factors that vary and the values for each of the factors. The capacities of the renovation sections are considered to be the input factors or decision variables. In the objective function, we used the resource utilisations and total throughput of the system with equal relative importance.

We used the optimisation utility of AutoStad to improve the performance of the existing system. In the next sections, we briefly explain the evaluation algorithm, application of the processes in the renovation unit, and give the computational results.

4.5.1. Evaluation algorithm

We performed evaluation using an optimisation algorithm called an evolution strategy algorithm (Genetic Algorithm). Evolution strategies process a population of solutions during each iteration of the search. The algorithm in

AutoStat tries to avoid finding a local optimum while seeking the global optimum.

Survival of the fittest

Evolution strategies are based on the theory of evolution. An initial population (made up of sets of factor values) combine to create the next generation of factor values (**children**). The children of that generation inherit traits from each of their **parents**, and they also have slight differences, called mutations. The fittest children of that generation (as defined by your fitness function) live to become the parents the next generation, and so on.

For the first generation:

1. Randomly create the first generation of children. Each generation contains 7N number of children, where N is the number of parents per generation. For example, we have defined the number of parents to be 3, therefore algorithm created 21 children. Each child is randomly assigned factor values. For example, assume you have defined 3 factors, and each factor's values are being varied from 1 to 4.

2. Make the runs for each child.

3. Based on the fitness score for each child, pick the best N children to use as parents for the next generation, where N is the number of parents per generation.

4. To create each child in the new generation, randomly pick two of the parents (selected in step 3), combine them (take some of the factor values from one parent and some values from the other), then mutate the factor

values slightly within the factor's set of defined values. Create 7N number of children, where N is the number of parents per generation. Because parents are chosen randomly, it is possible that the two parents for a generation may occasionally be the same.

5. Repeat steps 2 - 4 until either the termination criteria are met or until the runs are stopped.

Local versus global optimum

The search algorithm tries a wide variety of possible solutions before it narrows down its search. Some algorithms search a smaller area and find a solution that is not the best possible choice. Evaluation strategies algorithm in AutoStat uses a globally oriented search algorithm and does a wide search to find *the best overall solution*, not just the best solution in a limited area.

Mutation

A mutation is a change to a factor value within the factor's defined set of values. Each factor is mutated independently of other factors. Integer factors are mutated then rounded to the nearest integer value. For a given factor, if further mutation is not helping the fitness score, algorithm mutates it less and less until its optimal value is determined. Then the factor is set to the best value and is not changed any more (the factor has a mutation rate of zero). Other factors that are helping the fitness score continue to be mutated until the algorithm has focused in on the optimal combination of factor values.

Subsequent generations.

We assume that algorithm has already made two runs for a given child (set of factor values), the maximum allowed is five, and the most allowed per generation is two. Algorithm would set up the following runs:

In generation four, algorithm sets up the most allowed within a generation (two). The next time the same child is created (in generation five), four runs have been made and the maximum possible is five, so algorithm sets up one additional run for that child.

4.5.2. Application of Evaluation Process

In this section, we tried to answer how to increase the throughput with lowest additional capacity changes. First, we made a sensivity analysis on the bottleneck sections and lower utilised sections one by one and found their effects on the productivity (see *Table 4.5-1).* We observed that their effects on the productivity are the same. We also increased the capacities of these sections one and two units (at the same time) to determine the effects on the renovation unit. One unit increment in the capacities of bottleneck resources increases 15% for the large motors and 6% for the small motors. One more additional unit (two-unit) capacity increment in the bottleneck sections does not make further improvement as seen in *Table 4.5-2,* in diminishing rate of return. Also, the large motor bed preparation and 2nd repair sections becomes as new bottleneck sections after these capacity increments.

We defined the ranges of the factors in *Table 4.5-3*. The ranges of the factors are defined with respect to above sensivity analyses and by taking technical staffs` opinions (see *Table 4.5-4*). The minimum ranges of the bottleneck sections are set to the existing capacities of resources. The maximum ranges of these sections are set according to results of the sensitivity analyses and expert opinion. Note that the upper limits are set over the results found in the analyses, so that they do not become a tight constraint. Also, large motor bed preparation and 2nd repair sections are added to the factors and their ranges determined like other over utilised sections. For the lower utilised sections, maximum ranges are set to their existing capacities and minimum ranges are set lower than the defined values at the sensivity analyse to see whether we can achieve the same throughput level with the less number of resources.

Large Motor	Block Ren. S.	Crank Ren. S.	Piston Ren. S.	1st Rep. Sec.
Existing Cap.	2310 (3)	2310 (3)	2310 (3)	2310 (2)
Exist. Cap. +1	2321 (4)	2310 (4)	2319(4)	2310 (3)
Exist. Cap. +2	2321 (5)		2319 (5)	
Small Motor	Crank Ren. S.	1st Rep. Sec.	2nd Rep. Sec.	
Existing Cap.	3175 (3)	3175 (3)	3175 (3)	
Exist. Cap. +1	3322 (4)	3182 (4)	3182 (4)	
Exist. Cap. +2	3322 (5)	3182 (5)	3182 (5)	
Common Res	Test Section	Packing Sec.		
Existing Cap.	7303	7306		
Exist. Cap. -1	7303	7306		
Exist. Cap. -2	7303	7306		
Exist. Cap. -3	7280	7301		

Table 4.5-1. Sensivity analysis of the over utilised sections.

Large Motor		Small Motor	
Existing Cap.	2310	Existing Cap.	3175
Exist. Cap. +1	2635	Exist. Cap. +1	3368
Exist. Cap. +2	2635	Exist. Cap. +2	3368
Change	15% increase	Change	6% increase

Table 4.5-2. Results of increased capacities in the existing system.

Type	Name	Coefficient	Direction
Response	DISMANTLE SECTION	1	Maximise
Response	WASHING SECTION	1	Maximise
Response	LARGE BLOCK REN. SEC.	1	Maximise
Response	SMALL BLOCK REN. SEC.	1	Maximise
Response	UNİMOG BLOCK REN. SEC.	1	Maximise
Response	LARGE CRANK REN. SEC.	1	Maximise
Response	SMALL CRANK REN. SEC.	1	Maximise
Response	UNİMOG CRANK REN. SEC	1	Maximise
Response	LARGE PISTON REN. SEC.	1	Maximise
Response	SMALL PISTON REN. SEC.	1	Maximise
Response	UNİMOG PISTON REN. SEC.	1	Maximise
Response	LARGE BED REN. SEC.	1	Maximise
Response	SMALL BED REN. SEC.	1	Maximise
Response	UNİMOG BED REN. SEC.	1	Maximise
Response	LARGE 1ST REP. SEC.	1	Maximise
Response	SMALL 1ST REP. SEC.	1	Maximise
Response	UNİMOG 1ST REP. SEC.	1	Maximise
Response	LARGE 2ND REP. SEC.	1	Maximise
Response	SMALL 2ND REP. SEC.	1	Maximise
Response	UNİMOG 2ND REP. SEC.	1	Maximise
Response	LARGE MOUNTING SEC.	1	Maximise
Response	SMALL MOUNTING SEC.	1	Maximise
Response	UNİMOG MOUNTING SEC.	1	Maximise
Response	LARGE CARB. REN. SEC.	1	Maximise
Response	SMALL CARB. REN. SEC.	1	Maximise
Response	UNİMOG CARB. REN. SEC.	1	Maximise
Response	LARGE ELECT. REN. SEC.	1	Maximise
Response	SMALL ELECT. REN. SEC.	1	Maximise
Response	UNİMOG ELECT. REN. SEC.	1	Maximise
Response	TESTING SECTION	1	Maximise
Response	PACKING SECTION	1	Maximise
Response	**THROUGHPUT and UTILISATION**		**Maximise**

Table 4.5-3. List of elements in the objective function:

Type	Name	Minimum	Maximum
Response	LARGE BLOCK REN. SEC.	3	7
Response	LARGE CRANK REN. SEC.	3	7
Response	SMALL CRANK REN. SEC.	3	6
Response	LARGE PISTON REN. SEC.	3	6
Response	LARGE BED REN. SEC.	3	6
Response	LARGE 1^{ST} REP. SEC.	3	6
Response	SMALL 1^{ST} REP. SEC.	3	6
Response	LARGE 2^{ND} REP. SEC.	7	14
Response	SMALL 2^{ND} REP. SEC.	3	6
Response	TESTING SECTION	10	14
Response	PACKING SECTION	5	7

Table 4.5-4. List of factors.

In this model, we assume that the relative importance of each fitness function term is the same. The evaluation parameters are; Maximum replication per solution is five, and number of parents per generation is three.

When there is no improvement more than 5% or in the last 30 generation. This compares the best fitness score of the current generation to the best score of the previous N^{th} generation. If there is not desired improvement in the fitness score between these generations, algorithm stops making runs.

Best fitness - the best fitness score seen so far in any generation.

Best fitness in this generation - the best fitness score of a child in that generation.

Parents' average fitness - the average fitness of all the parents of the generation.

Children's average fitness - the average fitness of all the children of the generation.

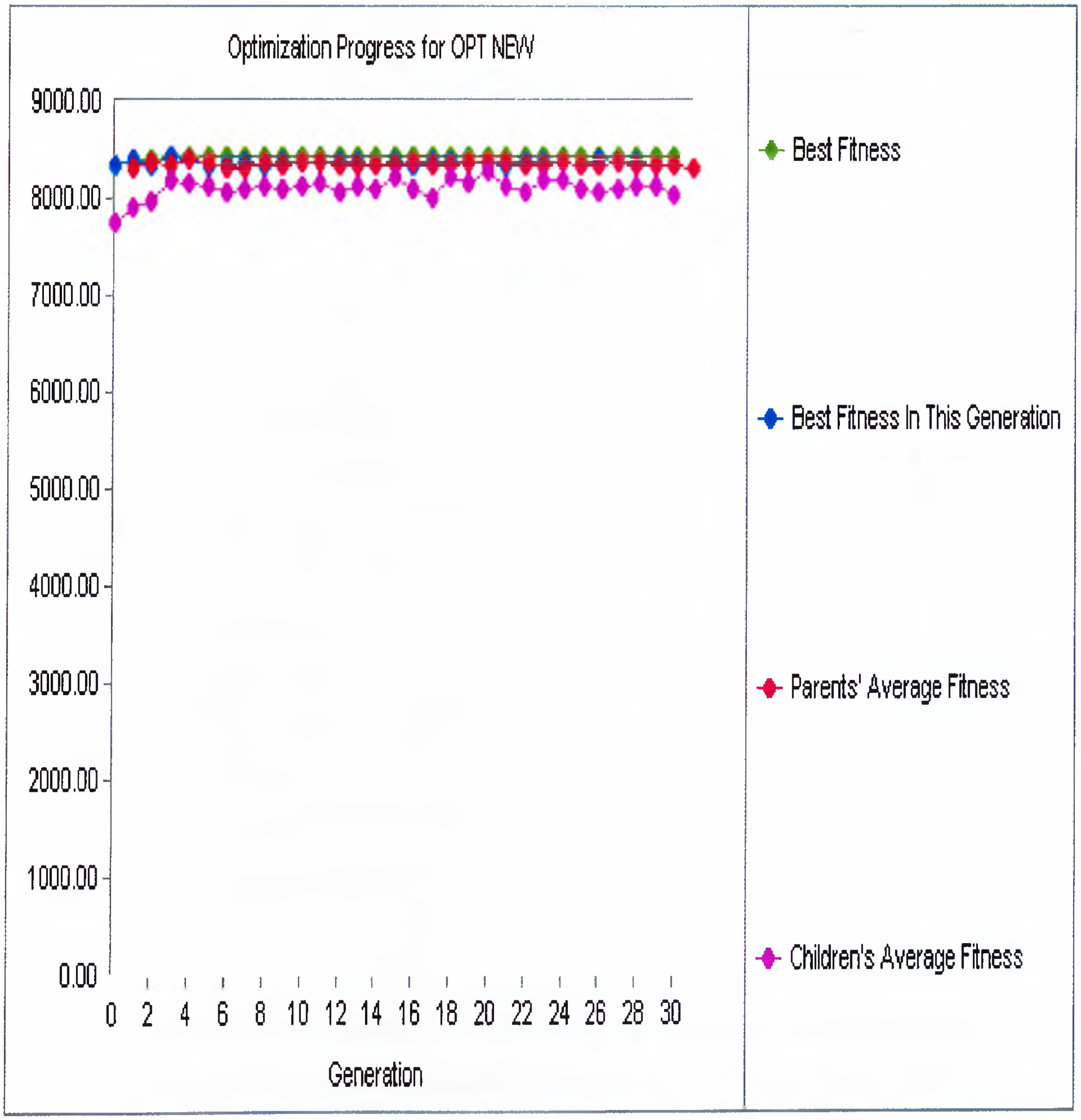


Figure 4.5-1 Progress graph motor renovation unit.

In our implementation, the algorithm stops the search when the termination criteria is met (624 runs and 32 generations later). When the progress graph is plotted, we observed that as the optimisation process proceed, the response value increases and all the lines converge, indicating that this is *the best solution that the evaluation algorithm* could find. According to graph (see *Figure 4.5-1*) and the summary results (see *Table 4.5-6*), the best score is obtained at the 28st generation. The results indicate that our objective function increases of the section's capacities. Further discussions are made in the following section.

Name	Existing Capacity	Optimised Capacity	Change
LARGE BLOCK REN. SEC.	3	6	+3
LARGE CRANK REN. SEC.	3	6	+3
SMALL CRANK REN. SEC.	3	4	+1
LARGE PISTON REN. SEC.	3	5	+2
LARGE BED REN. SEC.	3	4	+1
LARGE 1ST REP. SEC.	2	3	+1
SMALL 1ST REP. SEC.	3	4	+1
LARGE 2ND REP. SEC.	8	12	+4
SMALL 2ND REP. SEC.	3	5	+2
TESTING SECTION	14	13	-1
PACKING SECTION	7	1	-1

Table 4.5-5. Capacity changes.

If all the lines on the graph did not converge, the algorithm would continue to search up to defined maximum number of generations. There are times when the best so far and the best of a generation might deviate due to an outlier (a response value that is very different than other values).

4.5.3. Results and Discussions

In this section, a comparison between the existing and improved system optimised by GA is made to see if any improvement is obtained due to the changes in the system. The improvement is obtained in the numbers of the renovated motors. Overall results are given in *Table 4.5-6.* (Also see Appendix B.3 for more detailed results)

Throughput (number of motors)	**Existing System**	**Improved System**	**Improvement (%)**
Large motor	2331	3047	+30%
Small motor	3189	3476	+12%
Unimog motor	1852	1929	+4%

Table 4.5-6. Overall results of the optimisation procedure.

The throughput of the renovation unit improved is about 17%. The highest increase is observed for the large type motors (about 30%). The small motors renovation and unimog motors renovation increases 12% and 4%, respectively.

Even though this much of important is significant, the optimised system requires many additional resources. Specifically, the optimised system requires extra resources in nine sections and reduction in two sections (*Table 4.5-4).*

Also, we observed that the utilisations decrease at the sections that their capacities are increased and the utilisations increase at the sections that their capacities are decreased.

This additional requirement on resources decreases the probability of the application of the improved system.

Chapter 5

PRE-CONTROL & REPAIR SECTION

5.1 Purpose

The purpose of the pre-control & repair section is to increase the productivity of Army Depot's renovation units. (By increasing the number of the renovated & repaired vehicle motors.)

a. To increase the level of movement capacity of the combat units.

b. Designing new supportive logistics systems for expansion to meet demands.

c. Promote combat efficiency of the armed services as a whole by prevention of unnecessary duplication of facilities.

5.2. Introduction

The pre-control and repair section checks only large and small type of vehicle motors before entering the motor renovation unit. The unimog type of vehicle motors will not be handled in this section, since this kind of vehicle motors will be disposed up to 5 years in the Turkish Army. The control mechanism defines the status of the vehicle motors and the type of breakdowns. To admit vehicle motors, maintenance technicians examine if the motors are renovated 1 years ago or unused or not so much used according to the their physical appearances.

First, the vehicle motor is tested to define the type of breakdown. As a result of testing procedure, if the breakdown is undefined the vehicle motor is sent to the renovation unit. Otherwise if the breakdown is defined, the maintenance technicians decide according to the criteria to repair the breakdown at the subdivision of the pre-control & repair section or to send the renovation unit. (See *Figure 5.2-1*)

After repair operation, the vehicle motor tested at the bremze (testing) facility for final inspection. Then it is either sent to main depot's mixed goods accountancy or it is sent back to renovation unit.

5.2.1. Advantages of the pre-control & repair section:

Using the pre-control and repair section, we expect to:

a. Increase the total number of renovated and repaired vehicle motors.

b. Prevent unnecessary duplication of renovation activities.

c. Improve the availability of the spare parts. Since this section is located in main repair depot.

d. Decrease the time needs and increase cost effectiveness.

5.2.2. Disadvantages of the pre-control & repair section:

a. If this facility does not define the type of breakdown at vehicle motor then the cost for renovation increases. Due to additional maintenance works. Also this section require extra 4 people and some equipment.

b. If the motor is only operated in this section, the later problems at the repaired motor (but not renovated motors) can cause much more damage.

c. In order to obtain the benefit from this section there should be better recordings and information about motors. This requires an additional study and time.

5.2.3. The subdivisions of pre-control & repair section and technical data

There are three main subdivisions:

Control facility: selects the motors that will be repaired and defined the type of breakdown.

In 1011 Main Repair Depot, based on the past experiences, the technicians expect that 20-25% of previously renovated small type of vehicle motors and 1-3% of large type of vehicle motors come again to the renovation unit. The process of defining the type of breakdown can take minimum 45 minutes, average 60 minutes, and maximum 75 minutes. From the historical data, we know that breakdown probability is 90% for large type of vehicle motors and 85% for small type of vehicle motors.

The percentage of the selection rate for the small type of vehicle motors is very high. This is because of the age of vehicle motors.

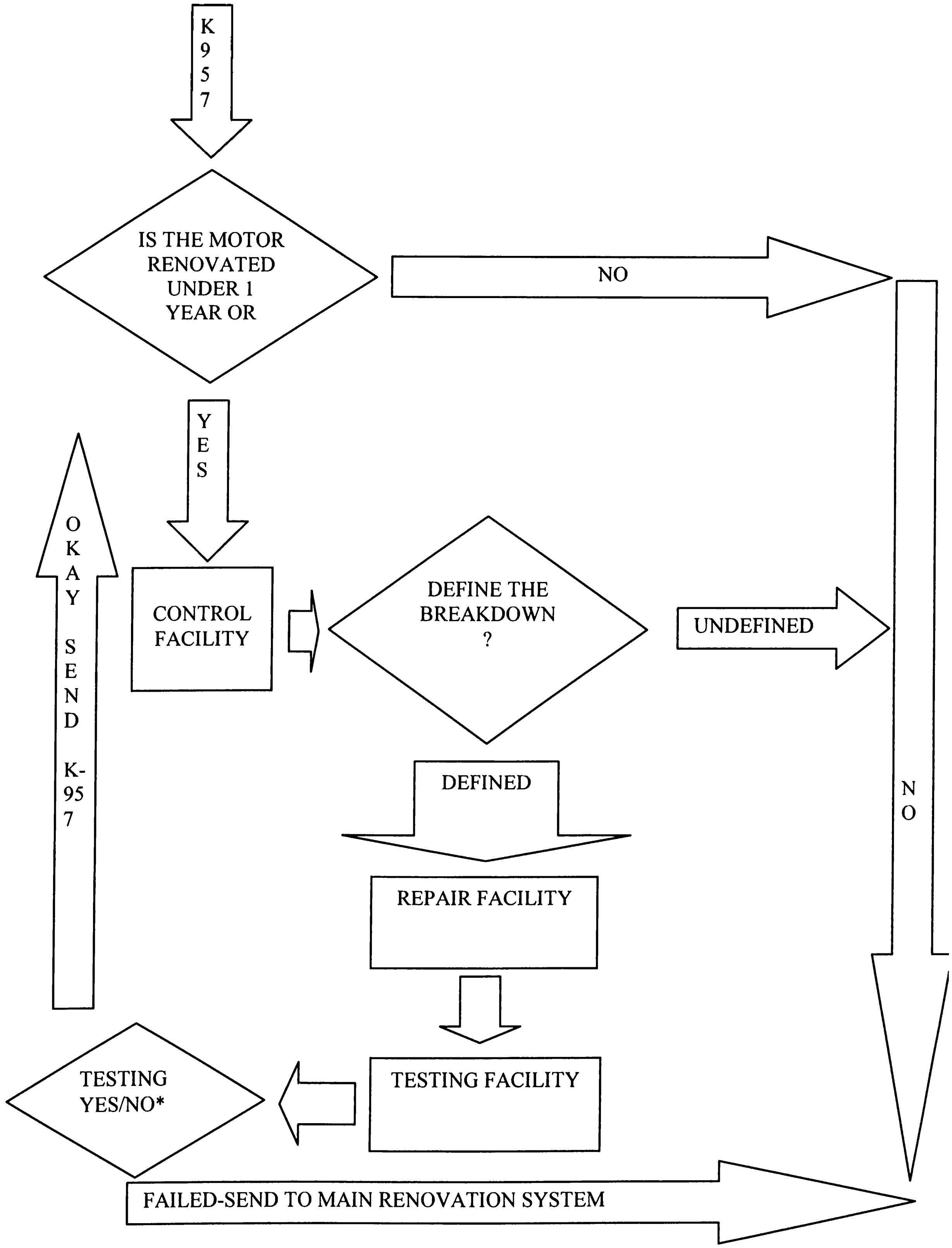
Repair facility: is used for repair process of the motor.

The technicians expect that the repair activity can take minimum 50 minutes, average 120 minutes, and maximum 150 minutes for large vehicle motors and minimum 60 minutes, average 125 minutes, and maximum 160 minutes for small vehicle motors.

Testing facility: tests the vehicle motors according to standards of renovation unit.

The testing operation is expected to take 55 minutes, average 60 minutes, and maximum 70 minutes, and the probability of success is 97 %.

Figure 5.2-1. The Logical model of the pre-control & repair section.



5.2.4. Simulation code of the pre-control & repair section.

The simulation code for pre-control & repair section is developed in Automod 9.0 (1999). The code is given in Appendix C.1.

5.3. The Results

In this section, we compare the existing system with the pre-control and repair section added to existing system to see if there is some improvement in the system performance due to this control facility. Ten replications are taken and the results are presented in *Table 5.3-1.* In this table, each row represents the different replication results. The cumulative sums, averages, standard deviations, and confidence intervals (α=0.05) are showed at the end of the each column. The second, third, and the fifth columns show the number of renovated motors and sixth, and seventh columns show the number of repaired motors. To obtain direct comparisons with the existing system, the second and fifth columns, and third and sixth columns are combined in *Table 5.3-2.*

When the results (given in *Table 5.3-1* and 3.3.2-1) are compared, it is clear that there are improvements about 2.5% (the amount of improvement is found as =Existing sys. results-Proposed sys. results / Existing sys. results) for large motors and 29% for small motors. The common random numbers (CRN) are used to increase the precision in comparisons. Namely, the paired-t test is applied and obtained the point and interval estimates on the difference in the mean performance of the proposed system and the existing system (Eg. $\theta_1-\theta_2$).

Replication	Large motors	Small motors	Unimog motors	Large repaired	Small repaired
1	2297	3121	1855	64	979
2	2354	3158	1893	64	973
3	2330	3137	1893	60	960
4	2329	3132	1856	47	987
5	2335	3149	1876	64	975
6	2344	3140	1826	61	1024
7	2321	3112	1844	66	1007
8	2345	3134	1866	66	1014
9	2340	3141	1836	62	988
10	2321	3179	1855	72	976
CUM. SUM	23316	31403	14909	492	7919
AVERAGE	2331,875	3135,375	1863,625	61,5	989,875
STAN. DEV	17,58601311	14,57921711	23,35402994	6,6440	22,5669
CON. INT.	12,18624337	10,10268141	16,18319574	4,6039	15,6377

Table 5.3-1 The number of renovated & repaired vehicle motors.

Replication	Large type of motors	Small type of motors	Unimog type of motors
1	2361	4100	1855
2	2418	4131	1893
3	2390	4097	1893
4	2376	4119	1856
5	2399	4124	1876
6	2405	4164	1826
7	2387	4119	1844
8	2411	4148	1866
9	2402	4129	1836
10	2393	4155	1855
CUM SUM=	23942	41286	14909
AVERAGE=	2393,375	4125,25	1863,625
STAN.DEV=	18,84476055	22,57527092	23,35402994
CON.INT=	11,67987168	13,99202005	14,47469031

Table 5.3-2 The combined number of renovated & repaired vehicle motors.

5.3.1. Comparison of the number of renovated & repaired large type motors.

The paired–t approach is applied to test if the differences between the systems is significant. The results are given in *Table 5.3.1-1.* In this table, the rows represent the results for the number of renovated and repaired large type motors at the each replication. This table also displays the average differences between the systems, standard deviation, and confidence interval (α=0.05) on the mean differences.

There is on the average 2.50% increase in the number of renovated and repaired motors due to the proposed changes. *Figure 5.3.1-1* also displays the differences between these two systems for each replication. The paired-t test results show that the proposed system is better than the existing system (because the average difference plus and minus confidence interval (-58.3±14.91) does not include the zero).

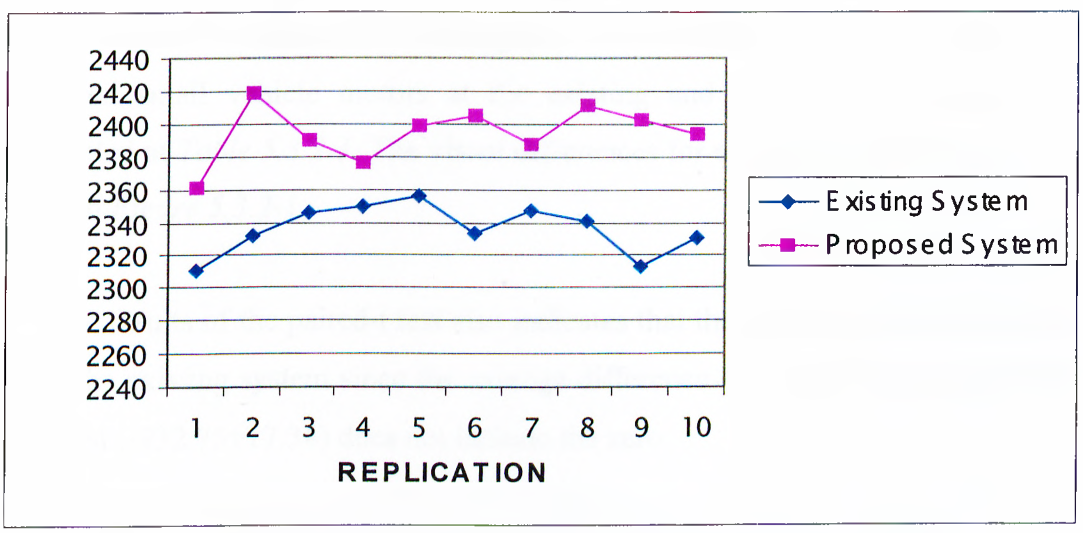


Figure 5.3.1-1 The number of renovated & repaired large type motors.

LARGE MOTOR	EXISTING SYSTEM	PROPOSED SYSTEM	DIFFERENCE
1	2310	2361	-51
2	2332	2418	-86
3	2346	2390	-44
4	2350	2376	-26
5	2356	2399	-43
6	2333	2405	-72
7	2347	2387	-40
8	2341	2411	-70
9	2313	2402	-89
10	2331	2393	-62
	CHANGE	-2.50%	INCREASE
AVERAGE. DIFF	-58.3		
STAND. DEV	6.598063689		
VARIANCE	43.53444444		
95% CI. T-test	14.91162394		

Table 5.3.1-1 The number of renovated & repaired large type motors.

5.3.2. Comparison of the number of renovated & repaired small type motors

When the analysis is done for the small type of motors, we observe that there is on average 29% increase in throughput. The number of the renovated and repaired small vehicle motors at the existing and the proposed system are presented at *Table 5.3.2-1*. The visual differences for each replication can also be seen in *Figure 5.3.2-1.*

The results of the paired-t test also indicates that the proposed system is better than the existing system since the average difference plus and minus confidence interval (-932.75±17.34) does not include the zero.

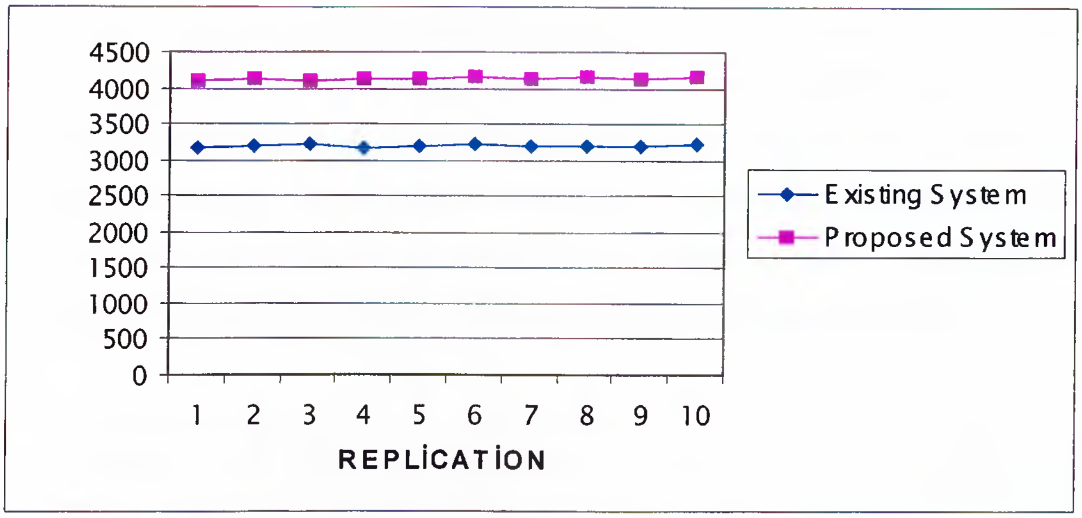


Figure 5.3.3-2 The number of renovated & repaired small type motors.

SMALL TYPE	EXISTING SYSTEM	PROPOSED SYSTEM	DIFFERENCE
1	3175	4100	-925
2	3204	4131	-927
3	3212	4097	-885
4	3168	4119	-951
5	3204	4124	-920
6	3216	4164	-948
7	3176	4119	-943
8	3185	4148	-963
9	3184	4129	-945
10	3218	4155	-937
	CHANGE	-29%	INCREASE
AVERAGE. DIFF	-932.75		
STAND. DEV	7.67323735		
VARIANCE	58.87857143		
95% CI. T-test	17.34151641		

Table 5.3.2-1. The number of renovated & repaired small type vehicle motors.

5.3.3. Comparison of the number of renovated & repaired unimog type motors.

In the unimog type motors case, we observe that there is on the average 1.7 % increase in the number of motors (*Table 5.3.3-1)*. This table presents the differences at each replication and *Figure 5.3.3-1* displays the visual differences. The paired-t test results show that there is difference between the systems for the number of the renovated unimog motors, since the average difference plus and minus confidence interval (-32.375 ±28.77) does not include the zero. Even though, the unimog type of vehicle motors is not operated at the pre-control & repair section, we observed an insignificant increase in terms of throughput, because of the proposed system's relaxation effect on the renovation unit.

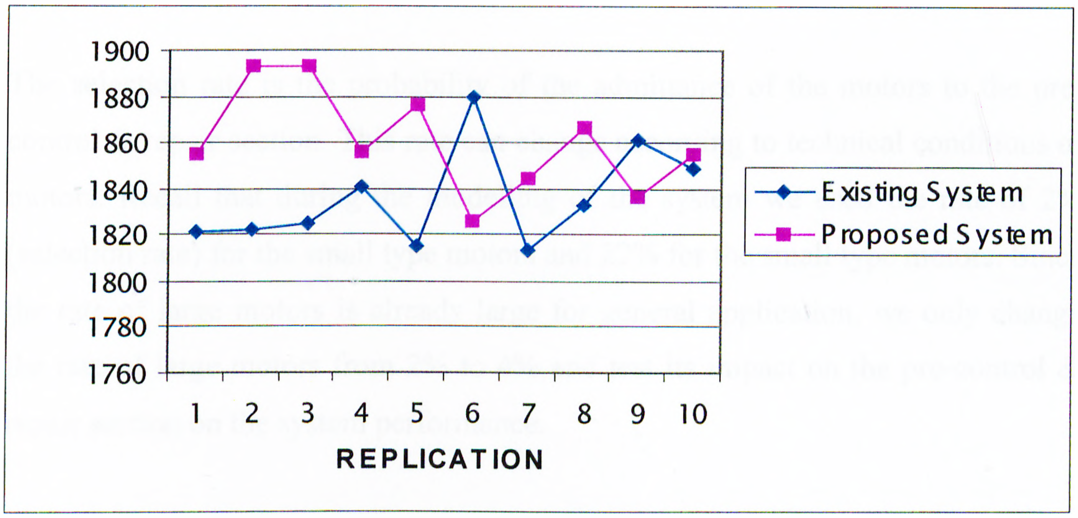


Figure 5.3.3-1 The number of renovated & repaired small type.

UNIMOG TYPE	EXISTING SYSTEM	PROPOSED SYSTEM	DIFFERENCE
1	1821	1855	-34
2	1822	1893	-71
3	1825	1893	-68
4	1841	1856	-15
5	1815	1876	-61
6	1880	1826	54
7	1813	1844	-31
8	1833	1866	-33
9	1861	1836	25
10	1849	1855	-6
	CHANGE	-1.7%	INCREASE
AVERAGE. DIFF	-32.375		
STAND. DEV	12.73234071		
VARIANCE	162.1125		
95% CI. T-test	28.77509001		

Table 5.3.3-1. The number of renovated & repaired unimog type vehicle motors.

5.3.4. The alternative situations:

In this section, we further investigate the performance of the pre-control & repair section to answer some more what if questions (ie. what happens if the parameters or input variables are changed?).

5.3.4.1. The changes in selection rate.

The selection rate is the probability of the admittance of the motors to the pre-control & repair section. This rate can change according to technical conditions of motors. Recall that during the modelling of the system we used the rate of 2% (selection rate) for the small type motors and 22% for the small type motors. Since the rate of large motors is already large for general application, we only change the rate of large motors from 2% to 4% and test its impact on the pre-control & repair section on the system performance.

Selection rate	Large Type Motor	Small Type Motor	Utilisation of C. F.
2	64	979	0.709
4	110	939	0.708
6	184	977	0.76
8	238	1042	0.83
10	291	1019	0.863
12	345	975	0.859
14	414	977	0.91
16	467	999	0.959
18	519	985	0.975
20	570	995	0.998
22	613	944	0.999
24	630	935	0.999
26	672	863	0.999
Average	393.6153846	971.4615385	
Stan. Dev	204.9941212	44.50021608	
C.I. for 0.05	111.433861	24.19011269	

Table 5.3.4-1 The maximum selection rate and the number of repaired motors.

The results of the simulation experiments are given in *Table 5.3.4-1*. In general, the number of repaired large motors is increased 71.8% (from 64 unit to 110 units). But there is no significant change for the number of renovated and repaired small and unimog type vehicle motors (See *Tables 5.3.1-1* and *5.3.3-1*). These results were normal, since we used the extra capacity (since the additional repair section has under utilisation) and we do not change the other type of motors` selection rates.

Next, we try to determine the effect of the selection rate on the pre-control & repair section. As expected, the throughput of the large motors increases at a decreasing rate (diminishing rate of return). We did not increase the selection rate for the small type motors since the operation times for both motors are almost same.

The results for the various selection rates are displayed at *Table 5.3.4-1*. We observed that the number of repaired large motors (after the rate of 20%) is increasing at a decreasing rate but the number of repaired small motors is decreasing (see *Figure 5.3.4-1*). Because the additional repair section's utilisation reaches one. Also the utility of the control facility (the other additional section) is reaching the upper bound (see *Figure 5.3.4-2).*

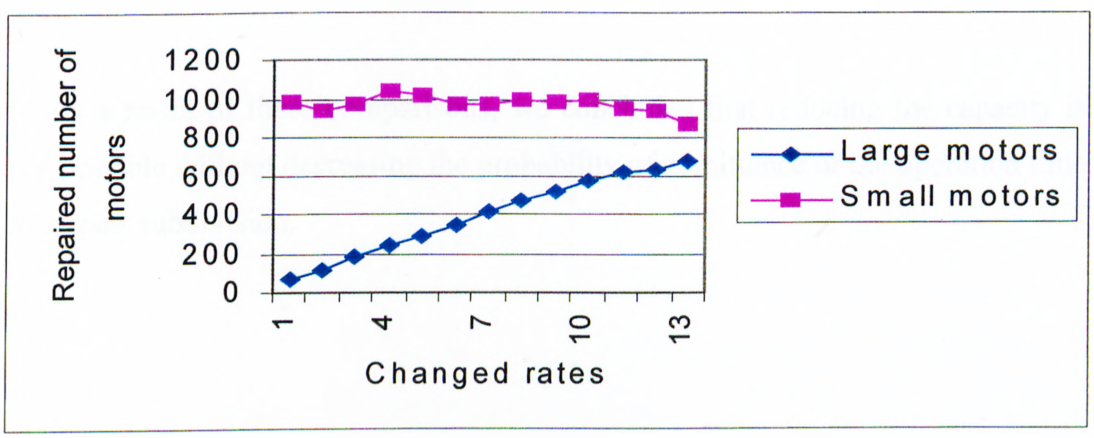


Figure 5.3.4-1 The observations at the number of repaired motors.

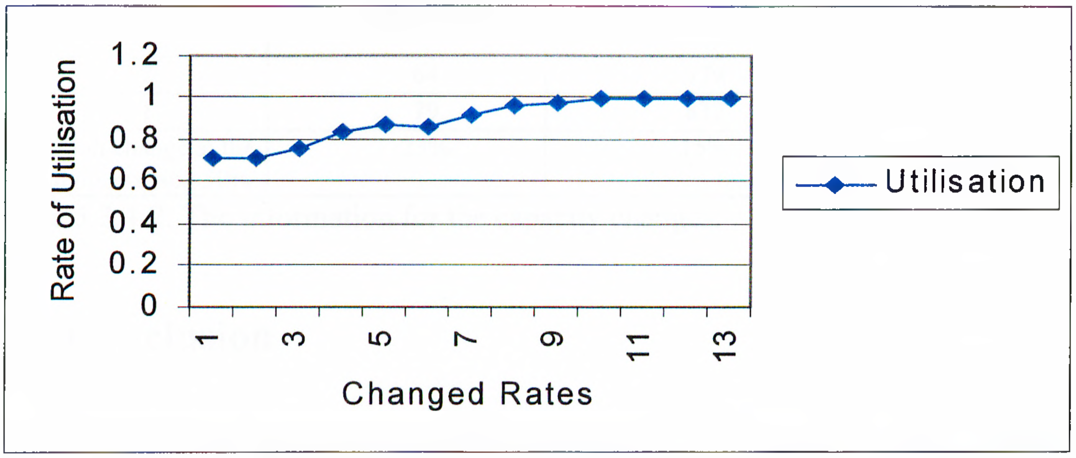


Figure 5.3.4-2 The utility of the "Addrepair" Subdivision at different selection rates.

5.3.4.2. Decreasing the capacity of the resources.

In this section, we decreased the capacity of additional repair section.

The results are presented in *Table 5.3.4-2*. In this table, the rows represent the number of repaired motors and the utility of the repair activity at each replication. We observed that when the capacity of the repair subdivision decreased by one unit, the amount of repaired large motors increases 23% and small motors decreases about 15% are decreases, and the subdivision's utility is increases about 75.6%.

As a result of these comparisons, we concluded that reducing the capacity is not possible without decreasing the probability of admittance or the operation time for repair subdivision.

Capacity	Large Motor	Small Motor	Utilisation of Addrepair
2	64	979	0.567
1	79	832	0.996
Utilisation change at the capacity decreased 50%	23%	15%	75%

Table 5.3.4-2. The information for the capacity changes.

5.4. Conclusion

In this chapter, we propose a new "pre-control & repair section" to improve the number of the renovated and repaired motors. The results indicate that the proposed system increases the number of the renovated and repaired motors. In short, we investigated the cost effectiveness of the pre-control and repair section implementation (see Appendix D-1 for implementation cost of the proposed system 1). As seen in the *Table 5.4-1*, when the proposed system is implemented, the average cost of renovation of the large motors and small motors decrease 5 Million. TL and 6 Million TL, respectively. The effect of the implementation of the proposed system on the total cost is about 9.7% increase. Therefore, we can conclude that the pre-control and repair section should be captivated in the plans to support to maintenance for 1011 Main Repair Depot.

Large Motors	Total Cost	Average Productivity	Average Cost Per Motor
Existing System	605 Billion. TL	2331 (per year)	259 Million. TL
Proposed System 1	608.5 Billion. TL	2393 (per year)	254 Million. TL
Small Motors			
Existing System	247 Billion. TL	3190 (per year)	77 Million. TL
Proposed System 1	296 Billion. TL	4125 (per year)	71 Million. TL

Table 5.4-1. Average cost changes for the proposed system 1

Chapter 6

COMBINED PARALLEL RESOURCES SYSTEM

6.1. Purpose

The purpose of the combined resources system is to decrease the lost capacity and to increase the productivity of Army Depot's renovation units.

6.2. Introduction

Recall that the existing system works as a flow shop. On the other hand, the proposed system is a generalisation of the flow shop and the parallel machine environments. Instead of m machines in series, there are s stages in series with a number of machines in parallel at each state. Each job has to be processed first at stage 1, then at stage 2, and so on. A stage functions as a bank of parallel machines; at each stage job j requires only one machine and, usually, any machine can process any job. The queues between the various stages operate under FIFO discipline [26]. In the combined parallel resources system, the number of the machines and workers are the same as the existing system. For this system, we do not increase the number of resources. We only form a common queue for some of the identical resources instead of each one having a separate queue formed in front of it. It is theoretically shown that a common queue approach generally decreases average waiting time in queues. These resources are block renovation sections, crank renovation sections, cylinder bed preparation sections, piston renovation sections, repair groups, and main assemble groups.

First, the motors are grouped with respect to their types, before admitting them to the system. The vehicle motor is disassembled, washed, and separated as in the existing system, then the grouped motors are sent to renovation sections.

When different types of vehicle motors arrive at the renovation sections, some of the machines at renovation sections should be adjusted with respect to the type of motor. These sections are block renovation sections, crank renovation sections, cylinder bed preparation sections and piston renovation sections. These four sections require a major set-up, approximately 90 minutes.

Repair groups and main assemble groups do not need any set-up.

After renovation operation, the vehicle motor is tested at the bremze (testing) facility for final inspection and then it is packed. Afterwards, it is sent to main depot (mixed goods accountancy).

6.2.1. Advantages and disadvantages of the combined parallel resources system:

Let us now discuss about the benefits from the combined parallel resources system. These are:

a. increase the total number of renovated vehicle motors (ie. increasing the throughput).

b. decrease the bottleneck situations of the renovation activities due to alternative resources available for each operation.

c. decrease the lost capacity of the resources (under utilisation).

d. Reduce the breakdown impact of the system.

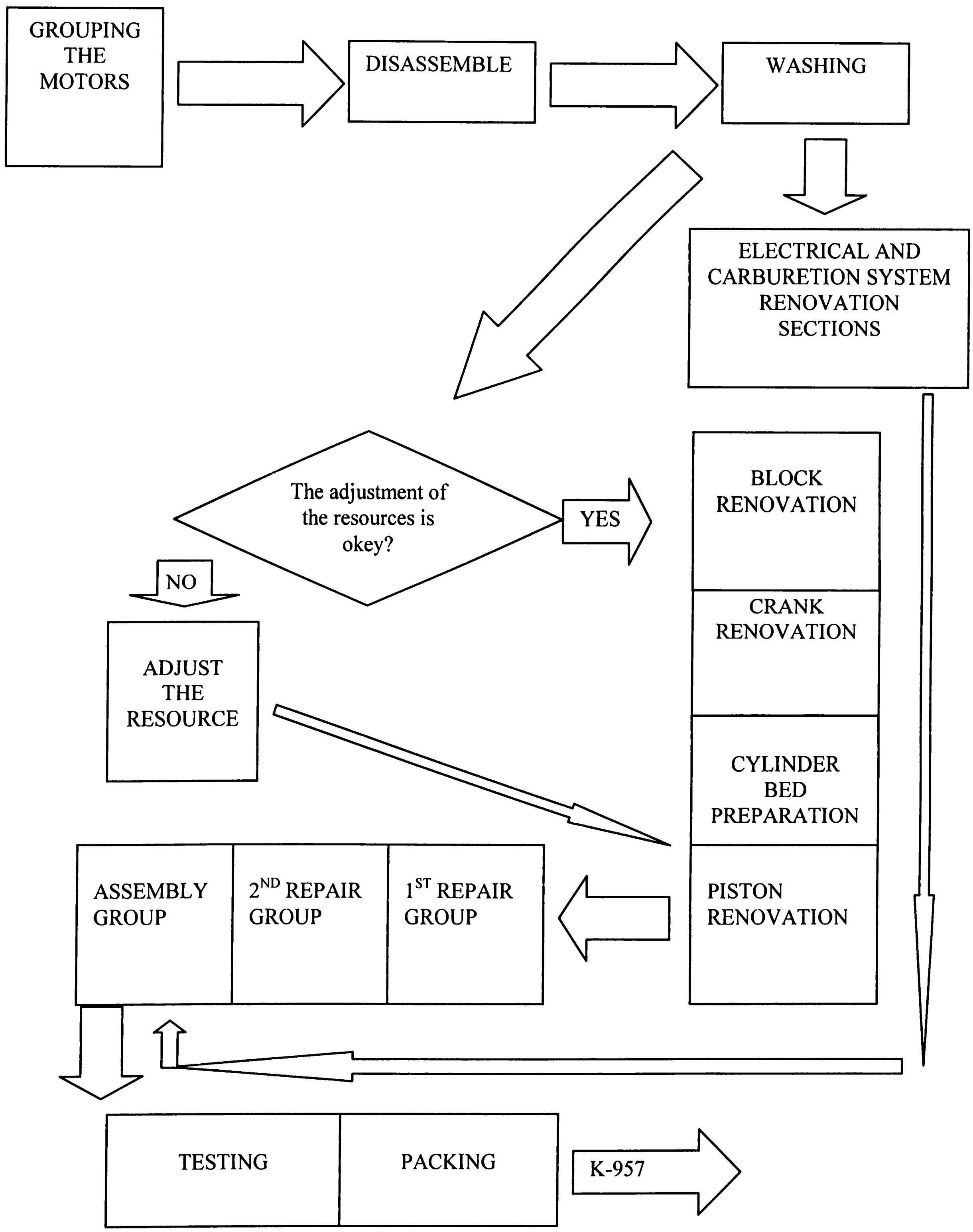
In addition to these benefits the proposed system might incur some additional cost as well:

a. More complicated scheduling problem might appear.

b. Additional set-up is required for some operations.

The effects of some of these factors will be tested in the next section.

Figure 6.2.1. The logical system of the combined parallel resources system.



6.2.2. Simulation code of the combined parallel resources system.

The simulation code for combined parallel resources system is developed in Automod 9.0 (1999). The code is again given in Appendix C.2.

6.3. The Results

In this section, a comparison between the existing and proposed system is made to see if any improvement is obtained due to the combined parallel resources system. In general, an improved is observed in the numbers of the renovated motors. Overall results based on the ten replications for the proposed system are given in *Table 6.3-1.*

REPLICATION	LARGE TYPE	SMALL TYPE	UNIMOG TYPE
1	2446	3378	2030
2	2670	3389	1978
3	2634	3367	1984
4	2502	3349	1979
5	2611	3513	2020
6	2554	3540	1980
7	2712	3204	2003
8	2304	3553	2239
9	2786	3333	1815
10	2304	3357	2123
CUM. SUM	25523	33983	16213
AVERAGE	2552,3	3398,3	2015,1
STAN. DEV	163,7179485	107,9084489	109,0794308
CON. INT.	101,4714209	66,88102151	67,60678922

Table 6.3-1. The results of the combined parallel resources system.

These results will be also compared the results of the existing system (given in *Table 6.3-1 and 3.3.2-1).* Recall that the common random numbers (CRN) are used and the common random input sequences are used to increase the precision in the estimation of the difference between alternative systems.

6.3.1. Comparison of the Existing & Proposed Systems

The Paired-t test is applied to the difference in the mean performance of between the proposed and existing system. The results are presented for each for each motor type in detail in the next sections.

6.3.1.1. Comparison of the existing and proposed systems for large motors.

The results are given in *Table 6.3-1*. In this table, the rows represent the results for the number of renovated large motors at each replication. The table also displays the average differences between systems, standard deviation, and confidence interval on the mean differences

The results indicate that the throughput of the proposed system improve about 9.37% (=216.4/2335.9). *Figure 6.3-1* displays the differences between the systems for each replication. The Paired-t test results show that, the performance of the proposed system is statistically better than the existing system, as the confidence interval on the difference between mean performances (216.4±79.84) does not include zero.

LARGE MOTOR	EXISTING SYSTEM	PROPOSED SYSTEM	DIFFERENCE
1	2310	2446	-136
2	2332	2670	-338
3	2346	2634	-288
4	2350	2502	-152
5	2356	2611	-255
6	2333	2554	-221
7	2347	2612	-265
8	2341	2404	-63
9	2313	2686	-373
10	2331	2404	-73
	CHANGE	9.37%	INCREASE
AVERAGE. DIFF	-216.4		
STAND. DEV	33.83233168		
VARIANCE	1144.626667		
95% C.I T-test	79.84430276		

Table 6.3.1-1. The number of renovated large type motors for both systems

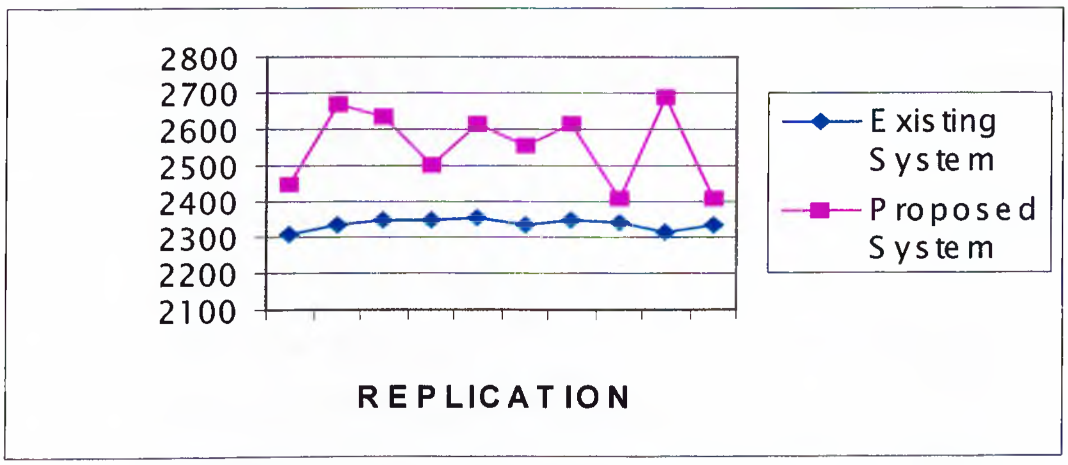


Figure 6.3.1.1. The differences between systems, for large motors.

6.3.1.2. Comparison of the existing and proposed systems for small motors.

When the same procedure is implemented for the small vehicle motors (with respect to the number of renovated small motors), we observed 6.90% increase in throughput (*see Table 6.3.1-2*). *Figure 6.3.1-2* shows the improvement for each replication.

SMALL TYPE	EXISTING SYSTEM	PROPOSED SYSTEM	DIFFERENCE
1	3175	3378	-203
2	3204	3389	-185
3	3212	3367	-155
4	3168	3349	-181
5	3204	3513	-309
6	3216	3490	-274
7	3176	3264	-88
8	3185	3553	-368
9	3184	3333	-149
10	3218	3357	-139
	CHANGE	-%6.90	INCREASE
AVERAGE. DIFF	-220.375		
STAND. DEV	28.64159687		
VARIANCE	820.3410714		
95% CI. T-test	67.59416862		

Table 6.3.1-2. The number of renovated small motors for both systems.

The Paired-t test results indicate that the proposed system is statistically better than the existing system, as the average difference plus and minus confidence interval (216.4±79.84) does not include zero.

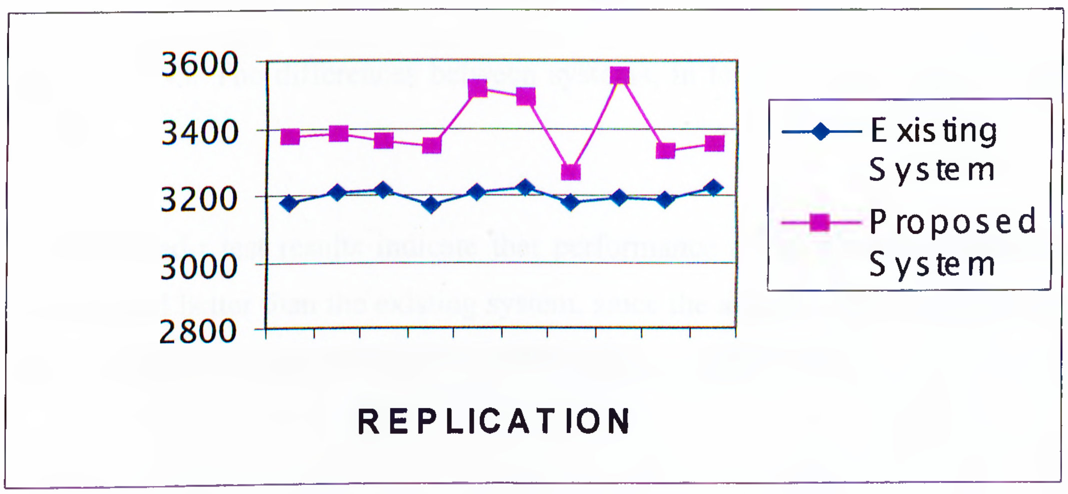


Figure 6.3.1-2. The differences between system, for small motors.

6.3.1.3. Comparison of the existing and proposed systems for unimog motors.

When the same procedure is applied to the unimog vehicle motors (with respect to the number of renovated unimog motors), there is 10.67% increase (*see Table 6.3.1-3*). Also *Figure 6.3.1-3* shows the differences between the systems for each replication visually.

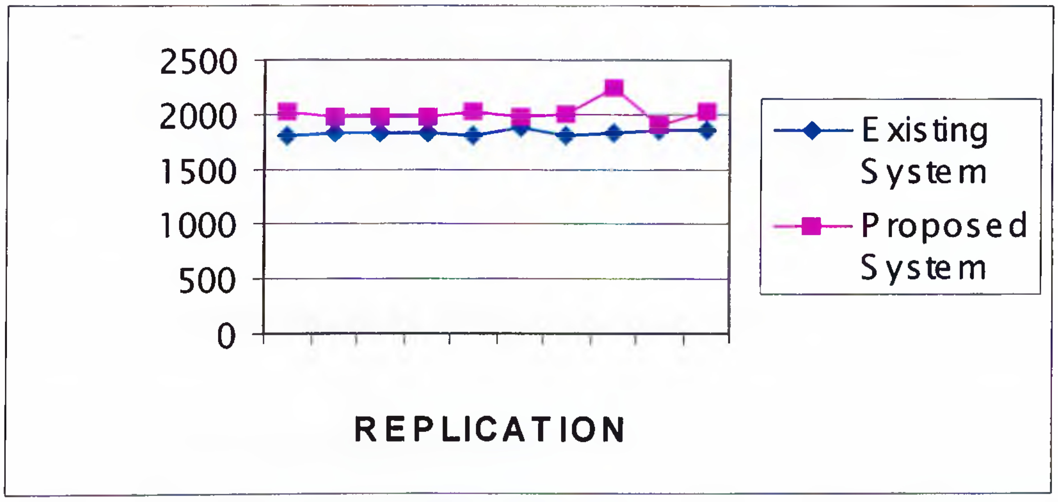


Figure 6.3.1-3. The differences between systems, in terms of renovated Unimog motors.

The Paired-t test results indicate that performance of the proposed system is different and better than the existing system, since the average difference plus and minus confidence interval (195.375±79.84) does not include zero.

UNIMOG TYPE	EXISTING SYSTEM	PROPOSED SYSTEM	DIFFERENCE
1	1821	2030	-209
2	1822	1978	-156
3	1825	1984	-159
4	1841	1979	-138
5	1815	2020	-205
6	1880	1980	-100
7	1813	2003	-190
8	1833	2239	-406
9	1861	1915	-54
10	1849	2023	-174
	CHANGE	-10.67%	INCREASE
AVERAGE. DIFF	-195.375		
STAND. DEV	29.26037667		
VARIANCE	856.1696429		
95% CI. T-test	69.05448894		

Table 6.3.1-3. The number of renovated unimog motors for both systems.

6.3.1.4. Comparison of the queue lengths.

In this section we investigate the effects of the proposed system on the queue lengths. The reason for comparing the queue lengths is to investigate final situation of the bottlenecks.

a. Block renovation section queue.

The simulation results are given in *Table 6.3.1-4*. In this table, to get direct comparisons with the proposed system, the second column is derived from adding each type of the block renovation queues.

The results of the paired-t test indicate that the proposed system is statistically different with the existing system but there is about 61% decrease at the queue length of the block renovation section. *Figure 6.3.1-4* also displays the difference at the queue for each replication.

REPLICATION	EXISTING SYSTEM.			PROPOSED SYSTEM			DIFFERENCE		
1		105			33			72	
2		102			56			46	
3		136			26			110	
4		21			15			6	
5		158			32			126	
6		2			60			-58	
7		7			0			7	
8		117			27			90	
9		143			32			111	
10		125			41			84	
AVERAGE=		91.6			32.2			59.4	
VARIANCE=								343.5	
CONF.INT(t)		41.89		The proposed model is statistically different than the existing system.					

Table 6.3.1-4. Comparison of the block renovation section`s queue lengths.

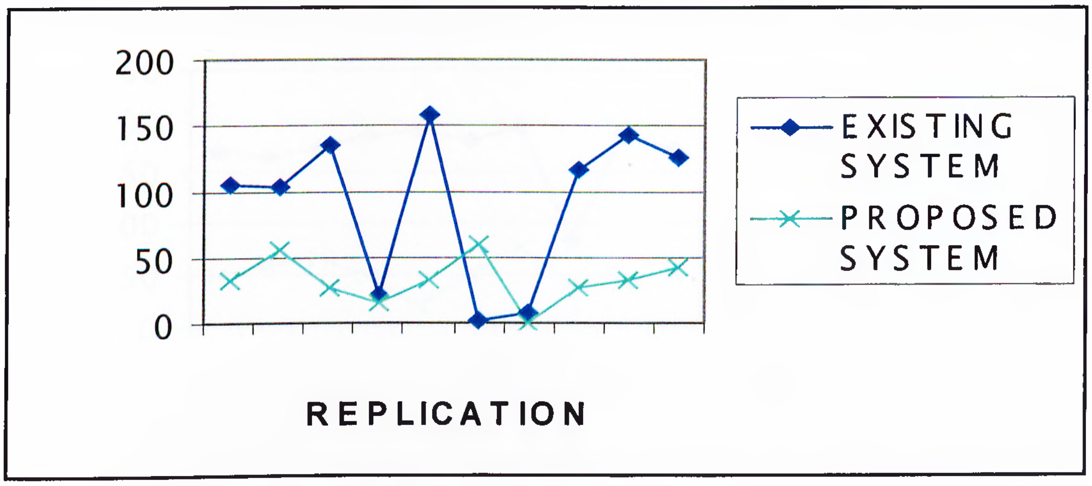


Figure 6.3.1-4. Queue length differences between systems.

b. Crank renovation section queue lengths.

For the crank renovation section, the simulation results are given in *Table 6.3.1-5. Figure 6.3.1-5* is also depicted as a visual aid to show the differences. In general, the improvement is about 69% (91-32 / 91). This is also testified by the paired t-test.

REPLICATION	EXISTING SYSTEM.	PROPOSED SYS.	DIFFERENCE
1	171	50	121
2	165	60	105
3	175	67	108
4	185	48	137
5	191	19	172
6	181	51	130
7	195	77	118
8	83	40	43
9	186	42	144
10	172	53	119
AVERAGE=	170.4	50.7	119.7
VARIANCE=			110.8
CONF.INT	24.84	The proposed model is statistically different than the existing system.	

Table 6.3.1-5. Comparison of the crank renovation section queue lengths.

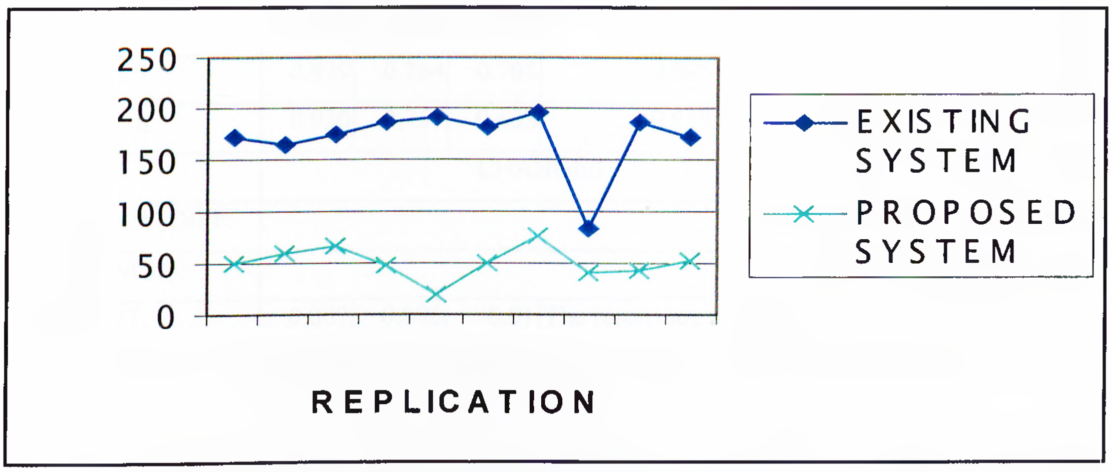


Figure 6.3.1-5. Queue length differences between systems.

6.3.1.5. Comparison of the Utilisations.

We also look at the utilisation to see positive effects of the proposed system on the system performances. Recall that, the proposed system combined the same type of resources in a parallel machine environment. Therefore some departments utilisations of some departments are changed. In the following sub-sections, we observed and investigated the differences of the systems.

a. Block Renovation Section:

The simulation results are given in *Table 6.3.1-6*. In this table, the utilisations at the existing system are shown separately so that one can make a direct comparison between the systems.

REPLICATION	EXISTING SYSTEM.			PROPOSED SYSTEM	DIFFERENCE		
1	0.954	0.783	0.703	0.874	0.080	-0.09	-0.17
2	0.926	0.753	0.687	0.864	0.062	-0.11	-0.18
3	0.934	0.763	0.687	0.867	0.067	-0.1	-0.18
4	0.938	0.764	0.700	0.86	0.078	-0.1	-0.16
5	0.924	0.762	0.683	0.868	0.056	-0.11	-0.19
6	0.933	0.759	0.723	0.874	0.059	-0.12	-0.15
7	0.939	0.764	0.706	0.865	0.074	-0.1	-0.16
8	0.930	0.757	0.682	0.859	0.071	-0.1	-0.18
	CHANGES				0.073	-0.14	-0.24
AVERAGE. DIFF.					0.068	-0.1	-0.17
VARIANCE					1E-05	7E-06	2E-05
CON. INT.	0.007	0.006	0.01	The results are significant			

Table 6.3.1-6. Comparison of the block renovation section`s utilisations.

On the averages, we observed about 7.8% decrease in the utilisation of the large block renovation section and combined parallel block renovation section. Also, we observed the deterioration in the utilisation performance 18% and 35% for the small and unimog motors, respectively according to existing system averages, due to common usage of the resource (see *Figure 6.3.1-6*). The Paired-t test implies that the simulation results are statistically significant and the proposed system is decreasing the effect of the bottleneck at the large type of block renovation section in the existing system. In short, the proposed system is increasing the average utilisation of the resources.

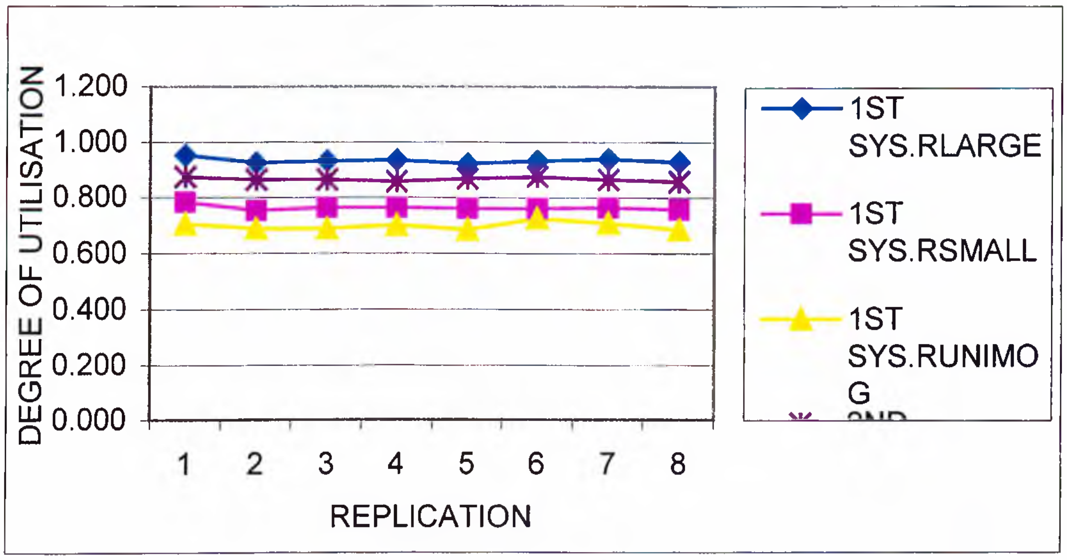


Figure 6.3.1-6. Utilisation differences between systems.

a. Crank Renovation Section:

When the same procedure is repeated for the crank renovation section, we observed average 3% and 2.9% decrease for the large and small crank renovation sections, respectively and 82% increase at the unimog type crank renovation section *(See Table 6.3.1-7.)*, with respect to utilisation of the combined parallel block renovation unit. (See *Figure 6.3.1-7*)

The Paired-t test show that the statistical significance and the proposed system decrease the effects of bottlenecks the large and small type of block in renovation sections of the existing system.

REPLICATION	EXISTING SYSTEM.			PROPOSED SYSTEM			DIFFERENCE		
1	0.994	0.996	0.538		0.963		0.031	0.033	-0.43
2	0.995	0.991	0.524		0.966		0.029	0.025	-0.44
3	1	0.998	0.527		0.971		0.029	0.027	-0.44
4	0.999	0.999	0.536		0.966		0.033	0.033	-0.43
5	0.996	0.996	0.52		0.969		0.027	0.027	-0.45
6	1	0.995	0.551		0.972		0.028	0.023	-0.42
7	0.999	1	0.54		0.97		0.029	0.03	-0.43
8	1	0.998	0.522		0.965		0.035	0.033	-0.44
	CHANGES						0.03	0.029	-0.82
AVERAGE. DIFF.							0.030	0.029	-0.44
VARIANCE							9E-07	2E-06	1E-05
CON. INT	The results are significant						0.002	0.003	0.009

Table 6.3.1-7 Comparison of the crank renovation section`s utilisations.

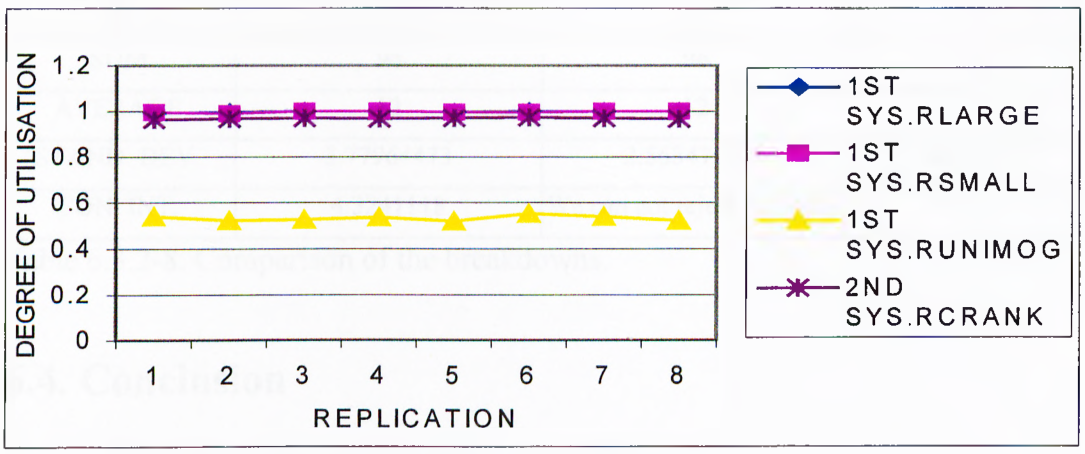


Figure 6.3.1-7. Utilisation differences between systems.

6.3.2.5. Comparison of the breakdowns in systems.

In real life, there are always random breakdowns that adversely effect the system performance. In this section, our objective is to see the effects of the breakdowns on the existing and proposed systems, in order to reveal basic differences between the systems.

The results are given in *Table 6.3.1-8.* In this table, the rows represent the number of breakdowns at each replication. There is no change in the number of the breakdowns and also the results are not statistically significant, since the same creation block is used to show the effects of the breakdown on the systems.

REPLICATION	EXISTING SYSTEM	PROPOSED SYSTEM	DIFFERENCES
1	9	14	-5
2	13	9	4
3	16	10	6
4	18	13	5
5	11	8	3
6	12	14	-2
7	6	14	-8
8	11	14	-3
SUM	96	96	0
AVERAGE	12	12	0
STAND. DEV.	3.77964473	2.563479778	5.182387756
CON. INT.	4.3241118	We can not reject the model	

Table 6.3.2-8. Comparison of the breakdowns.

6.4. Conclusion

We had mentioned that the basic mission of the Army Logistics System is to support the soldier in the field with what is needed, when, where, and in the condition and quantity required at minimum expenditure of resources.

The implementation of proposed system 2 increase the rate of renovated vehicle motors and decreases the effects of breakdowns due to usage flexibility of common resources.

Large Motors	Total Cost	Average Productivity	Average Cost Per Motor
Existing System	605 Billion. TL	2331 (per year)	259 Million. TL
Proposed System 2	630 Billion. TL	2552 (per year)	247 million. TL
Small Motors			
Existing System	247 Billion. TL	3190 (per year)	77 Million. TL
Proposed System 2	259 Billion. TL	3398 (per year)	76 Million. TL
Unimog Motors			
Existing System	470 Billion. TL	1852	254 Million. TL
Proposed System 2	500 Billion. TL	4125	248 Million. TL

Table 6.4-1. Average cost changes for the proposed system 1

Also, we investigated the cost effectiveness of the combined parallel resources proposed system 2 (see Appendix D-2 for implementation cost of the proposed system 1). As seen in the *Table 6.4-1*, when the proposed system is implemented, the average cost of renovation of the large motors, small motors, and unimog motors decrease 12 Million. TL, 1 Million TL, and 6 Million. TL, respectively. The implementation of the combined parallel resources system requires approximately 5 days, according to technical staff in the depot, and this implementation time only decreases the number of renovated motors in the unit for once.

Chapter 7:

INCREASED READY SPARE PART USAGE IN THE EXISTING SYSTEM.

The spare part is purchased product that processed or semi-processed out of depots.

7.1. Purpose

The purpose of the increasing the usage of the spare parts is to increase the capacity of the existing system. We expected that spare parts can

a. decrease the processing times.
b. increase the usage time of the renovated motors.
c. increase the quality of the renovated motors.
d. increase the level of movement capacity of the combat units.

7.2. Introduction

In the proposed system, the facilities are the same as the existing system. Recall that the existing system works as a flow shop. The ready spare part usage in the existing system decreases the processing times and the need for human resources. The effect of the ready spare part usage is applied on the large and small type motors.

The experimentations will be done by using the following ready spare parts.

Change 1: Using motor blocks on the small type of motors.

Change 2: Using cranks on the small type of motors.

Change 3: Using mild covers on the large type of motors.

7.2.1. Advantages and disadvantages of using the spare parts:

It is expected that the proposed system will:

a. increase the total number of renovated large and small type of vehicle motors.
b. decrease the work time on the processes.
c. decrease the needs for human resources.

In addition to these benefits the proposed system might incur some disadvantages:

a. The cost can increase due to the increase in purchase cost.
b. The under utilisation of the resources can be observed.

Hence, both the advantages and disadvantages of the using spare parts at the proposed system are should be considered. One of the purposes of this simulation is to generate the data that helps to make this decision effectively. With the simulation runs, we try to quantify the true advantages and disadvantages of the proposed approach.

7.2.3. The technical data to support the effect of the spare part usage.

The technical data is obtained by the help of the technicians. These are:

- The use of motor block for the small type of motors decreases about 45 minutes of processing time of the block renovation section.

- The use of crank for the small type of motors decreases about 40 minutes of processing time of the crank renovation section.

- The use of mild cover for the large type of motors decreases about 45 minutes of processing time of the crank renovation section.

Also, the code for this chapter is presented in Appendix C.3.

7.3. The Results

In this section, we compare the existing and proposed system to see if there is any improvement due to spare parts. Ten replications are taken for proposed system. The average throughput of each type of motor is obtained and presented in the *Table 7.3-1.*

In general, the results indicate that the only improvement is obtained for only small type of motors (about 6%). The improvements for the small type motors seem as significant. The details of the results are presented in the following sections.

REPLICATION	LARGE TYPE	SMALL TYPE	UNIMOG TYPE
1	2318	3320	1846
2	2339	3404	1856
3	2323	3398	1802
4	2340	3395	1861
5	2341	3405	1823
6	2314	3388	1806
7	2338	3399	1808
8	2299	3413	1852
9	2311	3401	1826
10	2343	3394	1876
CUM. SUM	23266	33917	14654
AVERAGE	2326.6	3391.7	1835.6
STAN. DEV	15.60056979	26.10257544	26.00940001
CI. for 0.05	9.669141339	16.17822263	16.12047305

Table 7.3-1. The results of the changes on the existing system.

7.3.2. Comparison of the Existing & Proposed System for Large Motors.

Again, in the experiments the common random numbers (CRN) are used to increase the precision. Specially, the confidence interval approach in the correlated sampling is used to identify the difference between the mean performance [15].

The paired-t results are given in the *Table 7.3-2.* In this table, the rows represent the results for the number of renovated motors at each replication. This table also displays the average difference between systems, standard deviations, and confidence intervals on the mean difference.

LARGE MOTOR	EXISTING SYSTEM	PROPOSED SYSTEM	DIFFERENCE
1	2310	2309	1
2	2332	2336	-4
3	2346	2339	7
4	2350	2318	32
5	2356	2337	19
6	2333	2319	14
7	2347	2342	5
8	2341	2340	1
9	2313	2330	-17
10	2331	2329	2
	CHANGE	%0.401344	DECREASE
AVERAGE. DIFF.	9.375		
STAND. DEV.	13.35830994	4.22426851	
VARIANCE	178.4444444	17.84444444	
95% CI. T-test	9.546846833	The systems are statistically the same.	

Table 7.3-2. Comparison of the number of renovated large type motors.

The results indicate that the performance of the proposed system is worse than the existing system about 0.4%. But this difference is not statistically significant.

Processed soft cover usage at the block renovation section for large motors are used in the proposed system. The number of renovated large motors should have increased by the implementation. But the other implementations especially at the small motors detoriate the renovation of the other types, due to high increase of the small motors in the percentage of common resources usage and the bottlenecks at the large motor renovation sections prevent the increase of the throughput. In order to make further investigation, the rate of change at the utilisation of the block renovation section is compared and found that the new average utilisation of this section is 0,936 and 0,282 decrease is observed and this difference is significant. Next, we tried to decrease capacity of the existing system at the crank renovation section. As a result of simulation, decreasing the capacity of the section is decreasing the throughput of the system.

7.3.3. Comparison of the Existing & Proposed System for the Small motors.

When the same procedure is applied for the small motors (with respect to the number of renovated small motors), we observed 6.183% increases in the throughput of the small motors (see *Table 7.3-3)*.

SMALL TYPE	EXISTING SYSTEM	PROPOSED SYSTEM	DIFFERENCE
1	3175	3320	-145
2	3204	3404	-200
3	3212	3398	-186
4	3168	3395	-227
5	3204	3405	-201
6	3216	3388	-172
7	3176	3399	-223
8	3185	3413	-228
9	3184	3401	-217
10	3218	3394	-176
	CHANGE	-%6.183082	INCREASE
AVERAGE. DIFF.	-197.5		
STAND. DEV.	27.58924911	8.724486613	
VARIANCE	761.1666667	76.11666667	
95% CI. T-test	19.71733975	The systems are statistically different each other.	

Table 7.3-3. Comparison of the number of renovated small motors.

The Paired-t test results indicate that the improvement is significant and thus, the performance of the proposed system is statistically better than the existing system.

Our further investigation indicate that, the small motor block and crank renovation sections\` utilisation rate decreases and the utilisations of the 1st and 2nd repair sections\` increases about 0.980 are observed. Also, the 1st and 2nd repair sections appeared as new bottlenecks in the small motor renovation sections by the implementation.

7.3.4. Comparison of the Existing & Proposed System for Unimog Motors.

When the same procedure is applied for the unimog motors, we observe the deterioration of performance 0.021% for the proposed system *(see Table 4.3-5)*.

UNIMOG	1ST SYSTEM	2ND SYSTEM	DIFFERENCE
1	1821	1846	-25
2	1822	1856	-34
3	1825	1802	23
4	1841	1861	-20
5	1815	1823	-8
6	1880	1806	74
7	1813	1808	5
8	1833	1852	-19
9	1861	1826	35
10	1849	1876	-27
	CHANGE	%0.021786	DECREASE
AVERAGE. DIFF.	0.4		
STAND. DEV.	34.2999838	10.84660725	
VARIANCE	1176.488889	117.6488889	
95% CI. T-test	24.51333239	The systems are statistically the same.	

Table 7.3-5. Comparison of the number of renovated unimog motors.

The Paired-t test results (see *Table 7.3-5*) indicate that the performance of the proposed system is statistically the same as the existing system (Note that the average differences plus and minus confidence intervals include zero), eventhough the proposed system yield numerically worse performance than the existing system. The reason for such an inferior performance is that the percentages of the large and small motors are greater than the unimog motors due to spare parts application. Therefore unimog motors have less common resources, which in term decreases the throughput of the motors.

7.4. Conclusion:

The results are indicating that when the ready spare part usage increase, the throughput is not always increasing as seen in our case due to multiple bottlenecks and common resources. The resources utilisations are decreasing when the spare parts are used. We observed that, when the spare parts are used, sometimes the other resources could appear as new bottleneck resources. Recall that, these results are tied to our experimental settings. If we have used different experimental setting, the results can be changed.

The average total cost increase or decrease due to purchasing spare parts and increasing the number of renovated motors should be carefully calculated for the decision of the using spare parts.

Chapter 8

CONCLUSION

In this study, we gave a brief introduction to the army maintenance system and 5^{th} level renovation system and, we lay down the background of this research. It appears that research in maintenance systems in army needs fast approaches to solve the operational problems.

Then, we developed a simulation model to analyse the behaviour of the existing and proposed 5^{th} level renovation maintenance systems of the Turkish Army in 1011 Main Repair Depot is developed. The model can be adapted to represent other depot`s renovation maintenance systems.

The objectives of the study are:

- To understand the behaviour of the existing system.

- To detect the bottlenecks in the existing system.

- To optimise the performances of existing the system.

- To develop the alternative systems.

The simulation model also enables to investigate the effect of several changes on the simulation model. In this study, we evaluate the existing system with using genetic algorithms and also we designed new proposed systems and compare with the existing system.

First, we gave the meaning of the study for the army and then presented general conclusion and later study areas.

8.1. What does it mean for the army?

(When the new improvement is done.)

Logistics is the application of time and space factors to war. It is the economics of warfare, and it comprises, in the broadest sense, the three large M`s of warfare-material, movement, and maintenance. If international politics is the 'art of possible,' and war is its instrument, logistics is the art of defining and extending the possible. It provides the substance that physically permits an army to live and move and have its being.

The basic mission of the Army Logistics System is to support the soldier in the field with what is needed, when, where, and in the condition and quantity required at minimum expenditure of resources. Therefore, whatever is done to increase the support rate is important and necessary.

8.2. General Conclusion

8.2.1. Existing and Improved System

In the existing system, we get the following results:

- The existing system is the non-terminating system.

- The existing system has the bottleneck sections due to over utilisation.

- The breakdowns in the existing system increase the waiting time in queues.

- Also, the existing system has lower utilised sections due to higher capacities.

Then, we improved the existing system with using genetic algorithm utility of the AutoStad. In our optimisation problem, we try to increase total throughput of the system and utilisations of the sections for departments and, a comparison between the existing and improved system is made to see if any improvement is obtained due to the changes in the system:

- The improvement obtained in the numbers of the renovated motors is on the averagely 17% and it is significant.

- The optimised system requires many additional resources.

- The utilisations are decreased at the sections that their capacities are increased.

- The utilisations are increased at the sections that their capacities are decreased due to optimisation process.

This additional resource requirement decreases the probability of the application of the improved system.

8.2.2. Pre-control and Repair Section

The purpose of the pre-control & repair section is to increase the productivity of Army Depot's renovation units. (By increasing the number of the renovated & repaired vehicle motors.)

The pre-control and repair section checks only large and small type of vehicle motors before entering the motor renovation unit. The unimog type of vehicle motors will not be handled in this section, since this kind of vehicle motors will be disposed up to 5 years in the Turkish Army. The control mechanism defines the status of the vehicle motors and the type of breakdowns. To admit vehicle motors, maintenance technicians examine if the motors are renovated 1 years ago or unused or not so much used according to the their physical appearances.

The results indicate that the proposed system increases the number of the renovated and repaired motors.

8.2.3. Combined Parallel Resources System

The purpose of the combined resources system is to decrease the lost capacity and increase the productivity of Army Depot's renovation units. The proposed system is a generalisation of the flow shop and the parallel machine environments. Some of the resources are combined for the use of free space at the machines. The results showed that:

- The throughput of the system increases.

-The application of the proposed system decreases the bottleneck and breakdown effects on the system.

-We can not obtain decreases at in the number of breakdowns but we can continue to renovation process due to usage

8.2.4. Increased Ready Spare Part Usage

The purpose of the increasing the usage of the spare parts is to increase the capacity of the existing system. The facilities are the same as the existing system. The ready spare part usage in the existing system decreases the processing times and the need for human resources. The effect of the ready spare part usage is applied on the large and small type motors. Our experimental results are indicating that:

- The throughput does not always improve, due to multiple bottlenecks and common resources.

- The resources utilisations are decreasing when the spare parts are used.

- The spare part usages in the existing system result new bottlenecks.

The average total cost increase or decrease due to purchasing spare parts and increasing the number of renovated motors should be carefully calculated for the decision of the using spare parts.

8.2.5. Comparison of all the proposed systems

We also made comparisons among the proposed systems with respect to throughput of the systems. The results are given in *Table 8-1*. In this table, the rows represent the results for the average number of renovated large motors at each proposed system (based on ten replications).

Throughput of the Proposed Systems	Existing system	1st Proposed System (Pre-control and Repair Section)	2nd Proposed System (Combined Parallel Resources System)	3rd Proposed System (Increased Ready Spare Part Usage)
Large Motors	2336	2394	2552	2326
Small Motors	3194	4128	3398	3391
Unimog Motors	1836	1860	2015	1835

Table 8-1. Average renovated motors in the proposed systems.

According to Large Motors,
The Paired-t test results show that the performance of the 2nd proposed system is statistically better than the 1st and 3rd proposed system and 1st proposed system is better than the 3rd proposed system.

According to Small Motors,
The Paired-t test results show that the performance of the 1st proposed system is statistically better than the 2nd and 3rd proposed system and 2nd proposed system is better than the 3rd proposed system.

According to Unimog Motors,
The Paired-t test results show that the performance of the 2nd proposed system is statistically better than the 1st and 3rd proposed system and 1st proposed system is better than the 3rd proposed system.

Also, we investigated the cost effectiveness of the pre-control and repair section` (proposed system 1) and combined parallel resources system (proposed system 2) (see sections 5.4 and 6.4). The best system for the cost impact is the proposed system 1. Therefore, we can conclude that the 2nd proposed system is the best system of all the systems.

8.2.6. Further Research Areas

In order to further investigate the potential of simulation in military maintenance systems are high, since there is limited research in the Turkish Army. We suggest the following research directions: Investigating the effects of system configurations, investigating the effects of the scheduling procedures, investigating the effect of system disturbances, and developing rules or guidelines.

Bibliography

[1] F. Azadivar and J. Victor Shu. Use of Simulation in Optimisation of maintenance Policies. In *Proceeding of the 1998 Winter Simulation Conference*. 1998.

[2] J. Banks. *Handbook of Simulation.* John Wiley & Sons, Inc., New York. 1998.

[3] M. A. Centano and M. Florencia Reyes. So You Have Your Model: What To Do Next A Tutorial on Simulation Output Analysis. In *Proceeding of the 1998 Winter Simulation Conference*. 1998.

[4] B. Dengiz, F. Sen and A. Bulgak. Optimisation of Stochastic Systems Using Genetic Algorithms. *Technical Report*, Gazi University, Turkey. 1997.

[5] G. W. Evans and W. E. Biles. Simulation of Advanced Manufacturing Systems. In *Proceeding of the 1992 Winter Simulation Conference*. 1992.

[6] P. A. Farrington and J. J. Swain. Designing Simulation Experiments for Evaluating Manufacturing Systems. *In Proceeding of the 1994 Winter Simulation Conference*. 1994.

[7] The Army Logistics System, *FM 700-80*, Chapter 1. Introduction. 1998

[8] The Army Logistics System, *FM 700-80*, Chapter 11. Maintenance management. 1998.

[9] The Army Logistics System, *FM 700-80*, Chapter 18. Automated Logistics Management Information Systems. 1998.

[10] R. F. Garzia, M. R. Garzia, and B. P. Zeigler. Discrete-Event Simulation. *IEEE Spectrum*, December, pages. 32-36. 1986.

[11] D. Harrel and M.Tumay. *Simulation Made Easy: A Manager's Guide.* Industrial Engineering and Management Press. Norcross. Ga. 1996.

[12] R. G. Harvey, D. V. McElveen, P. H. Miyares, and T. F. Schuppe. C-141 Depot Maintenance: Using Simulation to Define Resource Requirement. In *Proceedings of the 1992 Winter Simulation Conference*. 1992.

[13] J. D. Ianni. Maintenance Simulation: Research and Applications. Available from *Department of Air Force Research Laboratory* web site, http://www.he.afrl.af.mil/hes/programs/depth/pubs /r_and _a.htm. 2000.

[14] L. Jenkins. A Simulation Model for Scheduling Inspection of Machine Components. *Application of Simulation Areas,* University of Pittsburg Press. 1987.

[15] Union maintenance For the Ordonnance Goods. *KKTT 9-2810*, Chapter 1. Introduction. 1997.

[16] Union maintenance For the Ordonnance Goods. *KKTT 9-2810*, Chapter 2. Maintenance Rules and Sections. 1997.

[17] 5th Level Maintenance System, *KKY 314-2*, Chapter 1. Introduction. 1981.

[18] 4th Level Maintenance System, *KKY 314-1*, Chapter 1. Introduction. 1996.

[19] K. Kang and R. J. Roland. Military Simulation, *Handbook of Simulation, Edited by Jerry Banks,* John Wiley & Sons, Inc., New York. 1998.

[20] A. M. Law and W. D. Kelton. *Simulation Modelling and Analysis*, 2nd Edition, McGraw-Hill Book Company. 1991.

[21] A. M. Law and M. G. McComas. Simulation of Manufacturing Systems. In *Proceeding of the 1999 Winter Simulation Conference*. 1999.

[22] T. H. Naylor, J. L. Balintfy, D. S. Burdick, and K. Chu. *Computer Simulation Techniques.* John Wiley & Sons, Inc., New York. 1966.

[23] R. D. Painter. Object-Oriented Military Simulation Development and Application. In *Proceeding of the 1995 Winter Simulation Conference*. 1995.

[24] D. J. Parsons and L. C. Krause. Tactical Logistics and Distribution Systems (Tloads) Simulation. . In *Proceeding of the 1999 Winter Simulation Conference*. 1999.

[25] A. A. B. Pritsker. *Introduction to Simulation and Slam II*, Third Edition, John Wiley & Sons, Inc., New York. 1986.

[26] M. Pinedo. *Scheduling Theory, Algorithms, and Systems*, Prentice Hall Englewood, New Jersey. 1995.

[27] L. K. Piplani, J. G. Mercer, and R. O. Roop. Systems Acquisition Manager's Guide for the Use of Modelling and Simulation, *Defence Systems Management College*, Ft. Belvoir, Va. 1994.

[28] R. G. Sargent. Verification and Validation of Simulation Models. In *Proceeding of the 1994 Winter Simulation Conference*. 1994.

[29] B. Stuckman, G. Evansand M. Mollaghasemi. Comparison of Global Search Methods for Design Optimisation Using Simulation. In *Proceeding of the 1994 Winter Simulation Conference*. 1994.

[30] T. Savolainen, D. Beeckmann, P. Groumpos, and H. Jagdev. Positioning Of Modelling Approaches, Methods and Tools. *Computer Industry*, Vol. 25, pages 255-262. 1995.

[31] E. Tekin and I. Sabuncuoglu. Simulation Optimisation: A Comprehensive Review on Theory and Applications. *Journal of Industrial Engineering Operation Research*, vol. 11. 1998.

[32] P. D. Welch. The Statistical Analysis of Simulation Results. In *Computer Performance Modelling Handbook*, Ed. S. Lavenberg, New York: Academic Press. 1983.

[33] W. Whitt. Simulation Run Length Planning. In *Proceedings of the 1989 Winter Simulation Conference*. 1998.

Appendix A

1. Data Requirements

Renovation lead times for the products are known approximately by the past experiences. Production planning is done very roughly based on these data, the state of machines and workers (See *Table A-1*).

Processes	General	Large Motor		Small Motor		Unimog Motor	
	Capacity	Minutes	Capacity	Minutes	Capacity	Minutes	Capacity
Dismantle	12	90		125		90	
Washing	10	90		80		90	
Electrical Ren.		300	7	300	9	229	5
Carburation Ren.		285	7	200	8	267	6
Block Ren.		146	3	145	5	133	3
Crank Ren.		150	3	110	3	68	2
Piston Ren.		156	3	75	3	110	2
Bed Prep. Ren.		133	3	43	2	100	2
1st Repair		102	2	108	3	266	5
2nd Repair		361	8	106	3	208	4
Mounting		106	3	80	3	158	3
Testing	14	171		154		175	
Packing	7	85		72		90	
Breakdown Repair		240		240		240	

Table A-1. The capacities and operation times.

Data collection and analyse is very important part of the modelling and it is generally necessary to represent each source of randomness by a probability function. Also, we know that failure to choose the correct distribution affects the accuracy of model`s results (validity of the output) and the results can be differed 30-60% from the reference model.

In our study, data collection on the random variables of interest can not implemented perfectly, since the number of required probability distributions was large and time available for the simulation study. Therefore, we applied the triangular approach to the data. In the triangular approach, the experts are asked for subjective estimate of the most likely time to perform the task. This most likely value c is the mode of the distribution of X. Given a, b, and c, the random variable X is then considered to have a triangular distribution on the interval [a, b] with mode c (Cinlar, 1975, chap.4). The difficulty with this triangular approach was that it required subjective estimates of the absolute minimum and maximum possible values a and b. For this reason, we investigate the minimum and maximum values of the processes by interviewing the technicians and the workers at the renovation unit and then applied the triangular approach to the process times (See *Table A-2*).

Processes	Large Motor			Small Motor			Unimog Motor		
	a	c	b	a	c	b	a	c	b
Dismantle	55	90	100	60	125	160	54	90	99
Washing	54	90	100	48	80	88	54	90	99
Electrical Ren.	180	300	330	180	300	330	137	229	252
Carburation Ren.	171	285	313	120	200	220	160	267	293
Block Ren.	87	146	160	86	145	160	93	133	146
Crank Ren.	90	150	180	66	110	132	48	68	74
Piston Ren.	93	156	171	45	75	83	77	110	122
Bed Prep. Ren.	79	133	146	25	43	48	70	100	110
1st Repair	60	102	112	63	108	118	186	266	292
2nd Repair	216	361	397	64	106	116	145	208	228
Mounting	63	106	116	48	80	88	110	158	173
Testing	102	171	188	92	154	167	105	175	192
Packing	51	85	93	43	72	79	54	90	99
Breakdown Rep.	60	240	800	60	240	800	60	240	800

Table A-2. Applied triangular approach to the processing times.

Also, in *Table A-3* the parameters of the model and *Table A-4* the entity flow rates are presented. The probabilities of the parameters are converted by using the historical enlistment of the depot. The entity flow rates are also constant except breakdown appearance. The breakdowns are created by using exponential distribution since we obtained only the appearance per year and exponential distributions are mostly used for interarrival times of breakdowns to a system that occur a constant rate.

Parameters	Large Motor	Small Motor	Unimog Motor
Dispose Rate	10%	30%	45%
Dispose Rate of Electrical Parts	0%	15%	20%
Dispose Rate of Carburation Parts	0%	14%	22%
Breakdown At Mot. Block Ren. Sec	9%	12%	5%
Breakdown At Mot. Crank Ren. Sec	9%	11%	5%
Breakdown At Mot. Piston Ren. Sec	8%	11%	5%
Breakdown At Mot. Bed Prep. Ren. Sec	9%	11%	5%

Table A-3. Parameter set of the model.

TYPES OF MOTOR	(ENTITY)
LARGE MOTORS	Constant 32min
SMALL MOTORS	Constant 45min split 1
UNIMOG MOTORS	Constant 65min split 1
BREAKDOWN	Exponential 150hour

Table A-4. Entity flow rates.

2. Conceptual Model

2.1. Events:

- The orders are given by Turkish Land Forces.
- Production plans are prepared.
- The motor arrives at the disassembling section.
- The disassembling section begins dismantling.
- The disassembling section completes dismantle.

- Washing section begins washing the motor parts.
- Washing section completes washing the motor parts.
- Block renovation section begins renovating the block of the motor.
- Block renovation section completes renovating the block of the motor.
- Crank renovation section begins renovating the crank of the motor.
- Crank renovation section completes renovating the crank of the motor.
- Cylinder bed preparation section begins to prepare the motor.
- Cylinder bed preparation section completes prepare the motor.
- Piston renovation section begins renovating the piston of the motor.
- Piston renovation section completes renovating the piston of the motor.
- 1st repair section begins the repair of the motor.
- 1st repair section completes the repair of the motor.
- 2nd repair section begins the repair of the motor.
- 2nd repair section completes the repair of the motor.
- Assembly section begins assembling the motor.
- Assembly section completes assembling the motor.
- Testing section begins testing of the motor.
- Testing section completes testing of the motor.
- Packing section begins packing of the motor.
- Packing section completes packing of the motor.
- Electrical section begins to renovate electrical parts of the motor. (Part availability.)
- Electrical section completes renovating electrical parts of the motor. (Part availability.)
- Fuel oil section begins to renovate fuel oil parts of the motor.

- Fuel oil section completes the renovation of fuel oil parts of the motor.

2.2. Activities:

- Disassembling section.
- Washing section.
- Large motor block renovation section.
- Large motor crank renovation section.
- Large motor cylinder bed preparation section.
- Large motor piston renovation section.
- Large motor 1st repair section
- Large motor 2nd repair section
- Large motor assembly section
- Small motor block renovation section.
- Small motor crank renovation section.
- Small motor cylinder bed preparation section.
- Small motor piston renovation section.
- Small motor 1st repair section
- Small motor 2nd repair section
- Small motor assembly section
- Unimog motor block renovation section.
- Unimog motor crank renovation section.
- Unimog motor cylinder bed preparation section.
- Unimog motor piston renovation section.
- Unimog 1st repair section
- Unimog 2nd repair section
- Unimog assembly section
- Testing section.

- Packing and painting section.
- Fuel oil systems renovation section.
- Electrical systems renovation section.

3. Simulation Code.

```
begin Pselection arriving
   set acont to triangular 45,60,75
   set aadb to triangular 55,60,70
   if load type=Lbig then
   begin
   set adis1 to triangular 55,90,100
   set aw1  to triangular 54,90,100
   set are1 to triangular 180,300,330
   set arc1 to triangular 171,285,313
   set ad1  to triangular 50,120,150
   set ab   to triangular 87,146,160
   set acr  to triangular 90,150,180
   set abe  to triangular 79,133,146
   set ap   to triangular 93,156,171
   set a1r  to triangular 60,102,112
   set a2r  to triangular 216,361,397
   set am   to triangular 63,106,116
   set abr1 to triangular 102,171,188
   set ap1  to triangular 51,85,93
   end
    else if load type=Lsmall then
   begin
   set ad2  to triangular 60,125,160
   set aw2  to triangular 48,80,88
   set are2 to triangular 180,300,330
   set arc2 to triangular 120,200,220
   set asmall   to triangular 445,743,817
   set asb  to triangular 86,145,160
   set asc  to triangular 66,110,132
   set asbe to triangular 25,43,48
   set asp  to triangular 45,75,83
   set as1r to triangular 63,108,118
   set as2r to triangular 64,106,116
   set asm  to triangular 48,80,88
   set ap2  to triangular 43,72,79
```

```
    set abr2 to triangular 92,154,167
    end
    else if load type=Lunimog then
    begin
    set adis3 to triangular 54,90,99
    set aw3  to triangular 54,90,99
    set are3 to triangular 137,229,252
    set arc3 to triangular 160,267,293
    set aub  to triangular 93,133,146
    set auc  to triangular 48,68,74
    set aube to triangular 70,100,110
    set aup  to triangular 77,110,122
    set au1r to triangular 186,266,292
    set au2r to triangular 145,208,228
    set aum  to triangular 110,158,173
    set au   to triangular 936,1560,1716
    set abr3 to triangular 105,175,192
    set ap3  to triangular 54,90,99
    end
    send to Pfirst

begin Pfirst arriving
    set Atimestamp to ac
    inc Vinsystem by 1
    if load type=Lbig then
    begin
        move into Q1dis
        wait until Vbigblock <=70
        wait until Vbigcrank <=70
    end
    else if load type=Lsmall then
    begin
        move into Q2dis
        wait until Vs <=100
        wait until Vsc <=100
    end
    else if load type=Lunimog then
    begin
        move into Q3dis
        wait until Vu <=100
    end
    if load type=Lbig then use Rdis for adis1 min
    else if load type=Lsmall then use Rdis for  adis2 min
    else if load type=Lunimog then use Rdis for adis3 min
   /* move into conv:geton
```

```
      travel to conv:getonwash*/
      send to Pw
  end
  begin Pw arriving
      if load type=Lbig then move into Q1wash
      else if load type=Lsmall then move into Q2wash
      else if load type=Lunimog then move into Q3wash
      if load type=Lbig then use Rw for aw1 min
      else if load type=Lsmall then use Rw for aw2 min
      else if load type=Lunimog then use Rw for aw3 min
      if load type=Lbig then
       begin
         move into Q1washf
      end

      else if load type=Lsmall then
      begin
        /* move into conv:getoffwash2
         travel to conv:getonren2*/
      end
      else if load type=Lunimog then
      begin
      end
      send to Ptype
 end
 begin Ptype arriving
     if load type=Lbig    then
     begin
        set Atype to 1
        send to oneof(0.10:Pdelet,0.9:Pdup)
     end
     else if load type=Lsmall  then
     begin
        set Atype to 2
        send to oneof(0.30:Pdelet,0.70:Pdup)
     end
     else if load type=Lunimog then
     begin
        set Atype to 3
        send to oneof(0.45:Pdelet,0.55:Pdup)
     end
end

begin Pdelet arriving
    if load type=Lbig    then inc Vbigdel by 1
```

```
        else if load type=Lsmall  then inc Vsmalldel by 1
        else if load type=Lunimog then inc Vunimogdel by 1
      /*if load type=Lbig    then  print"Deleted  Type "Atype , Vbigdel to
message
        else if load type=Lsmall  then print"Deleted  Type "Atype , Vsmalldel to
message
        else if load type=Lunimog then print"Deleted  Type "Atype , Vunimogdel
to message
     end
     begin Pdup arriving
        if load type=Lbig  then clone 1 loads to Pelect new load type L1elect
        else if load type=Lsmall  then clone 1 loads to Pelect nlt L2elect
        else if load type=Lunimog then clone 1 loads to Pelect nlt L3elect
        if load type=Lbig  then clone 1 loads to Pcarb new load type L1carb
        else if load type=Lsmall  then clone 1 loads to Pcarb nlt L2carb
        else if load type=Lunimog then clone 1 loads to Pcarb nlt L3carb
        send to P1
   end

   begin Pdup1 arriving
        if load type=Lbig  then clone 1 loads to Pelect new load type L1elect
        else if load type=Lsmall  then clone 1 loads to Pelect nlt L2elect
        else if load type=Lunimog then clone 1 loads to Pelect nlt L3elect
        if load type=Lbig  then clone 1 loads to Pcarb new load type L1carb
        else if load type=Lsmall  then clone 1 loads to Pcarb nlt L2carb
        else if load type=Lunimog then clone 1 loads to Pcarb nlt L3carb
        send to die
   end
   begin Pelect arriving
        if load type=L1elect then
        send to Pelect1
        else if load type=L2elect then
        begin
           send oneof (15:die,85:Pelect1)
        end
        else if load type=L3elect then
        begin
           send oneof (20:die,80:Pelect1)
        end
   end
   begin Pelect1 arriving
        if load type=L1elect then
        begin
          move into Q1elect
          use R1elect for are1 min
```

```
    end

    else if load type=L2elect then
    begin
       move into Q2elect
       use R2elect for are2 min
    end

    else if load type=L3elect then
    begin
       move into Q3elect
       use R3elect for are3 min
    end

    if load type=L1elect then inc Velectbig by 1
    else if load type=L2elect  then inc Velectsmall by 1
    else if load type=L3elect then inc Velectunimog by 1

    send to die
end
begin Pcarb arriving
    if load type=L1carb then
       send to Pcarb1
    else if load type=L2carb then
    begin
       send oneof (14:die,86:Pcarb1)
    end
        else if load type=L3carb then
    begin
       send oneof (18:die,78:Pcarb1)
    end

end
 begin Pcarb1 arriving

    if load type=L1carb then
    begin
       move into Q1carb
       use R1carb for  arc1 min
    end

    else if load type=L2carb then
    begin
       move into Q2carb
       use R2carb for arc2 min
```

```
        end

        else if load type=L3carb then
        begin
           move into Q3carb
           use R3carb for arc3 min
        end

        if load type=L1carb then inc Vcarbbig by 1
        else if load type=L2carb  then inc Vcarbsmall by 1
        else if load type=L3carb then inc Vcarbunimog by 1

        send to die
    end
    begin Pbreak arriving

        if load type=Lbreak then
           begin
              increment Vbreak by 1
              print"Total breakdown" , Vbreak to message
              set Abreak to continuous
(.09:1,.18:2,.27:3,.35:4,.47:5,.58:6,.69:7,.80:8,.85:9,.90:10,.95:11,1:12)
              if Abreak=1 then
              begin
                 take down Rblockb
                 wait for triangular 60,240,800 min
                 print"Breakdown at Rblockb "  to message
                 bring up Rblockb
              end

              else if Abreak=2 then
              begin
                 take down Rcrankb
                 wait for triangular 60,240,800 min
                 print"Breakdown at Rcrankb "  to message
                 bring up Rcrankb
              end

              else if Abreak=3 then
              begin
                 take down Rbedb
                 wait for triangular 60,240,800 min
                 print"Breakdown at Rbedb "  to message
                 bring up Rbedb
              end
```

```
else if Abreak=4 then
begin
    take down Rpistonb
    wait for triangular 60,240,800 min
    print"Breakdown at Rpistonb "  to message
    bring up Rpistonb
end

else if Abreak=5 then
begin
    take down Rblocks
    wait for triangular 60,240,800 min
    print"Breakdown at Rblocks "  to message
    bring up Rblocks
end
else if Abreak=6 then
begin
    take down Rcranks
    wait for triangular 60,240,800 min
    print"Breakdown at Rcranks "  to message
    bring up Rcranks
end
else if Abreak=7 then
begin
    take down Rbeds
    wait for triangular 60,240,800 min
    print"Breakdown at Rbeds "  to message
    bring up Rbeds
end

else if Abreak=8 then
begin
    take down Rpistons
    wait for triangular 60,240,800 min
    print"Breakdown at Rpistons "  to message
    bring up Rpistons
end
else if Abreak=9 then
begin
    take down Rblocku
    wait for triangular 60,240,800 min
    print"Breakdown at Rblocku "  to message
    bring up Rblocku
end
```

```
            else if Abreak=10 then
            begin
                take down Rcranku
                wait for triangular 60,240,800 min
                print"Breakdown at Rcranku "  to message
                bring up Rcranku
            end
            else if Abreak=11 then
            begin
                take down Rbedu
                wait for triangular 60,240,800 min
                print"Breakdown at Rbedu "  to message
                bring up Rbedu
            end
            else if Abreak=12 then
            begin
                take down Rpistonu
                wait for triangular 60,240,800 min
                print"Breakdown at Rpistonu "  to message
                bring up Rpistonu
            end
            send to die
        end
end

begin P1 arriving

    if load type=Lbig then

    begin
        increment Vbigblock by 1
        move into Qblockb
        use Rblockb   for  ab min
        decrement Vbigblock by 1
        increment Vbigcrank by 1
        move into Qcrankb
        use Rcrankb   for  acr min
        decrement Vbigcrank by 1
        move into Qbedb
        use Rbedb     for  abe min
        move into Qpistonb
        use Rpistonb  for  ap min
        move into Q1repairb
        use R1repairb for  a1r min
        move into Q2repairb
```

```
    use R2repairb for  a2r min
    move into Qmountingb
  if Vbig >= Velectbig then
    begin
        wait until Vbig <= Velectbig
    end

    else if Vbig >= Vcarbbig then
    begin
       wait until Vbig <= Vcarbbig
    end
    use Rmountingb for am min
   /* move into conv:getoffren1
    travel to conv:goodby1 */
end

else if load type=Lsmall then

begin
    increment Vs by 1
    move into Qblocks
    use Rblocks for asb  min
    decrement Vs by 1
    increment Vsc by 1
    move into Qcranks
    use Rcranks for asc min
    decrement Vsc by 1
    move into Qbeds
    use Rbeds for asbe min
    move into Qpistons
    use Rpistons for asp min
    move into Q1repairs
    use R1repairs for as1r min
    move into Q2repairs
    use R2repairs for as2r min
    move into Qmountings
    if Vsmall >= Velectsmall then wait until Vbig <= Velectsmall
    else if Vsmall >= Vcarbsmall then wait until Vbig <= Vcarbsmall
    use Rmountings for asm min
   /* move into conv:getoffren2
    travel to conv:goodby2 */
end

else if load type=Lunimog then
```

```
      begin
          increment Vu by 1
          move into Qblocku
          decrement Vu by 1
          use Rblocku for aub min
          move into Qcranku
          use Rcranku for auc min
          move into Qbedu
          use Rbedu for aube min
          move into Qpistonu
          use Rpistonu for aup min
          move into Q1repairu
          use R1repairu for au1r min
          move into Q2repairu
          use R2repairu for au2r min
          move into Qmountingu
          if Vunimog >= Velectunimog then wait until Vunimog <=
Velectunimog
          else if Vunimog >= Vcarbunimog then wait until Vunimog <=
Vcarbunimog

          use Rmountingu for aum min
        /*  move into conv:getoffren3
          travel to conv:goodby3 */
      end

      send to Pinc
  end

  begin Pinc arriving
      if load type=Lbig then inc Vbig by 1
      else if load type=Lsmall  then inc Vsmall by 1
      else if load type=Lunimog then inc Vunimog by 1
    /* if load type=Lbig then print"Renovated Type "Atype , Vbig to message
      else if load type=Lsmall then print"Renovated Type "Atype, Vsmall to
message
      else if load type=Lunimog then print"Renovated Type "Atype, Vunimog
to message
    */  send to Pbrem
  end

  begin Pbrem arriving
      move into Qbrem
      get Rbrem
      if load type=Lbig    then wait for  abr1 min
```

```
    else if load type=Lsmall  then wait for abr2 min
    else if load type=Lunimog then wait for abr3 min
    free Rbrem
    move into Qpack
    get Rpacking
    if load type=Lbig    then wait for ap1 min
    else if load type=Lsmall  then wait for  ap2 min
    else if load type=Lunimog then wait for  ap3 min
    free Rpacking
    send to die
end
```

Appendix B

1. Additional Warm-up Figures.

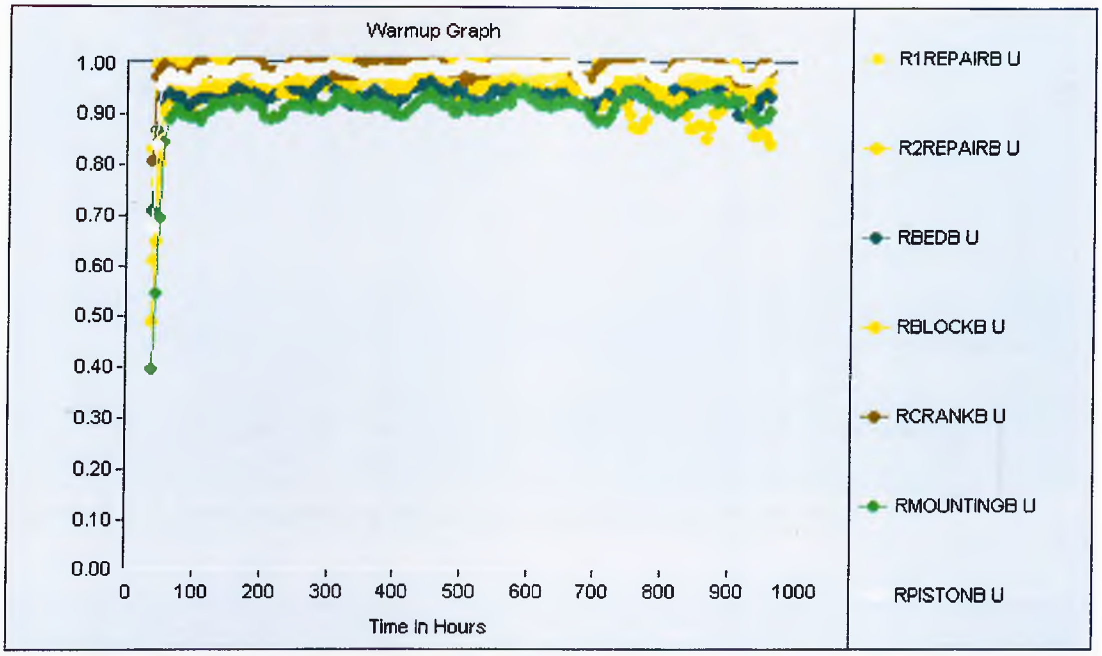


Figure B.1-1 Large motors sub-section`s utilisations versus time.

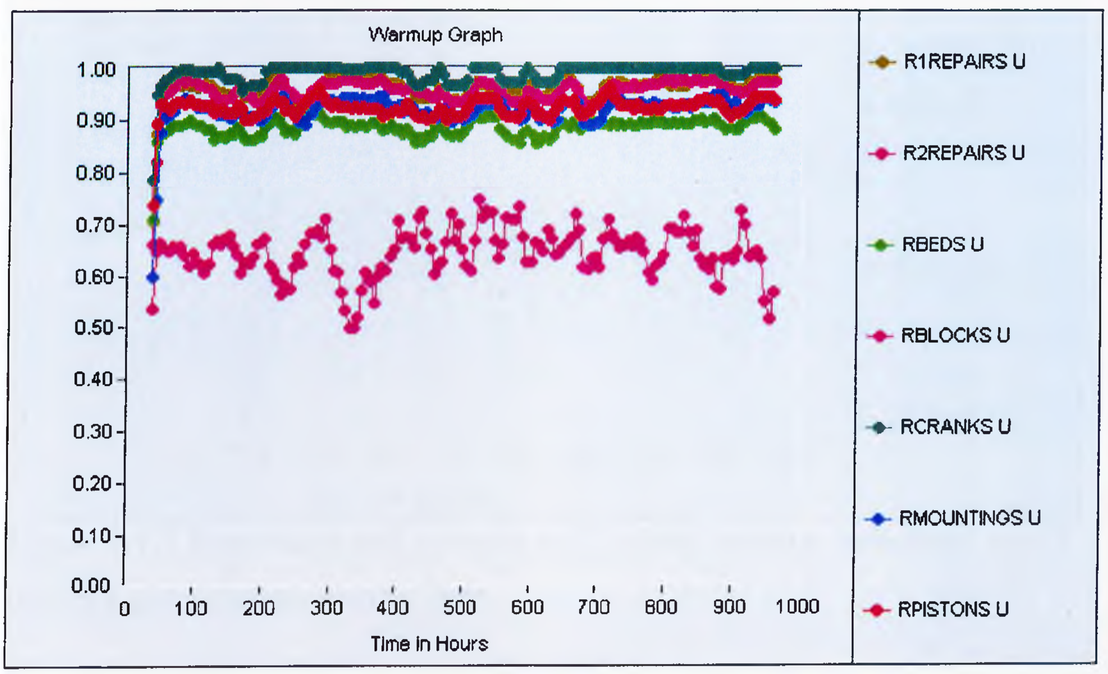


Figure B.1-2 Small motors sub-section`s utilisations versus time.

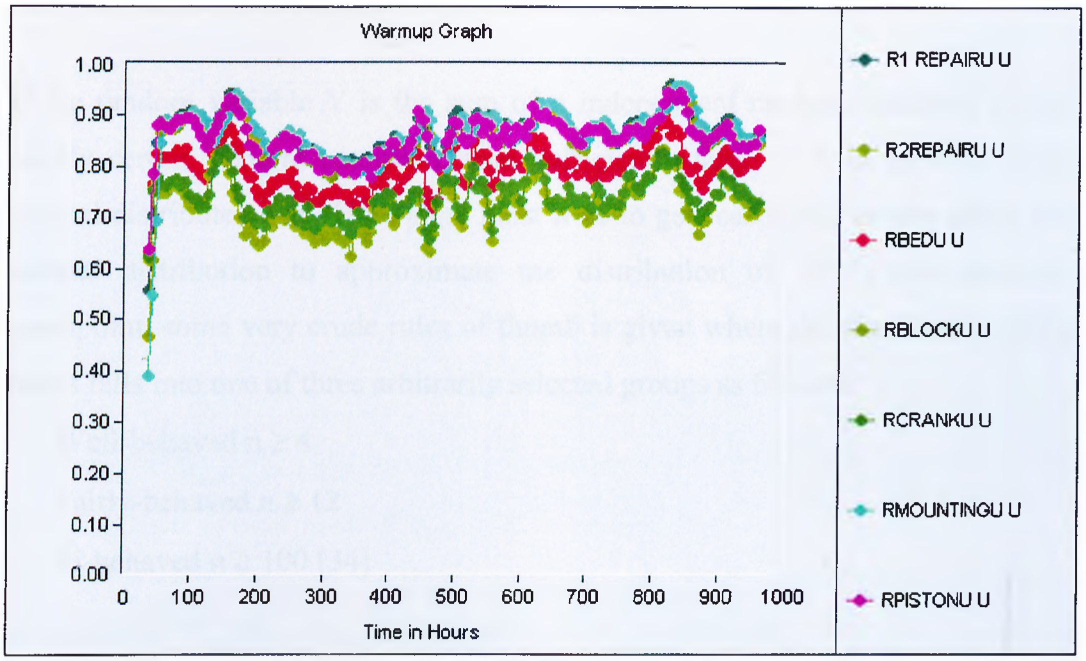


Figure B.1-3 Unimog motors sub-section`s utilisations versus time

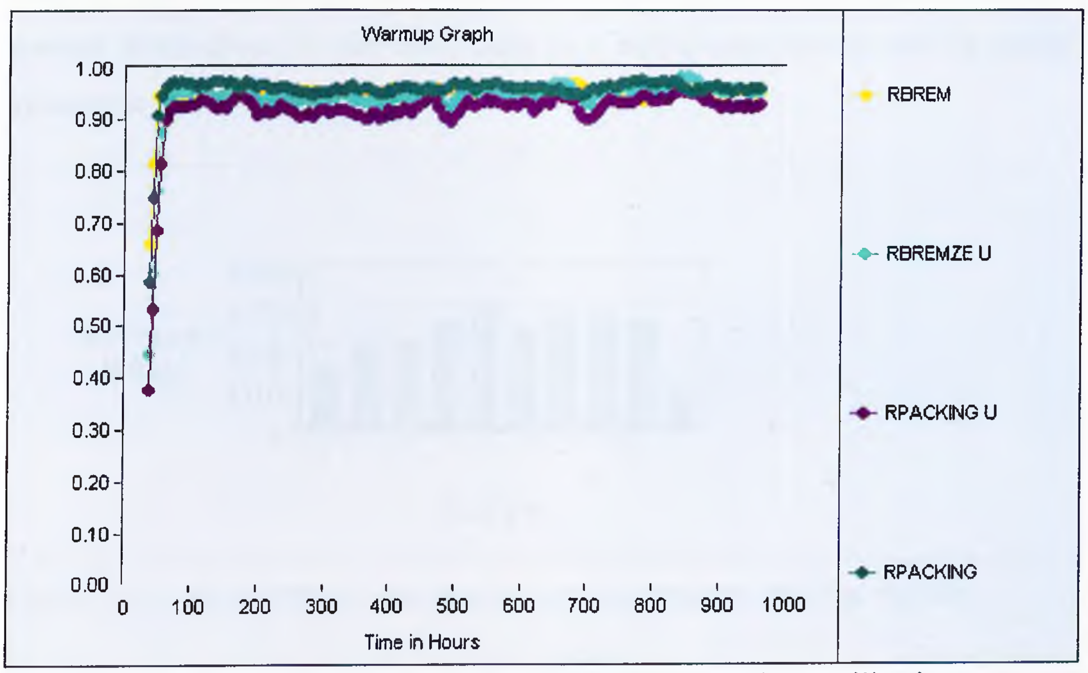


Figure B.1-4 Renovation unit`s testing and packing sections utilisations versus time and processing time versus time.

2. Normality Check

If the random variable Y is the sum of n independent random variables which satisfy certain general condition, then sufficiently large n, Y is approximately normal distributed and "How large must n be to get reasonable results using the normal distribution to approximate the distribution of Y?" From practical standpoint, some very crude rules of thumb is given where the distribution of X_i terms falls into one of three arbitrarily selected groups as follows:

- Well-behaved $n \geq 4$
- Fairly-behaved $n \geq 12$
- Ill-behaved $n \geq 100$ [34]

Therefore we get the time in system results and make the histogram of the data as seen in Figure B.2-1. The distribution of X_i does not radically depart from the normal distribution. In our case, there is a bell-shaped density that is nearly symmetric and n = 2303.

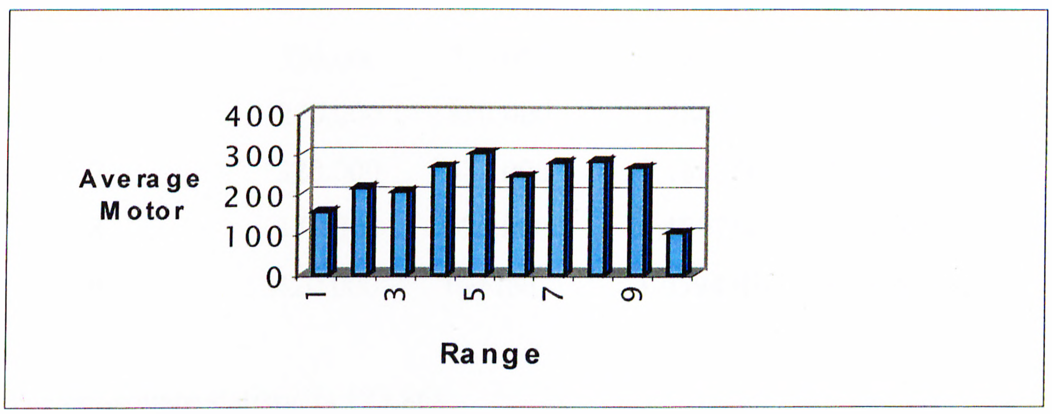


Figure B.2-1. Histogram of time in system measurements for large motors.

Also, we made the goodness-of-fit test to provide helpful guidance for evaluating the suitability. Chi-square test is applied to formalise the normality. This test is valid for the large sample sizes.

SAMPLE CHARACTERISTIC	VALUE
NUMBER OF OBSERVATIONS	2303
MINIMUM OBSERVATION	21.9167
MAXIMUM OBSERVATION	467.450
MEAN	251.800
MEDIAN	251.367
VARIANCE	1.60986E+ 4
COEFFICIENT OF VARIATION	.50389
COEFFICIENT OF SKEWNESS	-.09307
COEFFICIENT OF KURTOSIS	1.87088

CHI-SQUARE TEST TABLEAU :

	INTERVAL RANGE		FREQUENCIES	
INTERVAL	FROM	THROUGH	OBSERVED	MODEL
1	- INFIN	70.0000	1.07686E-1	7.59509E-2
2	70.0000	120.000	9.29223E-2	7.35040E-2
3	120.000	170.000	9.24881E-2	1.10105E-1
4	170.000	220.000	1.18541E-1	1.41490E-1
5	220.000	270.000	1.18541E-1	1.55979E-1
6	270.000	320.000	1.28094E-1	1.47514E-1
7	320.000	370.000	1.16804E-1	1.19681E-1
8	370.000	420.000	1.18975E-1	8.32987E-2
9	420.000	INFINI	1.05949E-1	9.24762E-2

The chi-square statistic is 123.868.

The theoretical probability of exceeding the observed statistic is with degrees of freedom 6 less than 0.001 and 8 less than 0.001. Therefore, we can safely assumed the normality.

3. Summary Statistics of the Existing System

No factors changed				0,9	0,95	0,99
LARGE TYPE MOT	Average	6,2966	CI Low	6,278548	6,274322	6,264596
	Std. Dev.	0,031142	CI High	6,314652	6,318878	6,328604
	Minimum	6,222	# of Runs	10	10	10
	Maximum	6,324				
	Median	6,3045				
	# of Runs	10				
SMALL TYPE MOT	Average	5,2603	CI Low	5,245087	5,241527	5,23333
	Std. Dev.	0,026243	CI High	5,275513	5,279073	5,28727
	Minimum	5,196	# of Runs	10	10	10
	Maximum	5,287				
	Median	5,2655				
	# of Runs	10				
UNIMOG TYPE MOT	Average	5,8707	CI Low	5,851462	5,846959	5,836594
	Std. Dev.	0,033187	CI High	5,889938	5,894441	5,904806
	Minimum	5,825	# of Runs	10	10	10
	Maximum	5,91				
	Median	5,8795				
	# of Runs	10				
Q1CARB	Average	127823,9	CI Low	123985,6	123087,2	121019,1
	Std. Dev.	6621,422	CI High	131662,2	132560,5	134628,6
	Minimum	114204,7	# of Runs	10	10	10
	Maximum	136096,8				
	Median	129700,5				
	# of Runs	10				
Q1DISMANTLE	Average	101364,2	CI Low	99749,07	99371,04	98500,8
	Std. Dev.	2786,275	CI High	102979,4	103357,4	104227,6
	Minimum	97910,54	# of Runs	10	10	10
	Maximum	107475,8				
	Median	101420				
	# of Runs	10				
Q1ELECT	Average	170449,9	CI Low	166532,7	165615,8	163505,2
	Std. Dev.	6757,519	CI High	174367,1	175283,9	177394,5
	Minimum	155834,9	# of Runs	10	10	10
	Maximum	177612,9				
	Median	172010,7				
	# of Runs	10				
Q1REPAIRB	Average	7975,14	CI Low	7820,271	7784,024	7700,581

	Std. Dev.	267,162	CI High	8130,009	8166,256	8249,699
	Minimum	7515,7	# of Runs	10	10	10
	Maximum	8369,37				
	Median	7989,17				
	# of Runs	10				
Q1REPAIRS	Average	7302,77	CI Low	6898,234	6803,552	6585,589
	Std. Dev.	697,859	CI High	7707,306	7801,988	8019,951
	Minimum	6448,08	# of Runs	10	10	10
	Maximum	8768,3				
	Median	7028,66				
	# of Runs	10				
Q1REPAIRU	Average	15587,98	CI Low	15528,06	15514,04	15481,75
	Std. Dev.	103,3666	CI High	15647,9	15661,92	15694,21
	Minimum	15473,6	# of Runs	10	10	10
	Maximum	15849,08				
	Median	15569,68				
	# of Runs	10				
Q1WASH	Average	13140,2	CI Low	12528,6	12385,45	12055,92
	Std. Dev.	1055,067	CI High	13751,8	13894,95	14224,48
	Minimum	12046,08	# of Runs	10	10	10
	Maximum	15073,37				
	Median	12848,15				
	# of Runs	10				
Q2CARB	Average	31881,56	CI Low	28978,17	28298,63	26734,28
	Std. Dev.	5008,596	CI High	34784,95	35464,49	37028,84
	Minimum	20649,95	# of Runs	10	10	10
	Maximum	40379,66				
	Median	32027,78				
	# of Runs	10				
Q2DISMANTLE	Average	27518,98	CI Low	22755,51	21640,61	19074,06
	Std. Dev.	8217,396	CI High	32282,45	33397,35	35963,9
	Minimum	7816,87	# of Runs	10	10	10
	Maximum	38350,35				
	Median	28499,25				
	# of Runs	10				
Q2ELECT	Average	386770,3	CI Low	359678,4	353337,5	338740,5
	Std. Dev.	46735,82	CI High	413862,2	420203,1	434800,2
	Minimum	321417,8	# of Runs	10	10	10
	Maximum	467128,7				
	Median	376868,5				
	# of Runs	10				
Q2REPA0RU	Average	12191,79	CI Low	12164,87	12158,57	12144,07
	Std. Dev.	46,43529	CI High	12218,71	12225,01	12239,51
	Minimum	12114,75	# of Runs	10	10	10
	Maximum	12260,16				
	Median	12195,16				
	# of Runs	10				
Q2REPAIRB	Average	19489,6	CI Low	19446,27	19436,13	19412,78

	Std. Dev.	74,75204	CI High	19532,93	19543,07	19566,42
	Minimum	19371,73	# of Runs	10	10	10
	Maximum	19598,2				
	Median	19480,97				
	# of Runs	10				
Q2REPAIRS	Average	6306,67	CI Low	6231,834	6214,318	6173,996
	Std. Dev.	129,099	CI High	6381,506	6399,022	6439,344
	Minimum	6112,01	# of Runs	10	10	10
	Maximum	6595,99				
	Median	6299,21				
	# of Runs	10				
Q2REPAIRU	Average	12191,79	CI Low	12164,87	12158,57	12144,07
	Std. Dev.	46,43529	CI High	12218,71	12225,01	12239,51
	Minimum	12114,75	# of Runs	10	10	10
	Maximum	12260,16				
	Median	12195,16				
	# of Runs	10				
Q2WASH	Average	12191,79	CI Low	12164,87	12158,57	12144,07
	Std. Dev.	46,43529	CI High	12218,71	12225,01	12239,51
	Minimum	12114,75	# of Runs	10	10	10
	Maximum	12260,16				
	Median	12195,16				
	# of Runs	10				
Q3CARB	Average	46027,65	CI Low	43093,82	42407,15	40826,4
	Std. Dev.	5061,112	CI High	48961,48	49648,15	51228,9
	Minimum	39260,55	# of Runs	10	10	10
	Maximum	55964,1				
	Median	45277,23				
	# of Runs	10				
Q3DISMANTLE	Average	7021,06	CI Low	6909,784	6883,739	6823,784
	Std. Dev.	191,961	CI High	7132,336	7158,381	7218,336
	Minimum	6686,21	# of Runs	10	10	10
	Maximum	7296,5				
	Median	7006,17				
	# of Runs	10				
Q3ELECT	Average	81461,39	CI Low	64716,52	60797,34	51775,23
	Std. Dev.	28886,35	CI High	98206,26	102125,4	111147,6
	Minimum	53756,26	# of Runs	10	10	10
	Maximum	150391,5				
	Median	71640,79				
	# of Runs	10				
Q3WASH	Average	11799,02	CI Low	11497,51	11426,94	11264,49
	Std. Dev.	520,1302	CI High	12100,53	12171,1	12333,55
	Minimum	11215,27	# of Runs	10	10	10
	Maximum	12693,11				
	Median	11672,3				
	# of Runs	10				
QBEDB	Average	7555,39	CI Low	7387,39	7347,99	7257,49

	Std. Dev.	289,9	CI High	7723,39	7762,79	7853,29
	Minimum	7237,48	# of Runs	10	10	10
	Maximum	8159,17				
	Median	7568,15				
	# of Runs	10				
QBEDS	Average	2543,53	CI Low	2476,543	2460,865	2424,772
	Std. Dev.	115,558	CI High	2610,517	2626,195	2662,288
	Minimum	2408,99	# of Runs	10	10	10
	Maximum	2793,28				
	Median	2516,04				
	# of Runs	10				
QBEDU	Average	9250,13	CI Low	8995,033	8935,326	8797,88
	Std. Dev.	440,065	CI High	9505,227	9564,934	9702,38
	Minimum	8667,42	# of Runs	10	10	10
	Maximum	9900,86				
	Median	9215,55				
	# of Runs	10				
QBLOCKB	Average	185063,2	CI Low	181749,5	180973,9	179188,5
	Std. Dev.	5716,437	CI High	188376,9	189152,5	190937,9
	Minimum	175708,1	# of Runs	10	10	10
	Maximum	195335,4				
	Median	184242,9				
	# of Runs	10				
QBLOCKS	Average	26800,68	CI Low	24054,43	23411,66	21931,99
	Std. Dev.	4737,518	CI High	29546,93	30189,7	31669,37
	Minimum	15438,59	# of Runs	10	10	10
	Maximum	33469,47				
	Median	27350,31				
	# of Runs	10				
QBLOCKU	Average	11073,59	CI Low	10715,62	10631,84	10438,97
	Std. Dev.	617,526	CI High	11431,56	11515,34	11708,21
	Minimum	10399,71	# of Runs	10	10	10
	Maximum	12152,88				
	Median	10934,37				
	# of Runs	10				
QBREMZE	Average	8820,17	CI Low	8811,685	8809,699	8805,127
	Std. Dev.	14,638	CI High	8828,655	8830,641	8835,213
	Minimum	8798,21	# of Runs	10	10	10
	Maximum	8847,88				
	Median	8820,565				
	# of Runs	10				
QCRANKB	Average	160600,7	CI Low	159159,7	158822,4	158046
	Std. Dev.	2485,812	CI High	162041,7	162378,9	163155,3
	Minimum	156932,4	# of Runs	10	10	10
	Maximum	163694,1				
	Median	160926,2				
	# of Runs	10				
QCRANKS	Average	163942,1	CI Low	156393,3	154626,5	150559,2

	Std. Dev.	13022,35	CI High	171490,9	173257,7	177325
	Minimum	139305	# of Runs	10	10	10
	Maximum	185740,7				
	Median	163449,1				
	# of Runs	10				
QCRANKU	Average	4031,28	CI Low	3970,001	3955,658	3922,641
	Std. Dev.	105,712	CI High	4092,559	4106,902	4139,919
	Minimum	3940,31	# of Runs	10	10	10
	Maximum	4280,4				
	Median	3989,88				
	# of Runs	10				
QMOUNTINGB	Average	5863	CI Low	5851,242	5848,49	5842,155
	Std. Dev.	20,2838	CI High	5874,758	5877,51	5883,845
	Minimum	5832,66	# of Runs	10	10	10
	Maximum	5895				
	Median	5856,78				
	# of Runs	10				
QMOUNTINGS	Average	4291,53	CI Low	4284,697	4283,098	4279,417
	Std. Dev.	11,787	CI High	4298,363	4299,962	4303,643
	Minimum	4276,61	# of Runs	10	10	10
	Maximum	4306,3				
	Median	4291,95				
	# of Runs	10				
QMOUNTINGU	Average	9573	CI Low	9533,265	9523,965	9502,556
	Std. Dev.	68,5462	CI High	9612,735	9622,035	9643,444
	Minimum	9477,85	# of Runs	10	10	10
	Maximum	9666,28				
	Median	9574,825				
	# of Runs	10				
QPACK	Average	4357,23	CI Low	4354,174	4353,459	4351,812
	Std. Dev.	5,27156	CI High	4360,286	4361,001	4362,648
	Minimum	4350,29	# of Runs	10	10	10
	Maximum	4367,6				
	Median	4355,78				
	# of Runs	10				
QPISTONB	Average	34616,73	CI Low	27173,96	25431,96	21421,81
	Std. Dev.	12839,42	CI High	42059,5	43801,5	47811,65
	Minimum	14697,24	# of Runs	10	10	10
	Maximum	62704,96				
	Median	33163,82				
	# of Runs	10				
QPISTONS	Average	4131,21	CI Low	4075,869	4062,916	4033,098
	Std. Dev.	95,4684	CI High	4186,551	4199,504	4229,322
	Minimum	4034,94	# of Runs	10	10	10
	Maximum	4340,46				
	Median	4119,33				
	# of Runs	10				
QPISTONU	Average	11559,3	CI Low	10674,5	10467,41	9990,68

	Std. Dev.	1526,358	CI High	12444,1	12651,19	13127,92
	Minimum	10264,48	# of Runs	10	10	10
	Maximum	15518,96				
	Median	11113,92				
	# of Runs	10				
R1CARB U	Average	0,8944	CI Low	0,889199	0,887982	0,88518
	Std. Dev.	0,008972	CI High	0,899601	0,900818	0,90362
	Minimum	0,884	# of Runs	10	10	10
	Maximum	0,912				
	Median	0,8915				
	# of Runs	10				
R1CARB	Average	15336,2	CI Low	15313,9	15308,68	15296,66
	Std. Dev.	38,47464	CI High	15358,5	15363,72	15375,74
	Minimum	15288,9	# of Runs	10	10	10
	Maximum	15406,29				
	Median	15333,04				
	# of Runs	10				
R1ELECT U	Average	0,9401	CI Low	0,935065	0,933887	0,931175
	Std. Dev.	0,008685	CI High	0,945135	0,946313	0,949025
	Minimum	0,927	# of Runs	10	10	10
	Maximum	0,953				
	Median	0,94				
	# of Runs	10				
R1 ELECT	Average	16123,62	CI Low	16112,48	16109,87	16103,87
	Std. Dev.	19,21505	CI High	16134,76	16137,37	16143,37
	Minimum	16097,89	# of Runs	10	10	10
	Maximum	16157,54				
	Median	16121,75				
	# of Runs	10				
R1REPAIRB U	Average	0,9691	CI Low	0,966051	0,965338	0,963695
	Std. Dev.	0,005259	CI High	0,972149	0,972862	0,974505
	Minimum	0,959	# of Runs	10	10	10
	Maximum	0,976				
	Median	0,9685				
	# of Runs	10				
R1 REPAIRB	Average	5449,51	CI Low	5441,32	5439,403	5434,99
	Std. Dev.	14,1292	CI High	5457,7	5459,617	5464,03
	Minimum	5421,77	# of Runs	10	10	10
	Maximum	5471,22				
	Median	5452,41				
	# of Runs	10				
R1REPAIRS U	Average	0,9337	CI Low	0,930608	0,929884	0,928218
	Std. Dev.	0,005334	CI High	0,936792	0,937516	0,939182
	Minimum	0,924	# of Runs	10	10	10
	Maximum	0,94				
	Median	0,935				
	# of Runs	10				
R1 REPAIRS	Average	5749,26	CI Low	5741,819	5740,077	5736,067

	Std. Dev.	12,8372	CI High	5756,701	5758,443	5762,453
	Minimum	5730,48	# of Runs	10	10	10
	Maximum	5767,37				
	Median	5749,78				
	# of Runs	10				
R1 REPAIRU U	Average	0,8405	CI Low	0,83441	0,83299	0,8297
	Std. Dev.	0,0105	CI High	0,84659	0,84801	0,8513
	Minimum	0,826	# of Runs	10	10	10
	Maximum	0,853				
	Median	0,8445				
	# of Runs	10				
R1 REPAIRU	Average	14841,08	CI Low	14828,12	14825,09	14818,1
	Std. Dev.	22,35679	CI High	14854,04	14857,07	14864,06
	Minimum	14787,74	# of Runs	10	10	10
	Maximum	14874,92				
	Median	14844,96				
	# of Runs	10				
R2CARB U	Average	0,8281	CI Low	0,82074	0,81901	0,815
	Std. Dev.	0,0127	CI High	0,83546	0,83719	0,8412
	Minimum	0,811	# of Runs	10	10	10
	Maximum	0,854				
	Median	0,826				
	# of Runs	10				
R2 CARB	Average	10746,82	CI Low	10730,46	10726,63	10717,81
	Std. Dev.	28,22772	CI High	10763,18	10767,01	10775,83
	Minimum	10712,28	# of Runs	10	10	10
	Maximum	10794,81				
	Median	10741,69				
	# of Runs	10				
R2ELECT U	Average	1	CI Low	1	1	1
	Std. Dev.	0	CI High	1	1	1
	Minimum	1	# of Runs	10	10	10
	Maximum	1				
	Median	1				
	# of Runs	10				
R2 ELECT	Average	16129,17	CI Low	16109,21	16104,54	16093,78
	Std. Dev.	34,43711	CI High	16149,13	16153,8	16164,56
	Minimum	16071,95	# of Runs	10	10	10
	Maximum	16186,66				
	Median	16122,49				
	# of Runs	10				
R2REPAIRB U	Average	0,8646	CI Low	0,861318	0,86055	0,858782
	Std. Dev.	0,005661	CI High	0,867882	0,86865	0,870418
	Minimum	0,854	# of Runs	10	10	10
	Maximum	0,871				
	Median	0,8665				
	# of Runs	10				
R2 REPAIRB	Average	19406,39	CI Low	19372,32	19364,35	19345,99

	Std. Dev.	58,7731	CI High	19440,46	19448,43	19466,79
	Minimum	19314,45	# of Runs	10	10	10
	Maximum	19494,35				
	Median	19403,18				
	# of Runs	10				
R2REPAIRS U	Average	0,924	CI Low	0,920642	0,919856	0,918047
	Std. Dev.	0,005793	CI High	0,927358	0,928144	0,929953
	Minimum	0,91	# of Runs	10	10	10
	Maximum	0,931				
	Median	0,9255				
	# of Runs	10				
R2 REPAIRS	Average	5691,15	CI Low	5684,268	5682,657	5678,949
	Std. Dev.	11,8719	CI High	5698,032	5699,643	5703,351
	Minimum	5671,88	# of Runs	10	10	10
	Maximum	5707,02				
	Median	5692,115				
	# of Runs	10				
R2REPAIRU U	Average	0,8184	CI Low	0,813207	0,811991	0,809193
	Std. Dev.	0,008959	CI High	0,823593	0,824809	0,827607
	Minimum	0,805	# of Runs	10	10	10
	Maximum	0,828				
	Median	0,8215				
	# of Runs	10				
R2 REPAIRU	Average	11571,33	CI Low	11560,38	11557,82	11551,92
	Std. Dev.	18,89093	CI High	11582,28	11584,84	11590,74
	Minimum	11539,17	# of Runs	10	10	10
	Maximum	11594,74				
	Median	11574,24				
	# of Runs	10				
R3CARB U	Average	0,9849	CI Low	0,983936	0,98371	0,983191
	Std. Dev.	0,001663	CI High	0,985864	0,98609	0,986609
	Minimum	0,983	# of Runs	10	10	10
	Maximum	0,988				
	Median	0,9845				
	# of Runs	10				
R3 CARB	Average	14325,34	CI Low	14306,83	14302,5	14292,53
	Std. Dev.	31,92778	CI High	14343,85	14348,18	14358,15
	Minimum	14288,37	# of Runs	10	10	10
	Maximum	14381,68				
	Median	14311,19				
	# of Runs	10				
R3ELECT U	Average	0,9969	CI Low	0,995206	0,994809	0,993896
	Std. Dev.	0,002923	CI High	0,998594	0,998991	0,999904
	Minimum	0,992	# of Runs	10	10	10
	Maximum	1				
	Median	0,997				
	# of Runs	10				
R3 ELECT	Average	12306,35	CI Low	12294,32	12291,51	12285,03

	Std. Dev.	20,74649	CI High	12318,38	12321,19	12327,67
	Minimum	12270,74	# of Runs	10	10	10
	Maximum	12334,34				
	Median	12303,44				
	# of Runs	10				
RBEDB U	Average	0,849	CI Low	0,846134	0,845463	0,843919
	Std. Dev.	0,004944	CI High	0,851866	0,852537	0,854081
	Minimum	0,841	# of Runs	10	10	10
	Maximum	0,855				
	Median	0,85				
	# of Runs	10				
R BEDB	Average	7125,06	CI Low	7113,383	7110,65	7104,359
	Std. Dev.	20,1434	CI High	7136,737	7139,47	7145,761
	Minimum	7082,31	# of Runs	10	10	10
	Maximum	7152,42				
	Median	7121,505				
	# of Runs	10				
RBEDS U	Average	0,5573	CI Low	0,555614	0,55522	0,554311
	Std. Dev.	0,002908	CI High	0,558986	0,55938	0,560289
	Minimum	0,551	# of Runs	10	10	10
	Maximum	0,562				
	Median	0,558				
	# of Runs	10				
R BEDS	Average	2289,01	CI Low	2285,286	2284,415	2282,408
	Std. Dev.	6,4238	CI High	2292,734	2293,605	2295,612
	Minimum	2280,75	# of Runs	10	10	10
	Maximum	2300,08				
	Median	2291,14				
	# of Runs	10				
RBEDU U	Average	0,788	CI Low	0,782473	0,78118	0,778202
	Std. Dev.	0,009534	CI High	0,793527	0,79482	0,797798
	Minimum	0,773	# of Runs	10	10	10
	Maximum	0,798				
	Median	0,792				
	# of Runs	10				
R BEDU	Average	5573,96	CI Low	5567,418	5565,886	5562,361
	Std. Dev.	11,2862	CI High	5580,502	5582,034	5585,559
	Minimum	5560,9	# of Runs	10	10	10
	Maximum	5589,01				
	Median	5568,7				
	# of Runs	10				
RBLOCKB U	Average	0,9548	CI Low	0,952098	0,951465	0,950009
	Std. Dev.	0,004662	CI High	0,957502	0,958135	0,959591
	Minimum	0,946	# of Runs	10	10	10
	Maximum	0,959				
	Median	0,9565				
	# of Runs	10				
R BLOCKB	Average	7814,77	CI Low	7801,882	7798,865	7791,921

	Std. Dev.	22,233	CI High	7827,658	7830,675	7837,619
	Minimum	7791,84	# of Runs	10	10	10
	Maximum	7861,52				
	Median	7805,325				
	# of Runs	10				
RBLOCKS U	Average	0,7804	CI Low	0,776651	0,775774	0,773754
	Std. Dev.	0,006467	CI High	0,784149	0,785026	0,787046
	Minimum	0,769	# of Runs	10	10	10
	Maximum	0,789				
	Median	0,7825				
	# of Runs	10				
RBLOCKS	Average	7780,61	CI Low	7765,414	7761,857	7753,67
	Std. Dev.	26,2144	CI High	7795,806	7799,363	7807,55
	Minimum	7734,15	# of Runs	10	10	10
	Maximum	7805,94				
	Median	7784,02				
	# of Runs	10				
RBLOCKU U	Average	0,698	CI Low	0,693984	0,693044	0,69088
	Std. Dev.	0,006928	CI High	0,702016	0,702956	0,70512
	Minimum	0,688	# of Runs	10	10	10
	Maximum	0,706				
	Median	0,7015				
	# of Runs	10				
RBLOCKU	Average	7402,57	CI Low	7396,054	7394,529	7391,018
	Std. Dev.	11,241	CI High	7409,086	7410,611	7414,122
	Minimum	7389,39	# of Runs	10	10	10
	Maximum	7426,86				
	Median	7400,63				
	# of Runs	10				
RBREM	Average	8819	CI Low	8810,613	8808,65	8804,132
	Std. Dev.	14,4679	CI High	8827,387	8829,35	8833,869
	Minimum	8797,48	# of Runs	10	10	10
	Maximum	8846,17				
	Median	8819,555				
	# of Runs	10				
RBREMZE U	Average	0,7092	CI Low	0,70679	0,706226	0,704927
	Std. Dev.	0,004158	CI High	0,71161	0,712174	0,713473
	Minimum	0,704	# of Runs	10	10	10
	Maximum	0,715				
	Median	0,709				
	# of Runs	10				
RCRANKB U	Average	0,996	CI Low	0,99286	0,992126	0,990434
	Std. Dev.	0,005416	CI High	0,99914	0,999874	1,001566
	Minimum	0,984	# of Runs	10	10	10
	Maximum	1				
	Median	0,9975				
	# of Runs	10				
RCRANKB	Average	8355,61	CI Low	8345,607	8343,266	8337,876

	Std. Dev.	17,2563	CI High	8365,613	8367,954	8373,344
	Minimum	8337,18	# of Runs	10	10	10
	Maximum	8385,67				
	Median	8354,48				
	# of Runs	10				
RCRANKS U	Average	0,9954	CI Low	0,992506	0,991828	0,990269
	Std. Dev.	0,004993	CI High	0,998294	0,998972	1,000531
	Minimum	0,983	# of Runs	10	10	10
	Maximum	1				
	Median	0,997				
	# of Runs	10				
RCRANKS	Average	6129,28	CI Low	6123,049	6121,591	6118,233
	Std. Dev.	10,7491	CI High	6135,511	6136,969	6140,327
	Minimum	6115,74	# of Runs	10	10	10
	Maximum	6147,92				
	Median	6130,285				
	# of Runs	10				
RCRANKU U	Average	0,533	CI Low	0,529587	0,528788	0,526949
	Std. Dev.	0,005888	CI High	0,536413	0,537212	0,539051
	Minimum	0,522	# of Runs	10	10	10
	Maximum	0,539				
	Median	0,5365				
	# of Runs	10				
RCRANKU	Average	3769,91	CI Low	3763,621	3762,149	3758,76
	Std. Dev.	10,8497	CI High	3776,199	3777,671	3781,06
	Minimum	3757,23	# of Runs	10	10	10
	Maximum	3793,17				
	Median	3771,51				
	# of Runs	10				
RDISMANTLE U	Average	0,3758	CI Low	0,373931	0,373493	0,372486
	Std. Dev.	0,003225	CI High	0,377669	0,378107	0,379114
	Minimum	0,369	# of Runs	10	10	10
	Maximum	0,38				
	Median	0,376				
	# of Runs	10				
R DISMANTLE	Average	2727,38	CI Low	2713,594	2710,367	2702,939
	Std. Dev.	23,7828	CI High	2741,166	2744,393	2751,821
	Minimum	2675,56	# of Runs	10	10	10
	Maximum	2758,71				
	Median	2727,2				
	# of Runs	10				
RMOUNTINGB U	Average	0,6718	CI Low	0,669374	0,668806	0,667499
	Std. Dev.	0,004185	CI High	0,674226	0,674794	0,676101
	Minimum	0,662	# of Runs	10	10	10
	Maximum	0,676				
	Median	0,673				
	# of Runs	10				
RMOUNTINGB	Average	5665,06	CI Low	5657,043	5655,166	5650,846

	Std. Dev.	13,8308	CI High	5673,077	5674,954	5679,274
	Minimum	5647,01	# of Runs	10	10	10
	Maximum	5683,96				
	Median	5662,785				
	# of Runs	10				
RMOUNTINGS U	Average	0,6954	CI Low	0,693583	0,693158	0,692179
	Std. Dev.	0,003134	CI High	0,697217	0,697642	0,698621
	Minimum	0,689	# of Runs	10	10	10
	Maximum	0,7				
	Median	0,696				
	# of Runs	10				
RMOUNTINGS	Average	4284,78	CI Low	4277,995	4276,406	4272,75
	Std. Dev.	11,7055	CI High	4291,565	4293,154	4296,81
	Minimum	4270,61	# of Runs	10	10	10
	Maximum	4299,95				
	Median	4285,78				
	# of Runs	10				
RMOUNTINGU U	Average	0,8275	CI Low	0,821506	0,820103	0,81687
	Std. Dev.	0,01034	CI High	0,833494	0,834897	0,83813
	Minimum	0,812	# of Runs	10	10	10
	Maximum	0,841				
	Median	0,83				
	# of Runs	10				
RMOUNTINGU	Average	8776,99	CI Low	8765,923	8763,333	8757,37
	Std. Dev.	19,0918	CI High	8788,057	8790,648	8796,61
	Minimum	8752,51	# of Runs	10	10	10
	Maximum	8807,33				
	Median	8773,33				
	# of Runs	10				
RPACKING U	Average	0,6935	CI Low	0,691259	0,690734	0,689527
	Std. Dev.	0,003866	CI High	0,695741	0,696266	0,697473
	Minimum	0,689	# of Runs	10	10	10
	Maximum	0,7				
	Median	0,693				
	# of Runs	10				
RPACKING	Average	4314,82	CI Low	4312,351	4311,773	4310,443
	Std. Dev.	4,25885	CI High	4317,289	4317,867	4319,197
	Minimum	4310,55	# of Runs	10	10	10
	Maximum	4324,81				
	Median	4313,51				
	# of Runs	10				
RPISTONB U	Average	0,9913	CI Low	0,987593	0,986725	0,984728
	Std. Dev.	0,006395	CI High	0,995007	0,995875	0,997872
	Minimum	0,976	# of Runs	10	10	10
	Maximum	0,997				
	Median	0,9925				
	# of Runs	10				
RPISTONB	Average	8356,62	CI Low	8339,105	8335,005	8325,568

	Std. Dev.	30,2158	CI High	8374,136	8378,235	8387,672
	Minimum	8310,98	# of Runs	10	10	10
	Maximum	8399,84				
	Median	8355,225				
	# of Runs	10				
RPISTONS	Average	4024,89	CI Low	4019,107	4017,753	4014,637
	Std. Dev.	9,97698	CI High	4030,673	4032,027	4035,143
	Minimum	4009,31	# of Runs	10	10	10
	Maximum	4045,35				
	Median	4022,155				
	# of Runs	10				
RPISTONS U	Average	0,6539	CI Low	0,652162	0,651755	0,650819
	Std. Dev.	0,002998	CI High	0,655638	0,656045	0,656981
	Minimum	0,647	# of Runs	10	10	10
	Maximum	0,658				
	Median	0,654				
	# of Runs	10				
RPISTONU U	Average	0,8687	CI Low	0,862503	0,861053	0,85771
	Std. Dev.	0,01069	CI High	0,874897	0,876347	0,87969
	Minimum	0,853	# of Runs	10	10	10
	Maximum	0,88				
	Median	0,873				
	# of Runs	10				
RPISTONU	Average	6145,42	CI Low	6138,651	6137,067	6133,42
	Std. Dev.	11,6764	CI High	6152,189	6153,773	6157,42
	Minimum	6132,08	# of Runs	10	10	10
	Maximum	6164,37				
	Median	6140,78				
	# of Runs	10				
RWASH	Average	4593,54	CI Low	4589,056	4588,006	4585,59
	Std. Dev.	7,736	CI High	4598,024	4599,074	4601,49
	Minimum	4583,65	# of Runs	10	10	10
	Maximum	4606,54				
	Median	4591,54				
	# of Runs	10				
RWASH U	Average	0,7572	CI Low	0,75387	0,75309	0,75129
	Std. Dev.	0,00575	CI High	0,76053	0,76131	0,76311
	Minimum	0,749	# of Runs	10	10	10
	Maximum	0,768				
	Median	0,7585				
	# of Runs	10				

Table B.4-1 Summary statistics of the existing system.

4. Summary Statistics of the Improved System.

BEDL		4
BLOCKL		6
CRANKL		6
CRANKS		4
PISTONL		5
REP2L		12
REP2S		5
REPL		3
REPS		4
Score		8466.876
Age		28
INCREASE1	Average	8466.876
	Std. Dev.	Infinity
	Minimum	8466.876
	Maximum	8466.876
	Median	8466.876
	# of Runs	1
PACKING TOTAL	Average	8448
	Std. Dev.	Infinity
	Minimum	8448
	Maximum	8448
	Median	8448
	# of Runs	1
Q1CARB	Average	449008.02
	Std. Dev.	Infinity
	Minimum	449008.02
	Maximum	449008.02
	Median	449008.02
	# of Runs	1
Q1DISMANTLE	Average	4876.41
	Std. Dev.	Infinity
	Minimum	4876.41
	Maximum	4876.41
	Median	4876.41
	# of Runs	1
Q1ELECT	Average	611862.14
	Std. Dev.	Infinity
	Minimum	611862.14
	Maximum	611862.14
	Median	611862.14
	# of Runs	1
Q1REPAIRB	Average	6079.83

	Std. Dev.	Infinity
	Minimum	6079.83
	Maximum	6079.83
	Median	6079.83
	# of Runs	1
Q1REPAIRS	Average	5945.05
	Std. Dev.	Infinity
	Minimum	5945.05
	Maximum	5945.05
	Median	5945.05
	# of Runs	1
Q1REPAIRU	Average	15340.53
	Std. Dev.	Infinity
	Minimum	15340.53
	Maximum	15340.53
	Median	15340.53
	# of Runs	1
Q1WASH	Average	4868.63
	Std. Dev.	Infinity
	Minimum	4868.63
	Maximum	4868.63
	Median	4868.63
	# of Runs	1
Q2CARB	Average	10947.86
	Std. Dev.	Infinity
	Minimum	10947.86
	Maximum	10947.86
	Median	10947.86
	# of Runs	1
Q2DISMANTLE	Average	0
	Std. Dev.	Infinity
	Minimum	0
	Maximum	0
	Median	0
	# of Runs	1
Q2ELECT	Average	411088.74
	Std. Dev.	Infinity
	Minimum	411088.74
	Maximum	411088.74
	Median	411088.74
	# of Runs	1
Q2REPA0RU	Average	12112.13
	Std. Dev.	Infinity
	Minimum	12112.13
	Maximum	12112.13

	Median	12112.13
	# of Runs	1
Q2REPAIRB	Average	19438.76
	Std. Dev.	Infinity
	Minimum	19438.76
	Maximum	19438.76
	Median	19438.76
	# of Runs	1
Q2REPAIRS	Average	5681.06
	Std. Dev.	Infinity
	Minimum	5681.06
	Maximum	5681.06
	Median	5681.06
	# of Runs	1
Q2REPAIRU	Average	12112.13
	Std. Dev.	Infinity
	Minimum	12112.13
	Maximum	12112.13
	Median	12112.13
	# of Runs	1
Q2WASH	Average	12112.13
	Std. Dev.	Infinity
	Minimum	12112.13
	Maximum	12112.13
	Median	12112.13
	# of Runs	1
Q3CARB	Average	28639.56
	Std. Dev.	Infinity
	Minimum	28639.56
	Maximum	28639.56
	Median	28639.56
	# of Runs	1
Q3DISMANTLE	Average	4847.78
	Std. Dev.	Infinity
	Minimum	4847.78
	Maximum	4847.78
	Median	4847.78
	# of Runs	1
Q3ELECT	Average	71595.09
	Std. Dev.	Infinity
	Minimum	71595.09
	Maximum	71595.09
	Median	71595.09
	# of Runs	1
Q3WASH	Average	4849.63

	Std. Dev.	Infinity
	Minimum	4849.63
	Maximum	4849.63
	Median	4849.63
	# of Runs	1
QBEDB	Average	8206.46
	Std. Dev.	Infinity
	Minimum	8206.46
	Maximum	8206.46
	Median	8206.46
	# of Runs	1
QBEDS	Average	2730.7
	Std. Dev.	Infinity
	Minimum	2730.7
	Maximum	2730.7
	Median	2730.7
	# of Runs	1
QBEDU	Average	6721.83
	Std. Dev.	Infinity
	Minimum	6721.83
	Maximum	6721.83
	Median	6721.83
	# of Runs	1
QBLOCKB	Average	7826.54
	Std. Dev.	Infinity
	Minimum	7826.54
	Maximum	7826.54
	Median	7826.54
	# of Runs	1
QBLOCKS	Average	8367.14
	Std. Dev.	Infinity
	Minimum	8367.14
	Maximum	8367.14
	Median	8367.14
	# of Runs	1
QBLOCKU	Average	8054.51
	Std. Dev.	Infinity
	Minimum	8054.51
	Maximum	8054.51
	Median	8054.51
	# of Runs	1
QBREMZE	Average	8864.48
	Std. Dev.	Infinity
	Minimum	8864.48
	Maximum	8864.48

	Std. Dev.	Infinity
	Minimum	8531.05
	Maximum	8531.05
	Median	8531.05
	# of Runs	1
QPISTONS	Average	4338.45
	Std. Dev.	Infinity
	Minimum	4338.45
	Maximum	4338.45
	Median	4338.45
	# of Runs	1
QPISTONU	Average	8191.35
	Std. Dev.	Infinity
	Minimum	8191.35
	Maximum	8191.35
	Median	8191.35
	# of Runs	1
R1CARB U	Average	1
	Std. Dev.	Infinity
	Minimum	1
	Maximum	1
	Median	1
	# of Runs	1
R1CARB	Average	15339.13
	Std. Dev.	Infinity
	Minimum	15339.13
	Maximum	15339.13
	Median	15339.13
	# of Runs	1
R1ELECT U	Average	1
	Std. Dev.	Infinity
	Minimum	1
	Maximum	1
	Median	1
	# of Runs	1
R1 ELECT	Average	16194.92
	Std. Dev.	Infinity
	Minimum	16194.92
	Maximum	16194.92
	Median	16194.92
	# of Runs	1
R1REPAIRB U	Average	0.848
	Std. Dev.	Infinity
	Minimum	0.848
	Maximum	0.848

	Median	0.848
	# of Runs	1
R1 REPAIRB	Average	5426.66
	Std. Dev.	Infinity
	Minimum	5426.66
	Maximum	5426.66
	Median	5426.66
	# of Runs	1
R1REPAIRS U	Average	0.761
	Std. Dev.	Infinity
	Minimum	0.761
	Maximum	0.761
	Median	0.761
	# of Runs	1
R1 REPAIRS	Average	5743.38
	Std. Dev.	Infinity
	Minimum	5743.38
	Maximum	5743.38
	Median	5743.38
	# of Runs	1
R1 REPAIRU U	Average	0.871
	Std. Dev.	Infinity
	Minimum	0.871
	Maximum	0.871
	Median	0.871
	# of Runs	1
R1 REPAIRU	Average	14800.18
	Std. Dev.	Infinity
	Minimum	14800.18
	Maximum	14800.18
	Median	14800.18
	# of Runs	1
R2CARB U	Average	0.856
	Std. Dev.	Infinity
	Minimum	0.856
	Maximum	0.856
	Median	0.856
	# of Runs	1
R2 CARB	Average	10770.94
	Std. Dev.	Infinity
	Minimum	10770.94
	Maximum	10770.94
	Median	10770.94
	# of Runs	1
R2ELECT U	Average	1

	Std. Dev.	Infinity
	Minimum	1
	Maximum	1
	Median	1
	# of Runs	1
R2 ELECT	Average	16146.77
	Std. Dev.	Infinity
	Minimum	16146.77
	Maximum	16146.77
	Median	16146.77
	# of Runs	1
R2REPAIRB U	Average	0.759
	Std. Dev.	Infinity
	Minimum	0.759
	Maximum	0.759
	Median	0.759
	# of Runs	1
R2 REPAIRB	Average	19431.08
	Std. Dev.	Infinity
	Minimum	19431.08
	Maximum	19431.08
	Median	19431.08
	# of Runs	1
R2REPAIRS U	Average	0.602
	Std. Dev.	Infinity
	Minimum	0.602
	Maximum	0.602
	Median	0.602
	# of Runs	1
R2 REPAIRS	Average	5676.01
	Std. Dev.	Infinity
	Minimum	5676.01
	Maximum	5676.01
	Median	5676.01
	# of Runs	1
R2REPAIRU U	Average	0.852
	Std. Dev.	Infinity
	Minimum	0.852
	Maximum	0.852
	Median	0.852
	# of Runs	1
R2 REPAIRU	Average	11567.1
	Std. Dev.	Infinity
	Minimum	11567.1
	Maximum	11567.1

	Median	11567.1
	# of Runs	1
R3CARB U	Average	0.983
	Std. Dev.	Infinity
	Minimum	0.983
	Maximum	0.983
	Median	0.983
	# of Runs	1
R3 CARB	Average	14350.62
	Std. Dev.	Infinity
	Minimum	14350.62
	Maximum	14350.62
	Median	14350.62
	# of Runs	1
R3ELECT U	Average	0.999
	Std. Dev.	Infinity
	Minimum	0.999
	Maximum	0.999
	Median	0.999
	# of Runs	1
R3 ELECT	Average	12305.08
	Std. Dev.	Infinity
	Minimum	12305.08
	Maximum	12305.08
	Median	12305.08
	# of Runs	1
RBEDB U	Average	0.836
	Std. Dev.	Infinity
	Minimum	0.836
	Maximum	0.836
	Median	0.836
	# of Runs	1
R BEDB	Average	7142.05
	Std. Dev.	Infinity
	Minimum	7142.05
	Maximum	7142.05
	Median	7142.05
	# of Runs	1
RBEDS U	Average	0.607
	Std. Dev.	Infinity
	Minimum	0.607
	Maximum	0.607
	Median	0.607
	# of Runs	1
R BEDS	Average	2289.19

	Std. Dev.	Infinity
	Minimum	2289.19
	Maximum	2289.19
	Median	2289.19
	# of Runs	1
RBEDU U	Average	0.817
	Std. Dev.	Infinity
	Minimum	0.817
	Maximum	0.817
	Median	0.817
	# of Runs	1
R BEDU	Average	5568.74
	Std. Dev.	Infinity
	Minimum	5568.74
	Maximum	5568.74
	Median	5568.74
	# of Runs	1
RBLOCKB U	Average	0.611
	Std. Dev.	Infinity
	Minimum	0.611
	Maximum	0.611
	Median	0.611
	# of Runs	1
R BLOCKB	Average	7826.44
	Std. Dev.	Infinity
	Minimum	7826.44
	Maximum	7826.44
	Median	7826.44
	# of Runs	1
RBLOCKS U	Average	0.827
	Std. Dev.	Infinity
	Minimum	0.827
	Maximum	0.827
	Median	0.827
	# of Runs	1
RBLOCKS	Average	7796.67
	Std. Dev.	Infinity
	Minimum	7796.67
	Maximum	7796.67
	Median	7796.67
	# of Runs	1
RBLOCKU U	Average	0.727
	Std. Dev.	Infinity
	Minimum	0.727
	Maximum	0.727

	Median	0.727
	# of Runs	1
RBLOCKU	Average	7435.73
	Std. Dev.	Infinity
	Minimum	7435.73
	Maximum	7435.73
	Median	7435.73
	# of Runs	1
RBREM	Average	8830.46
	Std. Dev.	Infinity
	Minimum	8830.46
	Maximum	8830.46
	Median	8830.46
	# of Runs	1
RBREMZE U	Average	0.814
	Std. Dev.	Infinity
	Minimum	0.814
	Maximum	0.814
	Median	0.814
	# of Runs	1
RCRANKB U	Average	0.651
	Std. Dev.	Infinity
	Minimum	0.651
	Maximum	0.651
	Median	0.651
	# of Runs	1
RCRANKB	Average	8334.19
	Std. Dev.	Infinity
	Minimum	8334.19
	Maximum	8334.19
	Median	8334.19
	# of Runs	1
RCRANKS U	Average	0.81
	Std. Dev.	Infinity
	Minimum	0.81
	Maximum	0.81
	Median	0.81
	# of Runs	1
RCRANKS	Average	6114.08
	Std. Dev.	Infinity
	Minimum	6114.08
	Maximum	6114.08
	Median	6114.08
	# of Runs	1
RCRANKU U	Average	0.552

	Std. Dev.	Infinity
	Minimum	0.552
	Maximum	0.552
	Median	0.552
	# of Runs	1
RCRANKU	Average	3765.45
	Std. Dev.	Infinity
	Minimum	3765.45
	Maximum	3765.45
	Median	3765.45
	# of Runs	1
RDISMANTLE U	Average	0.419
	Std. Dev.	Infinity
	Minimum	0.419
	Maximum	0.419
	Median	0.419
	# of Runs	1
R DISMANTLE	Average	2833.53
	Std. Dev.	Infinity
	Minimum	2833.53
	Maximum	2833.53
	Median	2833.53
	# of Runs	1
REN LARGE TOTAL	Average	3047
	Std. Dev.	Infinity
	Minimum	3047
	Maximum	3047
	Median	3047
	# of Runs	1
REN SMALL TOTAL	Average	3476
	Std. Dev.	Infinity
	Minimum	3476
	Maximum	3476
	Median	3476
	# of Runs	1
REN UNIMOG TOTAL	Average	1929
	Std. Dev.	Infinity
	Minimum	1929
	Maximum	1929
	Median	1929
	# of Runs	1
RMOUNTINGB U	Average	0.88
	Std. Dev.	Infinity
	Minimum	0.88
	Maximum	0.88

	Median	0.88
	# of Runs	1
RMOUNTINGB	Average	5679.89
	Std. Dev.	Infinity
	Minimum	5679.89
	Maximum	5679.89
	Median	5679.89
	# of Runs	1
RMOUNTINGS U	Average	0.76
	Std. Dev.	Infinity
	Minimum	0.76
	Maximum	0.76
	Median	0.76
	# of Runs	1
RMOUNTINGS	Average	4298.01
	Std. Dev.	Infinity
	Minimum	4298.01
	Maximum	4298.01
	Median	4298.01
	# of Runs	1
RMOUNTINGU U	Average	0.858
	Std. Dev.	Infinity
	Minimum	0.858
	Maximum	0.858
	Median	0.858
	# of Runs	1
RMOUNTINGU	Average	8747.25
	Std. Dev.	Infinity
	Minimum	8747.25
	Maximum	8747.25
	Median	8747.25
	# of Runs	1
RPACKING U	Average	0.798
	Std. Dev.	Infinity
	Minimum	0.798
	Maximum	0.798
	Median	0.798
	# of Runs	1
RPACKING	Average	4331.04
	Std. Dev.	Infinity
	Minimum	4331.04
	Maximum	4331.04
	Median	4331.04
	# of Runs	1
RPISTONB U	Average	0.785

	Std. Dev.	Infinity
	Minimum	0.785
	Maximum	0.785
	Median	0.785
	# of Runs	1
RPISTONB	Average	8374.37
	Std. Dev.	Infinity
	Minimum	8374.37
	Maximum	8374.37
	Median	8374.37
	# of Runs	1
RPISTONS	Average	4035.01
	Std. Dev.	Infinity
	Minimum	4035.01
	Maximum	4035.01
	Median	4035.01
	# of Runs	1
RPISTONS U	Average	0.713
	Std. Dev.	Infinity
	Minimum	0.713
	Maximum	0.713
	Median	0.713
	# of Runs	1
RPISTONU U	Average	0.901
	Std. Dev.	Infinity
	Minimum	0.901
	Maximum	0.901
	Median	0.901
	# of Runs	1
RPISTONU	Average	6135
	Std. Dev.	Infinity
	Minimum	6135
	Maximum	6135
	Median	6135
	# of Runs	1
RWASH	Average	4601.19
	Std. Dev.	Infinity
	Minimum	4601.19
	Maximum	4601.19
	Median	4601.19
	# of Runs	1
RWASH U	Average	0.817
	Std. Dev.	Infinity
	Minimum	0.817
	Maximum	0.817

	Median	0.817
	# of Runs	1
TESTED TOTAL	Average	8455
	Std. Dev.	Infinity
	Minimum	8455
	Maximum	8455
	Median	8455
	# of Runs	1

Table B.5-5 Summary statistics of the optimised system.

Appendix C

1. Additional Code For Pre-Control And Repair Section

```
begin Pselection arriving

    set acont to triangular 45,60,75
    set aadb to triangular 55,60,70
    if load type=Lbig then
    begin
    set adis1 to triangular 55,90,100
    set aw1  to triangular 54,90,100
    set are1 to triangular 180,300,330
    set arc1 to triangular 171,285,313
    set ad1  to triangular 50,120,150
    set ab   to triangular 87,146,160
    set acr   to triangular 90,150,180
    set abe  to triangular 79,133,146
    set ap   to triangular 93,156,171
    set a1r  to triangular 60,102,112
    set a2r  to triangular 216,361,397
    set am   to triangular 63,106,116
    set abr1 to triangular 102,171,188
    set ap1  to triangular 51,85,93
    end
      else if load type=Lsmall then
    begin
    set ad2  to triangular 60,125,160
    set aw2  to triangular 48,80,88
    set are2 to triangular 180,300,330
    set arc2 to triangular 120,200,220
    set asmall   to triangular 445,743,817
    set asb  to triangular 87,145,160
    set asc  to triangular 66,110,132
    set asbe to triangular 25,43,48
    set asp  to triangular 45,75,83
    set as1r to triangular 63,108,118
    set as2r to triangular 64,106,116
    set asm  to triangular 48,80,88
    set ap2  to triangular 43,72,79
    set abr2 to triangular 92,154,167

    end
```

```
else if load type=Lunimog then
begin
set adis3 to triangular 54,90,99
set aw3  to triangular 54,90,99
set are3 to triangular 137,229,252
set arc3 to triangular 160,267,293
set aub  to triangular 93,133,146
set auc  to triangular 48,68,74
set aube to triangular 70,100,110
set aup  to triangular 77,110,122
set au1r to triangular 186,266,292
set au2r to triangular 145,208,228
set aum  to triangular 110,158,173
set au   to triangular 936,1560,1716
set abr3 to triangular 105,175,192
set ap3  to triangular 54,90,99
end

if load type=Lbig then
begin
   set Atype to 1
   send to oneof(4:Pinspection,96:Pfirst)
end

else if load type=Lsmall then
begin
   set Atype to 2
   send to oneof(22:Pinspection,78:Pfirst)
end

else if load type=Lunimog then
begin
   set Atype to 3
   send to Pfirst
end
end

begin Pinspection arriving

      move into Qcontrol
      begin
         if load type=Lbig then
         begin
            use Rcontrol for acont min
         end
```

```
            else if load type=Lsmall then
            begin
               use Rcontrol for  acont min
            end
         end
         move into Qdefine
         begin
            if load type=Lbig then
            set define to  oneof(10:0, 90:1)
            else if load type=Lsmall then
            set define to  oneof(15:0, 85:1)
         end
         if define=0 then send to Pfirst
         else if define=1 then
            begin
               move into Qaddrepair
               if load type=Lbig then
               use Raddrepair for ad1  min
               else if load type=Lsmall then
               use Raddrepair for ad2 min
            end
            move into Qaddbremze
            use Raddbremze for aadb min
            set test to oneof(3:0,97:1)
            if test=0 then send to Pfirst
            else if test=1 then
            begin
               if load type=Lbig then
               inc Vprebig by 1
               else if load type=Lsmall then
               inc Vpresmall by 1
            end
            begin
                if load type=Lbig then print"Repaired Type "Atype , Vprebig to
message
                else if load type=Lsmall then print"Repaired Type "Atype,
Vpresmall to message
            end

               send to die
   end
```

2. Code For Combined Parallel Resources System.

```
begin Pselection arriving

   increment Vinsystem by 1
   set priority to Vinsystem
   set acont to triangular 45,60,75
   set aadb to triangular 55,60,70
   if load type=Lbig then
   begin
   set adis1 to triangular 55,90,100
   set aw1  to triangular 54,90,100
   set are1 to triangular 180,300,330
   set arc1 to triangular 171,285,313
   set ad1  to triangular 50,120,150
   set ab   to triangular 87,146,160
   set acr   to triangular 90,150,180
   set abe  to triangular 79,133,146
   set ap   to triangular 93,156,171
   set a1r  to triangular 60,102,112
   set a2r  to triangular 216,361,397
   set am   to triangular 63,106,116
   set abr1 to triangular 102,171,188
   set ap1  to triangular 51,85,93
   end
    else if load type=Lsmall then
   begin
   set ad2  to triangular 60,125,160
   set aw2  to triangular 48,80,88
   set are2 to triangular 180,300,330
   set arc2 to triangular 120,200,220
   set asmall   to triangular 445,743,817
   set asb  to triangular 87,145,160
   set asc  to triangular 66,110,132
   set asbe to triangular 25,43,48
   set asp  to triangular 45,75,83
   set as1r to triangular 63,108,118
   set as2r to triangular 64,106,116
   set asm  to triangular 48,80,88
   set ap2  to triangular 43,72,79
   set abr2 to triangular 92,154,167

   end
```

```
    else if load type=Lunimog then
    begin
    set adis3 to triangular 54,90,99
    set aw3  to triangular 54,90,99
    set are3 to triangular 137,229,252
    set arc3 to triangular 160,267,293
    set aub  to triangular 93,133,146
    set auc  to triangular 48,68,74
    set aube to triangular 70,100,110
    set aup  to triangular 77,110,122
    set au1r to triangular 186,266,292
    set au2r to triangular 145,208,228
    set aum  to triangular 110,158,173

    set au   to triangular 936,1560,1716
    set abr3 to triangular 105,175,192
    set ap3  to triangular 54,90,99
    end

    send to Pfirst
end

begin Pfirst arriving

    set Atimestamp to ac
    inc Vinsystem by 1
    if load type=Lbig then
    begin
       move into Q1dis
       wait until Vbigblock <=70
       wait until Vbigcrank <=70
    end
    else if load type=Lsmall then
    begin
        move into Q2dis
        wait until Vs <=100
        wait until Vsc <=100
    end
    else if load type=Lunimog then
    begin
       move into Q3dis
       wait until Vu <=100
    end
    if load type=Lbig then use Rdis for adis1 min
    else if load type=Lsmall then use Rdis for  adis2 min
```

```
    else if load type=Lunimog then use Rdis for adis3 min
   /* move into conv:geton
    travel to conv:getonwash*/
    send to Pw
end

begin Pw arriving
    if load type=Lbig then move into Q1wash
    else if load type=Lsmall then move into Q2wash
    else if load type=Lunimog then move into Q3wash
    if load type=Lbig then use Rw for aw1 min
    else if load type=Lsmall then use Rw for aw2 min
    else if load type=Lunimog then use Rw for aw3 min

    if load type=Lbig then

    begin
        move into Q1washf
     /* move into conv:getoffwash1
        travel to conv:getonren1*/
    end

    else if load type=Lsmall then

    begin
       /* move into conv:getoffwash2
        travel to conv:getonren2*/
    end

    else if load type=Lunimog then

    begin
      /*  move into conv:getoffwash3
        travel to conv:getonren3*/
    end
    send to Ptype
end

begin Ptype arriving
    if load type=Lbig   then
    begin
        set Atype to 1
        send to oneof(0.10:Pdelet,0.9:Pdup)
    end
    else if load type=Lsmall  then
```

```
      begin
         set Atype to 2
         send to oneof(0.30:Pdelet,0.70:Pdup)
      end
      else if load type=Lunimog then
      begin
         set Atype to 3
         send to oneof(0.45:Pdelet,0.55:Pdup)
      end
end

begin Pdelet arriving
      if load type=Lbig    then inc Vbigdel by 1
      else if load type=Lsmall  then inc Vsmalldel by 1
      else if load type=Lunimog then inc Vunimogdel by 1
    /*if load type=Lbig    then  print"Deleted  Type "Atype , Vbigdel to message
      else if load type=Lsmall  then print"Deleted  Type "Atype , Vsmalldel to
message
      else if load type=Lunimog then print"Deleted  Type "Atype , Vunimogdel to
message
    */send to Pdup1
end

begin Pdup arriving
      if load type=Lbig  then clone 1 loads to Pelect new load type L1elect
      else if load type=Lsmall  then clone 1 loads to Pelect nlt L2elect
      else if load type=Lunimog then clone 1 loads to Pelect nlt L3elect
      if load type=Lbig  then clone 1 loads to Pcarb new load type L1carb
      else if load type=Lsmall  then clone 1 loads to Pcarb nlt L2carb
      else if load type=Lunimog then clone 1 loads to Pcarb nlt L3carb
      send to P1
end

begin Pdup1 arriving
      if load type=Lbig  then clone 1 loads to Pelect new load type L1elect
      else if load type=Lsmall  then clone 1 loads to Pelect nlt L2elect
      else if load type=Lunimog then clone 1 loads to Pelect nlt L3elect
      if load type=Lbig  then clone 1 loads to Pcarb new load type L1carb
      else if load type=Lsmall  then clone 1 loads to Pcarb nlt L2carb
      else if load type=Lunimog then clone 1 loads to Pcarb nlt L3carb
      send to die
end

begin Pelect arriving
```

```
    if load type=L1elect then
    send to Pelect1

    else if load type=L2elect then
    begin
       send oneof (15:die,85:Pelect1)
    end

    else if load type=L3elect then
    begin
       send oneof (20:die,80:Pelect1)
    end

end

begin Pelect1 arriving

    if load type=L1elect then
    begin
       move into Q1elect
       use Relect for are1 min
    end

    else if load type=L2elect then
    begin
       move into Q2elect
       use Relect for are2 min
    end

    else if load type=L3elect then
    begin
       move into Q3elect
       use R3elect for are3 min
    end

    if load type=L1elect then inc Velectbig by 1
    else if load type=L2elect  then inc Velectsmall by 1
    else if load type=L3elect then inc Velectunimog by 1

    send to die
end

begin Pcarb arriving

    if load type=L1carb then
```

```
        send to Pcarb1
    else if load type=L2carb then
    begin
        send oneof (14:die,86:Pcarb1)
    end

    else if load type=L3carb then
    begin
        send oneof (18:die,78:Pcarb1)
    end

end

begin Pcarb1 arriving

    if load type=L1carb then
    begin
        move into Q1carb
        use Rcarb for  arc1 min
    end

    else if load type=L2carb then
    begin
        move into Q2carb
        use Rcarb for arc2 min
    end

    else if load type=L3carb then
    begin
        move into Q3carb
        use R3carb for arc3 min
    end

    if load type=L1carb then inc Vcarbbig by 1
    else if load type=L2carb  then inc Vcarbsmall by 1
    else if load type=L3carb then inc Vcarbunimog by 1

    send to die
end



begin P1 arriving

    if load type=Lbreak then
```

```
    begin
        increment Vbreak by 1
        print"Total breakdown" , Vbreak to message
        set Abreak to continuous (.30:1,.55:2,.80:3,1:4)
        if Abreak=1 then
        begin
            set Rblock capacity to 8
            wait for triangular 60,240,800 min
            print"Breakdown at Rblock repaired"  to message
            set Rblock capacity to 11
        end

        else if Abreak=2 then
        begin
            set Rcrank capacity to 6
            wait for triangular 60,240,800 min
            print"Breakdown at Rcrank repaired"  to message
            set Rcrank capacity to 8
        end

        else if Abreak=3 then
        begin
            set Rbed capacity to 5
            wait for triangular 60,240,800 min
            print"Breakdown at Rbed repaired "  to message
            set Rbed capacity to 7
        end

        else if Abreak=4 then
        begin
            set Rpiston capacity to 6
            wait for triangular 60,240,800 min
            print"Breakdown at Rpiston repaired"  to message
            set Rpiston capacity to 8
        end

        send to die
    end

else if load type=Lbig then

 begin
    increment Vbigblock by 1

    move into Qblock
```

```
    set Vb to 1
    if Vb<>Vbf then wait for 90 min
    use Rblock   for  ab min
    set Vbf to 1
    decrement Vbigblock by 1

    move into Qcrank
    set Vc to 1
    if Vc<>Vcf then wait for 90 min
    use Rcrank   for  acr min
    set Vcf to 1
    move into Qbed
    set Vbe to 1
    if Vbe<>Vbef then wait for 90 min
    use Rbed     for  abe min
    set Vbef to 1
    move into Qpiston
    set Vp to 1
    if Vp<>Vpf then wait for 90 min
    use Rpiston  for  ap min
    set Vp to 1
    move into Q1repair
    use R1repair for  a1r min
    move into Q2repair
    use R2repair for  a2r min
    move into Qmounting
  if Vbig >= Velectbig then
    begin
        wait until Vbig <= Velectbig
    end

    else if Vbig >= Vcarbbig then
    begin
       wait until Vbig <= Vcarbbig
    end
    use Rmounting for am min

end

else if load type=Lsmall then


begin
   increment Vs by 1
```

```
        move into Qblock
        set Vb to 2
        if Vb<>Vbf then wait for 90 min
        use Rblock for asb  min
        set Vb to 2
        decrement Vs by 1
        increment Vsc by 1
        move into Qcrank
        set Vc to 2
        if Vc<>Vcf then wait for 90 min
        use Rcrank for asc min
        set Vcf to 2
        decrement Vsc by 1
        move into Qbed
        set Vbe to 2
        if Vbe<>Vbef then wait for 90 min
        use Rbed for asbe min
        set Vbef to 2
        move into Qpiston
        set Vp to 2
        if Vp<>Vpf then wait for 90 min
        use Rpiston for asp min
        set Vp to 2
        move into Q1repair
        use R1repair for as1r min
        move into Q2repair
        use R2repair for as2r min
        move into Qmounting
        if Vsmall >= Velectsmall then wait until Vbig <= Velectsmall
        else if Vsmall >= Vcarbsmall then wait until Vbig <= Vcarbsmall
        use Rmounting for asm min

end

else if load type=Lunimog then


begin
      increment Vu by 1
      increment Vuc by 1
      move into Qblock
      set Vb to 3
      if Vb<>Vbf then wait for 90 min
      decrement Vu by 1
      use Rblock for aub min
```

```
        set Vbf to 3
        move into Qcrank
        set Vc to 3
        if Vc<>Vcf then wait for 90 min
        use Rcrank for auc min
        set Vcf to 3
        move into Qbed
        set Vbe to 3
        if Vbe<>Vbef then wait for 90 min
        use Rbed for aube min
        set Vbef to 3
        move into Qpiston
        set Vp to 3
        if Vp<>Vpf then wait for 90 min
        use Rpiston for aup min
        set Vp to 3
        move into Q1repair
        use R1repair for au1r min
        move into Q2repair
        use R2repair for au2r min
        move into Qmounting
        if Vunimog >= Velectunimog then wait until Vunimog <= Velectunimog
        else if Vunimog >= Vcarbunimog then wait until Vunimog <=
Vcarbunimog

        use Rmounting for aum min

    end




    send to Pinc
end

begin Pinc arriving
    if load type=Lbig then inc Vbig by 1
    else if load type=Lsmall  then inc Vsmall by 1
    else if load type=Lunimog then inc Vunimog by 1
    if load type=Lbig then print"Renovated Type "Atype , Vbig to message
    else if load type=Lsmall then print"Renovated Type "Atype, Vsmall to
message
```

```
    else if load type=Lunimog then print"Renovated Type "Atype, Vunimog to
message
    send to Pbrem
end

begin Pbrem arriving
    move into Qbrem
    get Rbrem
    if load type=Lbig    then wait for  abr1 min
    else if load type=Lsmall  then wait for abr2 min
    else if load type=Lunimog then wait for abr3 min
    free Rbrem
    move into Qpack
    get Rpacking
    if load type=Lbig    then wait for ap1 min
    else if load type=Lsmall  then wait for  ap2 min
    else if load type=Lunimog then wait for  ap3 min
    free Rpacking
    send to die
end
```

3. Code For Increased Ready Spare Parts Usage.

```
begin Pselection arriving

    set acont to triangular 45,60,75
    set aadb to triangular 55,60,70
    if load type=Lbig then
    begin

    set adis1 to triangular 55,90,100
    set aw1  to triangular 54,90,100
    set are1 to triangular 180,300,330
    set arc1 to triangular 171,285,313
    set ad1  to triangular 50,120,150
    set ab   to triangular 61,101,110  /*45 min decrease semi-processed spare part
*/
    set acr   to triangular 90,150,180
    set abe  to triangular 79,133,146
    set ap   to triangular 93,156,171
    set a1r  to triangular 60,102,112
    set a2r  to triangular 216,361,397
    set am   to triangular 63,106,116
```

```
    set abr1 to triangular 102,171,188
    set ap1  to triangular 51,85,93
    end
      else if load type=Lsmall then
    begin
    set ad2  to triangular 60,125,160
    set aw2  to triangular 48,80,88
    set are2 to triangular 180,300,330
    set arc2 to triangular 120,200,220
    set asb  to triangular 60,100,110  /*45min decrease semi-processed spare part
*/
    set asc  to triangular 42,70,77      /* 40min decrease semi-processed spare part
*/
    set asbe to triangular 25,43,48
    set asp  to triangular 45,75,83
    set as1r to triangular 63,108,118
    set as2r to triangular 64,106,116
    set asm  to triangular 48,80,88
    set ap2  to triangular 43,72,79
    set abr2 to triangular 92,154,167

    end
    else if load type=Lunimog then
    begin
    set adis3 to triangular 54,90,99
    set aw3  to triangular 54,90,99
    set are3 to triangular 137,229,252
    set arc3 to triangular 160,267,293
    set aub  to triangular 93,133,146
    set auc  to triangular 48,68,74
    set aube to triangular 70,100,110
    set aup  to triangular 77,110,122
    set au1r to triangular 186,266,292
    set au2r to triangular 145,208,228
    set aum  to triangular 110,158,173
    set au   to triangular 936,1560,1716
    set abr3 to triangular 105,175,192
    set ap3  to triangular 54,90,99
    end

    send to Pfirst
end
```

Appendix D

1. Costs of proposed system implementations

In this section, we investigated the implementation and process costs of the proposed systems 1 and 2 in *Table D-1 and D-2*. These informations are obtained from the technical staff in the renovation unit.

Pre-control and Repair Section	
Implemantation Cost	
Control Section	6500000000. TL
Repair Section	3500000000. TL
Test Section	6500000000. TL
Total	16500000000.TL
Processing Cost (per year)	
4 Worker (for a year)	12000000000.TL
Spare parts needs and general expenses	24000000000. TL
Total	36000000000. TL
Total cost	52500000000. TL
The effects on the Large Motors	3500000000. TL
The effects on the Small Motors	4900000000. TL

Table D-1. Cost of pre-control and repair section implementation

Combined Parallel Resources System	
Implemantation Cost	
5 days are used to convert existing system. This cost can be ignored according to technical staff.	Averagely 8 large motors, 10 small motors, and 6 unimog motors can not renovated.
Processing Cost (per year)	
Spare parts needs and general expenses	25000000000. TL for large motors
	12000000000. TL for small motors
	30000000000. TL for unimog motors
Total Cost Increase	57000000000. TL

Table D-2. Cost of combined parallel resources system implementation