مركز المنظومات الإلكترونية والبرمجيات
ELECTRONIC Systems AND programming center

Refereed Journal
www.epc.ly

Journal of Electronic Systems and Programming

Issue N0:4 December 2021

Journal of Electronic Systems and Programming

Electronic Systems and Programming Center

Journal of Electronic Systems and Programming

Issue No: 4 - 2021

Editorial:
Fourth Issue – Journal of Electronic Systems and Programming

We are delighted to announce the publication of the fourth issue of the Journal of Electronic Systems and Programming (JESP).

First of all, we would like to welcome the new Editorial Board members and we want to take this opportunity to express our gratitude to the outgoing members of the Editorial Board for their important contributions to the JESP journal over many years, and have worked hard to achieve our goals.

The big challenge to any journal is to provide facilitating peer-review process, and finding willing, expert, and independent reviewers across a wide range of research areas. To help ensure that JESP can continue to meet this challenge, a new Editorial Advisory Board has been appointed to ensure that manuscripts are reviewed to a high standard.

Finally, we thank our reviewers and authors for their fundamental contribution to the fourth release of the Journal. We still hope authors could consider JESP to be a place where to publish their work.

Editorial Board

Table of Contents:

1

INVESTIGATING THE EFFECTS OF WHEEL/RAIL SURFACE ROUGHNESS ON WHEEL/RAIL WEAR

Investigating the effects of wheel/rail surface roughness on wheel/rail wear

A. Shebani
College of Computer Technology – Zawya
amershebani@gmail.com

Abstract

Wear can be defined as the removal of material from solid surfaces by mechanical action. Wheel/rail wear is an everyday experience, and has been observed and studied for a very long time. Many railway accidents have occurred that were related to poor maintenance and wheel/rail wear. Therefore, there is a need to study the wheel and rail wear, so as to make informed decisions on wheel and rail maintenance, and reduce maintenance costs. This work focus on investigate the relationship between the wheel/rail surface roughness and wheel/rail wear. To study the relationship between the wheel/rail roughness parameters and the wheel/rail wear, a twin disc rig test and Alicona profilometer were used. This study can use to eliminate the wheel/rail wear and improve the design of the wheel and rail. This paper presented an Alicona profilometer and replica material as an effective tool to study the wheel/rail surface texture.

Keywords: wheel wear, rail wear, roughness, Alicona profilometer, replica material, twin disc rig test.

1. Introduction

1.1 Surface texture (Roughness and waviness)

Surface texture is an important issue when the main interest is to understand the nature of material surfaces and it plays an important role in the functional performance of many engineering components [1]. When two nominally rough surfaces are placed in contact, surface roughness causes contact to occur at discrete contact spots (junctions) such as in Figure:. (1). Then sum of the areas of all the contact spots constitutes the real (true) area of contact or simply contact area, and for most materials with applied load, this will be only a small fraction of the apparent (nominal) area of contact (that which would occur if the surfaces were perfectly smooth). The real area of contact is a function of the surface texture, material properties and interfacial loading conditions. The proximity of the asperities results in adhesive contacts caused by interatomic interactions. During the contact of two surfaces, contact will initially occur at only a few points to support the normal load (force). As the normal load is increased, the surfaces move closer together, a large number of higher asperities on the two surfaces come into contact, and existing contacts grow to support the increasing load. Deformation occurs in the region of the contact spots, establishing stresses that oppose the applied load. The mode of the surface deformation may be elastic, or plastic, and depends on nominal normal shear stresses (apparent contact area), surface roughness, and material properties [2].

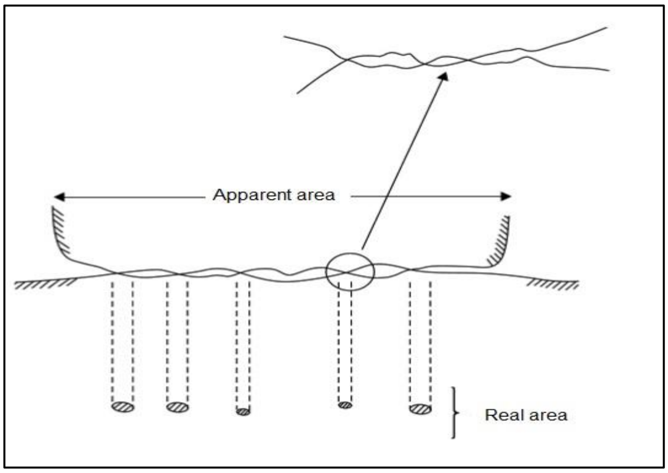


Figure: 1 Apparent area and real area [2]

All surfaces are rough on a microscopic scale, and when the two rough surfaces are in contact the real area is very small compared to the apparent area of the contact. When loading presses two rough surfaces together, only some peaks of the surfaces will be in contact such as in Figure 2 [3].

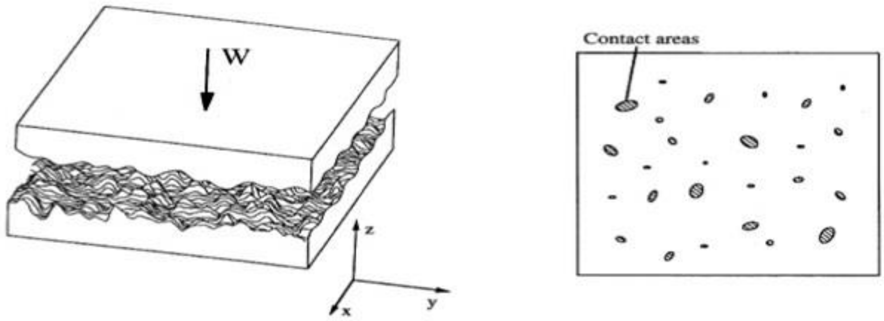


Figure: 2 Schematic of the real area of contact [3]

1.2 Analysis of surface roughness

Surface texture is the repetitive or random deviation from the nominal surface that forms the three-dimensional topography of the surface. Surface texture includes: roughness (nanoroughness), waviness (macroroughness), lay, and flaws. Nano- and macroroughness are formed by fluctuations in the surface of short wavelengths, characterized by hills (asperities) and valleys of varying amplitudes and spacing, and these are large compared to molecular dimensions. Asperities are referred to as peaks in a profile (two dimensions) and summits in a surface map (three dimensions). Figure 3 shows the surface roughness and waviness [4].

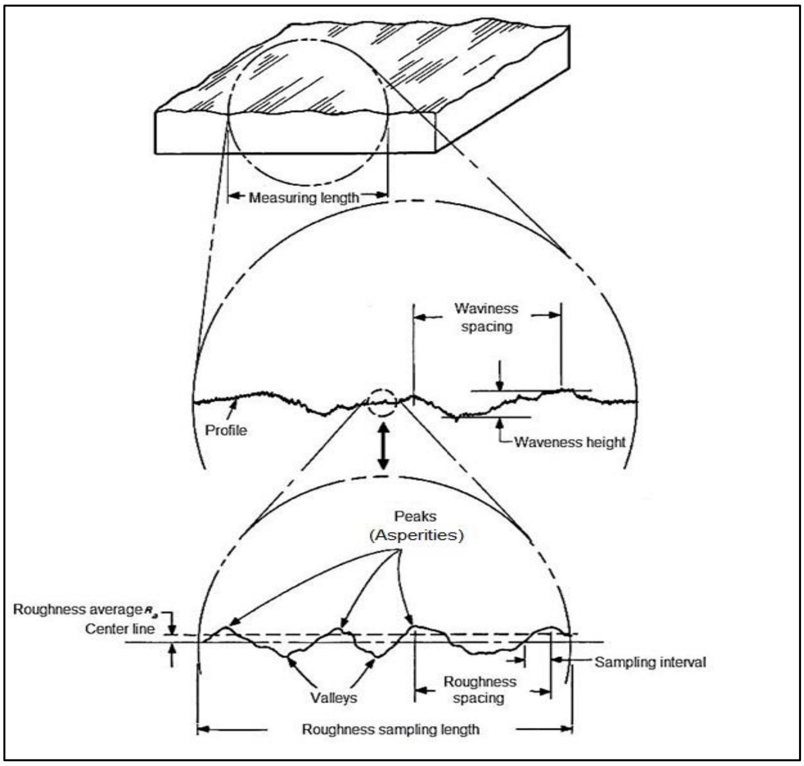


Figure: 3 Surface roughness and waviness [4]

Figure 4 shows the original profile, waviness profile, and roughness profile

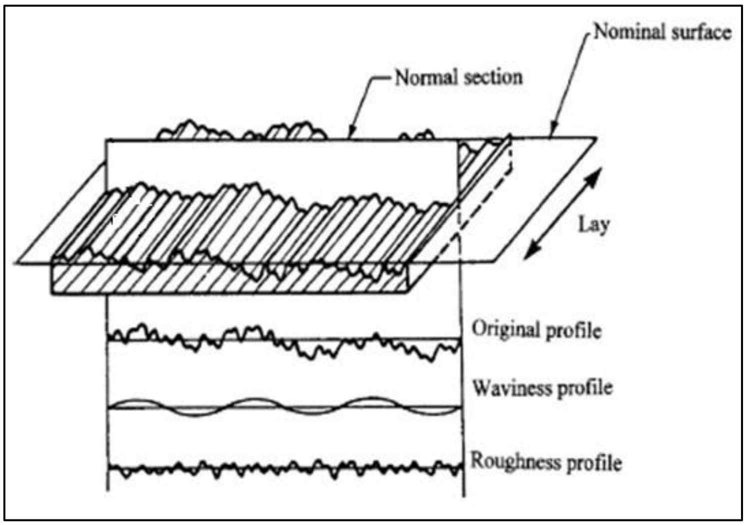


Figure: 4 Original profile, waviness profile, and roughness profile [5]

1.3 Basic definitions in roughness surface

- Surface: The surface of an object is the boundary which separates that object from another substance. Its shape and extent are usually defined by a drawing or descriptive specifications [6].
- Profile, it can be defined It is the contour of any specified section through a surface [6].
- Nominal surface, it is the intended surface. The shape and extent of a nominal surface are usually shown and dimensioned on a drawing. The nominal surface does not include intended surface roughness [7].
- A real surface is the actual boundary of an object. It deviates from the nominal surface as a result of the process that created

the surface. The deviation also depends on the properties, composition, and structure of the material the object is made of [7].

- Measured profile can be defined as the profile obtained with some measuring profilometer [8].
- Nominal profile is the straight or the smoothly line of intersection of the nominal surface with a plan which perpendicular to the surface [8].
- Primary profile is the sum of all the deviations of the measured profile from the nominal profile [8].
- Waviness profile is a recurrent deviation from a flat surface, much like waves on the surface of water [6].
- Roughness profile is defined as closely spaced, irregular deviations on a scale smaller than that of waviness. Roughness is expressed in terms of its height, its width, and its distance on the surface along which it is measured [6].
- Lay, it is referring to the predominant direction of the surface texture. Ordinarily lay is determined by the particular production method and geometry used [7].

1.4 Profile Filter

Filters play a fundamental role in the analysis of surface texture measurement. It is the process of extracting, or suppressing, certain wavelengths or spatial frequencies in the total profile, for example the attempts to reduce the effects of instrument noise and imperfections. The Profile filter is the filter that separates profiles into longwave and shortwave components. There are three filters used in instruments for measuring surface roughness, waviness and primary profiles. They all possess the same transmission characteristics, but have differing cut-off wavelengths such as in Figure 5 [9].

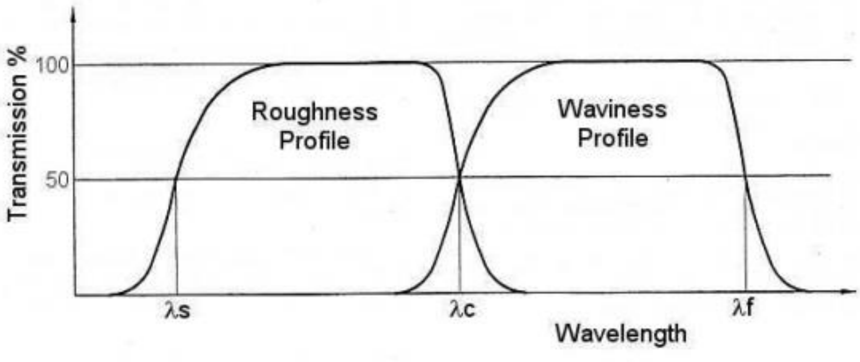


Figure: 5 Transmission characteristic of roughness and waviness profiles [9]

λs Profile Filter This is the filter that defines the intersection between the roughness and shorter wave components, such as instrument noise, present in a surface [9].
λc Profile Filter This is the filter that defines the intersection between the roughness and waviness components [9].
λf Profile Filter This is the filter that defines the intersection between the waviness and longer wave components, such as form, present in a surface [9].

1.5 Roughness parameters

Roughness average (R_a), it called centre line average value or arithmetic average. Among Height Parameters, the roughness average (Ra) is the most widely used because it is a simple parameter to obtain when compared to others. The roughness average is described as follows [1]:

$$R_a = \frac{1}{L} \int_0^L |Z(x)| \tag{33}$$

Where $Z(x)$ is the function that describes the surface profile analyzed in terms of height (Z) and position (x) of the sample over the evaluation length "L" Figure 6.

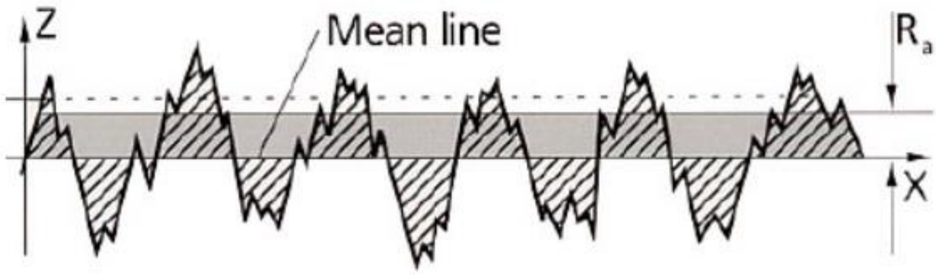


Figure: 6 Profile of a surface (Z). It represents the average roughness Ra [1]

2. Surface Measuring

2.1 Introduction

It is essential to consistently measure the surface roughness of work pieces as broad definitions of 'smooth surface' or 'rough surface' are inadequate. The ideal method by which to characterize surface roughness is through direct measurement with an appropriate tool. In this context surface roughness is usually quantified by the vertical deviations of a surface from its mean profile. Historically, the stylus instrument was the most widely accepted method to measure surface roughness as it gave more accurate results compared to other methods. Presently, many surface measuring instruments are used in manufacturing. For example, stylus instruments are widely used in the automotive industry, while optical instruments are often used in manufacturing where a non-contact method is required. Optical instruments do not make contact with surfaces and thus do not leave any trace or damage. Surface measuring instruments currently used in manufacturing can be sorted into two groups as follows [10].

2.2 Non-Contact approach

Optical profilometer measures surface roughness by using the laws of reflection as shown in Figure:. (7). Instruments that use this technique make use of focus detection, interferometry, and projected light. The optical interference method is suitable for measuring roughness of soft materials which are easily damaged. It is widely used in manufacturing of lenses and hard disk equipment [10].

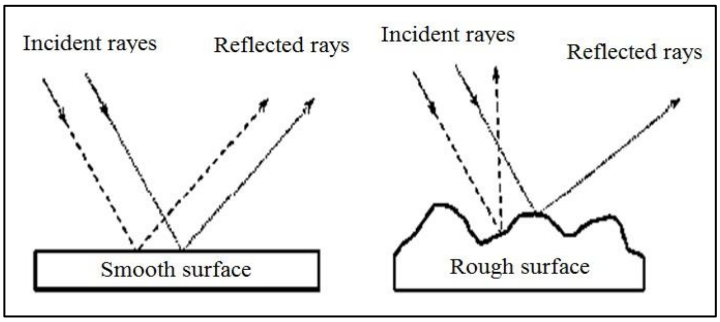


Figure: 7 Representative scheme for reflecting of light with smooth and rough surfaces [10]

Figure 8 shows the schematic of the optical surface profile. The main idea of the optical interference instrument is to identify the wavelengths of the light beams that are reflected from the test material and the reference mirror. Both wavelengths of the light beam will be used to calculate the height differences over a surface profile [10].

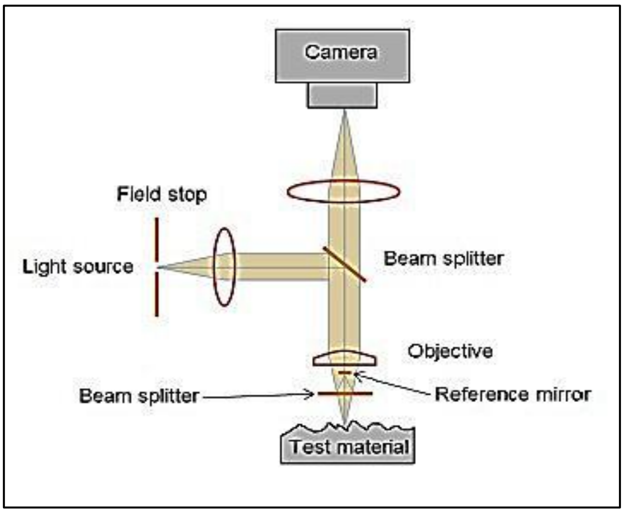


Figure: 8 Schematic of the optical surface profile [10]

2.3 Contact approach

Surface profilometer, also known as the 'Stylus method', is operated by moving a stylus along a surface. The stylus moves up and down following the peaks or valleys in the surface, and records this in terms of Cartesian coordinates as shown in Figure:. (9) [10].

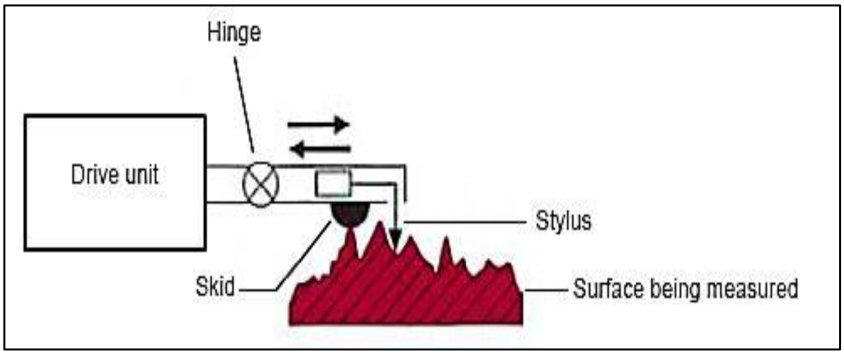


Figure: 9 Schematic of the working principle of a stylus profilometer [10]

The stylus technique which is the most commonly used class of surface texture measuremens instrument. Figure:ure:. (10) shows a schematic diagram of a stylus instrument. The stylus traverses the surface peaks and valleys, and the vertical motion of the stlus is converted by the transducer into an electrical signal which may be analyzed by digital or analog techniques. In many kinds of modern instruments, the signal undergoes analog-to-digital conversion. The resulting digital profile is stored in a computer and canbe anlyzed subseqeuntly for roughness parameters [11].

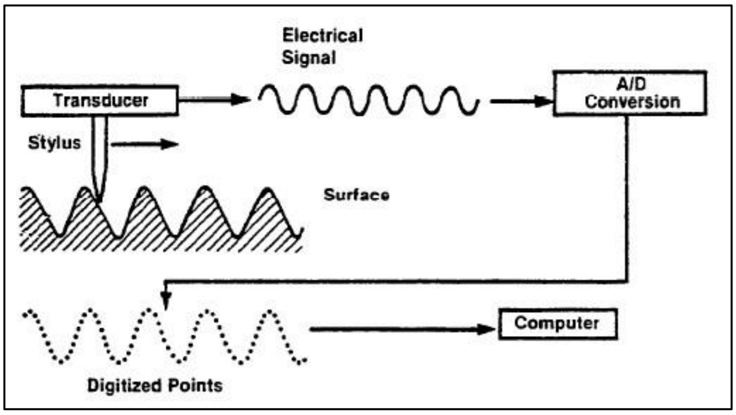


Figure: 10 Schematic diagram of stylus instrument [11]

Figure 11 shows a schematic diagram of a stylus transducer, this transducer is called a linear variable differential transformer (LVDT). The stylus is fastened to a ferrite core, which is positioned between two coils that form part of an AC bridge. As ferrite core moves up and down between the coils, the balance of the bridge is changed. The resulting output, after suitable demodulation and amplification, is a voltage signal that is proportional to the displacement of the stylus. The output of the LVDT, therefore, is directly proportional to surface height and hence yields the profile as the surface is scanned [11].

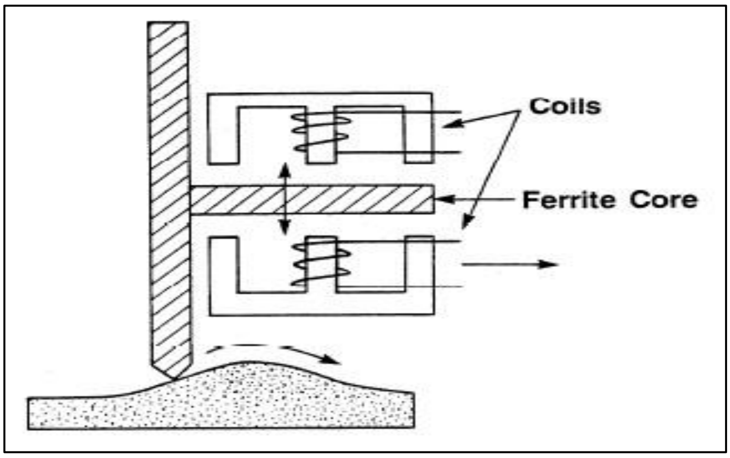


Figure: 11 Schematic diagram of LVDT transducer [11]

3. Wheel and rail wear

Due to the importance of the wheel and rail wear issue which is related to safety and economy, many experimental tests were carried out in this work using the twin disc rig to investigate the effects of some important parameters such as load, yaw angle, and speed on the wheel/rail wear under dry conditions. Figure: (12) shows the wheel/rail wear positions which is needs to be measured in this project (wheel wear and rail wear).

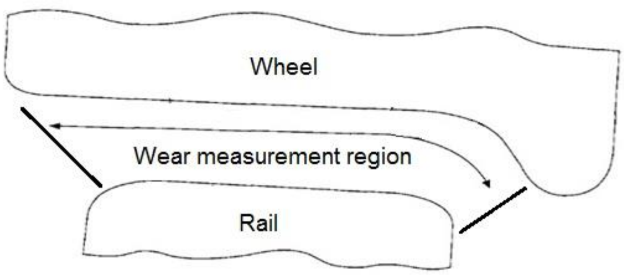


Figure: 12 Wheel and rail wear

The twin-disc system is simple and efficient; it consists of the use of two rollers pressed into contact, the variation of the relative velocity and of the contact pressure allows performing of the test under different conditions [12]. The twin disc approach has been shown to provide a good approach for comparing adhesion levels under a range of wheel/rail contact conditions, with and without contaminants [13]. Twin disc wear testing, used extensively for studying wear of wheel and rail materials [14].

The twin disc rig used in this work consists of an upper steel wheel of 310mm diameter and a lower steel wheel with diameter of 290mm. The rollers and shafts are made of EN24T steel. Vertical force of up to 4KN can be applied on the rollers through a jacking mechanism. The rig consists of a rotary table to allow a relative yaw angle between the rollers; this yaw angle is indicated by markings on the handle of the rotary table. In this work the replica technique used for wear and roughness measurements of the twin disc rollers, where the replica used to make a copy of the wheel/rail surfaces of two rollers before the test and after each test, and then the Alicona profilometer used to measure the wheel/rail wear. The name of the replica material which is used in this work is AccuTrans [15].

The Alicona profilometer generates the wheel/rail roughness profiles using different filters such as in a standard ISO 2478 [16].

The Alicona profilometer (INFINTE FOCUS G4) which is shown in Figure: (13) was used in this work for pin wear and disc roughness and wear measurements.

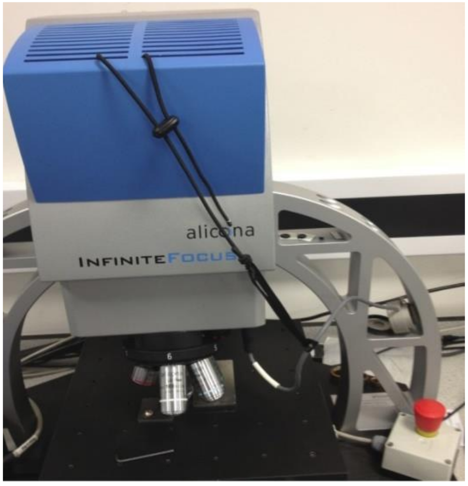


Figure: 13 Alicona (INFINTE FOCUS G4) [15]

Figure 14 shows a sample of the rail wear measurement using the Alicona profilometer which was carried out in this paper using the twin disc rig.

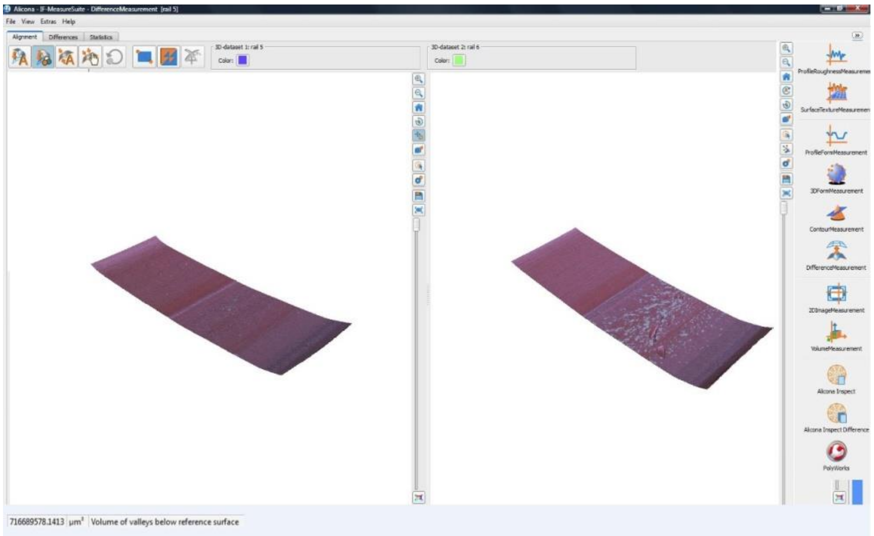


Figure: 14 Rail sample measurements using Alicona profilometer

Figure 15 shows a sample of replica surface for wheel under the lens of the Alicona profilometer. The sample dimensions were 5mm width and 35mm length; the roughness profile was shown in this Figure.

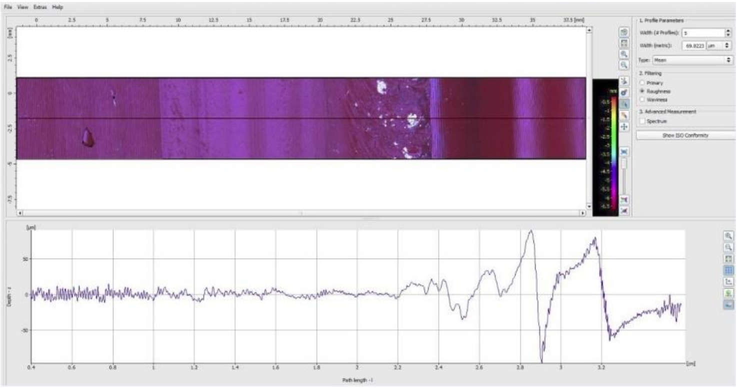


Figure: 15 Sample of replica of wheel profile under the Alicona profilometer

Figure 16 shows a sample of replica surface for rail under the lens of the Alicona profilometer. The sample dimensions were 5mm width and 20mm length; the roughness profile was shown in this Figure.

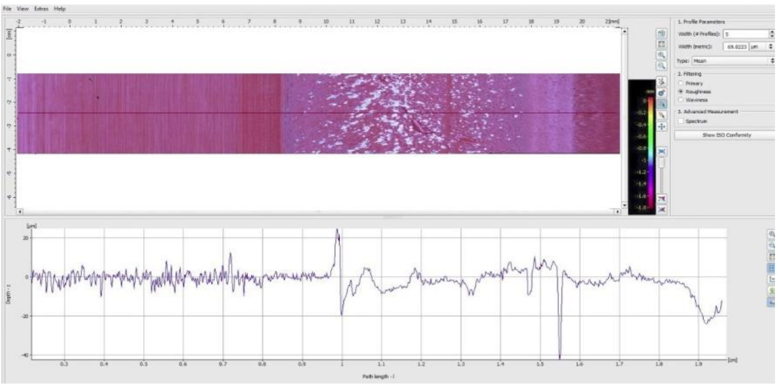


Figure: 16 Sample of replica of rail profile under the Alicona profilometer

4. Wheel/rail wear, and wheel/rail roughness measurements

In this work, several tests for wheel/rail wear, and wheel/rail roughness measurement using the twin disc rig, and Alicona profilometer were carried out under several conditions such as:

Effect of load on wheel/rail wear, and wheel/rail roughness over fixed time; such as in Table (1); under dry condition. (Speed =960rpm, test time = 60 min, and yaw angle = 0.4°). Table (1) shows the arithmetic average roughness and wheel/rail wear for wheel and rail after applying different values of load. The arithmetic average roughness and wear were measured using the Alicona profilometer.

Table 1 Effect of load on wheel/rail wear and Ra

No	Load (N)	Wheel wear (mm^3)	Rail wear (mm^3)	R_a for v (μm)	R_a for r (μm)
1	1000	1.1886	0.7166	3.5371	2.1232
2	1400	2.1225	1.3221	4.0957	3.8251
3	1800	3.6387	2.3639	4.3696	4.2429
4	2200	4.3870	3.4815	4.8643	4.6598
5	2600	5.8493	4.8744	5.1319	5.0726
6	3000	6.6227	5.3814	5.2357	5.1158
7	3400	8.9447	7.5931	5.6357	5.4151

Figure 17 shows the variation of arithmetic average roughness (R_a) with different values of load for wheel/rail surfaces.

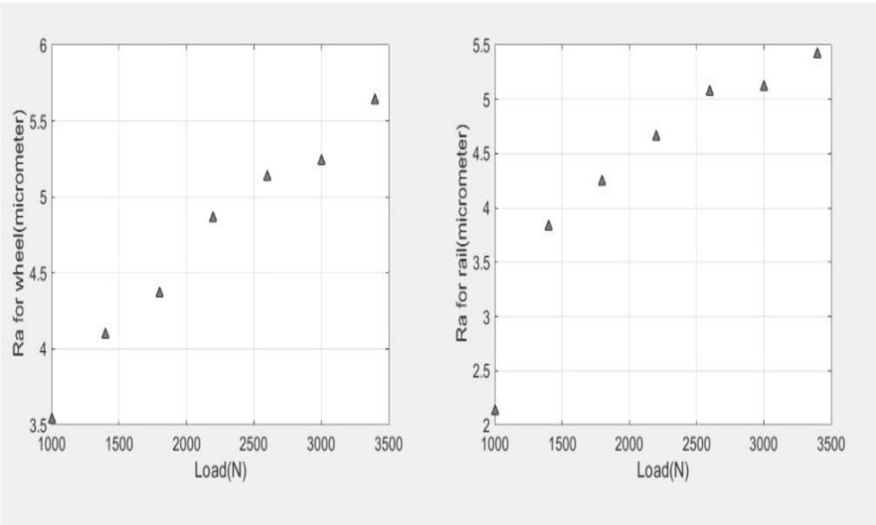


Figure: 17 Variation of arithmetic average roughness (R_a) with different values of load.

Effect of yaw angle on wheel/rail wear, and wheel/rail roughness over fixed time; such as in Table (2); under dry condition. (Speed =960rpm, load 2200N, and test time = 60 min). Table (2) shows the arithmetic average roughness and wheel/rail wear for wheel and rail after applying different values of yaw angle. The arithmetic average roughness and wear were measured using the Alicona profilometer.

Table 2 Effect of yaw angle on wheel/rail wear and Ra

No	Yaw angle (degree)	Wheel wear (mm^3)	Rail wear (mm^3)	R_a for w (μm)	R_a for r (μm)
1	0.1	1.2351	1.1721	2.6459	2.3576
2	0.2	1.6765	1.6265	3.1597	2.8613
3	0.3	2.1225	3.0609	3.2085	3.4548
4	0.4	3.1702	3.6511	3.9405	3.6072
5	0.5	5.8254	4.6919	5.4428	4.1522
6	0.6	7.4042	7.216	6.9407	5.8913
7	0.7	9.2830	8.7792	7.4647	7.2941

Figure 18 shows the variation of arithmetic average roughness (R_a) with different values of yaw angle for wheel and rail surfaces.

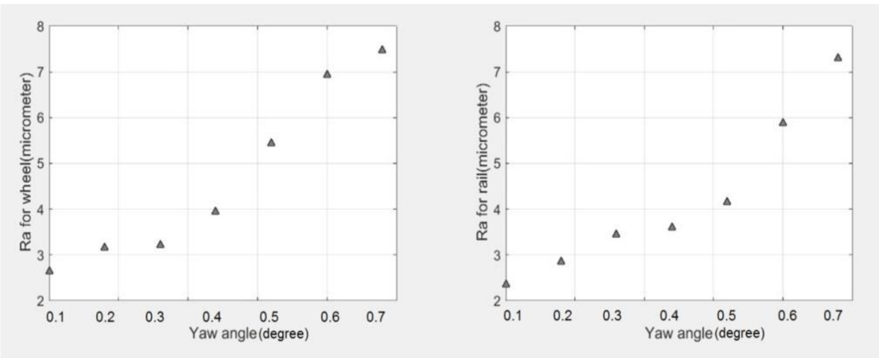


Figure: 18 Variation of arithmetic average roughness (R_a) with different values of yaw angle

Effect of speed on wheel/rail wear, and wheel/rail roughness over fixed time over fixed time under dry condition with different values of speed, such as in Table (3). (Load =1000N, and yaw angle = 0.1°, and test time = 10 min). Table (3) shows the arithmetic average roughness and wheel/rail wear for wheel and rail after applying different values of speed. The arithmetic average roughness and wear were measured using the Alicona profilometer.

Table 3 Effect of speed on wheel/rail wear and Ra

No	**Speed (rpm)**	**Wheel wear (mm^3)**	**Rail wear (mm^3)**	**R_a for** (μm)	**R_a for** (μm)
1	300	0.0823	0.0769	1.1062	1.0742
2	420	0.1997	0.1646	1.3783	1.2202
3	540	0.3668	0.3405	1.6430	1.4360
4	660	0.4536	0.4104	1.9292	1.8075

Figure 19 shows the variation of arithmetic average roughness (R_a) with different values of speed for wheel and rail surfaces.

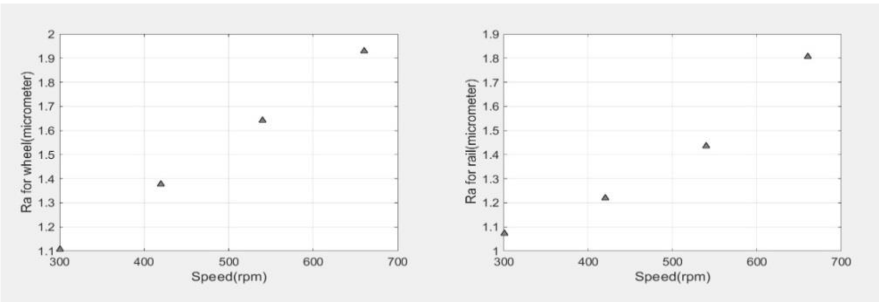


Figure: 19 Variation of arithmetic average roughness (R_a) with different values of speed

5. Conclusion

Several tests were carried out using the twin disc rig machine, and then, these measurements were used to establish how the relation between the wheel/rail surface roughness parameters and the wheel/rail wear. This work presented that the wheel/rail wear and roughness can be measured using and replica material and Alicona profilometer. For wheel/rail wear measurements, the replica material applied to the wheel/rail surfaces of the twin test rig to make a copy of both surfaces, then, the replica samples were scanned using an optical profilometer and the results were processed to establish wheel/rail wear and roughness parameter. The relation between wheel/rail wear and wheel/rail roughness parameter under different of load, yaw angle and speed were investigated. Test results show that the relation between the wheel/rail surface roughness parameter and wheel/rail wear, when the arithmetic average roughness R_a increased, as a result, the wheel/rail wear occurred faster in case of change of load, yaw angle, and speed. The arithmetic average roughness of wheel/rail, and the wheel/rail wear were bigger in case of change of yaw angle, it was bigger than the cases of change of load and speed, that because of the contact was closed to the flange rather than the tread in case of yaw angle tests.

References

[1] V. Bellitto, ATOMIC FORCE MICROSCOPY IMAGING, MEASURING AND MANIPULATING SURFACES AT THE ATOMIC SCALE. Croatia, 2012.

[2] B. Bhushan, Introduction to tribology: John Wiley & Sons, 2013.

[3] S. Tavares, "Analysis of surface roughness and models of mechanical contacts," University of Pisa, 2005.

[4] B. Bhushan, Modern Tribology Handbook, Two Volume Set: Crc Press, 2000.

[5] B. Muralikrishnan and J. Raja, Computational surface and roundness metrology: Springer Science & Business Media, 2008.

[6] U. Khandey, "Optimization of surface roughness, material removal rate and cutting tool flank wear in turning using extended taguchi approach," Master thesis, National institute of technology India, 2009.

[7] "Surface Metrology Guide - Surfaces and Profiles," http://www.htskorea.com/tech/spm/profile.pdf.2015

[8] Z. Dimkovski, "Characterization of a Cylinder Liner Surface by Roughness Parameters Analysis," Blekinge Institute of Technology, Karlskrona, Sweden, 2006.

[9] J. K. Brennan, "Algorithms for surface texture profiles and parameters," University of Huddersfield, 2010.

[10] S. Srirattayawong, "CFD study of surface roughness effects on the thermo-elastohydrodynamic lubrication line contact problem," Department of Engineering, 2014.

[11] a. J. R. T. V. Vorburger, "Surface Finish Metroology Tutorial," USA1990.

[12] N. Bosso and N. Zampieri, "Experimental and numerical simulation of wheel-rail adhesion and wear using a scaled roller rig and a real-time contact code," Shock and Vibration, vol. 2014, 2014.

[13] E. Gallardo-Hernandez and R. Lewis, "Twin disc assessment of wheel/rail adhesion," Wear, vol. 265, pp. 1309-1316, 2008.

[14] R. Lewis, R. Dwyer-Joyce, U. Olofsson, and R. Hallam, "Wheel material wear mechanisms and transitions," 2004.

[15] A. Shebani, "PREDICTION OF WHEEL AND RAIL WEAR USING ARTIFICIAL NEURAL NETWORKS," PhD, Computing and Engineering, University of Huddersfield UK, 2017.

[16] Alicona, Optical 3D micro coordiate measurement from & roughness vol. Version 3.9.1 EN 2011.

2

Develop Clamping and Drilling System Using PLC

Develop Clamping and Drilling System Using PLC

Esmail Mohammed
Department of Electrical and Electronics Engineering
Higher Institute of Science and Technology Zawia, Libya
ismaeber@yahoo.com

Salah Aboharba
Department of Electrical and Electronics Engineering
Higher Institute of Science and Technology Zawia, Libya
sabohrba@ymail.com

Abstract

Many industrial applications require that a hydraulic systems working automatically after an operator presses a START pushbutton. Such as cutting, bending, drilling, clamping and forming metals, that needs more force. Clamping and drilling system one of them, that need control for ability to clamp and drilling the workpiece with required force without damage or deforming the work piece. In this study, clamping and drilling system was developed and improvement by using PLC control system. The PLC is programed for control requirements to the hydraulic system of workpiece clamp and drilling. The Programming Software for GLOFA series GMWIN is selected and programming (LAD) software was used to program the system control process of the system. Using PLC control hydraulic system, can be easy to improve power the productivity greatly and increase the system stability, reliability, also security for factory workers.

Keywords: clamping and drilling system - hydraulic system- PLC control system- workpiece.

1. Introduction

Recently, the increasing use of electrically controlled hydraulic systems in industry comes from the need for fast, low-cost means of production with better quality, less waste, and more energy. Electrically controlled hydraulic systems offer many other advantages. A few of these are decrease fatigue, micro control and small size [1, 2]. Before the technical progress in the industrial field, all drilling operations are carried out related to the vibration which results in the noise environment as well as more fatigue of the workers [3].

The improved performance of the clamping and drilling machine must be developed to meet the demands of the times. In order to meet Manufacturing requirements for complex and precise products, by using PLC control the clamping and drilling machine is developed into Precision machine for clamping drilling machine.

Fixtures can be enhancing the production quality if they are properly designed. Fixture serves both purposes Fixtures can be enhance the production quality if they are properly designed. Fixture serves both purposes. Work clamping and drilling system consists of work piece detection, clamping and unclamping, drilling and undrilling. Work piece detection is achieved by using proximity sensors. Many industrial processes involve some type of system or process of automation. Milling, folding, tensioning and grinding are just a few of these processes. Operations are often controlled by some type of sequential circuit so that the operation of the machines occurs at a specific moment during the work cycle. The clamp cylinder should always be turned on first [4,5].

The drilling process is a common automated machining process. Many machining functions are accomplished with a drill connected to a hydraulic cylinder. Drilling the hole, tapping and threading the hole, and countersinking the top edge of the hole are all mechanical jobs that call for drilling [6].

2. Clamping and Drilling System

Clamping and drilling system consist of hydraulic part and electric control part, as shown in figure1.

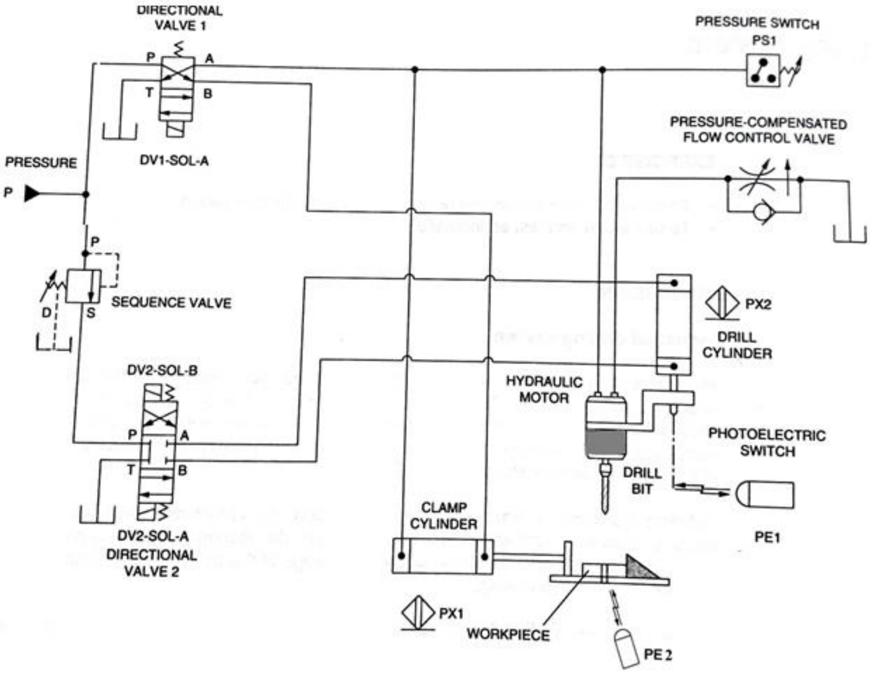


Figure 1: Clamping and Drilling System

Figure 1 shows a typical drilling system that includes a clamp cylinder and a drilling cylinder that lowers and raises a hydraulic drill. The sequence of operation is as follows: directional valve 1 is shifted first to extend the clamp cylinder. When the workpiece is held in place, the hydraulic motor of the drilling unit begins to rotate. Then the gear valve 2 is shifted to extend the drilling cylinder, reducing the drilling unit. When the workpiece is drilled, the directional valve 2 is shifted to retract the drilling cylinder, which raises the drilling unit. When the drilling cylinder is retracted, the valve is returned, directional valve1 is returned to its normal condition to retract the clamp cylinder and stop the hydraulic motor of the drill unit.

The pressure- compensated flow control valve connected downstream of the hydraulic motor provides a constant motor speed by compensating for pressure changes upstream and downstream.

Electro control part that includes magnetic proximity sensors, pushbutton switches for start and reset, and Relays as actuators for solenoid valves as shown in the figure2.The magnetic proximity switches and diffuse Reflective Photoelectric Switch require a 24-V DC voltage to operate. Therefore, make sure to connect the plus and the negative terminals of these sensors or switches to the 24-V DC power supply. Magnetic proximity sensors so that it is activated when the bore cylinder rod is fully retracted. Leave the cylinder rod in the fully retracted position.

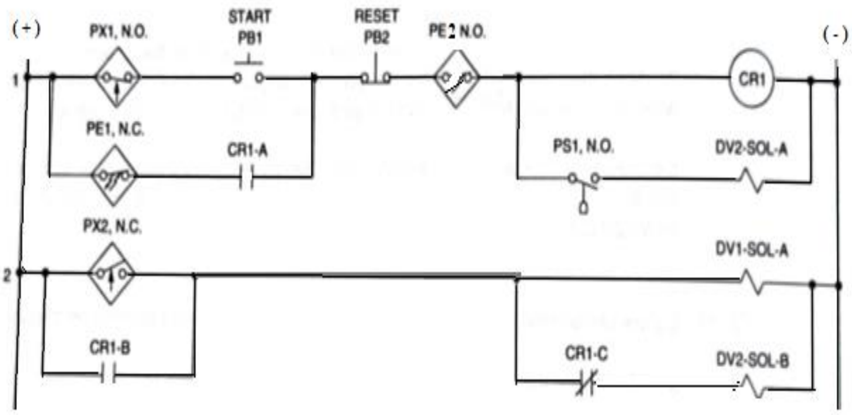


Figure 2: Electro control part

3. Programmable Logic Controller (PLC)

It is a digital logic device used to control for typical industrial processes, such as control of machinery on factory assembly lines, electro hydraulic [7, 8]. It is used in many industries applications. The interlocking between the limit switch or

a proximity sensors is given for continuous motion of the machine. PLC can be programing by many method such as LAD, STL, FBD, etc. The PLC hardware consists of input and output units, where the sensors and switches connected with PLC inputs, and the actuators connected with PLC outputs [9]. The microprocessor-based CPU is the one that controls the operations inside the PLC. The read input is read and its state is stored in the data memory, the data is transferred to the application program and processed, the updated data memory is updated and finally the executed of the output [10].

In this system the PLC is working as brain to control of this system, where the PLC is received the signal from a proximity sensors or limit switches, then send a signal or command to actuators or loads which connected by output units. By using ladder diagram (LAD), the electrical diagram as shown in figure 2 can be rewritten as ladder diagram. The (LAD) is one of methods of GLOFA series GMWIN PLC programming. Figure 3 shows the ladder diagram that control clamping and drilling system or machine. Table.1 shows the symbols used in the electrical diagram and scalar diagram.

Figure 3: ladder diagram (LAD)

Table-1: the symbols used in the electrical diagram and scalar diagram

Section	Indirect Variable	I/O Assignment	comment
Input	START	%IX1.0. 0	Push Switch-1
	RESET	%IX1.0.2	PushSwitch-2
	PX1.N.O	%IX1.0.5	Proximity Switch
	PE1.N.O	%IX1.0.4	Photoelectric Sensor-1
	PE2.N.O	%IX1.0.3	Photoelectric Sensor-2
	PS1.N.O	%IX1.0.6	Pressure Switch
Output	SV11	%Qx0.0.0	Solenoid Valve-A Extend Cy1
	SV21	%Qx0.0.2	Solenoid Valve-B Extend Cy2
	SV22	%Qx0.0.3	Solenoid Valve-B Retract Cy2
	MOTOR S	%Qx0.0.5	Motor Operation
	SV12	%Qx0.0.1	Solenoid Valve-A Retract Cy1

4. Clamping and drilling system operation and control

- Firstly, The workpiece to drill is positioned by hand in the drill place. When the operator presses a START pushbutton, solenoid A of directional valve 1 is activated, but if there is no the workpiece on drill place the system not operate. This shifts the valve to the straight-arrows condition to extend the clamp cylinder. Since the extending clamp cylinder requires a pressure, the hydraulic motor of the drill unit does not rotate now.

- When the clamp cylinder stress the workpiece, the pressure rises quickly behind its piston, the hydraulic motor of the drill unit starts rotating. Also, pressure switch PS1 is activated, indicating that the workpiece is correctly clamped. This activates solenoid A of directional valve 2 and shifts the valve to the straight-arrows condition to extend the drill cylinder. The sequence valve stays partially closed so that most of the pumped oil goes to the motor, while a small amount of oil goes through the sequence valve to extend the drill cylinder. The drill cylinder, will extends at slow speed, which is beneficial because the drill bit will not be damaged by impacting the workpiece.

- The drill cylinder extends and lowers the drill unit to drill a hole in the workpiece, When this cylinder becomes fully extended, it activates a photoelectric sensor or switch, PE1 This activates solenoid B of directional valve 2, shifting the valve to the crossed - arrows condition to retract the drill cylinder and raise the drill unit, During retraction of the drill cylinder, solenoid A of directional valve 1 is kept energized so the drill motor continues to turn and the clamp remains constant.

-When the drill cylinder becomes fully retracted, it will activates magnetic proximity switch PX2, which de-energizes solenoid B of directional valve 2 and returns the valve to the center condition.

This also de-energizes solenoid A of directional valve 1, that will causing the clamp cylinder to retract and the drill motor to stop rotating.

- When the clamping cylinder becomes fully retracted, it activates magnetic proximity switch PX1, which stops the cycle. The drilled workpiece can then be removed. Several refinements can be added to the drilling system to provide special control functions by reprograming PLC. They may include, for example, a reset pushbutton as a safety feature to permit the operator to stop the system at any sequential step and reset it back to the initial state where both cylinders are fully retracted and the hydraulic motor is stopped.

5. Conclusion

In this study, clamping and drilling system was developed and improvement by using PLC control system. The PLC is programed for control requirements to the hydraulic system of workpiece clamp and drilling. The Programming Software for GLOFA series GMWIN is selected and programming (LAD) software was used to program the system control process of the system. Using PLC control hydraulic system can be easy to improve power the productivity greatly and increase the system stability, reliability, also security for factory workers.

The PLC program can be modified to increase the speed of work and thus increase the production capacity easily without any problem.by using PLC in drilling machine leads to Optimization process and drilling is smooth and reduce noise when compared to other system or machine.

References

[1] Guanghuai Wang. Application of Hydraulic technology. Harbin Institute of technology Press; 2005

[2] Dey, A. (1995), "Drilling machine", Latest development of heavyearth moving

[3] Machinery, Annapurna Publishers,pp.120-228. Charbulova, M., Matusova and M., Caganova, D., "Intelligent Production Systems and Clamping Systems for Intelligent Production Systems", Journal.

[4] J. W. Luo, Hydraulic Drive and Control [M].Chongqing: Chongqing University Press. (2006. 7.)

[5] Minke, G, "Building with earth: design and technology of a sustainable architecture", Walter de Gruyter, (2013).

[6] Bassily, Hany, et al. "A Mechatronics Educational Laboratory – Programmable Logic Controllers and Material Handling Experiments." Mechatronics 17.9 (2007): 480-8.

[7] S. Ilango and V. Soundararajan, "Introduction to Hydraulics and pneumatics", PHI Learning Private Limited, New Delhi,2nd Edition, 2009.

[8] Zuperl, U., Cus, F., Vukelic, D., "Variable Clamping Force Control for an Intelligent Fixturing", Journal of Production Engineering, Vol. 14, 2009, pp. 19-22.

[9] Wong, Kiing Ing, and Teck Ung Siaw. "PLC and SCADA Laboratory Experiments for a Final Year Instrumentation Course." International Journal of Information and Education Technology, vol. 5.11 (2015)

[10] Charbulova, M., Matusova and M., Caganova, D.,“Intelligent Production Systems and Clamping Systems for Intelligent Production Systems”, Journal of Production Engineering, Vol. 14, 2009, pp. 63-66.

3

Exploring the Suitability of UML as a Business Process Modeling Tool

Exploring the Suitability of UML as a Business Process Modeling Tool

Adel Smeda, El-Bahlul Fgee, Mariam Mohamed

Department of Computer Science, School of Basic Sciences Libyan Academy, Janzor, Libya
adel.smeda,bahlul.fgee@academy.edu.ly

Abstract

In this paper, we explore the applicability of UML (Unified Modeling Language) as a Business Process Modeling tool (BPM). BPM is a critical component of any business improvement process; it allows capturing a broad outline and procedures that govern what it is a business does. This model provides an overview of where the proposed system will fit into the organizational structure and daily activities. Although there are quite a lot of modeling techniques, there is no a well-defined standard. Moreover, these techniques support wide range of definitions and notations. Meanwhile UML is based on well-defined meta model governed by an international research group (OMG: Object Modeling Group) and becomes a de-facto standard for software modeling. Its rich modeling tool, with different visual diagrams and notations, makes it a promising candidate for modeling any business process. A case study is presented to better understand how UML can be used as a BPM tool.

Keywords Business Process Modeling, Unified Modeling Language, Modeling, Meta Modeling.

1. Introduction

Business organizations worldwide work competitively, which leads to the need of improving their competitively. This involves increasing the quality of their products and services, and to reduce costs and improve profits. To gain competitively, organizations must improve their inner operations and this process needs a business model that represents the business composition, letting the organizations to accomplish the analysis and the simulation of work before its execution [1]. Previously, the business has been represented with hierarchical models of organizational structures. However, the business can be improved through the optimization of its business processes. A business process represents the organizations system of work, beside allowing a study of the workflow through their inside and outside limits [1]; Business processes are essential since they define the order of work, keep the competitive advantage, increase the quality and decrease maintenance costs [2]. Subsequently business processes can become multifaceted and hence hard to analyse, understand, or explain. Due to the fact that business processes tend to grow larger and more complex with age, then they should be managed properly. Therefore, models of processes are used to represent a business process [3]. Although there are quite a lot of modeling techniques and these techniques use different notations, but they are used to model similar processes. The focus was on visual notations, which have a readability advantage over non-visual notations. Visual notations make it possible for organizations to improve their business processes and communicate them with partners to simplify business to business transactions. Meanwhile, in software projects UML (Unified Modeling Language) is becoming a de-facto standard for modeling software systems with its well defined meta-model and its different verities of diagrams that represent different views of the system [13]. In this paper, we explore the applicability and suitability of UML as a Business Process Modeling (BPM) tool.

2. Business Process Modeling

To keep up and stay competitive, companies and organizations must evaluate the quality of their products and the efficiency of their services. They should consider their competitors, subcontractors, dealers, and clients as well the ever-changing laws and regulations. Besides, they must independently examine their products or services, by answering such inquiries as: Is my internal process running smoothly? If possible improve my product or service in different way? If the production processes are efficiently? Possibility of developing the product or service portfolios to get new markets and customers? [1]. Recently, business group must also evaluate their own information systems, if their way of business effectively supported, if the systems adapt to changing, if the used information are an significant resource in the business, if the information satisfactory and correct. All businesses will benefit by gaining a deeper understanding of how their business interacts with its environment, which comes from answering these questions. To answer these questions, it is necessary to create a model of the business, a simplified view of a complex reality (Figure 1). It means creating abstraction that enables you to eliminate irrelevant details and focus on one or more important aspects at a time. Effective models also facilitate discussions among different stakeholders in the business, helping them to reach agreement on the key fundamentals and to work toward common goals. Modeling is an accepted and established ways of analyzing and designing systems.

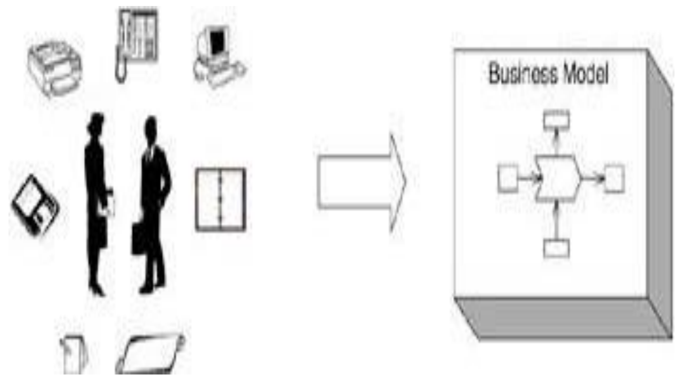


Figure 1: A business model is a simplified view of a business.

A business model is an abstraction of how a business functions. Its details differ according to the perspective of the person creating the model, each of whom will naturally have a slightly different viewpoint of the goals and visions of the business, including its efficiency and the various elements that are acting in concert within the business. This is normal, and the business model will not completely resolve these differences. What the business model will do is provide a simplified view of the business structure that will act as the basis for communication, improvements, or innovations, and define the information systems requirements that are necessary to support the business.

2.1 Business Concepts

A business, an enterprise, is a complex system that has a specific purpose or goal. All the functions of the business interact to achieve this goal. The business system in addition can be interlinked with and influenced by the decisions and events that occur in other systems, where can't be separately analyzed. This leads to difficult defining the boundaries of the business. The organization's resources can have separate goals that do not forever reflect those of the business [1]. The most important elements in a business, namely, customers, suppliers, laws, and regulations are outside of the business and are not defined within the business itself. Therefore, the business system is an open system whose objects and parts are often also parts of other business systems. As such, it cannot be viewed as a black box system, which is analyzed by looking only at the input to and output from the system, but as a system whose parts are visible, as shown in figure 2 [1].

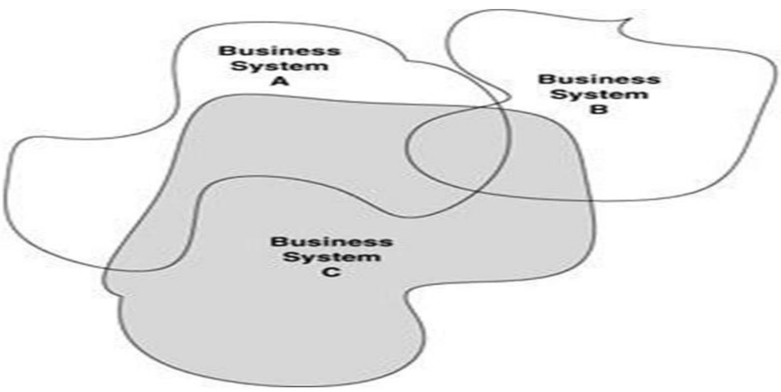


Figure 2: A business system is interlinked with other business systems.

2.2 Business Process

The business processes are the active part of the business. They describe the functions of the business, and involve resources that are used, transformed, or produced. It is an abstraction that shows the cooperation between resources and the transformation of resources in the business. It emphasizes how work is performed, rather than describing the products or services that result from the process. A more formal definition of a business process is a collection of activities that takes one or more kinds of input and creates an output that is of value to the customer. A business process has a goal and is affected by events occurring in the external world or in other processes. Figure 3 shows a general model of business processes [2].

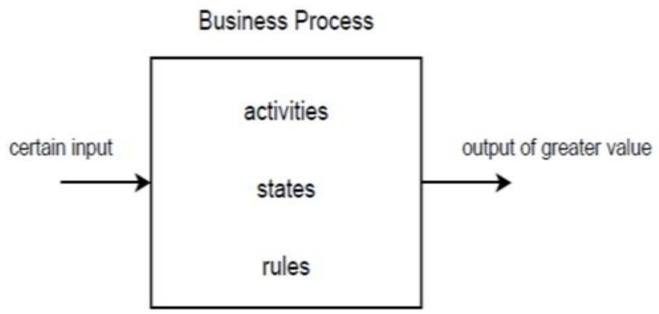


Figure 3: A general model of business processes.

2.3 Business Process Modeling Technique

Business process models are created based on a specific modeling technique. In practice several different techniques exist which are suitable for different business domains and purposes. For the creation of business process models, an appropriate technique must be chosen. Each technique consists of two major parts (figure 4), a modeling language (also called modeling grammar) and a modeling method. The modeling language can be further divided into notations (at least one), syntax, and semantics. The modeling notation defines graphical symbols that process modelers can use to model processes [4]. The syntax states rules for combining the symbols within business process models. For process modelers it is mandatory to stick to the rules specified by the syntax. The semantic description binds a meaning to each graphical symbol to clarify its specific use. The modeling method defines the procedures which can be applied to create a business process model. Following these procedures ensures that the resulting model is compliant to the modeling notations used. In practice, modeling tools play an important role as they are used to create, maintain, and apply business process models.

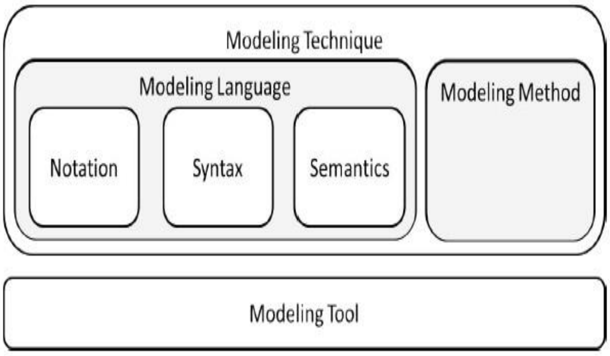


Figure 4: Components of a modeling technique.

2.4 Business Process Modeling Tool

Business process modeling plays an essential role in the business process management regulations. For the last twenty years several methods and tools have been proposed to model a business process [5, 8, 11]. In this section, we present a brief survey of most of these tools including flowcharts, Petri nets, Role Activity Diagrams, Event-driven Process Chain, Integrated Definition for Function Modeling, and Business Process Modeling Notations.

2.4.1 Flowcharts

A flowchart is a diagram that represents a process as a chain of activities and decisions. Flowcharts are the first and most fundamental process related to modeling methodology, their first use back to the early twenties, where they were used in by mechanical engineers to illustrate machine behavior. Basic flowchart entities are activities, decisions, start and end points.

2.4.2 Petri nets

A Petri net is a formal modeling language that describes the behavior of simultaneous processes. Carl Adam Petri presented the graphical notations of Petri nets in 1939 [6], initially they were used to describe chemical processes. Compared to other methodologies, Petri nets provide a well-defined mathematical, this makes that all well-formed Petri nets can be construed and executed by a machine. While most other methodologies focus on representing the structure of a process, Petri nets focus on the actual behavior of a process. Furthermore, the mathematical basis of Petri nets makes them appropriate for different kinds of automated analysis [6].

2.4.3 Role Activity Diagrams

A Role Activity Diagram (RAD) is a business process modeling methodology presented by Praxis Plc to model and analyse business processes [7] RAD is similar to UML activity diagrams, but there main difference is that RAD emphasizes on responsibilities, while UML activity diagrams emphasizes on orchestrating the activities [13].

2.4.4 Event-driven Process Chains

An Event-driven Process Chain (EPC) is a business process modeling technique designed to produce business understandable models. It was developed in 1992 by August-Wilhelm and it has grown to become one of the most common business process modeling methodology. Its focus on logical connectors and functions make it appears. technically oriented, while it is in fact helps the business stakeholders, rather than the IT stakeholders [9].

2.4.5 Integrated Definition for Functional Modeling

Integrated Definition for Functional Modeling (IDEF) is defined as a series of modeling languages initially designed to be used in the field of software engineering. The first IDEF modeling language created in 1975 as a side-product of the Integrated Computer-Aided Manufacturing program of the United States Air Force [10]. Since then, the IDEF group has grown to a set of 16 modeling languages. However, only the first five IDEF languages have matured into well-accepted modeling languages, while the rest were never developed any further than their initial definition.

2.4.6 Business Process Modeling Notation

Business Process Modeling Notation (BPMN) is a methodology of business process modeling intended to provide human-clear representations of business processes [11,12]. The original version of BPMN was developed by the Business Process Management Initiative (BPMI) in 2004, but the Object Modeling Group (OMG) adopted the language as a standard for business process modeling in 2006 [15]. Five years later, BPMN became the most common notation for modeling business processes.

3. UML as a BPM tool

UML (Unified Modeling Language) is a standard language for modeling, visualizing, constructing, and documenting the artifacts of software systems. UML was created by the Object Management Group (OMG) and UML 1.0 specification draft was proposed to the OMG in January 1997 [13]. It was initially started to capture the behavior of complex software and non-software system and now it has become a de-facto standard for modeling software systems. Although UML is used for non-software systems, the emphasis is on modeling software applications (especially object oriented software applications). It provides different diagrams that can be used to model the static and the dynamic behavior of systems. With its diverse number of diagrams, we think that UML can be used as a BPM tool. For this task, we chose use case diagram, activity diagram, and sequence diagram since these three diagrams. Throughout this work we use an inventory control system of the ELMERGAB cement factory as a case study to show how UML can be used to model a business process.

3.1 Organizational chart of ELMERGAB cement factory

ELMERGAB cement factory is one of a cement factories owned and administrated by the Al- Arabiya Company of cement. It is located in the town of Alkhoms and provides regular construction cement for local market. Figure 5 shows the organizational chart of company.

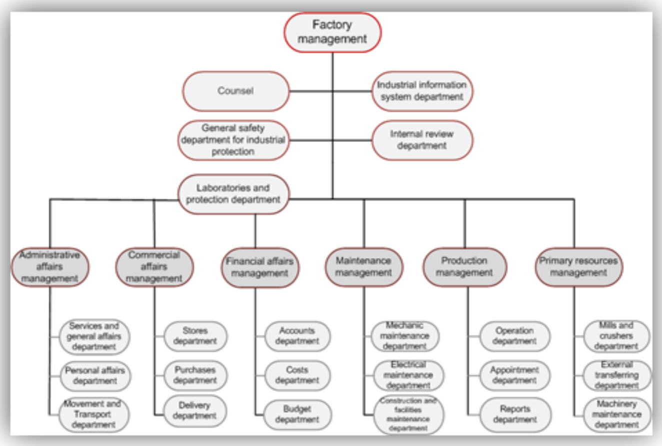


Figure 5: Organizational chart of ELMERGAB Company

In our study, we took the inventory control system of the company as our case study. First we defined the main business processes of the system, which are:

- Materials request.
- Internal purchase request.
- External purchase request.
- Materials receive.
- Returning materials or Discount on purchase
- Examining materials.
- Adding materials.

Next, we used UML to model these business processes with different UML diagrams.

3.2 Modeling the business processes of the system using UML

To model the business process of the studied system we used the following UML diagrams: use case diagram to define the system procedures, activity diagram to model the dynamic behavior of the system, and sequence diagram to express the activities of the system in terms of data massages.

3.2.1 Defining the system procedures with uses case diagrams

Use Case diagram is a set of tasks, subtasks, and a set of participants. For the studied system, we defined the following procedures: request materials, internal purchase request, and external purchase request, receive materials, and communicate with the suppliers. Figures 6, 7, 8, 9 show the uses case diagrams of the procedures respectively

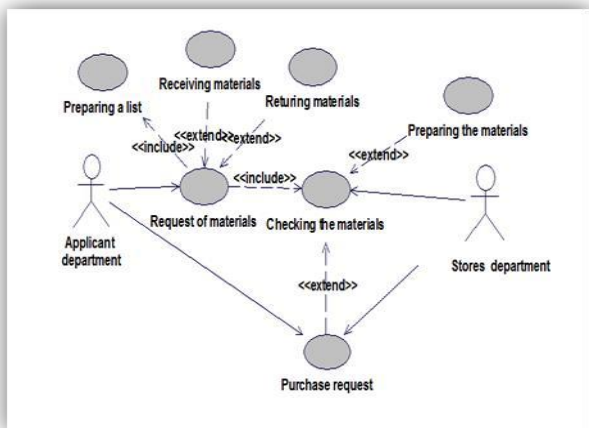


Figure 6: Use case diagram of request materials

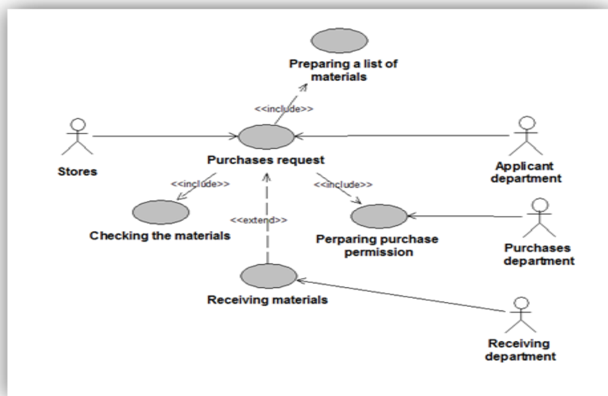


Figure 7: Use case diagram of internal request.

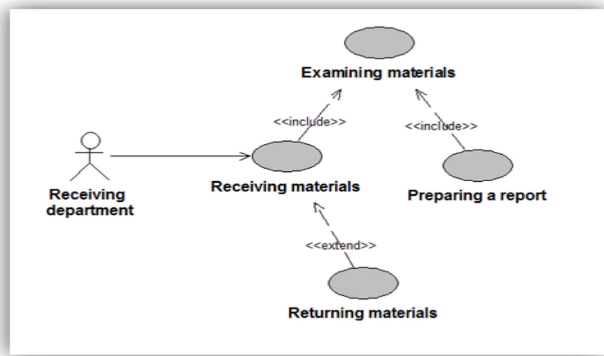


Figure 8: Use case diagram of receive materials

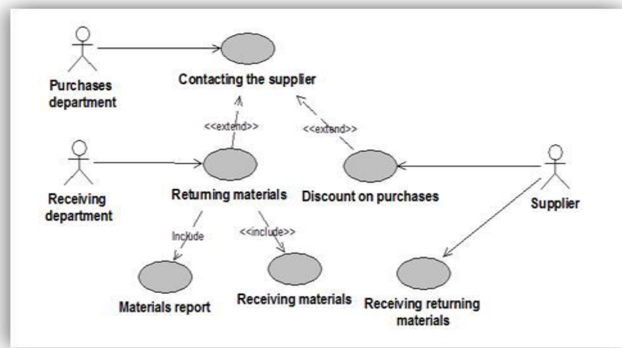


Figure 9: Communicate with the suppliers

3.2.2 Modeling dynamic behavior of the system with activity diagrams

Activity diagrams show the dynamic behavior of a system. They present certain activities that a group of objects conduct and explain the way in which a number of events and decisions happen in a sequence via a number of activities [13]. Activity diagrams are oriented towards the functional and informational perspectives of the business process [14]. The business processes of the system are represented by a set activity diagrams, as shown in figures 11, 12, 13, 14, 15, 16, 17. For instance, figure 10 represents the activity diagram of the process "request material". The first activity in the process is issuing a request, followed by exchange of materials with exchange permission, next checking the materials. After that, we have two branches in the workflow, one if the materials are accepted in case of match and the other one if the materials are not accepted because of non-match, which leads the flow to another activity process, i.e. return materials and the activities workflow continue . The activities continue to flow in accordance with the processes of the system. The activity diagrams focus on how the work is performed and what information entities are created.

Figure 10: Activity diagram of request materials.

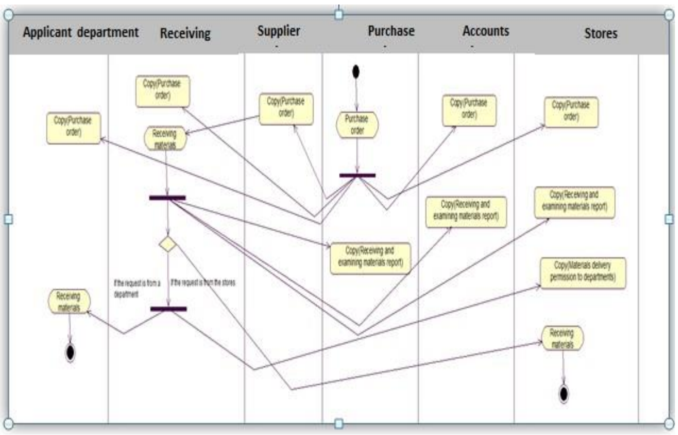


Figure 11: Activity diagram of request internal purchase.

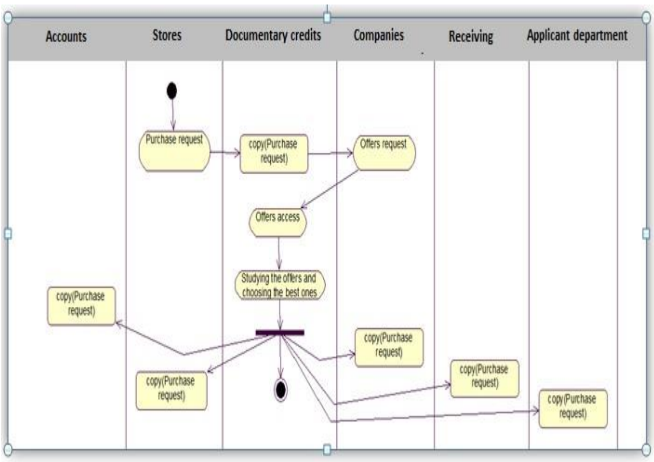


Figure 12: Activity diagram of external purchase request.

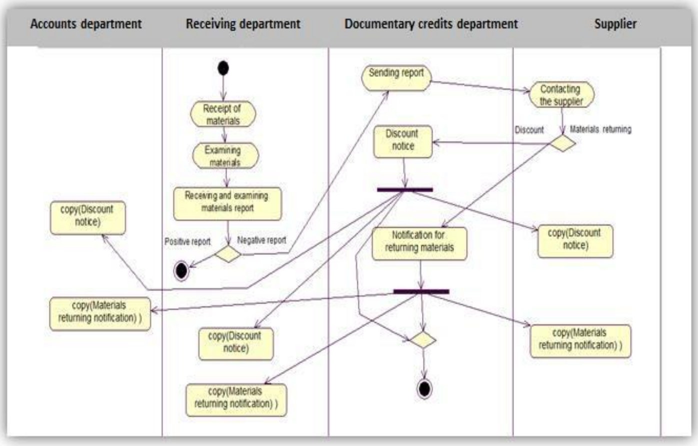


Figure 13: Activity diagram of receiving materials.

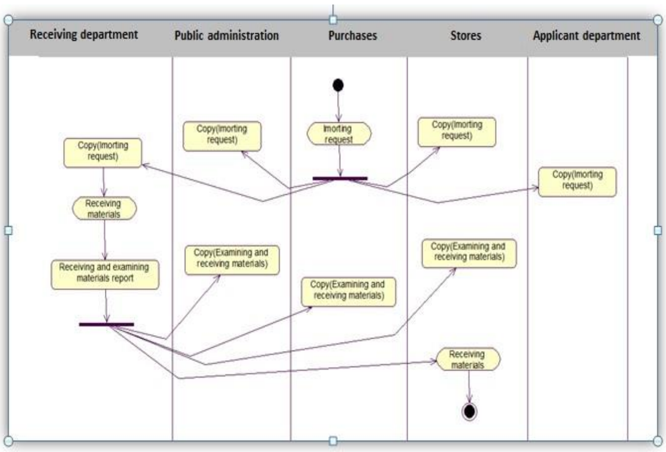


Figure 14: Activity diagram of request importing.

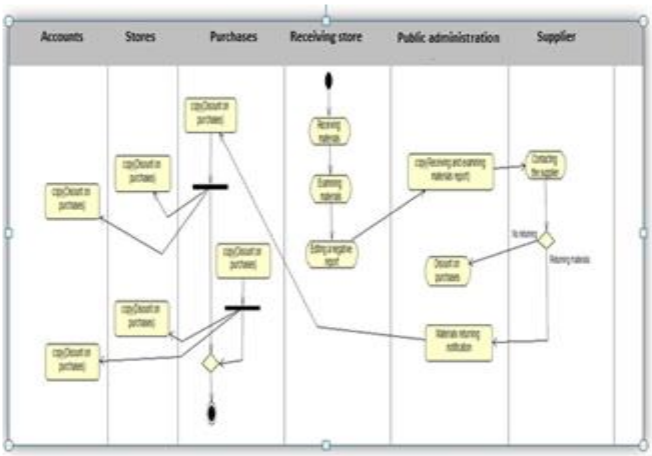


Figure 15: Activity diagram of returning materials, and discount on purchase

3.2.3 Sequence diagram as a BPM

Sequence diagrams illustrate the order of flowing of the data messages in a system according time order [13]. They describe how objects collaborate over the process step in time and meet the requirements of the behavioral perspective.

The business processes of our case study system are represented using sequence diagrams as shown in figures 18, 19, 20, 21, 22, and 23. Figure 18, for example, shows the sequence diagram of the process “request materials”. The diagram displays the chronological order of the processes that take place in the process “request materials”.

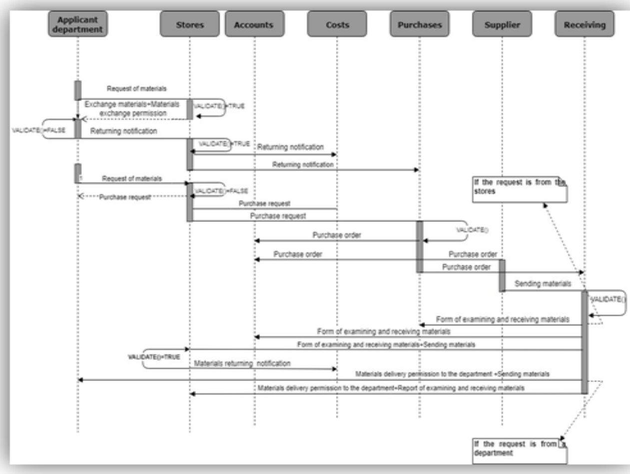


Figure 16: Sequence diagram of request materials.

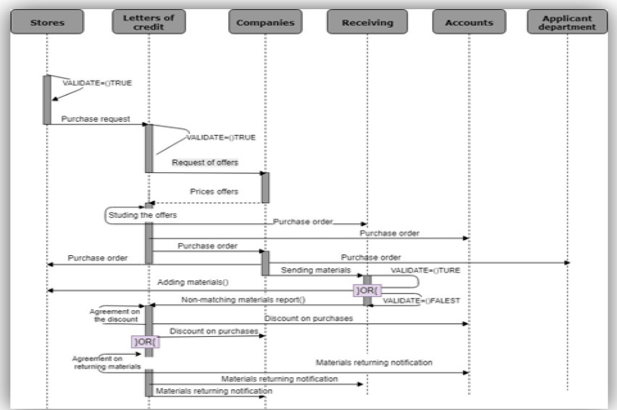


Figure 17: Sequence diagram request purchase.

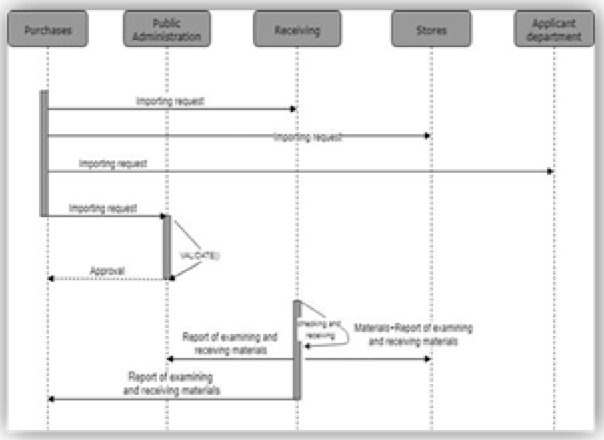


Figure 18: Sequence diagram of external request.

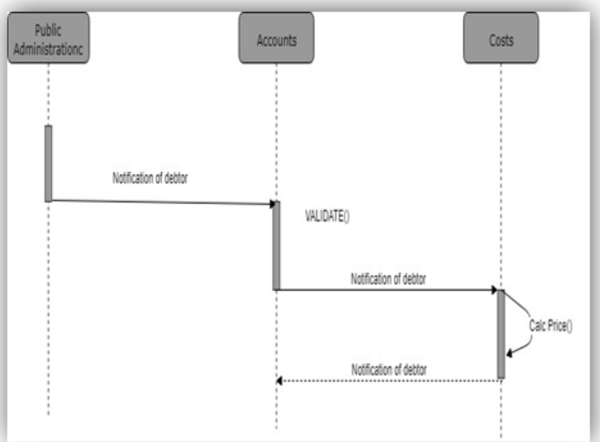


Figure 19: Sequence diagram of notification of debtor.

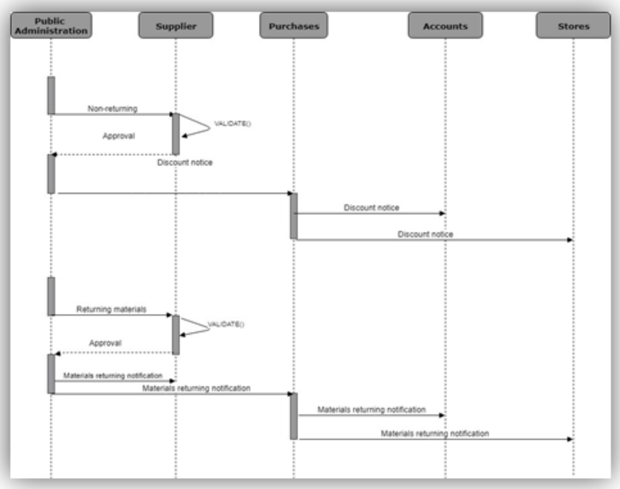


Figure 20: Sequence diagram of returning materials

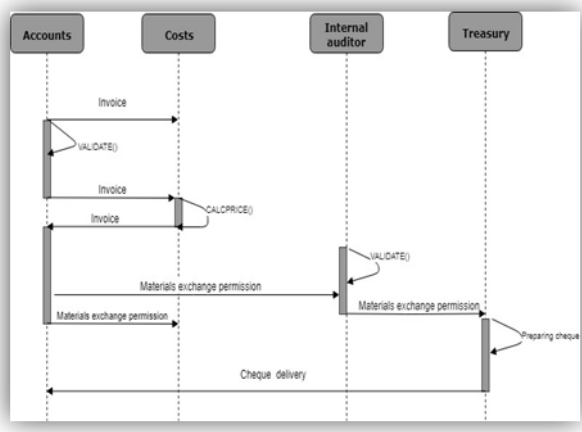


Figure 21: Sequence diagram of invoice and cheque preparing

4. Conclusion

In this paper, we explored the stability of using UML as a Business Process Modeling tool (BPM). BPM is a critical component of any business improvement process; it allows capturing a broad outline and procedures that govern what it is a business does. We showed that different diagrams of UML, such as use case diagram, activity diagram and sequence diagram, can be used to model the business process of any case. By using UML as a BPM tool, one can profit from the advantages of UML including: its popularity in the industrial community as well as in the academic community, its standardization by a met- model governed by an international research group (OMG: Object Modeling Group), and its rich modeling tool with different visual diagrams and notations. A case study was presented to better understand how UML can be used as a BPM tool.

References

[1] Aguilar-Saven and Ruth Sara, Business process modelling: Review and framework, International Journal of production economics 90.2, 2004, pp. 129-149.

[2] Grzegorzm Nalepa and Krzysztofm Kluza Sebastian Ernst, Modeling and Analysis of Business Processes with Business Rules, In Jason A. Beckmann, Eds., Business Process Modeling: Software Engineering, Analysis and Applications, Nova Science Publishers, 2011.

[3] J.Cardoso, I.Vanderfeesten, H. Reijers, Computing coupling for business process models (2010),

http://eden.dei.uc.pt/~jcardoso/Research/Papers/Old%20paper%20format / Caise-19th- Coupling-Cardoso-

[4] J. Becker, M.Rosemann, C.von Uthmann, Guidelines of Business ProcessModeling. In: van der Aalst W., Desel J., Oberweis A. (eds) Business Process Management. Lecture Notes in Computer Science, vol 1806. Springer, Berlin, Heidelberg, 2000.

[5] Hafedh Mili, Guitta Bou Jaoude, Éric Lefebvre, Guy Tremblay, Alex Petrenko, Business process modeling languages: Sorting through the alphabet soup, ACM Computing surveys, Volume 43 Issue 1, November 2010.

[6] J. L. Peterson, Petri Nets Theory and the Modeling of Systems, Englewood Cliffs, NJ: Prentice-Hall, 1981.

[7] John Murdoch and John, A Modelling Engineering Design Process with Role Activity Diagrams, Journal of Integrated Design & Process Science, vol. 4, no. 2, pp. 45-65, 2000.

[8] Michaelzur Muehlena and Marta Indulska, Modeling languages for business processes and business rules: A representational analysis, Elseveir Information Systems, Volume 35, Issue 4, June 2010, pp. 379-390.

[9] A. Scheer, O Thomas, O Adam, Process Modeling Using Event-Driven Process Chains, Process-aware information systems, Wiley Online Library, 2005.

[10] Integrated Definition for Functional Modeling (IDEF 0), Federal Information Processing Standards Publication 183, 1993, http://www.idef.com/ Downloads/pdf/idef0. pdf.

[11] Katalina Grigorova and Kaloyan Mironov, Comparison of business process modeling standards, International Journal of

Engineering Sciences & Management Research, September 2014.

[12] Stephen White, BPMN 2.0 Handbook Second Edition: Methods, Concepts, Case Studies and Standards in Business Process Management Notation. Future Strategies Inc, 2011.

[13] James Rumbaugh, Grady Booch, Ivar Jacobson, The Unified Modeling Language Reference Manual, Addison-Wesley, 2010.

[14] Hans-Erik Eriksson, UML toolkit, John Wiley &Sonc, 1998.

[15] A. Schatten and S. Josef , Agile business process management with sense and respond, proceedings of the IEEE International Conference on e-Business Engineering (ICEBE'07), Hong Kong, Oct. 24-26/2007.

4

Comparative Analysis of Various Cost Models on the basic of Certain Parameters

Comparative Analysis of Various Cost Models on the basic of Certain Parameters

Khari A. Armih
Al-zawiya College of Computer Technology
Khari.armih@gmail.com

Moktar M. Lahrashe
Al-zawiya College of Computer Technology
mlahrashe@gmail.com

Abstract

With the growth in heterogeneity, the current focus of designing parallel performance cost models is on providing low-level details of parallel execution to the programs to enable resource-aware partitioning and dynamic load balancing procedures, in particular, for heterogeneous parallel architectures.
This Paper presents a survey of a classification of current and emerging cost models for parallel and distributed environments as well as algorithmic skeletons, and addressing major challenges such as complexity, target architectures, Optimisation and Skeleton-based Approachs.

Keyword: Parallel, heterogeneous, Algorithmic skeletons, Cost model

1. Introduction

Models of parallel computation play an important role in designing and optimising parallel algorithms and applications. These models assist the developer in understanding all-important aspects of the underlying architecture without knowing unnecessary details. Moreover, parallel computational models ware used to predict the performance of a given parallel program on a given parallel machine.

The common way of predicting the performance of parallel program is to derive a symbolic mathematical formula that describes the execution time of that program. This formula has a set of parameters that usually include the size of program, number of processors, and other hardware and algorithm characteristics that affect the execution time of the program. These parameters will be given by a programmer, benchmarking, or profiling tools.

Skeleton-based and similarly structured frameworks have employed these parallel computational models to predict the performance of the parallel algorithms in the early stages of the design process. Consequently, these computational models can assist and guide scheduling algorithmic skeletons on a wide variety of architectures.

Several parallel computational models had been developed for parallel-distributed systems to guide parallel algorithm designers. Good general surveys of early research are given in [5, 6, 7, 10] and a more recent survey is given in [8].

In this paper, we survey several well-known parallel cost models that have been proposed for parallel and distributed environments as well as algorithmic skeletons.

Finally, this survey doesn't aim to give a comprehensive survey of cost models, which would be much beyond the scope of this report, but we classify and discuss essential aspects of the existing performance cost models, and give references for further reading.

2. The Family of PRAM Models

The most widely-used cost model in parallel computing is the Parallel Random Access Machine (PRAM) model [9]. The PRAM model was based on the RAM model [10] of sequential computation. The model consists of a global shared memory and a set of sequential processors that operate synchronously. The model assumes that at each synchronous step, each processor can access any memory location in one unit time regardless of the memory location. The PRAM model provides a useful guide for parallel algorithm designers and thereby allows them to ignore all the architecture details of the underlying hardware and concentrate on application-specific issues.

Despite the useful basis provided by the PRAM model for parallel algorithm design, it cannot reflect all the costs of a real parallel machine. This results in non-portable programs due to a number of assumptions made by the model by ignoring the cost of some parallel activities such as synchronisation, memory contention, and communication latency or bandwidth.

Therefore, several realistic variants of PRAM-based models have been introduced to make PRAM more practical. These variants attempt to account for the cost issues of real parallel machines. For example, models such as Block PRAM (BPRAM) [11], Local-Memory PRAM (LPRAM) [12], and Asynchronous PRAMs [13] seek to include the latency cost with the standard PRAM model.

Another PRAM variant is asynchronous PRAMs that add some degree of asynchrony into the basic PRAM model in order to ease the restriction on processors synchronisation. These models differ in the way of the processors are synchronised. They include the

Asynchronous Parallel Random Access Machine model (APRAM) [14] that addresses the synchronisation assumption of the basic PRAM model to allow asynchronous execution, and the Hierarchical PRAM (HPRAM) model [15], which uses the PRAM model as a sub-model, consists of a collection of synchronous PRAMs that operate asynchronously from each other. Another asynchronous model by Gibbons et al. [16] allows the processors to run in an asynchronous manner.

The CRCW PRAM model [17] and the QRQW PRAM model [18, 19] account for memory locations contention, where the read and write to shared memory locations are done concurrently.

3. BSP and Variants

The Bulk Synchronous Parallel (BSP) model [20, 21] is a parallel computation model that provides a simple way of writing parallel programs for a wide range (architecture-independent) of parallel architectures by offering a bridging model that links software and architecture. In addition, it provides a straightforward way for realistic performance prediction for application design on a variety of different parallel architectures including distributed-memory systems, shared-memory multiprocessors, and networks of workstations. Practically, the BSP model aims to provide a bridge model between the software and hardware.

The BSP model consists of a collection of processors that communicate using message passing. The computations in the BSP model are formulated as a series of super steps. Conceptually, each super step is divided into three stages. In the first stage, all processors concurrently compute using only local data. In the second stage, processors exchange messages with each other. In the third stage, all of the processors execute a barrier synchronisation, after they finished sending and receiving messages.

Compared with the PRAM model the BSP model is more realistic, since it accounts for two cost issues of the real parallel machines, namely communication cost and memory latency cost.

Since BSP programs are based on sequential super steps, the model provides a very straightforward approach to cost estimation by firstly, calculating the cost of each super step, and secondly, calculating the cost of the whole BSP program by summing the cost of the super steps. The cost of each super step in a BSP program is given by:

$$T_{superstep} = w + hg + l$$

Where w reflects the cost of the longest running local computation in any of the processors, l is a constant cost (the cost of the barrier synchronisation) that depends on the performance of the underlying hardware, h is the number of messages sent or received per processor and g captures the measurement of the ability of the communication network to deliver these messages. A number of parallel implementations have been proposed using the BSP model [22, 23, 24, 25];

4. The LogP Model Family

LogP [29, 30] is an architecture-independent parallel computation model for designing and analysing parallel algorithms. It is a model for distributed-memory multiprocessors where processors communicate using message passing. LogP provides a good balance between abstraction and simplicity by using a few parameters to characterise the parallel computers and enabling the user to ignore all unnecessary details.

Like the BSP model, LogP is more realistic, since both models try to capture the communication latency and bandwidth through parameters [31], and both models allow the processors to work in a completely asynchronous manner.

Nevertheless, the LogP model gives a more realistic picture than BSP, since LogP has more control over the machine resources by capturing the communication overhead. Furthermore, LogP can be use in parallel systems that are constructed from a collection of complete computers connected by a communication network.

Conceptually, the LogP model consists of a collection of sequential processors interacting through a communication network by exchanging messages, where each processor has direct access to a local memory. The parallel program is executed in an asynchronous way by all processors in the LogP machine.

The LogP model seeks to capture the communication network cost by describing the parallel computer in terms of four elements:

P: the machine's number of processors.

g: communication bandwidth for short message (gap).

L: communication delay (latency).

o: communication overhead (overhead) .

The latency is an upper bound on the time required to send a message from a source processor to its target processor. The overhead is the fixed amount of time that a processor requires to prepare for sending or receiving a message; during this time, the processor cannot perform other operations. The gap is the minimum time interval between sending two messages on the same processor. The gap is the inverse of the available per-processor communication bandwidth for a short message. Several researchers [31, 32, 33] have shown that the LogP model delivers good and accurate predictions for small messages. A number of different extensions of the classic LogP model have been developed to improve prediction accuracy by addressing different communication network issues:

4.1 LogGP

The LogGP model by Alexandrov et al.[34, 35] is an extension of the basic LogP model. Since LogP facilitates only short-message communication transmission between processors and ignores long messages, the LogGP model extended LogP to provide a simple linear model that can model both short- and long-messages.

Just as in the original LogP model, LogGP is developed for distributed memory multiprocessors, where each processor has access to local memory. The processors work in an asynchronous way and communicate with other processors by point-to-point messages.

LogGP uses the parameters (latency, overhead, gap, and number of processors) that were introduced by the LogP model to characterise communication performance. In addition, it introduces a new additional parameter, Gap per byte, G, which captures the communication bandwidth for long message. Thus, the LogGP model uses 1/g for short message and 1/G for long message. In the LogP model, sending a k bytes message between two processors requires sending [k/w] messages, where w is the underlying message size of the machine. This takes:

$$O + \left(\left[\frac{K}{W}\right] - 1\right) * \max(g, O) + l + O\ cysles$$

While sending everything as a single large message in the LogGP model takes:

$$O + (K - 1) * G + L + O\ cycles$$

4.2 LogGPS

The LogGPS model [36] is a parallel computational model that extends LogGP to include the synchronisation cost. As in the original LogP model, LogGP eliminates the synchronisation cost that is needed in other models such as PRAM and BSP. This elimination might make LogGP not accurate enough, while it ignores the need for synchronisation when sending a long message in programs that use high-level communication libraries such as MPI. The LogGPS model has been proposed to address this shortcoming in LogGP.

Sending a long message between two processors is often performed by sending a small message to the receiver to check if it is ready to receive the original message. The process causes the sender processor to be synchronised with the receiver processor and adds a synchronisation cost to the overhead. Thus, the LogGPS model adds one additional parameter, S, which reflects the message-size threshold for synchronising sends.

4.3 HLogGP

Another extension of the LogGP model is the Heterogeneous LogGP [37] model. HLogGP has been specifically proposed for heterogeneous parallel systems to capture the heterogeneity in both communication networks and computational nodes. Since the underlying architecture of the LogGP model is very similar to the cluster architecture, it is considered an appropriate starting point for developing HLogGP.

The HLogGP model extends LogGP by transforming its scalar parameters into matrices. Conceptually, the parameters for overhead and gap are replaced by vector parameters, and latency and Gap is replaced by matrix parameters. Furthermore, to capture the heterogeneity in the computational nodes, the parameter for the number of processors is replaced by a computational power vector, which describes the physical features for every node in the system. The model has been shown to deliver an accurate prediction on heterogeneous clusters.

4.4 Other LogP extension

Besides the previously mentioned LogP extensions, other extensions have been proposed that aim to address different issues in communication. We briefly discuss some here:

LogP-HMM [38] is a parallel computational model based on the LogP model. The idea of LogP-HMM is to develop an accurate model that accounts for the impact of both network communication and multilevel memory on the performance of parallel algorithms and applications. Therefore, the LogP HMM model extends LogP with the HMM model [39], where the LogP model deals with network communication and the HMM model addresses the memory hierarchy.

LoGPC [40] is a simple model that extends LogP and its extension LogGP to address another aspect of communication networks. It uses the features of both models to account for short message as well as long message bandwidth. Practically, the LoGPC model is intended to capture the impact of message transmissions of size m.

Parametrised LogP [41] or (pLogP for short) is a slight extension of the LogP and LogGP models. This model can accurately predict the completion time of collective operations in message passing models such as MPI.

Five parameters are used in the pLogP model to characterised the network. Like the LogP model, it uses P as the number of processors and L is the end-to-end latency, but the original parameters o and g are replaced by a function of message size, where o_s (m) and o_r (m) are the sender and receiver overheads of the message size m, and g(m) is the delay between consecutive message transmissions of size m.

5. HiHCoHP

The HiHCoHP model [42, 43] is a realistic communication model for hyperclusters (multi-level clusters of clusters of processors) with heterogeneous processors. It aims to capture the important features of a real hypercluster such as bandwidth and transmission cost.

The HiHCoHP model is based on several parameters that reflect the heterogeneity of hyperclusters:

• P_i ("computing power"): HiHCoHP considers the computing power as N heterogeneous nodes that may differ in computational power (computation and memory speed).

• (" message processing"): the P_a and P_b set up fixed communication cost is $(\sigma_a^{(k)} + \sigma_b^{(k)})$; and the cost of message packing in P_a is $\pi_a^{(k)}$ and message unpacking in P_b is $\pi_b^{(k)}$.

• $\lambda^{(k)}$ ("network latency"): or the end-to-end latency is the amount that is required to send one packet between the source node and destination node at level-k of the network.

• $\beta^{(k)}$ ("link-bandwidth"): the amount of data that can be sent between two nodes at level-k of the network.

• $k^{(k)}$ ("Network capacity"): the maximum number of packets that can be transmitted at once.

So the total end-to-end communication time of sending p-packet message from node P_a to node P_b is given by:

$$(\sigma_a^{(k)} + \sigma_b^{(k)}) + (\pi_a^{(k)} + \pi_b^{(k)})p + \lambda^{(k)} + \Delta(p)$$

Where $\Delta(p) = (p-1)/\beta^{(k)}$ in a pipeline network, and $\Delta(p) = \lambda^{(k)}(p-1)$ in a store-and-forward network.

6. DRUM

Another type of parallel computational model are architectures-aware cost models. One of the well-known models is the Dynamic Resource Utilisation Model [44, 45] or (DRUM). DRUM is developed to support resource-aware load balancing in a heterogeneous environment such as clusters and hierarchical clusters (clusters of clusters, or clusters of multiprocessors).

DRUM accounts for the capabilities of both network and computing resources. In particular, DRUM is intended to encapsulate information about the underlying hardware, and provide monitoring facilities for hardware capabilities evaluation. Benchmarks are used to assess the capabilities of computational, memory and communication resources.

Each node in the tree structure of the DRUM model has been given a single value called "power", which represents the portion size of the total load that can be assigned to that node based on its processing and communication power.

The power of node n in the DRUM model is calculated as the weighted sum of processing power p_n and communication power c_n:

$$power_n = w_n^{comm} c_n + w_n^{cpu} p_n, \quad w_n^{comm} + w_n^{cpu} = 1$$

7. Skeletons

To improve the performance of parallel applications, performance cost models are associated with algorithmic skeletons to accurately predict the costs of parallel applications. More precisely, the aim of these performance models is to assist the parallel skeletons, either implicitly or explicitly, to guide scheduling on a wide variety of architectures.

This section deals with skeleton-associated performance cost models. Several skeleton-based and similarly structured frameworks have employed performance cost models for various kinds of skeletons. Some of the skeleton-based frameworks employ the well-known cost models and their variants such as the models that were previously mentioned, and others use their own performance prediction tools to estimate the performance of a given program.

Here, we briefly outline the skeleton-based frameworks that employ high-level cost models.

7.1 Darlington's group

Performance models are proposed in [46] for processor farms, divide and conquer (DC), and pipeline skeletons. For example, a performance model has been proposed for a divide and conquer skeleton to provide a prediction of the execution time for given program, which is used to guide resource allocation.

In this model, the total execution time required to solve a problem of size N on P processors is given by:

$$T_{solN} = \sum_{i=1}^{log(p)} \left(T_{\mathrm{divN/2i-1}} + T_{\mathrm{combN/2i}} + T_{comms}\right) + T_{\mathrm{solN/P}}$$

Where T_{divN} is the time to divide a problem of size N, T_{combN} is the time to combine the two results, and T_{comms} is the communication time between processors.

7.2 BSP-based Approaches

Several authors associate the BSP model with algorithmic skeletons for performance optimisation.

For example, Skel-BSP [23, 47] is a subset of P3L that uses an extension of the BSP model called the Edinburgh-Decomposable-BSP model to achieve performance portability for skeletal programming. EdD-BSP extends the BSP model by adding partition and join operations to partition and reunify BSP submachines which allows subset synchronisation as in D-BSP.

Compared to the standard BSP model, EdD-BSP replaces the g parameter with two parameters, which are g∞ and $N_1/2$, and then estimates the cost of two kinds of supersteps:

a) The cost of computational supersteps is given by:

$$T = W + hg\infty(N_{1/2}/h + 1) + L$$

b) The cost of partition and join superstep is given by L

Another BSP-based approach is Bulk-Synchronous Parallel ML (BSML) [151].

BSBML is a functional data parallel language for programming BSP algorithms using a set of high-level parallel primitives. It uses the BSP model to predict the performance of a given program on a wide variety of parallel architectures.

7.3 P3L

P3L uses a variant of the LogP model to predict and optimise program performance on parallel systems. An analytic model is presented in [49] for the basic forms of parallelism to be used by the template-based compiler of the P3L language.

This model is more complex than LogP, since it is intended to capture several hardware features, such as the speed of processor, node architecture, and network bandwidth and latency.

Here we briefly describe the analytical model for the high level template that is related to this work.

The Map construct is implemented on an N dimension grid of processors. The computation time T of input granularity k is given by:

$$T(\mathrm{k}) = k\left(T_{dis}() + T_c \prod_{i=1}^{N} d_i + T_{col}\right)$$

Where:

T_c : seq. computation time.

d_i : data granularity for dimension i.

T_{dis}: data distribution time.

T_{col}: time for collecting results.

7.4 HOPP

The HOPP (Higher-order Parallel Programming) model [50, 51] is a methodology based on the BMF (Bird-Meertens Formalism)[52], where the program is expressed as a composition of higher-order functions.

The HOPP model uses a cost model introduced in [52] to predict the costs of programs. This cost model is implemented as an analyser for calculating the costs of possible implementations for a given program on a given distributed-memory machine.

In the HOPP model, the cost of a program is computed in terms of n steps:

$$cost = \sum_{i=1}^{i=n} C_{pi} + \sum_{i=0}^{i=n-1} C_{i,i+1}$$

where C_{pi} is the cost of phase i which depends on the number of processors and sequential implementation of the functions in that step, and $C_{i,i+1}$ is the cost of communication that may be incurred between step i and step $i + 1$.

7.5 SkelML

SkelML [53] gives performance models for a number of skeletons such as pipeline, farm, and fold Processor Chain skeletons. These models are based on the communication overhead and computation time that are involved in application execution. The skeleton performance models and profiling information help the SkelML compiler to determine useful parallelism.

8. Resource Metrics for Parallel Cost Models

The performance of parallel machines is dependent on the underlying architecture features. These features are referred to as resource metrics that characterise the parallel computational model. Thus, a computational model can be identified by a set of these resource metrics. We now consider some resource metrics that are visible in all parallel computational models.

Number of processors the number of processor in the machine.

Communication Latency is the time needed to transfer a message from one processor to another processor; this depends on both the network topology and technology.

Communication Bandwidth is the amount of data that can be sent within a given time; this is a limited resource in practice and depends on the network interface.

Communication Overhead is the period of time that is needed by the processor for sending and receiving message. The amount of overhead depends on network topology features such as communication protocols.

Computational power Computational power is the amount of work finished by one processor in a given time for a specific task; this value depends on the processor's capabilities and the task being processed.

Synchronous/Asynchronous In a synchronous model, all processors are synchronised after executing each instruction. Processors may run semi asynchronously, where the computations occur asynchronously within each phase and all processors are synchronised at each phase.

Table 1 shows how these resource metrics contribute in forming the computational models considered above.

Table 1: Resource metrics for parallel computation

Model	Procs	Latency	Bandwidth	Overhead	Computational Power	Synch Asynch
PRAM	√					Synch
BSP	√	√	√			Semi-synch
LogP	√	√	√	√		Asynch
LogGP	√	√	√	√		Asynch
HLogGP	√	√	√	√	√	Asynch
SkelML	√			√		Asynch
P3L	√	√	√	√	√	Asynch
Ske-BSP	√	√	√			Semi-synch
HOPP	√	√	√	√	√	Asynch

9. Conclusion

Here, we have carried out the comparative systematic study of some software cost estimation models in conjunction with their relevant techniques. Although it would be very difficult to say, which model is preeminent as it is vastly based on the size of software and certain other underlying hardware specifications. We claim performance cost models that based on architectural details of a parallel machine to provide cost estimation of a given program on a given machine, provides a reasonable trade-of between the accuracy and simplicity needed for our heterogeneous skeletons.

Reference

[1] P. B. Gibbons, Y. Matias, and V. Ramachandran. The QRQW PRAM: Accounting for Contention in Parallel Algorithms. In Proceedings of the Fifth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA '94, pages 638-648, Philadelphia, PA, USA, 1994. Society for Industrial and Applied Mathematics.

[2] P. Gibbons, Y. Matias, and V. Ramachandran. E_cient Low-Contention Parallel Algorithms. In The 1994 ACM Symp. on Parallel Algorithms and Architectures, pages 236-247, 1994.

[3] L. G. Valiant. A Bridging Model for Parallel Computation. Commun. ACM, 33:103-111, August 1990.

[4] D. B. Skillicorn, J. M. D. Hill, and W. F. McColl. Questions and Answers about BSP. Scientific Programming, 6(3):249-274, 1997.

[5] J. M. D. Hill, B. McColl, D. C. Stefanescu, M. W. Goudreau, K. Lang, S. B. Rao, T. Suel, T. Tsantilas, and R. H. Bisseling. BSPlib: The BSP Programming Library. Parallel Computing, 24(14):1947-1980, 1998.

[6] A. Zavanella. Skel-BSP: Performance Portability for Skeletal Programming. In Proceedings of the 8th International Conference on High-Performance Computing and Networking, HPCN Europe 2000, pages 290-299, London, UK, UK, 2000. Springer-Verlag.

[7] M. Goudreau, K. Lang, S. Rao, T. Suel, and T. Tsantilas. Towards Efficiency and Portability: Programming with the BSP Model. In Proceedings 216 Bibliography of the Eighth

Annual ACM Symposium on Parallel Algorithms and Architectures, SPAA '96, pages 1-12, New York, NY, USA, 1996. ACM.

[8] A. Goldchleger, A. Goldman, U. Hayashida, and F. Kon. The implementa-tion of the BSP Parallel Computing Model on the InteGrade Grid Middle-ware. In Proceedings of the 3rd International Workshop on Middleware for Grid Computing, MGC '05, pages 1-6, New York, NY, USA, 2005. ACM.

[9] P. de la Torre and C. Kruskal. Submachine Locality in the Bulk Syn-chronous Setting. In Luc Boug, Pierre Fraigniaud, Anne Mignotte, and Yves Robert, editors, Euro-Par'96 Parallel Processing, volume 1124 of Lecture Notes in Computer Science, pages 352-358. Springer Berlin / Heidelberg, 1996. 10.1007/BFb0024723.

[10] B. H. Juurlink and H. G. Wijsho_. The E-BSP model: Incorporating General Locality and Unbalanced Communication into the BSP Model. In Luc Boug, Pierre Fraigniaud, Anne Mignotte, and Yves Robert, editors, Euro-Par'96 Parallel Processing, volume 1124 of Lecture Notes in Computer Science, pages 339-347. Springer Berlin Heidelberg, 1996.

[11] L. G. Valiant. A Bridging Model for Multi-core Computing. In Proceedings of the 16th annual European symposium on Algorithms, ESA '08, pages 13-28, Berlin, Heidelberg, 2008. Springer-Verlag.

[12] D. Culler, R. Karp, D. Patterson, A. Sahay, K. E. Schauser, E. Santos, R. Subramonian, and T. von Eicken. LogP: Towards A Realistic Model of Parallel Computation. In Proceedings of the Fourth ACM SIGPLAN Symposium on Principles and

Practice of Parallel Programming, PPOPP '93, pages 1-12, New York, NY, USA, 1993. ACM.

[13] D. E. Culler, R. M. Karp, D. Patterson, A. Sahay, E. E. Santos, K. E. Schauser, R. Subramonian, and T. von Eicken. LogP: A Practical Model of Parallel Computation. Commun. ACM, 39(11):78-85, November 1996.

[14] G. Bilardi, K. T. Herley, A. Pietracaprina, G. Pucci, and P. Spirakis. BSP vs LogP. In Proceedings of the Eighth Annual ACM Symposium on Parallel Algorithms and Architectures, SPAA '96, pages 25-32, New York, NY, USA, 1996. ACM.

[15] T. Hoeer, L. Cerquetti, and F. Mietke. A Practical Approach to the Rating of Barrier Algorithms Using the LogP Model and Open MPI. In Proceedings of the 2005 International Conference on Parallel Processing Workshops, ICPPW '05, pages 562-569, Washington, DC, USA, 2005. IEEE Computer Society.

[16] D. Culler, L. T. Liu, R. P. Martin, and C. Yoshikawa. LogP Performance Assessment of Fast Network Interfaces, 1996.

[17] A. Alexandrov, M. F. Ionescu, K. E. Schauser, and C. Scheiman. LogGP: incorporating long messages into the LogP model | one step closer towards a realistic model for parallel computation. In Proceedings of the seventh annual ACM symposium on Parallel algorithms and architectures, SPAA '95, pages 95-105, New York, NY, USA, 1995. ACM.

[18] A. Alexandrov, M. F. Ionescu, K. E. Schauser, and C. Scheiman. LogGP: Incorporating Long Messages into the LogP Model for Parallel Computation. Journal of Parallel and Distributed Computing, 44(1):71-79, 1997.

[19] F. Ino, N. Fujimoto, and K. Hagihara. LogGPS: A Parallel Computational Model for Synchronization Analysis. In Proceedings of the Eighth ACM SIGPLAN Symposium on Principles and Practices of Parallel Programming, PPoPP '01, pages 133-142, New York, NY, USA, 2001. ACM.

[20] J. L. Bosque and L. Pastor. A Parallel Computational Model for Heterogeneous Clusters. IEEE Trans. Parallel Distrib. Syst., 17:1390-1400, December 2006.

[21] Z. Li, P. H. Mills, and J. H. Reif. Models and Resource Metrics for Parallel and Distributed Computation. In Proceedings of the 28th Hawaii International Conference on System Sciences, HICSS '95, pages 51-, Washington, DC, USA, 1995. IEEE Computer Society.

[22] J. S. Vitter and E. A. M. Shriver. Optimal disk I/O with Parallel Block Transfer. In Proceedings of the Twenty-Second Annual ACM Symposium on Theory of Computing, STOC '90, pages 159-169, New York, NY, USA, 1990. ACM.

[23] C. A. Moritz and M. Frank. LoGPC: Modeling Network Contention in Message-Passing Programs. IEEE Trans. Parallel Distrib. Syst., 12(4):404-415, 2001.

[24] T. Kielmann, H. E. Bal, and S. Gorlatch. Bandwidth-E_cient Collective Communication for Clustered Wide Area Systems. In Parallel and Distributed Processing Symposium, 2000. IPDPS 2000. Proceedings. 14th International, pages 492 -499, 2000.

[25] F. Cappello, P. Fraigniaud, B. Mans, and A. L. Rosenberg. HiHCoHP: Toward a Realistic Communication Model for Hierarchical HyperClusters of Heterogeneous Processors. In

Proceedings of the 15th International Parallel & Distributed Processing Symposium, IPDPS '01, pages 42-, Washington, DC, USA, 2001. IEEE Computer Society.

[26] A. L. Rosenberg. Sharing PartitionableWorkloads in Heterogeneous NOWs: Greedier Is Not Better. In Proceedings of the 3rd IEEE International Conference on Cluster Computing, CLUSTER '01, pages 124-, Washington, DC, USA, 2001. IEEE Computer Society.

[27] K. D. Devine, E. G. Boman, R. T. Heaphy, B. A. Hendrickson, J. D. Teresco, J. Faik, J. E. Flaherty, and L. G. Gervasio. New Challenges in Dynamic Load Balancing. Appl. Numer. Math., 52(2-3):133{152, February 2005.

[28] J. Faik, J. D. Teresco, K. D. Devine, J. E. Flaherty, and L. G. Gervasio. A Model for Resource-aware Load Balancing on Heterogeneous Clusters. Technical Report CS-05-01, Williams College Department of Computer Science, 2005.

[29] J. Darlington, A. J. Field, P. G. Harrison, P. H. J. Kelly, D. W. N. Sharp, and Q. Wu. Parallel Programming Using Skeleton Functions. In PARLE '93: Proceedings of the 5th International PARLE Conference on Parallel Architectures and Languages Europe, pages 146-160, London, UK, 1993. Springer-Verlag.

[30] A. Zavanella. Skeletons and BSP: Performance Portability for Parallel Programming. PhD thesis, UNIPI, December 1999.

[31] F. Gava. BSP Functional Programming: Examples of a Cost Based Methodology. In Proceedings of the 8th international conference on Computational Science, Part I, ICCS '08, pages 375-385, Berlin, Heidelberg, 2008. Springer-Verlag.

[32] D. Pasetto and M. Vanneschi. Machine-independent analytical models for cost evaluation of template-based programs. In PDP, pages 485{492, 1997.

[33] R. Rangaswami. Compile-Time Cost Analysis for Parallel Programming. In Proceedings of the Second International Euro-Par Conference on Parallel Processing-Volume II, Euro-Par '96, pages 417-421, London, UK, 1996. Springer-Verlag.

[34] R. Rangaswami. A Cost Analysis for a Higher-order Parallel Programming Model. PhD Thesis. University of Edinburgh, Department of Computer Science, 1996.

[35] R. S. Bird. Algebraic Identities for Program Calculation. Comput. J., 32(2):122-126, April 1989.

[36] D. B. Skillicorn and W. Cai. A Cost Calculus for Parallel Functional Programming. J. Parallel Distrib. Comput., 28(1):65{83, 1995.

[37] T. A. Bratvold. Skeleton-Based Parallelisation of Functional Programs. PhD thesis, Heriot-Watt University, November 1994.

[38] D. G. Lowe. Object Recognition from Local Scale-Invariant Features. In ICCV '99: Proceedings of the International Conference on Computer Vision-Volume 2, page 1150, Washington, DC, USA, 1999. IEEE Computer Society.

[39] D. G. Lowe. Distinctive Image Features from Scale-Invariant Keypoints. Int. J. Comput. Vision, 60(2):91-110, 2004.

[40] S. Gupta. Performance Analysis of GPU Compared to Single-Core and Multi-Core CPU for Natural Language Applications.

IJACSA - International Journal of Advanced Computer Science and Applications, 2(5):50-53, 2011.

[41] M. McCool and S. D. Toit. Metaprogramming GPUs with Sh. AK Peters Ltd, 2004.

[42] AMD Corporation. ATI Stream Computing User Guide, Version 2.01. Technical report, 2010.

[43] K. Komatsu, K. Sato, Y. Arai, K. Koyama, H. Takizawa, and H. Kobayashi. Evaluating Performance and Portability of OpenCL Programs. In The Fifth International Workshop on Automatic Performance Tuning, UC Berkeley - CITRIS, Sutardja Dai Hall, Berkeley, CA 94720, USA, June 2010.

[44] M. M. Baskaran, U. Bondhugula, S. Krishnamoorthy, J. Ramanujam, A. Rountev, and P. Sadayappan. A Compiler Framework for Optimization of A_ne Loop Nests for GPGPUs. In Proceedings of the 22nd Annual International Conference on Supercomputing, ICS '08, pages 225-234, New York, NY, USA, 2008. ACM.

[45] S. Lee, S. Min, and R. Eigenmann. OpenMP to GPGPU: A Compiler Framework for Automatic Translation and Optimization. SIGPLAN Not, 44(4):101-110, February 2009.

[46] J. Hoberock and N. Bell. Thrust: A Parallel Template Library @http://www.meganewtons.com, 2009.

[47] CUDPP: CUDA Data Parallel Primitives Library@http://gpgpu.org/developer/cudpp, 2009.

[48] M. Steuwer, P. Kegel, and S. Gorlatch. SkelCL - A Portable Skeleton Library for High-Level GPU Programming. In

Proceedings of the 2011 IEEE International Symposium on Parallel and Distributed Processing Workshops and PhD Forum, IPDPSW '11, pages 1176-1182, Washington, DC, USA, 2011. IEEE Computer Society.

[49] A. D. Malony, S. Biersdor_, S. Shende, H. Jagode, S. Tomov, G. Juckeland, R. Dietrich, D. Poole, and C. Lamb. Parallel Performance Measurement of Heterogeneous Parallel Systems with GPUs. In Proceedings of the 2011 International Conference on Parallel Processing, ICPP '11, pages 176-185,

Washington, DC, USA, 2011. IEEE Computer Society.

[50] Y. Ogata, T. Endo, N. Maruyama, and S. Matsuoka. An Efficient, Model-based CPU-GPU Heterogeneous FFT Library. In Parallel and Distributed Processing, 2008. IPDPS 2008. IEEE International Symposium on, pages 1-10, April 2008.

[51] C. Yang, F. Wang, Y. Du, J. Chen, J. Liu, H. Yi, and K. Lu. Adaptive Optimization for Petascale Heterogeneous CPU/GPU Computing. In Proceedings of the 2010 IEEE International Conference on Cluster Computing, CLUSTER '10, pages 19-28, Washington, DC, USA, 2010. IEEE Computer Society.

[52] V. Strassen. Gaussian Elimination is not Optimal. Numerische Mathematik, 14(3):354-356, 1969.

[53] M. Aldinucci, M. Danelutto, and P. Teti. An advanced Environment Supporting Structured Parallel Programming in Java. Future Gener. Comput. Syst., 19(5):611{626, July 2003.

5

Implementation of Two Axises Platform Using PID Controller

Implementation of Two Axises Platform Using PID Controller

Najeb Koni Omran
Higher Institute of Science and Technology - Zawia,
r9_nono@yahoo.com

Assadeg Mohamed Akra
Higher Institute of Science and Technology - Zawia,
AlsadekAkra@yahoo.com

Isam S M Jalil
Higher Institute of Science and Technology –Al-Shumookh, Tripoli
see_es@yahoo.com

Abstract

This paper describes the design and implementation of two axis platform controller. The term two axis platform is a machine which rotate in two degrees of freedom, mainly it has two axis, one rotate about vertical axis and the other rotate about horizontal axis. The paper deals with two areas, study the DC motor as actuator and how it could be controlled using the computer software ("MATLAB", "Integral") technique to the integrated joint dynamic model and an independent joint control scheme was drive using a classic approach. The results obtained gave the required movement for the platform in terms of fluidity and smoothness in moving from one position to another one.

Keywords: PID Controller, multi-axis, MATLAB, brushless DC motors MAXON**.**

1. Introduction

In the industrial applications, design for the single axis motion control systems has been well investigated with traditional or modern control strategies [1]. Recently, precise contour control for the multi-axis systems has attracted much attention. As an example, defined a position vector and applied a modified and transform to determine the dominant position error vector so as to correct the position error vector in a two axes platform. To achieve a high degree of position and deposition accuracy, a coordination controller using DC servo motor [2].

The main goal of this paper was to develop a model of the two axises platform, design axises control schemes for (current, velocity, position) using classical approach.

Pitch, yaw and roll are the three dimensions of movement when an object moves through a medium. The terms may be used to describe an aero plane’s movements through the air. They are also applied to fish moving through water, and spacecraft moving through space, there are in fact six degrees of freedom of a rigid body moving in three-dimensional space.

As the movement along each of the three axes is independent of each other and independent of the rotation about any of these axes, the motion has six degrees of freedom. As illustrated in Figure (1), the Pitch axis which is for nose up or tail up, Yaw axis which is for nose moves from side to side, and finally, the Roll axis which aids a circular (clockwise or anticlockwise) movement of the body as it moves forward.
The surfaces of a plane and the fins of a fish have a similar function. They serve to adjust the object's attitude as it moves through the fluid, submarines face the same dynamic control problems as fish do [1].

The platform can move in two planes (side to side, up and down), and can also rotate around two axis (horizontal, and vertical), so there are four options for changing direction of the motion.

The controllers were implemented using selected hardware which shows satisfactory results.

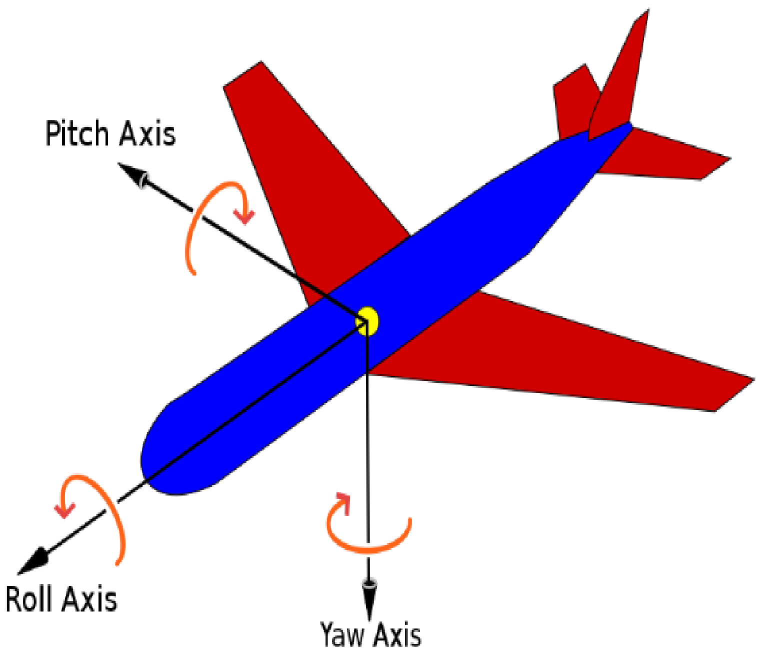


Figure 1: The position of all three axes, with the right-hand rule For describing the angle of its rotations

2. The Proposed System

The proposed system as seen in the Figure 2 consists of four main parts, two axis platform, two motors with drivers, a power supply, and wires for connections, and a programming unit.

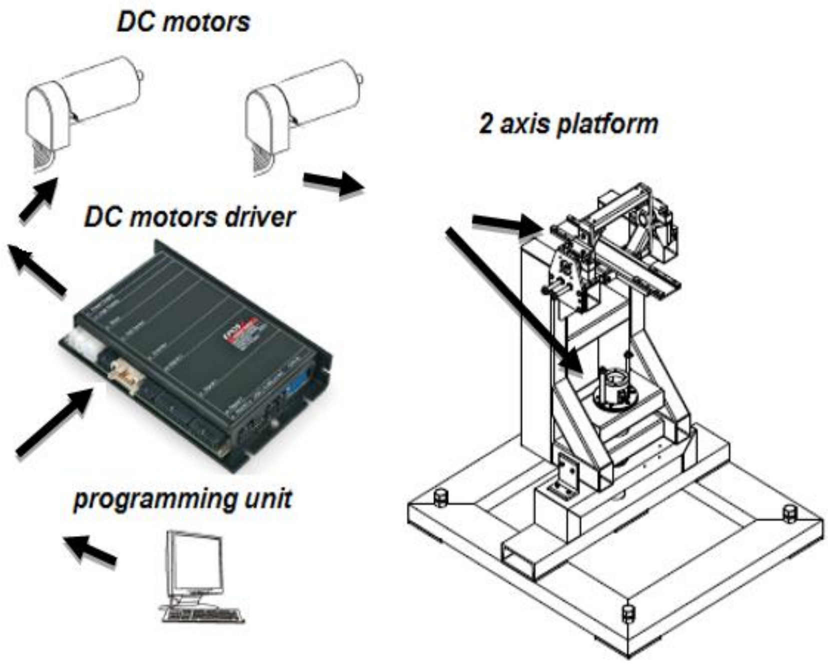


Figure 2: System Assembler

The real implementation of two axes platform is shown in Figure 3

Figure 3: Two axes platform with motors

A two brushless DC motors MAXON type were used as seen in Figure (4) to carry out the required movement and they will be controlled by motors driver **EPOS 7010**.

Figure 4: Brushless DC motors MSXON type

3. Controller Design

3.1 Integral controllers

A closed loop control system is one that determines a difference in the desired and actual condition (the error) and creates a correction control command to remove this error, one form of controller widely used in industrial process control is called three terms, or PID controller, PID control demonstrates three ways of looking at this error and correcting it as shown in Figure 5.

The first way is the (P of PID), the proportional term, this term represents, the bigger the error, the bigger the correction [3].
The (I in PID) is the second way which is for the integral of the error over time. The integral term produces a correction that considers the time where the error has been present. Stated in other words, the longer the error continues, the bigger the correction. Lastly, the (D in PID) way which is stands for derivative. In the derivative term, the corrective action is related to the derivative or change of the error with respect to time. [3]

In other words, the faster the error is changing, the bigger the correction. Control systems can use P, PI, PD, or PID in creating corrective actions. The problem generally is "tuning" the system by selecting the proper values in the terms [3].

The transfer function of the controller is:

$$\boldsymbol{Gc(s) = Kp + \frac{Ki}{s} + sKd} \quad (1)$$

$$\boldsymbol{Gc(s) = \frac{Kds^2 + Kps + Ki}{s}} \quad (2)$$

The controller provides a proportional term, an integration term, and a derivative term. The equation for the output in the time domain is:

$$u(t) = Kpe(t) + Ki\int e(t)dt + Kd\frac{de(t)}{dt} \quad (3)$$

The three-mode controller is also called a PID controller because it contains a proportional, an integral, and a derivative term [2].

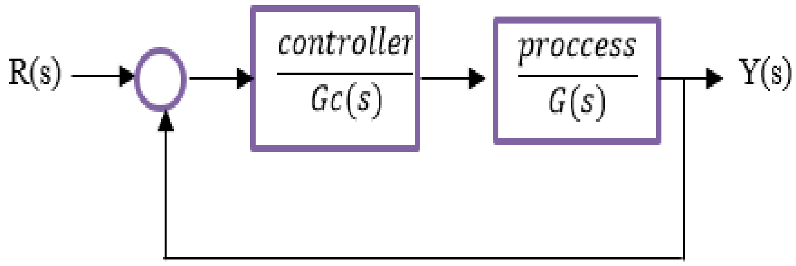


Figure 5: Closed-loop systems with a controller

3.2DC motor modeling and PID controller

3.2.1 Physical setup

A common actuator [6] in control systems is the DC motor which directly provides rotary, the electric equivalent circuit of the armature and the free-body diagram of the rotor are shown in Figure 6.

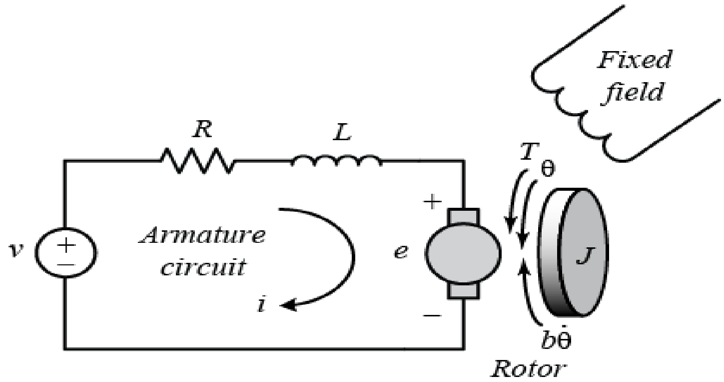


Figure 6: The electric DC motor equivalent circuit of the armature and the free-body diagram of the rotor

3.2.2 Plant to be controlled

The plant to be controlled as illustrated in Figure (7) is the actual DC motor base assembly with a simulated inertial load platform. The simulated moment of inertia is less than the actual DC motor moment of inertia. The effective gear ratio is 93:1 (from the motor armature shaft to the actual load) [6].

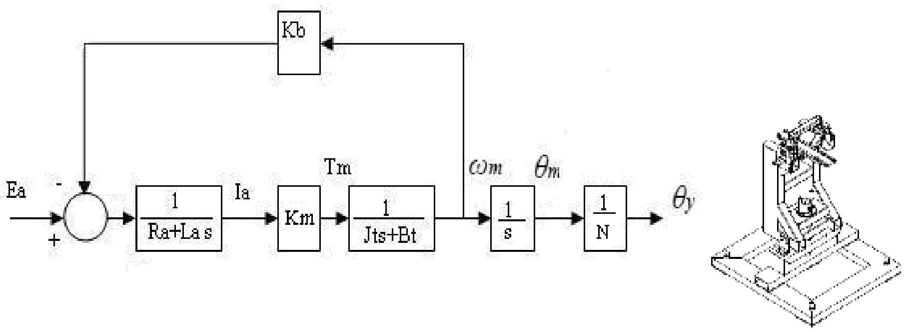


Figure 7: Block diagram of integrated model for (YAW or PITCH axis rotation)

The sensor used is a 500--line optical encoder. It is directly mounted to the armature shaft, so that it rotates at the armature speed. Either of the two encoder signals, A~or~B can be used to measure velocity. Since there are 500 pulses per armature revolution, the number of pulses per load revolution is (93 shaft rev./1 load rev.) * (500 pulses /shaft rev.) [5].

The open loop transfer function of single joint relating the angular displacement to the applied voltage of the joint platform is as follows [6]:

$$Gs = \frac{\theta y}{Ea} = \frac{Km}{Ns[Ra(Jts + b) + KbKm\, l} \qquad (4)$$

SO:

$$Gs = \frac{\theta y}{Ea} = \frac{Km}{Ns[sRaJt + Rab + KbKm]} \qquad (5)$$

$$Gs = \frac{Km}{Ns[Ra(Js + b) + KbKm]} \qquad (6)$$

$$Gs = \frac{Km\,/(Ra + KbKm)}{N\,s(\tau 1\, s\; +\; 1)} \qquad (7)$$

$$\tau 1 = \frac{RaJ}{Rab + KbKm} \qquad (8)$$

And the relation between the angular velocity *ωm(s)* to the armature voltage *Ea*is given by:

$$\frac{\omega m(s)}{Ea(s)} = \frac{Km}{[s\,Ra\,Jt\; +\; Ra\,b +\; KbKm]} \qquad (9)$$

Where:

Km= (Nm/A) torque constant
R= (ohm) resistance
L= (Hennery) inductance
*Kb= (rad*sec/V) back emf*
N= gear ratio

The transfer function of the Maxon Motor can be derived using data from Characteristics of brushless dc motor in the reference [5]

4. Controller Architecture

The EPOS controller architecture furthermore explained will be mapping of internal controller parameters to controller parameters in SI units, and vice versa. In addition to PID position regulation, are described.

The current control loop is used in all operation modes. In the position and velocity based modes there is also a superior position or velocity controller used.

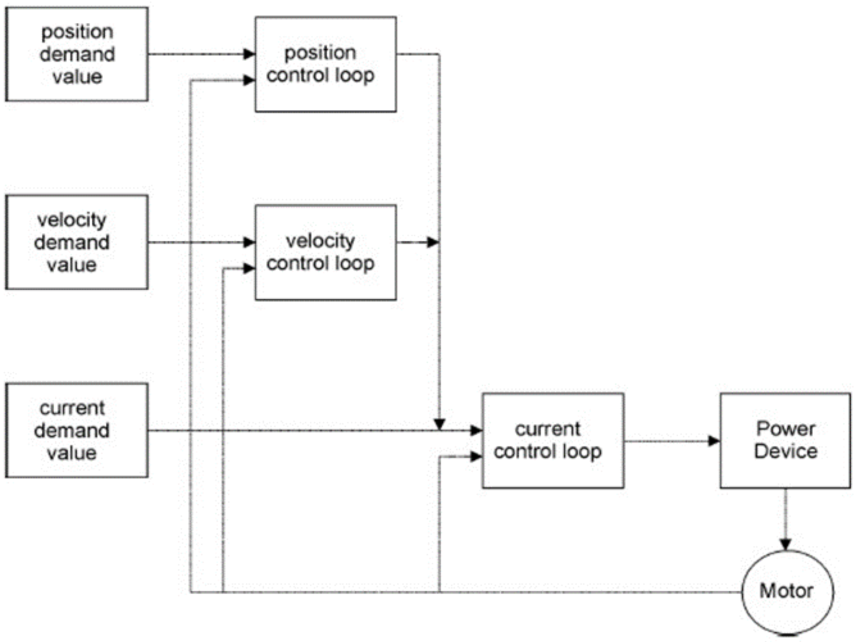


Figure 8: Controller Architecture

5. Plant Model using MATLAB Simulink.

In general, the mathematical equations representing a given system that serves as the basis for a Simulink model can be derived from physical laws. In this paper we will demonstrate how to derive a mathematical model and then implement that model in Simulink and how to employ Simulink to design and simulate the PID controller for a system [7]. The Block Diagram for plant using MATLAB Simulink is shown in Figure 9.

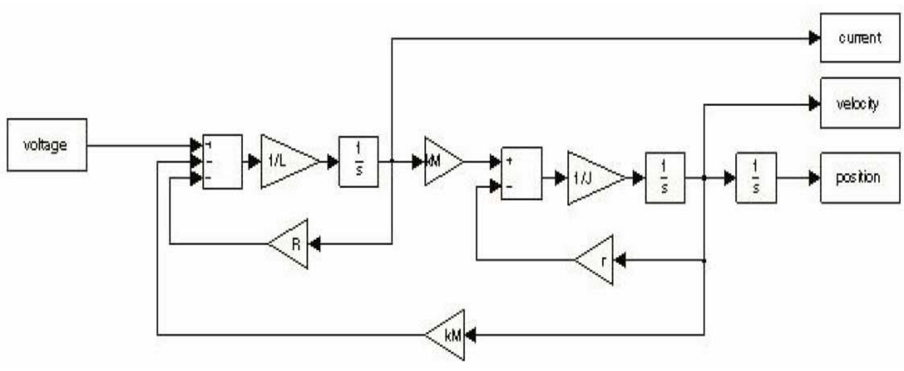


Figure 9: Block Diagram for plant using MATLAB Simulink

K_m = *147 mNm / A*

$J = J_{Motor} + J_{load}$

$$r = \frac{KmIo}{\frac{n0\ 2\pi\ rad}{1} \times \frac{1min}{60s}}$$

$$r = \frac{44.7mNm}{324.6\ rad^2} = 137.7\mu Nm\ /\ rad\ /s$$

6. Current Controller regulation

The plant is connected to the PI current controller.

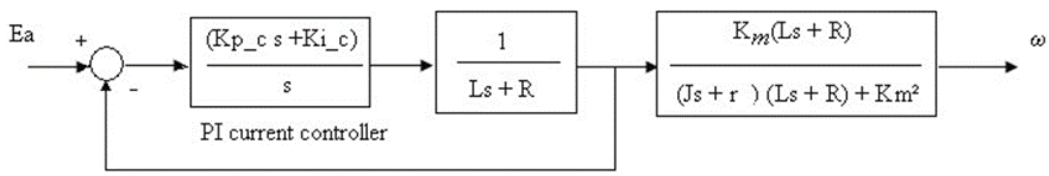


Figure 10: Current Regulation, Block Model

Current controller parameters can be used in analytical calculations, respectively numerical simulations via transfer function:

Ccurrent = *kp + ki / s*

We are used Ziegler-Nichols tuning methods to tuning PI current controller is:

$Kp\_c = 16\,\Omega$, $Ki\_c = 31.28\,k\,\Omega/s$

While:

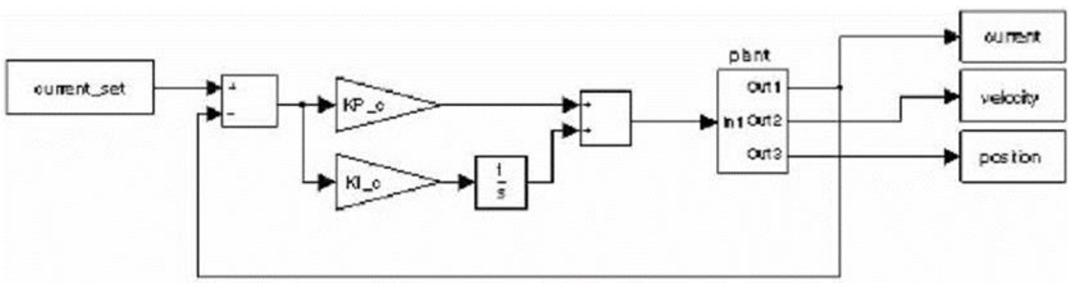


Figure 11: PI current controller block diagram

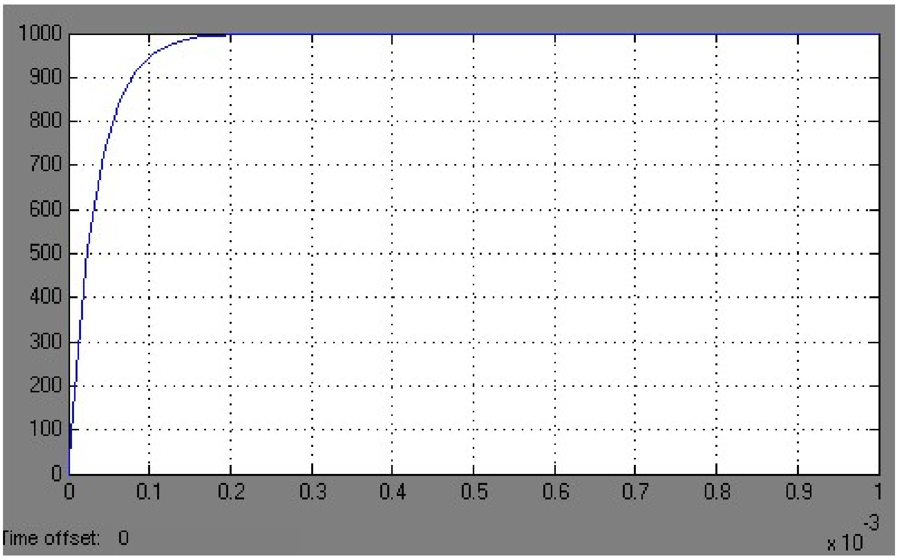


Figure 12: Current Regulation, simulated

7. Velocity Controller regulation

Designing a PI Compensator Using the Ziegler-Nichols Tuning Algorithm method.

$$G(s) = \frac{(Kp\_c\, s + Ki\_c) * Km\, (Ls + R)}{s\,[(Ls + R)\,[(Js + r)\,(Ls + R) + Km^2]]}$$

The PI velocity controller is connected to current regulation.

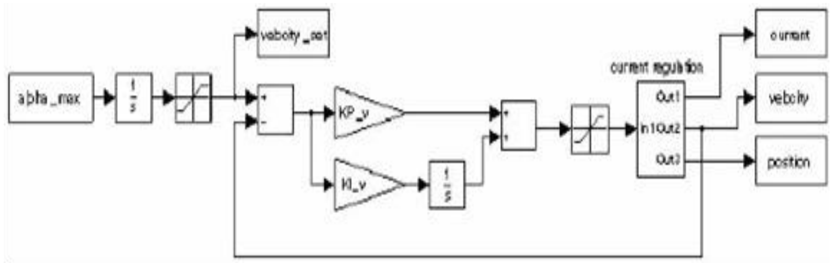


Figure 13: Velocity Regulations, Block Model

The PI velocity controller is:

$$Kp\_v = 100 \frac{mA}{(rad/s)} \quad ; \quad Ki\_v = 0.4 \frac{A/s}{(rad/s)}$$

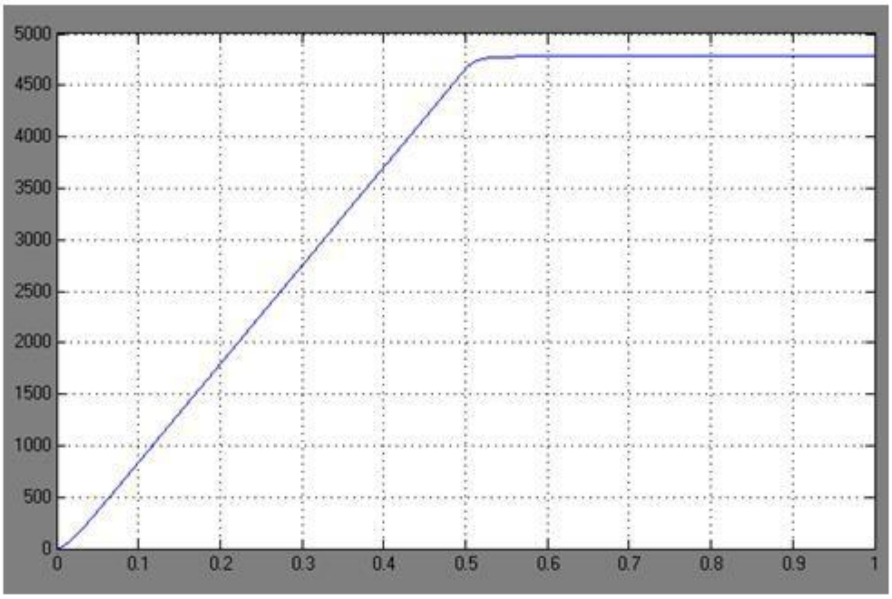


Figure 14: Velocity Regulation, simulated

8. Results and Discussion

Whereas the system Input is the voltage at the motor winding and the system outputs are current, velocity or position.

This was the result of the mathematical process of the integrated system (the platform, motors and system feedback). After the simulation process with the Simulink MATLAB program, an angular velocity is accrued which will be tested as in the Figure 15 by step response.

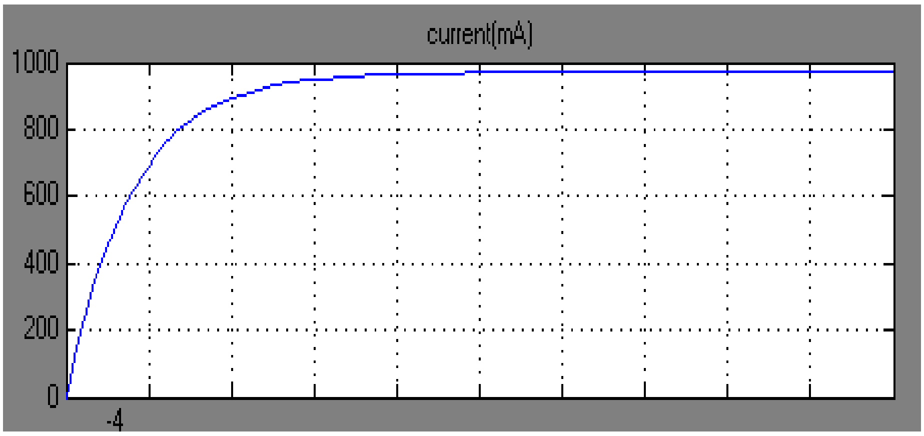


Figure 15: Open loop step response

In this system have three curves we observe current, velocity and position, observe the current curve was stable [5].

The response of the closed loop velocity control system to changes in gain agreed with theory, the closed loop system responds well to changes in load, the amount of error between the input command velocity and the motor velocity can be reduced by increasing the gain; this is what will be discussed in another research paper.

9. Conclusion and Future Work

This paper is to develop a model of the two axis platform, design axis control schemes for (current, velocity, position) using classical approach. The controllers were implemented using selected hardware which shows satisfactory results.

A model for the actuator motor and the platform axis load was developed. An electro-mechanical model of DC motor key control parameters were identified to develop theoretical and build MATLAB/Simulink model of DC motor. A transfer function for each axis actuator motor was developed.

Our work has a number of limitations, which we propose to address in future work:

- Using EPOS software to operate EPOS EC motor controller instead of using our designed MATLAB program.
- Carry on Comparative performance analysis between auto integral and manual tuning integral.

References

[1] R.C. Nelson, "Flight Stability and Automatic Control", McGraw Hill, Second Edition, 1998.

[2] LI Wei1, DUAN Cuifang, HUA Weijuan. Design of the Multi-axis Motion Control System [J]. Agricultural Science & Technology and Equipment, 2010.

[3] Nakamura, M. (Masatoshi), S. Gotō, and N. Kyūra. Mechatronic Servo System Control : Problems in Industries and Their Theoretical Solutions. Berlin ; Springer, 2004. Print.

[4] P.Pillay ad R.Krishnan, "Modeling, Simulation and Analysis of a Permanent Magnet Brushless DC motor drive part II: The brushless DC motor drive," IEEE Transactions on Industry application, Vol.25, May/Apr 1989.

[5] Documentation of Maxon EC motor, May 2010 edition, EC 60 Ø 60 mm, brushless, 400 Watt .

[6] Benjamin C. kuo, Automatic control systems Seventh edition.

[7] K. Ang, G. Chong, and Y. Li, "PID control system analysis, design and technology," IEEE Trans.Control System Technology, vol. 13, pp. 559-576, July 2005. (Pubitemid 41005115)

6

Subgroups of Diagram Group Using Covering Methods

Subgroups of Diagram Group Using Covering Methods

K. Alaswad
Gharyan University, Libya
Kalthom.alaswed@gu.edu.ly.com

N. Ghroda
Gharyan University, Libya
nassraddin2010@gmail.com

Abstract

Diagram groups are created from geometric objects named semigroup presentation S. These diagrams are drawn and considered as 2-complexes. The aim of this article is to determine the covering complex for diagram groups over union of two semigroup presentations by adding some initial generators and relations to alphabet and a set of relation correspondingly. The main aim is to study subgroups of diagram group, we present a method of producing normal subgroups of one generator. In addition, we design a new method for computing all generators and relations for the fundamental group $\pi_1(K({}^3S_1 \cup {}^3S_2, W)$.

Keywords: Diagram groups, Semigroup presentation, Generators, Relation.

1. Introduction

In [7] we obtained the connected 2-complex space ${}^{3}K_i$ that were obtained from ${}^{3}S =< a, b, c: a = b, b = c, c = a >$, also we prove that the connected 2-complex graph ${}^{3}K_{i+1}$ is the covering complex for ${}^{3}K_i$ for all $i \geq 2$.

In this paper we want to determine the semigroup presentation of union of two semigroup presentations by adding a relation.

Let ${}^{3}S_1 =< a, b, c: a = b, b = c, c = a >$, ${}^{3}S_2 =< x, y, z: x = y, y = z, z = x >$ and ${}^{3}S =< a, b, c, x, y, z: a = b, b = c, c = a, x = y, y = z, z = x, x = a >$ be semigroup presentations. Now, we consider the semigroup presentation ${}^{3}S$ obtained from union of initial generators and relations ${}^{3}S_1$ and ${}^{3}S_2$ by adding a relation $x = a$ (for more details see [3], and [5]).

In fact, Guba and Sapir (1997) have shown that if $S_1 =< X_1: r_1 >$, $S_2 =< X_2: r_2 >$ and $S =< X_1 \cup X_2: r_1 \cup r_2 >$are semigroup presentations, then for $W_1, W_2 \in D(S, W_1W_2)$is isomorphic to the direct product of$D(S, W_1)$ and $D(S, W_2)$. Also, they proved if one considers$S =< X_1 \cup X_2: r_1 \cup r_2 \cup \{W_1 = W_2\} >$ where $X_1, X_2$are disjoint sets, and the congruence class ofW_1modulesS_1does not contain words of the form YW_iZ, where Y, Zare words over$X_1, X_2$and YZ are not empty. Then $D(S, W_i)$is isomorphic to the free product$D(S_1, W_i)$and$D(S_2, W_i)$.

In section 2, we will determine the connected 2-complex space ${}^{3}K_i, i \in N$obtained from semigroup presentation ${}^{3}S =< a, b, c, x, y, z: a = b, b = c, c = a, x = y, y = z, z = x, a = x >$.

In section 3, we will compute the generators and relations for the fundamental group $\pi_1(K({}^{3}S_1 \cup {}^{3}S_2, W)$.

2. Determining the connected 2-complex graphs

Let $^3S =< a, b, c, x, y, z: a = b, b = c, c = a, x = y, y = z, z = x, a = x >$ be a semigroup presentation. Associated with semigroup presentation $S =< X: r >$ we have a connected 2-complex graph $^3K_i, i \in N$ where the vertices are word on set of alphabet X and the edges are the form $e = (W_1, R_\varepsilon \to R_{-\varepsilon}, W_2)$such that$\iota(e) = W_1R_\varepsilon W_2$, $\tau(e) = W_1R_{-\varepsilon}W_2$.

The connected 2-complex obtained from$S$is collections of subgraphs$^3K_i, i \in N$. Note that the 2-complexgraph$^3K_i(^3S_1)$obtained from semigroup presentation $^3S_1$ is just collection of subgraphs $^3K_i(^3S_1)$where $^3K_i(^3S_1)$ contains all vertices of length n and respective edges. Similar we obtain $^3K_i(^3S_2)$ from semigroup presentation $^3S_2$for the graph $^3K_i(^3S_2)$. Now for semigroup presentation ^3S, the graph $^3K_i(^3S) =^3K_i(^3S_1) \cup ^3K_n(^3S_2) \cup \{(u, a \to x, v\}$ such that the length $uv = n - 1$. IfW_n is a vertex in $^3K_n(^3S)$then W_nh, $h \in \{a, b, c, x, y, z\}$is a vertex in $^3K_{n+1}(^3S)$. Similarly, if $(u, R_\varepsilon \to R_{-\varepsilon}, v)$ is edge in $^3K_n(^3S)$, then $(u, R_\varepsilon \to R_{-\varepsilon}, vh)$is the respective edge in $^3K_{i+1}(^3S)$. Thus $^3K_{i+1}(^3S)$ is just six copies of $^3K_i(^3S)$togetherwithsixverticesandedges$(u, R_\varepsilon \to R_{-\varepsilon}, vh)$, $h \in \{a, b, c, x, y, z\}$.

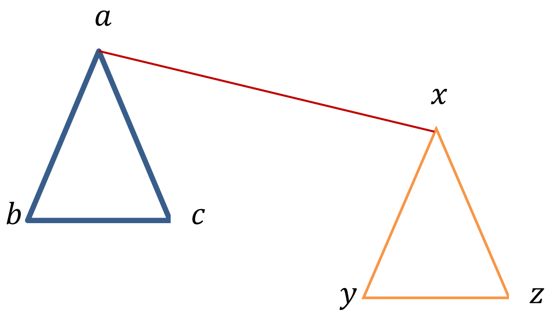


Figure 1: The connected 2-complex graph$^3K_1(^3S)$

While ${}^{3}K_2({}^{3}S)$is

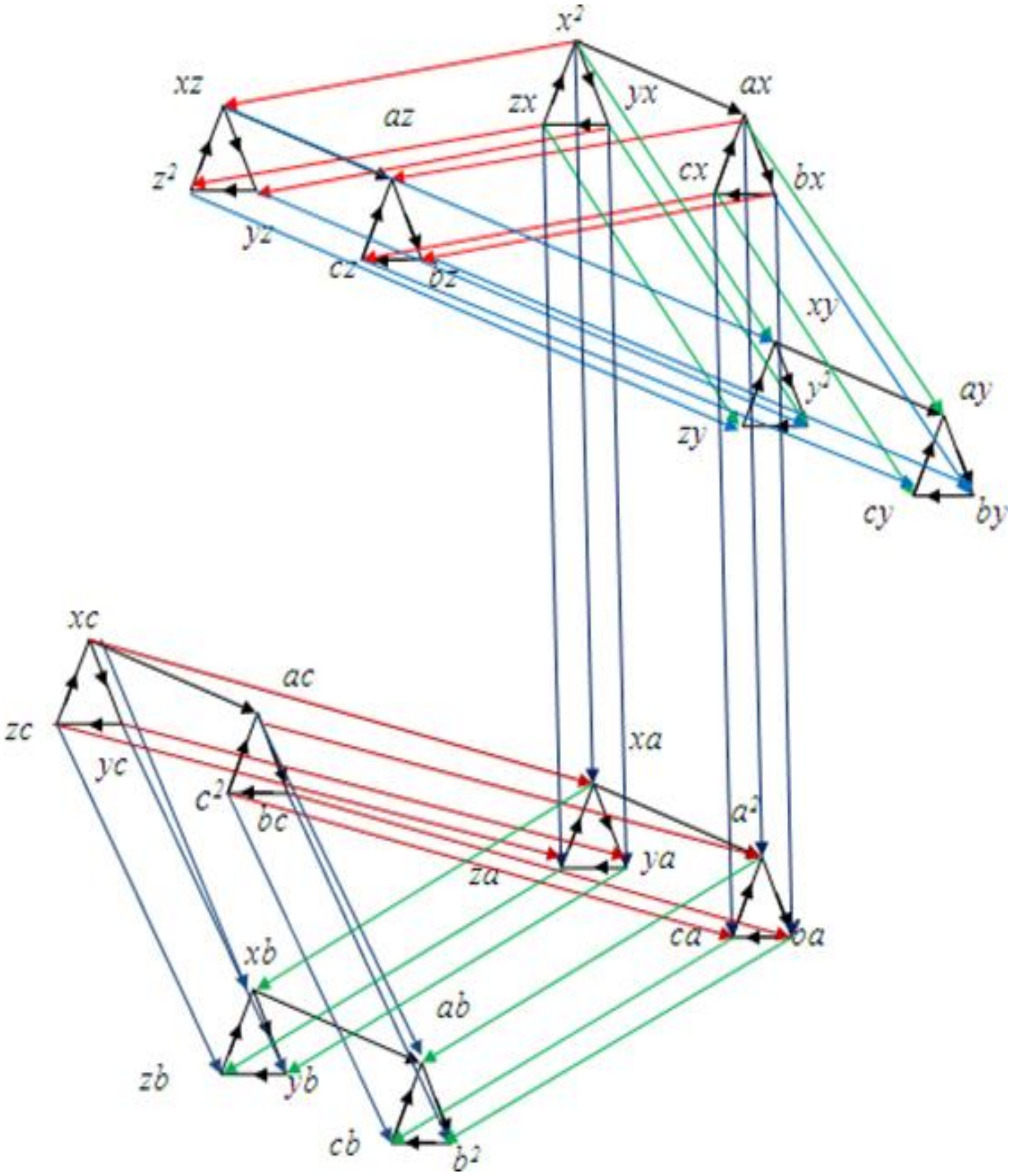


Figure 2: The connected 2-complex graph${}^{3}K_2({}^{3}S)$

Note that ${}^{3}K_2({}^{3}S)$is six copies of ${}^{3}K_1({}^{3}S)$and each vertex in each copy are joinedtogether respectively. Likewise with six copies of ${}^{3}K_3({}^{3}S)$, the 2-complex graph ${}^{3}K_2({}^{3}S)$may be obtained by repeating similarprocedureswithresult ${}^{3}K_4({}^{3}S)$and so on.

Observation2.1 A connected 2—complex graph${}^{3}K_i({}^{3}S)$contains 6^i vertices.

Observation2.2 A connected 2 -complex graph ${}^{3}K_{n+1}({}^{3}S)$is six copies of ${}^{3}K_i({}^{3}S)$.Thus,if there is $e_{l,m}$edges in ${}^{3}K_i({}^{3}S)$then the number of edges in ${}^{3}K_{i+1}({}^{3}S)$ is $6e_{l,m}$plus all edges between triangles in${}^{3}K_{i+1}({}^{3}S)$,which is $e = 6e_{i-1} + 6^i + 6^{i-1}$.

Observation2.3 Vertices$U$and$V$are connected if and only if $L(U) = L(V)$.

Lemma2.4 If$L(U) = L(V)$then$\pi_1({}^{3}K_i({}^{3}S),U) = \pi_1({}^{3}K_i({}^{3}S),V)$.

Lemma2.5 Vertices of ${}^{3}K_i({}^{3}S)$are all words of length i.

3. Subgroups of Diagram Group Using Covering Methods

Let W be a positive word on ${}^{3}S$. When the length of W equal to one, then we have 6^1possibilities vertices in the connected 2-complex graph ${}^{3}K_1({}^{3}S)$namely x, y, z, a, b, c as is shown Figure 1.

Definition 3.1

A 2-complex graph ${}^{3}K_H$ contains the following:

i. Vertices: The set of right cosets $H[\alpha]$of H, where $[\alpha] \in P_v, v \in V$.

ii. Edges: All ordered pairs $(H[\alpha], x^{\varepsilon})$where x is an edge in K and $\varepsilon = \pm 1$.

iii. Functions:

a. $i(H[\alpha], x^{\varepsilon}) = H[\alpha]$.

b. $\tau(H[\alpha], x^{\varepsilon}) = H[\alpha x^{\varepsilon}]$.

c. $(H[\alpha], x^{\varepsilon})^{-1} = (H[\alpha x^{\varepsilon}], x^{-\varepsilon})$.

Theorem 3.2 (Rotman 1995,2002)

The component ${}^{3}K_{H}$ is connected 2-complex graph.

Theorem 3.3 (Rotman 1995,2002)

The map φ_H:${}^{3}K_H({}^{3}S) \rightarrow {}^{3}K({}^{3}S)$, $\varphi_H(H[\alpha]) = v$, $\varphi_H(H[\alpha], x) = x$ is a mapping of connected 2-complex graphs.

Theorem 3.4 (Rotman 1995,2002)

The map φ_H:${}^{3}K_H({}^{3}S) \rightarrow {}^{3}K({}^{3}S)$, $\varphi_H(H[\alpha]) = v$, $\varphi_H(H[\alpha], x) = x$isa locally bijective.

Theorem 3.5 (Rotman 1995,2002)

The mapping $\varphi_H^{*}: \pi_1({}^{3}K_H({}^{3}S), v') \rightarrow \pi_1({}^{3}K({}^{3}S), v)$ is an injective if φ_H^{*}isa locally bijective.

Theorem 3.6

Consider the following connected 2-complex graph${}^{3}K_1({}^{3}S)$ as shown in Figure 1, such that $G = \pi_1({}^{3}K_1({}^{3}S), a)$ contains δ_1, where $\delta_1 =< e_{a,b}e_{b,c}e_{c,a} >$. If $H_{1_2}$is the smallest normal subgroup of Gcontaining $< \delta_1^2 >$, then the covering complex ${}^{3}K_{H_{1_2}}({}^{3}S)$ for${}^{3}K_1({}^{3}S)$is two hexagons' shapes.

Proof: Use the notion of $H[\gamma] = H[\beta] \Leftrightarrow [\gamma\beta^{-1}] \in H$. From ${}^3K_1({}^3S)$, $\pi_1({}^3K_1({}^3S))$ can be obtained. Fix a vertex a in ${}^3K_1({}^3S)$. Now, for any normal subgroup of$\pi_1({}^3K_1({}^3S), a)$, there exists a unique covering space, start by choosing basic $H[\alpha]$ where α is a path such that $i(\alpha) = a, \tau(\alpha) = v$ for every vertex $v$in ${}^3K_1({}^3S)$. As a result, these basic $H[1], H[e_{a,b}]$, and $H[e_{a,b}e_{b,c}]$ will be selected, and then all possible edges, vertices, can be determined as shown in Table 1 and Table 2.

Table 1: Vertices in ${}^3K_1({}^3S)$ and ${}^3K_{H_{1_2}}({}^3S)$

Vertices in ${}^3K_1({}^3S)$	Vertices in ${}^3K_{H_{1_2}}({}^3S)$
a	$H[1]$
b	$H[e_{a,b}]$
c	$H[e_{a,b}e_{b,c}]$
a	$H[e_{a,b}e_{b,c}e_{c,a}]$
b	$H[e_{a,b}e_{b,c}e_{c,a}e_{a,b}]$
c	$H[e_{a,b}e_{b,c}e_{c,a}e_{a,b}e_{b,c}]$
x	$H[e_{a,x}]$
y	$H[e_{a,x}e_{x,y}]$
z	$H[e_{a,x}e_{x,y}e_{y,z}]$
x	$H[e_{a,x}e_{x,y}e_{y,z}e_{z,x}]$
y	$H[e_{a,x}e_{x,y}e_{y,z}e_{z,x}e_{x,y}]$
z	$H[e_{a,x}e_{x,y}e_{y,z}e_{z,x}e_{x,y}e_{y,z}]$

Table 2: Edges in $^3K_1(^3S)$ and $^3K_{H_{1_2}}(^3S)$

Edges in $^3K_1(^3S)$	Edges in $^3K_{H_{1_2}}(^3S)$
$e_{a,b}$	$(H[1], e_{a,b})$
$e_{a,b}e_{b,c}$	$(H[e_{a,b}], e_{a,b}e_{b,c})$
$e_{a,b}e_{b,c}e_{c,a}$	$(H[e_{a,b}e_{b,c}], e_{a,b}e_{b,c}e_{c,a})$
$e_{a,b}$	$(H[e_{a,b}e_{b,c}e_{c,a}], e_{a,b}e_{b,c}e_{c,a}e_{a,b})$
$e_{a,b}e_{b,c}$	$(H[e_{a,b}e_{b,c}e_{c,a}e_{a,b}], e_{a,b}e_{b,c}e_{c,a}e_{a,b}e_{b,c})$
$e_{a,b}e_{b,c}e_{c,a}$	$(H[1], e_{a,b}e_{b,c}e_{c,a}e_{a,b}e_{b,c})$
$e_{a,x}$	$(H[1], e_{a,x})$
$e_{a,x}e_{x,y}$	$(H[e_{a,x}], e_{a,x}e_{x,y})$
$e_{a,x}e_{x,y}e_{y,z}$	$(H[e_{a,x}e_{x,y}], e_{a,x}e_{x,y}e_{y,z})$
$e_{a,x}e_{x,y}e_{y,z}e_{z,x}$	$(H[e_{a,x}e_{x,y}e_{y,z}], e_{a,x}e_{x,y}e_{y,z}e_{z,x})$
$e_{a,x}$	$(H[e_{a,x}e_{x,y}e_{y,z}e_{z,x})], e_{a,x}e_{x,y}e_{y,z}e_{z,x}e_{a,x})$
$e_{a,x}e_{x,y}$	$(H[e_{a,x}e_{x,y}e_{y,z}e_{z,x}e_{a,x})], e_{a,x}e_{x,y}e_{y,z}e_{z,x}e_{a,x}e_{x}$
$e_{a,x}e_{x,y}e_{y,z}$	$(H[1], e_{a,x}e_{x,y}e_{y,z}e_{z,x}e_{a,x}e_{x,y}e_{y,z})$

Now, suppose φ_H:$^3K_{H_{1_2}}(^3S) \rightarrow ^3K_1(^3S)$defined by $\varphi_H(H[1]) = a, \varphi_H[e_{a,b}] = b$,

$\varphi_H(H[1], e_{a,b}) = e_{a,b}$. This map can be viewed as a locally bijective. For this reason, $^3K_{H_{1_2}}(^3S)$is a covering graph for$^3K_1(^3S)$and it is of two hexagons shapes. Therefore, the covering space $^3K_{H_{1_2}}(^3S)$ for $^3K_1(^3S)$ in this case is of two hexagons shapes.

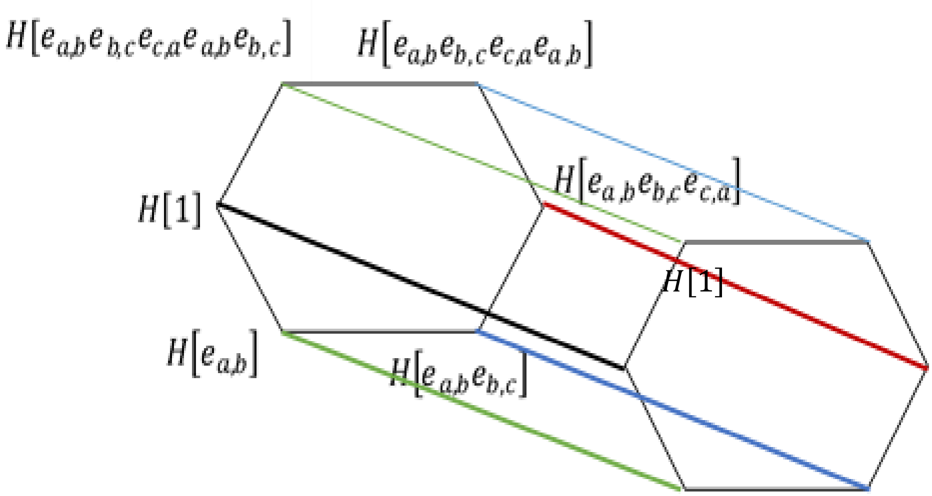


Figure 3: The connected 2-complex graph $^3K_{H_{1_2}}(^3S)$

Since a is a vertex of the connected 2-complex graph $^3K_2(^3S)$, and $H[1]$ lies over a, then by theorem 3.5 $\varphi_H^*: (\pi_1(^3K_{H_{1_2}}(^3S), H[1])) \rightarrow Im\varphi_H^* = H$. As a result, $H = \pi_1(^3K_{H_{1_2}}(^3S), H[1])$ can be obtained as a subgroup of $G = \pi_1(^3K_1(^3S), a)$.

The generators for $\pi_1(^3K_{H_{1_2}}(^3S), H[1])$ are computed here using maximal subtree methods, select $T(^3K_{H_{1_2}}(^3S))$ for $^3K_{H_{1_2}}(^3S)$ (see Figure 4).

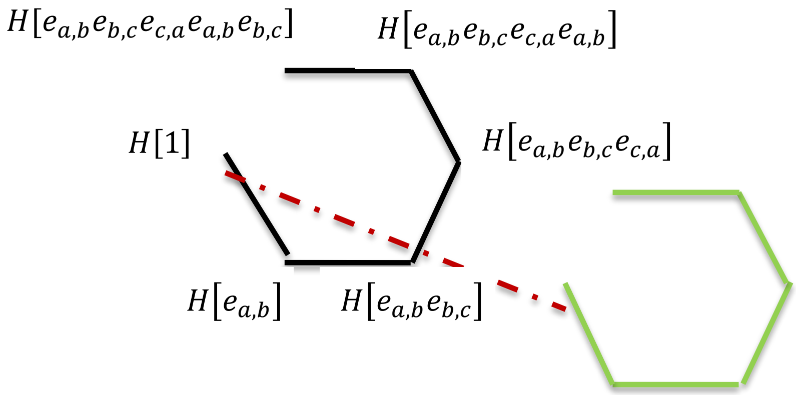


Figure 4: The maximal tree $T({}^3K_{H_{1_2}}({}^3S))$

The generators for the fundamental group $\pi_1({}^3K_{H_{1_2}}({}^3S), H[1])$ will be:

1. $g_1({}^3K_{H_{1_2}}({}^3S)) =$
   $(H[1], e_{a,b})(H[e_{a,b}], e_{a,b}e_{b,c})(H[e_{a,b}e_{b,c}], e_{a,b}e_{b,c}e_{c,a})$

2. $(H[e_{a,b}e_{b,c}e_{c,a}], e_{a,b}e_{b,c}e_{c,a}e_{a,b})$

$(H[e_{a,b}e_{b,c}e_{c,a}e_{a,b}], e_{a,b}e_{b,c}e_{c,a}e_{a,b}e_{b,c})(H[1], e_{a,b}e_{b,c}e_{c,a}e_{a,b}e_{b,c})^{-1}.$

3. $g_2({}^3K_{H_{1_2}}({}^3S)) = (H[1], e_{a,b})(H[1], e_{a,x})(H[e_{a,x}], e_{a,x}e_{x,y})$

$(H[e_{a,x}e_{x,y}], e_{a,x}e_{x,y}e_{y,z})(H[e_{a,x}e_{x,y}e_{y,z}], e_{a,x}e_{x,y}e_{y,z}e_{z,x})$

$(H[e_{a,x}e_{x,y}e_{y,z}e_{z,x})], e_{a,x}e_{x,y}e_{y,z}e_{z,x}e_{a,x})$

$(H[e_{a,x}e_{x,y}e_{y,z}e_{z,x}e_{a,x})], e_{a,x}e_{x,y}e_{y,z}e_{z,x}e_{a,x}e_{x,y})(H[1], e_{a,x}e_{x,y}e_{y,z}e_{z,x}e_{a,x}e_{x,y}e_{y,z})^{-1}.$

References

[1] A.G.B.Ahmad and A.M.AlOdhari. The graph of diagram groups constructed from natural numbers semigroup with a repeating generator.Jour.ofInst.ofMath&Com.Sci.(Math.Ser.) 17(2004), 67-69.

[2] A. G. Ahmad 2003. Triviality problem for diagram groups.Jour. Of Inst OfMaths.& Comp. Sci. 16(2)(2003): 105-107.

[3] V.Guba and M. Sapir. Diagram Groups .Memoirs of theAmericanMathematicalSociety, 1997.

[4] V. Guba and M. Sapir and G.N Azhantseva G.N, V.,Metrics on Diagram Groups and Uniform Embeddings in a Hilbert space, Commentarii Mathematici Helvetici, 4: 911-929.

[5] V. Kilibarda, On the Algebra of Semigroup Diagrams, Int. J. of Alg. &Comp.,7(1997):313-338.

[6] J. J. Rotman, An Introduction to the theory of groups. Forth edition. NewYork:Springer–Verlag.1994.

[7] K.Alaswed and Abd Ghafur Bin Ahmad.Covering Space for Diagram Group from Semigroup Presentation, Proceedingofthe5thAsianMathematicalConference, KualaLumpur, 22-26 [1].

[8] Rotman, J. 1995. An Introduction to theory of groups, 2nd edition. New York: Springer- Verlag

[9] Rotman, J. 2002. Advanced Modern Algebra. New Jersey: Pearson Education, In.

7

Using Data Mining by Universities: to Study Student Retention

Using Data Mining by Universities: to Study Student Retention

Soad Algaib
High Institute of Science Technology Yafren
soad_absent@yahoo.com

Abstract

Objective: Student retention is one of the most challenging problems in higher education. It affects university rankings, school reputation, and financial resources. In order to understand and solve the problem, it is very important to know the factors that affect student retention to build a model that predicts students who are at risk of dropping out of college. This can be obtained by using data mining techniques.

Methods: This paper describes a number of case studies that studied student retention using data mining and analysis their results.

Results: The analysis shows that student retention is affected mostly by scholastic performance, such as GPA and SAT, and the most used data mining technique to predict retention model is decision trees. Moreover, using sufficient data with proper variables and using balanced dataset for binary classification help data mining techniques to predict freshman student retention with about 80% precision. Finally, using rich set of features, more data and more variables can help improve the data mining results.

Conclusion: Universities and institutions can use their databases along with data mining techniques to predict student retention. They can build models that predict at risk students, so they can make a plan to retain them. Also, the factors that affect student retention can be monitored by the university and managed in somehow that improves retention. However, the success of data mining studies depends on the quantity and quality of data used to build models.

1. Introduction

Student retention is one of the most challenging problems in higher education. According to the U.S. Department of Education, Center for Educational Statistics, only about 50% get a bachelor degree from all students who start their higher education [4].

Dropping out in higher education is an old problem; Tinto [15] reported national dropout rates and BA degree completions rates from 1880 to 1980, and he found that they were constant from 50% to 60%, except for the World War II period Figure 1. The most important effects of the higher dropout of students in higher education are financial losses, low graduation rates, and bad school reputation; consequently, enrollment management and student retention have the highest attention by universities' administrators in the U.S. and other developed countries [4].

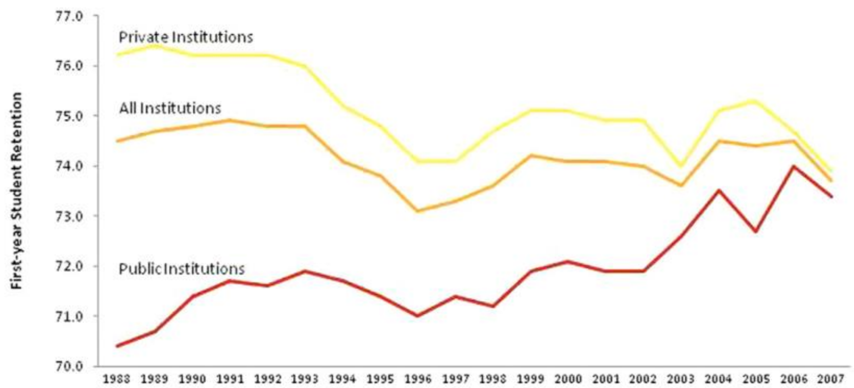
To improve student retention, decision makers should understand the significant reasons that lead students to dropout; moreover, they must identify accurately students who likely will dropout [4].

Figure 1: BA degree completion rates for the period 1880–1980

Higher education organizations need to predict students' paths, and react based on them. Predicting students' academic behaviors can be achieved by data mining [8]. Data mining helps these organizations to take advantage of students' reports in huge datasets to discover invisible patterns that will be used to build models to predict a student's behavior with high correctness [8]. Consequently, data mining enables higher education organizations to use their data resources and students' data in more effective way [8]. Using predictive data mining techniques in higher education would be similar to the use of them in marketing where they have been used for a long time and become very important to the success of this field [4]. For example, data mining can be used in marketing to identify clients who are very likely will leave the company, so the company can take some actions to keep them or the most significant ones [4]; likewise, in higher education, data mining can be used to identify students who are at a high probability of dropping out, so an institution or a university can react to these information [8].

2. Related Work

There are many studies have been done on retention problem. Generally, researchers found that from all the students who drop out of college, most of them do after the first year [3, 6]. Figure 2 shows the rates of first-year students who return for second year at four-year colleges with expected variance between public and private institutions [1]. As a result, most of the retention studies focused on the first year dropouts [13]. Many different methods of data mining have been used for retention prediction. Different models are built based on variant types of data, have different levels of prediction like institution or university, have different percentage of precision, and achieve different levels of success. The next paragraphs describe a number of case studies and their results.



	1988	1989	1990	1991	1992	1993	1994	1995	1996	1997	1998	1999	2000	2001	2002	2003	2004	2005	2006	2007
Private Institutions	76.2	76.4	76.2	76.2	76.2	76.0	75.2	74.8	74.1	74.1	74.7	75.1	75.1	74.9	74.9	74.0	75.1	75.3	74.7	73.9
All Institutions	74.5	74.7	74.8	74.9	74.8	74.8	74.1	73.8	73.1	73.3	73.6	74.2	74.1	74.1	74.0	73.6	74.5	74.4	74.5	73.7
Public Institutions	70.4	70.7	71.4	71.7	71.6	71.9	71.7	71.4	71.0	71.4	71.2	71.9	72.1	71.9	71.9	72.6	73.5	72.7	74.0	73.4

Figure 2: Percentage of first-year students at four-year colleges who return for second year [1]

Druzdzel [5] studied the dropout problem by applying a knowledge discovery algorithm on the US news college ranking data to discover the factors that affect student retention. They found that the most important factor was average test score (Figure 3) and other factors like student-faculty ratio, faculty salary, and university's educational expense did not directly affect student retention; based on that, they recommended that universities should enhance the student selectivity to increase retention.

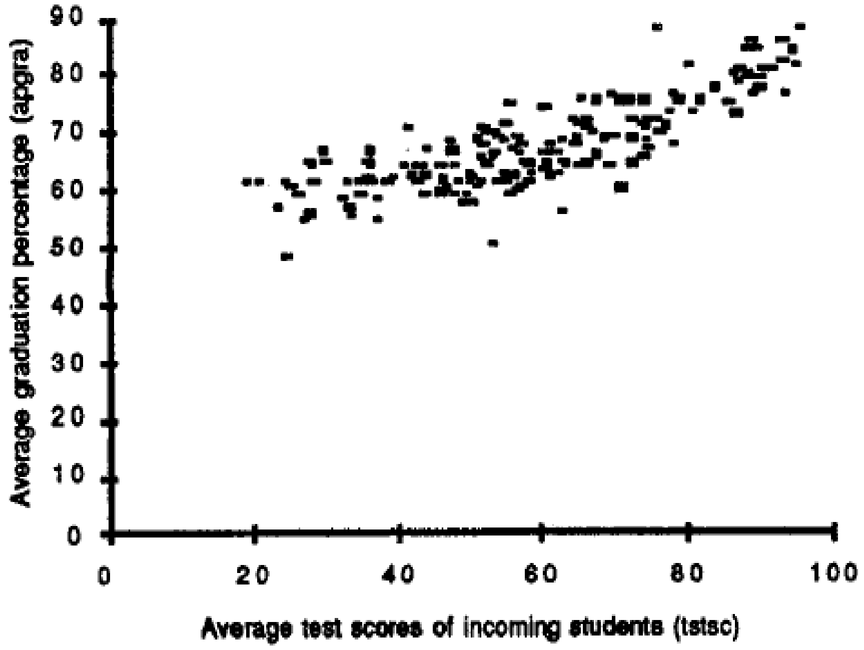


Figure 3: Relation between average test score and average graduation percentage [5]

Sanjeev and Zytkow [10] applied a pattern discovery process to student databases to look for patterns that influence student retention. They found that the GPA of high school is the most important factor that predicts student retention. Also, they found that financial aid does not in retaining students.

Stewart and Levin [11] used data mining and analysis on dataset of students in a community college to identify student characteristics that are associated with student retention. They found that the most important characteristics that would predict a student's retention are the GPA of a student, cumulative hours attempted, and cumulative hours completed. Also, they found that the new students at a higher risk.

Salazar [9] studied graduate student retention in Industrial University of Santander, Colombia, by using clustering and decision tree algorithms. Generally, they found that the higher marks in the pre-university test and the lower age predict the higher probabilities of a good academic performance and persistence.

Superby [12] wanted to classify new students into three groups, which are low risk (the probability of dropping out), medium risk, and high risk by applying discriminant analysis, neural networks, random forests and decision trees to dataset based on a questionnaire aimed to new year students at the University of Belgium. The authors found that the scholastic history of the student and his socio-familial background have the highest correlation to student's success. However, they found that the overall classification rates obtained were not remarkable; they were 40.63% for decision trees, 51.78% for random forests, 51.88% for neural networks, and 51.88% for linear discriminant analysis.

Herzog [7] studied student retention by applying decision trees, neural networks and logistic regression to American College Test's (ACT) student profile section data, NSC data, and the institutional student information system data and comparing the results. Decision trees built using C5.0 gave the highest correct classification rate. For example, C5.0 performed the best with 83% correct classification rate for degree completion time (three years or less) as it is shown in Figure 4.

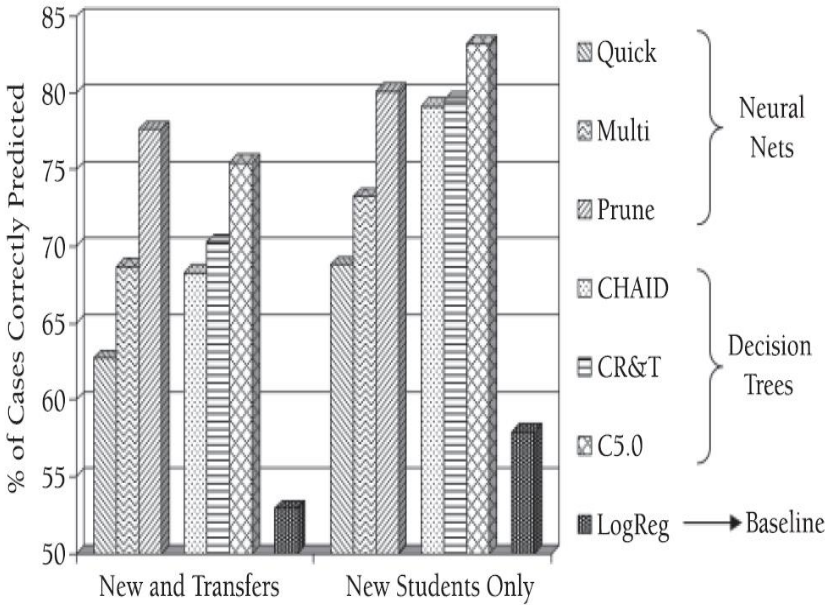


Figure 4: Model comparison for degree completion time: prediction accuracy with validation data [7]

Atwell [2] studied retention by applying data mining techniques to University of Central Florida’s student demographic and survey data. They used nearest neighbor algorithm to impute more than 60% observations that have one or more variables with missing values. They used several modeling techniques, such as logistic regression, neural network, decision trees, and clustering, and they found that these models can identify more than 88% of the students who dropped out in the test data. Also, they concluded that the quality of student learning experience such as High School GPA and SAT is the most significant factor in retention rate.

Delen [4] applied data mining techniques to five years of institutional data to study retention. The author used four classification methods, which are artificial neural networks, decision trees, support vector machines, and logistic regression. He found that support vector machines gave the best overall prediction accuracy with 81.18%, followed by decision trees (80.65%), artificial neural networks (79.85%), and logistic regression (74.26%).

Nandeshwar [14] used data mining to find patterns of student retention at American Universities. The authors applied classification techniques to data from a mid-size public university. In contrast to the last studies, the authors found that it is very difficult to predict first or second year retention. However, they found it is easier to predict third year retention, and the most important attributes affecting third-year retention were student’s wages, parent’s adjusted gross income, student’s adjusted gross income, mother’s income, father’s income, and high school percentile.

3. Analysis

From analyzing the last case studies’ results, we can conclude that the following: the most significant factor that affects student retention is scholastic performance, such as GPA and SAT. Also, the most used data mining technique to predict retention model is decision trees. Moreover, according to [4], data mining techniques are able to predict freshman student retention with about 80% precision but by using sufficient data with proper variables and using balanced dataset for binary classification; certainly, using rich set of features, more data and more variables can help improve the data mining results.

Delen [4] added that it is better to use decision trees because they depict more explicit model structure compared to support vector machines and neural networks [4].

4. Conclusion

Universities and institutions can use their databases along with data mining techniques to predict student retention. They can build models that predict at risk students, so they can make a plan to retain them. Also, the factors that affect student retention can be monitored by the university and managed in somehow that improves retention. However, the success of data mining studies depends on the quantity and quality of data used to build models.

References

[1] ACT National Collegiate Retention and Persistence to Degree Rates.

[2] Atwell, R. H., Ding, W., Ehasz, M., Johnson, S., & Wang, M., 2006. Using data mining techniques to predict student development and retention. *In Proceedings of the National Symposium on Student Retention*.

[3] Deberard, S. M., Julka, G. I., & Deana, L., 2004. Predictors of academic achievement and retention among college freshmen: a longitudinal study. *College Student Journal*, 38(1), pp. 66–81.

[4] Delen, D., 2010. A comparative analysis of machine learning techniques for student retention management. *Decision Support Systems*, 49(4), pp. 498-506.

[5] Druzdzel, M. J., & Glymour, C.,1994. Application of the TETRAD II program to the study of student retention in US colleges. *In Working Notes of the AAAI-94 Workshop on Knowledge Discovery in Databases (KDD-94) Seattle*, WA, pp. 419– 430.

[6] Hermaniwicz, J. C., 2003. College Attrition at American Research Universities: Comparative Case Studies, Agathon Press, New York.

[7] Herzog, S., 2006. Estimating student retention and degree-completion time: Decision trees and neural networks vis–vis regression. *New Directions for Institutional Research*, p. 131.

[8] Luan, J., 2006. Data Mining Applications in Higher Education.

[9] Salazar, A., Gosalbez, J., Bosch, I., Miralles, R., & Vergara, L., 2004. A case study of knowledge discovery on academic

achievement, student desertion and student retention. *In 2nd International Conference on Information Technology: Research and Education*, ITRE 2004, pp. 150–154.

[10] Sanjeev, A., & Zytkow, J., 1995. Discovering enrolment knowledge in university databases. *In First International Conference on Knowledge Discovery and Data Mining,* Montreal, Que., Canada, pp. 246–51.

[11] Stewart, D. L., & Levin, B. H., 2001. A model to marry recruitment and retention: A case study of prototype development in the new administration of justice program at Blue Ridge community college.

[12] Superby, J. F., Vandamme, J. P., & Meskens, N. (2006). Determination of factors influencing the achievement of the first-year university students using data mining methods. *In 8th International Conference on Intelligent Tutoring Systems (ITS 2006),* Jhongli, Taiwan, pp. 37–44.

[13] Thomas, E. H., & Galambos, N., 2004. What satisfies students? Mining student opinion data with regression and decision tree analysis, *Research in Higher Education,* 45 (3), pp. 251–269.

[14] Nandeshwar, Ashutosh & Menzies, Tim & Nelson, Adam. (2011). Learning patterns of university student retention.

[15] Tinto, Vincent. “Limits of Theory and Practice in Student Attrition.” The Journal of Higher Education 53, no. 6 (1982): 687–700. https://doi.org/10.2307/1981525.