International Journal of Mechanical Engineering and Robotics Research

ISSN 2278 – 0149 www.ijmerr.com
Vol. 2, No. 2, April 2013

Research Paper

PERFORMANCE OF A ROBOT TO REDUCE CYCLE TIME ESTIMATION BY C++ PROGRAMMING

Pratik P Gandhi[1]*

*Corresponding Author: **Pratik P Gandhi,** ✉ pratik7g@gmail.com*

In some daily tasks such as Pick and Place or Loading and Unloading application, the Cartesian robot is requested to reach with its end-effectors to a desired target location. Such tasks become more complex if it has to handle multiple points in shortest travelling time and space. It this reason the present study was conducted with the primary objective to develop a computational intelligent system that would contribute towards encouraging a productive and quality way of material handling and processing. The objective of this paper is to optimize the performance of a Cartesian (Gantry) robot to pick hot crown gear in a quenching press machine and to place our Tray Track line pallate board by using end-effectors. In this paper where actual robot perform in an automobile industries, where some distance taken, there Achleine Software was used to perform the Cartesian Robot. But now in this replace on based the C++ programming and Matlab Software. We calculated our actual robot cycle time and estimated new cycle time to increase the productivity and increase the efficiency of industries.

Keywords: Programming, Work envelope, Pick and place movement, Cycle time, Estimation, Travel path

INTRODUCTION

A Cartesian Robot is one or more principal axes of control are linear. They move in a straight line rather than rotate. Among other advantages is that this mechanical arrangement simplifies the robot control arm solution. Cartesian robots are being widely employed in industrial applications such as pick and place application lines that handle a variety of crown gear models. In order to avoid the risk factor in hot crown gear pick and place application, various steps can be taken. One of the prominent method is by substituting the human hands with the robotic arm in handling these dangerous and hazardous environments. It is with these reasons that this study was conducted with the primary objective to design and develop a new low-cost, cycle

[1] Faculty of Engineering, University Malaysia Sarawak (UNIMAS), Kota Samarahan 94300, Sarawak, Malaysia.

time reduce ,high-efficiency Cartesian robotic arm for application such as loading and unloading application. A new evolutionary computation method using Dynamic Programming to control and optimize the system performance in terms of its positioning and speed that would contribute towards encouraging a cycle time reduce, improve the productive and quality process will be developed. This population candidate controller is repeatedly grown according to crossover, mutation and other operators. The competition between different companies regarding price and performance of the Cartesian robot and control system has been the most important motivation. In case of cost saving, cycle time on robotics equipments, the solution is an alternative.

CARTESIAN COORDINATE ROBOT

A robot whose joints travel in right angle lines to each other, there are no radial motions. The profile of its work envelope represents a rectangular shape. And it is also referred to as Gantry Robot (Figures 1a, 1b and 1c).

Figure 1a: Cartesian Robot

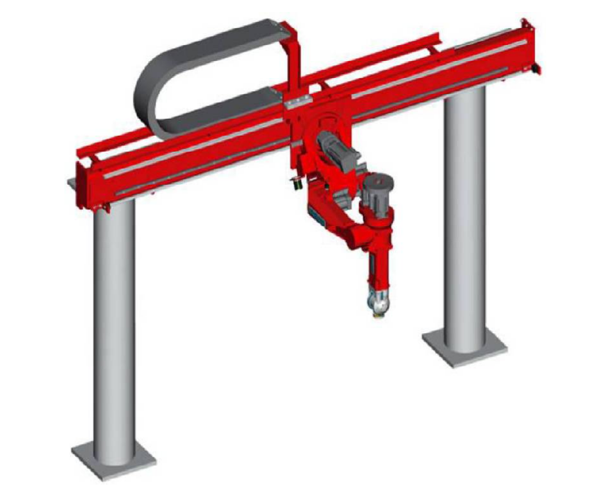
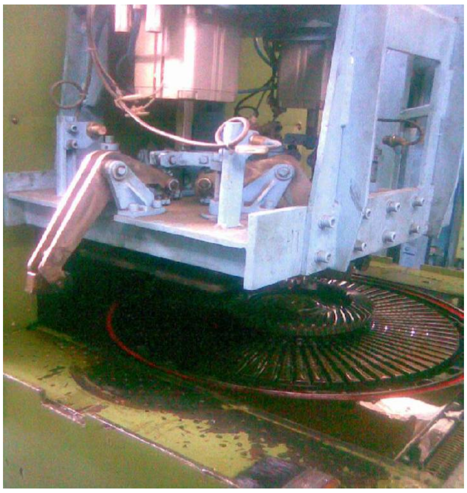


Figure 1b: Project Cartesian Robot

Figure 1c: Pick the Gear by Cartesian Robot



A coordinate system with axes or dimensions that are intersecting and perpendicular (orthogonal). The origin is the intersection of the three coordinates—x, y and z axes—that locate a point in space and measure its distance from any of three intersecting coordinate planes. The coordinates are used to identify points for the positioning of an end-effectors.

CONTROLLED-PATH ROBOT

This robot is taught its motions according to capabilities inherent in point-to-point and continuous-path systems: robot axes need not be specified, while the desired contour, acceleration, and deceleration are automatically generated. Special features of this kind of robot are path computations, programmable velocities, coordinated axis motions, ability to make changes in end-effector length, use of multi-robots, mirror imaging, and software editing and diagnosis.

DEFINITIONS OF KEY TERMS

Cycle Time: "Period required to complete one cycle of an operation; or to complete a function, job, or task from start to finish. Cycle time is used in differentiating total duration of a process from its run time" (Businessdictionary.com, 2010). Time required to perform a cycle.

Lead Time: "Number of minutes, hours, or days that must be allowed for the completion of an operation or process, or must elapse before a desired action takes place" (Businessdictionary.com, 2010).

Effectiveness: "Doing the right things to create the most value for the company" (Chase *et al.*, 2006, p. 8).

Efficiency: "Doing something at the lowest possible cost…the goal of an efficient process is to produce a good or provide a service by using the smallest input of resources" (Chase *et al.*, 2006, p. 8).

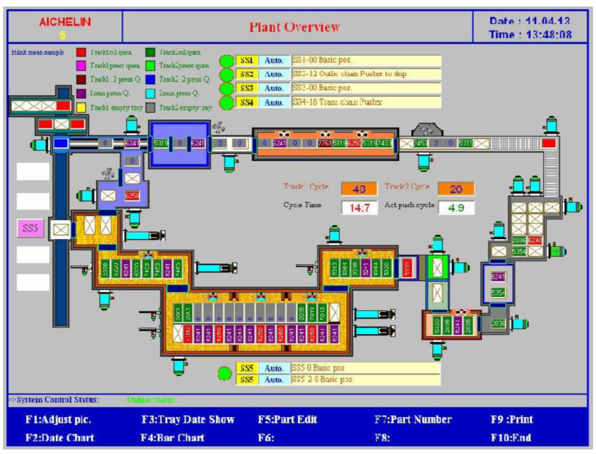
LEVEL OF TECHNOLOGY

Robots are often classified by their level of technology. These classifications are low-tech, medium tech, and high-tech. A low-tech robot is generally non servo and has only three or four axes. This type of robot has little feedback and very simple control units, and is typically used in pick and place tasks. Medium-tech robots have moderately sophisticated feedback systems and microprocessor-based control units. These robots have four to six axes. Medium-tech robots usually use teach pendants for programming. These are the most widely used types of robots, used for grinding, drilling, milling, and loading Numerically Controlled (NC) machines. High-tech robots are the most sophisticated type of robot. They employ state-of-the-art technology and use large mainframe computers as control units. High-tech robots have complex feedback systems, such as optical sensors and artificial intelligence. This type of servo robot is extremely flexible and can perform a variety of tasks, such as the assembly of television sets, personal computers, and stereo systems.

CARTESIAN ROBOT BASED PARAMETER ON AICHELIN SOFTWARE

In this Software were the Calculate of distance movement and pick the crown gear and place the tray track line board (Figure 2).

Figure 2: Simualation Working on Robot by Aichelin Software



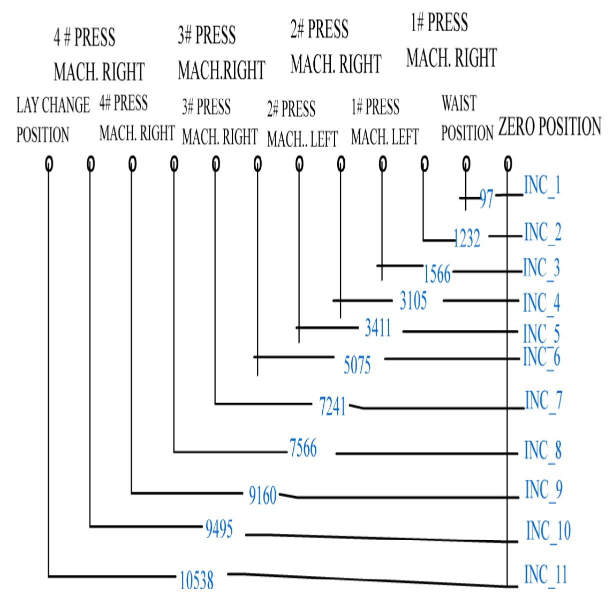
METHODOLOGY

Method of Dynamic Programming

Optimization Method

As shown in Figure 3, the total time taken by cartsian robot is sum of time taken during various presses.

Figure 3: Cartesian Robot Distance Movement



Suppose the time taken by the cartisian robot in one cycle time is T.

Then our goal is to Minimize cycle time

Hence the Optimization Function is as follow.

$$T\min = \min \sum_{i=1}^{17} ti[\ ]$$

where reduced Robot cartsian Cycle time is depends only on distance between patch 1 to patch 2. Because we can not make any anywhere else:

Hence

$$T\min = \min \left[\sum_{i=2}^{6}\right] ti$$

where $t2$, $t4$, $t6$ are constant. Because these are the necessary distance which have to travel by Cartesian robot.

Now our goal reduce the is only to reduce $t3$ and $t5$.

Only $t3$ and $t5$ are varying. Hence the optimization function is only depends upon $t3$ and $t5$.Means our goal is to minimise the distance travel by the Cartesian robot in time $t3$ and $t5$. So the final optimization function is:

$T\min = \min(t3 + t5);$

CONSTRAINTS

We have to minimize the distance travel during time t3 and t5 such that the the distance travel in these time should be grater then the height of stopper placed there.

Suppose the hight of stopper is hs and distance travel in time t3or t5 is d.

Then our constraints for minimization function is :-

d>hs

d = Distance Travel in Time $t3$ or $t5$

hs = Height of Stopper

Simulation of Actual and Estimated Time and Distance

Parameter of robot movement to pick and place crown gear to evaluate of reducing time parameter of cycle in comparison actual robot working cycle by calculating Matlab Software (see Appendix).

Flow Chart 1: Estimating Parameters of Cycle Time

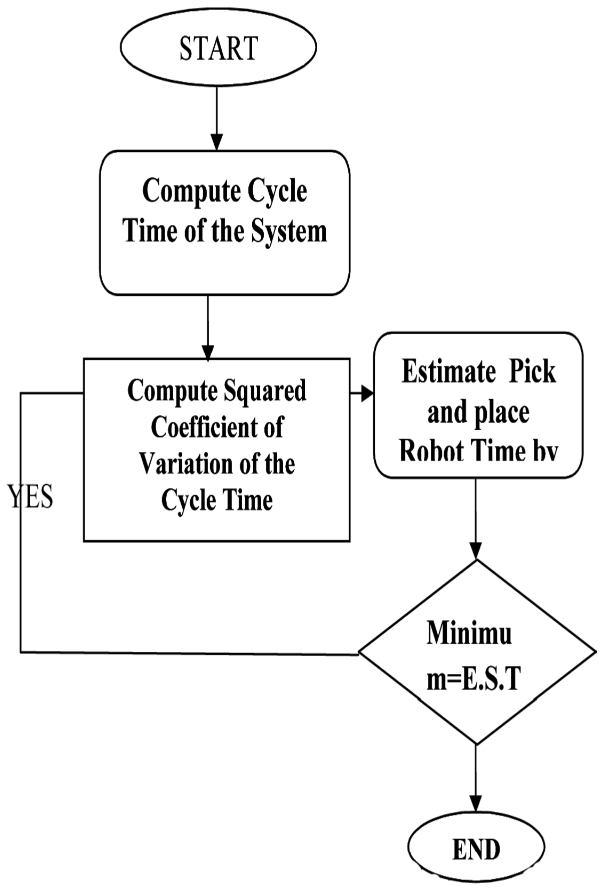
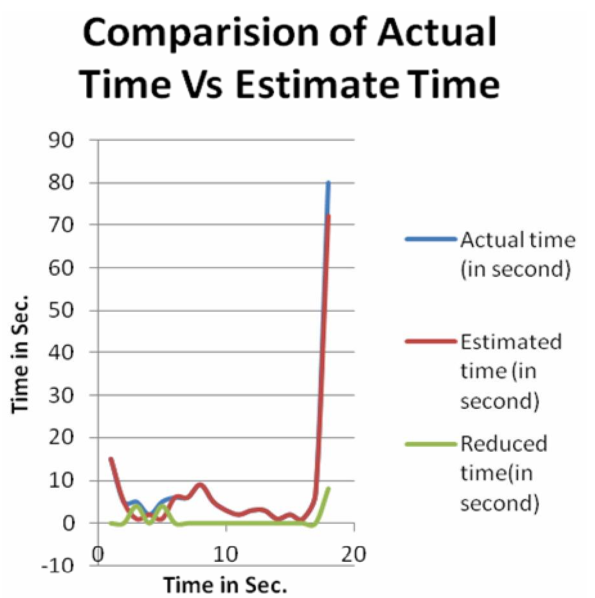


Table 1: Comparison of Actual vs. Estimated Time Data by Using C++ and Matlab Coding

Actual Time (in Second)	Estimated Time (in Second)	Reduced Time (in Second)
15	15	0
5	5	0
5	1	4
2	2	0
5	1	4
6	6	0
6	6	0
9	9	0
5	5	0
3	3	0
2	2	0
3	3	0
3	3	0
1	1	0
2	2	0
1	1	0
7	7	0
80	72	8

Graph 1: Actual vs. Estimates Time



Comparision Data Actual vs. Estimated Time

Therefore, may be the actual time working in industries and estimated time solving by C++ and Matlab Coding Programming (Table 1 and Graph 1).

RESULTS AND DISCUSSION

our research work are to be the estimated new cycle time of a robot movement is 72 sec per process, estimated no. of Cycle increases of a robot movement is 5 cycle per hour, automatically saving a time is 8 sec by Shortest Travelling problem to reducing the travel path of robot movement.

Calculation Robot Cycle Time Analytical

Robot cycle time based parameter was calculated with ACHLEINE Software, therefore may be consumed the time 80 millisecond per each one complete of cycle.

But new estimated robot cycle time 72 millisecond per each one complete of cycle Consumed to Reduced 8 millisecond there for may be productivity me be increased of per cycle.

Actual Robot Cycle Time = 45 cycle completed in One hr.

Estimated Robot Cycle Time = 50 cycle completed in One hr.

Productivity Robot Cycle Time Increased = Actual C.T.-Estimated C.T.

= 50 cycle-45 cycle

Productivity Robot Cycle Time Increases = 5 Cycle per hr Increased

Actual Robot Cycle Time = 80 sec per each process

Cycle Process

1 Robot cycle per hr = 45 cycle per hr

1 Robot cycle per shift = 360 cycle per shift

1 Robot cycle per day = 1080 cycle per day

1 Robot cycle per weak = 8400 cycle per weak

1 Robot cycle per month = 32400 cycle per month

1 Robot cycle per year = 3153600 cycle per year

Estimated Robot Cycle Time = 72 sec per each process

Cycle Process

1 Robot cycle per hr = 50 cycle per hr

1 Robot cycle per shift = 400 cycle per shift

1 Robot cycle per day = 1200 cycle per day

1 Robot cycle per weak = 7560 cycle per weak

1 Robot cycle per month = 36000 cycle per month

1 Robot cycle per year = 438000 cycle per year

No. of Cycle Difference = 43800 Cycle/year

The Robot increase per year cycle 43800 times, and efficiency increase per year 9% (Figure 4).

Figure 4: Output Plot 1: C++ Programming for Estimated Cycle Time

CONCLUSION

The Cartesian robot scheduling problem considered in this paper can be formulated as type of dynamic programming problem. Achleine Software was used to control the cycle time where to estimate our new control the cycle time by C++ and Matlab Programming. The actual cycle time is 80 sec completed the robot by one process but new estimated time is 72 sec. Therefore the 8 sec reducing time is automatically increase the productivity 5 cycles per hr. The main result of this paper was considered as, the problem can be solved in dynamic programming using C++ and Matlab software.

ACKNOWLEDGEMENT

Author acknowledge the help received from scholars whose articles are citied and included in reference of this manuscripts. The author are also grateful to authors/editors/publishers of all those articles, journals and books from where the literature for.

REFERENCES

1. Ayob M and Kendall G (2002), "A New Dynamic Point Specification Approach to Optimise Surface Mount Placement Machine in Printed Circuit Board Assembly", Proceedings of the IEEE ICIT'02, December 11-14, pp. 486-491, Bangkok.
2. Ayob M, Cowling P and Kendall G (2002), "Optimisation for Surface Mount Placement Machines", Proceedings of the IEEE ICIT'02, December 11-14, pp. 498-503, Bangkok.
3. Ayob M and Kendall G (2003a), "Real-Time Scheduling for Multi Headed Placement Machine", Proceedings of the 5th IEEE International Symposium on Assembly and Task Planning, ISATP'03, July 9-11, 128-133, Besançom, France.
4. Ayob M and Kendall G (2003b), "An Investigation of an Adaptive Scheduling for Multi Headed Placement Machines", Proceedings of the 1st Multidisciplinary International Conference on Scheduling: Theory and Applications, MISTA, August 13-16, pp. 363-380, Nottingham, UK.
5. Ayob M and Kendall G (2003c), "A Monte Carlo Hyper-Heuristic to Optimise Component Placement Sequencing for Multi Head Placement Machine", Proceedings of the International Conference on Intelligent Technologies, Intech'03, December 17-19, 132-141, Chiang Mai, Thailand.
6. Ayob M and Kendall G (2004), "A Nozzle Selection Heuristic to Optimise the Hybrid Pick and Place Machine", Proceedings of the 2004 IEEE Conference on Cybernetics and Intelligent Systems (CIS), pp. 1259-1264, Singapore.
7. Ayob M and Kendall G (2005a), "A Triple Objective Function with a Chebychev Dynamic Point Specification Approach to Optimise the SMD Placement Machine", *European Journal of Operational Research*, Vol. 164, pp. 609-626.
8. Ayob M and Kendall G (2005b), "A Variable Neighbourhood Monte Carlo Search for Component Placement Sequencing of Multi Headed Placement Machine in Printed Circuit Board Assembly", Submitted to the *Intelligent Manufacturing Journal*.

9. Ayob M and Kendall G (2005c), "A Weighted Nozzle Rank Heuristic to Optimise the Hybrid Pick and Place Machine", Submitted to the *International Journal of Production Research*.
10. Ball M O and Magazine M J (1988), "Sequencing of Insertions in Printed Circuit Board Assembly", *Operations Research*, Vol. 36, pp. 192-201.
11. Bard J F (1988), "A Heuristic for Minimizing the Number of Tool Switches on a Flexible Machine", *IIE Transactions*, Vol. 20, pp. 382-391.
12. Bard J F, Clayton R W and Feo T A (1994), "Machine Setup and Component Placement in Printed Circuit Board Assembly", *International Journal of Flexible Manufacturing Systems*, Vol. 6, pp. 5-31.
13. Bentzen B (2000), "SMD Placement, in the SMT in FOCUS", Url: Http://Www.Smtinfocus.Com/PDF/SMD_Placement.Pdf (September 25, 2002).
14. Blum C and Roli A (2001), *Metaheuristics in Combinatorial Optimization*, Overview and Conceptual Comparison, Technical Report TR/IRIDIA/2001-13, IRIDIA, Belgium.
15. Bonert M, Shu L H and Benhabib B (2000), "Motion Planning for Multi-Robot Assembly Systems", *International Journal of Computer Integrated Manufacturing*, Vol. 13, No. 4, pp. 301-310.
16. Burke E K, Cowling P I and Keuthen R (1999), "New Models and Heuristics for Component Placement in Printed Circuit Board Assembly", Proceedings of The 1999 IEEE on ICIIS99, pp. 33-140.
17. Burke E K, Cowling P and Keuthen R (2001), "The Printed Circuit Board Assembly Problem: Heuristic Approaches for Multi-Headed Placement Machinery", Proceedings of the IC-AI2001, pp. 1456-1462, Las Vegas, CSREA Press.
18. Cavalloro P and Cividadi E (1988), "An Expert System for Process Planning in PCB Assembly Line", CH2552-8/88/0000/0170, *IEEE*, pp. 170-174.
19. Chan D (1993), "Precedence Constrained TSP Applied to Circuit Board Assembly and No Wait Flow-Shop", *International Journal of Production Research*, Vol. 31, No. 9, pp. 2171- 2177.
20. Chan D and Mercier D (1989), "IC Insertion: An Application of the Travelling Salesman Problem", *International Journal of Production Research*, Vol. 27, pp. 1837-1841.
21. Chandra P, Li S and Stan M (1993), "Jobs and Tool Sequencing in an Automated Manufacturing Environment", *International Journal of Production Research*, Vol. 31, pp. 2911-2925.
22. Chang T C and Terwilliger J (1987), "Rule-Based System for Printed Wiring Assembly Process Planning", *International Journal of Production Research*, Vol. 25, pp. 1465-1482.
23. Chen W-S and Chyu C-C (2003), "A Minimum Setup Strategy for Sequencing Pcbs with Multi-Slot Feeders", *Integrated Manufacturing Systems*, Vol. 14, No. 3, pp. 255-267.

APPENDIX

Figure 1: Actual vs. Estimated (1)

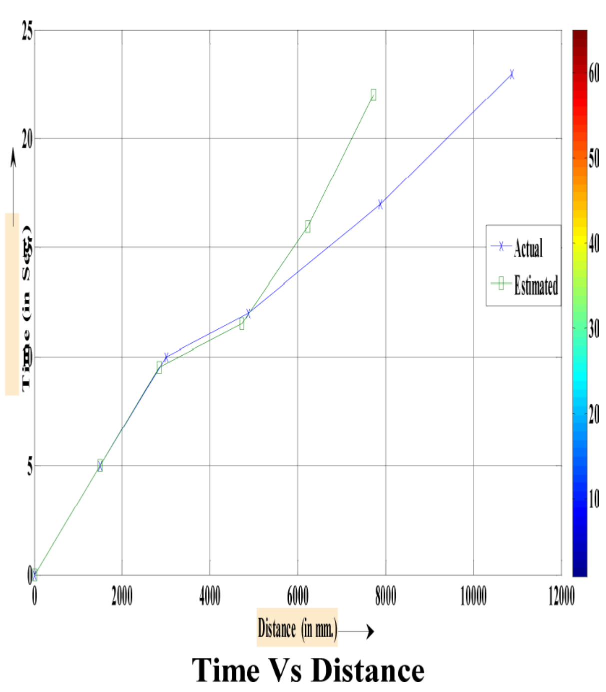


Figure 2: Actual vs. Estimated (2)

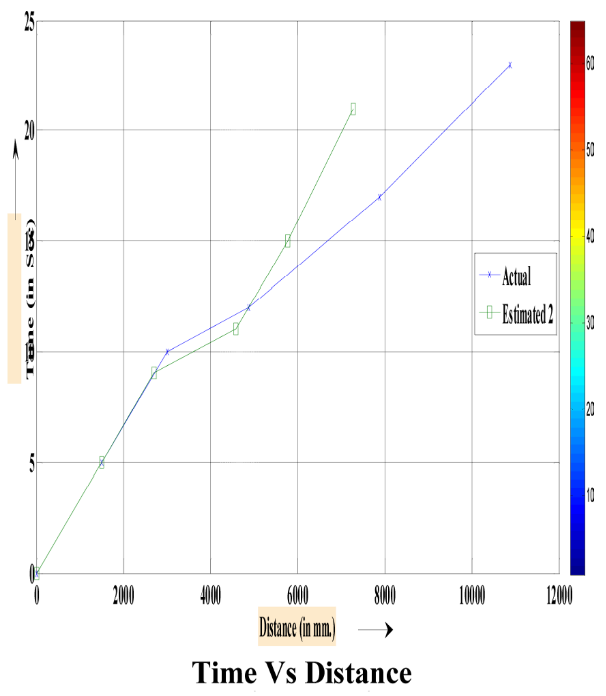


Figure 3: Actual vs. Estimated (3)

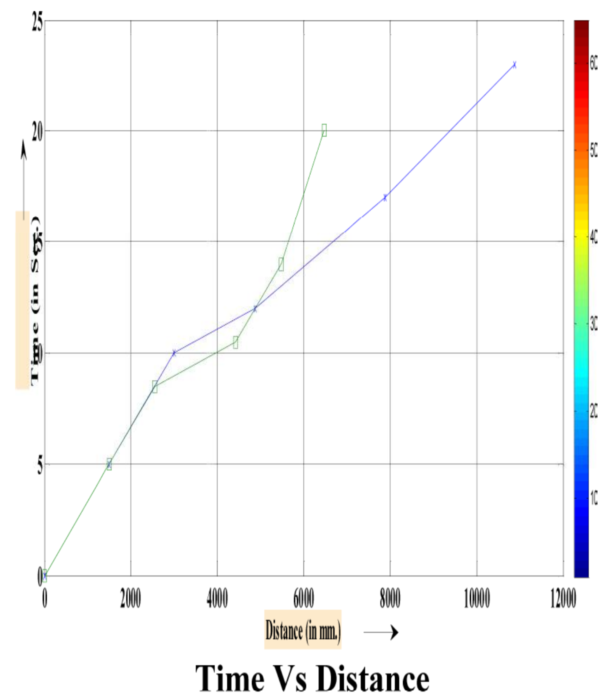
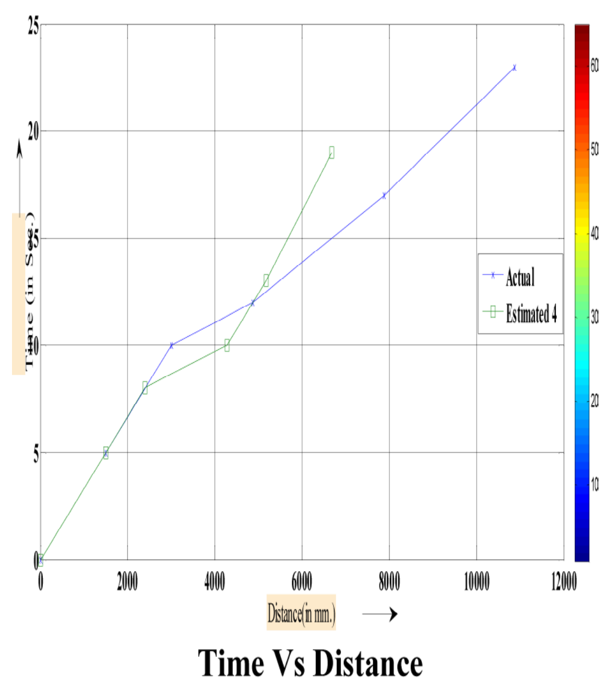


Figure 4: Actual vs. Estimated (4)



APPENDIX (CONT.)

Figure 5: Actual vs. Estimated (5)

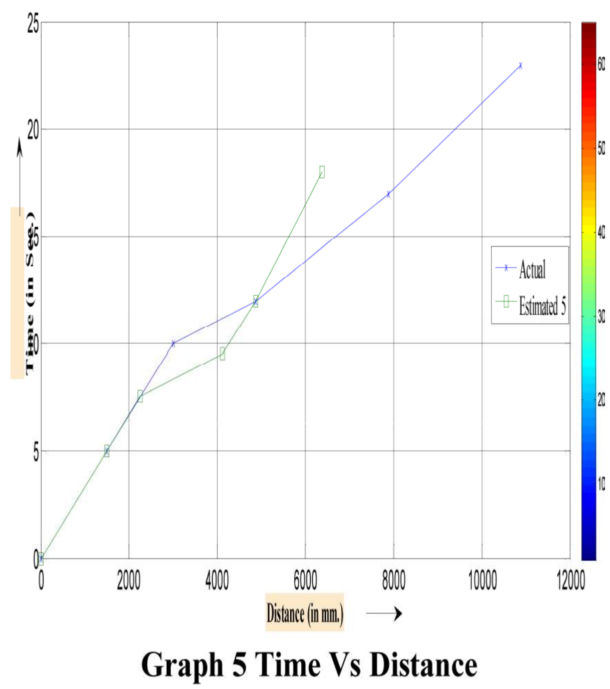


Figure 7: Actual vs. Estimated (7)

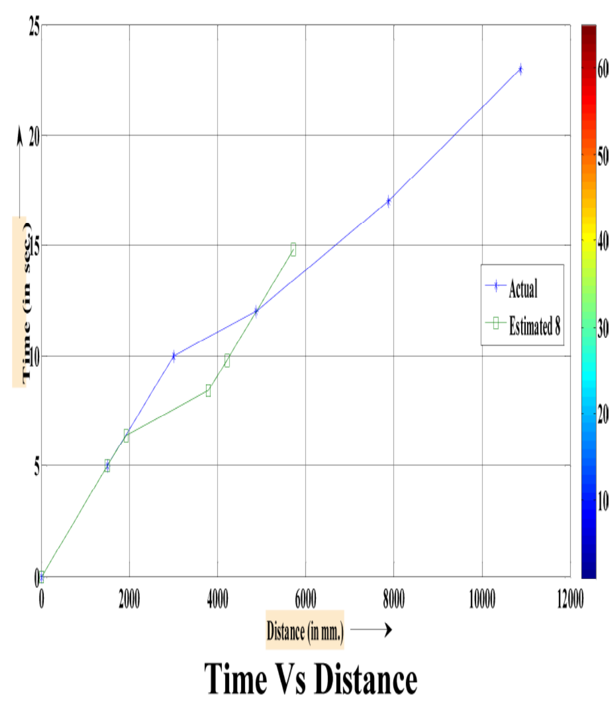


Figure 6: Actual vs. Estimated (6)

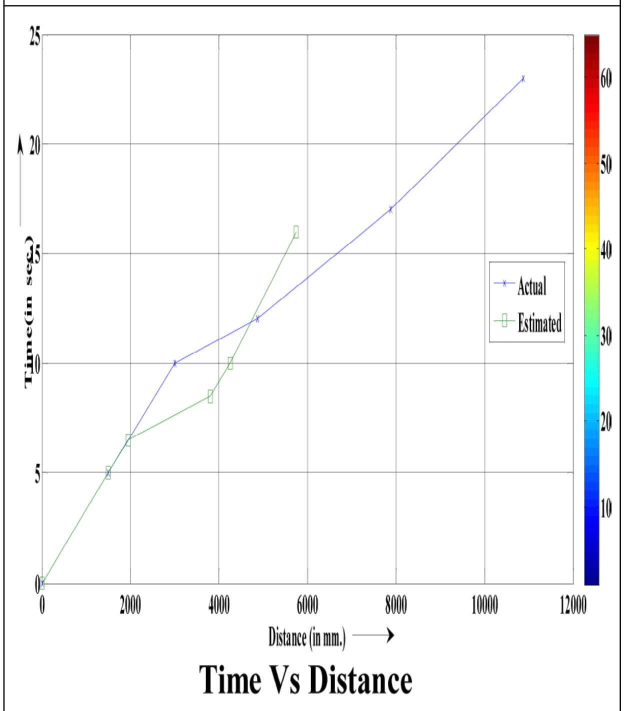
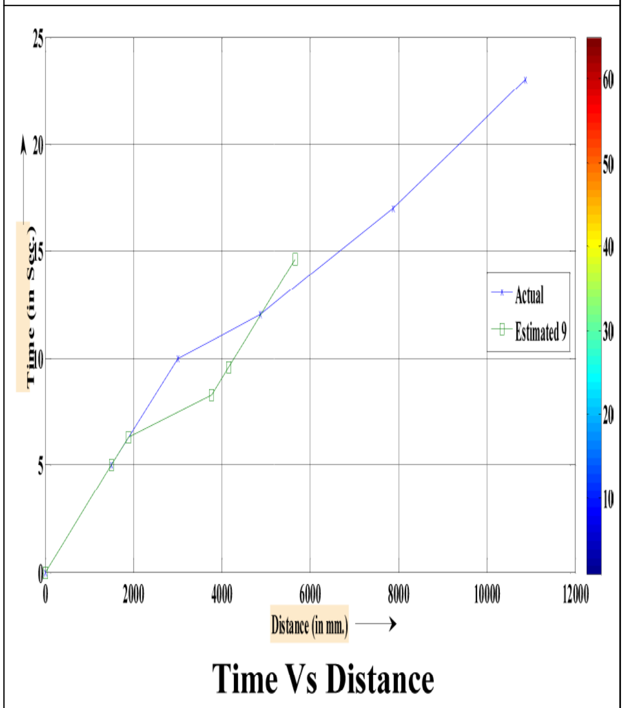


Figure 8: Actual vs. Estimated (8)



APPENDIX (CONT.)

Figure 9: Actual vs. Estimated (9)

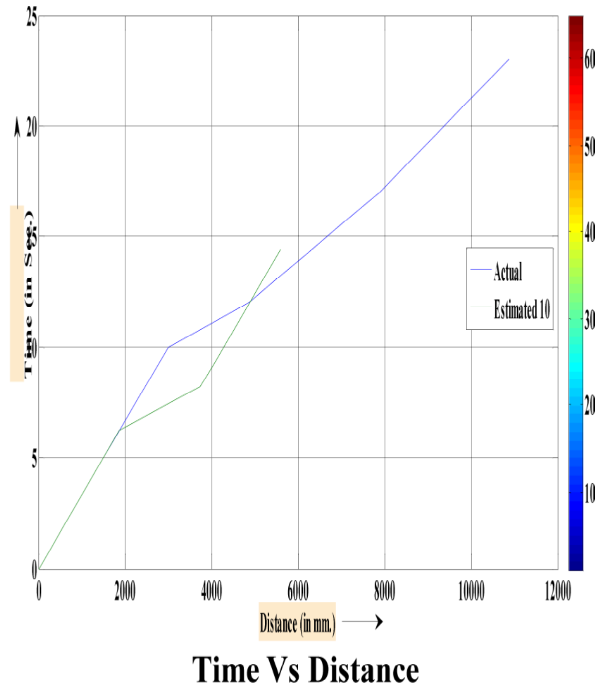


Figure 10: Actual vs. Estimated (10)