

Implementation of Conventional and Neural Controllers Using Position and Velocity Feedback

Expo Paper

Department of Electrical and Computer Engineering

By:

Christopher Spevacek

and

Manfred Meissner

Advisor:

Dr. Gary Dempsey

Abstract:

The project objective was to investigate and compare different algorithms for the calculation of velocity from position information. The best algorithm was applied to a small robot arm system which consists of a controller (PC software), analog-to-digital and digital-to-analog converter PC card, power amplifier, DC motor, gear train, and external load. In robotic systems a velocity calculation is difficult or impossible to implement because of noise. A neural network will be used to filter the noise from the position data before calculating velocity. The controllers that were implemented were a proportional controller, a feed-forward controller, and a two-loop controller consisting of the velocity algorithm and neural network filter. Experimental results show the benefits of our two-loop controller.

Introduction:

Our project objective is to design and evaluate different controllers for a small robot arm-motor platform from Quanser Consulting. The controllers and signal processing algorithms were implemented in C language on a 200 MHz Pentium Computer. An internal A/D and D/A converter card is connected to the external plant as shown in Fig. 1. The plant consists of an amplifier, DC motor assembly, external gear train, external load (robot arm), and potentiometer for the position sensor. The feedback voltage will be passed through an anti-aliasing filter, to the A/D converter, into the computer. The feedback voltage signal is proportional to the position of the robotic arm. The arm position, output position on Fig. 1, is the primary output of the system, although arm velocity will also be used.

Objective:

The objective of this project is to show how different types of controllers affect hydraulic and robotic systems. This project compares a proportional control, feed-forward control, minor loop

control, and minor loop control using neural networks. The proportional and feed-forward control use position feedback only, but the minor loop control uses velocity and position feedback. The velocity will be determined in software from the position feedback. Another major objective is to determine a linear model of the robot arm system. This was a critical part of the project because this model determines the different controller parameters. The final objective is to design a user-friendly interface to run and interchange these controllers on-line.

Significance of Research:

The hydraulic systems area is one of the more challenging areas for closed-loop control especially in the large machinery area. Hydraulic systems are expensive and require large laboratory space. Control algorithms are investigated on smaller systems such as robot arms. Our robot arm system exhibits dynamics (mathematics) similar to a hydraulic actuated system. The predominant method used in industry is human control. Many product parameters cannot even be specified because they rely on the person's experience and training. Automatic electronic controllers can eliminate the repetitive tasks performed by the human controllers as well as achieving better performance such as final position accuracy. Final positioning accuracy has only become a concern in the last several years. Good accuracy is difficult to obtain with conventional controllers.

The first step to improve the hydraulic system is to add a conventional closed-loop electronic controller. The PID controller is the most widely used controller in industry and also the easiest method to design and implement. However, there are many problems and challenges with the PID controller for the large machinery hydraulic applications. The nonlinear time-varying parameters of the hydraulic system present a challenge for this control design method. One example of a hydraulic system is a Caterpillar wheel loader. The primary limitation of the PID

controller is that it is not adaptive to the different load conditions found in the large hydraulic system area [1]. For example, the wheel loader's bucket load can vary from 0 to 33,000 pounds.

Student Involvement:

The two students who worked on this project were Christopher Spevacek and Manfred Meissner. We both performed the system identification of the robot arm, which required approximately three weeks. We also designed all the controllers and the neural network plus writing the code to program the controllers in C language. Dr. Gary Dempsey, our advisor, supplied papers on different velocity algorithms, which we investigated. The best approaches are discussed in the following text.

Methodology:

The first step, in the project was to determine a linear model of the robot arm shown in Fig 1. A block diagram of the robot arm system is shown in Fig. 2. The block diagram shows that the model is a third order system. A pole is located at the s-plane origin due to the integrator. The armature pole is at high frequency that will not affect the experimental results. Therefore, a second order plant was used for the model. Experimental measurement from the system yields the following model for the plant.

$$Gp = \frac{5.05 * e^{-0.025s}}{s(s/2 + 1)} \quad (1)$$

After the plant was found the next step was to design the proportional controller, because this is the easiest one to design. The proportional controller just adds a gain to the system. This was simulated and tested. The results are shown in Fig. 3. Note that an error (tracking error) exists between the command signal and the system output. Ideally, the two curves should overlap. The

feed-forward controller was added next, to reduce the tracking error. The feed-forward controller is the inverse of the plant model. A first-order approximation is

$$G_{FF} = \frac{s}{s/10+1} \quad (2)$$

The block diagram, Fig. 4., shows the location of the feed-forward controller. The gain k is the only parameter for the P-controller design. The output of the plant is shown in Fig. 5. Note that the tracking error has been reduced. The next step was to design the minor loop controller. This is different than the other two methods because the controller uses velocity feedback. To do this, position data had to be differentiated in order to get velocity. We researched different algorithms and the best was found to be Tustin's Method (3).

$$Tustin = \frac{2}{T} \frac{z-1}{z+1} \quad (3)$$

To further reduce the tracking and over-shoot error, a minor loop controller was designed. The minor loop is just a differentiation of the output signal, which gives the velocity. This loop does not affect the overall order of the system. After obtaining the overall transfer function of the minor loop (4), the proportional and feed forward controllers (5) had to be readjusted.

$$G_{ML} = \frac{s*0.35}{s/30+1} \quad (4)$$

$$G_{FF} = \frac{s*0.02}{s/30+1} \quad (5)$$

The location of the feed-forward and the minor loop is shown in Fig. 6. The output is shown in Fig. 7. Note that the tracking error and over-shoot are reduced. The neural network was used to filter the noise of the output signal in order to obtain better differentiating results (Fig.8, 9, 10, 11). Neural networks are programming techniques to emulate a brain. It is adaptive and so it can

react better to various changes in the surroundings or plant. In connection with the minor loop better results were expected, but the curve-fitting algorithm used did not do the job very well. The overshoot was just slightly smaller than without the neural network and the tracking was not as good as in the conventional minor loop (Fig.12).

Results:

The proportional controller is the easiest to design but does not have the best results as seen in Fig. 3. When the feed-forward controller was implemented the tracking was better but an overshoot was introduced to the output as shown in Fig. 5. With minor loop control, the tracking was almost perfect and the over-shoot was lowered (see Fig.7).

Conclusion:

With five out of six controllers working: proportional, feed-forward, minor loop, feed-forward with minor loop, proportional with minor loop, the one that did not perform as expected was the minor loop with the neural network. The neural network should have the best results because it smoothes out the position output signal. Further investigation of the neural algorithm is required but the simulation and experimental results are promising. Note the reduction in the level of noise from Fig.9 to Fig.10. The neural network problem has been determined. An extension to the neural network algorithm is currently under investigation to improve performance.

Reference:

[1] G. L. Dempsey. "ALS 990 Wheel Loader: Controller Design for Bucket and Steering Systems," Technical Report, Caterpillar Inc., Systems and Controls Research Department, November 1996.

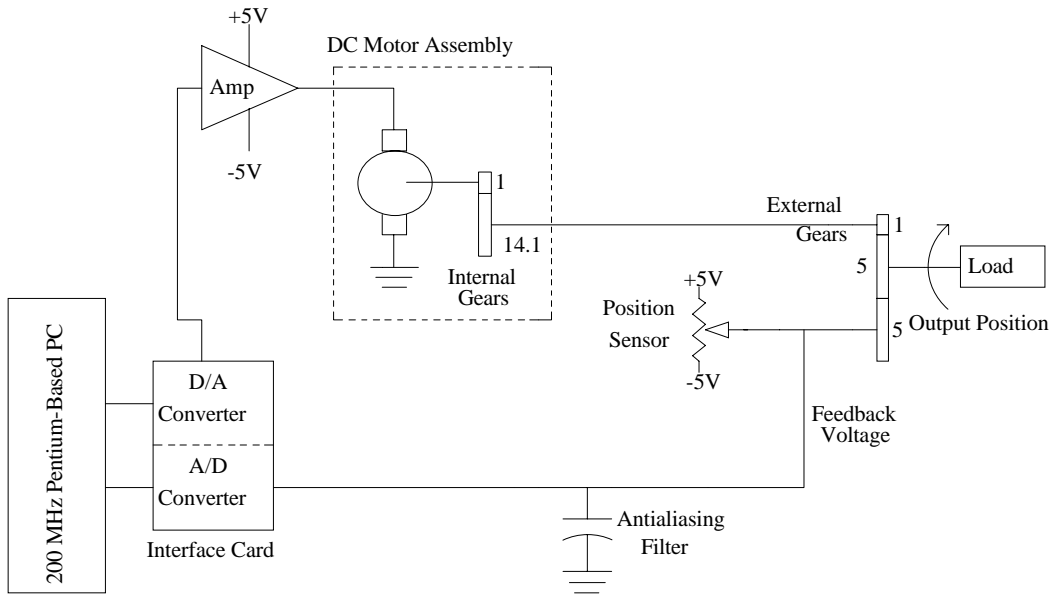


Fig. 1. Low Level System Block Diagram

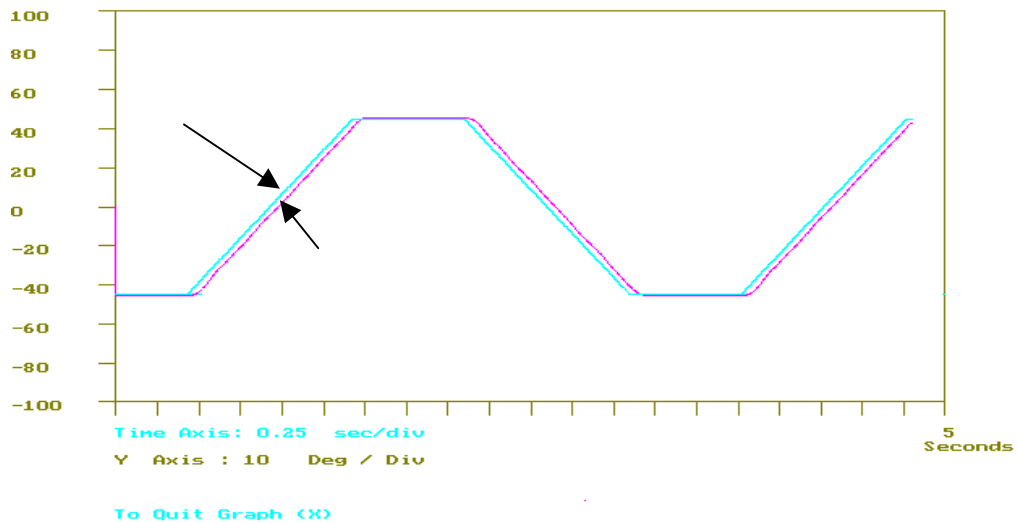
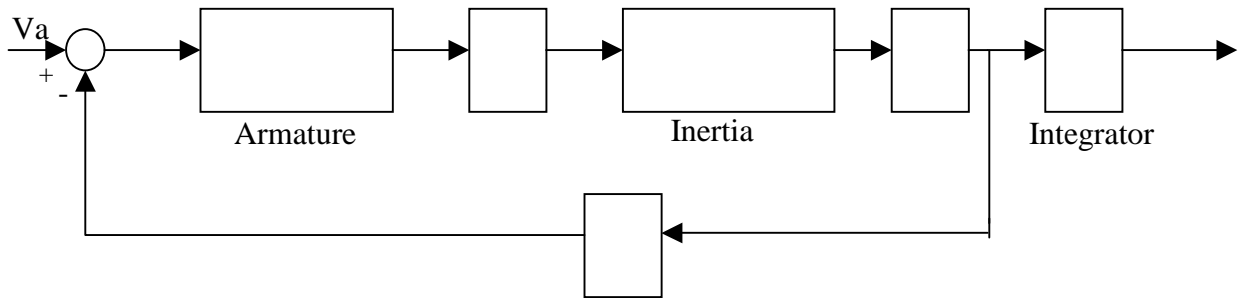


Fig. 3. P-Controller Output

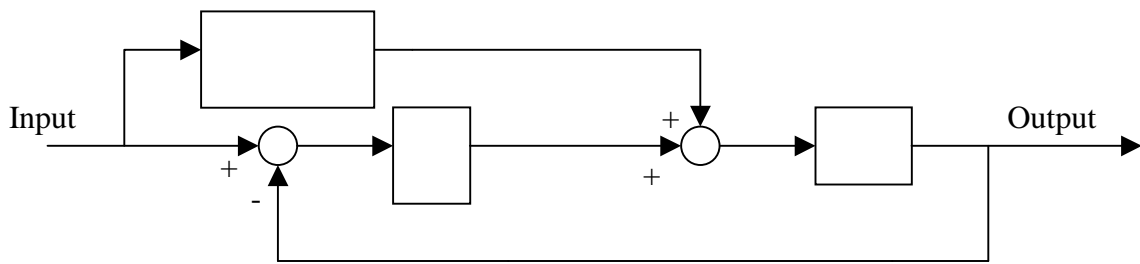


Fig. 4. Feed Forward P-Controller

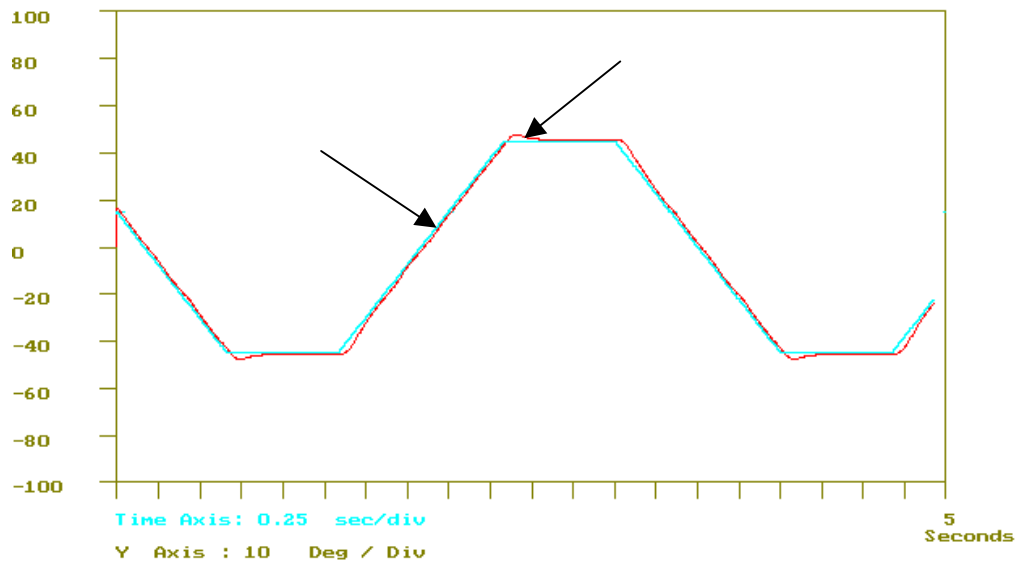


Fig. 5. Feed-Forward Output

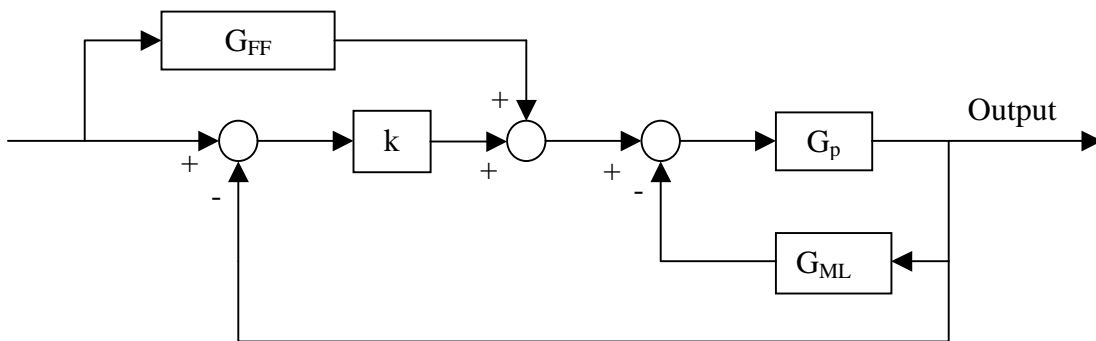


Fig. 6. Minor Loop Block Diagram

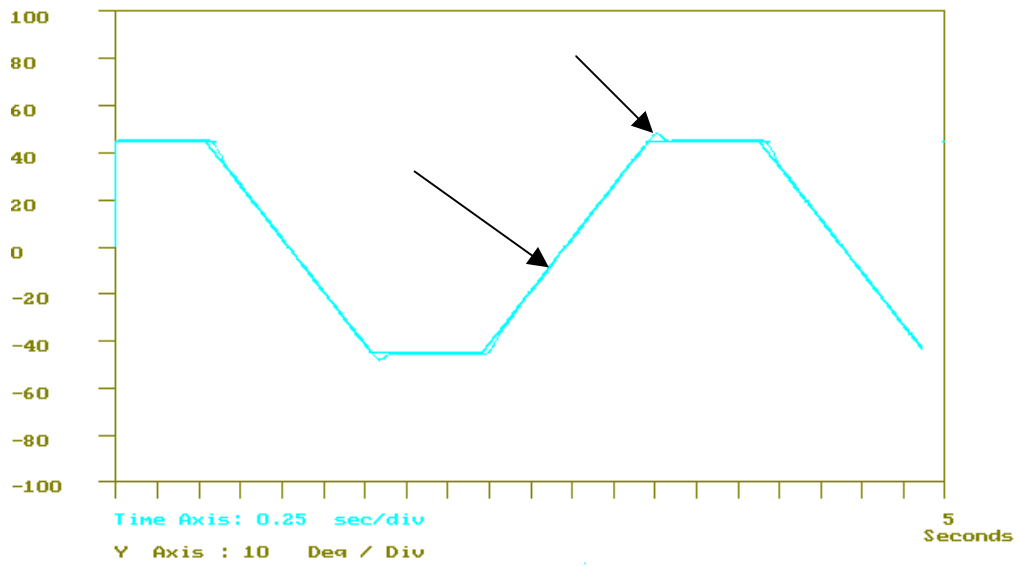


Fig. 7. Minor Loop Output

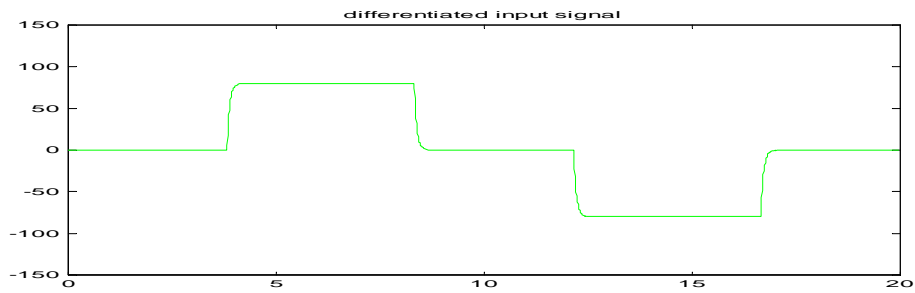


Fig.8. Ideal Differentiation of Input Signal

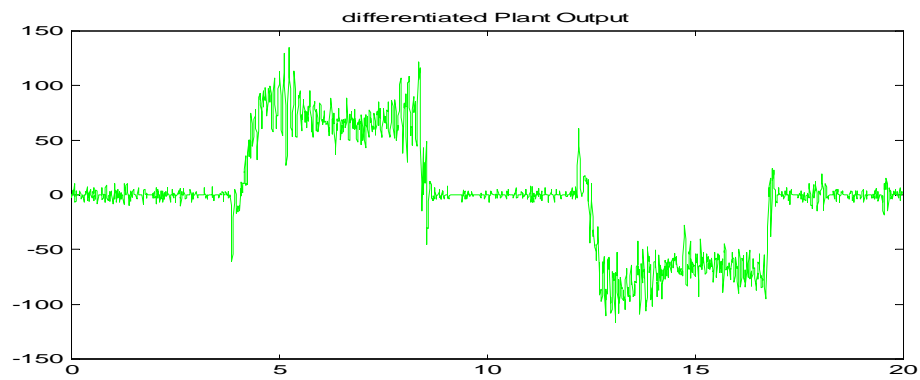


Fig.9. Differentiated Plant Signal without ANN

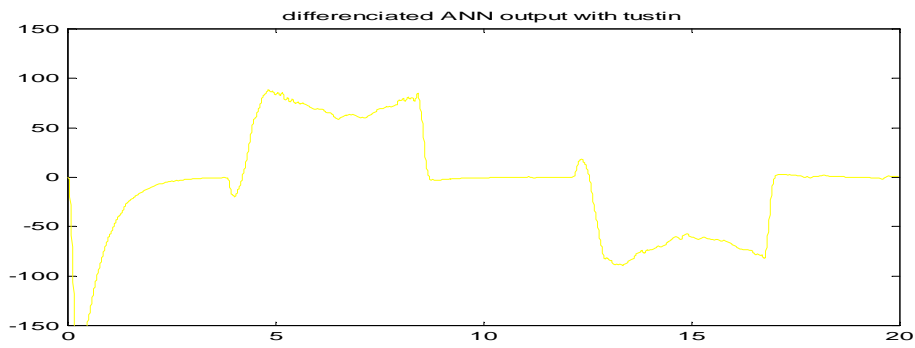


Fig.10. Differentiation with ANN and Tustin's Method

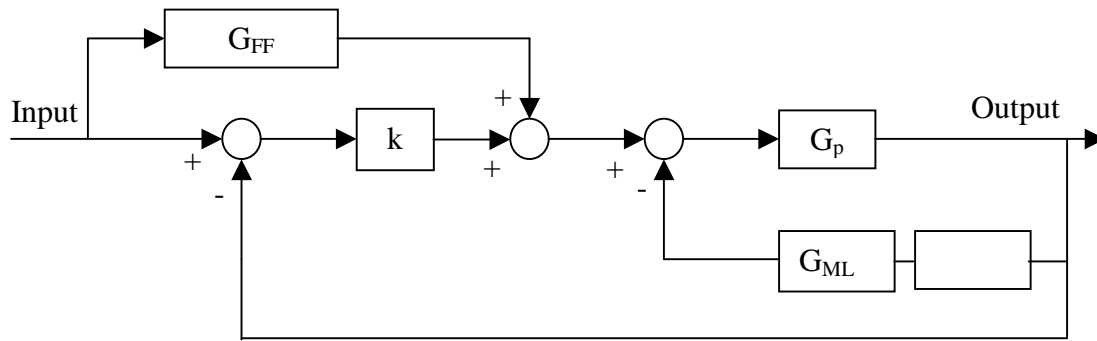


Fig. 11. Minor Loop with ANN Block Diagram

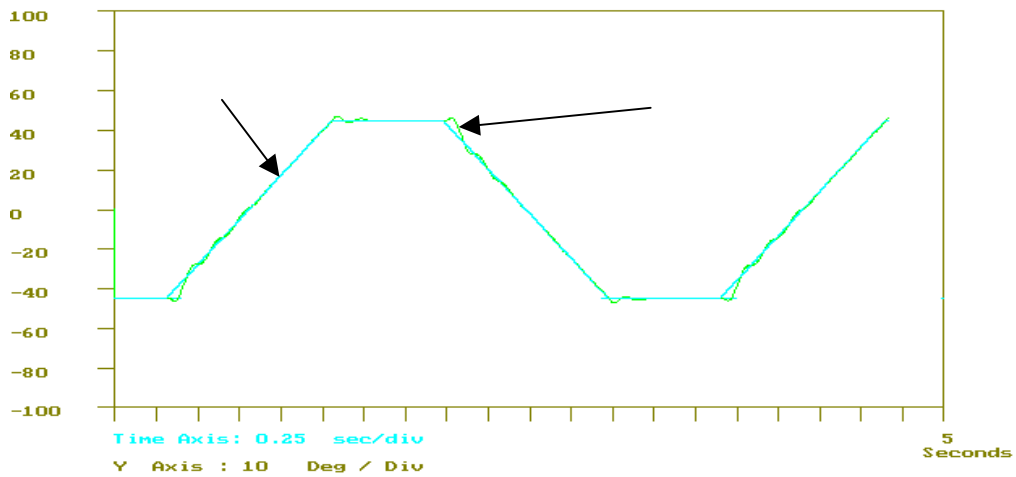


Fig. 12. Minor Loop with ANN Output