A Closed Loop Feedback Method for a Manual Bar Straightener

Robert J. Miklosovic, Zhiqiang Gao
Department of Electrical and Computer Engineering
Cleveland State University
Cleveland, Ohio, USA

Abstract—Automation of a unique manually controlled industrial bar straightener is proposed. A continuous-time closed loop model is constructed in Simulink for an event-driven process through the use of asynchronous timers. The system is simulated with linear and nonlinear PD controllers. A nonlinear filter, called the tracking differentiator, is introduced as an alternative to a linear approximate means of providing accurate derivative feedback in the presence of noise. In both cases, the nonlinear techniques outperformed their linear counterparts while retaining tuning simplicity.

I. BACKGROUND

Precision straightening of a cylindrical metal bar is largely based on the ability to precisely measure its geometry. A few fundamental measurements and how each influences the tolerance specification on straightness should first be understood. Methods for measuring roundness and straightness are covered to lay the groundwork for the problem formulation.

The basic operation of the machine is outlined in Section II, and its fundamental limitations and need for automation are discussed. Section III addresses the task of closing the loop through block diagrams and the role of new hardware in the process. Section IV contains descriptions of all of the blocks that are modeled in Simulink. The linear and nonlinear controller designs are discussed in Section V, the system is simulated in Section VI, and concluding remarks are made in Section VII.

A. Measuring Roundness

Roundness is a quantity derived from comparing the shape of a cross-sectional area at one distinct point along a cylinder’s length against a circle. A round metal bar that is arbitrarily long with respect to its diameter has to be checked for roundness in many locations lengthwise and averaged to insure overall consistency. Roundness is approximated by rotating the work piece one revolution in a Vee block while measuring the surface with an indicator. Taking the difference between the minimum and maximum indicator readings in this case is referred to as the total indicator reading (TIR) [1].

B. Measuring Straightness

Straightness is a quantity derived from comparing the axial centerline of a specific section of a cylinder’s length against a straight line. A simple method for approximating straightness is by rotating the bar one revolution between two Vee blocks that are a fixed distance (d) apart, while measuring in the center with an indicator. The distance that the axial centerline of the part deviates from a theoretically straight centerline directly below the indicator equals the extent to which the part is bowed, or warped, over length d. The maximum and minimum indicator readings (IX and IN) are physically represented in Fig. 1. From this, TIR is derived as:

\[ \text{TIR} = I_N - I_0 = (R + |\text{Bow}|) - (R - |\text{Bow}|) = 2|\text{Bow}| \] (1)

Deviations in roundness, outside diameter (OD) size, and finish can adversely affect the measurement.

Figure 1. Max. and min. indicator readings of a bowed part

C. Straightening

The straightening process, which can be broken into steps, simply involves correcting any error while checking for straightness. First, the part is measured for straightness. Then, it is rotated so that the bow is oriented 180 degrees away from the Vee blocks with the maximum indicator reading facing upwards. Finally, a counter-bending force replaces the indicator and straightens the work piece against the Vee blocks.

II. MACHINE OPERATION

The straightener to be automated uses a non-contact ultrasonic sensor in place of the indicator and rollers in place of the Vee blocks in an effort to minimize contact wear. The part slowly spirals through the machine. The indicator reading
Each is observed and taken into consideration when producing controller’s performance and slow it down by extending \( Y \). Becomes a continuous sinusoid at the sensor’s output, having a peak-to-peak value equal to the TIR each revolution. TIR is sampled from the sensor output and calculated each revolution, making the sample period of one revolution the minimum time between consecutive bends (\( Y_{SP} \)). TIR is the plant output (Y) to be controlled. When the part is straightened, the machine stops rotation with the bow facing upwards, but the part continues to feed lengthwise while an air cylinder counter-bends the part over a period of time. This bend time (\( B_2 \)) is the control variable (U). Fig. 2 illustrates this operation.

![Figure 2. The straightener to be automated](image)

**A. Process Limitations**

There are aspects of the process that can limit the controller’s performance and slow it down by extending \( Y_{SP} \). Each is observed and taken into consideration when producing an accurate simulation model:

1. The ultrasonic sensor introduces RFI noise into its output. The use of a feedback filter is essential.
2. A rough part surface finish adds distortion to the sensor output.
3. An out-of-round part superimposes harmonics on the sensor output sinusoid, placing a bound on the minimum steady state error that is achievable.
4. Inconsistent material density produces false measurements. The measured focal length of a transducer is dependent on the density of the material that is being measured [2]. The unit cannot measure accurately in the presence of a time-variant material density (i.e., hard spots). Although unavoidable, it can be detected, since TIR changes monotonically.
5. An inconsistent OD causes vertical shifts in the sensor output. A differential TIR measurement cancels these effects.
6. A twisted part condition is detected when the angular position of the maximum indicator reading slowly moves with each revolution. This condition is created when the part is not straightened at the precise angular location and occurs because of the quantization affect of the digital readout used by the operator. The new controller will use a continuous signal and the part can be straightened 30 to 45 degrees ahead of the twist when encountered.

**B. The Need for Automation**

Replacement of the operator with electronic hardware is beneficial in several ways. The cost of the electronics is much less than the ongoing hourly wage and schedule of an operator. The limitations associated with the digital readout are eliminated, which helps the machine to straighten faster with more precision. The process can be drastically sped up to produce more. Though the minimum sample time is one revolution, it does not need to be slow enough for human comprehension. A Programmable Logic Controller (PLC) can make several calculations and test the result against a set of rules many times faster than a human being.

**C. Research Methodology**

The focus is split between modeling and control design, since this is a new control problem. The process is manually controlled rather than being strictly manual in operation, meaning the machine needs only a new controller. There is no need for a complete mechanical overhaul, so the best method of straightening is not researched. Typical of a small company, time and money are limited. Gao and Huang [3] presented a new error-based control design framework including such innovations as a nonlinear tracking differentiator and a nonlinear proportional-integral-derivative (NPID) control method. These methods prove to be powerful and simple to tune, which make them ideal for use in an industrial environment.

**III. A CLOSED LOOP SOLUTION**

The task of automation can begin once the process is well defined. A straightforward system block diagram is developed, and each block is modeled in Simulink. Aspects of the hardware configuration are carefully considered.

**A. From Open Loop to Closed Loop**

The open loop multi-input-multi-output (MIMO) block diagram, in Fig. 3, represents the manually controlled process. The operator calculates TIR and monitors the angular position (\( \theta_B \)) of the bow with each revolution. When Y approaches a specified limit, the operator sets \( B_2 \) proportional to Y and its rate, and then pushes a button (\( B_2 \)) to straighten the part. The bend timer creates a pulse triggered by \( B_2 \) that counter-bends the part for \( B_2 \) minutes. This event forces the part to have a specific rate for a period of time. After the event, the part takes on a new rate and the process perpetuates. Therefore, \( Y' \) is a piece-wise continuous function of time. The operator’s involvement in the process is represented as a block in Fig. 4.

![Figure 3. Open loop block diagram](image)
Rearranging the blocks and breaking each function down into smaller more-manageable blocks reduces the representation to a usable closed loop, single-input-single-output (SISO) form. Fig. 5 shows how a SISO plant is obtained by combining the process with the task of sampling the TIR, since it can be consistently computed.

The plant has a variable-width pulse as the input (U) and the sampled TIR (Y) as the output to be controlled. The incoming changing position of the work piece is modeled as an unknown rate disturbance (D). The closed loop SISO block diagram is shown in Fig. 6.

### B. Hardware Configuration

Depicted in Fig. 7, an encoder and a PLC are the only hardware needed for automation. The encoder feeds the angular position of the part back to the controller, and the PLC handles all of closed loop functions outside of the process.

Fig. 8 depicts the hardware layout for plant data acquisition during manual operation. The PLC is also used to calibrate a strip chart recorder, shown in Fig. 9, which simplifies calibration for the operator and removes room for error during data acquisition. Using the PLC for both data acquisition and control keeps costs lower.

### IV. Modeling

Simulation modeling involves the construction of Simulink blocks for each of the blocks in the SISO diagram. Three modular blocks are first designed to accommodate the presence of an event-driven plant in a continuous-time environment where the control variable is an asynchronous pulse width. They are all based on creating asynchronous timers in Simulink by integrating a constant until it reaches a preset value, then shutting it off by feeding the input with a zero. Equation (2) will reach a value of one in the time interval equal to the average of u(t) over it. The accuracy increases if the interval is very small or u(t) is a constant function. A suitable difference equation is given in (3).

\[ y(t) = \int \frac{dt}{u(t)} \]  
\[ y(n) = \frac{1}{u(n)} + y(n-1) \]

#### A. Triggered Sample-and-Hold (TSH) Block

The output and rate of the plant immediately after the part has been straightened are dependent on \( B_x \) and the previous \( Y \) and \( Y' \), all of which occur over different time intervals. The function of the TSH block is to sample a value at one point in time so that it can be used at another time. Shown in Fig. 10, the block basically samples the input for a small period of time on the rising edge of an enabling pulse, and then holds that value constant at the output until the block is re-enabled. In Simulink, a constant value can be maintained for an arbitrary period of time by building the output of an integrator to a desired value. A T-second timer is created by integrating 1/T.
until it equals one. It is used to control the small time interval in (4) which takes a time average of the input.

\[ y(t) = \int_0^T u(t) \]  
\[ (4) \]

B. Pulse Block

The bend timer must be able to convert \( B_T \) from a value into the timed pulse, \( U \). The pulse block, illustrated in Fig. 11, was designed for this purpose. By incorporating (2), it creates a pulse that has a magnitude equal to the sign of the input and a pulse width equal to the input’s absolute value.

C. Pulse Width Block

The plant must be able to convert \( U \) from a timed pulse back into a value. The pulse width block, shown in Fig. 13, is designed for this, and is the functional inverse of a pulse block.

D. Plant Modeling

The initial rate \( (R_1) \), the bend rate \( (R_2) \), and the resulting rate \( (R_3) \) were tabulated from twenty-five strip chart recordings. \( R_2 \) proved to be a constant and the form of \( R_3 \) was chosen to be a linear combination of \( R_1 \) and \( B_T \). Coefficients were iterated to minimize the sum of the absolute value of the error between actual and calculated rates. \( R_3 \) is calculated in the following equation:

\[ R_3(n) = 0.6030719 R_1(n) - 0.0001182 B_T(n) \]

The plant was modeled to supply an integrator with different rates at different times, forming one piece-wise continuous rate \( Y' \). The first initial rate \( (R_0) \), is supplied until the first bending event, and then the plant toggles between the \( R_2 \) and \( R_3 \). The integrator converts the rate into position \( Y \). Fig. 14 depicts the plant block where \( R_0, R_2 \), the part roundness, and the initial \( Y \) are specified to create a practical simulation.
E. Bend Timer and Decision Modeling

The bend timer was modeled to create a $B_T$-minute pulse when $B_P$ is triggered. The pulse is only triggered if $B_T$ is large enough and the specified sample time has expired since a previous bend. The bend timer also supplies the rate disturbance (D). Fig. 15 depicts the bend timer block where the minimum and maximum $B_P$, $Y_{sp}$, and D are specified to create a practical simulation. The decision block, not shown, creates a bend pulse whenever the error signal is above a specified tolerance limit, $T_L$.

![Bend Timer Simulink block](image)

Figure 15. Bend timer Simulink block

V. CONTROL DESIGN

There are currently many different control structures available, the simplest of which is the PID controller design. For this reason, a PD controller is first applied to the closed loop model to verify its response and stability. It is used as a benchmark for other controllers. Next, a nonlinear PD (NPD) control scheme is introduced. It retains the tuning ease of the PD controller while improving performance. Last, a nonlinear filter, called the tracking differentiator (TD), is introduced as an alternate means of providing an accurate derivative feedback to the controller in the presence of noise, thus improving performance.

A. Linear PID Control

For many reasons, PID control is still used in 90% of industrial applications [3]. There are only three tuning parameters, each having direct physical significance to the error signal, not the model. This makes for easy tuning, without having to spend considerable resources on the construction of a linear model. Linear models are often inaccurate and require re-tuning when the real-world plants they represent are nonlinear and time varying. A control structure that is error-based, and not model-based, is more resilient to model uncertainties [3].

The PID control law in (6) represents the direct physical meanings of the three parameters, where $e$ is the error signal, $K_P$ is the proportional gain, $K_I$ is the integral gain, and $K_D$ is the derivative gain.

$$u = K_P e + K_I \int e \, dt + K_D \frac{de}{dt}$$  (6)

The $K_P$ term is most effective when the error is the largest, which occurs in the early stages of the transient and helps to initialize the transient. The $K_P$ term is most effective when the error is changing the quickest, which occurs in the transient region and helps prevent overshoot by adding damping [4]. The $K_I$ term is most effective when the error is small and changes the least, which occurs in the steady state region where it continues to drive the steady state error to zero.

Although asynchronous, $Y$ is based on two past rate values, thus making the plant behave like a double integrator. Furthermore, the operator intuitively sets a bend time based on $Y$ and $Y'$ when manually controlling the machine. Therefore, a PD controller is chosen as a starting point. The controller was tuned for the smallest possible settling time, overshoot, and steady state error. The closed loop system and benchmark controller gains are shown in Fig. 16.

![Closed loop model with a PD controller](image)

Figure 16. Closed loop model with a PD controller

B. Nonlinear PID Control

Although the PID parameters are easy to tune, they are limited by their linearity [3]. The proportional term is ineffective in minimizing steady state error when the error is small, unless $K_P$ is set very large. This can lead to overshoot and cause the control to saturate when the error is large. Hence, the $K_P$ term must be nonlinear in order to have a consistent effect on the response. Motion profiles are very effective because they attempt to keep the error in a small range, but they are only effective on the initial transient. Consequently, the proportional term in (6) is replaced by the nonlinear term in (7).

$$K_P f_p(e,a,d) = K_P \begin{cases} e \text{ sign}(e), & |e| > d \\ \frac{e}{d^{1-a}}, & |e| \leq d \end{cases}$$  (7)

When $0 < a < 1$, $f_p(e,a,d)$ is small for large errors (minimizing overshoot and saturation) and large for small errors (minimizing the steady state error). The exponential approach is an infinite gain when the error approaches zero, causing numerical issues of bang-bang chattering [4]. Inserting a linear region in the neighborhood of zero solves the problem [3]. When $a > 1$, the control is detuned. When $a = 1$ or $|e| < d$, $f_p(e,a,d)$ is linear.

The derivative term needs only to be effective in the transient region when $\dot{e}$ is relatively large. Since the error changes very little in the steady state, the $K_D$ term basically processes noise and degrades the control [3,4]. Setting $a > 1$ will allow the derivative to be effective when $\dot{e}$ is high (in the transient region) and inconspicuous when $\dot{e}$ is small (in the steady state) [8]. To further minimize the effects of noise, the
linear region is replaced by a dead zone, allowing it to only be operational when $\dot{e}$ is large enough [8]. The derivative term is rewritten in (8).

$$K_D f_D(\dot{e}, a, d) = K_D \begin{cases} |\dot{e}|^a \operatorname{sign}(\dot{e}), & |e| > d \\ 0, & |e| < d \end{cases}$$ (8)

The integral term is most effective in the steady state region when the error is small. Here it ensures that the (otherwise finite) error is driven to zero. Other places in the response, the integral is prone to saturation and windup. There is a 90-degree phase lag associated with the integrator that can push higher-order plants towards instability [3,4]. Sometimes the use of nonlinear proportional and derivative terms can eliminate the need for an integral term altogether. If it is needed, setting $-1 < a < 0$ helps to reduce these risks, making it only operational when the error is small. The integral term is rewritten in (9), and the control law, summarizing the NPD control design discussed in this section, is shown in (10).

$$K_I f_I(e, a, d) = K_I \begin{cases} \int e \operatorname{sign}(\int e), & |e| > d \\ \frac{e}{d^{1-a}}, & |e| < d \end{cases}$$ (9)

$$u = K_p f_p(e, a_p, d_p) + K_I f_I(e, a, d) + K_D f_D(\dot{e}, a_d, d_d)$$ (10)

These fundamental concepts were used to quickly design a nonlinear controller. The nonlinear controller is pictured in the simulation loop in Fig. 17, and the controller parameters are shown. Although the plant is unusual, the NPD concepts previously discussed were effective.

![Figure 17. NPD controller design](image)

**C. Nonlinear Derivative Approximation**

The derivative term of a PID controller is sometimes useless because it can amplify noise to unacceptable levels. The TD, represented in (11), is proposed as a nonlinear alternative to a second order linear-approximate derivative.

$$\begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = -M \operatorname{sign} \left( x_1 - v(t) + \frac{x_1 |x_1|}{2M} \right) \end{cases}$$ (11)

The TD is based on time optimal control of a double integrator and only has one design parameter. Furthermore, the parameter has direct physical significance to the maximum acceleration of the system. The TD is a filter that is not frequency dependent and produces a cleaner derivative to the controller. Fig. 18 shows how the TD is implemented in the simulation model.

![Figure 18. TD closed loop configuration](image)

**VI. SIMULATION**

The PD controller and the NPD controller are simulated in Simulink on the closed loop model. The significance of practical initial conditions and disturbances are considered. Simulation results from the two cases are presented and discussed. Next, the steady state error is compared under various noise and disturbance conditions. Finally, a second order approximation and a TD are implemented in the feedback loop and simulated with heavy feedback noise for both controllers.

**A. Transient Performance**

Simulating the system and individually adjusting plant parameters to emulate various real world conditions verified the design. Fig. 19 graphs the output and rate response for the PD controller. Notice the control action occurs when the rate is at ±$R_2$ (i.e. ±0.01 in this case).

![Figure 19. PD Controller Simulation](image)

The performance measures are defined as follows:

1. The settling time ($T_s$) is the time it takes for $Y$ to be within ±0.01, since this is the general object of straightening.
2. Overshoot (OS) is the maximum $|Y|$ after it has reached zero.
3. Steady state error (Ess) is the maximum $|Y|$ in the steady state region.

The linear and nonlinear PD controllers were tuned to achieve the same $T_s$. The results of the transient responses of
the two controllers for two widely different initial positions are tabulated in Table I, to compare OS and Ess. All units are in thousands except for settling time.

<table>
<thead>
<tr>
<th>Controller Type</th>
<th>.020 TIR Initial</th>
<th>.003 TIR Initial</th>
</tr>
</thead>
<tbody>
<tr>
<td>PD</td>
<td>19.929 0.08344 0.06395 2.230 2.3481 0.09357</td>
<td></td>
</tr>
<tr>
<td>NPD</td>
<td>19.929 3.1651 0.01354 2.220 0.0000 0.06848</td>
<td></td>
</tr>
</tbody>
</table>

The linear controller exhibited better overshoot than the nonlinear controller, partially due to a few less tuning parameters. On the other hand, the NPD controller outperformed the PD controller in every other way. When the initial position was changed by a factor of almost seven, the performance of the PD controller decreased, while the overshoot of the NPD controller improved. This is due to the nonlinear gains, which make the control more sensitive to small errors and transients.

B. Disturbance Rejection

Next, the results of the steady state error of each controller for various disturbance combinations are tabulated in Table II.

<table>
<thead>
<tr>
<th>Controller Type</th>
<th>Disturbance Combinations</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>D</td>
</tr>
<tr>
<td>PD</td>
<td>1.344</td>
</tr>
<tr>
<td>NPD</td>
<td>0.377</td>
</tr>
</tbody>
</table>

The two controllers exhibited similar responses to the feedback noise (Nf), which is examined further with the use of the derivative approximates in the feedback path. The nonlinear control largely outperformed its linear counterpart in the way of rate disturbance (D) and noise (Nd). This is a good indication that the nonlinear controller is more robust on this type of plant. The overall performance is better with the NPD controller.

C. Derivative Approximates

Finally, the feedback noise was simulated through the derivative approximates and the results are tabulated in Table III.

<table>
<thead>
<tr>
<th>Controller Type</th>
<th>NF Disturbance Present</th>
<th>All Disturbances Present</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>None</td>
<td>Linear Apx.</td>
</tr>
<tr>
<td>PD</td>
<td>1.916</td>
<td>1.789</td>
</tr>
<tr>
<td>NPD</td>
<td>1.920</td>
<td>2.052</td>
</tr>
</tbody>
</table>

Both of the controllers improved in performance with the use of the TD in the feedback path, and degraded with the implementation of the second order approximate. There are two possible reasons for this occurrence. The first is that the second order linear approximate forces the trade-off between noise rejection and performance. The second reason is that the tracking differentiator is a nonlinear filter that limits the maximum acceleration of the system and is not a frequency dependent filter. Both filters were easy to tune.

VII. Conclusion

By and large, the nonlinear control methods that were discussed have a performance advantage over the linear ones, while maintaining the ease of tuning. Both controllers are sufficient for physical implementation, but the nonlinear PD controller has better performance and disturbance rejection overall. The results show that the tracking differentiator has a performance advantage over the second order approximation by providing a cleaner derivative in the presence of noise, and maintains its tuning ease. The second order approximate derivative is not easily tuned in some cases.

Since many of the physical limitations of the plant were considered in the simulation model, the nonlinear control strategies outlined in this text are viable and ready for hardware-in-the-loop implementation. Based on the steps taken thus far, the following areas are recommended for further study:

1. The PLC algorithm that validates the sensor output against a set of rules can be designed.
2. The mathematical model that determines the resulting rate by a linear combination of variables can be studied further to find other linear combinations that produce less error to the raw data.
3. Other nonlinear controllers can be designed and simulated. The active disturbance rejection discussed in [3] can be implemented to reject model uncertainties by using an extended state observer.
4. The process can be modeled in discrete time and/or with different software. Writing the entire plant and bend timer in C may simplify the simulation model. From this, a class of problems can be clearly studied.
5. The process can be investigated and modeled as a finite state machine.
6. An alternative nonlinear function can be studied as a replacement to the function outlined in (7). The function defined as: \[ f(u, a) = (1 - e^{-|a|}) \text{sign}(u) \]
   is continuous and boasts only one design parameter.

REFERENCES