Iterative learning control for tailor rolled blanks

Iterative Learning Control

for Tailor Rolled Blanks

Zhengyang Lu

DCT 2007.066

Master Internship Report
Supervisors: Prof. Maarten Steinbuch (TU/e)
Camile Hol (Corus)
Eindhoven University of Technology
Department of Mechanical Engineering
Control System Technology

Eindhoven, July, 2006

Contents

1. Introduction

Tailor Rolled blanks, which are introduced in the car manufacturing industry, are rolled steel production with varying thickness. Great flexibility in this design is offered by TRB, because the thickness and hence the strength of the components are locally increased. In addition, it helps to reduce the overall weight of the components.

The rolling of TRB: after de-coiled, the strip enters the rolling machine. It is rolled by the working roll, which is connected to the motor by the back rolls. The screw position is controlled to make the exit strip have varying thickness. After that, the strip will be coiled.

Since the same blank is rolled subsequently many times, the process is repetitive in nature. By applying iterative learning control, we can benefit from this repetitive nature. A typical specification is a thickness transition from 2 to 1.5 mm with a tolerance on the thickness of 50 μm. A plot of the TRB profile is given in next page.

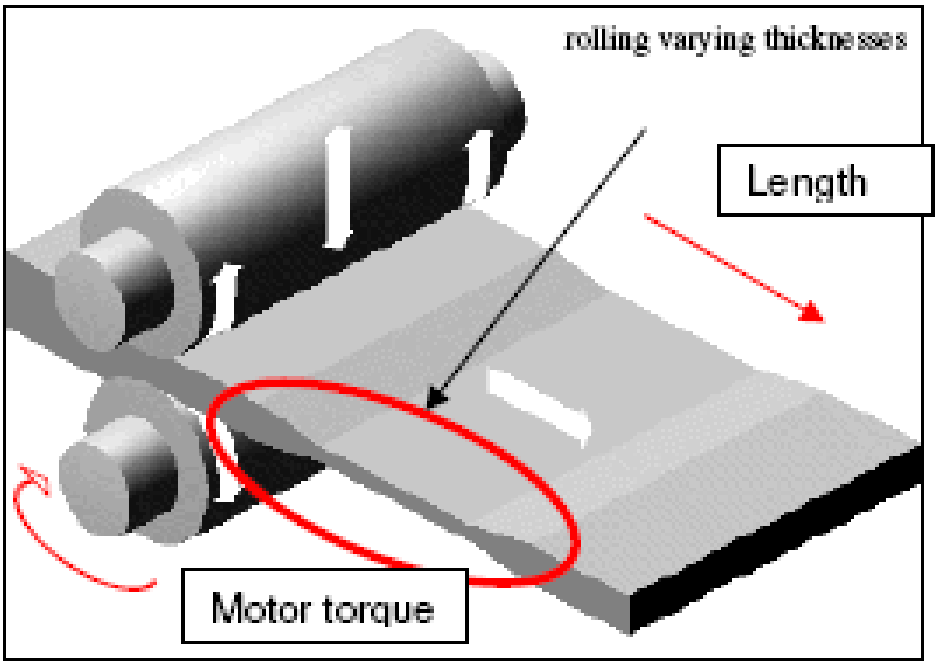

Figure 1. Rolling process

In this project, first, it is investigated if iterative learning control can be applied to reduce the thickness error in TRB. Since the relation between thickness and length is very important, the ILC is scheduled on length basis rather than a time basis.
Second, Torque adaptation is applied aiming to get a relative constant rolling speed, thus a better process condition.
Last but not least, the length uncertainty is analyzed, and a suitable level of uncertainty is implemented in the simulation model, to investigate if the system is robust enough for the uncertainty.

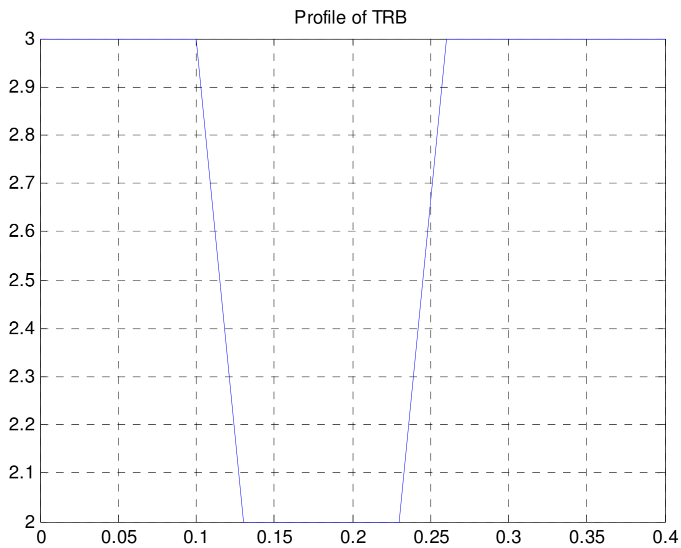

Figure 2. Profile of TRB

The figure above indicates that the length of each blank is 0.4 m, while the thickness varies from 3 mm to 2 mm.

2. Simulation model

To test a real-time control design a model of the system to be controlled is needed. The model comprises the basic relevant input and output signals with the static and dynamic behavior sufficient to test the designed controller before its real-time implementation.

The main part of the model is the screw position controller, the mill, and the deformation module next to the mechanical dynamics of the stand drive. These will be explained in more details below.

In the following paragraph, a basic introduction to the offline simulation model is given.

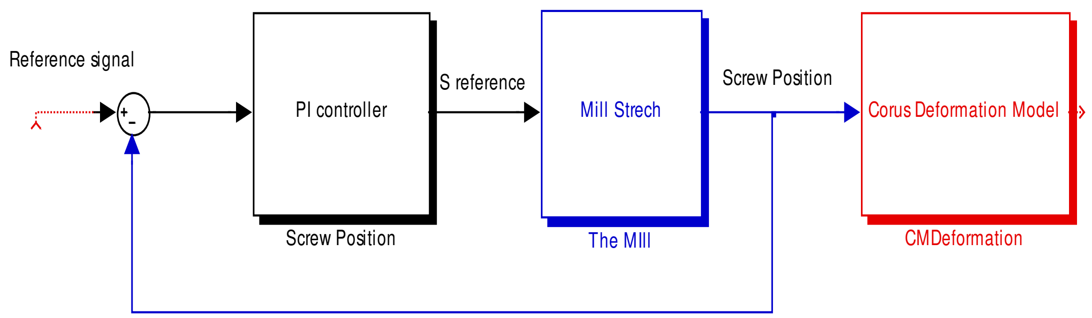

Figure 3. Offline Simulation Model

This basic layout shows how the model is constructed: the physical interaction between different subsystems is clearly showed.

The model consists of three blocks: named Controller, the Mill, and the Deformation boxes.

The controller is a conventional PI controller to control the varying thickness of the blanks by controlling the screw position, The Mill, which is regarded as the plant, represents all the dynamics of our system.

The Corus Deformation Model is defined as an S-function in Matlab, and is a static mapping. It calculates the necessary roll force, resulting entry and exit speeds and rolling torque as a function of entry and exit thickness, entry and exit tension stress and work roll speed as main input signals.

3. Iterative Learning Control

Iterative learning control is a method that has been developed from a practical environment, and there are many different ILC design techniques were developed.

First of all, to make the next paragraphs more understandable, the definition of two terminologies is given: the time period in which the system has to perform its repetitive task is called a trial and each task is called iteration.

Convergence is obviously the most important issue of Iterative Learning Control. In iterative learning control, experience from previous iteration or repetition is used such that the ILC system will gradually learn the control action that will result in prefect tracking. One could use the information from previous iteration to generate a new input u_k for the current iteration (where k is the iteration number), so that the tracking error will go to zero as the iteration numbers increase.

However, in real industry, what we want is that the tracking error will be decreased to be smaller than required tolerance in an acceptable numbers of iterations or a period of time. Less numbers of iterations or time indicates that the iterative learning control is powerful and efficient, because it will decrease the number of unqualified products, which have to be thrown away or re-manufactured. In other words, less number of iterations improves the production efficiency and lowers production cost.

The basic working principle of ILC that we have applied in for the Tailor Rolled Blanks can be illustrated by the figure 4.

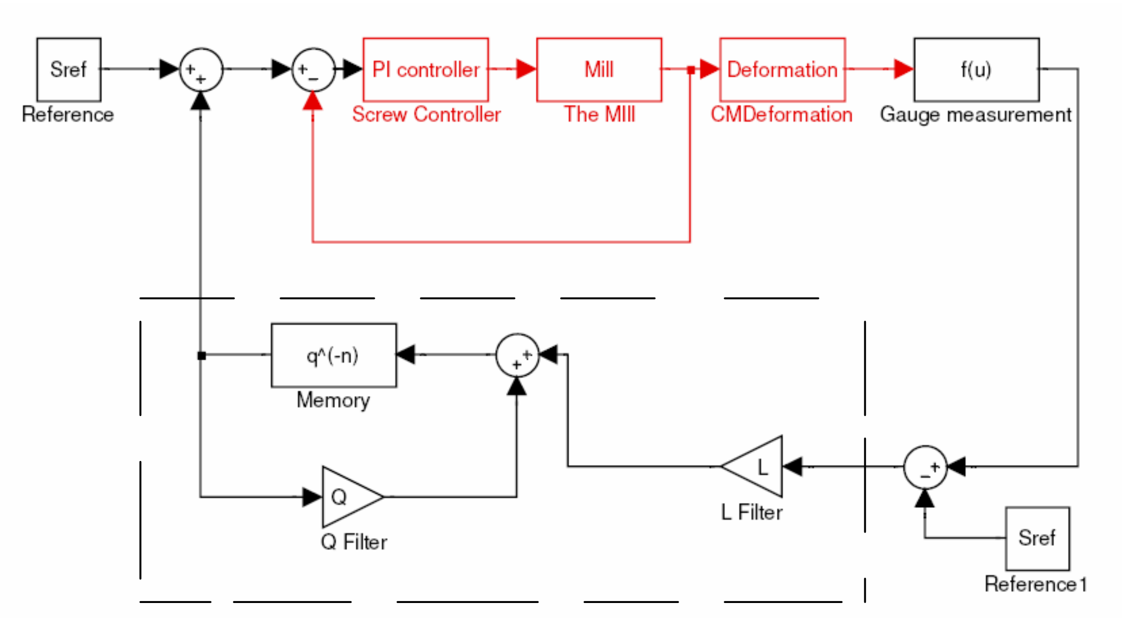

Figure 4. Feedback Loop and ILC Feedforward Loop

In the figure above, the whole loop consists of two parts: the offline simulation model and the iterative learning controller.
The offline simulation model, as we described before, is a PI controlled feedback loop. Its input is the feedforward compensation, which comes from the output of the iterative learning controller. And its output, which comes from the CMDeformation, is the thickness of the exit strip. Because it is impossible to measure the thickness of the rolled strip instantly, a delayed measurement is executed. The output of the gauge measurement is the so called “gauge meter”---- the thickness of exit strip on length basis. When the gauge meter is compared with the reference signal, difference is obtained as the input of the iterative learning controller.

The iterative learning controller comprises 3 parts: L filter, Q filter, and Memory, with error as its input and adaptation signal as output.

L filter: it could be inverse of the plant (with input (error) and output (delta u)) at low frequency, because the dynamics of the plant at high frequency is difficult to obtain. The L filter has no influence on the steady state performance as long as it provides a stable ILC. The coefficient of L filter is the learning gain. Larger value of learning gain will fasten the convergence, but it will also amplify the error, so there is always a compromise between fast convergence and stability system.

Q filter: it is the robustness filters to compensate for the influence of the model uncertainty at high frequency. A low pass (Butterworth) filter can be used.

Memory: the memory is used to compute and store the future error of current trial, which will be used to compute the control action for the next trial. This means that the ILC is non-causal: the future inputs can influent the current output.

However, our ILC algorithm can be described as below:

$$U_{k+1}(t+1) = Q * U_k(t) + f(e_k(t+1))$$
$$f(e_k(t+1)) = L * e_k(t+1);$$
$$where \quad L \quad is \quad \alpha * averaging; \quad \alpha \quad is \quad learing \quad gain;$$
$$Q \quad is \quad identity.$$

The ILC works in this algorithm:

As we have defined before, the time period in which the system has to perform its repetitive task is called a trial and each task is called iteration. In our case, time period is replaced by blank length, because the ILC is scheduled on a length basis.

In each trial, $U_{k+1}(t+1) = Q * U_k(t) + f(e_k(t+1))$. The new control signal for a new trial is the sum of the control signal for previous trial with Q filter, and a function of future error of previous trial. The function is a multiplication of future error and L filter. And the error equals $e = y_{ref} - y_{out}$, which is the difference between the reference and the actual output. Both the error and input are stored in the memory of the ILC, which will be used offline for calculating the new feedforward signal for the next iteration, according to user defined adaptation law.

Briefly, ILC is an offline method, or better described as open-loop feedback. It feeds back information as much the same way as conventional methods do: e.g. the error, the position etc. In addition, as offline method, ILC allows “noncausal” operations: the “future” of past trials can take part in the construction of the current control action.

$$U_{k+1}(t) = U_k(t) + f(e_k(t+1))$$

where k defines the trials, and t defines time.

The picture below maybe better describes what future error of past trial is, the difference between ILC and conventional feedback control.

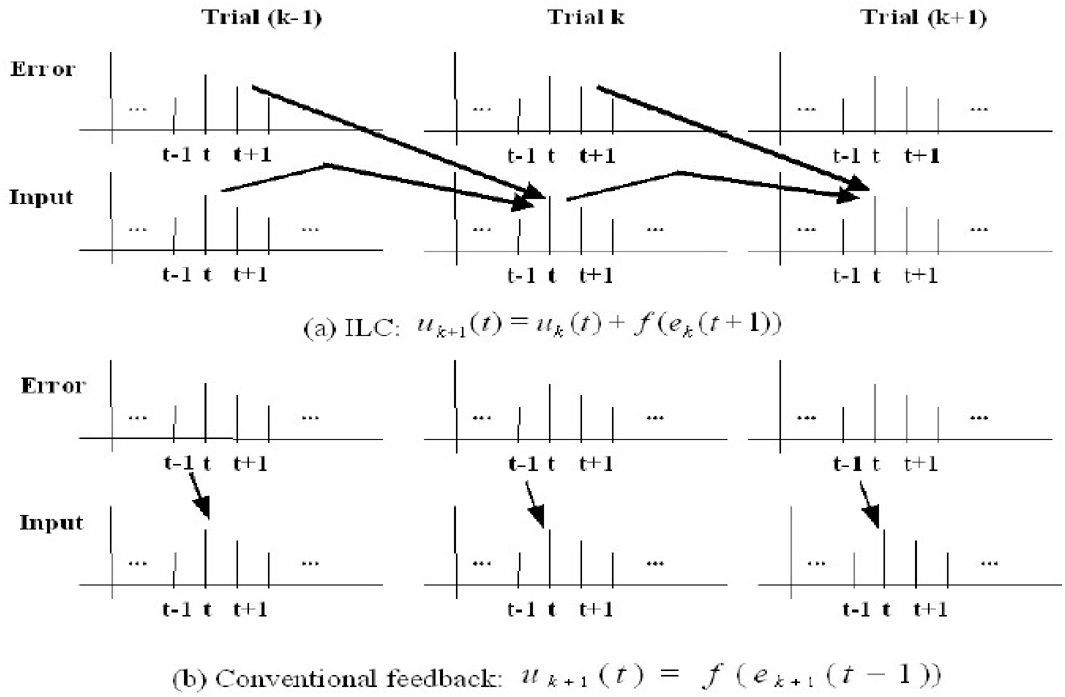

Figure 5. Difference between ILC and conventional feedback control.

4. Simulation

First of all, we define the control framework as below:

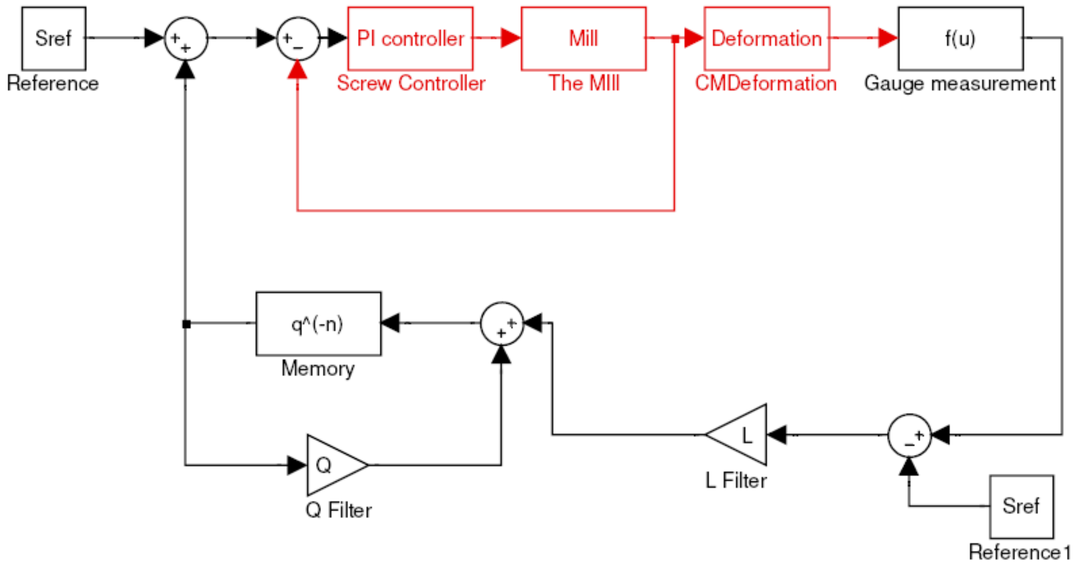

Figure 6. An overview of control framework

In the figure, the part of red blocks is the offline model, as we have already introduced.

Sref is the desired output, or reference signal.

“Gauge Measurement” is where the thickness of the blanks is measured. Because there is a distance between the work rolls and the gauge meter measurement for gauge, we get a delayed measurement: the time delay is about 3.3 sec. it changes due to the fluctuation of the exit speed.

4.1 Refined adaptation Law

Remember that in ILC the learning law is:

$$U_{k+1}(t+1) = Q * U_k(t) + f(e_k(t+1))$$

It can be redefined in our system like:

$$U_j^{adapt}(k+1) = U_j^{adapt}(k) - \alpha * e$$

Where k defines segment index; α is the learning gain

In the algorithm below, the reference gauge of the horizontal segments are recursively adapted using the measured exit profile, which is illustrated in Figure 7. The x- and y- coordinates are the length and gauge respectively. The adaptation corrects the reference profile in the (k+1)th blank using the measured gauge of the previous blank k.

The refined adaptation law in more clear way is as:

$$h_j^{adapt}(k+1) = h_j^{adapt}(k) - \alpha(h_j^{avg}(k) - h_j^{ref}(k))$$

α can be tuned for convergence speed.

The gauge measurements within a segment are average, resulting in the gauge average $h_j^{avg}(k)$ given by:

$$h_j^{avg}(k) = \frac{1}{L_j} \int_{L_j} h(x)\,dx \,.$$

Where L_j is the length from the k(th) sample point to the (k+1)th sample point. There are 100 sample points on the three horizontal segments of each profile, and this adaptation law is only applied on horizontal parts. And this method of calculating $h_j^{avg}(k)$ is only used for the starting point of each horizontal segment. A new way of calculating $h_j^{avg}(k)$ is defined for eliminating the phase shift. Phase shift is the change in phase of a periodic signal with respect to the reference. In this system, the phase shift was introduced by the one direction average: average the measurement gauge from k(th) sample point to (k+1)th point, and take this average value as the measured value for k(th) sample point.

$$h_j^{avg} = \frac{1}{(L_{j-1} + L_j)} \int_{L_{j-1}}^{L_j} h(x)\,dx\,;$$

Here, each averaging starts from (k-1)th sample point and ends at (k+1)th point. The profile is corrected with these averages according the adaptation law, resulting in the green curve in figure 7. These adaptation gauge $h_j^{adapt}(k+1)$ are used to compute the screw position for the next (k+1)st blank.

Take the second horizontal segment for example, the new adaptation (the green one) is larger than the old adaptation (the blue one), since the realized averaged gauge (the red one after averaging) was smaller than the reference. The reference for the second horizontal segment will therefore be increased, compared with it at previous blank, thus a smaller error is expected.
This adaptation law is applied in length basis instead of time basis.

The adaptation law is clearly showed with the help of figure 7.

The

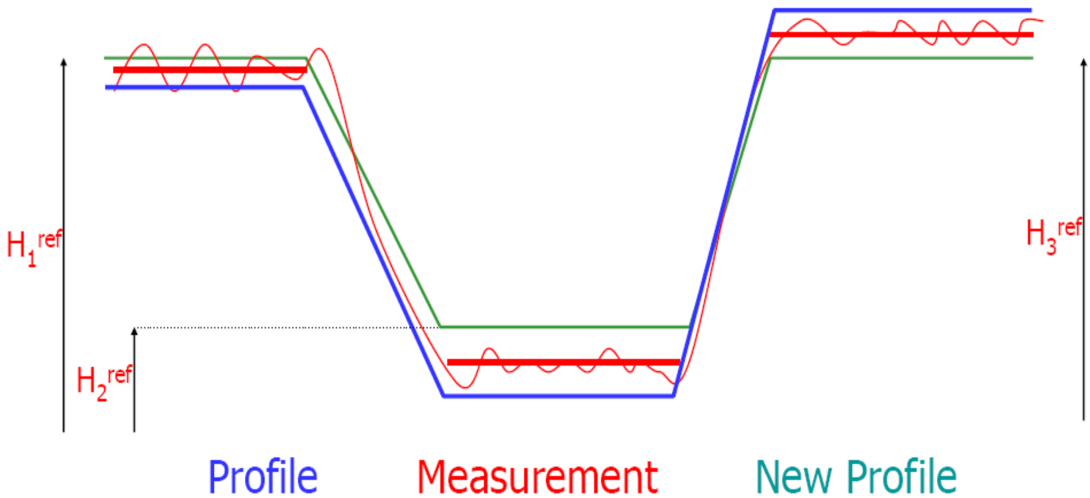

Figure 7. Scheme of Profile Adaptation

Stability Requirement

The stability requirement that we get for this system is:

$$\| Q - L(PS) \|_{\infty} < 1$$

This stability requirement is obtained by applying **Small Gain Theory**.

It is similarly applied in our control system as:

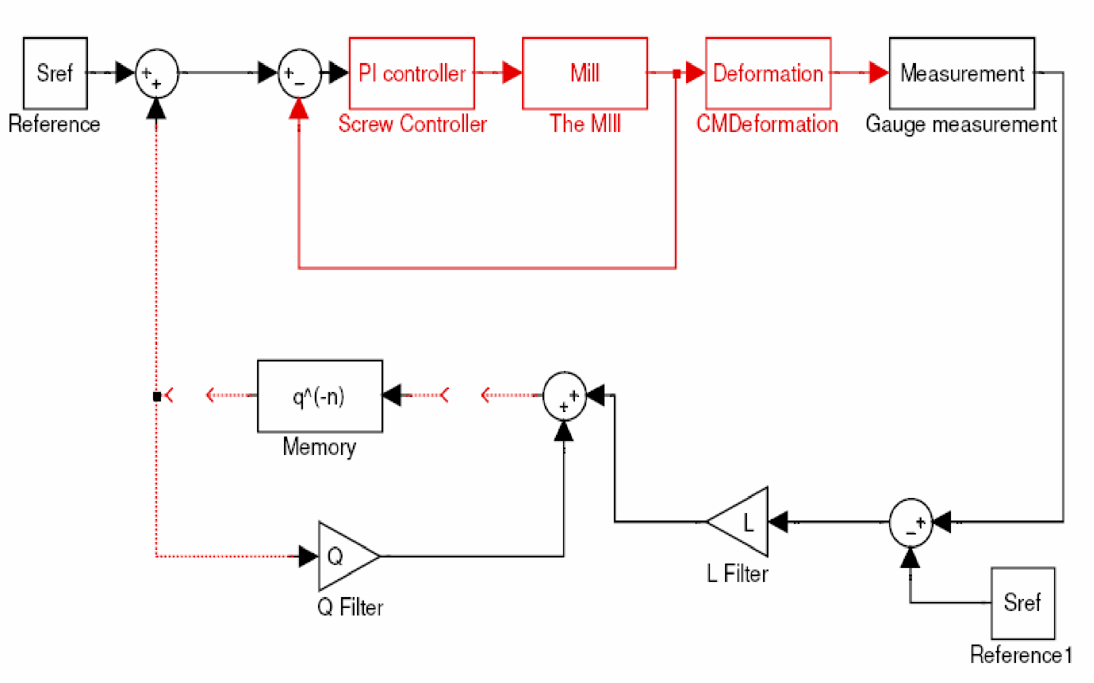

Figure 9. Disconnected control system

Simulation results:

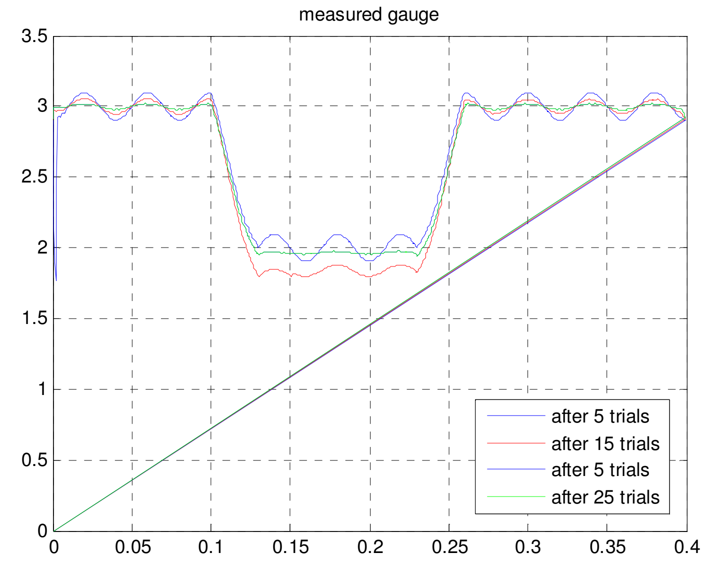

Figure 10. Simulation results for measured gauge

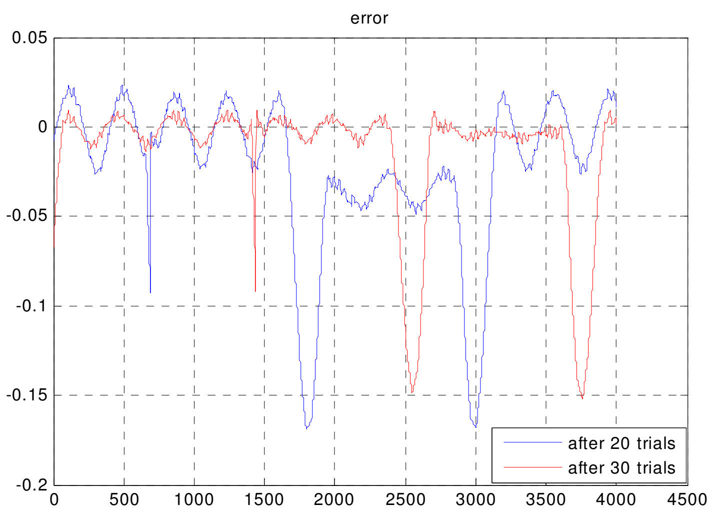

Figure 10. Simulation results for error

The figures above clearly show that the ILC improves the performance of the system after trials: the gauge is getting close to the reference, and the error is getting smaller. The peaks in the error figure are corresponding to the two slope parts of the profile, because there is no adaptation law applied. That means that they can not eliminate the repetitive disturbance, thus a relatively larger error here is reasonable.

4.2 Torque adaptation

The goal of the torque adaptation is to get a relatively constant rolling speed, thus a better rolling condition is obtained. By applying Newton's law (or better say "torque balance"), the equation is made as below:

$$I_{inertia} \dot{V}_p = -(T_{adapt} + T_{ff}) + Rolling\_Torque; \quad (1)$$

Move the adaptation torque to the left side of the equation:

$$T_{adapt} = Rolling\_Torque - T_{ff} - I_{inertia} \dot{V}_p; \quad (2)$$

T_{ff} is the feedforward torque given by the feedforward controller as currently used in the TRB control system. It is a linear function of the reference screw position s_{ref} as follow:

$$T_{FF}(t) = (K_k + C)s_{ref}(t), for \quad all \quad t > 0; \quad (3)$$

Where K_k the scaling factor of each trial, and C is a constant offset.

The gain for this feedforward signal was initially set to be K=20kN/mm, later an adaptation law is applied to make this value gradually converge to the optimal value, by looking at decreasing the maximum speed variation of trial after trial.

The principle of the adaptation is: Initially the scaling factor is set K=20kN/mm, which is obviously not large enough. In addition, we observe that by increasing this K, the maximum speed variation of each trial decreases, until K is increased to a certain value $K_{optimal}$. Because $K_{optimal}$ is not exactly known, the adaptation is used to make the K increase to $K_{optimal}$ automatically, by comparing the maximum speed variation of trial k-1 and trial k.

For example, if by applying $K_k = (1+\beta)K_{k-1}$, $V_{variation}(k) < V_{variation}(k-1)$ is obtained, $K_{k+1} = (1+\beta)K_k$ will be applied for next trial. The $K_{optimal}$ is achieved when $V_{variation}(k) < V_{variation}(k-1)$ doesn't hold anymore. β is a user defined constant value to determine how much the scaling factor increases (how large the step is) for each trial and it can be tuned for faster or slower convergence to $K_{optimal}$. $V_{variation}$ is the maximum speed variation of each trial.

After this, the T_{ff} is approximately the same as Rolling Torque, with slightly varying difference. If T_{adapt} could converge to this difference, the term $I_{inertia}\dot{V}_p$ in equation becomes 0. Because $I_{inertia}$ as a constant value is not zero, the $\dot{V}_p$ will be zero. Then a constant rolling speed is obtained.

The simulation result is shown as:

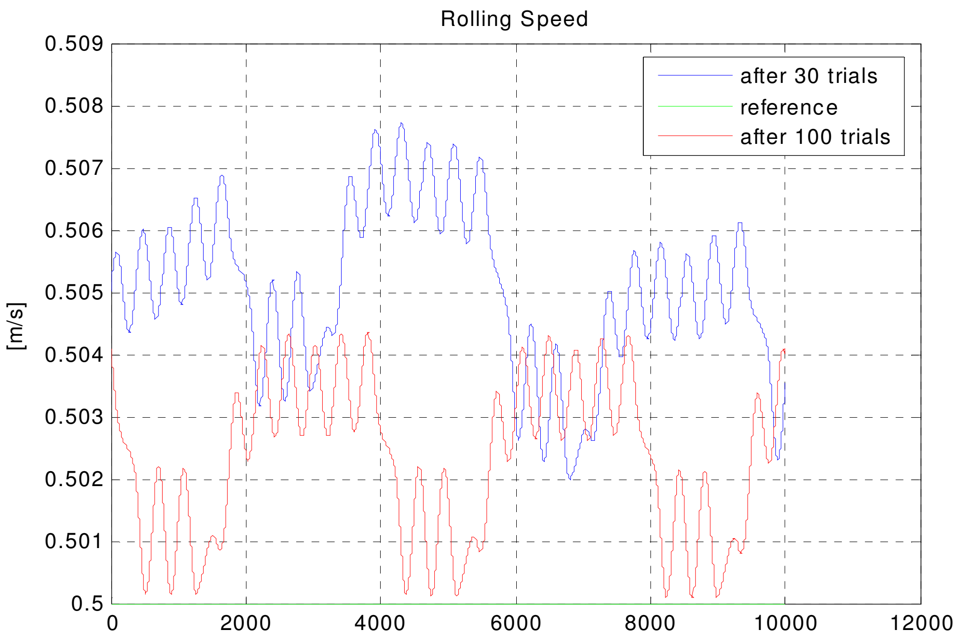

Figure 11. Rolling Speed Variation

Figure 11 clearly shows the decrease of the rolling speed variation. The green line is reference rolling speed 0.5 [m/s]. After 30 trials (the blue curve) the speed variation is reduced to 0.008 [m/s], and after 100 trials it decreases to about 0.004 [m/s], reduced by 50%. That is no more than 1% speed variation, so a relatively constant rolling speed is achieved.
Although the speed variation decreases significantly by the Torque adaptation and Feedforward Torque, a large improvement in the tracking of the gauge reference is not observed.

4.3 Length uncertainty

It is always very important that the system is robust to system uncertainty. Especially in simulation, a model of a system never exactly captures all system dynamics.

In our case, the most important uncertainty is probably (blank) length uncertainty: the uncertainty to detect the exact start and end point of each horizontal segment. This results in a potential misfit of the length-coordinate y which causes inclusion of wrong values from the transition section.

However, after analyzing the real time data (collected from real-time data of the rolling mill), we feel that the length uncertainty is not as large as we thought it could be. In addition, we found that the length uncertainty is mainly caused by slip (need to be further proved later). A small uncertainty is implemented in the system, and the system still performs well.

Simulation results:

First we checked the difference between entry strip volume and exit strip volume by the formula given as below:

$$\mathrm{Per}_{\mathrm{variation}} = \frac{\mathrm{EntryStripLength} * \mathrm{H} - \mathrm{ExitStripLength} * \mathrm{h}}{\mathrm{EntryStripLength} * \mathrm{H}}$$

Entry and exit variables are distinguished by a rule that a capital letter is used for the entry side and lower case for the exit (e.g. H is entry thickness and h is exit thickness).

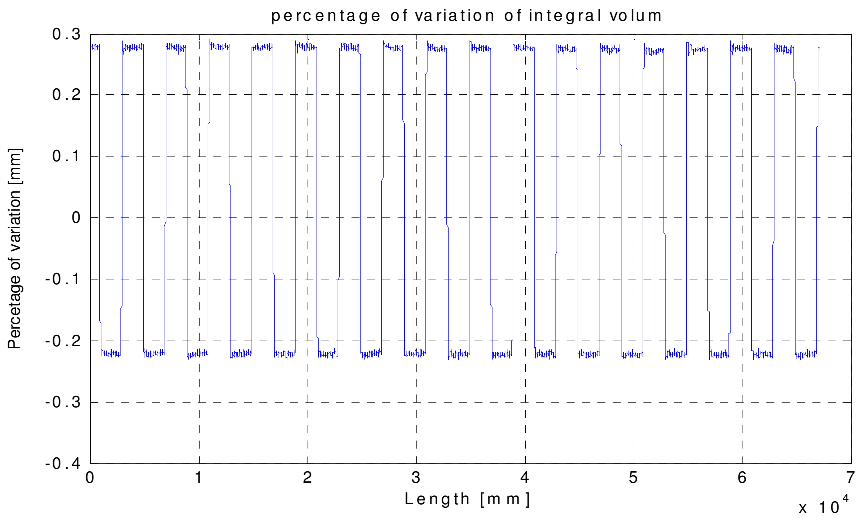

Figure 12. Percentage of variation of stripvolume

The difference between roll speed and the exit strip speed varies over the profile which results in varying slip. The slip is calculated from the exit strip speed and the work roll speed with the following equation:

$$slip = (v_{strip} - v_{roll}) / v_{roll}$$

The variation in slip during TRB rolling shows that TRBs can only be produced according to specification on a mill that is equipped with an accurate strip speed measuring device.

The following figure shows how large the length uncertainty is, after analyzing the real time data.

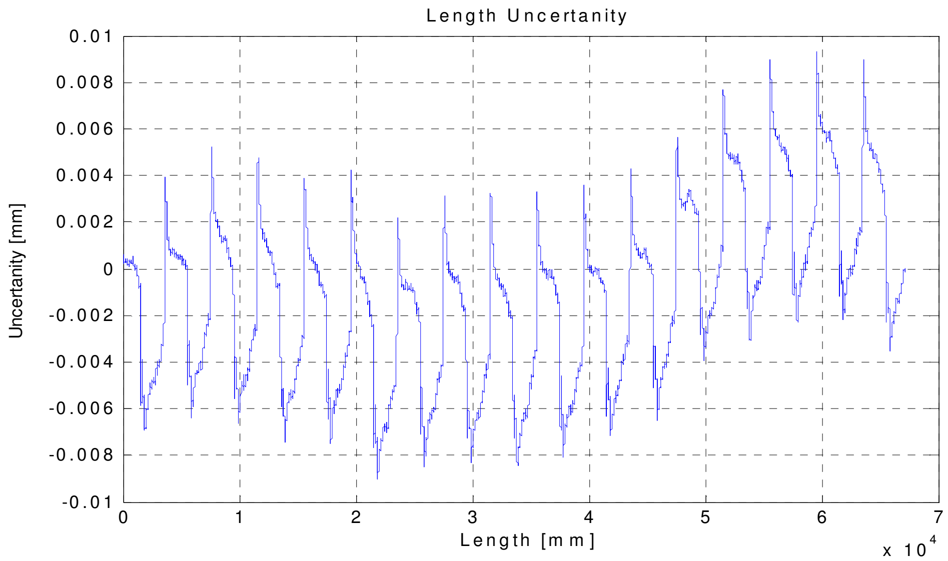

Figure 13. Length Uncertainty

Two times larger uncertainty (length basis) was installed in the simulation model, and the system still performs well. All the results showed before in this report were with this amount of uncertainty.

5. Conclusion and Recommendation

The iterative learning control method (the refined profile adaptation) proved to be very useful and helpful for TRB: it helps us to reduce the error in thickness significantly and results in a nice tracking of the reference gauges.

The simulation results of the torque adaptation are very satisfying: it leads to very small fluctuation (almost constant) in the roll speed and tension force. This torque adaptation will be applied next time in real time at the mill.

Some of the future works are: firstly, prove soundly that the slip is main reason for the length uncertainty. Secondly, considering the relatively larger error at the slope parts of the blank profile, a simple adaptation can be applied later, when it is necessary. Lastly, redefine the blank profile to decrease the error at the transition of continuous blanks, by adding the last horizontal segment of the last blank and the first horizontal segment of the next blank to the profile of the current blank.

6. Reference

[1] Iterative Learning Control with applications to a wafer stage
B.G.Dijkstra, Phd thesis, Concept Version 1.1

[2] Iterative Learning Control----a critical review
Mark Verwoerd, 2005

[3] Iterative Learning Control with Wavelet Filtering
R.J.E. Merry, Master Thesis, TU/e DCT 2005.54

[4] Iterative Learning Control on an Inkjet Printhead
Niels Johannes Maria Bosch, Master Thesis, TU/e DCT 2005.80

[5] Corus internal references

[6] Iterative Learning Control and Repetitive Control for Engineering practice
Richard W. Longman, 2000