PERFORMANCE COMPARISON OF TWO CONTROLLERS ON A NONLINEAR SYSTEM

Kenneth O.M. Mapoka[1], Dr Tranos Zuva[2], Howard Masebu[3] and Keneilwe Zuva[4]

[1]Department of Agricultural Engineering and Land Planning, Botswana College of Agriculture, Gaborone, Botswana

[2]Department of Computer Science, University of Tshwane, Pretoria, South Africa

[3]Department of Agricultural Engineering and Land Planning, Botswana College of Agriculture, Gaborone, Botswana

[4]Department of Computer Science, University of Botswana, Gaborone, Botswana

ABSTRACT

Various systems and instrumentation use auto tuning techniques in their operations. For example, audio processors, designed to control pitch in vocal and instrumental operations. The main aim of auto tuning is to conceal off-key errors, and allowing artists to perform genuinely despite slight deviation off-key. In this paper two Auto tuning control strategies are proposed. These are Proportional, Integral and Derivative (PID) control and Model Predictive Control (MPC). The PID and MPC controller's algorithms amalgamate the auto tuning method. These control strategies ascertains stability, effective and efficient performance on a nonlinear system. The paper test and compare the efficacy of each control strategy. This paper generously provides systematic tuning techniques for the PID controller than the MPC controller. Therefore in essence the PID has to give effective and efficient performance compared to the MPC. The PID depends mainly on three terms, the P (K_P) gain, I (K_I) gain and lastly D (K_D) gain for control each playing unique role while the MPC has more information used to predict and control a system.

KEYWORDS

Auto Tuning, Nonlinear system, Control law, PID & MPC

1. INTRODUCTION

There are two control strategies proposed in this paper. These are Auto tuning PID and auto tuning MPC controllers. The auto tuning methodologies were respectfully incorporated on PID and MPC controllers. These control strategies ascertains stability, effective and efficient performance on a nonlinear system. This paper generously provides systematic tuning techniques for the PID controller than the MPC controller. Therefore in essence the PID has to give desired performance compared to the MPC. This paper however scrutinizes the conundrum. The PID depends mainly on three terms, the P (K_P) gain, I (K_I) gain and lastly D (K_D) gain for control, inherently playing unique roles while MPC have more information or variables enabling prediction and control. We proposed one nonlinear system in this paper. The purpose of this nonlinear system was to help analyse and draw conclusions based on the responses gotten from simulations when compensators were administer. Thus, this paper presents auto tuning algorithm for MPC and many systematic tuning techniques for auto tuning PID controller.

DOI : 10.5121/ijccms.2013.2302

We proposed Ziegler Nichols (Z – N), Cohen Coon (C – C), Integral Time of weighted Absolute Error (ITAE) and Internal Model Control (IMC). these are systematic tuning techniques for the undisputed PID controller. These techniques auto tune the PID controller based on the first order plus time – delay (FOPTD) of the system. All three parameters of the PID tune simultaneously, together with their time constants $T_I \; and \; T_D$ during simulations.

1.1 Proposed Nonlinear System

We propose the following dynamic nonlinear system for this study. Matlab Simulink tool box enables to administer the nonlinearity to the system.

$$G_p(s) = \frac{0.09625\, s^2 + 0.3396\, s + 0.09806}{0.4208\, s^3 + 0.4608\, s^2 + 0.1659\, s + 0.01968} \qquad (1)$$

Corresponding discrete transfer function with sampling time of one second is expressed as:

$$G_p(z) = \frac{0.4398\, z^2 - 0.2558\, z - 0.04694}{z^3 - 2.086\, z^2 + 1.448\, z - 0.3345} \qquad (2)$$

Also can be expressed as

$$G_p(z) = z^{-1}\frac{0.4398 \; - 0.2558\, z^{-1} - 0.04694 z^{-2}}{1 - 2.086\, z^{-1} + 1.448\, z^{-2} - 0.3345\, z^{-3}} \qquad (3)$$

$G_p(z)$ Represent the discrete time system

2. AUTO TUNING PID AND MPC

The main aim of this section is to design a feedback controller to automatically control the system subjected to nonlinear adversity. Auto Tuning means any system controller inherently exhibiting self-monitoring or tuning and adaptation characteristics. The controller self-monitors and adjusts its parameters continuously to ascertain effective performance of the system regardless of irregularities or any other unforeseen internal or external disturbances. The auto tuned controller dictates system operations, for instance, in case of misfortune to the system the controller can halt the operation. Thus, Auto tuning methodologies incorporates the aspect of adaptation and power to dictate the operation success.

Various systems and instrumentation use Auto tuning method. For example, an audio processors, designed to control pitch in vocal and instrumental operations. Its main aim was originally to conceal off-key errors, and allowing artists to perform genuinely despite slight deviation off-key. On unrelated example above, Fractional order calculus (FOC) is also an example of a controller that utilizes Auto tuning principle [1].

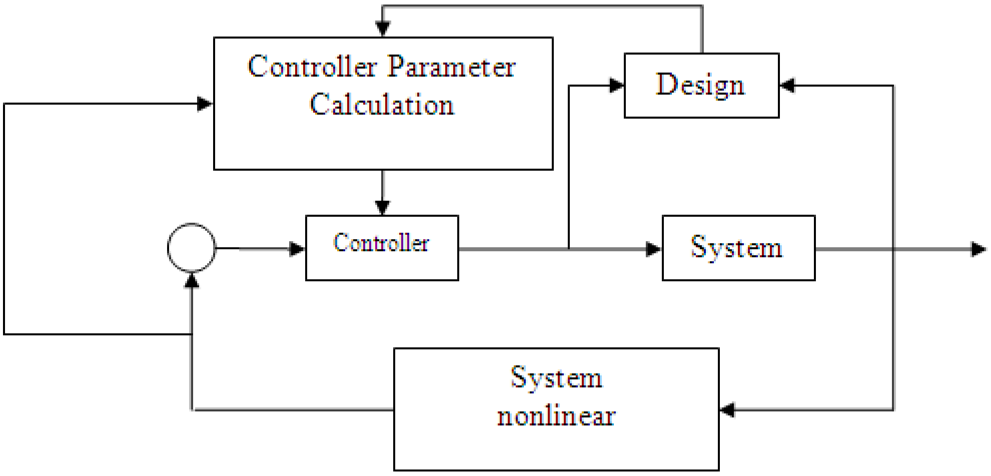

Figure 1. Overview of an Auto Tuning Controller

The block diagram in figure 1 holds performance criterion for the system and controller parameters on an embedded software i.e. Matlab code. The controller presents the control law implemented for the system i.e. we consider all control parameters and nonlinearity time varying.

3. TUNING THE PID CONTROLLER

The aim is to design PID controller that takes into account the system uncertainties, disturbances and high frequencies. Firstly, the phase margin and gain cross over frequency has long been an integral tool to measure robustness on the system. The robustness in the gain of the plant makes the system to obtain desirable transient responses. We propose a modified PID controller which handles high frequencies on the system well using the band limit frequency and when properly tuned it provides efficient desirable transient responses and robust performance. Modified PID control law;

$$(s) = K_p + \frac{K_i}{s} + \frac{K_d s}{\beta s+1} \quad \text{OR} \quad G_c(s) = K_c\left(1 + \frac{1}{\tau_I s} + \frac{\tau_D s}{\beta s+1}\right) \tag{4}$$

K_c ,τ_i *and* τ_d Present the proportional gain, integral and derivative times respectively. Incremental of the K_c value causes the system to respond quicker with inherent oscillations, while increasing τ_d yields a slower yet stable system (induces more damping), and decreasing τ_i provides a faster response with possible oscillations therefore the decrement of the term has to be monitored to avoid unintended oscillations. Let $\alpha = \frac{\tau_D s}{\beta s+1}$, represent the band limited frequency. This derivative term with a time filter, β, restrict the high frequency gain to a specified value. A constant value β could be any value ranging from 10 – 90% of the derivative term. Because of its simplicity in the form of structure, it has prevailed to solve control problems such as set – point regulation and disturbance attenuation.

The PID controller operates on three basic gains namely proportional gain (K_P) integral gain (K_I) and the derivative gain (K_D). Each gain has a definite function on stabilizing the system. Functions of each gain are outline below [2]

K_P – Proportional to the error or change in measurement and it determines the settling time of the system by increasing or reducing the process overshoot

K_I – This gain basically eliminates offset. It reduces steady state error to zero or keeps the error margin within the acceptable range for linear and nonlinear processes

K_D – Proportional to the rate of change of the measurement or deviation, therefore its function is to anticipate future errors by linear extrapolation. This function inhibits more rapid changes of the measurement than proportional action. It triggers the controller gain to move the unintended way when the measurement gets closer to the set-point. Usually it is used to prevent overshoot by stabilizing loops and adding phase lead there achieving more controller gain.

These terms are combined together to yield a control signal which then controls the system. Higher – order dynamics of system prevent the use of high proportional gain and the derivative gain provides damping hence slowing down the transient response [2].

4. EMPIRICAL TUNING RULES

T Industrial applications mainly use Empirical Tuning techniques/rules to adjust PID controller parameters [3]. A simple model typically first order plus dead time (FOPTD) model (equation 16) is commonly used as an initial benchmark to compute or estimate most of the empirical tuning rules for the PID controller. To get the best desired response, the system is approximated to a FOPTD or second order if the system is of higher order and the rules are obtained using this FOPTD or second order. The FOPTD equation is expressed as follows:

$$G(s) = \frac{K}{\tau s+1} e^{-s\tau_d} \qquad (5)$$

where, K, is the open loop gain of the system, τ, is the time constant and τ_d, is the dead time. One common procedure used by various researchers to estimate the parameters of the FOPTD model is called Reaction Curve procedure (PRC) [3]. PRC operates by imposing a step input on a system without a controller and the resultant is that of a FOPTD model approximation. The parameters of the FOPTD model can then be deduced from the step response plot as shown in Figure 2.

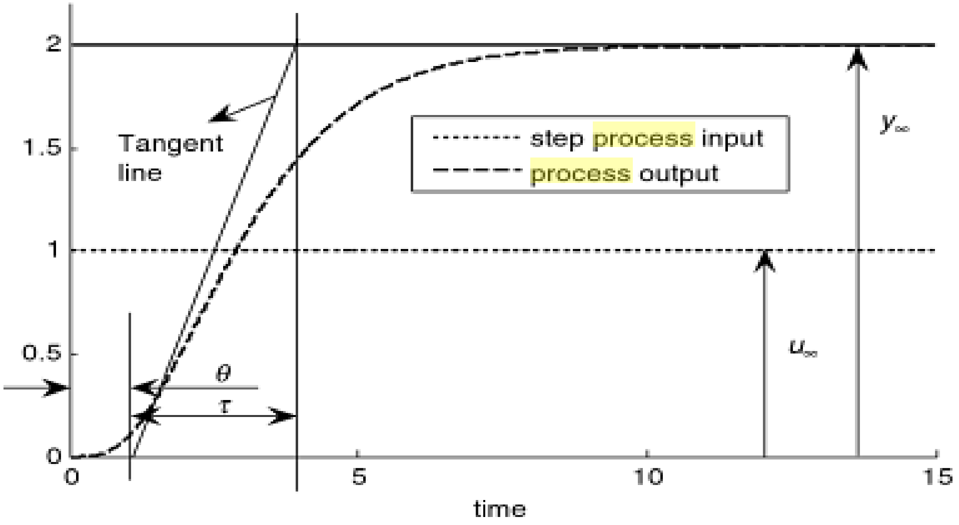

Figure 2. Process Reaction Curve: Illustrates how FOPTD model parameters are determined adopted from (Google text book) [4]

The point where the response stops curving upwards is referred to as the inflexion point. From this inflexion point a tangent line is drawn to a point where it crosses the step line, thereafter the system time constant can be determined. Time delay starts from zero to a point where the system response curves upwards.

Proposed empirical tuning rules in this report are Cohen Coon (C – C), Internal Model Control (IMC) and Integral Time and Weighted – Absolute Error (ITAE). These proposed empirical rules have been realized using the FOPTD model and the rules provide (mostly) desired performance of the industrial systems. These empirical rules are already available and presented in various control books and literature. The empirical tuning rules aims at determining the parameter gains and time constant of the PID controller. These parameters are the ultimate gain, K_c, the integral time constant, τ_i , and the derivative time constant, τ_d. The empirical rules for each respective technique are presented in Table 1 – 3.

4.1 Cohen – Coon Method

Ziegler Nichols tuning rules have been modified to form Cohen Coon rules. The main aim of the modification by these researchers was to enhance system performance of which the Ziegler Nichols rules could not meet. The response given by C – C technique is rather fast with slight overshoot, small settling time, and zero error unlike the Ziegler Nichols sluggish response. The C – C tuning rules also ensures an acceptable overshoot decay ratio until the system stabilizes [2]. Table 1 present C – C tuning rules.

Table 1. Cohen and Coon tuning Rules [5].

Gain Terms	K_c	T_I	T_D
PID	$\frac{\tau}{K\tau_d}(\frac{4}{3}+\frac{\tau_d}{4\tau})$	$\tau_d\frac{32+6\tau_d/\tau}{13+8\tau_d/\tau}$	$\tau_d\frac{4}{11+2\tau_d/\tau}$

4.2 Integral of time – weighted Absolute Error (ITAE)

ITAE is an optimisation technique as well as the Integral of Absolute Error (IAE), Integral of Squared Error (ISE) and Integral of Time Multiply Squared Error (ITSE). The purpose of these integrals is to minimize the output error. The ITAE handles continuous errors occurring for longer periods of time by gradually summing up the errors until a zero steady – state error is achieved [6]. The mathematical expression of the ITAE performance index is given as

$$ITAE = \int_0^\infty t|e(t)|dt \qquad (6)$$

t, is the time and, $e(t)$, represent the time varying error given as the difference between reference signal and output. The ITAE essential optimises the variation in the output signal (disturbance loads) and changes in the reference signal (set point loads). Table 2 present the empirical rules for the ITAE set point and disturbance load.

Table 2. ITAE Empirical Tuning Rules for Error Minimisation [7, 8]

Gain Terms	K_c	T_I	T_D
Load Disturbance (Load)			
PID	$\frac{1.357}{K}[\frac{\tau_{dt}}{T}]^{-0.947}$	$\frac{T}{0.842}[\frac{\tau_{dt}}{T}]^{0.738}$	$0.381T[\frac{\tau_{dt}}{T}]^{0.995}$
Set point (SP)			
PID	$\frac{0.965}{K}[\frac{\tau_o}{\tau}]^{-0.855}$	$\frac{\tau}{0.796 - 0.147[\frac{\tau_o}{\tau}]}$	$0.308\tau[\frac{\tau_o}{\tau}]^{0.929}$

4.3 Internal Model Control (IMC)

Modified IMC tuning rules have been well derived by different researchers and explicitly publicised [9]. The approximations of these tuning rules are realized by using the FOPTD model. Table 3 present the modified IMC rules.

Table 3. IMC – PID Tuning Rules [7]

Gain Terms	K_c	T_I	T_D
PID	$\frac{2T + \tau_d}{2k(\beta + \tau_d)}$	$T + \frac{\tau_d}{2}$	$\frac{T\tau_d}{2T + \tau_d}$

5. TUNING MPC CONTROLLER

F Model Predictive Control is an umbrella name encapsulating a family of predictive control algorithms developed in the early seventies by a French man called Richalet. Richalet's algorithm incorporated the MAC (Model Algorithmic Control). Predictive control has received a wide acceptance in different industries, as a results researchers such as Cutler, Ydstie, Clarke and so on developed sophisticated yet simple predictive control algorithms such as Generalized Predictive Control (GPC), Predictive Functional Control (PFC), Dynamic Model Control (DMC), IMC etc [10]. These predictive control algorithms are used predominantly in different industrial applications such as in automotive industries, control and automatic systems, chemical industries as well as health equipments. Industrial applications operate automatically and at times they do run for hours without any human interventions, therefore the excellent performance provided by such systems gives an insight as to how good and robust is the MPC. MPC control like any other controller it does have pros and cons on its overall performance. Though it does have few drawbacks, the MPC strategy has proven to be worthwhile in handling constraints, noise and other various perturbations hence providing efficient, robust and reliable control [10, 11].

5.1 MPC strategy

Unlike any other control scheme published in different literature such as PID which depends mostly on current and past inputs to control the system, the MPC strategy involves using current, past, future inputs as well as the optimizer. The optimizer in this case holds the weight which penalizes the input/output signals (cost function) for improving stability of the system as well as constraints handling mechanism. With these features the MPC control strategy is capable of predicting and updates its command signals for best control. Based on various literatures researchers including *(J.A. Rossiter 2003)* have well publicised that MPC control strategy is analogous to human thinking and different real life examples such as crossing the road and driving up/down the hill have been provided to show the similarities [10, 12]. This is true in the sense that human brains constantly use new information and past inputs/outputs to update our decision making as well as to make inferences whenever we are tasked or they is an objective to accomplish. For the MPC strategy to be plausible, the following components are important

- Control law/horizon N_u
- Output predictions horizon $N_2\ or\ n_y$
- Input and output weights β
- Receding horizon – every sampling time the output is updated.

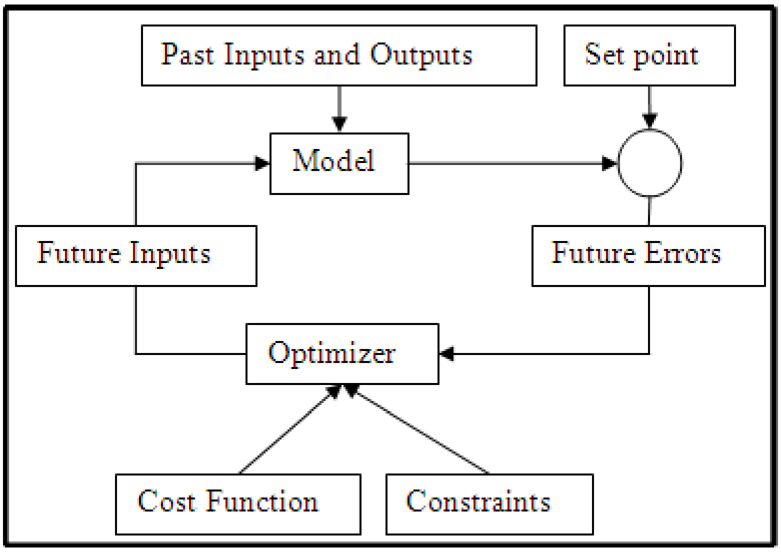

Figure 3. MPC Control Strategy [10]

5.1.1. Receding horizon strategy

A For instance, consider crossing the road. Whether the road is clear or not the pedestrian intending to cross the road is obliged to make some observations before crossing. Based on observations made a decision (control action) is made as to cross the road or wait for the oncoming traffic to pass then cross the road. As the pedestrian moves further away from the starting point a fraction of the road diminishes from their minds while the other part of the road continues to move with them on their mind (horizon), therefore the decisions (control actions) are made relative to that part of the road visible from where the pedestrian is on the road – that's receding. The part of the road visible (horizon) and traffic speeds dictates time one can take to cross the road (optimal response). Say, for instance a bike is travelling at 60 mph and they is a distance of 30 meters in between – chance of the pedestrian being hit is high if the pedestrian decide to cross the road anyway. To avoid the collision the prediction horizon for crossing the road has to be large to be able to account for all scenarios.

Researchers have published a number of control laws such as the DMC, LRPC, PFC but commonly published control law of all times and its known to perform well on various industrial applications is the GPC control law. Its implementation is simpler compared to its counterparts. The PFC on the other hand yields similar system response characteristics as the GPC [12] and several researchers have reached the same conclusion given by (*J.A. Rossiter 2003)*. Therefore the reader has to note that for the remainder of this chapter we focus solely on the GPC control law. The aim is to minimize the cost function and compute the GPC law.

5.1.2. Constraint handling

Apart from other parameters used for the MPC (GPC) controller, constraint handling is an important and advantageous aspect which makes the MPC (GPC) controller stands out from other control schemes by improving performance. The GPC is capable of optimising in the presence of constraints for a given predictive horizon, n_y, and appropriate control horizon, N_u. More importantly for better performance and constraint handling the prediction horizon has to be large as well as the control horizon to give the controller flexibility in dealing with constraint as well as the process dynamics. Constraint handling will not be explored in this report but the focus will be directed towards optimisation of the offset while the process inputs and outputs are weighted. Therefore the input and output responses of the system will not be bounded.

Optimisation is a procedure of reducing or minimizing the steady state error, therefore after optimization the error has to be a steady zero [12]

The performance index of the GPC algorithm is optimised using the following cost function [12]:

$$J_{gpc} = \sum_{j}^{n_y} \left\| R_{k+j} - Y_{k+j} \right\|_2^2 + \beta \sum_{j=0}^{N_u-1} \left\| U_{k+j} \right\|_2^2 \qquad (7)$$

R_{k+j} *and* U_{k+j} – *reference or input signal*

Y_{k+j} –*process output*

The best minimisation of the offset will happen if the variables (above) to be optimised are equal to each other. However in real process this is not the case. There are always mismatches of the input or output signal therefore input and output weights are included on the GPC algorithm to further illuminate/reduce the error, however the user/operator desires

5.2 GPC algorithm

A A standard GPC algorithm / law utilises the CARIMA representation [11]:

$$G(z^{-1}) = \frac{B(z^{-1})}{A(z^{-1})} u(t-1) + \frac{C(z^{-1})}{A(z^{-1})} \varepsilon(t) \qquad (8)$$

$A(z^{-1})$ and $B(z^{-1})$ are polynomials in the backward shift operator z^{-1}.

$\varepsilon(t)$ – Gaussian disturbance

Basically this CARIMA model formula presents the transfer function of a system or process in the backward shift (z^{-1}). This model is important for computing the GPC control law.

Researchers have derived the GPC law in different literatures including the most recent literature by *(J.A. Rossiter 2003)*. His vantage point was based on ‘Model – Based Predictive Control offering the practical approach’ paradigms. For brevity the GPC law established in [12] will be used.

$$\Delta u_j = P_r \vec{r} - N_j \overleftarrow{y} - \breve{D}_j \Delta \overleftarrow{u} \qquad (9)$$

An arrow, ← , represent past inputs while, →, represent future inputs

This control law is adopted directly from [12]. The above GPC control law is obtained by minimising the cost function with an assumption that there are no constraints on the system. This assumption is important for the GPC control law in the sense that one has to obtain the best performance without constraints otherwise the law won't work for constrained systems. When minimising the cost function an ideal process is assumed to allow the first derivative of the quadratic cost function to be zero see below:

$$\frac{dJ_{gpc}}{d\Delta\vec{u}} = 0 \equiv \Delta\vec{u} = (G^TG + \beta I)^{-1}G^T[\vec{r} - P\overleftarrow{y} - Q\Delta\overleftarrow{u}] \qquad (10)$$

This formula is realised after several substitutions of the estimated inputs and outputs on the cost function (7) before the first derivative can be taken.

G ,*is the dynamic matrix of the model*
β ,*weighting factor*
P *and* Q ,*polynomials*
I ,*identity matrix*

We know that for the GPC law to be implemented, the first row of the matrix has to be extracted to give the GPC control law in the form of equation (9), hence

$\Delta u_j = e_1^T\Delta\vec{u}$
$e_1^T -$ *1st row of the matrix transposed*

Note [12]

$$P_r = e_1^T(G^TG + \beta I)^{-1}H^T \qquad (11)$$

$$N_j = e_1^T(G^TG + \beta I)^{-1}H^TP \qquad (12)$$

$$\breve{D}_j = e_1^T(G^TG + \beta I)^{-1}H^TQ \qquad (13)$$

Based on these equations (11, 12, 13) plus tuning factor (explained next) a Matlab was written to be implemented together with a Simulink model.

5.2.1. Auto tuning rules for the GPC

R The main purpose of this subchapter is to clarify the effects of each tuning factor using the nonlinear system proposed. Intuitively is well known that nonlinear system require high gains of energy to be kept under control and they are also conditionally stable. Conditionally stable here means the gains of the nonlinear system are bounded [12]. Basically these GPC tuning factors are: Control horizon N_u, Minimum Output Horizon, N_1 *or sizey*, Maximum Output Horizon, N_2 *or* n_y and Weighting factor β

- Minimum output Horizon N_1

The nonlinear system used in this report is single input and single output therefore N_1 becomes one for all of the simulations demonstrations. Furthermore, N_1 intervenes in the cost function affecting only the output prediction. Remember N_1, can be any number from one (minimum output) up to, $N_2 - 1$ depending on the system outputs.

- Effects of varying $N_2, N_u \; and \; \beta$
- N_2, N_u Tuning Factors

For improved and desired performance, N_2, has to be large always therefore increasing N_u, to a bigger value will give the control input the flexibility it requires to deal with system dynamics. When the control horizon is greater than one ($N_u > 1$), the solution is likely to be optimal (best). Furthermore, the bigger the control horizon means more energy is being spent on the system, causing energy loses in case of overshoots – running costs etc.

- Impact of N_2 increasing depends on N_u. If N_u is large therefore larger N_2 is always better.
- If $N_u = 1$ and N_2 large implies open loop behaviour which means the closed loop poles move closer to the open – loop poles.

Choice of N_u with unstable Open Loop Poles
In order to have a sensible control optimisation, predictions have to be convergent. For example, consider an arbitrary system given below:

$$\varphi = \frac{buz^{-1}}{(1-2z^{-1})(1-3z^{-2})}$$

Then

$$buz^{-1} = (1-2z^{-1})(1-3z^{-2})\varphi$$

Therefore, if buz^{-1} does not cancel these poles on the right hand side then predictions will go to infinity. So the only way to force the numerator to cancel those poles we need at least two degrees of freedom i.e. as the number of poles increase so does the degree of freedom. Ideally N_u has to be greater than p. Most importantly increasing N_u gives the GPC controller more flexibility to change the control input. Furthermore, N_u has to be far less than, N_2 , otherwise the bits beyond N_2 would still have significant dynamics, therefore these dynamics are hard to ignore. The amount of information predicted is determined by N_2 .

- $N_u \geq p$, $\quad p \text{ is the number of unstable poles}$
- $N_u \ll N_2$
- $N_u = 1$, $\quad \equiv \text{Minimum variance control}$

Furthermore, If open loop performance or response is poor, this would only mean that $N_u = 1$, for the GPC controller will also be poor.

- Choice of N_u in general

The previous section has given some insight on how important the choice of N_u is on controlling the system. In a similar vein even when the system does not have unstable open loop poles, it may have some underlying dynamics which are undesirable and the only way to eliminate or modify these dynamics is by providing an input signal which is non – simple (not a step). Therefore for system which have non – simple dynamics a larger N_u is required in order to have flexibility in the input trajectory to modify the dynamics. As a caution, the reader has to note that increasing N_u means increasing the energy to control the system.

- The Control Weighting Sequence $\beta,$

These weighting of the input – output sequence is given in a form of range. For instance the weighting factor may be chosen from the following range, $10^{-3} \leq \beta \leq 10^{3}$. The weighting factor "penalises the control signal and improves the condition of the matrix to be inverted as well as bring the system response closer to desired closed loop performance [11, 12]". Since the nonlinear system used in this report is SISO system the weighting factors of the input and output are equivalent.

6. TUNING MPC CONTROLLER

This section proposes the Auto Tuning algorithm for the PID and MPC. We simulated and analysed different tuning and the best algorithm chosen for the PID as well as the effective GPC control law. We auto tuned the PID controller using the Model – Based PID Auto tuning detailed in [13], focussing on First Order Plus Dead Time (FOPDT), first-order lag and integrator plus dead time (FOIPDT), and so on. We incorporated the ITAE (load) control on model based Auto Tuning PID. The ITAE (load) is an optimisation technique which aims on minimising the system output error using relations in Table 2. These relations in Table 2 provide the closed loop gain of the system, integral time and the derivative times responsible for auto tuning the PID controller.

Table 4. PID Controller tuning and Auto tuned MPC Parameters with their respective parameter values

Controller	Parameter	Values
PID	K_c	0.3745
	T_i	0.3243
	T_d	0.0954
MPC	n_y	20
	N_u	3
	$W_u = W_y$	0.25
	$sisey$	1

These values are used in simulink models to auto tune the PID and MPC respectively. The simulation results are shown in Figure 4 and 5 and the comparison of the simulation results are in Table 5.

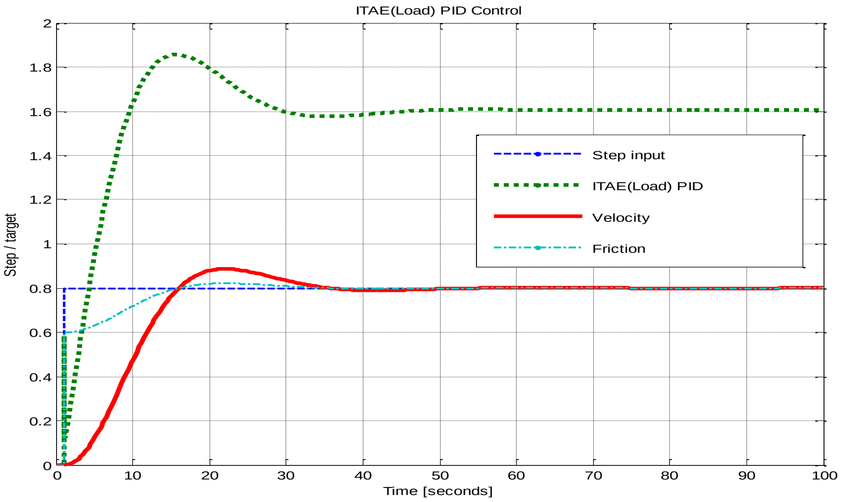

Figure 4. Auto tuned PID: Nonlinear System Velocity Control

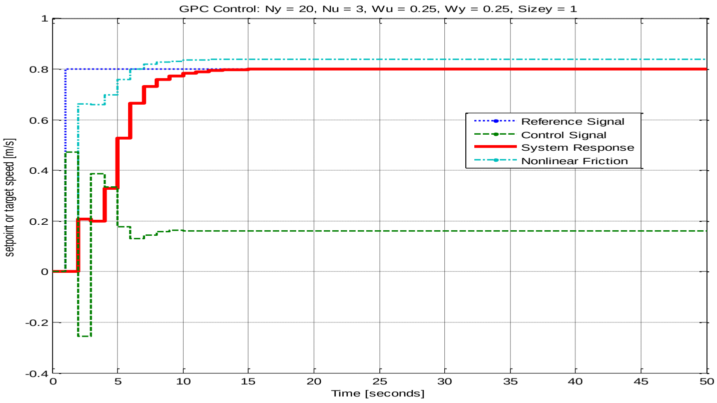

Figure 5. Auto tuned MPC Velocity Control

Table 5. IMC – PID Tuning Rules [7]

Auto tuned PID Controller	Auto tuned MPC Controller
Zero steady – state error ITAE (Load) tuning technique provides good, and desirable performance see Figure 4 Good for industrial applications with only little information to rely on to yield the best performance. Good stability	Good system stability. Zero steady – state error Best for industrial applications Relies on more information to make prediction Fast, reliable and stable response see figure 5

6.1 Discussions

It can be observed that both the auto tuned PID and auto tuned MPC provide good stability. The auto tuned MPC on the other hand has provided better control of the settling time, rise time, fast with zero overshoot which means a smooth transient response of the nonlinear system was attained. Although constraint handling is not investigated in this paper, for cases of constraint in the system the GPC has a better chance of controlling and giving optimal performance while for the PID only little can be said. The PID algorithm is not set for constraint handling. Auto tuning the MPC is the best control option for industrial application because it is well – known that most mechanical devices are subjected to constraints as well as different attributes that influences the behaviour of the system.

7. CONCLUSIONS AND FUTURE WORK

This paper has extensively looked at two different control strategies, capable of auto tuning in the presence of time varying parameters. The interactive nonlinear system was used to test the efficacy of these control strategies. These are PID auto tuned using the ITAE (with load disturbance) model based approach and MPC auto tuned using the GPC control law. For the PID control strategy, all three gains were utilized whilst P, PI and PD were neglected for brevity. Using all the PID control terms ensures better control for the system studied in this paper.

The tuning technique implemented for Auto Tune PID controller – ITAE (Load) tuning technique has been extensively discussed and mathematical formulas provided. The transient response for is good for the nonlinear system. Though the best method can be employed to auto tune the PID one problem may still remain that is constraint handling. Most of the PID controller algorithms are not set up to handle constraints. Therefore PID controllers are good or acceptable controllers for some of the industrial applications provided the system under control is constraint free.

The auto tuning algorithm for the MPC has proven to be more robust and reliable for industrial application i.e. nonlinear system than the PID control loops. Simulation results indicate quick yet stable response of the nonlinear system with zero steady – state error though subjected to nonlinear friction. It’s also good in constraint handling, although not explored in this report a useful avenue to look out further is Constraint Handling because the PID is not really set up for this but the predictive control is. Furthermore, based on the simulation results auto tuning MPC is the best alternative for industrial application because it relies on more data to make its optimal predictions as a result it a flexible controller when given a larger horizon to predict outcomes.
The comparison between auto tuned PID and auto tuned MPC was based on the simulation results. We auto tuned PID and auto tuned MPC controllers for a nonlinear system using the ITAE (load) to auto tune the PID controller and the GPC algorithm for the MPC. We observed that the GPC offers effective transient response i.e. quick rise time, fast settling time, fast with zero overshoot control compared to the ITAE (load) for the PID. Auto tuned MPC is an effective control option for industrial application.

In summary, this paper has given a preliminary study of different control strategies but we have only really looked at narrow range of PID control, one nonlinear example, and one particular variant of predictive control. Therefore, it will be logical to investigate more control strategies such as adaptive control, neural networks, fuzzy logic, nonlinear MPC, PID etc and try different examples of mechanical devices because they have different attributes and different things that are important. This paper however largely focused on single input and single output loop therefore a more interesting challenge will be to look at multi – variant loop for the PID and MPC.

ACKNOWLEDGEMENTS

The authors would like to thank their families for the support!

REFERENCES

[1] Concepcion A. Monje, B.M.V., Vicente Feliu, YangQuan Chen, (2008) *Tuning and auto-tuning of fractional order controllers for industry applications,*Vol. 16, No 7.

[2] Paz, R.A., (2001) *The Design of the PID Controller.* No 12.

[3] Prof. T.K. Ghoshal, P.S.S. (2010) *Process Instrumentation and Control*.

[4] Ltd, J.W.a.S.A.P., (2009) *Process Identification and PID Control*, Vol. 2, No (02 -01).

[5] Saeed Tavakoli, M.T., (2003) *Optimal Tuning of PID controllers for First Order Plus Time Delay models using Dimensional Analysis*, Montreal, Canada: The Fourth International Conference on Control and Automation (ICCA'03).

[6] Yamada, K., (1999) *Modified Internal Model Control for unstable systems.* Proceedings of the 7th Mediterranean Conference on Control and Automation, Vol. 99, No 7.

[7] Zalm, G.M.v.d., (2004) *Tuning of PID – Type Controllers: Literature Overview*, DAF, TU.

[8] Zollars, R.L., (2011) *Tuning Rules: Quarter Decay Ratio (Cohen and Coon)*, Voiland School of Chemical Engineering and Bioengineering, ChE 441 Process Control: Washington State University.

[9] Wen Tan, J.L., Tongwen Chen, Horacio J. Marquez, (2006) *Comparison of some well-known PID tuning formulas*, Publisher Computers and Chemical Engineering 30.

[10] Bordons, E.F.C.a.C., (2003) *Model Predictive Control*, Springer, Advanced Textbooks in Control and Signal Processing.

[11] Mahfouf, P.M., (2011) *Model – Based Predictive Control (MBPC)*, Sheffield: Advanced Control Systems, ACS 406 Notes

[12] J.A. Rossiter, (2003) *Model – Based Predictive Control*, Sheffield, CRC Press control Series.

[13] Leva, A. and M. Maggio, (2012) *Model-Based PI(D) Autotuning*, in *PID in the Third Millennium*, Springer-Verlag London.

Authors

Kenneth O.M. Mapoka is with the Department of Agricultural Engineering and Land Planning, Botswana College of Agriculture, Gaborone, BOTSWANA. He received Master of Engineering in Mechatronics from Sheffield University, United Kingdom in 2012. His special fields of interest include predictive and robust control applications.

Dr Tranos Zuva is currently with Department of Computer Systems Engineering, Tshwane University, Pretoria, South Africa.

Howard Masebu is a lecturer at Botswana College of Agriculture, Gaborone, Botswana in the Department of Agricultural Engineering and Land Planning.

Keneilwe Zuva is currently a lecturer at the University of Botswana in the Department of Computer Science. She received Masters of Engineering in Information Systems and Networking from the University of Essex, UK in 2001. Her research interests are networking, image processing, and network security.