

Pareto Analysis of Controller Design Methodologies for Integrator plus Dead Time Processes

Abstract—Controller design methodologies are commonly compared by publishing examples of individual controllers that were designed using competing approaches. In this paper, Pareto analysis is used to compare three methods of designing controllers for integrating second order plus dead time processes - a problem that has been discussed in the literature. The analysis shows that, in the example chosen, superior performance can be attained by adjusting the approach to tuning controllers. Solutions found by Pareto analysis that improve performance may be directly implemented. This finding demonstrates the usefulness of Pareto analysis in Control engineering, as either a design or research technique.

Index Terms—Disturbance rejection, Integrating processes, Pareto-frontier Differential Evolution, Pareto front, Smith Predictor

I. INTRODUCTION

The Smith Predictor has recently been extended to processes that include an integral or unstable pole. A standard Smith Predictor could not be applied because it cannot reject an input disturbance for these processes.

Mataušek and Micić [1,2] showed that adding an additional PD controller, called the disturbance modifier in this paper, allows the standard Smith Predictor to reject the input disturbance. This controller functions by modifying the disturbance rejection loop only, leaving the set point tracking behavior unchanged in the nominal case where the model is an exact match of the process. Majhi and Atherton [3] extended this controller to integrating and unstable processes by including a second PD controller to prestabilize the *model*, and feeding the output of this controller to the *process* as well. This effectively prestabilizes the *process* using the modeled response of the process without dead-time. These adaptations allow a wide range of processes to be controlled using a Smith predictor structure.

Kaya [4] provides an alternative means of specifying set point tracking behavior in terms of a settling time, T_s and damping factor, ζ . He also provides another set of tuning laws for the disturbance modifier when the process is an integral second order plus dead time (ISOPDT) process that he claims to be superior to the tuning laws provided in [3].

The author has shown [7] that the disturbance rejection behavior of these controllers is affected by set point tracking behavior, even for the nominal case.

Aström et al. [5] had published an earlier work dealing only with integrator plus dead time processes. This work fully decoupled the set point response of the system from the disturbance rejection behavior by removing feedback from the output of the process to the set point tracking controller. He

adds a single transfer function which consists of both dead time elements and polynomial elements that functions as a disturbance estimator.

When Liu et al. [6] investigated this field, they stated that [3] suffered from the coupling described above. In order to avoid this problem they also chose to remove the feedback from the output of the process to the set point tracking controller, resulting in a structure that may be viewed as an extension of [5] to ISOPDT processes and also to processes that include an unstable pole. The disturbance estimators proposed in [6] are not directly realizable, instead the authors proposed using a Maclaurin expansion (a Taylor expansion about the point $s = 0$) to approximate the function as a PID controller.

II. CONTEXT AND AIMS OF THIS WORK

References [3], [4], and [6] offer different solutions for the same set of processes, and clearly it is necessary for a designer to understand the benefits and disadvantages of each of these three options. In order to prove the superiority of the new controller over previous controllers, earlier papers showed simulations of various controllers for problems tackled by other authors. These simulations focused on set point tracking and input disturbance rejection, neglecting considerations of output disturbance and control effort. This process has lead to [3], [4], and [6] which are competitive papers, that all offer set point tracking with no overshoot in the nominal case and a smooth damped response for input disturbance rejection.

The fact that these controllers achieve reliable, acceptable set point tracking profiles allows the author to extend the comparison of these methods to consider output disturbance rejection and control effort. In this paper a single process is selected and the complete sets of potential controllers produced by [3], [4], and [6] are compared directly. These sets are then compared to controllers that have been optimized using Pareto-frontier Differential Evolution (PDE) [8]. By comparing the full set of potential controllers a more accurate understanding of the strengths of each method can be obtained.

III. PROBLEM

In this work the Pareto front for the solutions offered by [3], [4], and [6] to example 3 in [3] is studied. The model used is:

$$G_m e^{-\theta_m s} = \frac{ke^{-\theta_m s}}{s(T_1 s + 1)} = \frac{e^{-6.5672s}}{s(3.4945s + 1)} \quad (1)$$

This model is an approximation of a high-order process with dead time but it will be adopted as the actual process in this paper, omitting the modeling step. Robustness will be

controlled for as described in section VI part C.

IV. METHODS INVESTIGATED

The Majhi and the Kaya controllers can be represented in the same block diagram, because the methods vary only in the tuning of the controllers G_c , G_{c1} and G_{c2} , by the Majhi naming convention. Fig. 1 below gives a block diagram showing both the Majhi and Kaya naming conventions.

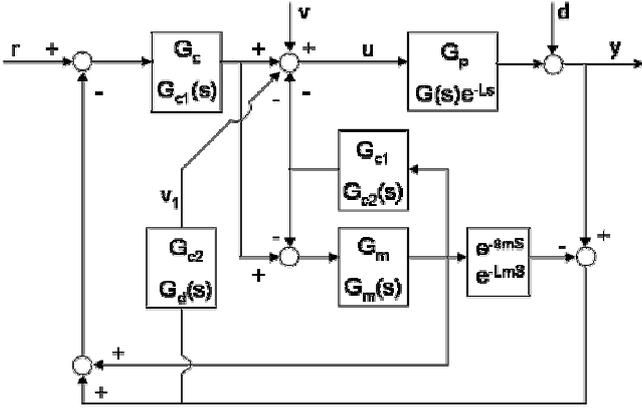


Fig. 1. Block diagram of Majhi and Kaya controllers showing naming convention from [3] top line and [4] bottom line.

A. The Majhi controller

Majhi and Atherton [3] specify their controllers as: $G_c = K_p(T_i s + 1)/(T_i s)$, $G_{c1} = K_f(T_f s + 1)$, $G_{c2} = K_d(T_d s + 1)/(T_d N s + 1)$ for ISOPDT systems (where $T_d/N \ll 1$ and can therefore be neglected in the analysis). The time constants are set to be equal to the process time constant, that is, $T_i = T_f = T_d = T_1$. The gains K_p and K_f are chosen to give the desired setpoint response, and the delay free portion of this desired setpoint response is given as $Y_r = 1/(\lambda s + 1)^2$. This gives $kK_p = T_1/\lambda^2$ and $kK_f = 2/\lambda$ with λ as a tuning parameter that Majhi and Atherton suggest is chosen as $\lambda \approx T_1$. The gain K_d is set by the equation:

$$K_d = \frac{\pi - 2\phi_m}{2k\theta_m} = \frac{0.5236}{k\theta_m} \quad (2)$$

where Φ_m is a tuning parameter, that they suggest to be $\Phi_m = 60^\circ$.

B. The Kaya controller

The controllers used by Kaya [4] are identical in form to the controllers in [3]. However Kaya defines a new variable, T_s , the closed loop settling time for the under-damped system. He chose the relationship $\omega_o = 2.5/\zeta T_s$. After choosing a value for ζ and T_s , the natural frequency of oscillation ω_o is fully specified. Then $K_p = T_1 \omega_o^2/k$, $K_f = 2 \zeta \omega_o/k$ and $T_f = T_i = T_1$ as before. This fully specifies the set-point tracking of the Kaya controller as $Y_r = \omega_o^2/(s^2 + 2 \zeta \omega_o s + \omega_o^2)$. Comparing the set-point tracking behavior of [3] to that of [4] it can be shown that if $\zeta = 1$ then $2\lambda = 2 \zeta / \omega_o$, and $T_s = 2.5\lambda$. A damping factor

of $\zeta = 1$ is often desirable for a system's response, and this is either implicitly or explicitly assumed in all examples in [4].

For the disturbance rejection controller, Kaya [4] adopts the method described in [2] and updates it for ISOPDT systems by defining an equivalent time constant $L_m = \theta_m + T_1$ (using notation from Eq.1). Then $T_d = \alpha L_m$ and

$$K_d = \frac{\frac{\pi}{2} - \phi_m}{K_p L_m \sqrt{(1 - \alpha)^2 + (\frac{\pi}{2} - \phi_m)^2 \alpha^2}} \quad (3)$$

Kaya follows the suggestion in [2] of setting the tuning parameters to $\Phi_m = 64^\circ$ and $\alpha = 0.4$ in the equation above.

C. Liu controller

Liu et al. [6] recently proposed a new controller that requires a different block diagram representation, as shown in Fig.2.

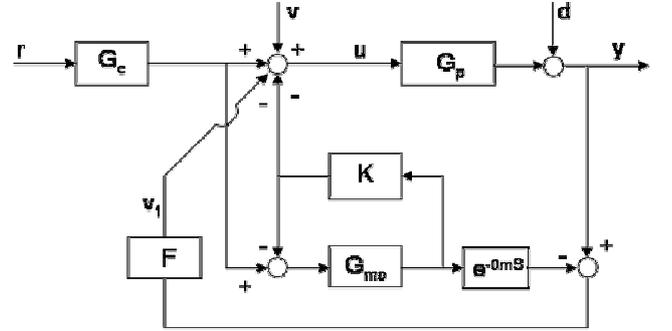


Fig. 2. Block diagram of Liu controller redrawn from [6].

For a ISOPDT process, the setpoint tracking controllers are specified as: $K = k_c = 1$, and

$$G_c(s) = \frac{T_1 s^2 + s + k_c k}{k(\lambda_c s + 1)^2} \quad (4)$$

where λ_c is a tuning parameter equivalent to λ in [3]. The disturbance estimator is defined as:

$$F_m(s) = \frac{s(T_1 s + 1)(a_1 s + 1)}{k[(\lambda_f s + 1)^3 - (a_1 s + 1)e^{-\theta_f s}]} \quad (5)$$

where $a_1 = 3\lambda_f + \theta$, λ_f a tuning parameter to tradeoff performance with robustness of the disturbance estimator.

This disturbance estimator is not realizable in this form, and a PID approximation to this desired estimator must be used: $F(s) = k_f + 1/(T_i s) + T_D s$. Liu [6] suggests that a low pass filter defined as $1/(\alpha_f T_D s + 1)$, with $0.01 < \alpha_f < 0.1$, should be used to make $F(s)$ realizable.

V. SIMULATION SETUP

A. Input parameters

Controller performance was simulated using software based on the 4th order Runge Kutta (RK4) algorithm. Setpoint tracking was initially ignored, since this problem has an adequate solution in all the methods. Thus the signal r is set to

zero in Fig. 1 and 2 above. From Fig. 2 it can be seen that if $r = 0$ the output from G_c , K and the model will always remain zero, even if the process varies or disturbances enter the loop. Therefore only the controller F and the process need to be simulated. For the Liu controller the input parameters, x , were specified as $x = [k_f T_1 T_D \alpha_f]$. The Majhi and Kaya controllers were simulated using code that implemented the control loop in Fig. 1. The code written to perform these simulations was tested by reproducing diagrams published in [3] and [4]. For these controllers $x = [K_d T_d (1/N) T_s]$.

B. Output (cost function)

All simulations for the data sets were 120s long, and a time step of $dt = 6.35\text{ms}$ was used for the RK4 algorithm. Three input signals were simulated for each controller: an input disturbance (v) step of 0.1 units (as used in [3] and [6]), an output disturbance (d) step of 0.2 units and a noise signal (n) with standard deviation of 0.1.

For each simulation two values were monitored, the error in the output value ($e = -y$ since $r = 0$) and the input value, u , to the process *after* the input disturbance has entered the loop. Measuring the u value after the simulated input disturbance, v , has entered, ensures that the final value of u is zero even for an input disturbance simulation. This is important because integral time squared error (ITSE) measures were applied to the first four costs. If the same steady state offset (equal to the input disturbance) remained in the measured u value, this constant value would have a larger contribution to the integral than the dynamic response.

For cost functions associated with the noise input the standard deviation of the monitored signals was found using recursive equations that are given in [9]. The output, f , from a specific x was defined to have the form $f = [U_v E_v U_d E_d U_n E_n]$, where U_v refers to the cost evaluated on signal u when input v is simulated.

VI. DESCRIPTION OF DATA SETS

A list of the datasets prepared for this paper is given in Table I. Each dataset was constructed using the nominal model as the process, and a second set of variation datasets were made using a positive 10% variation in dead time.

TABLE I
NAMING OF DATA SETS AND NO. OF ELEMENTS

Dataset	1	2	3	4	5	6
Name	TsPDE	mkPDE	LiuPDE	MajhiBase	KayaBase	LiuBase
Nominal	2000	295	1262	1457	1461	2267
Variation	2000	228	1127	1457	1461	2267

A. Optimized Data Sets

Datasets 1-3 were created using a PDE optimizer written in Matlab by Prof Greene from pseudo code in [8] and adapted to interface with the simulation software by the author.

Dataset 1 was created by allowing the optimizer to tune the

inputs K_d , T_d and T_s to obtain Pareto nondominated solutions for the Majhi / Kaya simulation code. The first four costs were considered during optimization. The filter term $(1/N)$ was kept fixed at 0.01 during the optimization. 1000 nondominated solutions were found; each of these solutions was re-evaluated with $(1/N) = 0.1$ to form the full set of 2000 elements.

For datasets 2 and 3 the optimizer ran until 5000 nondominated solutions were found for the nominal problem in each set. It was noted that a number of these nondominated solutions rejected noise well but performed poorly on the other costs. These datasets were tested for dominance again using only the first four costs, and the nondominated solutions were retained.

For dataset 2 the optimizer was allowed to tune K_d , T_d , and $1/N$ in the Majhi / Kaya simulation. The setpoint controllers were restricted to those published in [3] for this problem, that is $\lambda = 3.4945$, which has an equivalent settling time of $T_s = 8.74$. This gives $K_p=0.2862$, $T_i=3.4945$, $K_f=0.5723$ and $T_f=3.4945$ (from [3]).

Dataset 3 is based on [6] and allows all the parameters, k_f , T_1 , T_D , and α_f , to be tuned. Since [6] decouples the setpoint tracking from the disturbance rejection, the costs found for this dataset can be obtained for any desired settling time.

Thus datasets 1-3 consist of points that describe the best performance available when different restrictions are placed on the controllers. Dataset 3 requires a PID disturbance estimator, dataset 2 demands a settling time of 8.74s (by Kaya's definition) and dataset 1 only requires that the shape of the settling profile is the same as [3].

B. Base Data Sets

The base datasets (4-6) are intended to cover every controller that could be designed by a given method. This is done by stepping the tuning parameters available in each method from the smallest expected value to the largest expected value in a regular way. For each value of the tuning parameter a range of nondominant filters are tested in order to control for the effect of filter terms.

For datasets 4 and 5 the only parameter which is varied is T_s since [3] and [4] suggest that the tuning proposed for K_d and T_d is optimal and that the parameters available for tuning the disturbance modifier should be left unaltered. Therefore, the disturbance modifier was fixed at $K_d=0.0797$ and $T_d=3.4945$ for the Majhi method (dataset 4) and at $K_d=0.0719$ and $T_d=4.0246$ for the Kaya method (dataset 5). The settling time T_s was varied from 0.1 to 11.98 in steps of 0.03. Each settling time was tested for filters of 0.01, 0.03, 0.06 and 0.1. At values of $T_s < 2$ the larger filter terms resulted in numerical overflow and were automatically removed during simulation, dataset 4 and 5 consist of only those points which did not cause overflow. This suggests that for small values of T_s more derivative action is required to reject a disturbance. The effect of varying the tuning for K_d and T_d could be tested in a similar manner to verify the claims in [3] and [4].

Dataset 6 was created by varying λ_f from 0.2 to 18.2 in steps of 0.03. This covers the range that is suggested to be good by [4] (that is $0.50 < \lambda_f < 3.00$). Filters of 0.01, 0.03, 0.06, 0.1, 0.17, and 0.22 were tested. This range of filters covers the range suggested in [4] of 0.01~0.1 and includes two additional values. After creating this dataset, Pareto dominated solutions were removed leaving 2267 solutions for values of λ_f between 3.11 and 17.48.

C. Robustness Test Sets

A good controller must be robust to changes in the process. In [1]-[6] robustness of a controller was demonstrated by showing plots of its performance under a 10% perturbation of the dead time in either a positive sense only or a positive and negative sense. It can be seen from these papers that the peak error due to a disturbance is worst for a positive perturbation of the dead time. The peak error actually improves for a -10% perturbation in the published figures. Therefore to test robustness in this work the datasets' costs were re-evaluated for the case where the process dead time is increased by 10%, to 7.2239s, whilst the model dead time remains at 6.5672s. For datasets 2 and 3 the full 5000 element datasets were re-evaluated and the process of selecting nondominated solutions repeated. This re-evaluation explains the variation in the number of elements in the nominal set and the 10% variation set.

VII. REDUCTION OF DIMENSION

The datasets produced for this work have six dimensional cost sets; each element has the form $f = [U_v E_v U_d E_d U_n E_n]$. Data in high dimensions can be analyzed by statistical methods. However these methods provide limited insight into the problem. Comparisons of the input parameters which produced nondominated results can be made if the input parameters refer to a single method. However, competing methods cannot be compared. Studying costs 1-3 allows the datasets to be visualized directly in three dimensions. This dimensional reduction will be justified by considering the contribution of each omitted cost in turn.

Noise signals are included for generality in the datasets and are available for future work. However, these signals (hence costs 5 and 6) will not be studied in detail in this paper. Designing for disturbance rejection first and then improving noise rejection as a subsequent step is common practice in many design procedures. In this case E_n is small since the process studied has an integral characteristic. The process acts as an effective low pass filter, therefore E_n is likely to be acceptable in any design. The cost U_n is closely associated with U_d (they share a common transfer function), therefore optimizing the trade offs for U_d may yield an acceptable performance for U_n . If U_n is not acceptable in a design, adding suitable low pass filters can improve it, although a degradation in performance in other costs may be encountered. The fourth cost, E_d , shows strong correlation with U_v in all the datasets.

Thus E_d can be neglected if U_v is studied.

VIII. THREE DIMENSIONAL VISUALIZATION OF DATASETS

The datasets prepared for this work cover a general set of disturbance inputs for three separate methods and three optimizations of these methods. The datasets provide a large amount of information, from which a wide range of conclusions can be drawn. Visualizations presented in this section are chosen to best show the effect of restricting T_s on the attainable disturbance rejection.

The images are divided into three sections:

Section A gives two orthographic views of the data points in each dataset. These images are presented first to familiarize the reader with the general shape of the datasets.

Section B shows four sectional views of the datasets. These plots will be used to discuss the effect of varying T_s . The author suggests that the pictures in section A are then reviewed as the sectional views will help the reader visualize these images as three dimensional surfaces.

Section C consists of a single orthographic view of the variation datasets. This picture is included to show that the overall shape of the Pareto front and the relationship between datasets is largely unchanged by an increase in dead time.

A. Orthographic views of nominal datasets

In Fig. 3 datasets 2, 3 and 6 are on the Pareto front for very large values of E_v but turn off the Pareto front as E_v decreases. These datasets in fact turn out of the page at about 45° for larger values of U_d , this can be verified in Fig. 4 and the sectional views.

Datasets 4 and 5 (double light and dark grey line labeled in Fig. 4) are on the Pareto front for small values of E_v , but then turn away from the Pareto front.

Dataset 1 dominates datasets 2, 3 and 6, as Fig. 4 shows.

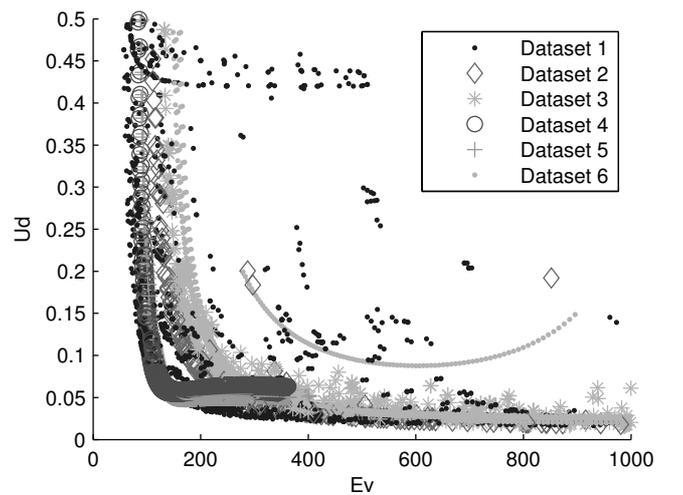


Fig. 3. Orthographic view of datasets 1-6 showing $U_d - E_v$ trade off. U_v axis out of page.

In Fig. 4 datasets 3 appears to be competitive with dataset 2, however Fig. 3 and the sectional views show that set 2 remains slightly dominant.

Dataset 2 was restricted to a value of $T_s = 8.74s$ during the optimization. The fact that it is dominated by dataset 1 shows that the value of T_s selected by Majhi and Atherton had a significant effect on the achievable disturbance rejection. Since dataset 3 consists of optimized PID controllers and hence represents the best performance attainable from a PID controller, it can be seen that PID control is not adequate for this problem.

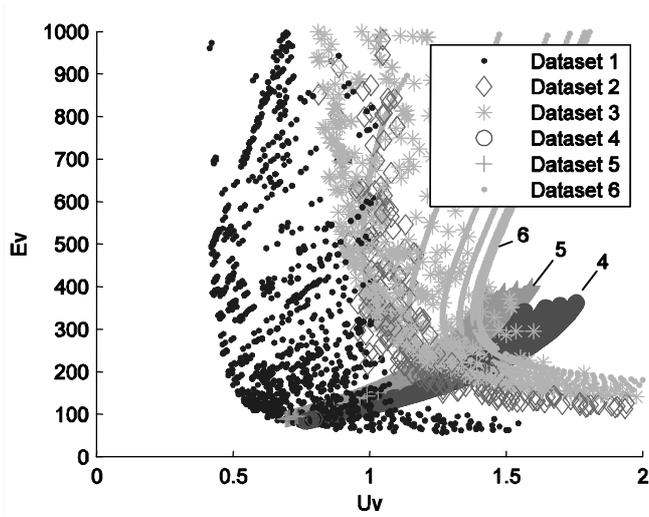


Fig. 4. Orthographic view of datasets 1-6 showing $E_v - U_v$ trade off. U_d axis out of page. Sectional plots given for this view.

B. Section views of Fig. 4

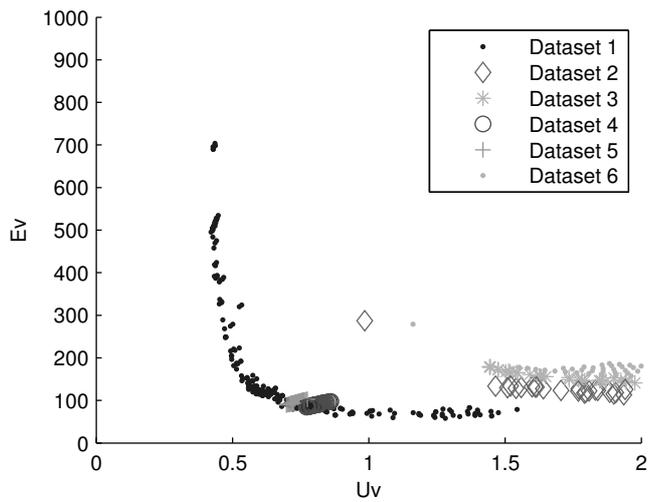


Fig. 5. Sectional view of Fig. 4. Datasets 4, 5 $T_s = 1.0-1.5$, $U_d = 0.20-0.44$ for remaining sets.

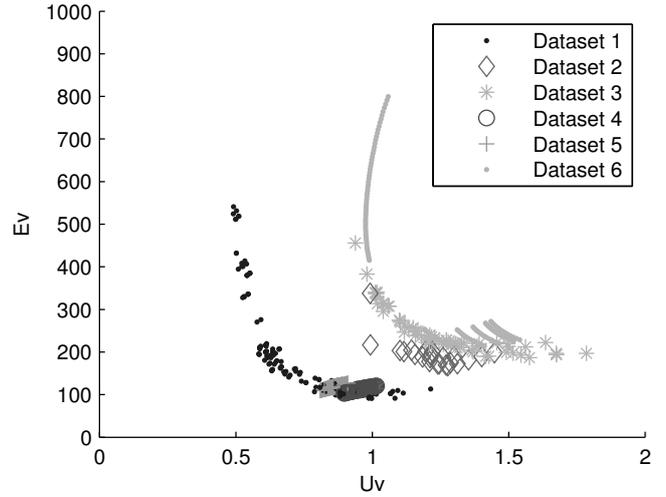


Fig. 6. Sectional view of Fig. 4. Datasets 4, 5 $T_s = 2.5-3.0$, $U_d = 0.058-0.077$ for remaining sets.

Fig. 5 – Fig. 7 were created by plotting points in datasets 4 and 5 within a range of values of T_s . Data points in other sets that lie in the same range of values of U_d as the points in dataset 5 were then added.

Dataset 1 defines a Pareto front for this problem. Datasets 4 and 5 are on or close to the Pareto front for a range of values of T_s however at the value of T_s selected for publication in [3] they no longer lie on the Pareto front, and datasets 3 and 6 are competitive with them. Since these controllers lie in the same region for this value of T_s claims of improving one element of the cost function are valid, but do not show the trade off with another cost. They also do not show the large improvement that can be attained for the methods published in [3] and [4] by reducing T_s .

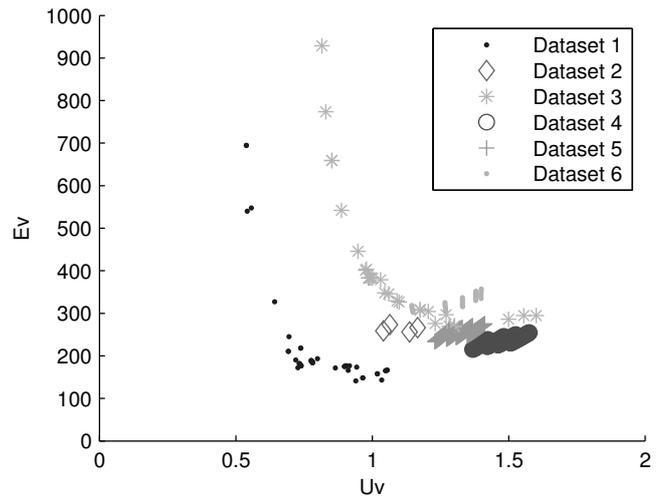


Fig. 7. Sectional view of Fig. 4. Datasets 4, 5 $T_s = 8.0-9.0$, $U_d = 0.048-0.054$ for remaining sets.

The method described in [6] is an optimal tuning of a PID controller for a range of values of U_d , but is not optimal for small values of U_d . This can be seen from Fig. 8

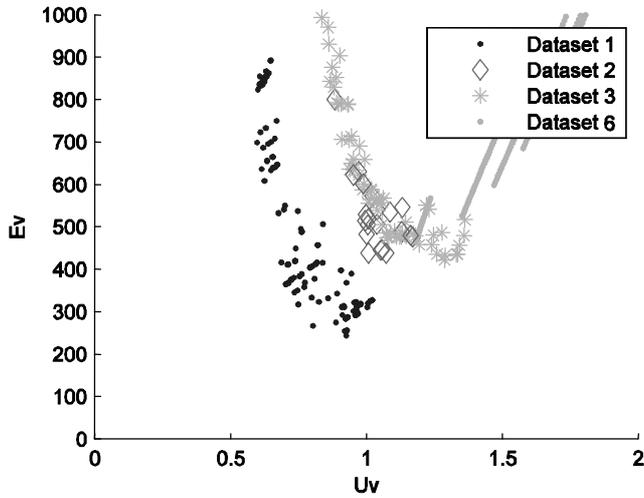


Fig. 8. Sectional view of Fig. 4. $U_d = 0.025-0.030$, datasets 4, 5 not visible in section.

C. Orthographic view of variation datasets

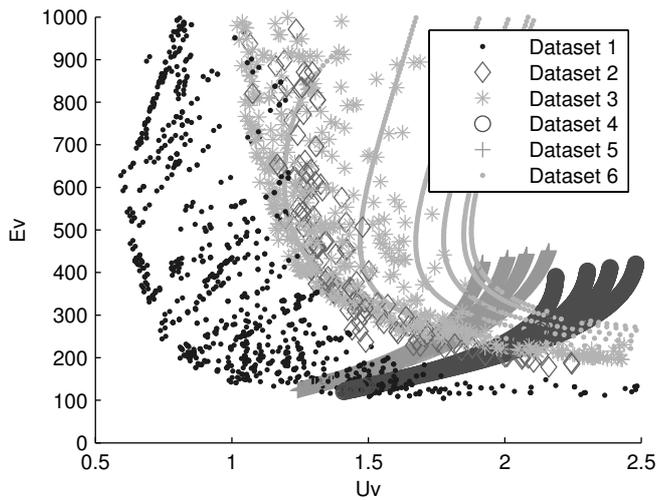


Fig. 9. Orthographic view of datasets 1-6 showing similar trends to Fig. 4 for the variation sets. Note U_v axis shifted by 0.5 units.

Comparing Fig. 9 to Fig. 4 shows that the relationship between the datasets is largely unchanged for a 10% positive variation in the process.

IX. SETPOINT RESPONSE

Setpoint response has not been discussed, but it is obviously affected by tuning T_s for [3] and [4]. The author has shown [7] that the robustness of the setpoint response is negatively affected by decreasing T_s . A designer may correct for setpoint response by adding a low pass prefilter to the setpoint, this

would allow a small value to be selected for T_s whilst maintaining the desired setpoint response. Alternatively, the disturbance estimator used in [3] and [4] may be migrated to the structure described in [6]. This modified controller would require two models of the process and other controllers, but could give superior disturbance rejection and accurate setpoint response.

X. CONCLUSION

In this work it has been shown that Pareto analysis is useful for analyzing controller design methodologies. It allows all significant tradeoffs to be addressed simultaneously. In this case the Pareto front was found to lie in three dimensional space. Visualizations of three dimensional data can be made using Matlab, however the key aspects of these moveable plots are difficult to capture in print. If the Pareto front lies in a higher dimension further work is required to analyze it.

The analysis has shown that current methods are being compared at a very restricted point, which does not lie on the Pareto front for disturbance rejection. PID controllers are not sufficient to obtain the best disturbance rejections possible for this example. For certain controllers the setpoint response time must be reduced to reach the Pareto front.

Using modern optimization techniques it is possible to tune a controller directly. However, these methods still require an existing controller strategy to optimize. Attempts by the author to tune generalized controllers to the integrating second order plus dead time process have been found to be computationally intensive and to yield poor results.

REFERENCES

- [1] M. R. Mataušek, and A. D. Micić, "A Modified Smith Predictor for Controlling a Process with an Integrator and Long Dead-Time," *IEEE Trans. Autom. Control*, vol. 41, no. 8, pp. 1199-1203, Aug. 1996.
- [2] M. R. Mataušek, and A. D. Micić, "On the Modified Smith Predictor for Controlling a Process with an Integrator and Long Dead-Time," *IEEE Trans. Autom. Control*, vol. 44, no. 8, pp. 1603-1606, Aug. 1999.
- [3] S. Majhi, and D. P. Atherton, "Obtaining controller parameters for a new Smith predictor using autotuning," *Automatica*, vol. 36, pp. 1651-1658, 2000.
- [4] I. Kaya, "Obtaining controller parameters for a new PI-PD Smith predictor using autotuning," *J. Process Control*, vol. 13, pp. 465-472, 2003.
- [5] K. J. Aström, C. C. Hang, and B. C. Lim, "A New Smith Predictor for Controlling a Process with an Integrator and Long Dead-Time," *IEEE Trans. Autom. Control*, vol. 39, no. 2, pp 343-345, Feb. 1994.
- [6] T. Liu, Y. Z. Cai, D. Y. Gu, and W. D. Zhang "New modified Smith predictor scheme for integrating and unstable processes with time delay," *IEE Proc.-Control Theory Appl.*, Vol. 152, No. 2, pp. 238-246, March 2005.
- [7] Internal report: "Interaction of Setpoint tracking and disturbance rejection in Smith predictors for integrating processes with dead-time," *Technical report for Control and Instrumentations Lab* Dec. 2006.
- [8] H. A. Abbass, R. Sarker, and C. Newton, "PDE: A Pareto-frontier Differential Evolution Approach for Multi-objective Optimization Problems," *Proc. IEEE Con. on Evolutionary Computation*, pp. 971-978, 2001.
- [9] A. Africa, and M. Braae, "An advanced analogue display for virtual instrumentation," *Elektron*, pp. 46-48, Feb. 2006.