Optimisation of Heterogeneous Ball Mill Systems (HBMS) Using Combined Multiestimator and Genetic Algorithm Based Switched Multicontroller System

Peter Tshuma
Department of Electronic Engineering,
National University of Science and Technology,
PO. Bag AC939 Ascot, Bulawayo,
Zimbabwe

Professor Emmanuel Nyakwende
Head of Department
Business Information Systems
University of Venda for Science & Technology
P Bag X5050 Thohoyandou 0950
Republic of South Africa

Professor Michael Collier
Chairman of Electronic Engineering,
National University of Science and Technology, PO. Bag AC939 Ascot, Bulawayo,
Zimbabwe

Abstract

Heterogeneous Ball Mills Systems (HBMS) are hybrid dynamic systems that are characterized by continuous and discrete behaviour. Their dynamics change with the wear and replenishment of spherical steel balls, raw material characteristics, or levels in the mill. The changes in the HBMS dynamics result in various regimes of control that form subsets of hybrid systems with increased operational costs and loss of efficiency. An appealing alternative is a combined Multiestimator and Genetic Algorithm Optimizer supervisory and multi-control system scheme, where different control laws are selected in each dynamic regime. We present a framework for multi-control and selection using combined Multiestimator and Genetic Algorithm Optimizer supervisory system, optimum performance states are derived for the normed-value of the HBMS's allowable unmodeled dynamics as well as for the system's ability for disturbance and error tracking.

Keywords: Hybrid systems-discrete-and-continuous systems, switching controllers, Multiestimator, and Genetic Algorithm Optimizer supervisory system, multi-control system HBMS, Ball mill system, spherical steel balls, and grinding circuit

1.1 Introduction

Heterogeneous Ball Mills Systems (HBMS) are hybrid dynamic systems that are characterized by continuous and discrete behaviour. The dynamic behaviour of Heterogeneous Ball Mills Systems (HBMS) is dependent on the combination of spherical steel balls and raw material levels (Lepore, Wouwer, Vande and Remy, 2002), mill speed and ball mill filling (Liddel and Moys, 1988, Dong and Moys, 2001), classifier settings (Rogers, Hassibi and Yang, 1999) and the state of the lifters or shell liners. The wear effect of spherical steel balls holdup diminishes the efficiency of the comminution process and introduces a strong nonlinearity through specific rate of breakage. (Lepore, Wouwer, Vande, and Remy, 2002) This means there is an irreversible accumulation of material in the mill with a decrease of the production as the balls wear. (Basting and Provost, No Date). Hence, it is necessary to control and maintain certain normal working level of spherical steel balls and raw material in the grinding circuit.

The control of Heterogeneous Ball Mill Systems (HBMS) remains a challenging problem and is the subject of considerable research; see (Basting and Provost, No Date, Buchholtz and Poschel in Wolf and Grassberger, (1997), Buchholtz, Freund, and Poschel, (2000), Jia and Li, (2000). HBMS have a number of autonomous trajectories that manifest themselves as the steel balls and shell liners wear or are replenished, and when the raw material changes its physical structure or properties. Generally, the control philosophy of HBMS optimizes the continuous element ignoring the discrete aspect of the behaviour, albeit with a reduced performance and productivity.

In the sequel, we provide a relatively uncluttered analysis of the HBMS dynamical behavior, an integrator and a multi-controller supervised by an estimator based and genetic algorithm (GA) optimizer employing dwell time switching. The estimator system has been considered previously in Morse, (1996), Morse, (No Date). It has been analyzed in one form or another in Kulkarni and Ramadge, (1996), Borrelli, Morse, and Mosca, (1997), Morse, (1997), Narendra and Balakrishnan, (1997) and elsewhere under various assumptions. Morse, (1997) has shown that the system's supervisor can successfully orchestrate the switching of a sequence of candidate multi-controllers into feedback with the system's imprecisely modeled process like the HBMS, so as to cause the output of the process to approach and track a constant reference input despite norm bounded unmodeled dynamics, and constant process disturbances and to insure that none of the signals within the overall system can grow without bound in response to bounded disturbance, be they constant or not.

Multiple Lyapunov functions were proposed for stability analysis of hybrid systems in Peleties and DeCarlo, (1991), Laferriere (1994), and Branicky, (1994), a controller design methodology based on multiple Lyapunov functions is described Malmborg, (1996), and Petterson and Lennartson (1996) and Johansson and Rantzer, (1997) makes an important contribution towards the application of multiple functions for practical controller design. Burridge and Rizzi (1996) employed the idea of guiding the system through a sequence of equilibrium points in order to stabilize the controller and they go on to make an assumption that is common to most of these works that every subsystem has the same equilibrium point, which has to be stabilized. However, hybrid systems can exhibit much richer behavior that can be used in HBMS: the system might switch between multiple equilibrium sets before reaching the final state. It is also commonly assumed that the switches between the controllers are either explicitly controlled, or that the switching surfaces can be explicitly characterized.

GA-based strategies Goldberg, (1984) Srinivans, (1994), and Zitzler (1999a, 199b), typically require some hundreds individuals for ensuring convergence. Moreover, when dealing with real-life optimization problems in control systems, the evaluation of each objective often requires a solution lasting several minutes Kim, (1998), Battistetti and others, (2000). This difficulty often makes the use of GA based strategies computationally unaffordable or highly unpractical from an industrial point of view.

The objective of this paper is to derive estimator based supervisory results with an addition of genetic algorithm for optimization of systems with changing dynamics like the HBMS processes This will be done for a supervisory control system in which the number of candidate controllers is finite, and the switching between candidate controllers is constrained in a sense to be made precise in the sequel. These restrictions not only greatly simplify the analysis in comparison with that given by Morse, (1997), but also make it possible to derive reasonably explicit upper bounds for the process's allowable unmodeled dynamics as well as for the system's disturbance-to-tracking error gain. We use a combined multiestimator and genetic algorithm optimizer for supervisory and switching of controllers with an optimum landscape.

Our work was influenced particularly by Pohleim and Hebner, (1996) they used genetic algorithm for optimal control of a greenhouse, Ursem, Krink, Jensen and Michalewicz (No Date) analyzed the modeling of control tasks in dynamic systems, Zefran and Burdick, (1998) designed the switching controllers for systems with changing dynamics. Sontag (1981), he used piecewise linear systems as underlying model for hybrid systems; Di Barba and others used, (2000). Multiobjective genetic algorithm for optimization of real time devices in electrical engineering Mehra, Smith, and Beard, (2000) used genetic algorithm for multi-spacecraft trajectory optimization. Laumanns, Laumanns, Laumanns, and Kitterer, (No Date) used evolutionary multiobjective integer programming for the design of adaptive cruise control system.

The description of Heterogeneous Ball Mill Systems(HBMS) is made in Section 1. The overall structure of multicontroller system, and family of realizations is described in Section 2. Section 3 gives the description of multiestimator based supervisory system. The genetic optimizer based system intended to support the multiestimator and its use on non-linear dynamics is described in Section 4. The main theorem characterizing the system's behavior is stated in Section 3. A simple, informal proof of the theorem is carried out in Section 4. The simulation process and results of the genetic algorithm Optimizer is given in Section 5. Section 6 is the discussion and evaluation of the use of genetic algorithm Optimizer on HBMS. The conclusion and future works are discussed in Section 7.

2. The Overall Multicontroller System

The aim of this section is to describe the structure of the supervisory control system to be considered in this paper. We begin with a description of the multicontroller system, and family of realizations,

2.1 The HBMS Multicontroller System

The overall problem of interest is to construct a supervisory system containing genetic algorithm optimizer used on processes like the HBMS. The HBMS process is presumed to admit the model of a non-linear system M whose transfer function from control input u(t) to measured output *y(t)* is a member of a known class of admissible transfer functions of the form

$$C_m = \bigcup_{m \in M} \{ v_m + \delta : |\delta| \leq \varepsilon_m \} \qquad (1)$$

$$v_m \overset{\Delta}{=} \alpha_m ! \beta_m \qquad (2)$$

is a prespecified, strictly proper, nominal transfer function, ε_m is a real non-negative number, δ is a proper stable transfer function whose poles all have real parts less than the negative of a prespecified stability margin $\lambda_c>0$, and $|\bullet|$ is the shifted infinity norm

$$|\delta| = \sup_{\omega\in\Re} |\delta(j\omega - \lambda_c)| \quad (3)$$

Prompted by the requirements of set-point control, it is assumed that the numerator of each transfer function in C_m is nonzero at s = 0. It is further assumed for each $m\in M$, that $\boldsymbol{\beta}_m$ is monic and that $\boldsymbol{\alpha}_m$ and $\boldsymbol{\beta}_m$ are coprime. All transfer functions in C_m are thus proper, but not necessarily stable rational functions. The specific model of the HBMS process to be controlled is shown in Figure 2. Here **y**(*t*) is the HBMS measured output and *d* is a disturbance.

The control task is to optimize the grinding circuit as the spherical balls wear or are replenished. Each state of the spherical steel balls and raw material is a sub-manifold m in a particular dynamic regime, $m\subseteq M_n$. Depending on the application; it might be necessary to achieve asymptotic optimization or maybe only convergence of the trajectories of the system to m. In both cases, the control task is complicated by the fact that it is not known in advance what manifolds the dynamical system will traverse. In particular, it is possible that the HBMS switches autonomously between different manifolds. Switching might also be unpredictable due to external disturbances such the change in raw material characteristics and the replenishment of steel balls.

Let the HBMS be in a given dynamic state. A natural way to control HBMS is to design a controller for each of the dynamic states. Therefore, for each manifold $m\in M$, we design a controller K:

$$K_i \stackrel{\Delta}{=} \{kp: m\in M\} \quad (4)$$

2.2 HBMS Family of Realizations

The evolution of the discrete state also depends on the discrete input such as the wear and replenishment rate of spherical steel balls. For each $m\in M$, $-\lambda_c$ is greater than the real parts of all of the closed loop poles of the feedback interconnection. It is presumed that a family of controllers is given, for each $k_m\in K$. These controllers are required to be chosen so that for each $m\in M_n$ is detectable and the controller is stabilizable. Morse et al, 1996 describes many different ways to construct such realizations, once one has in hand an upper bound n_m on the McMillan Degrees of the k_m Given the HBMS family of realizations, the sub-system to be supervised is thus of the form shown in Figure 2

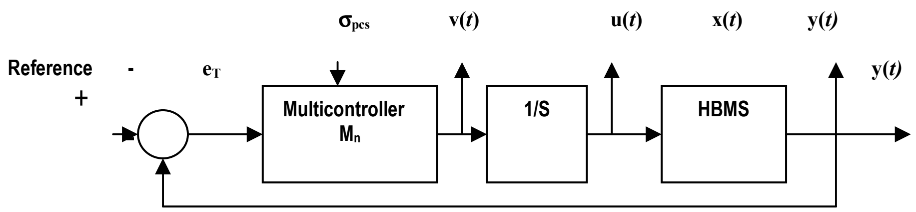

Fig: 2 Controller, the system, and the environment

where vector **x**(*t*) represents the internal state of the mill at time *t*, **v**(*t*) is the environment state, **u**(*t*) denotes the control signal, and **y**(*t*) is the output from the mill, M_n is the n_c dimensional state shared dynamical system called a multicontroller, **v**(*t*) is the input to the integrator, e_T is the tracking error and σ_{pcs} is a piecewise constant switching.

$$\dot{x}_{(m)} = A_\sigma x_m + b_\sigma e_T \quad (5)$$

$$v(m) = f_\sigma x_m + g_\sigma e_T \quad (6)$$

$$\dot{u}_{(t)} = v \quad (7)$$

$$e_T \stackrel{\Delta}{=} r - y(t) \quad (8)$$

3.1 Multiestimator Based Supervisor

The multiestimator shown in Fig 3 is an n_E-dimensional linear dynamic system. It generates $\mathbf{y}_r$ using an identifier estimator of the form:

$$\dot{x}_{(E)} = \begin{bmatrix} A_E & 0 \\ 0 & A_E \end{bmatrix} x_E + \begin{bmatrix} b_E \\ 0 \end{bmatrix} + \begin{bmatrix} b_E \\ 0 \end{bmatrix} y + \begin{bmatrix} b_E \\ b_E \end{bmatrix} \qquad (9)$$

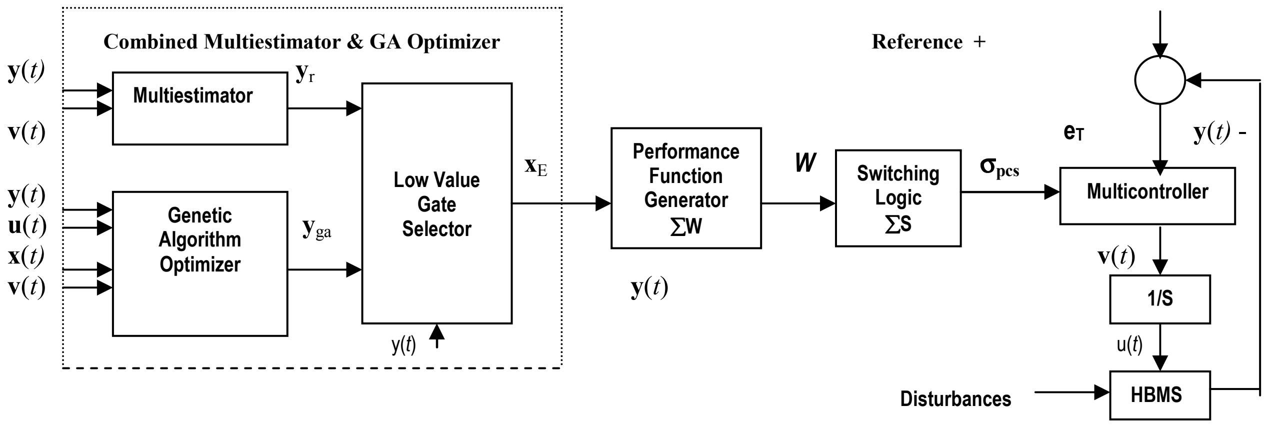

Fig 3 Combined Multiestimator, Genetic Algorithm Supervisor and Multicontroller

where

$$n_E \stackrel{\Delta}{=} 2(n_m+1)$$

and (A_E, b_E) is a parameter independent, (n_m+1) dimensional SISO controllable pair with $(\lambda_c I + A_E)$ stable. Here n_m is upper bound on the McMillan Degrees of v_m. Morse, (1996) explains in detail how to construct a function $m \rightarrow c_m$ so that for each $m \in M$

is a stabilizable realization of $(1/sv_m)$ whose uncontrollable eigenvalues have real parts less than $-\lambda_c$. The c_m enables the determination of estimation error x_E; through the low value gate selector. The performance function generator $\sum W$ processes the lowest of the two values y_r and y_{ga} .

4. $$\begin{bmatrix} A_E & 0 \\ 0 & A_E \end{bmatrix} + \begin{bmatrix} b_E \\ 0 \end{bmatrix} c_m + \begin{bmatrix} 0 \\ b_E \end{bmatrix} y + \begin{bmatrix} b_E \\ b_E \end{bmatrix}, c_m$$

The genetic optimizer based system is intended to support the multiestimator with selecting the non-linear dynamics of the HBMS. The HBMS environment, system, and control are monitored in order to determine the optimum switching value $\mathbf{y}_{ga}$. The objectives are defined to give a sufficient characterization of the optimizer-system's selection behavior considering energy consumption and cost of comminution. All these objective functions are computed within the simulation. Thus, the resulting genetic optimization problem can be stated as follows (where the function values of f and g calculated by the simulator).

$$\text{Minimize } f(x) = \begin{cases} f1(x) & \text{Energy consumption} \\ f2(x) & \text{Cost of comminution} \end{cases} \qquad (11)$$

$$\text{subject to } g(x) \geq E \min \text{ (minimum energy consumption)}$$
$$m \in M$$

For efficiency reasons, there is a constraint on the minimum energy consumption $g(x)$.

4.1 Genetic Algorithm Optimizer Modeling Shell

The genetic algorithm based supervisor optimizer shown in Fig 4 consists of a simulation shell that contains the current simulation step, the step length, the global time, an array for the performance measures, and three arrays for the control, system, and environment variables. Any variable is modeled by a data structure that contains its current value, a record of past values, the domain of the variable, and the parameters for drift, stochasticity, and periodicity. Moreover, the data structure contains a number of internal variables that are used for management of the genetic algorithm based supervisor optimizer variables (calculation of new state, resetting, etc.).

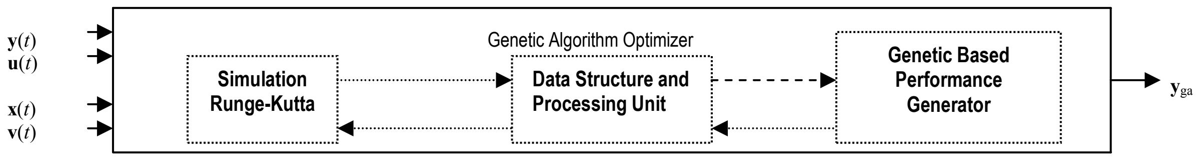

Fig 4 Genetic Algorithm Based Supervisor Optimizer

The HBMS Genetic Algorithm Based Supervisor Optimizer problem is modeled by the interactions between the controller, the system, and the surrounding environment. Here, vector $\mathbf{x}(t)$ represents the internal state of the mill at time t, $\mathbf{v}(t)$ is the environment state, $\mathbf{u}(t)$ denotes the control signal, and $\mathbf{y}(t)$ is the output from the mill. The change in HBMS state is usually modeled by a number of difference equations of the form:

$$x_i\,(t+h) = xi(t) + \Delta x_i(\mathbf{u};\, \mathbf{x};\, \mathbf{v};\, t;\, h) \qquad (12)$$

where x_i is the i_{th} system variable in $\mathbf{x}$, Δx_i (•) is the update function, t is the time, h is the length of a time-step, and $\mathbf{u}$, $\mathbf{x}$, and $\mathbf{v}$ are the control signals, the system state, and the environment state of previous time-steps (sometimes several steps in the past). Real systems are often described by a set of non-linear differential equations. In these cases, an approximation method, such as Runge-Kutta, is used as the update function Δx_i (•)

5. Genetic Algorithm Optimizer Simulation Process and Results

The optimization was carried out using the Multiple Population Genetic Algorithm. The following operators and parameters were used:

The chromosomes in the population are initialized and change with time. The peaks of the objective function are plotted as rings on the graph Figure 5. The objective function is both noisy and dynamic and dependent on the simulation process, therefore identifying the true optima at the centre is no use as it is not robust to the noise and dynamic behaviour.

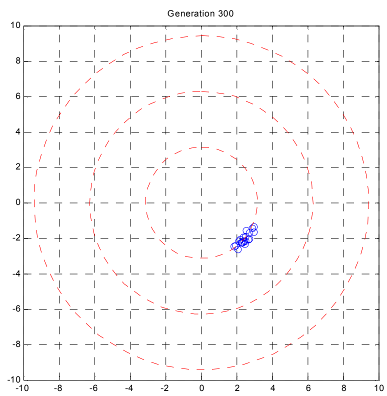

Fig 5: objective function are plotted as rings on the graph

The initial population converges rapidly to one or more clusters located on the rings. The sigma parameters converge quite quickly to a low value. The minimum value of σ_{ga} has been increased to make the algorithm search continuously for a better operating point. After a while, the population forms a single cluster, often on the side of a ring, which is quite robust. The outer rings are more robust to the dynamic behaviour, but have a slightly lower mean objective value. Occasionally the population may diverge, increasing σ_{ga} accordingly and settle on a new ring.

Using a diploid chromosome and a dominance mechanism. The chromosome has redundant information and is able to `remember' where it has been in the past. This mutating the dominance bits at the end of the chromosome may cause the chromosome to represent a very good past position. The algorithm tends to produce more transient clusters than the single chromosome implementation.

The essential goal of HBMS is that of identifying, in a completely automatic way, the system changes that affect behaviour and provide some prescribed performance for a given change, *e.g.* to minimize power consumption and materials cost or to maximize some output, taking into account physical constraints and geometrical bounds. Two objectives are defined to give a sufficient characterization of the HBMS behavior considering the efficiency and high throughput objective. The profit is equal to the income from the mill throughput and the minimum energy consumption. The penalty is enforced to avoid over flooding the mill and high-energy consumption. As stated earlier, direct control determines a solution's performance by simulating a number of steps into the future. In practice, this includes simulating the mill performance, or rather, predicting the mill performance in the control horizon. The change in HBMS state is modeled by non-linear differential equations (11) and simulated to initialize the genetic optimizer. The fitness of a solution σ_{ga} at time t is calculated as the profit achieved minus a penalty p:

$$Fit(x_m; t) = \sum^{t+\text{Control Horizon}} \Delta x \; (profit(j)\,)\; p(j) \qquad (13)$$

For control reasons there is a constraint on the minimum g (x)<0 and h (x)>0 between the estimator based supervisor and genetic algorithm optimizer.

The genetic algorithm was employed with the objective function of equation 11 and 13 the system model equation 11. All the standard parameters of the model were used. Fig 6 shows the evolution of the objective value for the best set of selection parameters $\mathbf{y}_{ga}$ found during an optimization.

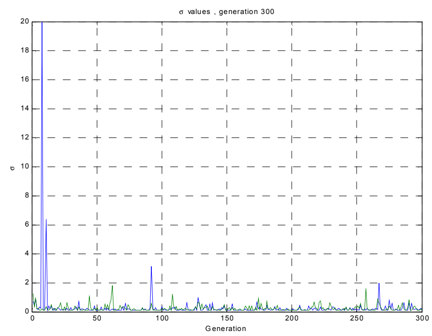

Fig. 6: Evolution of objective value for continuous system

6. Discussion and Evaluation on Use of Genetic Algorithm Optimizer on HBMS

The parallel use of multiestimator and genetic algorithm optimizer models to simulate real-world systems and select the best control for a given trajectory is the optimum solution for HBMS. The most important issue is the maximal allowed response time, which defines how fast the controller must react to ensure proper control. The main problem with genetic algorithm optimizer is that the calculation time for the response might be so long that the system state has changed substantially, thereby making the difference between the model and the real system too big. Hence, to overcome longtime delay in calculations, the multiestimator carries out the selecting task. This gives an opportunity of evolving controllers while the system is being controlled. If a better controller is evolved, it takes over the control of the real system. This technique allows the controller to adapt better to the HBMS and thereby compensate for long-term effects such as wear out of spherical steel balls or change in material properties.

7. Conclusions and Future Work

In this paper, we investigated the use of genetic algorithm for switching and selecting the optimum controller for use in real-world problems. The main motivation was the need for realistic optimization of dynamic systems, like the HBMS. In this context, we suggested a novel combined multiestimator and genetic algorithm optimizer for switching and controller selection for systems with changing dynamics. We demonstrated the potential of our combined multiestimator and genetic algorithm supervisory system in an HBMS.

It seems that the genetic algorithm optimizer introduced for switching and controller selection process are of value for modeling realistic dynamic problems. This conclusion is based on condition the following is taken into consideration:

- First, the models need direct interactions between the system components and creation of artificial dynamic problems where the shape and dynamics of the fitness landscape are introduced with relation to real problem.

- Second, even if the genetic algorithm optimizer could approximate the underlying dynamics by imitating the corresponding landscape one has to tune the system to allow the landscape of the real system to imitate it properly.

- Third, the previously introduced behaviour needs to be allowed for the optimization algorithm to affect the shape of the fitness landscape.

In our future work, we plan to concentrate on a few issues; these include: Real time control of the HBMS, (ii) investigation of discrete dynamic problems such as spherical steel balls replenishment, wear of shell liners on control and other mechanical failure problems.

8. References

1. Bastin G. and A. Provost, Feedback stabilisation with positive control of dissipative compartmental systems, Centre for Systems Engineering and Applied Mechanics (CESAME), Universite Catholique de Louvain, Belgium, (No Date)
2. Buchholtz, Volkhard and Poschel Thorsten, Force Distribution and Comminution in Ball Mills In: D. E. Wolf, P. Grassberger (Eds.) \Friction, Arching, Contact Dynamics" World Scientific (Singapore, 1997) p. 265-273.
3. Buchholtz, V., J.A. Freund, and T. Poschel Molecular dynamics of comminution in ball mills, Eur. Phys. J. B 16, 169{182 (2000) THE EUROPEAN PHYSICAL JOURNAL B EDP Sciences Societ_a Italiana di Fisica Springer-Verlag 2000 Comments on the history and current state. *IEEE Transactions on Evolutionary Computation 1*(1), 3–17.
4. Borrelli, D., A. S. Morse, and E. Mosca. Discrete time supervisory control of families of 2-degree of freedom linear set-point controllers. IEEE Transactions on Automatic Control, 1997.
5. Branicky, M. S. ``Stability of switched and hybrid systems," in Proceedings of the 33rd IEEE Conference on Decision and Control, (Lake Buena Vista, FL), pp. 3498--3503, 1994.
6. Burridge, R. R. Rizzi, A. A. and D. E. Koditschek, Sequential composition of dynamically dexterous robot behaviors." Preprint, 1996.
7. Di Barba, P.: A Fast Evolutionary Method for Identifying Non-inferior Solutions in Multicriteria Shape Optimization of a Shielded Reactor. Presented at 6th Intl OIPE Workshop, September 25-27, 2000, Torino (Italy).
8. Dong H. and Moys, M.H., Assessment of Discrete Element Method for one Ball Bouncing in a Grinding Mill, Int. J. Miner Process 65 (2002) 213-226, 17 August 2001.
9. Goldberg, D. E.: Genetic Algorithms in Search, Optimization, and Machine Learning. Addison Wesley, USA 1989
10. Jia Lixin and Li Xinzhong, SELF-OPTIMIZATION COMBINED WITH FUZZY LOGIC CONTROL FOR BALL MILL, Xi0an Jiaotong University, Xi0an, Shaan Xi, 710049, P. R. China, IJCSS, Vol.1, No.2, 2000
11. Johansson M. and A. Rantzer, ``On the computation of piecewise quadratic Lyapunov functions," in Proceedings of the 36th IEEE Conference on Decision and Control, (San Diego, CA), 1997
12. Kim, M. K., Lee, C., Jung, H.: Multiobjective Optimal Design of Three-phase Induction Motor using Improved Evolution Strategy. IEEE Trans. Mag., vol. 34, no. 5, pp. 2980-2983, 1998
13. Kulkarni S. R. and P. J. Ramadge. Model and controller selection policies based on output prediction errors. IEEE Transactions on Automatic Control, 41:1594--1604, 1996.
14. Laumanns, Nando, Laumanns, Marco and Kitterer, Hartmut Germany, Evolutionary Multi-objective Integer Programming for the Design of Adaptive Cruise Control Systems, (No Date)

15. . Liddel, K.S. and M.H. Moys, The effect of mill speed and filling on the behaviour of load in a rotary grinding mill, J.S. Afr, Inst. Metall, vol. 88, no 2 Feb 1988, pp49-57.
16. Laferriere, G. A. ``Discontinuous stabilizing feed back using partially defined Lyapunov functions," in Proceedings of the 33rd IEEE Conference on Decision and Control, (Orlando, FL), 1994.
17. Lepore, R. Wouwer, A. Vande and M. Remy, MODELING AND PREDICTIVE CONTROL OF CEMENT GRINDING CIRCUITS *Laboratoire d'Automatique, Faculté Polytechnique de Mons, 31 Boulevard Dolez, B-7000 Mons, Belgium* IFAC15th Triennial World Congress, Barcelona, Spain, 2002
18. Morse A. S., Supervisory control of families of linear set-point controllers - part Exact matching. IEEE Transactions on Automatic Control, pages 1413--1431,Oct 1996.
19. Morse A. S., A Bound for Disturbance to – Tracking- error Gain of Supervised set Point Control System, Springer-Verlag, Berlin, No Date.
20. Morse, A. S. Supervisory control of families of linear set-point controllers -part 2: Robustness. IEEE Transactions on Automatic Control, 42:1500--1515,Nov 1997.
21. Malmborg, J. Bernhardsson, B. M. and K. J. š MagNet Version 6, Getting Started Guide, Infolytica Corporation, 1999. http://www.infolytica.com/
22. Mehra, S.L., Smith, R., and Beard, R. Multi-spacecraft Trajectory Optimisation and Control Using Genetic Algorithm Techniques, IEEE, 2000
23. Narendra K. S. and J. Balakrishnan. Adaptive control using multiple models. IEEE Transactions on Automatic Control, pages 171 --187, fib 1997.
24. Narendra K. S. and A. M. Annaswamy. Stable Adaptive Systems. Prentice Hall, 1989.
25. Peleties P. and R. DeCarlo, ``Asymptotic stability of m-switched systems using Lyapunov-like functions," in American Control Conf., (Boston), pp. 1679--1684, 1991.
26. Pohleim, H, Genetic Algorithm Toolbox, Documentation Report, 1995].
27. Rogers, K.J., Hassibi. M. and M. Yang Advances in Fine Grinding & Mill System Application in the FGD Industry BR-1679 Presented to:EPRI-DOE-EPA Combined Utility Air Pollutant Symposium Control Atlanta, Georgia, U.S.A. August 16-20, 1999
28. Sontag, E. D. ``Nonlinear regulation: The piecewise linear approach," IEEE Transactions on Automatic Control, vol. AC26, pp. 346--358, 1981.
29. Srinivans, N., Deb, K.: Multiobjective Optimization using Non-dominated Sorting in Genetic Algorithms. E, vol. Comput. Vol. 2 no. 3, pp. 221-248, 1994.
30. Ursem, Rasmus, K., Thiemo Krink, Mikkel T. Jensen and Zbigniew Michalewicz, Analysis and Modeling of Control Tasks in Dynamic Systems, IEEE, (No Date)
31. Zefran M. Ÿ and J. W. Burdick, ``Stabilization of systems with changing dynamics," IEEE Transactions on Automatic Control, 1998.
32. Zitzler, E. E., Thiele, L.: Multiobjective Evolutionary Algorithms: A Comparative Case Study and the Strength Pareto Approach. IEEE Trans. E, vol. Comp. Vol. 3 no. 4, pp. 257-271, 1999a
33. Zitzler, Eckart Evolutionary Algorithms for Multiobjective Optimization: Methods and Applications, Thesis, Eidgenössische Technische Hochschule Zürich Swiss Federal Institute of Technology Zurich, November 11, 1999b