Proceedings of the 1999 Winter Simulation Conference
P. A. Farrington, H. B. Nembhard, D.T. Sturrock, and G.W. Evans, eds.

DESIGN AND EVALUATION OF A SELECTIVE ASSEMBLY STATION FOR HIGH PRECISION SCROLL COMPRESSOR SHELLS

Arne Thesen
Akachai Jantayavichit

Department of Industrial Engineering
University of Wisconsin-Madison
Madison, WI 53706, U.S.A.

ABSTRACT

Certain automotive parts call for assemblies to be produced to tolerances that cannot be economically reached using standard high volume machining practices. Shims are used instead. We show that the required precision may be reached by using selective assembly. An efficient selective assembly system is proposed. Simulation is used to evaluate the performance of this system, and configurations capable of tolerance improvements of up to 1/20 are suggested.

1 INTRODUCTION

The phrase *selective assembly* describes any technique used when components are assembled from sub-components such that the final assembly satisfies higher tolerance specifications than those used to make its sub-components. The use of selective assembly is inconsistent with the notion of interchangeable parts, and the technique is rarely used at this time. However, certain new technologies call for assemblies to be produced to a level of precision that is difficult to reach using standard high-volume machining practices. One such assembly is a scroll compressor produced by a major automobile manufacturer. Two highly machined cylindrical shells form the compression cavity in these compressors. In order to seal this cavity, the two shells must make physical contact at *both* ends. Unfortunately, the shells forming this cavity cannot be economically machined to sufficiently close tolerances to prevent leaks under high pressure. Instead, shims are used to reduce the size of the resulting gap to acceptable limits.

In this paper we introduce and evaluate a selective assembly station that permits top and bottom shells to be matched such that these higher tolerance requirement can be met. Shell tops and bottoms (Figure 1) arrive in separate input buffers. Here they queue up until the heads of each queue are jointly taken by a robot to a measuring station where they are classified and moved on to a small buffer storage of not-yet-assembled shells (Figure 2). This buffer is then searched for pairs of shells that meets the tolerance requirement. This search may yield no matches, in which case the buffer population has increased by two shells. Or, one of the incoming shells may have found a match, in which case the buffer population remains constant (although the content has changed). Finally each incoming shell may find a suitable match, in which case the buffer population will decease by two units.

Considerations in designing this assembly station include:

- How to identify matching shells?
- How much space should be used for buffer storage?
- What to do when the buffer is full?

We will show that the proposed selective assembly station can be used to reduce assembly errors by at least one order of magnitude. Furthermore, this performance gain is obtained without the need to maintain a large buffer storage, and without the need to reject any shell due to a failure to find a suitable match.

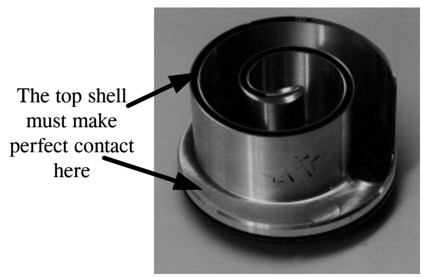

Figure 1: The "Bottom" Shell of an Automotive Scroll Compressor. The "Top" Shell is Similar.

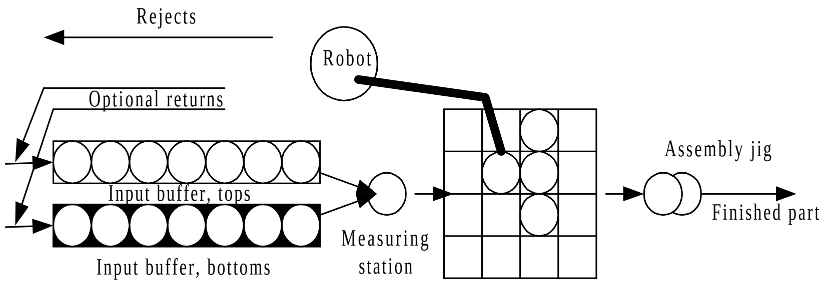

Figure 2: The Proposed Assembly Station.

2 PREVIOUS EFFORTS

Previous research in the area of selective assembly have mostly focused on the problem of matching pins to bushings such that a given tolerance specification is met (Gaillard 1945, Conway 1962; Wakefield 1964; Markov 1986; Bøjrke 1989). However, a few authors have also studied the problem ball bearing assembly (Rubenchik Mentov and Novikov 1979; Jones 1987 and 1989; Iyama et al. 1995). In all cases, the problem has been approached by first measuring incoming parts and assigning them to a suitable tolerance class, and then by matching parts belonging to corresponding classes. Some researchers have studied continuously operating systems where a buffer is replenished and depleted in real time (Boyer and Nazemetz 1985; Yamada 1994; Iyama et al. 1995). Others have studied batch processes where all the parts belonging to a given batch are simultaneously matches. (Pugh 1986 a and b and 1992 a and b; Fujino 1987; Fang and Zhang 1995 and 1996; Coullard Gamble and Jones 1997; Chan and Linn 1999). Specific contributions are discussed in the following paragraphs.

Boyer and Nazemetz (1985) proposed a statistical selective assembly method using a procedure by which the best component matches were chosen from two given sets of mating components. Using a simulation model, they showed that their method could greatly reduce assembly variability. Pugh (1986b) developed an enhanced computer program to generate group partitions for given component distributions and to determine the desired number of classes for selective assembly. He also used computer simulation to show that selective assembly is more powerful if components have nearly equal variability and if corresponding classes contain similar quantities of components (Pugh 1986a). He later used simulation to extend his work to compare the results of four statistical selective assembly techniques (Pugh 1992a). The results indicated a substantial improvement in the assemblies. Pugh (1992b) also studied the performance of selective assembly when components exhibit dissimilar variances. He found poor results with respect to selective assembly in this case.

Fujino (1987) proposed a new method, the “batch oriented system” which reduces the computing time when lot sizes are large. Lee, Hausman, and Guierrez. (1990) studied the problem of finding an optimal machine setting for the production of components in assembly operations. In addition, Berzak (1992) has discussed robotics techniques in selective assembly to improve product quality and reduce the cost. Yamada (1994) proposed a new method for optimizing the selective assembly process in an automated continuous production system. A method for evaluating the accuracy of a product based on a geometric design model was proposed by Arai (1992). His result indicated that increasing the number of classes improves the assembly accuracy, but decreases slightly the yield of the process. Iyama et al. (1995) used a Markov model to analyze a three-part ball bearing assembly. They found that the appropriate plan to produce component must consider both matching accuracy and buffer capacity.

Fang and Zhang (1995) presented an algorithm to minimize the number of surplus components by grouping the mating components based on two criteria: “balanced probability” and “unequal tolerance zone.” They recently studied another algorithm to quantitatively estimate the “matchable degree” in selective assembly via a predictive model with respect to set theory and probability method (Fang and Zhang 1996). This algorithm significantly reduces the surplus components after assembly by controlling both process capability and matchability degree. Coullard, Gamble, and Jones. (1997) considered the problem of maximizing yield of a matching problem within the context of a batch selective assembly system. They showed that when the cascading property is satisfied (as it is in most assembly problems), a greedy algorithm will find a matching that maximizes yield for all these problems. Zhang, Low, and Fang (1998) presented a new approach based on Process Capability Indices tolerancing. The goal of this approach is to jointly design tolerance and SPC parameters for selective assembly systems. Recently, Chan and Linn (1999) have proposed a new classification method for selective assembly. Their method can be generally applied to selective assembly of holes and shafts using any distribution estimates or true distributions.

Finally, we have not identified any published paper exploring the use of shims or the matching of parts in adjacent tolerancing classes. The second of these topics is a key focus of the research presented here.

3 STRATEGIES FOR SELECTIVE ASSEMBLY

Our approach to the selective assembly of compressor shells is based on the idea that the assembly station maintains a small buffer inventory of not-yet-assembled shell tops and bottoms. As new shells enter the station they are first measured and added to this buffer, then the buffer is searched to identify possible matches. There are many ways to implement this strategy. We consider a successful strategy to be one where:

- The average buffer population is small.
- No shells are rejected.
- Deadlock does not occur.

When the system is producing correctly machined parts, then there should be a suitable match for all incoming parts. Hence no discards should be called for. Note however that this is not the case if the two input processes are out of balance. In this case the use of shims or rejects may be necessary. This topic is beyond the scope of this paper.

The concern for deadlock stems from the fact that available buffer storage space will be finite in any implementation, hence the strategy must either guarantee that the buffer storage will never be full, or provide for corrective actions to be taken when full buffers are encountered. Strategies for meeting these goals are discussed in the following paragraphs. An evaluation of the strategies is given in Section 4.

3.1 Tolerance Classes

Assuming that the manufacturing errors for both top and bottom shells are normally distributed with identical standard deviations σ_m; then the gaps observed in assemblies using randomly selected components will be approximately normally distributed with a mean of 0 and a standard deviation σ_g equal to σ_m. Positive gaps are assumed to occur at the top of the assembly and negative gaps are assumed to occur at the bottom of the assembly. If shells with dimensional errors outside the $3\sigma_m$ range are discarded, then the largest observed gap (G_{max}) will be of size $6\sigma_m$. Furthermore, if the location of the gap is ignored, then the average size of a gap will be $0.78\sigma_m$ and the standard deviation will be $0.59\sigma_m$

In order to reduce the size of this gap it is necessary to have information about the actual dimensions of individual shells. However, dealing with actual measurements in a real time implementation can be difficult, and we will use a simpler classification scheme where the range of acceptable manufacturing errors (M_{max}) is partitioned into $2n$ equally sized, contiguous *tolerance classes* labeled [–n, -n+1, …,-1, 1,n ,,…, n-1, n.]. Incoming shells are measured and assigned an index corresponding to the corresponding tolerance class. From then on, shells are identified by their index only, and all information about actual measurements is lost. In this case the largest gap to be observed when two shells belonging to the same tolerance class are assembled will be M_{max} / $2n$. While it is beyond the scope of this paper, one of the advantages of using tolerancing classes is that the resulting system can be modeled as a Markov process, hence an analytic performance model can be developed at least for simple systems.

3.2 Matching Strategies

Two issues must be addressed when designing an algorithm for identifying acceptable matches. The first of these is to establish what constitutes an acceptable match. The second is to establish the order in which the items in the storage are to be evaluated. These considerations are briefly discussed below.

Acceptable matches. When the above classification scheme is used, then gaps no larger than $M_{max}/2n$ will result if shells belonging identical classes are matched. Similarly, gaps no larger than M_{max}/n will occur if shells belonging to adjacent classes (i.e. i and i+1 or i and i-1) are matched. The strategy of only matching shells belonging to identical classes will be refereed to as *one-to-one* matching, This strategy has the advantage that it is simple, and that the smallest possible gaps will occur for a given number of classes. Unfortunately, this strategy is not practical as it leads to overflowing buffer storages regardless of what size buffer is specified. The strategy of allowing matches of shells belonging to adjacent classes will be referred to as *one–to-three* matching. We will show that this approach leads to stable systems. A disadvantage of this approach is that twice as many classes are needed to reach the same level of precision as can be reached using the *one-to-one* strategy. Higher order strategies are also possible, but they are impractical as the number of classes needed to reach a given level of precision increases very quickly.

Search order. The search order is irrelevant when *one-to-one* matching is used. However, the search order will significantly affect the likelihood of obtaining successful matches when higher order matching strategies are used. A number of different search sequences are possible. We will search for matches in increasing order of their likelihood. This approach will be referred to as the *least-likely* strategy. Since $p_k < p_{k-1}$, incoming shells are more likely to find a match is their class index is small. We will therefore attempt to match parts with high index values before matching parts with smaller index values. The specific search sequence used for a system with 16 classes is presented in Figure 3. We believe that this sequence minimizes the average buffer population for all systems configurations. Proof of this is beyond the scope of this paper.

For reference purposes we will also consider two other strategies. Under the *most-likely* strategy, the search is performed in the reverse order of the least-likely strategy. We expect this strategy not to perform well, as shells that could be paired with hard-to-match shells are now more likely to be matched with more common shells. Under the *random* strategy, the search order is established at random whenever a search is to be made. An evaluation of these search sequences is given in Section 4.

3.3 Avoiding Deadlock

Deadlock will occur when there is no room for incoming shell in the buffer storage. Given the high volume of production anticipated for the proposed system, deadlock may eventually occur even for the simplest of designs. A strategy for dealing with these events is therefore required. We will consider three different strategies for dealing with incoming shells that cannot be mated to one of the shells in a full storage unit:

- *Ignore:* Try to allocate sufficient space to accommodate all space demands.
- *Discard:* Discard the incoming shells.
- *Return*: Return unmatched shells to a fixed position in the input buffer.

The *ignore* strategy can only be used when the largest possible buffer population is quite small. As we will see in the evaluation section, this is the case only when very few bins are used, and this strategy is impractical.

The *discard* strategy is simple to implement, and it may therefore be attractive. The performance of this strategy is measured by the fraction of shells that are rejected, and the strategy is economically feasible only when the system is configured such that few rejections are experienced.

The *return* strategy is a variation of the reject strategy; however, refused shells are now returned to a given position in the input buffer for later processing. This approach is likely to be successful if rejected shells are reinserted far enough back in the input stream such that the population in the storage buffer is sufficiently different when the shell reappears for processing.

3.4 Unbalanced Systems

An unbalanced system is one that has unequal production rates for tops and bottoms for at least two different tolerance classes. For example, if the tops are consistently shorter than the bottoms, then there will be a surplus of long bottoms and short tops. In this case it may not be possible to find suitable matches for all parts. In this paper we assume that an unbalanced production process is detected and corrected upstream. Work is underway to extend the design of the proposed system to identify this problem. Shims can then be used to compensate for systematic dimensional errors.

4 EVALUATION

In this section we present the results of simulation experiments designed to evaluate the performance of selective assembly systems with different storage capacities using the three deadlock avoidance strategies discussed above. Unless otherwise specified, all observations are from simulation experiments using 12 replications with 100,000 observations per replication, and confidence intervals are less than 2% of the observed mean. The following performance measures will be used:

$b(t,c,d)$ = Average buffer population.
$d(t,c,d)$ = Average discard rate.
$r(t,c,d)$ = Average return rate.

Where:

t = The number of tolerace classes used.
c = Maximum buffer capacity.
d = The deadlock avoidance policy used (r = return, d = discard, i = ignore).

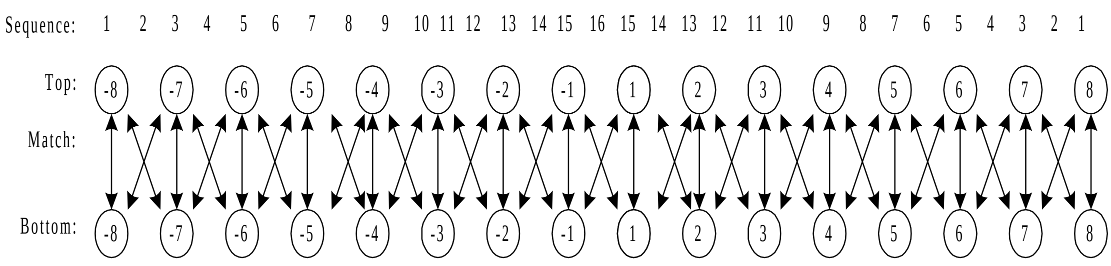

Figure 3: Search Sequence Used for Finding Matching Shells in a 16-class System.

4.1 Search Order

The relative performance of the three search strategies was evaluated for a wide range of configurations. In all cases it was found that the least *likely strategy* performed significantly better than the other strategies. In particular, the other strategies generate an excessive amount of rejects or returns. This is illustrated in Figure 4. Here the average buffer population for a 16 class system using the *return* strategy (i.e. *b(16,n,r)*) is shown for systems with different buffer capacities. Each point is the result of 12,000,000 observations. We note that the *least likely* search strategy leads to an average buffer population of about 12.7 when a buffer capacity of about 32 or more is used. The two other strategies performed significantly less well. Both strategies need a buffer capacity of at least 1,000 before reaching steady operations. Since this phenomenon was observed for all systems configurations we exclude the *random* and *most likely* strategies from further consideration. All results in the remainder of this paper are for systems using the least-likely search strategy.

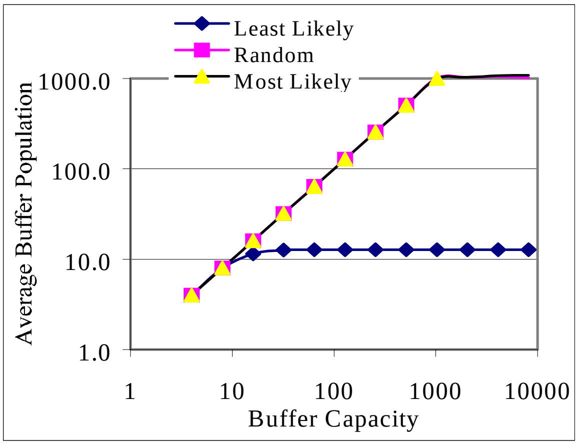

Figure 4: Performance of Three Search Strategies for Systems Using 16 Tolerance Classes and the *Return* Deadlock Avoidance Policy

4.2 Unlimited Buffer Capacity

We then estimated the average buffer population for systems with *t* tolerance classes and unrestricted buffer capacity ((*b(t, ,i)*). The results are seen in Figure 5. We note that the population is increasing at a decreasing rate as the number of classes (*t*) increases. We also collected data on how frequently different buffer populations were observed. To save space, we will not show the resulting empirical distribution functions. However, the resulting data support the conclusion that there is no theoretical upper bound on how large the population could be at any one point in time.

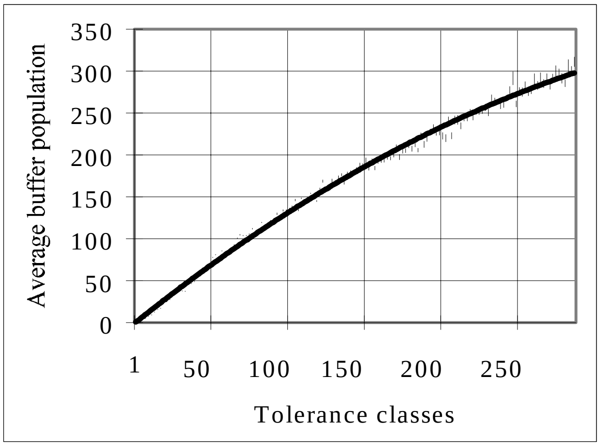

Figure 5: Observed Average Buffer Populations for Selective Assembly Systems with Unlimited Buffer Space. The Vertical Lines Indicate 95% Confidence Intervals. The Solid Curve is a Regression Estimate with $R^2 = 0.995$

4.3 Limited Buffer Capacity

Incoming parts will be discarded or returned if the buffer is full when they arrive. While this behavior is necessary to avoid deadlock, it is a non-productive use of time thatleads to increased cycle times. A large number of imulation experiments were conducted to evaluate the two policies designed to deal with this problem. The *discard* policy consistently led to a lower average buffer population than the *return* policy if all other factors remained the same. However, for the range of parameter values of interest in this study, their performances were quite similar, with the *discard* policy leading to slightly lower buffer utilization than the *return* policy. To save space, we will therefore only discuss the performance of the *return* policy in this section.

Few parts will be returned when the system's buffer capacity is fixed at a level significantly above the expected average buffer population. This is illustrated on the left side of Figure 6. For example, this figure shows that the average buffer population for a system using 18 tolerance classes is about 15 both for an unrestricted system and a system with a buffer capacity equal to 48. On the other hand, as the number of tolerance classes increases, the unrestricted buffer population increases freely while the restricted buffer population always stays equal to or below 48. This is illustrated on the right side of Figure 6 for a system with 48 tolerance classes. In this case, the average unrestricted population is about 65 while the average restricted population is about 47. A system's discard or return rate will increase significantly before the average buffer population reaches its limit. This is illustrated in the center of Figure 6 where the return rates for systems with different numbers of tolerance classes are shown. It is seen that the return rate increases exponentially if more than 24 tolerance classes are used. We note that the average buffer

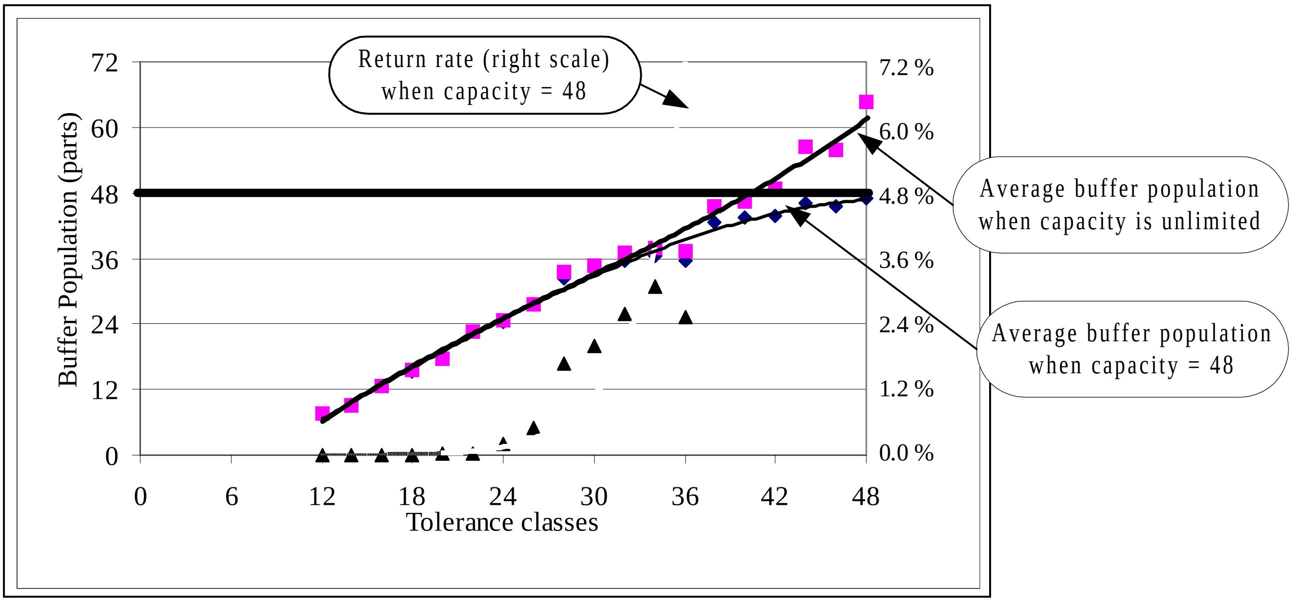

Figure 6: Performance of Selective Assembly System with a Buffer Capacity of 48 Parts. The Return Deadlock Avoidance Strategy is Used

population at this point is about half of the systems capacity limit. Similar behavior was observed for all other system configurations.

4.4 Size of Return Buffer

Since every return leads to an increase in the system's average cycle time, it is important to keep these to a minimum. Simulation studies were performed to estimate the return and discard rates for differently configured systems. Preliminary studies indicate that it is sufficient to insert a returned shell in a position in the input buffer corresponding to the system's buffer capacity, we will for simplicity restrict our presentation to a policy where shells are inserted in position 50 in the input buffer. (This corresponds a six-minute lag for systems with a six-second cycle time.)

4.5 Recommendations

Automated assembly systems producing parts such as the ones discussed here are likely to have cycle times in the 5 to 10 seconds per part range. This corresponds to a production rate of up to 720 parts per hour. We estimate that the operator should easily be able to handle one discard every 10 minutes. This corresponds to a discard rate of 0.33%. An automated returns system should be able to handle more returns. However, too many returns will reduce the average cycle time and it may lead to non-steady state operations. We arbitrarily estimate that an automated system should be able to handle no more than a 1% return rate. Using these rates, Table 1 gives recommended buffer allocations for systems designed to attain a given level of tolerance improvement.

Table 1: Recommended Buffer Capacity for Selective Assembly Systems with a Given Tolerance Improvement.

Tolerance improve-ment	Buffer capacity	Expected population	Discard rate (%)	Return rate (%)
1/4	12	3.1	0.01	0.4
1/8	28	12.5	0.29	0.6
1/12	48	24.3	0.16	0.8
1/16	60	36.3	0.29	0.8
1/20	72	45.0	0.29	1.0

5 SUMMARY AND CONCLUSION

We have introduced and evaluated a high-speed station for selective assembly of high precision automotive components. The station is capable of producing assemblies with a precision one order of magnitude higher than the precision of the incoming components. This improvement is reached by maintaining a small buffer storage of parts and by carefully matching parts with suitable measurements. An important design consideration is the capacity of this buffer storage. It is certain that the need to add parts to a full buffer will arise. This will cause deadlock unless a suitable deadlock avoidance strategy is adopted. Two strategies were evaluated. The first discards offending parts, while the second returns parts to the input buffer. We recommended configurations that would experience discards or returns at a rate of no more than six per hour. At this rate, an operator could easily handle the discarded parts, and the added cost of automatically returning parts to the input buffer may not be warranted. Instead, the robot could set aside all discards and they could be manually reintroduced in the system at the beginning of the next shift.

The systems evaluated here work well if errors for tops and bottoms are identically distributed. No selective assembly technique can compensate for production systems that produce components that do not match (i.e. too many small tops without a matching number of large bottoms). A modification of the present design using shims to compensate for systematic dimensional errors is under development

6 REFERENCES

Arai, T. 1992. A Simulation System on Assembly Accuracy. *Annuals of the CIRP* 41, 1: 37-40.

Berzak, N. Selective Robotic Assembly. 1992. *The 4th International Conference on Design Theory and Methodology,* 48:71-75. Scottsdale, Arizona, ASME Design Engineering Division.

Bøjrke, Ø.1989. *Computer-Aided Tolerancing*. 79-82. NewYork: ASME Press.

Boyer, D. E., and J. W. Nazemetz. 1985. Introducing Statistical Selective Assembly- A Means of Producting High Precision Assemblies from Low Precision Components. *Annual International Industrial Engineering Conference,* 562-570. Institute of Industrial Engineerings, Norcross, GA.

Chan, C. Ka and Linn J. Richard. 1999. A Grouping Method for Selective Assembly of Parts of Dissimilar Distributions. *Quality Engineering* 11, 2: 221-234.

Conway, H. G. 1962. *Engineering Tolerances.* 63-66. London: Sir Issac Pitman & Son Ltd.

Coullard, C. R., A. B. Gamble, and P. C. Jones. 1997. Matching Problems in Selective Assembly Operations. Working Paper, Dept. of I.E.M.S. Northwestern University.

Fang, X. D. and Y. Zhang. 1995. A New Algorithm for Minimizing the Surplus Parts in Selective Assembly. *Computer & Industrial Engineering* 28, 2: 341-350.

Fang, X. D. and Y. Zhang. 1996. Assuring the Matchable Degree in Selective Assembly via a Predictive Model Based on Set Theory and Probability Method. *Journal of Manufacturing Science and Engineering, Transactions of the ASME* 118, 2, May 1996: 252-258.

Fujino, Y. 1987. Matching Problem When Lot Size is Large. *Reports of Statistical Application Research,Union of Japanese Scientist and Engineers* 34, 4: 1-10.

Gaillaard, J. 1945. Selective Fits Are Easily Specified If Simple Instructions Are Followed. *American Machinist* 89,5, March 1945: 108-111.

Iyama, T., M. Mizuno, S. Goto, and T. Kogo. 1995. A Race-Matching Method for Ball Bearing Manufacture. *JSME International Journal Series C: Dynamics, Control, Robotics, Design and Manufacturing* 38, 1, Mar 1995: 167-17Jones, E., B. 1987. Sensors in Industrial Metrology. *Journal of Physics E: Scientific Instrument,* Europeans: Third Conference on Sensors and Their Applications 20, 9, Sep 1987: 1113-1126.

Jones, E., B. 1989. Industrial Metrology and Sensors. *Proceeding SPIE-Beam Deflection and Scanning Technologies, VLSI and Computer Peripherals*, 3/1-8. Institute of Electrical and Electronics Engineers, Piscataway, New Jersey.

Lee, H. L., Hausman, W. H., and Gutierrez, G. J. 1990. Optimal Machine Settings of Imperfect Component Production Processes for Assembly Operations. *Transactions on Robotics and Automation* 6, 6: 652-658. Institute of Electrical and Electronics Engineers, Piscataway, New Jersey.

Markov, G. N. 1986. Optimization of Resistor Selection for Voltage Dividers. *Measurement Techniques* 29, 4, 4 Apr. 1986: 337-339.

Pugh, G. A. 1986a. Group Formation in Selective Assembly. *ULTRATECH Conference Prooceeding,* 2: 2.117-2.123, Long Beach, CA.

Pugh, G. A. 1986b. Partitioning for Selective Assembly. *Computer & Industrial Engineering* 11, 1: 175-179.

Pugh, G. A. 1992a. A comparison of Selective Assembly Techniques. *Proceedings of the Summer Computer Simulation Conference,* 567-570. Society for Computer Simulation, San Diego, USA.

Pugh, G. A. 1992b. Selective Assembly with Components of Dissimilar Variance. *Computer & Industrial Engineering* 23, 4: 487-491.

Rubenchik, V. Ya, Mentov, K.V., and Novikov, V. V. 1979. Automatic Sorting of Rings by Size to Selective Assembly. *Measurement Techniques* 22, 1: 71-73.

Wakefield, L. P. 1964. *Dimensioning for Interchangeability,* 36-41. Oxford, New York, Pergamon Press LTD, Macmillan.

Yamada, Y. 1994. Combinatorial Optimization of Selective Assembly in Continuous Production. *Nihon Kikai Gakkai Ronbunshu. C Hen Transactions of the Japan Society of Mechanical Engineers, Part C.* 60, 573: 1877-1881.

Zhang, Y., Low S. Y. and Fang D. X. 1998. PCI-based Tolerances as an Interface between Design Specifications and Statistical Quality Control. *Computer & Industrial Engineering* 35, 1-2: 201-

AUTHOR BIOGRAPHIES

AKACHAI JANTAYAVICHIT is a graduate student at the University of Wisconsin-Madison. His research is in the area of selective assembly systems. He expects to earn his Ph.D. next spring.

ARNE THESEN is a Professor of Industrial Engineering at the University of Wisconsin-Madison. His current interests are in selective assembly and heuristic optimization methods.