Modeling and Throughput Prediction for Flexible Parts Feeders

Michael S. Branicky☆, Greg C. Causey†, Roger D. Quinn†
Case Western Reserve University (CWRU), Cleveland Ohio, 44106

Abstract

In this paper we illustrate a methodology for modeling and analyzing flexible feeders using generalized semi-Markov process (GSMP) models. Working through the simple case consisting of a single part being fed on a flexible feeder, we show how the throughput of the system may be obtained by both GSMP simulation and analytical techniques for GSMP models. Further, we demonstrate the predictive capability of such models. This is accomplished by generating and validating a model of the system feeding three distinct part types (at the same time) and then modifying the model to allow other feeding scenarios to be predicted. These scenarios include the effect of feeding the parts in a specific order, the effect of using a robot with different speed capabilities, and the effect of using a different sized presentation conveyor. We validate the predictions with physical testing.

1. Introduction

As the manufacturing market continues toward small-quantity, high-quality products, produced by reconfigurable automation, an ever increasing need for a truly universal parts feeding machine is being felt. While the complexities of the traditional "bin picking" problem are well known, vision-based flexible feeding systems (which solve a subset of the general problem) are beginning to be used and accepted in industry. Currently, a number of such feeders have been brought to market [1,2,3,4] or patented [5,6,7,8,9]. With such acceptance, however, comes the need for a more complete understanding of the operation of such systems. Without such understanding, the construction and programming of such systems will always be viewed as a ‘‘black art’’.

Feeding systems, as discussed above, while widely varying in physical design, can be characterized by three major features: a system for quasi-singulating parts, a system for determining the pose (position and orientation) of a singulated part, and a system for retrieving the part. This natural decomposition of the overall system into its constituent components allows flexible feeding systems to be modeled via general semi-Markov process (GSMP) models.

☆Department of Electrical Engineering and Computer Science
†Department of Mechanical and Aerospace Engineering
(msb11,gcc,rdq)@po.cwru.edu

1.1. Methodology

There are several steps required to construct and validate a GSMP model of a flexible parts feeding system. While the method currently requires physical testing and insight, it is expected that as the methodology matures, it will become less dependent on testing and more predictive in its capabilities. Four steps are required to construct and utilize a GSMP model for system simulation and throughput prediction:

1. Construct a model which is composed of discrete states and transitions from state to state. The physical system provides insight into the proper construction of the model.
2. Parameters for the model must be determined. Parameters include the distributions which describe the dwell times in each state and the probabilities of transitioning from one state to another.
3. Verify the model. This can be accomplished by two methods. This first is simply to simulate the model for a finite amount of time and examine the data which is produced. The second method is to analytically verify the model.
4. After the model has been verified, it may then be used to predict the throughput of the system under altered conditions. For example, if the part retrieval mechanism were replaced with one which was 50% faster, what would be the resulting throughput of the system?

An example application of the above procedure is undertaken in Section 2. The CWRU flexible parts feeder [10,11,12,13] was used for all the testing and modeling presented in the paper.

1.2. Previous Work

Some initial research has been performed in the area of feeder simulation. For example, several papers [14,15,16] have discussed methods of static and dynamic simulation of tumbling parts. The results are the percentage of parts which will rest in an orientation required for retrieval and assembly. Such techniques naturally complement the methodology above because they can be used to determine some of the model parameters alluded to above without testing. Goldberg and Gudmundsson [17] have examined the optimal conveyor speed setting for use in an Adept FlexFeeder by using a 2-D model of the physical system coupled with a statistical (Poisson process) model of the part arrivals. Goldberg *et al.* [18] estimate feeder throughput using conveyor and robot speed and the probability of stable parts poses. Branicky *et al.* [19] introduced the use of GSMPs to model flexible feeders and predict their performance.

1.3. Outline

In the following sections we endeavor to explain the method of modeling flexible feeders using GSMPs. We work through the modeling of a test case consisting of a single part being fed on a flexible feeder. We show how the throughput of the system may be obtained by both system simulation and by straight analysis of the GSMP. Next, we demonstrate the predictive capabilities of such models. This is accomplished by generating and validating a model of the system feeding three distinct part types (at the same time) and then modifying the model to allow other feeding scenarios to be predicted. These scenarios include the effect of feeding the parts in a specific order, the effect of using a robot with different speed capabilities, and the effect of using a different sized presentation conveyor. We validate the predictions with physical testing, and end with some conclusions.

2. GSMP Modeling

The aim of this section is to provide the background to enable modeling of flexible parts feeding systems via generalized semi-Markov process (GSMP) models. A GSMP (defined formally below) is basically a timed, probabilistic, finite-state machine with transition between states happening probabilistically and dwell times in any particular state given by a *general* distribution. Hence, ‘‘general’’ in the name arises from the fact that the distributions of dwell times are general and even may differ in their form (normal, exponential, etc.) from state to state. The ‘‘semi-Markov’’ of the name comes from the fact that the next discrete state is conditioned only on the current one, not the entire past (Markov); while the time remaining in the current state depends on how long we have already resided there (not Markov).

GSMPs have long been used in the modeling and analysis of queueing systems, which commonly arise in a variety of manufacturing applications [20]. We have found them to be similarly useful in modeling feeder systems [19], as we endeavor to illustrate in this paper. This is not surprising, since the feeder itself can be viewed as a tiny factory: parts arrive from the hopper, are queued by the conveyor, and processed by the vision system and robot.

The states of a GSMP model can be chosen using engineering insight. Typically, our states will denote conveyor advances, vision processing times, and the grasp times for various types of parts. The transition probabilities between states and the distributions of dwell times within states can be estimated from empirical data or derived from simpler models. Some examples are computed below. Once a model is constructed, it can be simulated to generate sample sequences of states and

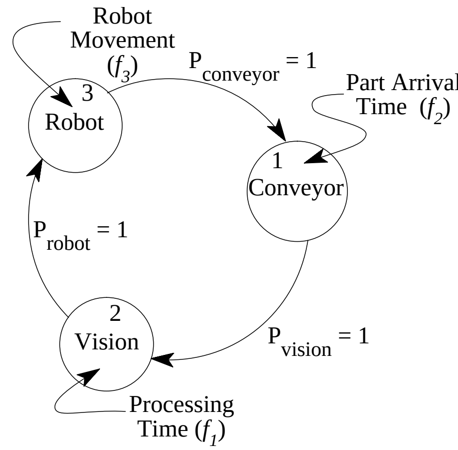

Figure 1: Simple Feeder Model

dwell times that are consistent with the statistics of the original underlying data. We also show examples of this below. Such simulations can be used to examine properties of the system that may be expensive (or impossible) to simulate in actuality (e.g., one can examine the effects of replacing the robot with one that is 20% faster). Finally, such models can be examined analytically in order to compute the means of certain variables. For instance, one could estimate the mean time until a certain collection of parts (necessary for some sub-assembly) could be fed by the system. We explain this type of analysis of GSMPs in detail.

2.1. GSMP Definition

For our purposes, a GSMP can be thought of as a finite set of states, S, with $|S|=n$; an n by n matrix of transition probabilities, $P=\{p_{ij}\}$, where p_{ij} is the probability of transitioning to state j given that the system is in state i; and a set of n probability distributions, f_i, representing the dwell time of the system in state i. By construction, the matrix P is a *stochastic matrix*, which is one with nonnegative entries whose rows add to one.

As an example, we briefly review the development of a model for the CWRU flexible feeder feeding a single part: 2¼’’ diameter plastic disks. See [19] for further details. Our feeder system contains three components: a parts presentation system (conveyor), a parts locating system (vision), and a parts retrieval system (robot). For the case of a single part, these sub-systems act sequentially, so it makes sense to consider a simple state model as shown in Figure 1. It has transition matrix

$$P = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix}$$

One can easily see that P is a stochastic matrix (despite the fact that the transitions are deterministic). It remains to determine the dwell time distributions representing part arrival times for the conveyor, processing time by the vision code, and robot movement time to pick and place the part. This was done using empirical testing as described in [19]. We found the conveyor dwell times (in seconds) to be modeled by an exponential distribution, f_1 (where λ_1=0.525); the vision by a shifted exponential, f_2 (λ_2=0.671, $t_{0,2}$=0.703); and the robot normal, f_3 (μ_3=1.60, σ_3=0.083).

As well as determining these distributions, we analyzed our empirical data [19] to obtain throughput information for the overall system and its various components. This data is reported in Table 1.

	Overall	Conveyor	Vision	Robot
Avg ppm	10.53	31.71	27.40	37.57
Std Dev	0.80	4.00	2.52	0.30

Table 1: Physical Throughputs

2.2. GSMP Simulation

The dynamics of a GSMP are as follows. Suppose that the system starts in state q. Then choose a dwell time, t_q, from the distribution f_q. At time t_q, transition to a new state r with probability p_{qr}. From there, pick a dwell time according to the distribution f_r, etc.

As an example, one may run simulations of the single-part model described above. In this case, the simulation is particularly simple, being given by the following pseudo-code:

Line 0: Set initial simulation time t=0, state transition index i=0, and initial state to be S_0=1, the Conveyor state. Goto Line S_0.

Line 1: Pick a time, t_i, from Conveyor's distribution. Add this to t. Increment i and compute the next state, S_i, according to the transition probabilities out-going from Conveyor. Goto Line S_i. (In this case, S_i always equals 2, the Vision state.)

Line 2: Pick a time, t_i, from Vision's distribution. Add this to t. Increment i and compute the next state, S_i, according to the transition probabilities out-going from Vision. Goto Line S_i. (In this case, S_i always equals 3, the Robot state.)

Line 3: Pick a time, t_i, from Robot's distribution. Add this to t. Increment i and compute the next state, S_i, according to the transition probabilities out-going from Robot. Goto Line S_i. (In this case, S_i always equals 1, the Conveyor state.)

The above simulation continues forever. If one wants to simulate the system for just T seconds, then one checks at each step whether the current dwell time, t_i, exceeds T-t. If it does, one breaks the simulation. There are various ways to deal with edge effects. For example, one may reset t:=t+t_i (obtaining a simulation time longer than T), use T-t as the last dwell time (obtaining a simulation time T), or throw it out (simulation time is t, which is less than T).

In any case, one can obtain throughput numbers by the resulting t_i and S_i from such simulations. Results for 30 hours of simulation time are tabulated in Table 2. Note that they compare quite nicely to the empirical data in Table 1.

	Overall	Conveyor	Vision	Robot
Avg ppm	10.26	30.93	26.90	36.86
Std Dev	0.53	4.52	2.68	0.28

Table 2: Simulated Throughputs

2.3. GSMP Analysis

In addition to simulation, one can use analytic means to glean certain information from GSMP models. Largely, these techniques are modifications of the ones that have been derived for discrete-time Markov chains [21], which we may think of simply as GSMPs with all dwell time distributions equal to the discrete value 1. The technique that we are interested in here is that of so-called *transient state analysis*. We quickly review the necessary mathematics to perform such analysis here, following [21], which should be consulted for more details and proofs of the stated results. Throughout, we use our single-part model as a running example for concreteness.

We say that state S_j is *accessible* from state S_i if by making only transitions that have nonzero probability it is possible to begin at S_i and arrive at S_j in some finite number of steps. A state S_i is always considered accessible from itself. For example, in Figure 1, all states are accessible from each other, in at most two steps. In general, accessibility can be determined by taking successive powers of the transition matrix P, since P gives the one-step transition probabilities, P^2 the two-step ones, etc. Specifically, let $p_{ij}^{(m)}$ be the ij^{th} element of the matrix P^m. If $p_{ij}^{(m)} > 0$ it is possible to go from S_i to S_j in m steps, since there is positive probability that the system would make such a transition. Thus, state S_j is accessible from S_i if and only if there exists $m \geq 0$ such that $p_{ij}^{(m)} > 0$. Back to our example, all states are accessible to themselves since $P^0 = I$ has nonzero diagonals. Examining $P^1 = P$ above, state 2 is accessible from 1, 3 from 2, and 1 from 3, each in a single step. Taking P^2, one sees that 3 is accessible from 1, 1 from 2, and 2 from 3, each in two steps.

Two states S_i and S_j are said to *communicate* if each is accessible from the other. In our example, every possible pair of states communicates. That is not usually the case. However, it is always possible to partition the set of states into *communicating classes*, in which each state within a class communicates with every other state in the class, but with no other state. In the special case where all states communicate, there is only one communicating class and the model is said to be *irreducible*. Otherwise, it is *reducible*. Our example is irreducible.

A communicating class C is said to be *closed* or *absorbing* if there are no possible transitions from the class C to any state outside C, i.e., no state outside C is accessible from C. A communicating class C is *transient* if it is not closed, i.e., if some state outside C is accessible from C. The states are called ‘‘transient’’ because (with probability one) the state of the model will eventually enter some closed communicating class.

As an example, consider the model shown in Figure 2. It contains three transient states, and one absorbing state. We call it a ‘‘success’’ model for the simple feeder since the absorbing state represents the fact that ‘‘one part was successfully fed, seen, and picked’’ given that the initial state is Conveyor. Its transition probability matrix is

$$P = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

This matrix is in terms of a *canonical form* in which the states are ordered to have all the transient states first, followed by those associated with the closed ones. In such a case, the transition matrix can be written in the partitioned form

$$P = \begin{bmatrix} Q & R \\ 0 & P_1 \end{bmatrix}$$

Assuming there are r transient states and n-r absorbing ones, the matrix Q is an $r \times r$ *substochastic matrix* (at least one row sum is less than one) representing the transition probabilities among transient states; R is an $r \times (n-r)$ matrix representing the transition probabilities from transient to absorbing states; and P_1 is an $(n-r) \times (n-r)$ stochastic matrix representing transitions within the closed classes.

The substochastic matrix Q completely determines the behavior of the model within the transient classes, and analysis questions concerning transient behavior can be stated in terms of Q. Actually, a central role in such analysis is played by the *fundamental matrix*

$$M = [I - Q]^{-1} = I + Q + Q^2 + Q^3 + \cdots .$$

By construction (see [21] for proofs), we have

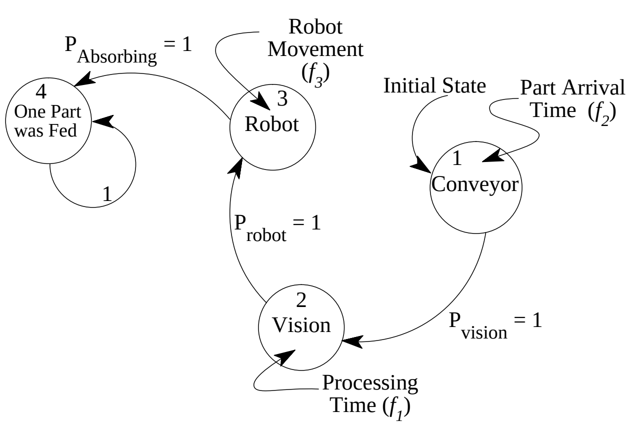

Figure 2: ‘‘Success’’ Model for Simple Feeder Model

Theorem 1 *The element m_{ij} of the fundamental matrix M is equal to the mean number of times the process is in transient state S_j if it is initiated in transient state S_i.*

Going back to our example, its fundamental matrix is

$$M = \left(\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} - \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} \right)^{-1}$$

$$= \begin{bmatrix} 1 & -1 & 0 \\ 0 & 1 & -1 \\ 0 & 0 & 1 \end{bmatrix}^{-1} = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{bmatrix}$$

Note that if we start in transient state i, each row gives the mean number of times we visit each other transient state before exiting to the absorbing state, which in this case represents a single part being fed, located, and placed. Now, if we want to know how much *time* the process spends in those transient states *in the mean*, we must simply multiply the row by the vector of means of the distributions f_i, $i = 1, 2, \cdots, r$. In our case, this vector is

$$\mu = \begin{bmatrix} 1/\lambda_1 \\ t_{0,2} + 1/\lambda_2 \\ \mu_3 \end{bmatrix} = \begin{bmatrix} 1.90 \\ 2.19 \\ 1.60 \end{bmatrix}$$

Thus, the mean time in seconds to feed one part starting in state 1 is given by the first component of $M\mu$: 5.69 s. The throughput is simply the reciprocal of this number, $1/5.69 = 0.1757 s^{-1}$ = 10.5 / min. This agrees quite well with the physical data!

In general, the mean amount of time that a GSMP stays in the set of transient states if started in transient state i is the i^{th} component of $M\,\mu$, which is given by

$$e_i\, M\, \mu,$$

where e_i is the row vector with a 1 in the i^{th} position and zeros elsewhere.

3. Predicting Feeder Throughput

In this section we begin to show the predictive capability of the GSMP model. First we review a GSMP model simulating feeding 3 different part types at once in a single feeder. This becomes the base-line case from which the other feeding scenarios (in which we predict system throughput) are based. First, a novel feeding situation is examined in which the same three parts (as the base-line) are fed, however, the parts must be retrieved in a specific order. Next we examine the effect of using a different robot (one which operates at a different speed) as the part retrieval mechanism. This is simulated in the physical system by changing (in software) the speed at which the robot operates. Lastly, we examine the situation in which the presentation conveyor is of a different size. This is simulated by altering the size (in software) of the processed vision window.

3.1. Base-Line Test: Parts in Any Order

In this case, which forms the base-line from which the predictive cases are based (first presented in [19]), a mixture of hex nuts and plastic sockets were fed.

3.1.1. Physical Testing

Data was collected (for approximately 30 hours) to determine the throughput of the overall system and sub-systems. Table 3 shows the average and standard deviation of the overall feeder and the feeder sub-systems for this test.

	Overall	Conveyor	Vision	Robot
Avg ppm	30.21	256.89	110.57	49.71
Std Dev	0.56	28.66	4.43	0.28

Table 3: Physical Throughput: Base-Line Test

3.1.2. GSMP Simulation

For simulation, a model as shown in Figure 3 was constructed. The vision system served as the central state (this is expected since the vision system ‘‘drives’’ the rest of the system, i.e., the vision determines when the conveyor needs to advance and when there is a part available for retrieval). From the vision system state, there are a total of five possible states into which the system may transition: retrieve a ⅜’’ nut, retrieve a 5⁄16’’ nut, retrieve a socket, big conveyor advance, or small conveyor advance. The probability of entering each of the states from the vision state is derived from experimental data. The distributions used and their parameters are described below. The transition probabilities, depicted in Figure 3, are shown in Table 4.

P_A	P_B	P_C	P_D	P_E
0.3394	0.3445	0.0633	0.0316	0.2212

Table 4: GSMP Transition Probabilities: Base-Line Test

The vision state is modeled by a time-shifted exponential distribution (λ=3.38, t_0=0.11), where the time-shift is the minimum time required to determine that the conveyor is empty.

The 2nd, 3rd, and 4th state is the retrieval of a ⅜’’ nut, 5⁄16’’ nut, and socket (respectively). The dwell times are normally distributed with data (μ=1.20, σ=0.063), (μ=1.20, σ=0.061), and (μ=1.28, σ=0.128) respectively.

The last two states represent the conveyor, which is modeled by one of two deterministic values. As discussed previously [19, 22, 23], the conveyor system operates by ‘‘loading’’ the horizontal conveyor in one large move and then advances parts into the workcell by smaller motions of only the horizontal conveyor. The two values are then the time for a large move (2.41) and the time for a smaller move (0.45).

The six distributions are summarized in Table 5.

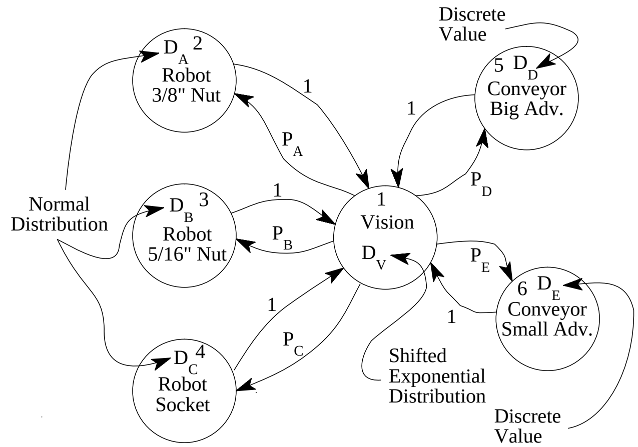

Figure 3: GSMP Model: Base-Line Test

State	Distribution	Parameters
Vision	D_V: Shifted Exp.	λ=3.38, t_0=0.11
Robot: ⅜’’ Nut	D_A: Normal	μ=1.20, σ=0.063
Robot: 5⁄16’’ Nut	D_B: Normal	μ=1.20, σ=0.061
Robot: Socket	D_C: Normal	μ=1.28, σ=0.128
Conv: Big Adv.	D_D: Discrete	t_{fixed} = 2.41
Conv: Small Adv.	D_E: Discrete	t_{fixed} = 0.45

Table 5: State Distribution Values: Base-Line Test

Together, these data represent a GSMP model of the flexible feeder system feeding three parts and can be used to determine system throughput. Table 6 shows the results of simulating the feeder.

	Overall	Conveyor	Vision	Robot
Avg ppm	30.01	259.90	110.38	49.38
Std Dev	1.09	55.07	7.41	0.25

Table 6: GSMP Simulation: Base-Line Test

3.1.3. GSMP Analysis

We may also use a ‘‘success’’ model derived from the one in Figure 3 to compute overall throughput for the base-line test. If we construct this model to achieve success if any one part is fed, it can be used to compute the mean time to grab a single part. That ‘‘success’’ model (not shown) has transition matrix

$$P = \begin{bmatrix} 0 & P_A & P_B & P_C & P_D & P_E & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

with Q being its principal 6×6 submatrix. The values of $P_A, \ldots, P_E$ were reported earlier, and it is easy to compute that the model has μ=[0.406 1.20 1.20 1.28 2.41 0.454]^T. In this case, 60/(e_1 M μ)=30.2 parts per minute. One can also compute the conveyor, vision, and robot throughputs by only adding the times used by those sub-systems in processing each part, that is by using, respectively,

$\mu_C = [0\ 0\ 0\ 0\ 2.41\ 0.454]^T$,

$\mu_V = [0.406\ 0\ 0\ 0\ 0\ 0]^T$,

$\mu_R = [0\ 1.20\ 1.20\ 1.28\ 0\ 0]^T$,

where T denotes the transpose. Using these mean vectors, we compute sub-system times in seconds of 0.2361, 0.5428, and 1.207, respectively. The throughputs in ppm for each sub-system are shown in Table 7.

	Overall	Conveyor	Vision	Robot
Avg ppm	30.2	254.1	111.5	49.7

Table 7: GSMP Analysis: Base-Line Test

Using the sub-system times, however, one can derive a simpler, ‘‘single-part model’’ for the base-line test. In this model, we reduce the state model (Figure 3) to include only three states. It is then much like the model shown in Figure 1. For future reference, we call it the Simplified Base-Line Model. It has the same P as the Equation in section 3.1.3, but the means of the distribution are changed to reflect those computed above. That is, the conveyor state has mean dwell time 0.2361 s, the vision state has one of 0.5428 s, and the robot’s is 1.207 s.

From this simple model, we can again determine the throughput of the overall system and sub-systems (using the GSMP analysis; the ‘‘success’’ model for the Simplified Base-Line Model is defined analogously to that in Figure 2). The analytic results are identical to those that appear in Table 7.

3.2. Effects of Parts in Order

The first example of using a GSMP model to predict feeder throughput is a novel feeding situation: feeding three part type in a specific order. This example was first shown in [19] and is reviewed here for completeness.

3.2.1. GSMP Simulation

To simulate this situation, a GSMP model of the system, shown in Figure 4, was created. In this model, the same transition probabilities and distribution parameters were used as in the base-line case. Using these values, the throughput data shown in Table 8 was generated by the simulation. The overall three-part system throughput (time to retrieve a single series of three parts) had mean 3.71 and standard deviation 0.53.

	Overall	Conveyor	Vision	Robot
Avg ppm	10.94	47.47	20.57	47.96
Std Dev	1.57	11.10	3.65	0.50

Table 8: GSMP Simulation: Part-in-Order

3.2.2. GSMP Analysis

Again, we may also compute the throughput of the system using analytical means. To compute overall throughput for this test, our ‘‘success’’ model has initial state S_1 and looks exactly like that in Figure 4, except that the arc from S_{12} to S_1 has destination state S_{13}, which has

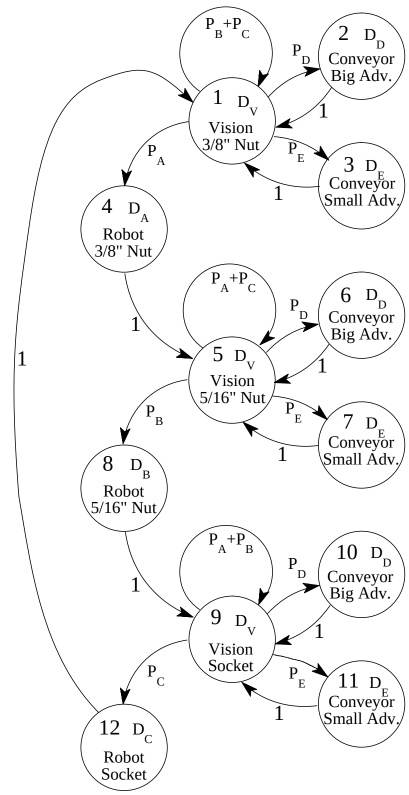

Figure 4: GSMP Model: Parts-in-Order

a deterministic (value = 1) self-transition. One can easily write down Q for this test model. It is the same as the P for the original GSMP model, except that the 1 in location $p_{12,2}$ is replaced with a 0. Computing three-part system throughput in this situation yields 3.69 ppm. This agrees closely with the simulation data. Again, one can compute the sub-system throughputs using

$\mu_C = [0\ 1\ 1\ 0\ 0\ 1\ 1\ 0\ 0\ 1\ 1\ 0]^T$,

$\mu_V = [1\ 0\ 0\ 0\ 1\ 0\ 0\ 0\ 1\ 0\ 0\ 0]^T$,

$\mu_R = [0\ 0\ 0\ 1\ 0\ 0\ 0\ 1\ 0\ 0\ 0\ 1]^T$.

To convert these back to atomic parts per minute (vs. complete assemblies per minute), we multiply the composite assemblies per minute and each of these numbers by three, arriving at the predictions in Table 9.

	Overall	Conveyor	Vision	Robot
Avg ppm	11.1	47.1	20.5	47.1

Table 9: GSMP Analysis: Parts-in-Order

3.2.3. Physical Testing

To test the validity of this simulation, the system was programmed to retrieve parts in the specified order. Approximately 24 hours of data was recorded and examined. Table 10 shows the results of this test. Part throughput had mean 3.14 and standard deviation 0.97.

	Overall	Conveyor	Vision	Robot
Avg ppm	9.43	28.35	21.06	49.73
Std Dev	2.91	13.43	7.04	0.66

Table 10: Physical Testing: Parts-in-Order

Comparing these results shows good agreement between the simulation and analysis and the physical test. The discrepancy is mostly due to the conveyor system. In the physical test, it was noticed that the system would occasionally be starved for sockets. This is because the sockets arrive in ‘‘batches.’’ This phenomena is not captured by the simulation and may account for the difference.

3.3. Effects of Different Feeder Hardware

Three simulations were conducted to determine the effect that altering a feeder sub-system would have on the overall system throughput. In the first two cases, simulating a robot with different operating characteristics (overall speed) was examined. The last case examines the effect that a different sized presentation conveyor would have on system throughput. These types of question are relevant to real world problems when one must determine the components to use in a feeder. For example, given a selection of robots from various manufacturers with different operating characteristics and costs, it would be beneficial to determine the appropriate robot which would successfully complete the feeding task while at the same time reducing overall system cost. In the same manner, one could examine the effect of the size of the presentation conveyor on the system throughput. Given the limited real estate of most table-top assembly robots, correctly sizing the presentation conveyor is important in maximizing the use of the robot’s work envelope.

To simulate a different model robot with a different speed characteristics, our current Adept 550 was programmed to run at reduced speeds while the system was operated. Two speed reduction tests were performed. The first operated the robot at 50% rated speed and the second operated the robot at 10% rated speed. To determine the true speed of the robot relative to operation at 100% rated speed (which was used in the base-line case), the robot was cycled repeatedly while cycle times were gathered. This was necessary due to the fact that the robot speed is not linearly related to the programmed percentage of full speed (i.e.50% programmed speed does not correspond ½ physical speed). Data was collected at 10% intervals on the range from 10% to 120% speed. From this data, the average speed of the robot at 10% and 50% was determined. (At 100% speed, the average robot move time was 1.08 s, at 50% speed, move time was 1.89 s, at 10% speed, move time was 8.15 s)

To simulate a change in the size of the feeding conveyor, the vision window in which parts are located was reduced in software. The window was half the size of the original window in both the *x* and *y* directions. This is analogous to reducing the width of the presentation conveyor by ½. (It was necessary to reduce the height of the vision window so that the same aspect ratio as the vision window on the full width conveyor would be realized.) This effectively reduced the area of the window to ¼ its original size.

In the following three situations, only the GSMP analysis, as discussed previously, was performed.

3.3.1. Robot at 50% Full Speed

3.3.1.1. GSMP Analysis

The ‘‘success’’ models for the Slow-Robot test are the same as in the base-line test. In fact, the only thing that changes is μ. In the case of 50% programmed robot speed, the means go to $\mu=[0.406\ 1.89\ 1.89\ 1.89\ 2.41\ 0.454]^T$. Hence, throughputs in ppm for the overall system and sub-systems are predicted to reduce (from the Base-Line Model) to those shown in Table 11.

	Overall	Conveyor	Vision	Robot
Avg ppm	22.5	254	111	31.7

Table 11: GSMP Analysis: 50% Robot Speed

3.3.1.2. Physical Testing

To test the validity of the simulation, the system was tested with the robot operating at 50% full speed. Table 12 shows the results of the physical test.

	Overall	Conveyor	Vision	Robot
Avg ppm	23.39	296.17	115.83	32.62
Std Dev	0.46	44.49	6.87	0.31

Table 12: Physical Test: 50% Robot Speed

3.3.2. Robot at 10% Full Speed

3.3.2.1. GSMP Analysis

Again, the only change from the base-line test is μ. In the case of 10% programmed robot speed, the means go to $\mu=[0.406\ 8.2\ 8.2\ 8.2\ 2.41\ 0.454]^T$. Hence, throughputs in ppm for the overall system and sub-systems are predicted to go down further to those shown in Table 13.

	Overall	Conveyor	Vision	Robot
Avg ppm	6.68	254	111	7.32

Table 13: GSMP Analysis: 10% Robot Speed

3.3.2.2. Physical Testing

To test the validity of the simulation, the system was tested with the robot operating at 10% full speed. Table 14 shows the results of the physical test.

	Overall	Conveyor	Vision	Robot
Avg ppm	6.67	322.24	113.11	7.26
Std Dev	0.16	69.45	11.63	0.15

Table 14: Physical Test: 10% Robot Speed

3.3.3. ½ Width Presentation Conveyor

3.3.3.1. GSMP Analysis

The ‘‘success’’ models for the Smaller-Vision-Window test are the same as the Simplified Base-Line Model defined in Section 3.1.3. However, with a window of ¼ the original size, the conveyor throughput goes down by ¼. Thus the inter-arrival time for the conveyor gets multiplied by 4, and the total mean time to retrieve a part is 4*(0.2361)+0.5428+1.207=2.694 s. This corresponds to a predicted throughput of 22.3 ppm.

	Overall	Conveyor	Vision	Robot
Avg ppm	22.3	63.5	111	49.7

Table 15: GSMP Analysis: ½ Width Conveyor

3.3.3.2. Physical Testing

During physical testing a single change was made to the software which controlled the conveyors. The distance which the presentation conveyor advanced was cut in half. This was necessary to prevent graspable parts from being moved pasted the smaller vision window. To offset this effect, the number of advances of the presentation conveyor before a loading process (operation of both the inclined and presentation conveyors) was doubled so that the total distance traveled by the presentation conveyor was consistent with the base-line case.

Table 16 shows the results of the physical test.

	Overall	Conveyor	Vision	Robot
Avg ppm	22.33	64.19	88.57	56.45
Std Dev	9.33	9.33	9.63	4.66

Table 16: Physical Test: ½ Width Conveyor

4. Conclusions

In this paper, we explained a methodology for modeling flexible feeders using generalized semi-Markov process (GSMP) models. Our methodology consists of the following steps:

- Examine the flexible feeding configuration.
- Create a GSMP model.
- Test the physical system to determine parameters.
- Validate models' correctness.
- Develop new model for a novel feeding situation.
- Verify new model's predictions.

Starting with the simple case of a single part being fed on a flexible feeder, we demonstrated the above in a variety of feeding situations. Specifically, we presented examples illustrating how to create a GSMP model and then populate the parameters describing its evolution. We also showed how the throughput of the system may be obtained by both system simulation and by pure analysis of the GSMP. We further demonstrated the predictive capability of such models. This was accomplished by generating and validating models of the system feeding three distinct part types (at the same time) and then modifying the model to allow other feeding scenarios to be predicted. These scenarios included the effect of feeding the parts in a specific order, the effect of using a robot with different speed capabilities, and the effect of using a different sized presentation conveyor. Finally, we validated the predictions with physical testing.

References

1 SonyFA, Advanced Parts Orientating System, T-APOS. Company Product Literature. 560 Route 303 Rangeburg, New York 10962.

2 Adept Robotics, Adept FlexFeeder 250, Company Product Literature, 150 Rose Orchard Way, San Jose, CA, Sept. 1995.

3 S. Gordon. Flexible feeding of small parts. *Proc. 1996 RIA Flexible Parts Feeding for Automated Handling and Assembly Workshop*. Minneapolis, MN, October 1996.

4 E.M. Ross. Flexible feeding devices for robotic assembly. *Proc. 1994 RIA Flexible Parts Feeding for Automated Handling and Assembly Workshop*. Cincinnati, OH, October 1994.

5 T. Nonaka and K. Otsuka. Photoelectric control system for parts orientation. U.S. Patent #4333558. 8 June 1982.

6 J.L. Goodrich and William L. Devlin. Programmable parts feeder. U.S. Patent #4608646. 26 August 1986.

7 B.R. Carlisle. Flexible parts feeder. U.S. Patent #5687831. 18 November 1997.

8 S.J. Gordon. Programmable reconfigurable parts feeder. U.S. Patent #5314055. 24 May 1994.

9 D. Herndon *et al*. Robotically controlled component feed mechanism visually monitoring part orientation. U.S. Patent #4909376. 20 March 1990.

10 R.D. Quinn *et al*. Design of an agile manufacturing workcell for light mechanical applications. *Proc. IEEE International Conf. Robotics and Automation*. Minneapolis, MN, April 1996.

11 F.L. Merat *et al*. Advances in agile manufacturing. *Proc. IEEE Conf. Robotics and Automation*, Albuquerque, NM, April 1997.

12 G. Causey and R. Quinn. Design of a flexible parts feeding system. *Proc. IEEE International Conf. Robotics and Automation*. Albuquerque, NM, April 1997.

13 G.C. Causey. *Elements of Agility in Manufacturing*. Ph.D. Dissertation, Dept. of Mechanical and Aerospace Engineering , Case Western Reserve University, January 1999.

14 B. Mirtich *et al*. Estimating pose statistics for robotic parts feeders. *Proc. IEEE International Conf. Robotics and Automation*. Minneapolis, MN, April 1996.

15 J. Craig. Simulation-based robot cell design in AdeptRapid. *Proc. IEEE International Conf. Robotics and Automation*. Albuquerque, NM, April 1997.

16 D. Berkowitz and J. Canny. Designing parts feeders using dynamic simulation. *Proc. IEEE International Conf. Robotics and Automation*. Minneapolis, MN, April 1996.

17 D. Gudmundsson and K. Goldberg. Tuning robotic part feeder parameters to maximize throughput. *Proc. IEEE International Conf. Robotics and Automation*. Albuquerque, NM, April 1997.

18 K. Goldberg *et al*. Estimating throughput for a flexible parts feeder. *Proc. International Symp. Experimental Robotics*. June 1995.

19 M.S. Branicky, *et al*. Toward a Science of Flexible Feeding. *Proc. IEEE International Conf. Advanced Intelligent Mechatronics (AIM)*. Atlanta, GA, Sept. 1999.

20 C.G. Cassandras, *Discrete Event Systems: Modeling and Performance Analysis*, Richard D. Irwin, Homewood, IL, 1993.

21 D.G. Luenberger, *Introduction to Dynamic Systems: Theory, Models, and Applications*, Wiley, New York, 1979.

22 G. Causey and R. Quinn. Testing and Analysis of a Flexible Feeding System. *Proc. IEEE International Conf. Robotics and Automation*. Detroit, MI, April 1999.

23 G.C. Causey. *Elements of Agility in Manufacturing*. Ph.D. Dissertation, Dept. of Mechanical and Aerospace Engineering , Case Western Reserve University, January 1999.