System Dynamics Models for Markov Processes

Pratap K. J. Mohapatra
Indian Institute of Technology Kharagpur

Rahul Kumar Roy
Indian Institute of Management Calcutta

Abstract : Markov models and System Dynamics models are aparently applicable to two completely different kinds of problems. However, structurally, they can be proved to be equivalent to each other. This paper establishes this equivalence. Critical observation have been made with regard to similarity and aparent differences between the two methodologies. The paper has also proposed a procedure for converting Markov models into system dynamics models. Examples have been drawn from Birth-Death process, M/M/1 Queue, Poisson Process and Yule Process to illustrate the method. It has been shown that such a framework makes the model for stochastic processes much more transparent and enables the system analyst to undestand the behaviour better.

P. K. J. Mohapatra : Prof. Mohapatra is a senior faculty member at the Department of Industrial Engineering & Management of IIT Kharagpur. His interest areas include modelling and simulation. He serves on the board of editors for "System Dynamics Review.

Rahul Kumar Roy : Sri Roy is an M. Tech '84 in Induatrial management from IIT Kharagpur. He currently works a Senior Systems Analyst at the computer centre of IIM Calcutta.

Acknowledgement : The authors greatly acknowledge various comments made by Prof. George P. Richardson on an earlier paper on the basis of which the current paper has been prepared.

1. Introduction

Markov and system dynamics models apparently belong to totally different fields. They differ in the way they approach the problem of systems analysis. The language and methodologcal instruments, each of them uses while modelling, also differ. In two articles published more than a decade ago, Sahin (1979a, 1979b) showed that stationary discrete-time Markov models are algebraically equivalent to a class of system dynamics models. He also delineated some advantages of this equivalence. However, these articles have apparently gone unnoticed by the mainstream system dynamicists. Wood (1983), Mosekilde and Rasmussen (1983) have also used system dynamics to a few cases of stochastic processes. They have however not been able to propose any generalized framework.

The present article reiterates the structural equivalence underlying the general Markov models and a class of system dynamics models. Examples are drawn from continuous time Markov processes. Both stationary and nonstationary situations are discussed. The paper begins with an introduction to the Markov processes in order to familiarize a reader with this branch of study. The structural equivalence is dealt with in subsequent chapters.

2. The Markov Modelling

A Markovian view considers a system as essentially non-deterministic or stochastic. Thus it uses the language of probability theory, a language normally used in the study of any stochastic process. A system governed by a stochastic process is assumed to be in any one of a finite (or countably infinite) number of states, each state representing a particular condition of the system. Let X(t) be the state occupied by the system at time t. Then X(t) is a random variable. These random variables are not independent since, usually, past and present states of the system influence the future states. The dependence relations among the random variables can be specified by giving the joint distribution function of every finite family $X(t_1)$, $X(t_2)$, ..., $X(t_n)$ of variables of the process. But this would be absurdly impractical to attempt for any real world situation. At this point we need Markov's simplifying assumption which states that given the present state, the past states have no influence on the future. This is the celebrated Markov property of forgetfulness (or memoryless). In formal terms, a stochastic process is Markovian if

$$\mathbf{P}\{a \le X(t) \le b \mid X(t_1)=x_1,\ldots,X(t_n)=x_n\} = \mathbf{P}\{a \le X(t) \le b \mid X(t_n)=x_n\}$$

whenever $t_1<t_2<\ldots<t_n<t$.

Markov models can be broadly classified into four kinds depending upon whether X is discrete or continuous and whether t is discrete or continuous. Discrete-state discrete-time models (also called Markov chains) and discrete-state continuous-time models (also called Markov processes) are the two popular kinds of Markov processes.

The possible values, X(t)'s can assume, may have either a qualitative or a quantitative meaning. For example, while in a queuing system, X(t) denotes the number of customers in the system at time t (a quantitative expression of the condition of the system), in a machine repair system, it denotes the physical conditions, such as operating or nonoperating, of a machine in a qualitative manner. However in every case it represents the state of the system at any point in time.

The change in a system is depicted in a Markov model by considering that the condition changes with the passage of a suitable time step. In other words the system is said to have made a transition from the current state to one of all possible states. These transitions take place according to a set of probabilities known as single step transition probability. If the system continues to occupy the same state then a virtual transition is said to have occurred. Transition probabilities hold the key to the representation of dynamics of a system. Thus a continuous time Markov process is completely described by its transition probability function $p_{ij}(t)$ which is the probability that the system is in state j at time t if it was in state i at time 0. $p_{ij}(t+\Delta t)$ is then given by

$$p_{ij}(t+\Delta t) = \mathbf{P}\{X(t+\Delta t)=j \mid X(0)=i\}$$

$$= \sum_k \mathbf{P}\{X(t+\Delta t)=j, X(t)=k \mid X(0)=i\}$$

By applying Bayes' theorem, one obtains

$$p_{ij}(t+\Delta t) = \sum_k \mathbf{P}\{X(t+\Delta t)=j \mid X(t)=k, X(0)=i\}\mathbf{P}\{X(t)=k \mid X(0)=i\}$$

By applying Markov's assumption of forgetfulness,

$$p_{ij}(t+\Delta t) = \sum_k \mathbf{P}\{X(t+\Delta t)=j \mid X(t)=k\}\mathbf{P}\{X(t)=k \mid X(0)=i\}$$

Thus it is the sum of the probabilities of transition from state i to j over all states i.

$$\varnothing_j(t) = \sum_{i=1}^{N} p_{ij}(t) \qquad [6]$$

The instantaneous rate of change of state probabilities can be derived as

$$\frac{d}{dt}\varnothing_j(t) = \sum_{i=1}^{N} \varnothing_i(t)\,\lambda_{ij} \qquad [7]$$

Defining $\Phi(t)$ as the row vector of state probabilities, these state equations can be obtained in the following vector-matrix form :

$$\frac{d}{dt}\Phi(t) = \Phi(t)\,\Lambda \qquad [8]$$

Taking transpose of both sides and defining $\underline{X}(t) = \Phi^T(t)$, and $A = \Lambda$ one obtains

$$\dot{\underline{X}}(t) = A\,\underline{X}(t) \qquad [9]$$

This is the familiar vector-matrix state differential equation of a linear system. Thus we conclude that stationary continuous time Markov processes are representative of autonomous linear systems.

3. The Similiarities and Differences between Markov and System Dynamics Modelling

System dynamics views systems as essentially deterministic though facilities exist to capture real life randomness by considering random variates of desired probability density functions. However, such random variates are generally used as mere noises contaminating the variables that are otherwise generated deterministically as functions of other variables in the system.

System dynamics normally delves deep into the cause-effect relationships working among the individual elements of a system. Such a view is conspicuously absent in Markov modelling and constitutes a fundamental paradigmic difference between the two approaches.

Like Markov models, states constitute the most important building blocks in system dynamics models. But they are conceptualised differently in the two approaches. In system dynamics, states are natural accumulations in physical flows (pure levels) or in information flows (smoothed levels), have physical meaning, and are invariably specified by quantitative measures. States are not random variables; thus neither the language of probability theory finds a place in the usual parlance of system dynamics, nor is it allowed that the system occupies only one state at any point in time.

The above mentioned differences notwithstanding, there are some apparent similarities between the two methodologies. The obvious similarity is that both the methodologies are applicable to dynamic systems. Both view a system as a collection of states. Both rest on a property of memoryless.

We recall that the Markov property of forgetfulness makes the future value of a state of a system dependent on its present value only, and not on the past values. The whole theory of Markov processes is founded on this assumption. Therefore, this property is explicit in Markov models.

System dynamics, on the other hand, makes no mention of such forgetfulness property. But the principles of system dynamics modelling rest on such assumption. Instantaneous change of state variables (rates) depends on the present value of state variables, not on their past values. While enunciating the principles of system (dynamics modelling), Forrester (1969) remarks " a policy governing a rate of flow can be responsive only to the available information at the particular point in system" . And it is imperative that the term *available information* implies that the magnitude of rate variables at any point in time depends on the value of level at that moment. Forrester's remarks therefore clearly point to the memoryless property of system dynamics models.

In this context, two features of system dynamics may be discussed here. They are models of (1) pipeline (discrete) delays, and (2) the boxcar trains (BOXLIN and BOXCYC functions) available in DYNAMO. Apparently, theses two features counter the memoryless property that we claim to be fundamental to system dynamics models.

A pipeline delay makes its output rate exactly equal to the input rate prevailing delay time constants back. This gives an impression that the present value of the output rate does not depend on the present state of the system, but depends on the past state. Since this past value has to be stored in the memory, it gives a feeling as if the pipeline delay does not display the memoryless property. We counter this argument in the following manner. Firstly, in its pure form, system dynamics is supposed to model continuous-time systems only and hence should not, strictly speaking, admit discrete delays. Secondly, such delays are modelled in practice as n cascaded first order delays where n equals the delay constant divided by the solution time interval DT. Thus, there is no necessity to store the past values of a rate variable to obtain present values of the rate variables in pipeline delays. If however, a software allows the values of input rate variables to be stored in memory for later use in evaluating the value of rate variables it definitely does not display a memoryless property; but concurrently, it also violates an important principle of system (dynamics modelling) by permitting a rate-to-rate coupling.

DYNAMO has a facility called boxcar train (BOXLIN and BOXCYC) where the contents of a level gets tranferred to its neighbouring level after a fixed time interval. Whereas BOXLIN allows the contents of the last level, in a chain of levels, to dissipate away, in BOXCYC the contents of the last level get transferred to the first level. One therefore gets a feeling that this phenomenon violates memoryless property. A close scrutiny of the mechanism however suggests that each of the boxcar trains have an alternating level-rate configuration with the rates dependent on the levels from where it originates and on a time varying parameter. The parameter assumes a value 1 only when the length of the simulation run divided by the time constant associated with each level-rate configuration is an integer, otherwise the parameter takes a value of zero. Modelled in this manner, there is no need to store old values of levels and so the boxcar trains can't be said to violate the memoryless property. We have to admit however, that both pipeline delays and boxcar trains can be modelled expeditiously by storing old values and using them later.

In summary we may say that if system dynamics models are strictly continuous, or even mildly discrete (meaning table function type of discontinuities in the information flow), then these models will display a memoryless property. While modelling such discrete phenomena as pipeline delays and boxcar trains, a model builder can take recourse to the expeditious way of storing old values and thus memorizing. In this context we may say that capturing history does not go against the idea of memoryless property. System dynamics models capture history by smoothing the variables, and by disaggregating level variables in physical flows. Smoothing attempts to capture history by weighing past information with a sequence of exponentially decreasing weight. A smoothed level does capture complete historical information. More and more historical information however becomes available as more and more of such smoothing variables are cascaded. As the number of such levels equals the smoothing time-constant divided by the solution time interval, the whole process becomes equivalent to a pipeline delay. In such a case full information on the past is captured in the model. However the property of memorylessness is never countered.

History can also be captured by disaggregating levels in physical flows. Sahin (1979a) suggests that defining four (age groups) levels in Meadows et al.'s world model, compared to only one in Forrester's world model, has enabled the former to capture history better.

It is to be noted that a model builder decides the extent to which history should be captured in his model. Having decided that, he has to fall back upon the principles of system dynamics modelling to define the flow rates as functions of present values of levels, thus displaying a memoryless property.

We end by saying that the founding principles of system dynamics modelling implicitly allow a memoryless property of the model. In order to capture complexities of real system and thus to enhance the scope of a model, a model builder often takes the liberty of incorporating features in his model, which are not strictly permissible by the principles of system dynamics modelling. Thus one may use rate-to-rate couplings and may even store past values of variables for defining some rate variables.

Apart from this strong similarity in the property of these models, some more similarities and difference exist between the two methodologies. System dynamics models are generally nonlinear. Linear system dynamics models are considered simple. The converse is true for Markov models. Most of the popular Markov models are stationary and therefore linear. Markov models with time-varying transfer rates are also linear.

Both descrete and continuous time systems are amenable to Markov modelling, requiring either difference equation or differential equation representation. System dynamics models, in contrast, make a continuous time assumption always ending up with differential equation representation. Though attempts have been made in the past to depict discrete events in system dynamics models, representation in system dynamics models depicting discrete-time systems in the form of difference equations are not available. Methodologically, system dynamics is not deficient at modelling discrete-time systems but it takes a philosophic stand that changes in systems take place only continuously. Since continuous changes are justified at higher levels of aggregation, system dynamics models tend to be very aggregative in nature.

Earlier it was shown that stationary Markov processes are equivalent to autonomous linear systems.

In the field of system dynamics there are instances where highly nonlinear models are linearized (Mohapatra 1978) for the purpose of analysis and subsequent design yielding differential equations similar to those for the continuous parameter Markov processes.

Another common feature of both the methodologies is the presence of feedback loops. System dynamics models build upon explicitly defined feedback loops. In Markov models although material feedback loops are clearly visible, the information feedback is rather less obvious. It is only when one studies the vecter matrix differential equation for the state probabilities, that the existence of information feedback becomes evident. These feedback loops are taken for granted and they are not mentioned explicitly.

Another commonality between the two approaches is the versatility of their use. Markov modelling has risen above all the competing methodologies in capturing stochasticity in systems but has mostly confined itself to problem areas of low level of aggregation. System dynamics has shown great promise in depicting causal relations in data-deficient systems but has mostly confined itself to problems of high level of aggregation.

4. Structural Equivalence between Markov and System Dynamics Models.

We have shown that the instantaneous rate of change of j^{th} state probability is given by

$$\frac{d}{dt}\O_j(t) = \sum \O_i(t)\,\lambda_{ij}$$

This can be written as

$$\frac{d}{dt}\O_j(t) = \O_j(t)\,\lambda_{jj} + \sum_{i\neq j} \O_i(t)\,\lambda_{ij}$$

or

$$\frac{d}{dt}\O_j(t) = \O_j(t)\left[-\sum_{k\neq j}\lambda_{jk}\right] + \sum_{i\neq j} \O_i(t)\,\lambda_{ij}$$

or

$$\frac{d}{dt}\O_j(t) = \sum_{i\neq j} \O_i(t)\,\lambda_{ij} - \sum_{k\neq j} \O_j(t)\,\lambda_{jk} \qquad [10]$$

One immediately recognizes that this is a level equation with $\O_j(t)$ as the level variable,

$\sum_{i \neq j} \emptyset_i(t)\, \lambda_{ij}$ is the total inflow into the level and $\sum_{j \neq k} \emptyset_j(t)\, \lambda_{jk}$ is the total flow out of the level. The inflow increases the probability $\emptyset_j(t)$ due to transitions to state j while outflow reduces $\emptyset_j(t)$ due to transitions out of j. The transfer rates λ_{ij} and λ_{jk} are the constants associated with the input and the output rates.

One observes that rates are linearly dependent on level variables from which they emerge. Thus stationary continuous time Markov models are algebraically equivalent to linear system dynamics models. In particular, the following comments can be made :

(i) State probabilities are nonnegative, a property which a state variable in any realistic system dynamics model must satisfy.

(ii) The state probabilities add up to 1. Thus they obey the principle of conservation as demanded by the physical flows in system dynamics models.

(iii) The initial state probabilities correspond to initial values of the state variables in the equivalent system dynamics model.

(iv) The transfer rates constitute the parameters of the system dynamics model.

(v) The outflow rates in the equivalent system dynamics model are functions of the state variables from which they originate. Thus they constitute simple first-order negative feedback loops. But these loops interact with each other and, in general, can be embedded in large positive feedback loops formed by the levels and rates which are related according to the transfer rate matrix. However, depending on particular situations, there may not be any positive feedback loop in a model.

(vi) Since stationary Markov models are analogous to autonomous linear system models, closed form solutions are possible for these models. Realistic systems may not satisfy the stationarity assumption even if Markovian assumption might hold. In non-stationary Markov models, transfer rates are time-varying. Markov models with time-varying transfer coefficients are still linear. Closed form solutions are possible, though difficult. It is argued in this paper that the stationarity assumption of Markov processes is quite restrictive. If one is ready to sacrifice the closed form solution for the sake of capturing more realism, then one may resort to the numerical solution procedure by converting the model into an equivalent system dynamics model.

(vii) Physical interpretation of state probabilities in system dynamics can sometimes be difficult. Since the state probabilities add up to unity, the individual state probabilities can be interpreted as fractions of total number of units in the particular physical flow system. For example, each level will represent the fraction of the total population in each grade in a manpower flow system, or a fraction of total number of machines in a particular category in a flow system representing machine conditions, etc.

(viii) The transfer rates are to be interpreted as the fraction per unit time that will flow out of the levels. For example, they will represent the fraction of population (manpower or machine, etc.) per unit time that will get transferred to the next level. In this context, it may be emphasized that often estimating transfer rates in the context of a Markov model may be difficult, whereas, when converted into its SD equivalent, their inverses are time constants, which are less difficult to estimate.

(ix) Mathematical formulation for transient solutions of Markov models tends to be complicated. Quite often, a Markov model deemphasizes transient solution and heavily focuses on steady state solution. A decision based on such steady state relationship may not be implementable in real life because the transients may not be acceptable. System dynamics models, on the other hand, give almost equal importance to both transient and steady state behaviours.

(x) System dynamics models are very transparent because the variables and the parameters have real life meaning. Therefore, SD equivalents of Markov models allow greater realism to be incorporated during the analysis by way of introduction of various realistic structural and policy changes.

5. Steady State Probabilities

Steady state solution to a stationary Markov model is very easily comprehensible in an equivalent system dynamics model. We know that in the steady state condition, the net flow into a level must be zero and the value of the level variables will become constant. This means that

$$\frac{d}{dt} \emptyset_j(t) = 0$$

and $$\emptyset_j(t) = \emptyset_j$$

So, from Eqn. 10, we must have

$$\sum_{i \neq j} \varnothing_i \lambda_{ij} = \sum_{k \neq j} \varnothing_j \lambda_{jk}$$

Using Eqn. 4, Eqn. 10 can be written as

$$\sum_{i \neq j} \varnothing_i \lambda_{ij} = -\varnothing_j \lambda_{jj}$$

The steady-state (or limiting) probability $\varnothing_j$ can be obtained as the following :

$$\varnothing_j = \frac{\sum_{i \neq j} \varnothing_i \lambda_{ij}}{\lambda_{jj}} \quad [11]$$

6. Converting Markov Models into System Dynamics Models

Equation (10) can be used to construct system dynamics flow diagrams for any Markov model. One notices in equation (10) that the terms λ_{ii} and λ_{jj} are not to be considered in the system dynamics model formulation implying that the virtual transition of a state into itself should be totally disregarded during the construction of an equivalent system dynamics model.

In the following sections we shall construct the equivalent system dynamics models for the birth and death process and its various ramifications. Both stationary and non-stationary Markov models will be considered.

6.1 The Birth and Death Process

In a birth-death process all transitions occur to the next state immediately above (a 'birth') or immediately below (a 'death') in the natural integer ordering of states. The transition diagram for such a process is given in Fig. 1. The system dynamics flow diagram for such a process is given in Fig. 2. The state equations for such a birth-death process are immediately computed as follows:

$$\frac{d}{dt}\varnothing_0(t) = \varnothing_1(t)\lambda_{10} - \varnothing_0\lambda_{01}$$

$$\frac{d}{dt}\varnothing_j(t) = \varnothing_{j-1}(t)\lambda_{j-1,j} + \varnothing_{j+1}(t)\lambda_{j+1,j} - \varnothing_j(t)\left[\lambda_{j,j-1} + \lambda_{j,j+1}\right]$$

In the steady state the first differentials will go to zero. Thus the steady state probabilities are given by

$$0 = \emptyset_1\lambda_{10} - \emptyset_0\lambda_{01}$$

$$0 = \emptyset_{j-1}\lambda_{j-1,j} + \emptyset_{j+1}\lambda_{j+1,j} - \emptyset_j\left[\lambda_{j,j-1} + \lambda_{j,j+1}\right]$$

Simulation of the system dynamics models of birth-death processes poses a particular difficulty. The diffulty is that one does not know in advance the numebr of states such a process will have nor is it possible to carry out a simulation with infinite number of states. In the following special applications of the birth-death process, we have assumed large numebr of states in the system to emulate the infinite state situations.

6.2 The M/M/1 Queuing System

The M/M/1 queuing system (Ravindran 1986) is a special case of birth-death process where

$$\lambda_{j,j+1} = \lambda \qquad \text{for } j = 0,1,2,....$$

and $$\lambda_{j,j-1} = \mu \qquad \text{for } j = 1,2,3.....$$

Thus the steady state equations for this system are

$$0 = \emptyset_1\mu - \emptyset_0\lambda$$

$$0 = \emptyset_{j-1}\lambda + \emptyset_{j+1}\mu - \emptyset_j\left[\mu + \lambda\right] \qquad \text{for } j = 1,2...$$

These, together with the normalizing equations

$$\sum_i \emptyset = 1$$

help to solve for the steady state probabilities which are well-known in queuing systems. To overcome the problem arising out of infinite system capacity (leading to infinite number of states), we have taken a large value of system capacity (equal to 20) while simulating the model. We have, however, also experimented with other values of system capacity, i.e. 5, 10, 15. In all these cases, in effect an M/M/1/N was simulated. In each case different values of λ/μ were taken to see its effect on the steady state value. The value of μ in all the cases was 8 but the magnitude of λ was varied (4,8 and 12). The solution interval in each case was taken as equal to 0.01. The initial value of states were 1,0,0,0......,0. The simulated results are given in Table 1. Table 1 also compares the simulated values of $\emptyset_0$ with the theoretically calculated values for M/M/1/N using closed form equations available in standard literature (for example Ravindran et al. 1987).

Table 1 : Simulation Results for M/M/1 Queuing System

λ	System Capacity	$\emptyset_0$ Theoretical	$\emptyset_0$ Simulated
4	5	0.51613	0.50794
	10	0.50049	0.50024
	15	0.50002	0.50001
	20	0.50000	0.50000
8	5	Zero	0.16667
	10	-do-	0.09091
	15	-do-	0.06250
	20	-do-	0.04762
12	5	-do-	0.04812
	10	-do-	0.00585
	15	-do-	0.00076
	20	-do-	0.00010

As expected the closest match with theoretical values are acheived when the queue capacity was 20, the highest. Effect of λ/μ was also as per expectation. For $\lambda/\mu < 1$ the steady state value for $\emptyset_0$ was less than 1 indicating a partial utilization of service. Whereas $\emptyset_0$ in steady state approached zero as λ/μ approached 1 indicating increase in utilization. For $\lambda/\mu > 1$ $\emptyset_0$ is zero in the steady state indicating a busy queue. The plot of $\emptyset_0$, average number of people in the system and average number of people in the queue, against transition for the three values of λ are given in Figures 3, 4 and 5 respectively. In Fig. 4 the value for *number of persons in the system* comes to a steady state when λ/μ is less than 1. But in the other two cases where λ/μ is equal to or greater than 1 the value for *number of persons in the system* does not attain a steady state value indicating an ever growing queue. Identical behaviour is demonstrated in Fig. 5 where the value for *number of person in the queue* is plotted. All these results conform to the results available in literature on queues.

6.3 The Poisson Process

The Poisson Process is a special case of Birth-Death process where the death rates are all zero and the birth rates $\lambda_{j,j+1}$ are all equal to a constant value λ. The transition diagram for such a process is given in Fig. 6. The equivalent system dynamics flow diagram is given in Fig. 7. From the flow diagram the state equations are given as

$$\frac{d}{dt}\Phi_0(t) = \Phi_0 \lambda$$

$$\frac{d}{dt}\Phi_j(t) = (\Phi_{j-1} - \Phi_j)\lambda \qquad \text{for } j = 1,2,\ldots$$

Solving the first equation one obtains

$$\Phi_0(t) = e^{\lambda t}$$

Using this solution for $\Phi_0(t)$ in the second equation recursively, one can show that the state probabilities follow a Poisson distribution. The outflow from each level variable is given by

$$R_j(t) = \lambda \, \Phi_j(t) \qquad \text{for } j = 0,1,2....$$

Thus $\mathbf{R_0(t) = e^{\lambda t}}$ and follows a negative exponential distribution. The flow diagram looks like an infinite-order exponential delay consisting of infinite number of cascaded first order exponential delays. Thus the outflow from the Rth state variable can be looked upon as the sum of K-1 number of random variables each following a negative exponential distribution. Therefore the outflow $\mathbf{R_k}$ **follows an erlang distribution of order k+1.**

The steady state transition probabilities can be equal to one another :

$$Ø_0 = Ø_1 = Ø_2 = ...$$

Thus if the Poisson process is finite (maximum population size N), then each of these steady state probabilities must be equal to $(N)^{-1}$ since all of them sum to unity. The system dynamics model for Poisson process was simulated with avalue of λ equal to 4. DT was taken as 0.001 and number of states were 15. The stady state proabilities are as given in Table 2.

Table 2 : Simulation Results for Poisson Process

Ø[0]	Ø[1]	Ø[2]	Ø[3]	Ø[4]	Ø[5]	Ø[6]	Ø[7]	Ø[8]
0.000	0.000	0.000	0.000	0.002	0.004	0.010	0.019	0.034

Ø[9]	Ø[10]	Ø[11]	Ø[12]	Ø[13]	Ø[14]	Ø[15]
0.051	0.071	0.089	0.102	0.108	0.106	0.403

Theoretically steady state values for all level variables, excepting for Ø, are supposed to be equal to zero. The steady state value for a few initial variables are indeed zero. But for some others the value is close to zero but not exactly equal to it. The steady state values for all such variables would have been equal to zero if a large number of runs were taken.

6.4 The Yule Process

The Yule process is a very special case of pure birth process where $\lambda_{j,j+1}$ depends on state j in the following manner (Feller 1966)

$$\lambda_{j,j+1} = j\lambda \qquad j \geq 1$$

State j indicates the population size. Therefore, j=0 will not be a state in the Yule process and $\lambda_{0,1}$ = 0. The equivalent SD flow diagram will not contain $\emptyset_0$ or R_0. The flow diagram is given in Fig. 8. Like the previous case, a large value of states (15) was taken. From Fig.6 it is obvious that the fifteenth state is a trapping state and that in the steady state probability system in the trapping state is 1 and probability of system in any other state is zero.

The model was simulated using a solution interval of 0.001. The initial values were taken as 1,0,0.....,0. The value of λ was 10. The simulation results are given in Table 2.

Table 3 : Simulation Results for Yule Process

Ø[1]	Ø[2]	Ø[3]	Ø[4]	Ø[5]	Ø[6]	Ø[7]	Ø[8]	Ø[9]
0.010	0.010	0.010	0.009	0.009	0.009	0.009	0.009	0.009
Ø[10]	Ø[11]	Ø[12]	Ø[13]	Ø[14]	Ø[15]			
0.009	0.009	0.009	0.009	0.009	0.862			

Theoretically, in the steady state, the value for all level variables excepting the trapping state should be zero. However the simulation this is not the case. This may be attributed to the the following factor. The change in the value of any level during a solution interval depends on level variable itself. As the level variable approaches steady state, its value becomes almost equal to zero. The change in its value at that point becomes so small that it appears that the level has reached a steady state. However, if one takes the simulation run over a large time period, it is certain that the value of the level variables will eventually come to zero value.

7. Final Remarks

This paper shows that Markov processes are structurally equivalent to system dynamics models. This opens up the possibility of applying system dynamics to problems in stochastic systems. Queueing and Reliability systems seem to be two ready candidates for such applications. We also expect that susch equivalence will help model builders to take note and make use of the wealth of analytical results available in the field of Markov modelling. The authors are working in these areas and hope to bring the results to light once they are ready.

8. Referrences

[Feller 1966] W. Feller, *An Introduction to Probability Theory and Its Applications*, Vol I, John Wiley & Sons, New York, 1966

[Forrester 1972] J. W. Forrester, *Principles of Systems*, Wright Allen Press, 1972.

[Mohapatra 1976] P. K. J. Mohapatra, "Some Generalized Results on Exponential Delays" in *Transaction of IMACS*, Vol. XVIII, No. 4, October 1976.

[Mohapatra 1980] P. K. J. Mohapatra, "Nonlinearity in System Dynamics Models" in *Dynamica*, Vol. 6 Part I, Summer 1980

[Mossekilde 1983] E. Mosekilde E & S. Rasmussen, "Random Processes in System Dynamics" in *Dynamica*, Vol. 9 Part I, Summer 1983

[Ravindran 1986] A. Ravindran, D. T. Phillips, J. J. Solberg, *Operations Research - Principles and Practices*, John Wiley & Sons, Second Ed. 1986, pp. 278-285

[Sahin 1979a] K. E. Sahin, "System Dynamics models : Some Obscurities" in *IEEE Transaction on System, Man and Cybernatics*, Vol. SMC-9, No. 2, Feb. 1979.

[Sahin 1979b] K. E. Sahin, "Equivalence of Markov Models to A Class of System Dynamics Models" in *IEEE Transaction on System, Man and Cybernatics*, Vol. SMC-9, No. 7, July 1979, pp 398-402.

[Wood 1983] W. T. Wood, "Modelling Stochastic Processes with System Dynamics" in *Proceeding of International System Dynamics Conference* 1983,Boston, Massachusetts, Vol. I, pp 369-376.

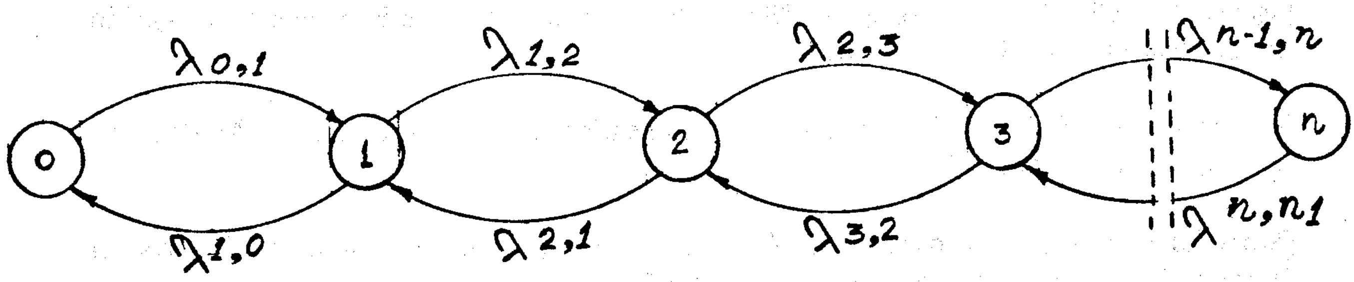

Fig. 1. : Transition Diagram for the Birth-Death Process

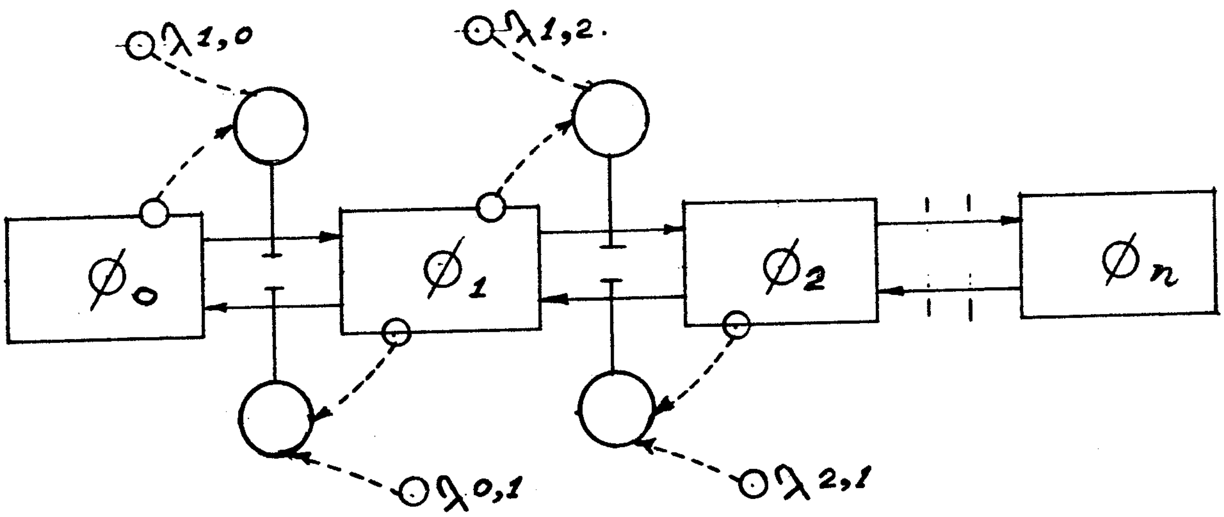

Fig. 2. : Flow Diagram for the Birth-Death Process

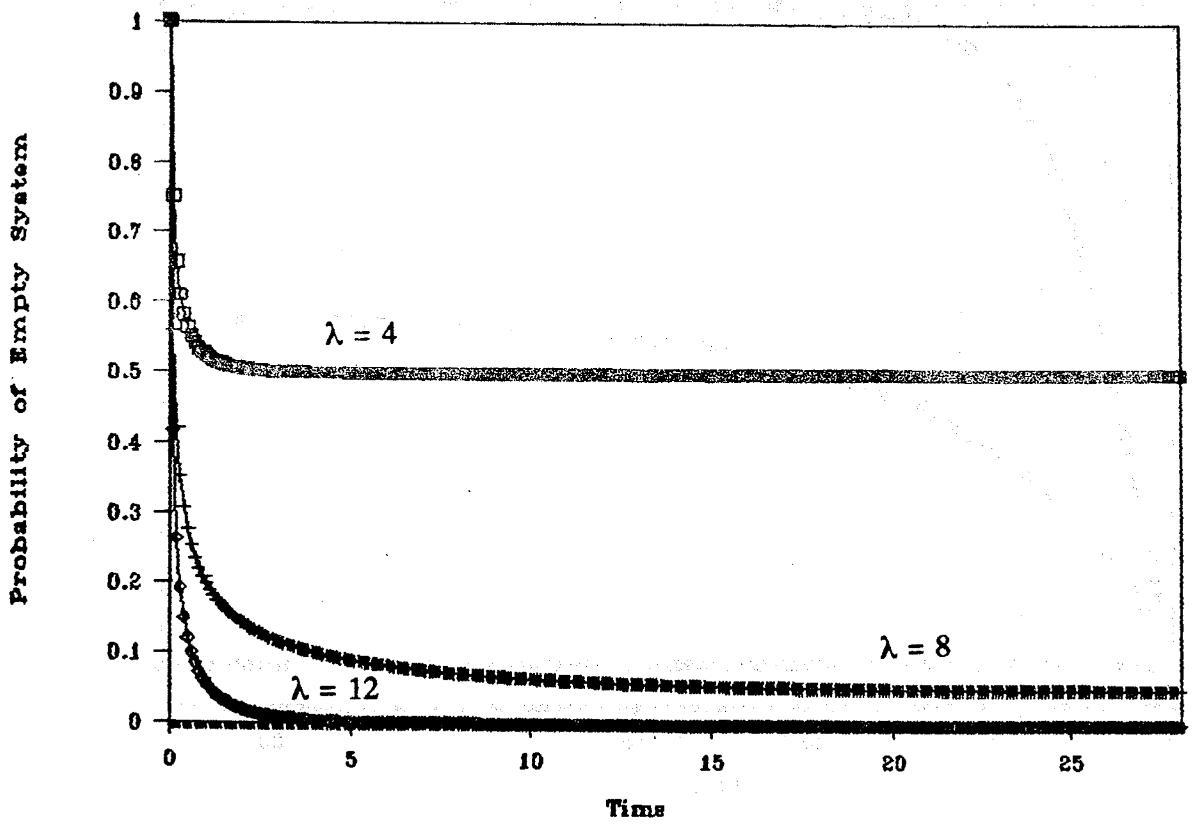

Fig. 3. : Variation of Probability of the System (M/M/1 Queue) being Empty

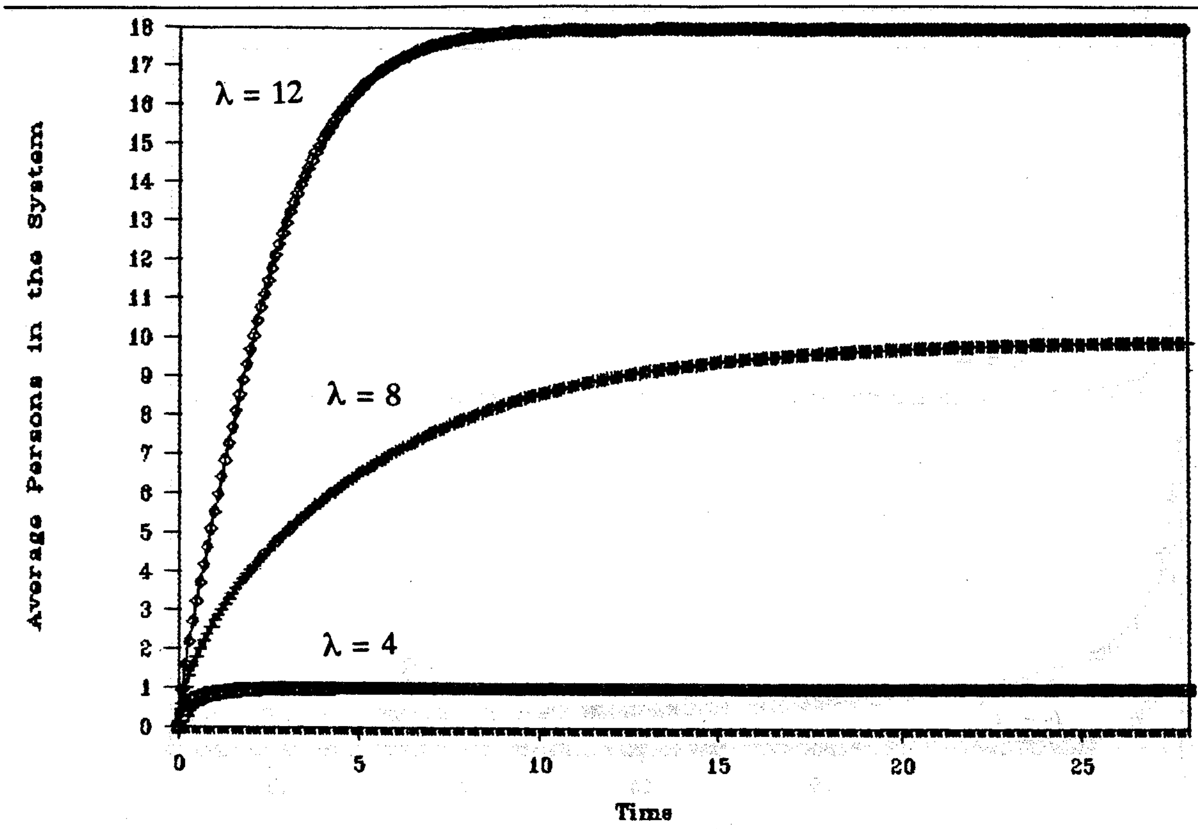

Fig. 4. : Variation of People in the System

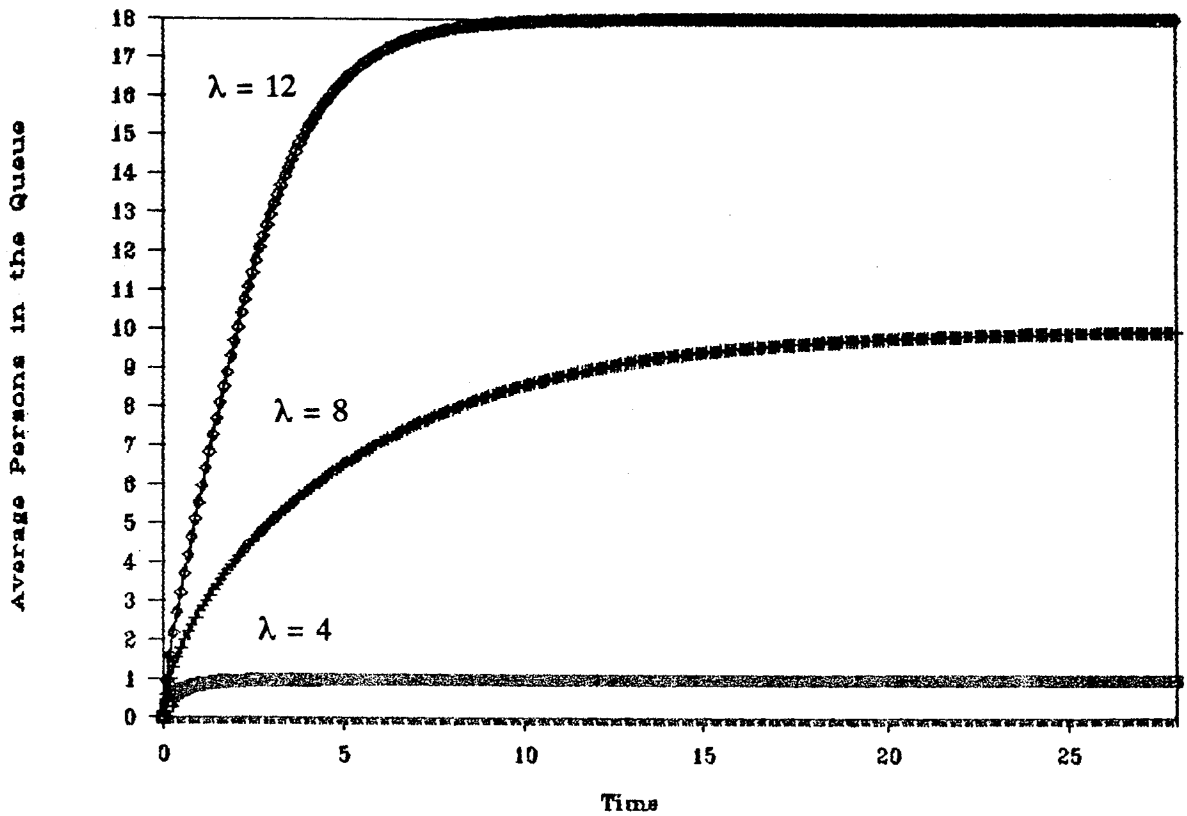

Fig. 5. : Variation of People in the Queue

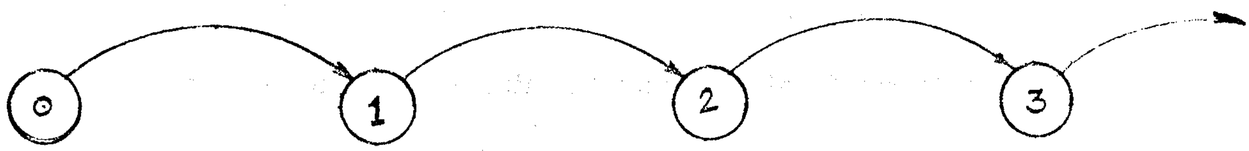

Fig. 6. : Transition Diagram for the Poisson Process

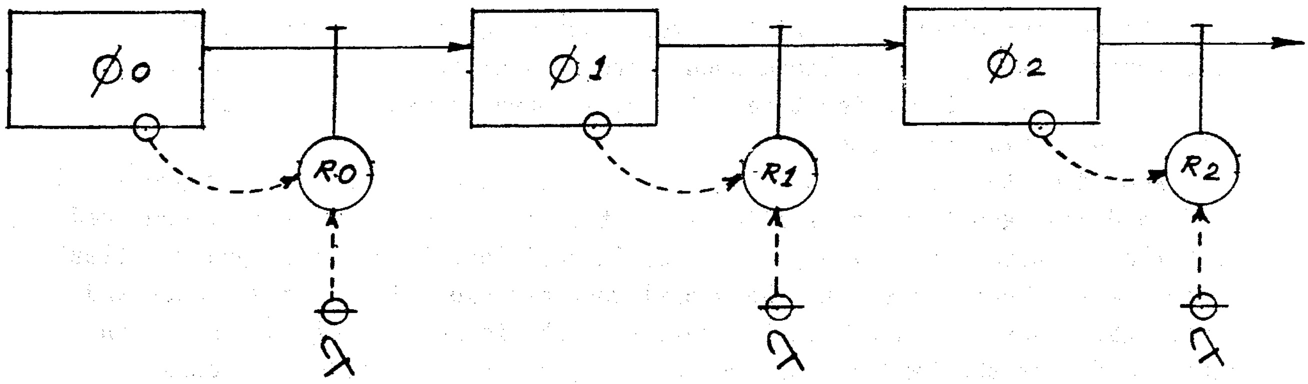

Fig. 7. : Flow Diagram for the Poisson Process

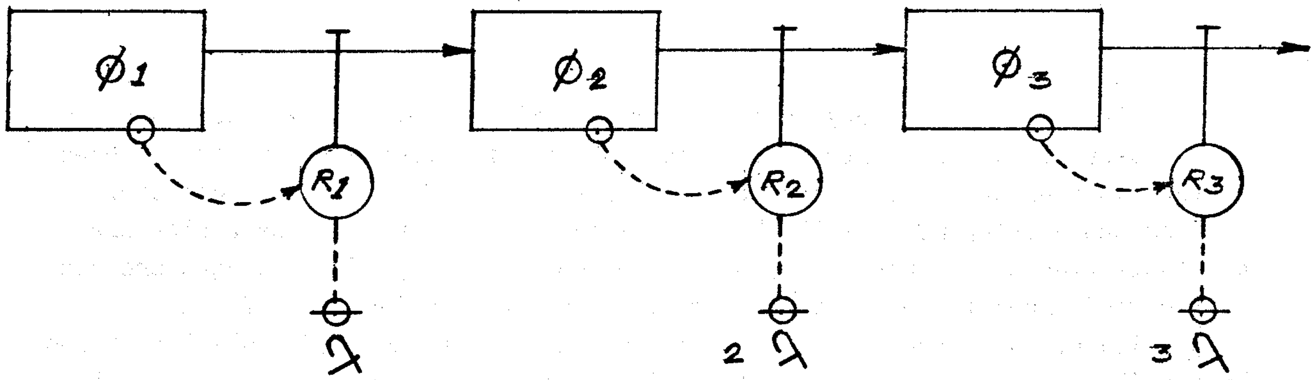

Fig. 8. : Flow Diagram for the Yule Process