

実時間処理システム－MARS 101 のシミュレーション*

大 須 賀 節 雄**

1. ま え が き

近來計算機技術の進歩に伴って、多くの分野で実時間処理システムが採用されてきている。実時間処理システムには、待時式処理システムにはない多くの固有の問題が含まれるため、それらの問題点を明らかにし、かつ解決してよりよいシステムを作るための努力がなされねばならない。本稿は実時間処理システムについてできるだけ一般的な立場から研究を行なうための一環として、日本国有鉄道の列車座席予約システム MARS 101 を対象にしたシミュレーションの結果である。

このシステムは、日本全国主要駅に散在する端局装置から中央処理システムに集まってくる多量の呼を実時間で処理する。また、このシステムの特徴の一つは、中央制御装置の制御のもとで、いくつかの独立した外部制御処理装置が、各呼について一連の処理手順のうちの異なった部分を受けもっており、これによってシステムとしての効率を増し、処理能力を高めていることである。

システム各部のマッチングを行なうことはどのような実時間処理システムでも必要なことであるが、本システムではこのような特徴のため、特にシステムのバランスを保つことが重視されねばならない。

本シミュレーションはこのような問題を含めて、システムの一般的な傾向を把握するために、パラメータをいろいろに変えて試行を行なった。本稿ではそのうち、手法の説明までを行ない、結果については追って報告する。なおシミュレーションには MARS 101 の中央制御装置 HITAC 3030 を用いた。

2. MARS 101 の概要

シミュレーションの対象である MARS 101 座席予約システムについては、これまでにいろいろな報告や

発表がなされているから、詳細な説明は省き、シミュレーションを行なう実時間処理関係の概略説明のみを行なうこととする。

2.1. システムの構成 (実時間処理関係)

MARS 101 システムのうち実時間処理に関するものは、中央制御装置 (SC)、テーブル計算機 (TC)、ファイル計算機 (FC)、送受信制御装置 (TRC) および磁気テープ装置 (MTC) である。ただし、このうち MTC は入出力情報が書き込まれるのみで、その処理は実時間処理と無関係であるから、実時間処理部分から除いて考えてもさしつかえない。それより問題となるのは2重処理システムによる影響である。すなわちこのシステムでは誤まりの発見を確実にし、かつ事故の際にも処理が中断されることを防ぐために、2重システムが用いられ、両システムで同一の呼を独立に処理し、結果の照合を行なっているからである。

そこで、シミュレーションに際して、この影響を考慮している。SC、TRC、TC および FC の機能は次のとおりである。

(i) SC-システムの中心となる制御装置であり、各種制御、演算機能を備えている。また TRC、TC、FC その他の外部制御装置からの自動割り込み機能を有し、これら各割り込みに対応して独立のプログラムが一つずつ準備されているマルチ・プログラミング方式である。各割り込みには優先順位がつけられ、この順位に応じてハード・ウェアによる自動割り込み選択がなされる。反面これら割り込みの成立あるいは禁止はプログラムによって制御することができ、これを含めて、SC はシステム内の呼の分布を全体的に制御する。また SC 自身でも呼の各種の処理 (料金計算、編集、コード変換その他) を行なう。外部諸装置との情報交換は、データ・チャンネルを通じて、メモリーシェアーにより演算部・制御部とは独立にかつ相互に並列に行なわれる。

(ii) TRC-多数入出力回線の並列制御を行なう。SCのデータ・チャンネルに接続され、電信回線を通じて送られてくる呼を1ビットずつ (50ボーの割合で)

* Simulation of Real-Time Processing System -MARS 101 by Setsuo Osuga (Institute of Space and Aeronautical Science, University of Tokyo)

** 東京大学宇宙航空研究所

SC 内の主記憶装置に直接書き込む。このような電信回線を最大 96 回線まで同時に制御する。

(iii) TC—このシステムが取り扱う列車に関する固有の情報(停車駅名, 走行区間その他)を記憶しており, SC の指令によってこれら情報を索引する。

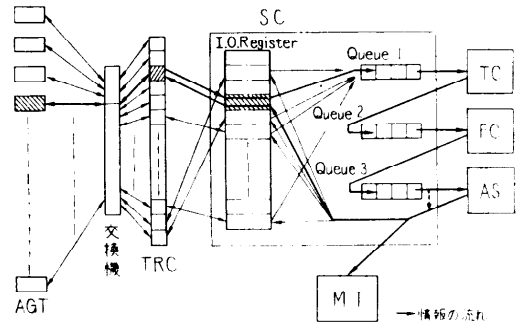
(iv) FC—全列車の予約期間中および各走行区間ごとの全座席が記憶されており, SC の指令によって必要な座席を見出し, また内容を修正する。

以上の各装置は, SC からの指令を受け取ると以後は各装置が有している制御機構によって独立に処理を行ない, 結果を SC の主記憶装置に書き込み, 割り込みによって SC に処理終了を知らせる。SC は割り込みが生ずると, それに対応したプログラムを実行して新しく指令を与える。

2.2. システム内の呼の移動

端局装置 (AGT) で発生した入力呼は, 上記各装置を一定の順序で通りながら処理が行なわれる。各装置の配置と, 呼の流れの順序を示したのが第 1 図である。前述のように, このシステムは 2 重系になっているが, 図では一方のみを示している。ただし 2 重系による影響を含めるために, もう一方のシステムを, TC や FC と同じような外部装置の一つのごとくにみなして, 図中 AS と示している。

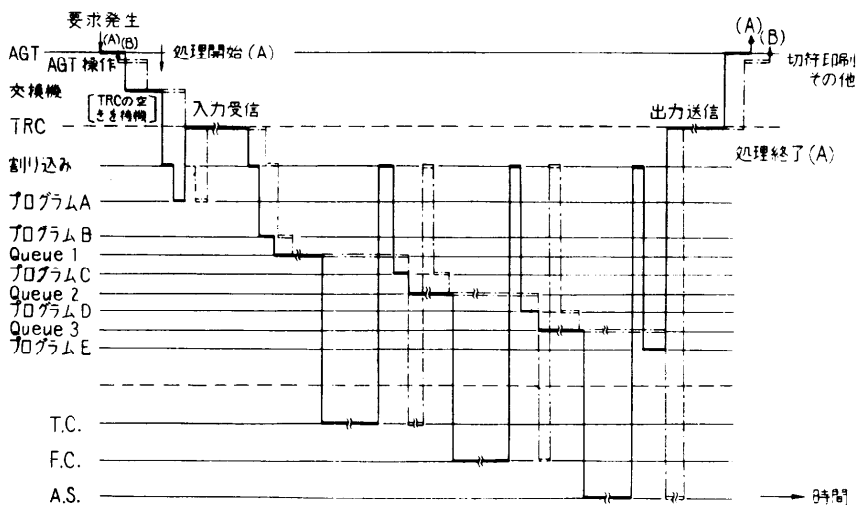
矢印で示されたものが呼の流れである。AGT からの入力呼は, 途中で中断点を経て交換機にはいり, 次いで TRC を介して 1 ビットずつ SC 内の主記憶装置に書き込まれる。この書き込みのアドレスは, SC



第 1 図 システムの構成と呼の流れ図

内でプログラムによって空いている場所が捜され, TRC に指示される。そのため, 入力呼に先だって要求信号が出され, これが TRC を介して SC に割り込み, SC 内でそれに対応するプログラム (Pr. A) が実行される。TRC は同時に n 回線の制御ができるものとし, 以後この数 n を TRC 送受信チャンネル数あるいは単に TRC チャンネル数と呼ぶことにする。TRC 送受信チャンネルが全部ふさがっている場合, 呼は交換機内蓄積装置に待たされる。通信は 50 ボードで行なわれるから, 入力呼のビット数を m_i とすると, 入力時間 $T_i = m_i/50$ 秒が必要である。入力呼のビット数は呼によって多少変化するが平均 125 ビット程度であり, したがって $T_i \approx 2.5 \text{ sec}$ である。

入力受信が終了すると, TRC は再び SC に割り込み, それに対応するプログラム (Pr. B) が実行されると, この呼が取り出されて SC 内でまず一連の処



第 2 図 処理シーケンス

理をうけ、次いで行列1 (Q1) に並べられる。この行列は TC 用のもので、この中の呼は先頭より一つずつ取り出されて TC で処理される。TC は一つの呼の処理を終了すると割り込みをかける。これに応じて SC 内でプログラム (Pr. C) が行なわれ、TC で処理された呼を SC 内で再処理し、行列2 (Q2) に並べ、また、Q1 内の次の呼を TC に送り込み TC を起動させる。Q2 は FC 用の行列で、以下同様にして FC で処理された呼は SC 内のプログラム (Pr. D) で処理されて出力呼が形成され、これが次の行列 (Q3) に並べられる。この行列は他システムとの照合をとるための行列で、両システムで照合すべき呼がそろうと、プログラム (Pr. E) で照合が行なわれ、この後、呼は再び TRC を介して入力の場合と同様に1ビットずつ送信される。なお照合のとられた呼は、入力呼とともに磁気テープに書き込まれる。出力呼のビット数を m_0 とすると、出力時間 $T_0 = m_0/50$ 秒が必要である。 m_0 は平均 100 ビット、したがって $T_0 \approx 2 \text{ sec}$ である。

以上の一連のシーケンスを示したのが第2図である。この図は横軸に時間を取り、呼が順次レベルを変えながら処理されていく模様を示している。図には A および B と記した二つの呼が示されているが、実際にはもっと多くの呼が存在し、それによって SC, TC, FC などが並列の処理を行ない、システムの処理能力が増すことになる。

3. シミュレーションの準備

3.1. モデル作成上の一般的事項

シミュレーションを行なう際、システムのモデル化には細心の注意が必要である。作成されたモデルがシステムを忠実に再現し得るものであると同時に、実際のシステムより取り扱いが容易で、シミュレーションの所要時間も短いものであることが望ましい。シミュレーションは、事象のパターンを記録あるいは観察することであり、個々の事象の意味・内容は問題にならないから、状態の時間的移り変わりだけが実際のシステムに合致するように定式化が行なわれる。

以後の記述において、送受信中あるいは待ち行列中の呼の数などのように、システムの状態を示す諸量を状態量と呼び、これを変化させる要因を事象と呼ぶ。

第2図に示されているように、このシステムは入力の発生を含めて各レベルで各種の装置が並列な動作を行なっている。モデル内でこのような諸装置に対応するものを Item と呼ぶことにする。これら各 Item は

事象を発生するが、計算機によってシミュレーションを行なうためには、これら事象を時間的に直列なものに変換せねばならない。しかし、これら各装置をすべて一様に取り扱うのは得策でないので、そのいくつかを Sub-Item とし、これらが集まって一つの Item を形成するように局部的な直列化が行なわれている。この結果、システム全体を表わすモデルは、総計8種の独立な Item にまとめられ、これらがさらに直列化されている。各 Item について、事象発生間の時間間隔がランダムなものは、モデル内で各分布に従う乱数が発生される。

モデル内では、状態量はすべて待ち行列内の呼の数として表わされ、これら待ち行列は“First-in, First-out”の方式に従う。これら各 Item ごとの状態量はモデル内で仮想的に進められる時間とともに変化し、この状況を記録することによってシミュレーションが実行される。状態の変化は、Item ごとの固有の規則に従って、シミュレーション制御プログラムにより行なわれる。なお、シミュレーションの対象には、個々の AGT の状態までは含めない。AGT は全体として入力発生源として扱われている。

3.2. システム各部の局部的なモデル化

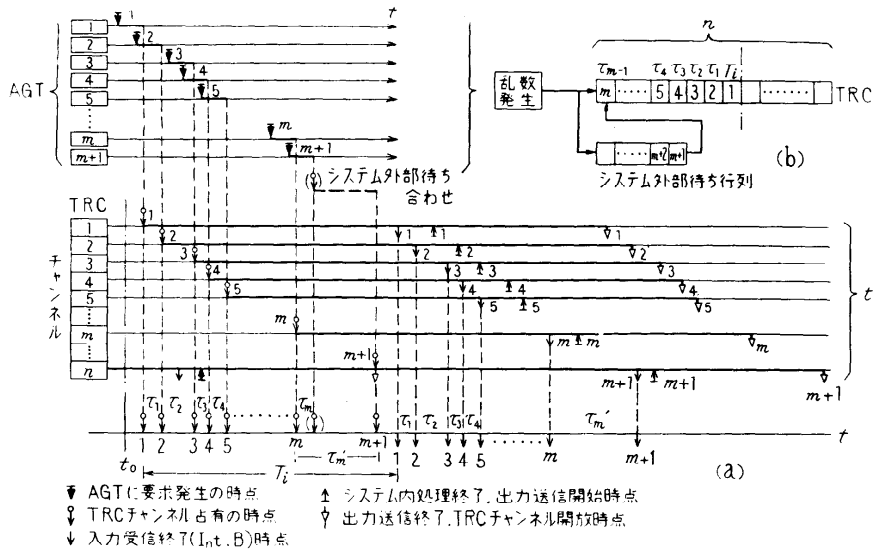
システム全体のモデルを構成する前に、システム各部の局部的なモデル化が行なわれる。モデルを、システムとの対応から、外部モデル (入力発生)、中間部モデル (送受信部)、内部モデル (処理部) に分ける。

3.2.1. 外部モデル

外部モデルは入力発生機構のモデルである。個々の AGT の状態は含めていないから、AGT 全体から中央処理システムに集まってくる入力呼の割合のみを問題とすれば良い。独立な AGT が十分多数存在するとき、システム入口での入力呼の割合は、ポアソン分布に従うことが良く知られている。したがって、モデルでは全 AGT を一つの乱数発生器でおきかえている。この乱数は、呼間の時間間隔が指数分布に従う。

3.2.2. 中間部モデル

中間部モデルは送受信を制御する TRC のモデルである。入力呼は TRC のチャンネルに空いているものがあれば、それを順次占有してゆく。空きがない場合には交換機内蓄積装置に待ち合わせが生ずる。また場合によっては回線がふさがっているために、AGT の前に客の待ち合わせが生ずることもあり、このような潜在的な待ち合わせを含めて、中央処理装置内にはいり得ずに待ち合わせている呼を、システム外部



第3図 入力呼と TRC の関係

待ち合わせと呼ぶことにする。占有された TRC の各チャンネルは入力を受信、システム内部の処理、処理結果の送信という一連の段階を経てはじめて開放される。

この情況の一例を第3図に示す。TRC は n チャンネルを有しているが、時刻 t_0 で m チャンネルが空いているとする。占有されているチャンネルが開放される前に m 個以上の呼が生ずると、最初の m 個は空いているチャンネルを占有するが $m+1$ 番目以後はシステム外部待ち合わせとなる。図ではその後 n 番目のチャンネルが空いて、そこに $m+1$ 番の呼が入る場合が示されている。図の下端に示されているように、各呼が TRC チャンネルを占有した時点および入力を受信が終了して処理要求の割り込み (Int. B) となった時点を、1本の時間軸上に記してみると、割り込み (Int. B) の時点での各呼の時間間隔は、それが TRC を占有した時点での時間間隔に等しい。そこで、モデルでは TRC を1本の待ち行列でおきかえる。呼が TRC チャンネルを占有した順序に、モデル内では呼が行列に並ぶものとする。行列内の呼は行列の先頭に特定の時間だけ止まったのち行列を離れる。この時間は TRC を占有するときの前の呼との時間間隔である。ただし、この時間間隔が T_i より大なときは T_i である。これは、単一窓口の前に行列があり、そのサービス時間が上述のように決められるとした場合と等価で

あるから、以後 TRC を1本の行列と一つの窓口（あるいは処理装置）があるかのごとく考える。

TRC チャンネルに空きがない場合は溢れ呼となってシステム外部待ち合わせ行列に並ぶので、このモデルの例が第3図 (b) に示されている。実際のモデルは、システム・モデルを形成する他の Item と同じ形にするため（行列の前で分岐点が生じないように）さらに修正が行なわれている。すなわち入力呼はすべてシステム外部待ち行列にはいり、TRC チャンネルに空きがある場合には、経過時間 0 で入力受信の行列に移る。これによって、行列が直列に連なることになる。

以上のことは出力送信の場合にもあてはまるが、この場合には溢れ呼は生じないので、システム外部待ち合わせ行列に相当するものは不要である。

入力受信が終了して割り込み (Int. B) となる呼は処理されるために直ちにシステム内部にはいり得るとはかぎらない。システム内部にすでに呼がいっぱいにはいっていると、受信終了の状態のまま待つことになり、ここに処理待ちの行列ができる。全行列を直列のものとし、行列の前での分岐をなくすために、入力受信中の行列を出た呼は、1度無条件にこの処理待ちの行列にはいるものとする。したがって実際に割り込み (Int. B) を生ずるのは、この行列に呼がはいっている場合のみである。

このようにして、TRC の n チャンネルは、それを占有している呼の状態によって、入力受信 (n_i)、処理待ち (n_w)、処理中 (n_p)、出力送信中 (n_o) および空き (n_e) に分けられる。いうまでもなく、 $n_i + n_w + n_p + n_o + n_e = n$ であり、 $n_e = 0$ の場合に新しい呼があるとシステム 外部待ち 合わせが実際に 生ずることになる。

3.2.3 内部モデル

内部モデルはシステム内部の処理機能のモデルであり、呼の受信を終了して処理にはいってから、送信を開始するまでを受けもつ。ここでは呼は TC, FC および SC で処理され、他システムの結果と照合される。TC, FC での処理および他システムとの照合は独立な Item であり、固有の待ち行列を有している。呼は行列の先頭にくると、その位置で処理または照合されるものとする。これら処理時間または照合のための待ち時間（以後これを他システムによる処理のように考え一括して処理時間と呼ぶ場合がある）は、それぞれ固有の分布に従う乱数である。

なお SC による処理は各 Item から事象が生じたとき、それに引き続いて行なわれるので、独立な事象ではない。したがって SC の処理時間は各 Item の待ち時間としてモデルに含まれている。たとえば Item i で処理が終了（事象が発生）した後、SC でそれに対応するプログラムがみかけ上実行されて、次の呼が与えられるまでは Item i は次の処理に入れずに待たされる。プログラム実行時間は Item によって異なるが、同一の Item については常に一定とする。

3.3. システム全体のモデル構成

システム全体のモデルはシステム各部でモデル化された主要な Item から構成されている。これら各 Item のうち入力の発生以外はそれぞれ固有の待ち行列を有しており、いずれも行列内に呼が存在し、かつその先頭の呼が Item ごとに固有の方法で決められる時間だけ留まった後事象が発生する。そこで、入力の発生に関しても仮想的に待ち行列を与えることにする。この行列には常に呼が存在していて、他の場合のように呼の数が 0 となったために事象が発生しないという場合が生じないようにしておけば良い。このようにしてシステムを構成する Item または待ち行列は次の 8 種である。

- | | |
|-----------------|----------|
| (1) 入力発生 | (Item 1) |
| (2) システム外部待ち合わせ | (Item 2) |
| (3) 入力受信中 | (Item 3) |

- | | |
|---------------|----------|
| (4) 処理待ち合わせ | (Item 4) |
| (5) TC | (Item 5) |
| (6) FC | (Item 6) |
| (7) 他システムとの照合 | (Item 7) |
| (8) 出力送信中 | (Item 8) |

上記 8 種の待ち行列は、この順序で直列に連なっていて、それぞれ独立に事象が発生する。ただし、(3) ~ (7) の各 Item では、行列の許容の長さが有限であるため、これらのうちのいずれかまたはすべての行列がいっぱいになっている場合がある。このような場合には、いっぱいになっている行列の一つ前の Item では、たとえ行列の先頭の呼が所定の時間を経過しても事象は生じ得ない。また各 Item とも、行列内に呼が存在しなければ事象は生じないことはいうまでもない。実際の場合には、これは SC による割り込みの制御という形をとるが、モデル内でもこれに相当した制御を行なうため、各 Item ごとに 2 種類のマスク・ビットが用いられる。すなわち次の行列がいっぱいのとき用いられるもの (MA) および行列内に呼が存在しないときのもの (MB) である。

(2) および (4) の行列は溢れ呼によって作られ、この中の呼は次の行列に空きができればすぐ移り得るような性質のものであるが、これは呼のサービス時間が常に 0 であるとすれば良い。また一般には事象は SC への割り込みであり、SC 内でそれに対応するプログラムが実行されて後、呼が次の行列に移される。そのため Item ごとに固有のプログラム実行時間が必要である。SC がプログラム実行中に生じた他の事象は、プログラム終了まで待たされる。ただし、(1) から (2) および (3) から (4) への移行は自動的な状態の変化であり、SC を仲介としないが、これもプログラム実行時間が 0 であるとすれば、他の場合と同様に扱うことができる。

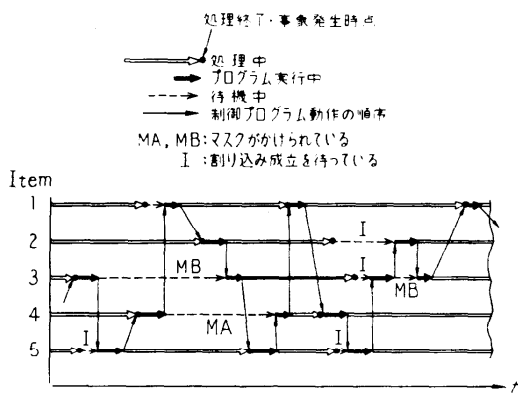
これら 8 種の Item は並列なものであるが、モデルでは事象の発生時点のみを問題にすれば良いのであるから、各事象は発生した順にとり出される。これは制御プログラムにより行なわれ、次に発生する事象はどれであるかが監視されている。モデル内での時間が経過すると、各 Item とも次に事象の発生するまでの残り時間がその分だけ減少する。そのために、この残り時間がリストにのせられており、これがサーチされて次に最初に生ずる事象が見出される。経過時間が差し引かれるためリスト内容は常に変化している。

以上のことを、簡単のため Item 数を仮りに 5 個と

した場合について第4図に示す。

この図で横軸は時間であり、水平線上の各種矢印は各 Item の状態を示し、これらを結んでいる細い実線の矢印は、事象を発生した順に取りあげてゆく制御プログラムの動作を示している。なお点線で示された待機時間は、システムの使用効率 (Availability) に関係するので、この時間が積算されて記録される。制御プログラムの動作は、具体的には前述のようにリスト・サーチであり、図の右側に示されているようなリスト内の残り時間をサーチして最初に 0 となるものを見出す。ただし、マスクのかけられている Item (マスク・ビットが 1 になっているもの) はサーチの対象からはずされる。

また、このシステムでは割り込みの優先順位が決めてられており、同時に二つ以上の割り込みが生じた場合



第4図 事象の発生と処理のシーケンス

第1表 各 Item に固有の性質

Item No.	システムとの対応	サーチ順位	MA	MB	S_i	τ_i'
1	入力呼の発生 (AGT)	3	0	0	0	R.N. (指数分布)
2	システム外部待ち合わせ (交換機)	8	0/1	0/1	S_2	0
3	入力受信 (TRC1)	1	0	0/1	0	$t_k - t_{k-1}$ or T_i
4	処理待ち合わせ (TR2)	7	0/1	0/1	S_4	0
5	TC	5	0/1	0/1	S_5	R.N. (TC 分布)
7	FC	4	0/1	0/1	S_6	R.N. (FC 分布)
7	他システムとの照合 (AS)	6	0	0/1	S_7	R.N. (指数分布)
8	出力送信中 (TRC3)	2	0	0/1	0	$t_k - t_{k-1}$ or T_0

あるいは SC がプログラム実行中に 2 種以上の割り込みが生じて待機しているような場合、優先順位の高いものから順に割り込みが成立する。これに対応してモデル内では残り時間のサーチ順序を優先順位と同じにきめ、そのサーチ順序がリスト内に記されている。ただし、(1), (3) および (8) のように自動的に状態の変化するものは、他のものより高位におかれる。リスト内にはこの他にも Item ごとの各種の性質を示す数値やパターンが記録されている。

各 Item と実際のシステム各部との対応および各 Item のいくつかの性質をまとめて第1表に示す。

マスク項で 0 となっているものは、マスク不要のものである。 τ_i' は各 Item ごとの呼の処理時間の性質を示す。R.N. と示されているものは、それぞれ固有の分布に従う乱数が用いられるもの、 $(t_k - t_{k-1})_i$ と示

されているのは、すでに述べたように入出力送受信の際、各行列の窓口の仮想的サービス時間が、行列にはいるときの呼間の時間間隔で決められることを示し、Item 2 および 7 の事象が生じたときに求められて表に記録されている。0 と記されているのは、常に $\tau_i' = 0$ を示す。このように求められる τ_i' は、Item i に関して引き続いて生ずる事象間の時間を規定するもので、Item i の事象が生じたときリスト内の残り時間の項に新しく書き込まれる。

残り時間	サーチ順位	マスク A (例)	マスク B (例)	実行時間
1	τ_1	4	0	S_1
2	τ_2	5	0	S_2
3	τ_3	2	0	S_3
4	τ_4	1	1	S_4
5	τ_5	3	0	S_5

3.4. モデル内の時間

シミュレーションの方法はモデル内の時間の進め方によって異なるものとなる。シミュレーションを実時間で行なう必要はないから、時間の単位のとおり方は任意であるが、時間の進め方には、(1) 一定間隔の微小時間を独立に進めてゆく方法と、(2) 時間間隔は不定で引き続いて生ずる事象間の時間を 1 単位とする方法がある。

(1) の方法では精度をあげるために微小時間の単位を非常に小さくせねばならないうえに、事象の生じないときも試行を繰り返さね

MA, MB: マスク。(0 は常にマスク不要のもの 0/1 はマスク使用のもの)

S_i : プログラム実行時間

τ_i : 行列の先頭にとどまる時間

R.N.: 新乱数が用いられる

0: 一常に 0

$t_k - t_{k-1}$: Item i を出てきた呼間の時間間隔

$S_2 = 1 \text{ ms}$ or 0

$S_4 = 3 \text{ ms}$ or 0

$S_5 = 3 \text{ ms}$ or 0

$S_6 = 4 \text{ ms}$ or 0

$S_7 = 3 \text{ ms}$ or 0

ばならないため時間がかかることになるので、本シミュレーションでは(2)の方法を用いている。具体的にはリスト内の時間をサーチして残り時間が最小のものを見出して、そのItemが次に事象を発生するものとして取り出され、同時にその事象発生時点まで時間を進めるため、リスト内の他の残り時間からその分だけ差し引かれる。

実際にはさらにシミュレーションの速度を速めるため、この見出された最小残り時間と、そのItemのプログラム実行時間(S_i)との和をとって、その分だけ一度に時間を進めている。ただし待機中であったItemの場合には、残り時間が負になっているが、このときは時間はプログラム実行時間だけ進められる。したがって一般的に示すと時間は $\text{posl.}(\tau_i) + S_i$ で進められる。ここで τ_i は見出された正値最小または負の残り時間である。このような進め方をすると、Itemごとに時間的な調整を必要とする場合が生ずるが、それは各Itemごとにそのつど処理される。時間は最小単位時間の計数によって表わされるが、この時間の1単位は 0.5^{ms} にとられている。

3.5. 制御プログラム

シミュレーション制御プログラムは、いずれかのItemで事象が生ずると、所定の規則に従って状態量を変化させ、次の事象発生の準備をする。また状態量によって事象の発生を制御する。この制御プログラムの主要な役割は次のようなものである。

(1) 事象の選択: リストをサーチして、残り時間がすべて正の場合、その最小のものあるいは、最初に現われる0または負のものを見出す。

(2) 時間を進める: $\text{posl.}(\tau_i) + S_i$ だけ時間を進める。すなわち、この値を他のItemの残り時間から差し引く。

(3) 乱数の発生: 各Item固有の分布に従う乱数を発生してリストにのせる。

(4) 事象の発生にともなって、各Itemごとに行なわれる状態の変化および各種の制御。

(5) 事象の発生にともなって、全Item共通に行なわれる状態の変化および各種の制御。

(6) 試行回数の監視、各種のデータ作成および印刷その他。

3.6. 乱数の発生

モデル内において、Item 1, 5, 6 および 7 には、それぞれ固有の乱数が用いられる。このうちItem 1(入力発生)はポアソン分布(時間間隔は指数分布)、Item

5 および 6 (TC および FC) は、それぞれドラムへのアクセス回数や処理の仕方によって決められる分布に従うものを必要とする。Item 7 (照合) に関しては、両システムでの呼の処理時間分布によってきまる待ち合わせ分布が求められねばならないが、両システムでの処理時間分布が不明のため、指数分布で代用する。次稿で述べるように、このシステムの特性は FC や TRC チャンネル数によって決定される場合が多いので、この影響は小である。

このような乱数の発生は

(1) (0, 1) 内の一様分布乱数の発生

(2) 一様分布乱数の特定分布への変換

という2段階に分けられる。

3.6.1. 一様分布乱数の発生

一様分布乱数は乗積法による疑似乱数を用いる。 j 回目につくられる一様分布乱数を x_j とすると、

$$x_j = C \cdot x_{j-1} \quad (1)$$

である。ここで

$$C = 5^{17}, x_0 = 1, \text{ modulo } 2^{40},$$

とする。このようにして作られた40ビットの乱数のうち、MSB (Most Significant Bit) 側10ビットを取り出して、一様分布乱数として用いる。

3.6.2. 特定分布乱数への変換

通常行なわれるように、各分布ごとに累積確率関数を横軸上で等分割し、各区分ごとに確率の幅を求め、これを一様分布に対応させる。

3.6.3. 各Itemごとの分布関数

(1) TC 分布 - TC での処理はドラムからの情報読み出しのみであり、独立に5回のアクセスが行なわれる。したがって、この分布は一様分布に従う5個の確率変数の和の分布であり、分布の重畳によって得られる。

この結果として得られる分布関数は正規分布によって良く近似されるので、TC 分布は、分布のすその部分を除いて正規分布を用いている。TC 分布関数を $f_T(\tau)$ とすると、これは次のようなものである。

$$f_T(\tau) = \begin{cases} 0, & \tau < 1 \text{ or } \tau > 5\tau_T + 1, \\ (\tau - 1)^4/24, & 1 < \tau \leq \tau_T + 1, \\ (5\tau_T - \tau + 1)^4/24, & 4\tau_T + 1 \leq \tau < 5\tau_T + 1, \\ \frac{1}{\sqrt{2\pi}\sigma} \exp\left\{-\frac{(\tau - m_T)^2}{2\sigma_T^2}\right\}, & \\ \frac{3}{2}\tau_T > |\tau - m_T|, & \end{cases} \quad (2)$$

$$\text{ただし, } m_T = \frac{5}{2}\tau_T + 1, \quad \sigma_T = \sqrt{\frac{5}{12}\tau_T}$$

ここで TC のドラム・アクセス時間のほかに処理時間 1ms が加えられている。 τ_T はドラム 1 回転時間であり、 m_T および σ_T は、それぞれ TC 分布の平均値および標準偏差で、

$$\tau_T = 6.6\text{ms}, \quad m_T = 17.5\text{ms}, \quad \sigma_T = 4.26\text{ms} \text{ である。}$$

(2) FC 分布 — FC の場合は呼の性質によって三つの場合がある。第 1 は標準の場合で、ドラムへのアクセス回数は 2 回である。ただし第 1 回目はドラムからの情報の読み出し、第 2 回目は読み出された情報を再び元の場所に書き込む。その間に処理が行なわれる。したがって第 1 回目のアクセス時間は一様分布に従うが、第 2 回目は第 1 回目のアクセス後、 $\tau_F + \alpha$ 時間後である。 τ_F は FC のドラム 1 回転時間であり、 α はドラム上で列車インデックス以後、必要とする座席情報のはいつている位置までの回転時間である。これは一般・指定予約や解約などの場合である。これに対し、第 2 の場合は予約でも空席のない場合（無効呼）や、照会などの場合であり、情報の読み出しと、わずかな処理時間のみで処理が済み、アクセスも最初の 1 回のみとなる。さらに第 3 の場合として誤指定の情報があり、この場合には FC にはいる前に検出されるので、FC での処理時間は 0 として扱われる。

以上 3 種の呼が一定の割合で含まれるので、これらの割合を p_1 , p_2 および p_3 とし、このおのおのの場合の分布関数をそれぞれ、 $f_{F1}(\tau)$, $f_{F2}(\tau)$ および $f_{F3}(\tau)$ とすると、FC の分布 $f_F(\tau)$ はこれらの混合で、

$$f_F(\tau) = \sum_{i=1}^3 p_i f_{Fi}(\tau), \quad (3)$$

$$f_{F1}(\tau) = \frac{1}{\tau_F} [u(\tau - \tau_F - \alpha) - u(\tau - 2\tau_F - \alpha)] \quad (3-a)$$

$$f_{F2}(\tau) = \frac{1}{\tau_F} [u(\tau - \alpha) - u(\tau - \tau_F - \alpha)] \quad (3-b)$$

$$f_{F3}(\tau) = \delta(\tau) \quad (3-c)$$

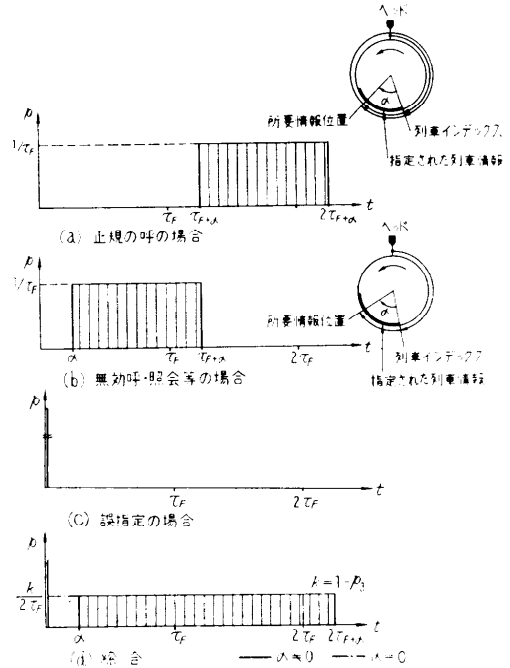
$u(\tau)$; 単位ステップ関数

$\delta(\tau)$; デルタ関数

である。本シミュレーションでは、 p_i の配分を 2 種類用い、また便宜上多少の修正を行なって、次のような 2 種類の FC 分布を用いている（第 5 図参照）。

$$\begin{aligned} \text{FA 型; } f_F(\tau) &= 0.03 \delta(\tau) \\ &+ \frac{0.485}{\tau_F} [1 - u(\tau - 2\tau_F)] \end{aligned} \quad (4)$$

$$\text{FB 型; } f_F(\tau) = \frac{1}{\tau_F} [u(\tau - \tau_F - 1)]$$



第 5 図 FC 処理時間分布

$$-u(\tau - 2\tau_F - 1)] \quad (5)$$

ここで $\tau_F = 22\text{ms}$ である。

(3) 指数分布 — 指数分布 $f_E(\tau)$ は

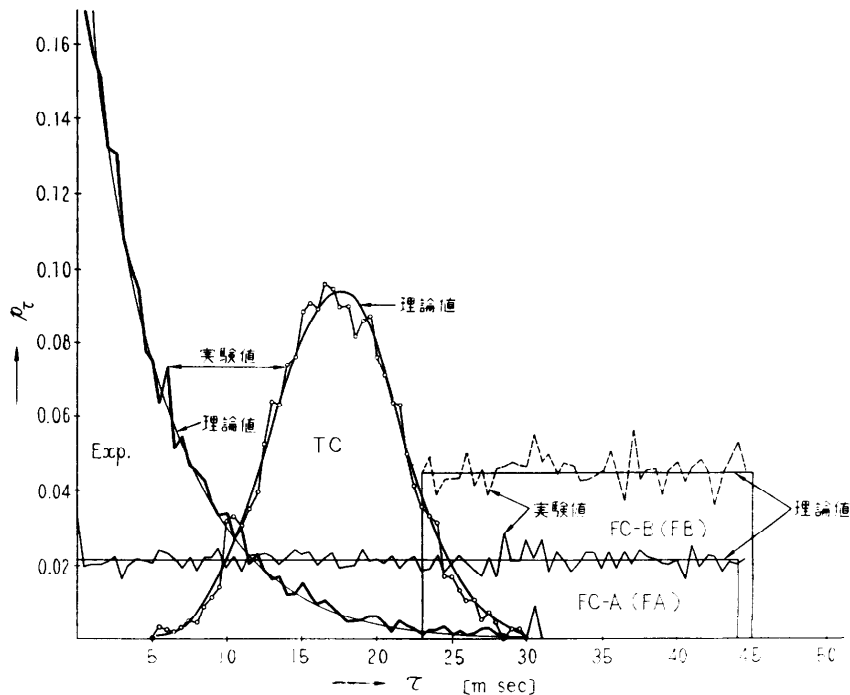
$$f_E(\tau) = \lambda e^{-\lambda\tau}, \quad (6)$$

であるから λ が決まりさえすれば直ちに分布が決まる。指数分布が入力発生に用いられるとき、このパラメータ λ の値は広範囲に変化させる必要がある。そのため基準となる分布を一つ作成しておき、それからすべての場合を導くことにする。基準となる分布は

$$f_E(\tau) = \frac{2}{11} e^{-\frac{2}{11}\tau} \quad (6)$$

である。これは平均値が 5.5ms の分布である。平均値がたとえば 16.5ms のものを必要とするときは、この分布より得られる τ を 3 倍して用いる。指数分布は分布の幅が無限に広がるためリミットを設けて有限のものにしている。以後指数分布を Exp. 分布と表わす。また指数分布の平均値を表示する必要があるときには Exp. (16.5) などと示す。

以上 4 種類の分布について、度数分布を求めたものが第 6 図である。これは FB 分布については 5,000 回、その他の分布については 11,000 回の乱数を発生させ記録したものである。また得られた各分布の平均



第6図 各 Item の分布密度

値および標準偏差を示すと次のようになる。カッコ内は理論値を示し、単位はすべて 1/1,000 秒である。

TC 分布; $m_T=17.27$ (17.50) $\sigma_T=4.14$ (4.26)

FA 分布; $m_F=21.85$ (21.83) $\sigma_F=13.29$ (13.35)

FB 分布; $m_F=34.72$ (34.00) $\sigma_F=6.62$ (6.35)

Exp. 分布; $m_E=5.72$ (5.50) $\sigma_E=5.36$ (5.50)

3.7. 結果の記録

シミュレーションを行なう際、どのようなデータをとるか、を決定することは重要である。本シミュレーションでは、性質の異なった種類のデータを記録・印刷している。一つは試行中の状態の変化をすべて記録する、カレント・データであり、他は結果を累積し各種統計量として試行終了後印刷する統計的データである。

3.7.1. カレント・データの内容

カレント・データとして記録しているのは、

- (1) 試行回数
- (2) 生じた事象間の時間間隔
- (3) 事象の生じた Item No.
- (4)~(10) Item 2~8 の行列内呼数
- (11) TRC チャンネル占有個数
- (12) 試行開始よりの絶対時間

である。

3.7.2. 統計的データ

統計量として、システムの分析に重要な次の6種がとられている。

- (1) 行列内待ち合わせ呼数ごとの占有時間 (Item 3~8)。

シミュレーション実行中に、行列内に ν 個の呼が存在していた状態の時間の総和が各行列ごとにとられている。Item i の行列に ν 個の呼が存在していた時間の和を $T_{i\nu}$ とし、シミュレーション総時間を T とすれば、

$$T = \sum_{\nu=0}^{\nu_{i0}} T_{i\nu}, \nu_{i0}; \text{Item } i \text{ の}$$

行列 i にはいり得る呼数 (8)

であり

$$p_{i\nu} = T_{i\nu} / T \quad (9)$$

が行列 i に ν 個の呼が待ち合わせている確率となる。

- (2) 各待ち合わせ呼数ごとの有効待ち時間 (Item 3~8)。

有効待ち時間は次のように定義される。上記 $T_{i\nu}$ は

$$T_{i\nu} = T_{i\nu_1} + T_{i\nu_2} \quad (10)$$

と分けられ、このうち $T_{i\nu_1}$ は処理装置 i が動作して

いる状態、 T_{iv_2} はこれが停止している状態にある時間である。 T_{iv_1} が有効時間であり、

$$T_{i_1} = \sum_{v=1}^{v_{i0}} T_{iv_1}, \quad T_{i_2} = \sum_{v=1}^{v_{i0}} T_{iv_2} \quad (11)$$

を作ると、処理装置 i が実働の割合

$$e_i = T_{i_1} / T \quad (12)$$

が得られる。

(3) 各待ち合わせ呼数ごとの占有時間分布 (Item 5~7)。

待ち合わせ呼数が k で、かつ処理の残り時間が x と $x+dx$ の間にあるような場合の度数分布をとったものである (詳細略)。

(4) 処理呼数

シミュレーション中に処理された呼の数であり、これを N とする。 T/N はシステム平均処理時間を表わす。

(5) 呼あたりの処理時間の和

呼が受信終了後 SC 内で処理が開始されてから、処理が済んで送信にはいるまでの時間を呼ごとに求め、これの和 T' を作る。 T'/N は呼あたりの平均処理時間を表わす。

(6) 前記 (5) の呼あたりの処理時間の 2 乗和
呼あたりの処理時間の分散を得ることができる。

3.8. パラメータ

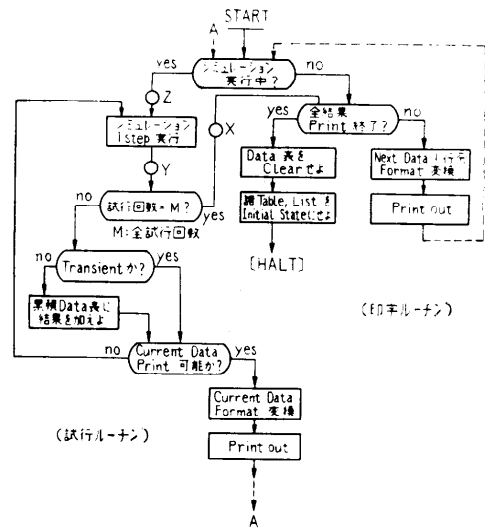
シミュレーションによって、システムの一般的な傾向を見出すため、次のように各種パラメータを変化させている。

- (1) 入力平均時間間隔 (16.5ms~110.0ms)。
- (2) TRC チャンネル数 (8~96)。
- (3) 通信時間 { (受) 2,500ms および { (受) 625ms
 { (送) 2,000ms { (送) 500ms
- (4) システム内部待ち行列の許容長さ。(1~4)
ただし $v_{05}=v_{06}=v_{07}=v_{08}$ としている。
- (5) プログラム実行時間 ($S_i=0 \ i=1\sim 8$) および
($S_1=0, S_2=1\text{ms}, S_3=0, S_4=3\text{ms}, S_5=3\text{ms},$
 $S_6=4\text{ms}, S_7=3\text{ms}, S_8=0$)
- (6) TC および FC の処理時間分布
すでに述べた TC, FC 分布のほか、指数分布を用いた場合を検討している。

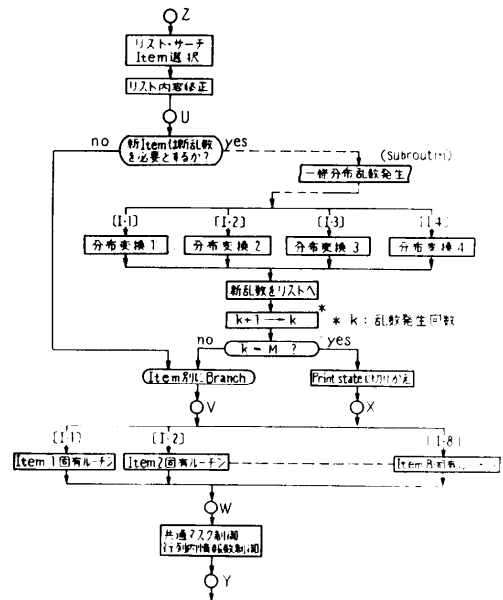
4. シミュレーション・プログラム

シミュレーション・プログラムの全体の構成およびそのうちのシミュレーション 1 ステップ実行ルーチンのフロー・チャートをそれぞれ第 7 図および第 8 図に

示す。



第 7 図 シミュレーション全体概略流れ図



第 8 図 シミュレーション試行ルーチン

4.1. プログラムの全体の構成

シミュレーション・プログラムは大きく分けて、試行ルーチンと結果の印字ルーチンから成っている。

試行ルーチンはループを形成して、事象が生ずるたびに 1 サイクルが実行される。この内容はシミュレーション 1 ステップの実行と、統計的データの作成、試

行回数の監視, カレント・データの記録印字などを行なう。ただし統計的データの作成は, 過渡状態を終了して定常状態にはいった後行なう。カレント・データはプリント時間を短縮するために, 1 行に 3 データずつまとめられているので, 3 ステップに 1 度の割合でプリントされる。

結果の印字ルーチンは, シミュレーション中に作成された統計的データを, 所定のフォーマットに従ってプリントし, プリント終了後リストや諸テーブル内容をシミュレーション実行前の状態に戻しておく。これはパラメータを変えて引き続き試行を繰り返すことを容易にするためである。

4.2. シミュレーション試行ルーチン

シミュレーションの 1 ステップ試行ルーチンは, まずリスト・サーチおよびリスト内容の修正を行なう。これはすでに述べたように, 8 種の Item の残り時間をサーチして, 最初に現われた 0 または負のものあるいはすべて正の場合にはその最小のものを見出し, $\text{posi.}(\tau_i) + S_i$ を作って, 他の Item の残り時間から差し引くことである。サーチは優先順位に従って行なわれ, また, マスクのあるものはスキップする。

求まった Item No. が 1, 5, 6 および 7 のものについては, それぞれ固有の分布に従う乱数が作られ, リストに書き込まれる。試行回数は乱数発生回数によって規定しているので, この数を監視し, 所定回数に達したら結果の印字ルーチンに移る。

次いで各 Item ごとの固有のルーチンにはいり, 行列内の呼数修正や, マスク制御が行なわれる。また Item 2, 3, 4 および 8 については, 次の事象発生ま

での時間がリストに書き込まれる。Item 2 および 7 では, 3 および 8 のためのサービス時間 (3.3. 参照) が求められ, 表に記録される。この後各 Item で共通に行なわれる状態量の変化やマスク制御がまとめて行なわれる。

5. む す び

本稿は MARS 101 座席予約システムが実用段階にはいる前に, その中央制御装置 HITAC 3030 を用いて行なったシミュレーションの方法ならびにプログラムについて述べたものである。

本シミュレーションはパラメータの各種組み合わせ約 150 組についてデータを求めた。この結果は一部 MARS 101 のプログラム作成の資料として用いられた。シミュレーションの実施ならびにその結果の分析については追って報告する。

シミュレーションの実施に際しては, 東大宇宙航空研究所穂坂衛教授に各種のご指導を頂き, また日立製作所阿部城一氏はじめ方式課の諸氏および浅井政幸氏には, ご多忙の折にもかかわらず多大なご協力を頂いた。ここに厚く感謝する次第である。

参 考 文 献

- 1) 穂坂 衛: 座席予約, 昭和 39 年電気 4 学会連合大会予稿集, S5.
- 2) 斉数賢次郎: 国鉄における座席予約の自動化について, 情報処理 Vol. 5, No. 1
- 3) 日立評論 座席予約システム特集号: Vol. 46 No. 6, 1964 年 6 月

(昭和 39 年 6 月 26 日受付)