7,5 (sete e meio)

ESCOLA POLITÉCNICA DA
UNIVERSIDADE DE SÃO PAULO
DEPARTAMENTO DE ENGENHARIA MECATRÔNICA

SIMULADOR DE SISTEMAS HÍBRIDOS

AUTOR: CHEN YEN-TSANG

ORIENTADOR: PROF. DR. PAULO EIGI MIYAGI
COLABORADORA: MESTRE ENG. EMILIA VILLANI

SÃO PAULO 10 DE DEZEMBRO DE 2001

RESUMO

O presente trabalho consiste na implementação de um simulador de sistemas híbridos utilizando as ferramentas Matlab e Simulink.

O fundamento teórico para o desenvolvimento do trabalho é baseado em "Redes Predicado/Transição Diferencial", que incorpora as propriedades de uma "Rede de Petri" com as características de um sistema de variáveis contínuas.

O algoritmo do simulador e a metodologia de montagem de um modelo de sistema híbrido foram estudadas e um protótipo do simulador foi implementada

O simulador foi utilizado para análise de um estudo de caso para avaliar a sua eficiência no apoio ao desenvolvimento de sistemas.

Os resultados indicam que a ferramenta proposta é efetiva na concepção e especificação funcional de sistemas híbridos.

1 Introdução

Em um sistema a eventos discretos (SED), o estado do sistema é alterado pela ocorrência de eventos considerados instantâneos, isto é, onde o tempo de duração do evento não é relevante em relação aos estados discretos do sistema. A ocorrência destes eventos depende apenas da satisfação de pré e pós condições do sistema [Miyagi, 1996]. Por outro lado em um sistema de variáveis contínuas (SVC) a evolução dos estados do sistema é contínua em função do tempo [Villani, 2000]. O estado do sistema é representado por variáveis que são também modificadas de forma contínua assumindo infinitos valores.

Existem ainda casos de sistema em que se verifica a ocorrência de características de SED e SVC simultaneamente. Tais casos são designados como "Sistemas Híbridos" [Villani, 2000].

1.1 Objetivo

O objetivo deste trabalho é o desenvolvimento de um simulador para sistemas híbridos. O simulador deve permitir a edição e simulação de modelos de sistemas que tenham uma parte puramente discreta, uma parte puramente contínua e uma parte híbrida que representa a interface entre a parte contínua e a parte discreta.

Há várias maneiras de se modelar um SED, mas no caso deste trabalho optou-se pelas Redes de Petri (RdP) devido as suas bem conhecidas características para descrição de fenômenos como paralelismo, concorrência,

sincronismo, etc [Arata, 1994], além da simplicidade de representação gráfica aliada a uma formulação matemática. Para modelagem de SVC utiliza-se sistemas de equações diferenciais. Para a parte híbrida, consideram-se técnicas derivadas de Redes de Petri que incluem a representação de equações diferenciais.

Este trabalho está dividido em duas partes: A primeira constitui na implementação de um simulador híbrido através do desenvolvimento de um simulador de Redes de Petri e de uma interface com um simulador de sistemas de equações diferenciais.

A segunda parte do projeto constitui na aplicação do simulador em um estudo de caso real que serve de base para a avaliação do simulador.

1.2 Motivação

Como a pesquisa na área de sistemas híbridos é relativamente recente [Villani, 2000], técnicas de análise destes sistemas baseiam-se principalmente em simulação, isto é, o estudo e projeto desta classe de sistema é baseado na análise da verificação do comportamento dinâmico de modelos. O desenvolvimento de um simulador é portanto de grande utilidade.

Atualmente, existem vários simuladores de sistemas à eventos discretos baseados em Redes de Petri disponíveis no mercado assim como para sistemas de variáveis contínuas. No entanto o mesmo não é válido para sistemas híbridos. Apesar de já existirem alguns simuladores baseados em Redes de Petri para sistemas híbridos, eles não apresentam flexibilidade para a modelagem da parte contínua e discreta simultaneamente, um exemplo é o Visual Object Net desenvolvido e baseado nas Redes de Petri Híbridas [Alla &

David, 1998]. Portanto considera-se relevante o desenvolvimento de um simulador que possa representar de forma flexível ambas as partes: discreta e contínua.

2 Estrutura dos Sistemas Híbridos

Os sistemas híbridos considerados neste trabalho são aqueles que apresentam a seguinte estrutura (Figura 1):

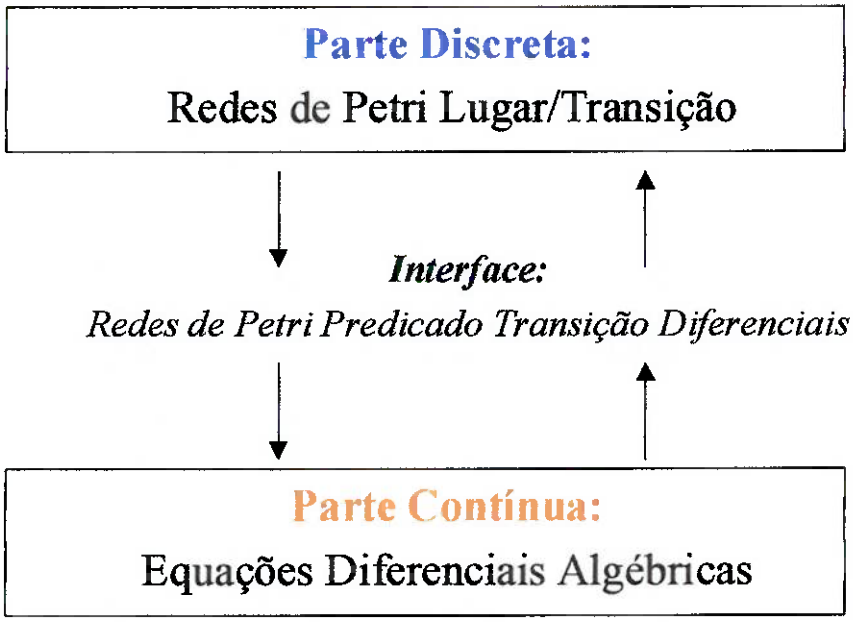


Figura 1 - Tipos de Sistemas

Tem-se uma parte puramente discreta para a qual escolheu-se como técnica de modelagem as Redes de Petri Lugar/Transição [Miyagi,1996] as quais foram introduzidos "arcos habilitadores" e "arcos inibidores" e tem-se uma parte puramente contínua modelada através de equações diferenciais.

A interface híbrida entre a parte contínua e discreta é realizada através das Redes de Petri Predicado/Transição Diferenciais [Villani,2000].

A escolha destas ferramentas de modelagem se encontra justificada no Capítulo 3, onde são apresentados os fundamentos teóricos e as características destas técnicas.

3 Fundamentos Teóricos

3.1 Redes de Petri

A Rede de Petri é uma técnica que tem-se mostrado efetiva para o estudo, análise e modelagem de sistemas a eventos discretos. Ela pode ser representada de forma gráfica (grafos) ou na forma de matrizes.

A representação gráfica da Rede de Petri é baseada em grafos bipartidos e orientados, isto é, são considerados dois tipos de nós, os "lugares" (*places* – representados por círculos) e as "transições" (*transitions* – representados por barras ou retângulos). A ligação entre os nós é feita por "arcos orientados" (setas). A Figura 2 abaixo mostra os elementos de uma RdP.

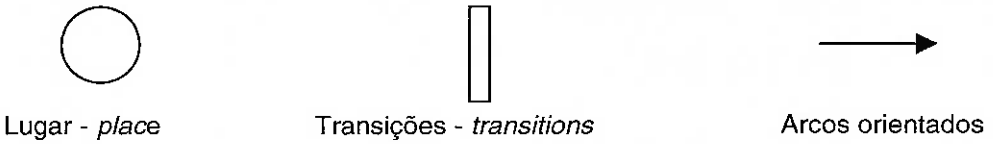


Figura 2 – Elementos de uma Rede de Petri

A Rede de Petri é um multigrafo, isto é, ela permite múltiplos arcos de um nó do grafo para outro, no entanto estas ligações devem respeitar uma condição:

Os arcos orientados não podem ligar diretamente lugares sem passar por uma transição ou transições diretamente sem passar por um lugar, isto é, não pode haver ligações entre nós do mesmo tipo. Esta regra está ilustrada na Figura 3 e Figura 4 abaixo.

Figura 3 - Exemplos de construções inconsistentes

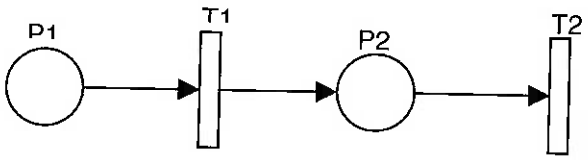


Figura 4 – Exemplo de modo consistente de representação

A rede de Petri é "marcada" quando há "marcas" no interior dos lugares. No exemplo citado na Figura 5, a rede de Petri é uma rede marcada com marcas nos lugares P1 e P2.

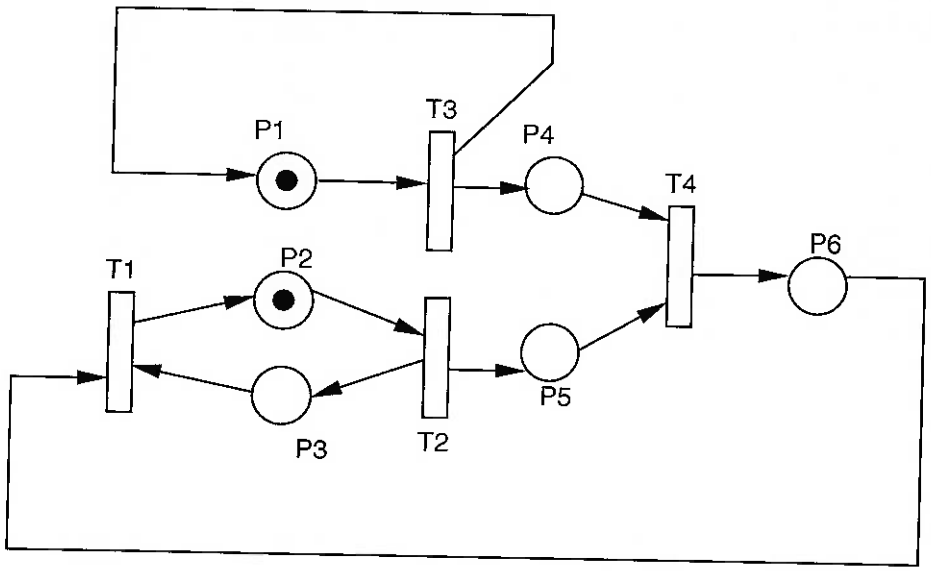


Figura 5 – Exemplo de rede de Petri

Ainda há os casos de arcos orientados com "pesos" e lugares com determinada "capacidade" de marcas, que serão apresentados e explicitados posteriormente.

A rede de Petri pode ser subdividida em três grandes categorias:

- Rede Condição/Evento – neste tipo de rede cada lugar pode receber apenas uma marca.
- Rede Lugar/Transição – ao contrário das redes condição/evento, os lugares podem conter mais de uma marca.
- Redes de Petri de Alto Nível – as marcas são individualizadas, isto é existem marcas de diferentes tipos no grafo.

3.1.1 Regras de execução

A "execução" da rede de Petri é baseada no número e distribuição das marcas ao longo dos lugares da rede. A rede de Petri é executada através dos "*disparos*" das transições. As transições só podem ser disparadas quando estão "*habilitadas*".

Diz-se que um "elemento A está à entrada do elemento B" quando o arco orientado de A para B tem a origem no elemento A, e diz-se que um "elemento A está à saída do elemento B" quando o arco orientado de B para A tem o destino no elemento A. Uma transição está habilitada quando todos os lugares que estão à sua entrada estão marcados e todos os lugares que estão à sua saída podem receber marcas.

No exemplo ilustrado pela Figura 6 tem-se uma rede de Petri do tipo Condição/Evento e pode-se notar que os lugares à entrada da transição T1

estão marcadas e o lugar à saída está livre, portanto a transição T1 está habilitada.

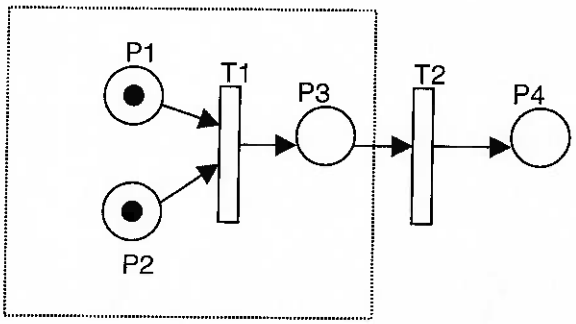


Figura 6 – Transição T1 habilitada, antes do disparo

A transição estando habilitada, dispara. Ao ocorrer o disparo as marcas dos lugares que estão à entrada da transição são retiradas em número definido pelo peso dos respectivos arcos e novas marcas são colocadas nos lugares à saída da transição também em número definido pelo peso dos respectivos arcos. No exemplo anterior após o disparo da T1 a rede fica com a configuração da Figura 7

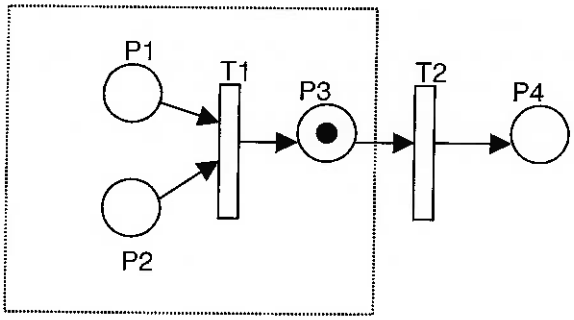


Figura 7 – Marcação obtida após o disparo de transição T1

Para redes Lugar/Transição o lugar de saída não necessita estar livre, basta que o lugar à saída tenha uma capacidade estabelecida suficiente para receber as marcas adicionais. A capacidade é o número máximo de marcas que um lugar pode comportar.

Caso os arcos de entrada da rede Lugar/Transição possuam um peso n, então deve-se retirar n respectivas marcas dos lugares à entrada. E no caso dos arcos de saída da rede Lugar/Transição terem um peso m, então deve-se acrescentar m respectivas marcas nos lugares à saída. Para ilustrar a regra citada tem-se a Figura 8 e a Figura 9

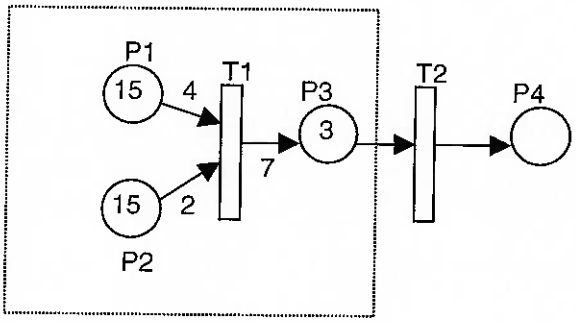


Figura 8 – Exemplo de rede de Petri Lugar-Transição com peso nos arcos.

Após o disparo do T1 na Figura 8, a rede fica com a nova marcação mostrada na Figura 9.

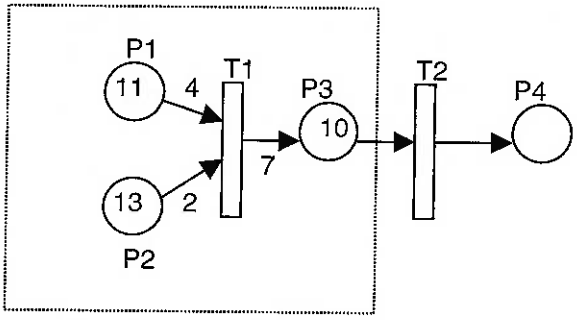


Figura 9 – Marcação da rede da Figura 8 após o disparo da transição T1

3.1.2 Representação matricial

Uma outra forma de representação da rede de Petri é através de relações de matrizes.

Na forma matricial as colunas estão relacionadas com as transições, as linhas estão relacionadas aos lugares e os elementos correspondentes representam o peso dos arcos.

Os elementos que compõem a representação matricial são:

- Matriz de pré-condição: representa os pesos dos arcos que ligam um lugar à uma transição.
- Matriz de pós-condição: representa os pesos dos arcos que ligam uma transição à um lugar.
- Vetor de estado inicial: é um vetor que representa a "marcação inicial da rede".
- Vetor de disparo (*firing vetor*): é o vetor que indica as transições que podem ser disparadas.

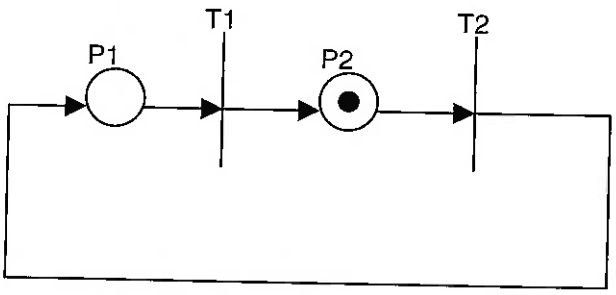


Figura 10 – Exemplo de uma rede de Petri.

Para a rede da Figura 10 tem-se as seguintes representações matriciais.

A matriz de pré-condição fica com os seguintes valores:

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

Para a matriz A, de uma rede condição/evento, entende-se que para ativar a transição T1 é necessária uma marca no lugar P1 ($a_{11} = 1$) e para ativar a transição T2 deve-se ter uma marca no lugar P2 ($a_{22} = 1$)

A matriz de pós-condição tem os seguintes valores para o caso citado na Figura 10:

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

De acordo com a matriz B, após o disparo da transição T1, não haverá acréscimo de marca no lugar P1 ($b_{11} = 0$) e haverá a adição de uma marca no lugar P2 ($b_{21} = 1$). Após a o disparo da transição T2 haverá um acréscimo de uma marca no lugar P1 ($b_{12} = 1$) e não haverá acréscimo de marca no lugar P2 ($b_{22} = 0$).

A matriz C chamada de "matriz de efeito líquido", é a diferença entre a matriz B e a matriz A .

$C = B - A$.

Para o caso estudado tem-se:

$$C = \begin{bmatrix} -1 & 1 \\ 1 & -1 \end{bmatrix}$$

De acordo com a matriz C, após o disparo da transição T1 haverá a retirada de uma marca no lugar P1 ($c_{11} = -1$) e o acréscimo de uma marca no lugar P2 ($c_{21} = 1$). O efeito após o disparo da transição T2 é o acréscimo de uma marca no lugar P1 ($c_{12} = 1$) e retirada de uma marca no lugar P2 ($c_{22} = -1$).

O vetor de marcação inicial M_0 neste caso é da seguinte forma:

$$M_0 = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

A interpretação do vetor de marcação inicial é de nenhuma marca no lugar P1 ($m_{01} = 0$), e de uma marca no lugar P2 ($m_{02} = 1$).

O vetor de transições habilitadas apresenta os seguintes valores:

$$\sigma = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

A regra de execução é evidentemente a mesma da representação gráfica, porém para observar o fluxo das marcações é necessário a aplicação da equação fundamental:

$$M = M_0 + C \cdot \sigma$$

Onde:

M: vetor de marcação atual;

M_0: vetor de marcação inicial;

C: matriz resultante da subtração entre as matrizes de pós-condição e pré-condição

σ: Vetor das transições habilitadas.

Aplicando a equação acima pode-se observar o resultado do fluxo das marcações da rede de Petri desejada.

Para o exemplo da Figura 10, tem-se:

$$M = \begin{bmatrix} 0 \\ 1 \end{bmatrix} + \begin{bmatrix} -1 & 1 \\ 1 & -1 \end{bmatrix} \cdot \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

O resultado desta manipulação algébrica é:

$$M = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

que é, exatamente, o estado da rede após o disparo da transição T2. ilustrada na Figura 11.

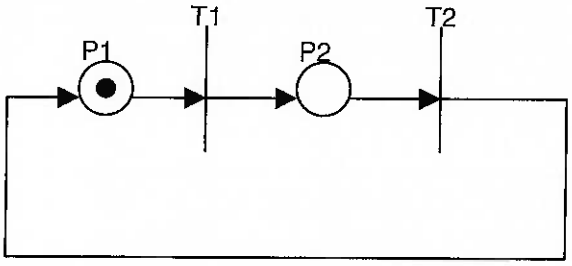


Figura 11 – Rede de Petri da Figura 10 após o disparo de T2

3.1.3 Introdução de Arcos Habilitadores e Inibidores

Na definição original da Rede de Petri Lugar/Transição, elas possuem apenas arcos orientados, no entanto, neste trabalho, para aumentar o poder de modelagem da parte discreta, introduz-se o conceito de "arcos habilitadores" e "inibidores".

A definição destes arcos foi baseada na definição dos "gates habilitadores" e "inibidores" do MFG [Miyagi,1996].

Um "arco habilitador" ou "inibidor" conecta um lugar ou um "elemento externo" à transição. O conceito de "elemento externo" também vem do MFG e será utilizado para realizar a conexão com a parte contínua, conforme descrito no Capítulo 5.

A representação de arcos habilitadores e inibidores está indicado na Figura 12 na página seguinte

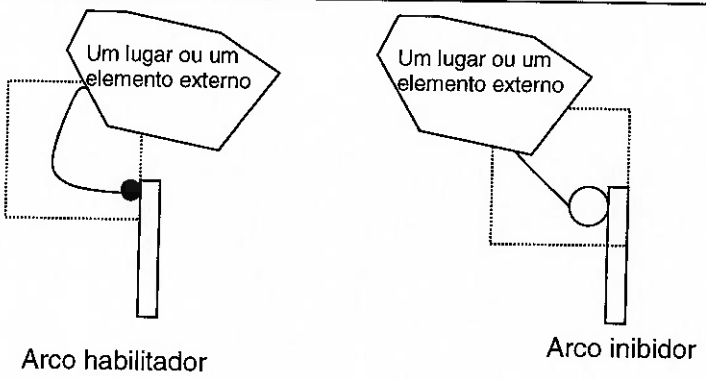


Figura 12 – Arcos habilitadores e arcos inibidores

Os arcos habilitadores e inibidores também podem ser considerados na representação matricial.

Para o arco habilitador, a transição a ele conectado estará habilitada se o lugar de origem do arco conter uma marca ou se o elemento externo corresponder a um sinal válido (equivalente a 1), e as pré e pós-condições da transição forem satisfeitas, Figura 13.

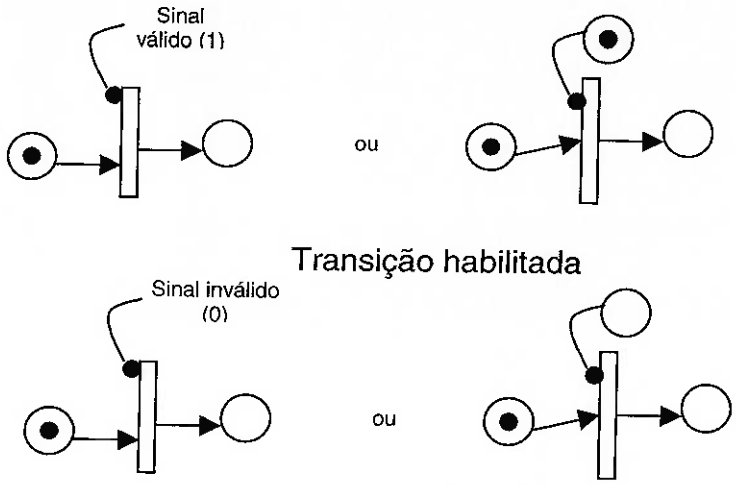


Figura 13 – Arcos habilitadores

O arco inibidor tem um funcionamento análogo mas inverso ao habilitador. Para um arco inibidor, a transição a ele conectado estará habilitada se o lugar de origem do arco não tiver nenhuma marca ou se o elemento

externo corresponder a um sinal inválido (equivalente a 0), e se as pré e pós condições da transição forem satisfeitas, Figura 14.

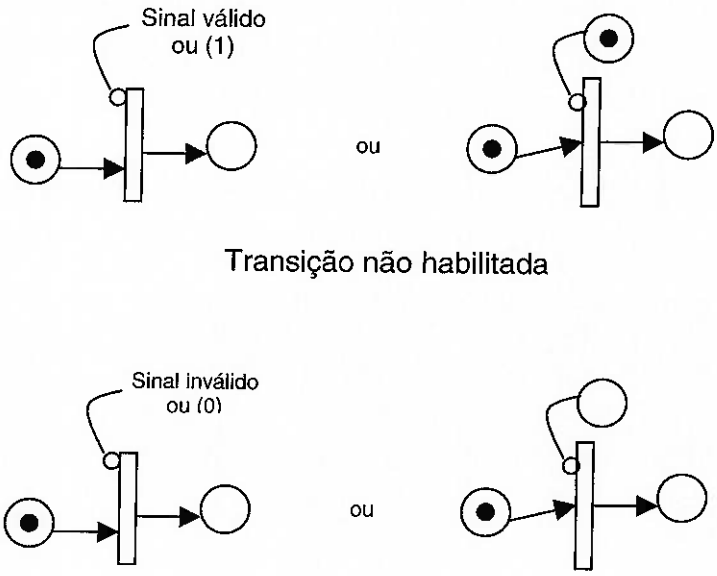


Figura 14 – Arcos inibidores

No disparo da transição, a diferença entre um lugar conectado à transição por um arco orientado e lugar conectado por um arco habilitador ou inibidor está na marcação após o disparo. Enquanto no primeiro caso retira-se uma marca do lugar, no segundo caso a marcação do lugar permanece inalterada, Figura 15.

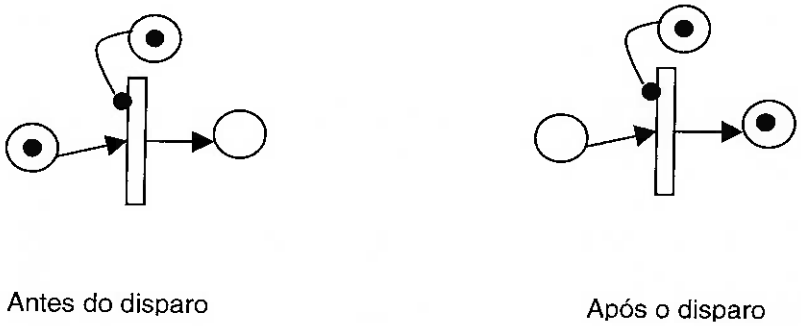


Figura 15 – Diferença de marcação entre lugares conectado por diferentes tipos de arcos

Do ponto de vista exclusivo da habilitação de uma transição, um arco habilitador pode ser considerado como sendo uma pré-condição como ilustra a Figura 16 abaixo:

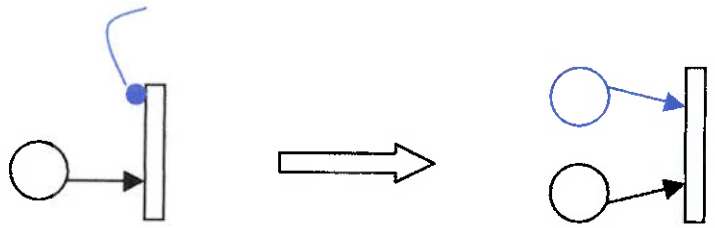


Figura 16 – Modelagem de arco habilitador

Da mesma forma, do ponto de vista exclusivo da habilitação de uma transição, os arcos inibidores podem ser considerados como uma pós condição para o disparo dela com indica o exemplo da Figura 17, onde o lugar correspondente ao arco inibidor tem capacidade unitária:

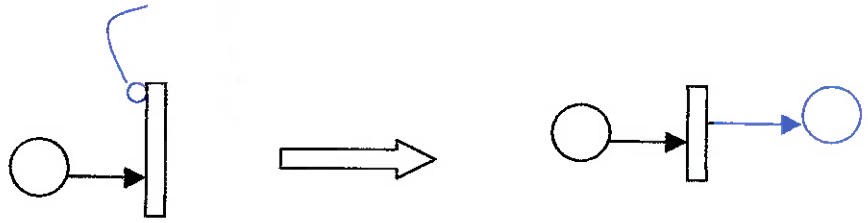


Figura 17 – Modelagem de arco inibidor

Os lugares introduzidos para considerar os arcos habilitadores e inibidores têm capacidade unitária e o disparo da transição não deve mudar a marcação destes. Assim, esses lugares são equivalentes aos chamados **pseudo-boxes** do MFG [MATSUSAKI,1998].

Para melhor compreensão destas considerações, apresenta-se o exemplo da Figura 18.

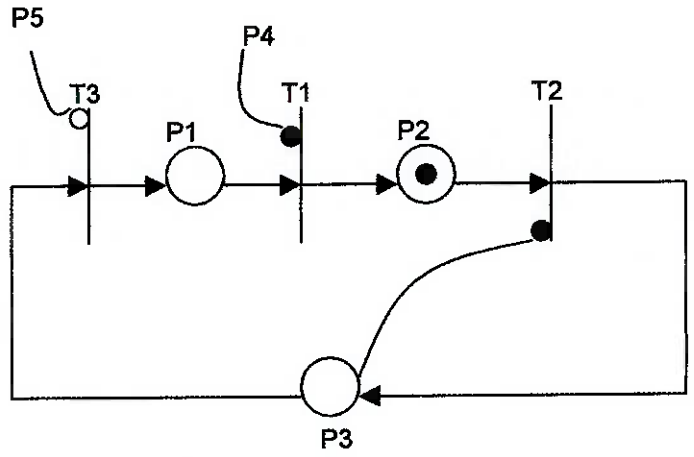


Figura 18 – Exemplos de arcos habilitadores e inibidores

Considerando-se que os arcos habilitador e inibidor podem ser modelados através da introdução do conceito de **pseudo-box,** a rede da Figura 18 pode ser transformada para grafo da Figura 19.:

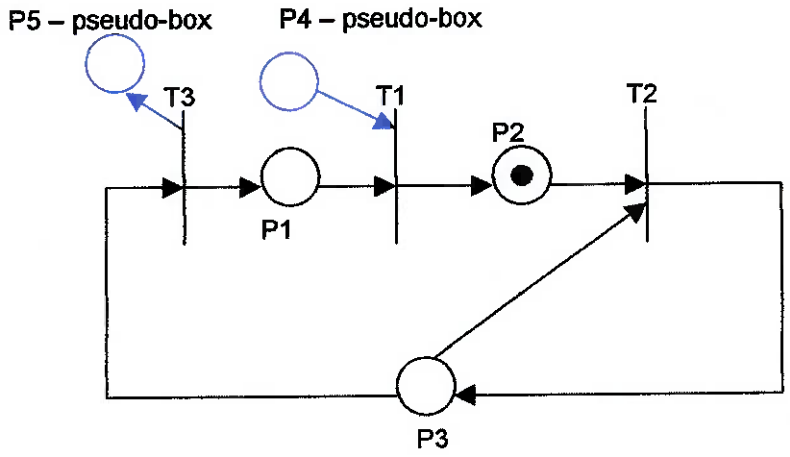


Figura 19 – Rede equivalente à rede da Figura 18

As matrizes de incidência da rede ficam assim da seguinte maneira, onde A representa a matriz de pré condição e B representa a matriz de pós condição:

$$A = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 1 & 0 \\ \hline 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

Parte dos lugares normais

Parte dos pseudo-boxes

$$B = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ \hline 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Parte dos lugares normais

Parte dos pseudo-boxes

Observação: O conceito de arcos habilitadores e inibidores não existe na teoria original das redes de Petri. Os conceitos apresentados acima fazem parte da teoria do PFS/MFG, no entanto são devidamente adicionadas à rede de Petri para implementar e modelar a interface entre os sistema contínuo e discreto do simulador.

3.1.4 Habilitação e disparo das transições

Para verificar se uma transição está ou não disparável tem-se os seguintes passos:

1. Verificar quais as transições cujas pré-condições estão todas satisfeitas.
2. Verificar quais as transições cujas pós-condições estão todas satisfeitas.
3. Fazer a interseção dessas transições e guardar o resultado em um vetor I.

4. Verificar quais as transições que estão habilitadas pelos arcos habilitadores;
5. Verificar quais as transições estão habilitadas pelos arcos inibidores;
6. Fazer a interseção dessas transições e guardar o resultado em um vetor II;
7. Fazer interseção entre os vetores I e II assim obter o vetor com as transições disparáveis.

Para a implementação do simulador foram adotados dois tipos de Redes de Petri. A parte do sistema discreto é implementada através do modelo baseado em Redes de Petri Lugar/Transição [Miyagi,1996], o sistema híbrido é modelado com Redes de Petri Predicado/Transição Diferencial [Villani,2000].

A capacidade máxima dos lugares da Rede de Petri Lugar/Transição será definido pelo usuário, no entanto, as capacidades dos lugares da Rede de Petri Predicado/Transição Diferencial e dos lugares que modelam arcos habilitadores e inibidores serão todas unitárias de acordo com [Villani, 2000].

O peso dos arcos orientados é definido pelo usuário nas matrizes de incidência das redes. Vale lembrar que o peso dos arcos da Rede de Petri Predicado/Transição Diferencial e dos arcos habilitadores e inibidores são todos unitários.

Quanto à temporização das transições, esta não será implementada neste trabalho, pois isso envolve a sincronização dos tempos entre a parte discreta e a parte contínua do simulador, além disso, se houver necessidade de

utilização de uma transição temporizada, esta pode ser representada através de um modelo de elemento com variáveis contínuas.

3.2 Redes de Petri para Sistemas Híbridos

Nas Redes de Petri para sistemas híbridos se verifica a presença simultânea de partes descritas por **variáveis discretas** (estados discretos que são modificados pela ocorrência de eventos instantâneos) e outra parte descrita por **variáveis contínuas** (variáveis que evoluem continuamente no tempo).

Há vários tipos de Redes de Petri que são capazes de representar esses sistemas. Citando alguns exemplos [Villani,2000]:

- Redes de Petri Híbridas – A Rede de Petri Híbrida é uma extensão da Rede de Petri. Nela coexistem elementos discretos e contínuos, os "lugares contínuos" apenas admitem marcações com números reais e não negativas e às "transições contínuas" são associadas velocidades máximas de disparo representando um fluxo contínuo de marcas. A "velocidade de disparo" de uma transição é a velocidade máxima definida para ela enquanto a marcação dos lugares de entrada for suficiente, caso contrário a velocidade de disparo é limitada pela somatória da quantidade de marcas presentes nos lugares de entrada e dos fluxos de marcas que estão entrando nestes lugares.
- Redes de Petri Diferenciais – Neste caso é introduzida a definição de "transição diferencial" (onde a velocidade de disparo pode ser constante, uma combinação linear ou uma função não linear da marcação dos "lugares diferenciais" ligados à equação diferencial) e "lugar diferencial" (onde a marcação é um número real). Neste tipo de rede, associa-se uma

"freqüência de disparo" a cada transição, representando o passo de integração utilizado no cálculo da integral da equação da transição.

- Redes de Petri de Alto Nível – Neste caso tem-se conceito de orientação a objetos. Cada marca é um objeto. A linguagem utilizada na definição de objetos e classes é baseada em C++. Entretanto, tal tipo de Rede de Petri é de grande complexidade, o que pode comprometer a efetiva análise de sistemas.
- Redes de Petri Predicado/Transição Diferencial (PTD) – Neste caso, observa-se que os lugares representam configurações do sistema. Assim a cada lugar é associada um sistema de equações diferenciais para a evolução das variáveis contínuas no tempo, ou seja, o sistema de equações associado à variável contínua determina a sua evolução no tempo.

A rede adotada no presente trabalho para modelar um sistema híbrido é a Rede de Petri Predicado/Transição Diferencial (PTD). A adoção desta rede é em função de sua flexibilidade na modelagem de sistemas e pela facilidade na compreensão de suas regras.

3.2.1 Redes Predicado/Transição Diferenciais (PTD)

Este tipo de Rede de Petri para modelagem de sistemas híbridos diferencia-se da Rede de Petri Lugar/Transição em dois aspectos:

1. Os lugares representam configurações do sistema e são associados a sistemas de equações;
2. As transições possuem "funções de habilitação" e "funções de junção".

Quando um lugar está marcado o sistema de equações a ele associado determina a evolução no tempo das variáveis continuas.

Para o disparo das transições, os valores das variáveis continuas que são responsáveis pela sua habilitação são fornecidas como parâmetros de entrada de uma **função de habilitação** do tipo: "se a expressão 1 for maior, igual ou menor que a expressão 2 então a transição está habilitada". É preciso lembrar que neste tipo de rede o disparo segue também todas as regras da Rede de Petri Lugar/Transição, porém é adicionada uma condição que depende das variáveis contínuas.

Ainda a respeito de transições, o segundo tipo de função associada à transição que existe na rede PTD é chamada de **função de junção**. Este tipo de função atualiza o valor das variáveis contínuas quando a transição é disparada, em outras palavras, ela modifica as variáveis contínuas segundo algumas regras pré-estabelecidas pelo modelo, de uma forma discreta.

Uma restrição considerada na utilização da rede PTD é que os lugares comportam apenas uma marca, que define quais são as equações utilizadas naquele instante para as variáveis contínuas A rede PTD deve ser

construída de modo que exista sempre um lugar desta rede que esteja marcado, garantindo assim a consistência das variáveis contínuas, isto é, em qualquer instante a variável apresentará um valor coerente sem interrupção ou descontinuidade nos valores assumidos ao longo do tempo. Para melhor compreensão das Redes de Petri Predicado/Transição Diferencial apresenta-se um exemplo a seguir.

3.3 Exemplo de Sistema Híbrido

O exemplo da Figura 20 ilustra o processo de montagem das laterais de um perfil de um trem com junção das peças através de uma soldagem O processo pode ser modelado como sendo um sistema com características híbridas.

A peça dos laterais dos trens são reunidas no estágio 1 onde verifica-se o número das peças e a sua classificação. Verificada a quantidade e a conformidade das peças, inicia-se a montagem. Nesta atividade cada peça é colocada nas suas respectivas posições e fixadas, temporáriamente, com grampos até se iniciar o processo de soldagem.

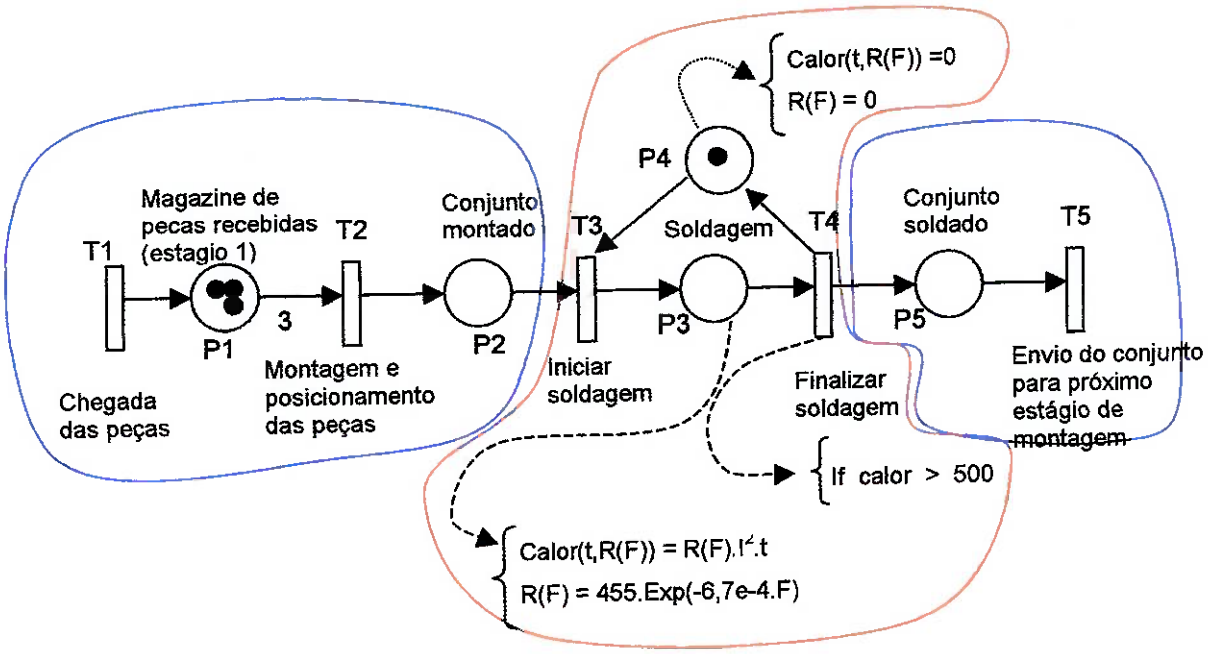


Figura 20 – Exemplo de uma RdP Predicado/Transição

Nesta primeira parte do sistema, a ocorrência dos eventos depende apenas da satisfação das condições e o tempo não é relevante neste caso, portanto pode-se tratar esta parte como sendo um SED.

Após o posicionamento das peças, inicia-se o processo de soldagem. Na soldagem por resistência (solda a ponto), as peças metálicas a serem unidas, são presionadas uma contra a outra, por meio de eletrodos, fazendo-se passar por estes uma alta corrente elétrica em um curto periodo de tempo. Essa corrente aliada à resistência entre as superficies em contato sob pressão do eletrodo gerará calor, por meio do efeito Joule, elevando a temperatura, na região de contato, até a temperatura de fusão das peças [FUTEMA, 1997].

Essa atividade de soldagem respeita a seguinte matemática:

$Calor(t,R(F)) = R(F).I^2.t$

$R(F) = 455.Exp(-6,7e-4.F)$

onde I = Corrente - constante

t = Tempo

R(F) = Resistência em função da força aplicada pelos eletrodos.

Através da relação acima, observa-se que o calor gerado varia com o decorrer do tempo, portanto a atividade de soldagem pode ser considerado como um SVC.

Após a soldagem das peças, o conjunto é liberado e é transportado para um outro estágio. Novamente, esta atividade de liberação e transporte do conjunto concluído depende somente das pré e pós condições, portanto é considerado com sendo um SED.

Como o sistema apresenta tanto características de SED e SVC, ela é considerada um Sistema Híbrido,

Na rede da Figura 20, a parte em azul representa a parte discreta do sistema, enquanto a parte em vermelho representa a parte híbrida. No lugar P1, cada marca representa uma peça da lateral do trem. Para começar a montagem, representada pela transição T2, as quantidades das marcas em P1 deve ser de pelo menos 30, isto é, a quantidade necessária para a tarefa de montagem. Atingida essa marcação T2 é habilitada e dispara. Após o disparo de T2, 30 marcas do P1 são retiradas e uma marca é colocada no lugar P2, e representa a lateral do trem montado. Se o lugar P3 está livre, a transição T3 é habilitada e dispara, isso representa o início do processo de soldagem. A marca de P2 é retirada e uma marca é colocada em P3. P3 ao ficar marcado, inicia o processamento das variáveis com o decorrer do tempo e a cada intervalo adequado os valores das variáveis serão verificados pela função de habilitação da transição T4. Uma vez que os valores das variáveis satisfaçam a função de habilitação da T4, esta fica habilitada e dispara. O disparo desta transição remove a marca de P3 e coloca uma marca nos lugares P4 e P5. O lugar P4 indica que as variáveis do sistema contínuo devem retornar ao valor inicial, isto é necessário para manter a continuidade da função modelada. O lugar P5 indica que a lateral do trem está soldada e o conjunto pronto para ser transportado para o próximo estágio de montagem.

4 Implementação e metodologia para criação do modelo de um sistema híbrido no simulador

A implementação do simulador é dividida em três partes:

- Parte discreta que é implementada em Matlab [Matsumoto,2001], usando a representação matricial;
- Parte contínua em sistemas de equações diferencias, que é implementada em Simulink [Matsumoto,2001], usando uma interface de diagrama de bloco;
- Parte da interface entre o sistema discreto e contínuo que é feita através da Rede de Petri Predicado/Transição Diferencial. A implementação é realizada por meio da interface Matlab-Simulink.

A simulação de Rede de Petri foi implementada com base na representação matricial. O programa utilizado para implementação foi o Matlab.

Para a modelagem da parte contínua do sistema utiliza-se o programa SIMULINK, que tem demostrado possuir uma boa capacidade e versatilidade como um simulador de um sistemas de variáveis contínuas. O SIMULINK fornece todo o suporte necessário para a integração entre os modelos adotados para implementação do simulador.

No processo de integração dos sistemas, o SIMULINK efetua as simulações das partes contínuas e devolve ao sistema discreto os resultados em valores binários, isto é, 0 ou 1.

Estes valores são os resultados utilizados pelas funções de habilitação da Rede de Petri Predicado/Transição Diferencial. Do ponto de vista

do simulador discreto, eles se comportam como sinais provenientes de elementos externos que habilitam ou não transições através de arcos habilitadores/inibidores.

Por outro lado, o sistema discreto devolve à parte contínua a marcação da rede Predicado/Transição Diferencial, que será utilizada para habilitar os sistemas de equações diferenciais correspondentes.

A função de junção é simulada da seguinte forma: a parte discreta do simulador fornece um sinal binário como saída e de acordo com este sinal, a parte contínua atualiza ou não o valor das variáveis contínuas.

Dentro do bloco ***"Matlab function"*** está o simulador de sistemas discretos, as suas entradas são as matrizes de pré e pós condição da Rede de Petri, os arcos habilitadores e inibidores, as marcações iniciais e atuais e as capacidades dos lugares.

Quando há um disparo de uma transição é feita a atualização das marcações dos lugares e dos valores das variáveis contínuas. As atualizações das variáveis continuas vão depender das funções de junção correspondentes.

Para confirmar se uma transição está ou não habilitada é necessário verificar as marcações da rede e também se as condições das funções de habilitação estão satisfeitas. A verificação das condições das funções de habilitação só se aplica nas transições das Redes de Petri Predicado/Transição Diferencial.

Apesar do simulador estar sendo desenvolvido para sistemas híbridos, observa-se que ele pode ser utilizado também para sistemas puramente discretos ou sistemas puramente contínuos.

4.1 Arquitetura geral do simulador

De acordo com o apresentado acima o simulador apresenta a arquitetura ilustrada na Figura 21:

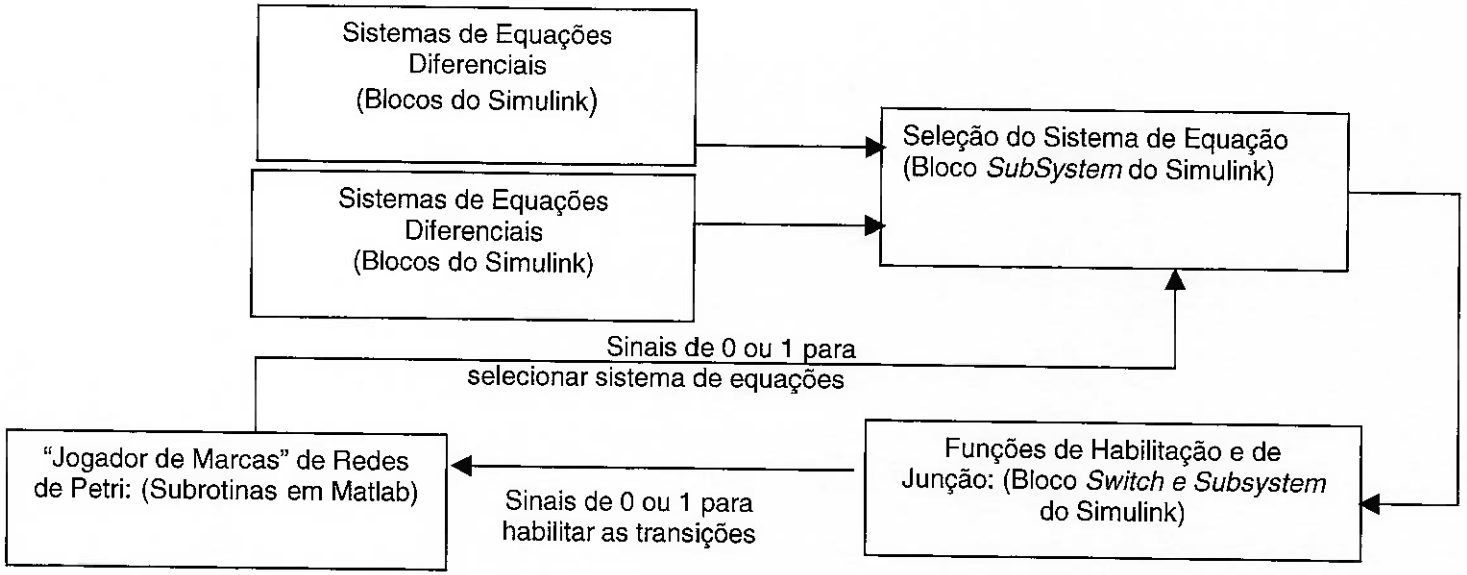


Figura 21 – Arquitetura geral do simulador

4.2 Arquitetura do "Jogador de Marcas" de Redes de Petri

No que se refere a implementação da parte discreta, a função principal – o "jogador de marcas" - está estruturada da seguinte forma - Figura 22:

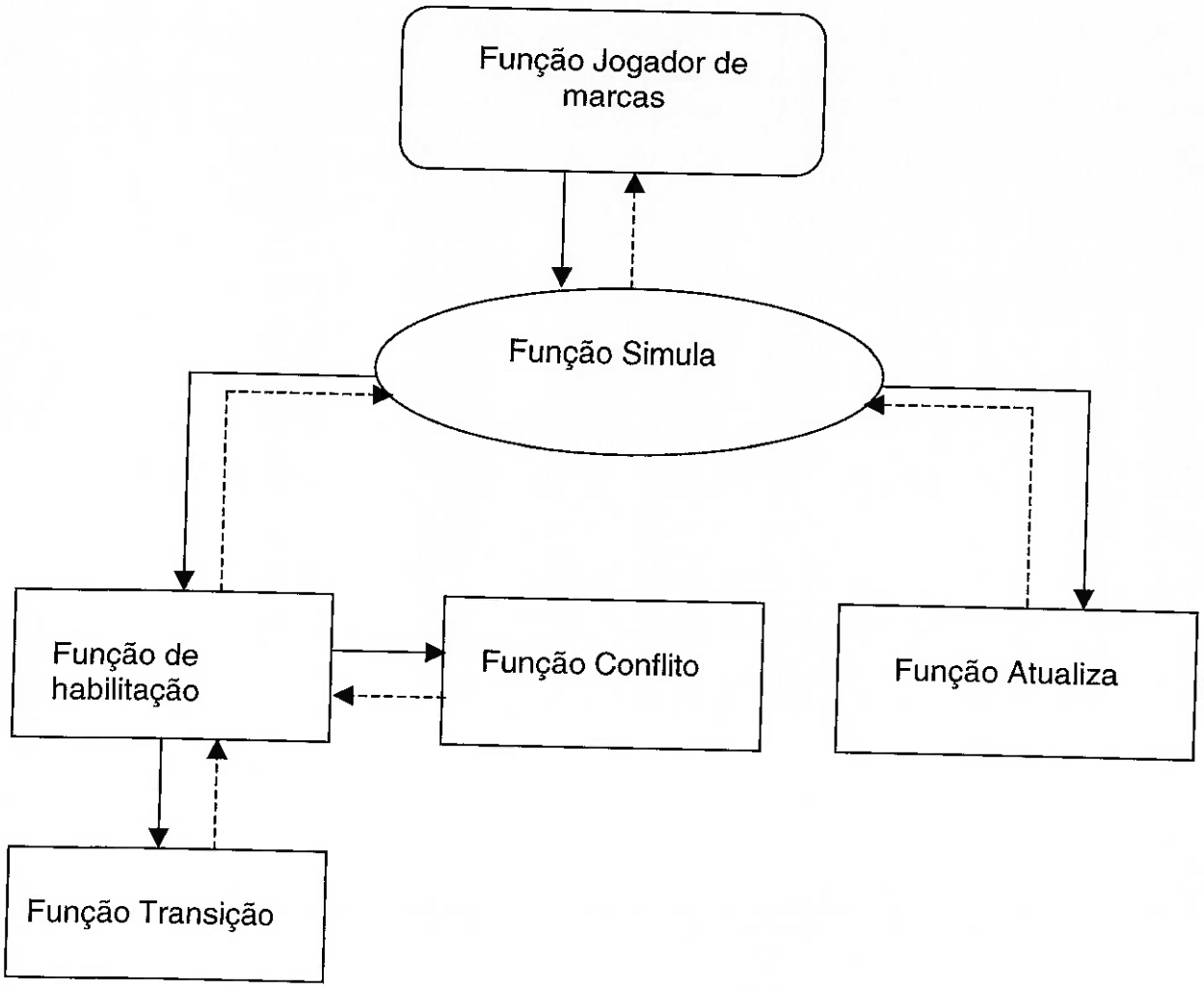


Figura 22 – Arquitetura do Jogador de Marcas

A "função de habilitação" tem como objetivo detectar as transições que estão habilitadas e prontas para disparo. Para fazer isso, ela chama a "função transição" que verifica se as pré e pós condições de cada transição são ou não satisfeitas. "A função transição" também resolve os "conflitos", isto é, caso haja uma situação de conflito, a função escolherá aleatoriamente uma transição para ser disparada.

Após a verificação de todas as transições e determinado quais delas estão habilitadas, a função principal chama a "função atualiza" que dispara todas as transições habilitadas e atualiza o estado do sistema modelado.

A "função conflito" é chamada pela função habilitação com o objetivo de verificar se o sistema a ser modelado apresenta ou não situações de conflito. Caso existam situações de conflito eles deverão ser tratados dentro da "função transição".

4.3 Modelagem da parte discreta do sistema

Para iniciar o uso do simulador de sistema híbrido, primeiramente é necessário modelar o sistema usando a técnica de Redes de Petri Predicado/Transição Diferencial [Villani, 2000]

Uma vez modelada o sistema, a parte em Rede de Petri deve ser transformada de acordo com a representação matricial indicada no capítulo 3.1.2 e 3.1.3

Usa-se o programa Simulink para executar o modelo implementado.

4.3.1 Representações matriciais no Simulink.

Para as representações matriciais em Simulink será utilizada a forma vetorial, isto é, os dois primeiros elementos do vetor de algarismos representam a dimensão da matriz (linha x coluna) e posteriormente vêm os elementos da matriz na seqüência das linha. Por exemplo:

$$A = \begin{bmatrix} 1 & 6 & 11 & 16 \\ 2 & 7 & 12 & 17 \\ 3 & 8 & 13 & 18 \\ 4 & 9 & 14 & 19 \\ 5 & 10 & 15 & 20 \end{bmatrix}_{5x4}$$

A matriz A é uma matriz 5x4, e a forma vetorial para representa-la é da seguinte forma:

A= [5 4 1 6 11 16 2 7 12 17 3 8 13 18 4 9 14 19 5 10 15 20].

Para um elemento nulo, por exemplo $B = [0]$

A forma vetorial da sua representação é B=[1 1 0]

4.3.2 Dados necessários a serem fornecidos.

Para iniciar o uso do simulador, é necessário fornecer os seguintes dados:

1. Marcação inicial;
2. Matriz de pré condição sem arcos habilitador e inibidor;
3. Matriz de pós condição sem arcos habilitador e inibidor;
4. Marcação inicial considerando somente os arcos habilitadores e inibidores e o estado inicial dos sinais externos
5. Matriz de pré condição considerando somente os arcos habilitadores e inibidores (sessão 3.1.3);
6. Matriz de pós condição considerando somente os arcos habilitadores e inibidores (sessão 3.1.3);
7. Capacidade dos lugares

4.3.3 Marcação inicial

Uma vez que se tem a rede na forma matricial, inicia-se o uso do simulador com a definição das marcações iniciais da rede. Como a marcação de uma rede é variável e precisa ser atualizada a cada disparo das transições habilitadas, utiliza-se os bloco *Data Store Memory* (Biblioteca Signals & Systems),

Figura 23. A marcação inicial da rede é o valor inicial fornecido ao bloco funcional.

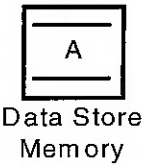


Figura 23 – Bloco *Data Store Memory*

Exemplo:

A marcação inicial de uma rede é $M_0 = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 2 \\ 0 \end{bmatrix}$

Passa-se a marcação inicial para a forma vetorial:

$M_0 = [\,5\,1\,1\,0\,0\,2\,0\,]$

A Figura 24 abaixo mostra os parâmetros a serem fornecidos ao bloco *Data Store Memory* para a definição da marcação inicial.

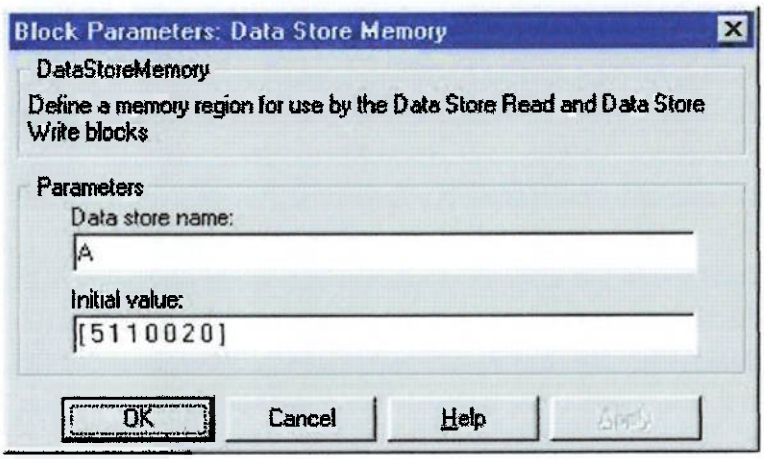


Figura 24 – Fornecendo as marcações iniciais

Para atualizar as marcações será utilizado o bloco *Data Store Write* (Biblioteca Signals & Systems), ilustrado na Figura 25 abaixo.

Figura 25 – Bloco *Data Store Write*

E para fazer a leitura dos dados do bloco *Data Store Memory* usa-se o bloco *Data Store Read* (Biblioteca Sings & Systems), ilustrado na Figura 26 abaixo.

Figura 26 – Bloco *Data Store Read*

A metodologia de fornecimento de marcações iniciais aplica-se também para indicar a marcação inicial dos lugares da rede dos quais originam-se arcos habilitadores e inibidores e o estado inicial dos sinais externos.

4.3.4 Matrizes de pré e pós condição.

Tendo criado os blocos que contêm as marcações iniciais, a próxima etapa é definir as matrizes de pré e pós condição.

Como as matrizes de pré e pós condição representam a estrutura do sistema e não seu estado, elas não são alteradas no decorrer da simulação, portanto usa-se o bloco *Constant* (Biblioteca Sources) para definição das matrizes,Figura 27.

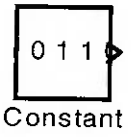


Figura 27 – Bloco *Constant*

Para fornecer os valores das matrizes, deve-se primeiro transforma-las para representação vetorial como indicado na sessão 4.3.1

4.3.5 Matrizes pré e pós considerando somente arcos habilitadores e inibidores e vetor marcação inicial considerando somente arcos habilitadores e inibidores e o estado inicial dos sinais externos.

Tanto o vetor marcação inicial considerando somente os arcos habilitadores e inibidores e o estado inicial dos sinais externos quanto as matrizes de pré e pós considerando-se somente arcos habilitadores e inibidores são criados de forma análoga à marcação inicial da rede, citada na sessão 4.3.3, e as matrizes de pré e pós condição, citada no sessão 4.3.4.

4.3.6 Capacidade dos lugares

Para definir a capacidade dos lugares é utilizada o bloco *Constant* (Biblioteca Sources). Os dados a serem fornecidos também seguem as mesmas regras para a representações vetoriais citadas na sessão 4.3.1

Não é permitido lugares com capacidades nulas, a capacidade mínima é unitária e não existe um limite superior.

Não é necessário a definição das capacidades dos pseudo-boxes que modelam os arcos habilitadores e inibidores. Estes por definição têm capacidade unitária [MATSUSAKI,1998].

4.3.7 *Organização dos dados de entrada*

Tendo definido todos os dados de entrada, a próxima etapa é colocá-los numa seqüência definida pelo simulador antes de iniciar o uso. Para juntar e organizar todos os dados é utilizado o bloco *Mux* (Biblioteca Signals & Systems), ilustrado na Figura 28 abaixo:

Figura 28 – Bloco *Mux*

A seqüência dos dados está ilustrado na Figura 29 abaixo:

1. Marcação inicial dos lugares;
2. Matriz de pré condição;
3. Matriz de pós condição;
4. Vetor da marcação inicial considerando-se somente os arcos habilitadores e inibidores e o estado inicial dos sinais externos;
5. Matriz de pré condição da rede considerando-se somente arco habilitadores e inibidores;
6. Matriz de pós condição da rede considerando-se somente arco habilitadores e inibidores;
7. Capacidade dos lugares (Excluindo os pseudo-boxes).

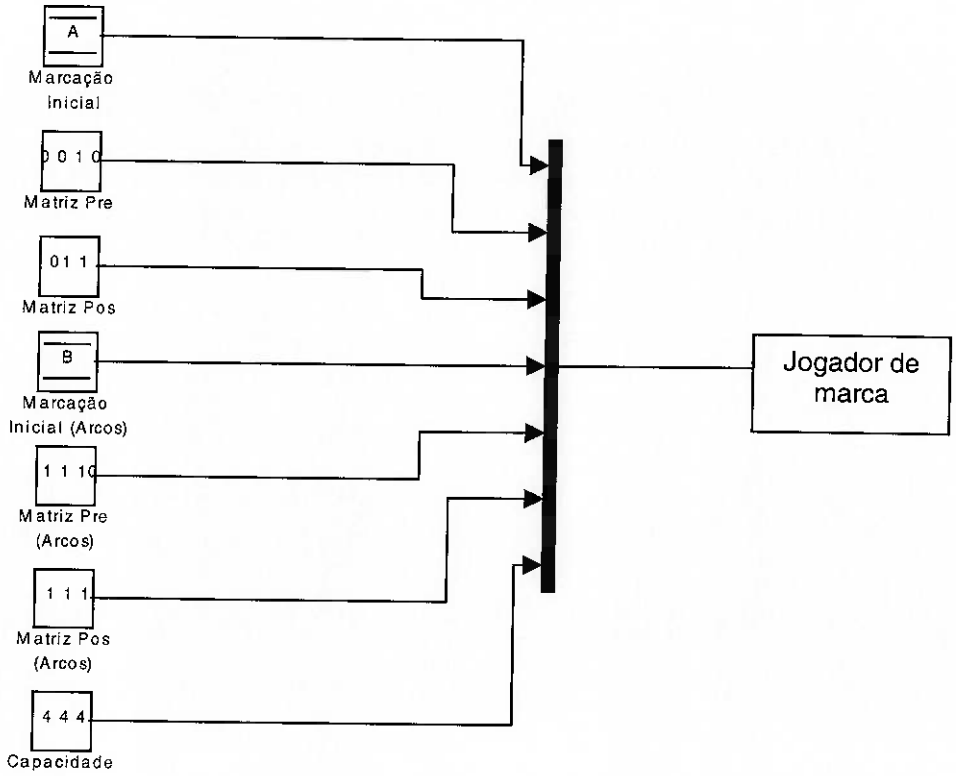


Figura 29 – Organização e seqüência dos dados de entrada

4.3.8 "Jogador de marcas"

O "jogador de marcas" é implementado em *linguagem m* e é encapsulado no bloco *Matlab Function* (Biblioteca Functions & Tables). O bloco é ilustrado na Figura 30

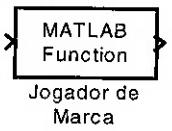


Figura 30 – Bloco *Matlab Function*

A configuração do bloco é a seguinte:

- A função a ser utilizada é a função *Jogador.m*

- O tamanho do vetor de saída é igual ao número de lugares da rede. Por exemplo, numa rede de 5 lugares, o tamanho do vetor saída do bloco é de dimensão 5.

Para exemplificar melhor, a configuração do bloco está ilustrada na Figura 31 abaixo, supondo que a rede a ser simulada tem 12 lugares.

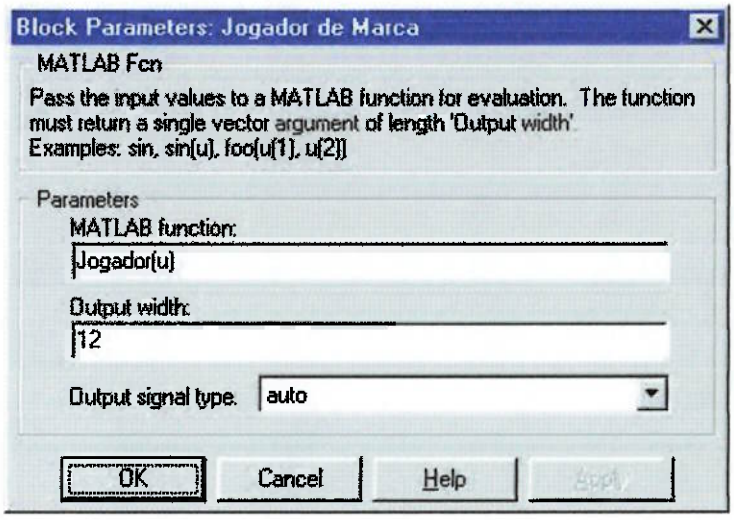


Figura 31 - Parâmetros do bloco *Matlab Function*- Jogador de marcas

Uma vez configurado o bloco *Matlab Function*, deve-se ligar a saída do bloco multiplexador, que reúne os dados de entrada em um único vetor, à entrada do "jogador de marcas". Este procedimento está representado esquematicamente na Figura 32 da página a seguir:

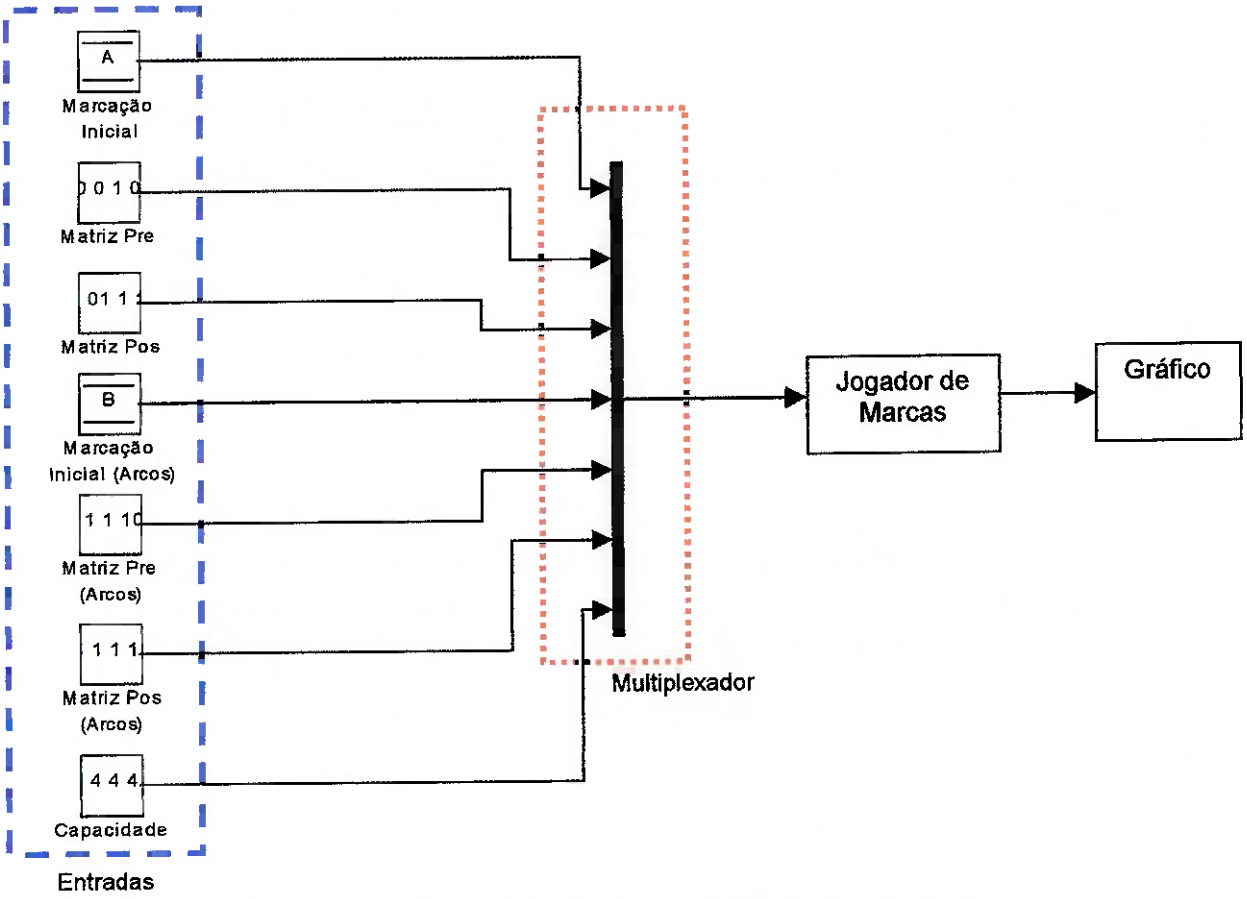


Figura 32 – Esquema de ligação do "Jogador de Marcas"

A saída do "jogador de marcas" pode ser conectado a um bloco de leitura de dados, tais como *Scope* (Biblioteca Sink), *Display* (Biblioteca Sink) ou *To File* (Biblioteca Sink). A escolha dos blocos a serem utilizados é a critério do usuário, no esquema da Figura 32 foi utilizado o bloco *Scope*.

4.3.9 Atualização das marcações

Para atualização das marcações utiliza-se o bloco *Data Store Write* (Biblioteca Signals & Systems) conforme mencionado na sessão 4.3.1. Juntamente com o bloco, é necessário a utilização de um bloco *Matlab Function* (Biblioteca Functions & Tables) para transformar a saída do "jogador de marcas" em um vetor com as características mencionadas na sessão 4.3.1.

A função Matlab encapsulada é *Transforma.m*. A saída da função *Transforma.m* é um vetor cujo tamanho é igual a *"tamanho de entrada"* + 2. Os dois elementos acrescentados são as dimensões da entrada. Por exemplo, o vetor de entrada tem dimensão 5, o vetor de saída terá dimensão 7 e a configuração do bloco está mostrada na Figura 33 a seguir.

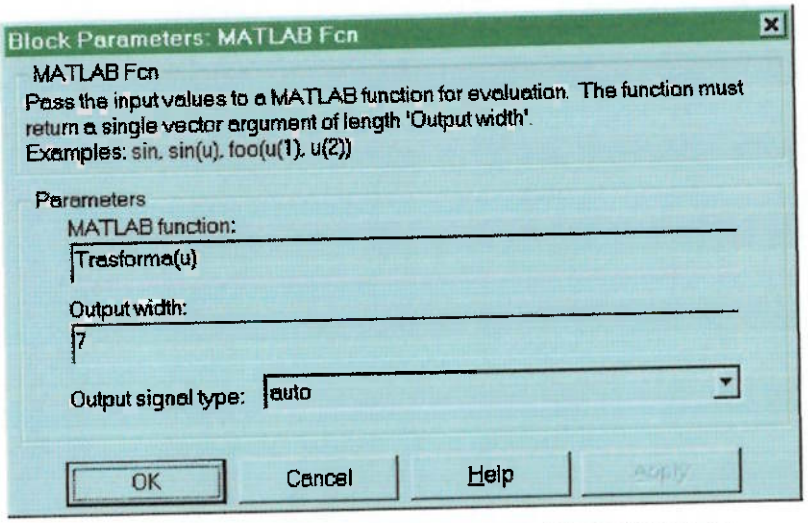


Figura 33 - Configuração da função *Transforma.m*

Por exemplo, a entrada é um vetor v = [1 0 0 1 5], o vetor de saída será v_s=[1 5 1 0 0 1 5].

Tendo configurado o bloco da *Transforma.m*. Agora deve-se ligar o bloco à saída do "jogador de marcas", e a saída do bloco *Transforma.m* ao bloco de memória. A Figura 34 abaixo esquematiza a utilização deste bloco e a atualização das marcações.

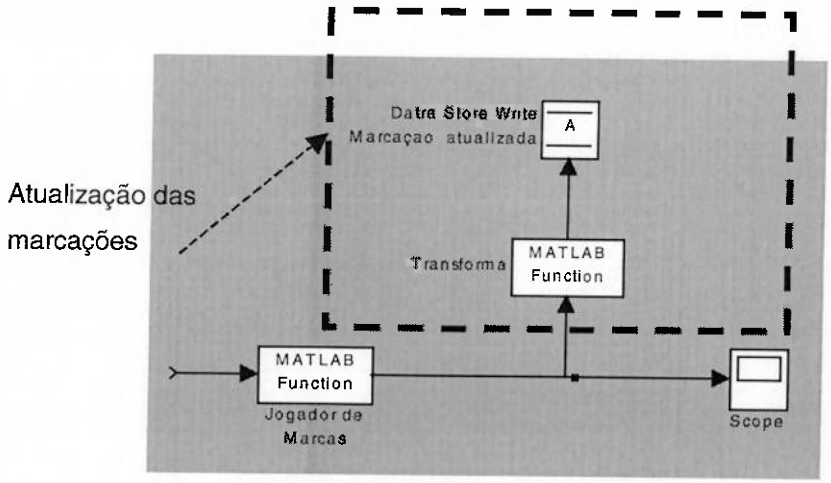


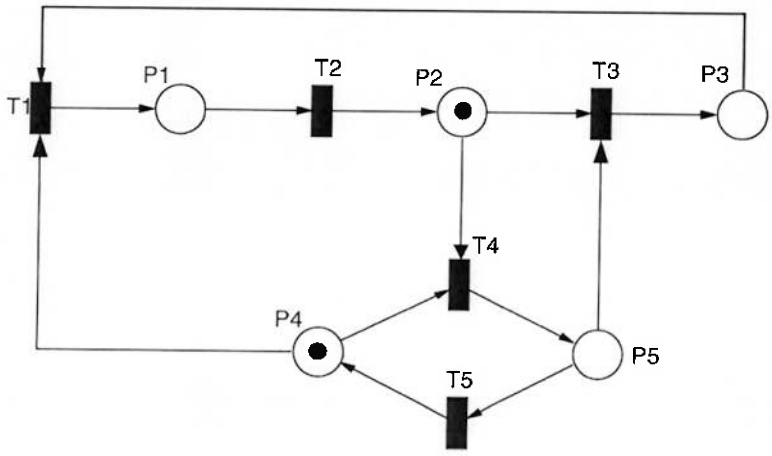
Figura 34 - Esquema de atualização das marcações

4.3.10 Exemplo de montagem um modelo

Tendo definidos todos os elementos necessários, deve-se agora conectar os blocos antes de iniciar a simulação.

A seguir apresenta-se um exemplo de modelagem usando os blocos definidos de acordo com as regras citadas na sessões anteriores.

O modelo considerado é ilustrado abaixo.



As matrizes de pré e pós condição da rede são:

$$A\ (\text{matriz de pré condição}) = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 \end{bmatrix}$$

$$B\ (\text{matriz de pós condição}) = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$

$$\text{Marcação inicial - } M_0 = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 1 \\ 0 \end{bmatrix}$$

$$\text{Capacidade} = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}$$

Como não há arcos habilitadores nem inibidores, as matrizes A' e B' que representam o modelo da rede considerando-se somente os arcos e os pseudo-boxes têm valores nulos, e o vetor das marcações iniciais considerando-se somente os arcos habilitadores e inibidores e o estado inicial dos sinais externos (M_0') para a rede com os arcos também é nula.

$$A' = B' = M_0' = [0]$$

Tendo definidos os parâmetros, agora deve-se montá-los na seqüência e passá-los para a função "jogador de marcas" como foram citados nas sessões anteriores.

Transformando os parâmetros na representação vetorial citado na sessão 4.3.1 tem-se:

A = [5 5 0 1 0 0 0 0 0 1 1 0 1 0 0 0 0 1 0 0 1 0 0 0 1 0 1];

B = [5 5 1 0 0 0 0 0 1 0 0 0 0 0 1 0 0 0 0 0 0 1 0 0 0 1 0];

M_0 = [5 1 0 1 0 1 0]

Capacidade = [5 1 1 1 1 1 1]

A' = B' = M_0' = [1 1 0]

As configurações dos blocos funcionais serão:

1. Marcações iniciais

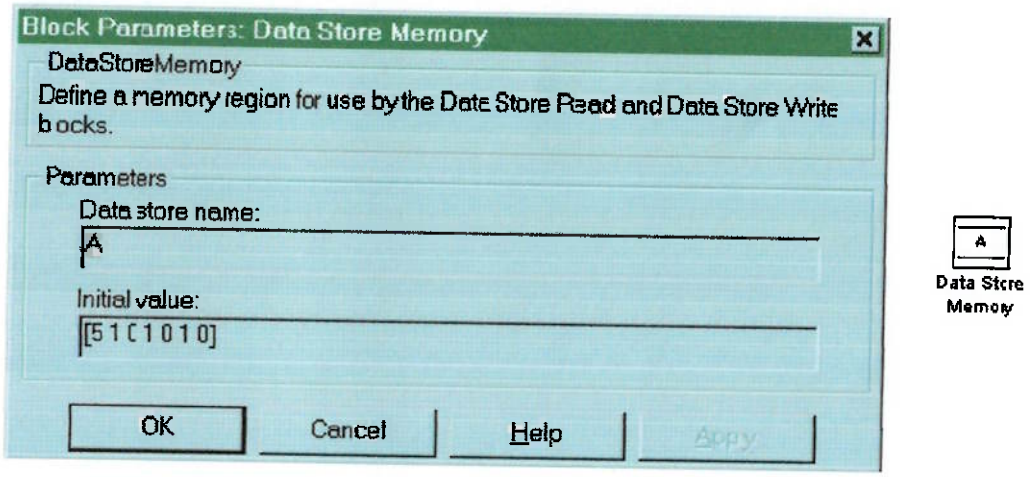


Figura 35 - Marcação inicial

2. Matrizes de pré e pós condição

Figura 36 - Matriz de pré condição

Figura 37 – Matriz de pós condição

3. Capacidade

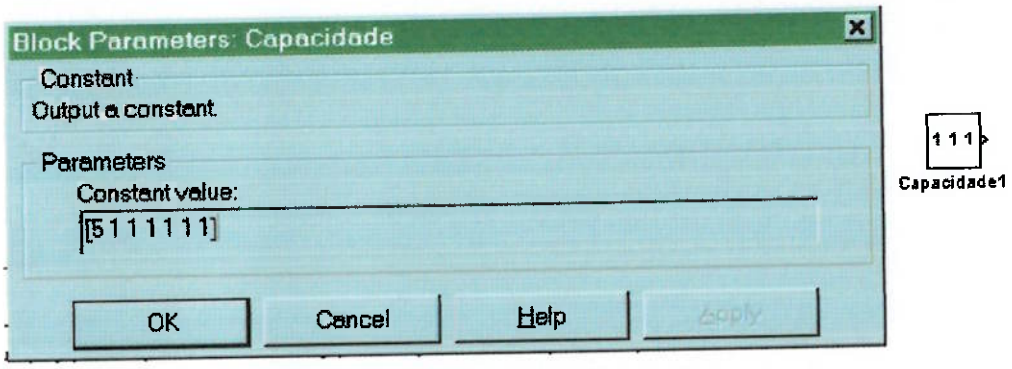


Figura 38 – Capacidade dos lugares

4. Marcação considerando-se somente os arcos habilitadores e inibidores e o estado inicial dos sinais externos

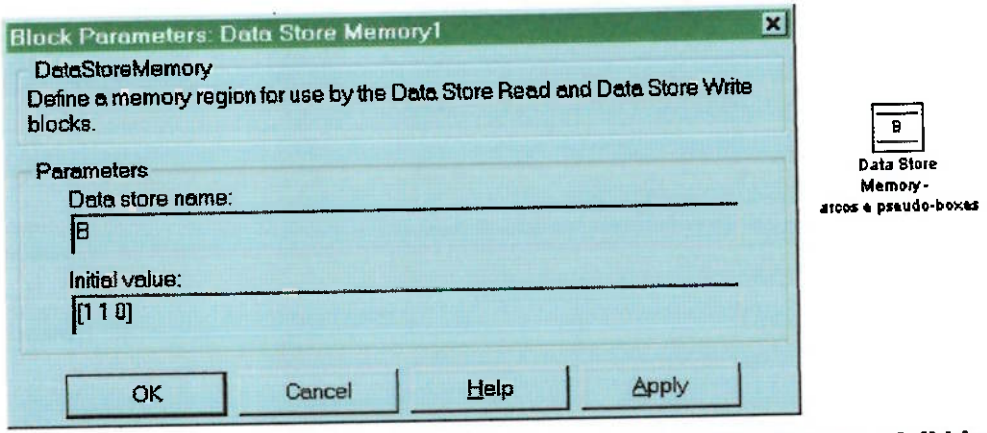


Figura 39 – Marcação considerando-se somente os arcos habilitadores e inibidores e o estado inicial dos sinais externos

No caso deste exemplo, a marcação dos lugares originário dos arcos habilitadores e inibidores e o estado inicial dos sinais externos pode ser representada pelo bloco funcional constante, já que não haverá variação da marcação.

5. Matrizes de pré e pós condição da rede considerando-se somente os arcos habilitadores e inibidores

A matriz de pré e pós condição da rede incluindo ambos os arcos é definida da mesma forma que as matrizes pré e pós condição da rede. A única diferença é o valor atribuído ao bloco funcional.

6. "Jogador de marcas"

A configuração do bloco funcional que encapsula o "jogador de marcas" está mostrada na Figura 31 acima. O parâmetro a ser mudado do bloco é a dimesão da saída, no caso do exemplo é 5.

7. Função *Transforma.m*

A função *Transforma.m* também é configurada da mesma forma como foi mostrada na Figura 33 acima. O parâmetro da função a ser mudado também é a dimensão da saída. Neste exemplo a saída terá dimensão 7, que por coincidência é o mesmo da Figura 33

8. Saída do "jogador de marcas"

A saída do "jogador de marcas" pode ser conectado diretamente à entrada da função *Transforma.m*, entretanto no exemplo a saída será conectada a um bloco de display e a um bloco gráfico.

O modelo implementado está mostrado na Figura 40 abaixo.

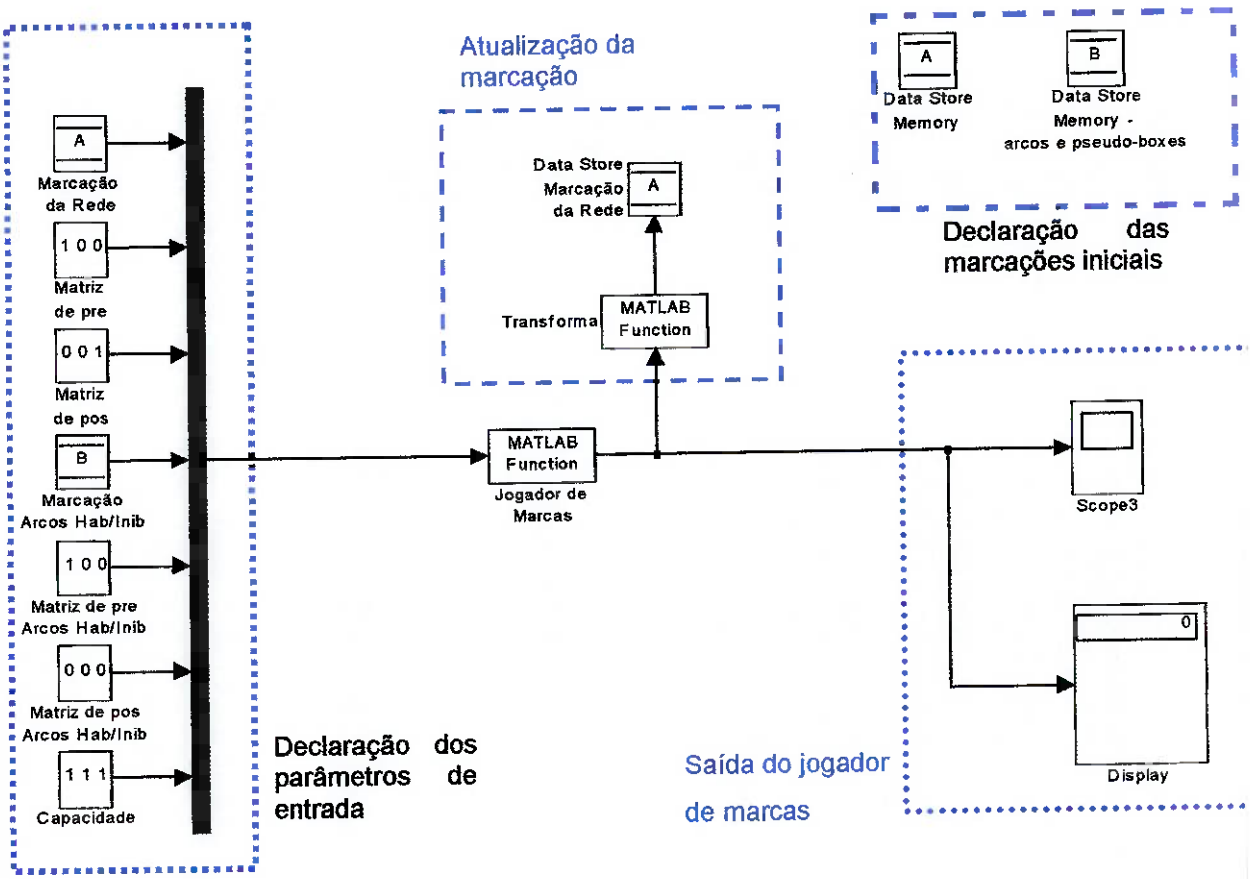


Figura 40 – Modelo implementado

4.4 *Modelagem da parte contínua do sistema*

A metodologia citada até agora refere-se a utilização do simulador para a parte discreta do modelo.

Para a parte contínua do modelo, a modelagem envolve o procedimento apresentado a seguir.

A saída do "jogador de marcas" é o sinal de habilitação de cada sistema de equações diferenciais. Para isso utiliza-se o bloco *Demux* (Biblioteca Signals & Systems) ilustrado na Figura 41 abaixo, que decompõe o vetor de marcação em sinais distintos:

Figura 41 – Bloco funcional *Demux*

Cada sinal de saída do bloco demultiplexador corresponde à marcação de um lugar. A seqüência é idêntica a nomenclatura dos lugares utilizados na Rede Predicado/Transição Diferencial. Para melhor entendimento vide Figura 42 abaixo:

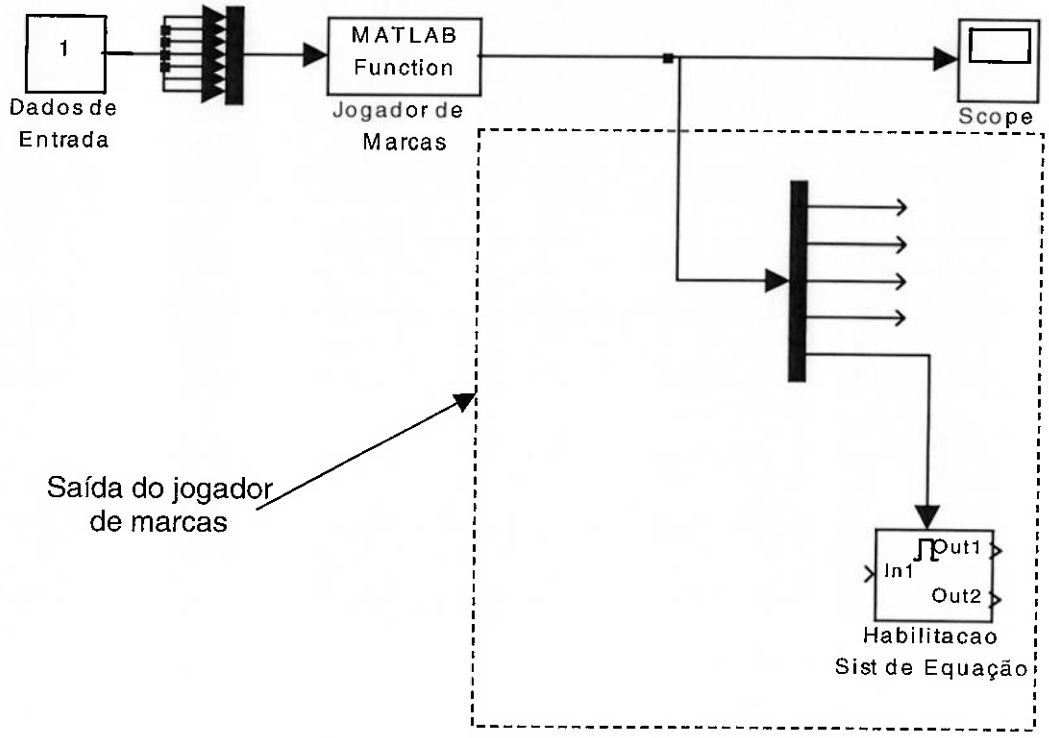


Figura 42 – Saída do "jogador de marcas" – habilitação de sistema de equações.

Para criar o lugar com o sistema de equações diferenciais da Rede Predicado/Transição Diferencial, é utilizado o bloco funcional *SubSystems* (Biblioteca Signals & Systems). O sistema de equações deve ser modelado na forma de diagrama de blocos. Para maiores detalhes é recomendável a consulta do [Matsumoto, 2001].

Para simular e atualizar os valores das variáveis do sistema de equações diferenciais, é necessário fornecer dados de entrada aos blocos funcionais que contem o sistema. Esses valores são definidos dentro de cada subsistema que modela as equações diferenciais ou derivam das transições que apresentam função de junção.

O sinal de habilitação do subsistema, como mencionado anteriormente, vem da saída do "jogador de marcas". Quando o sinal muda de 0 para 1, isto significa que a marcação do lugar passa de 0 para 1, então lugar está marcado e deve-se iniciar a simulação do sistema de equações diferenciais nele contido.

O subsistema, modelado pelo lugar com o sistema de equações diferenciais, devolve 0 ou 1 como resultado das funções de habilitação.

Resumindo, a modelagem do lugar com o sistema de equações diferenciais pode ser representado pela Figura 43

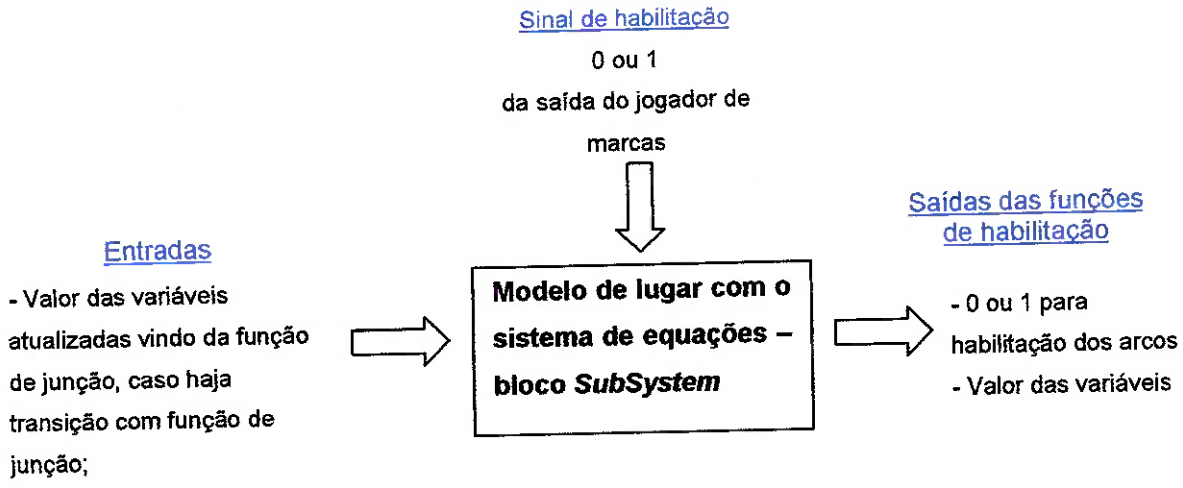


Figura 43 – Esquemático do modelo do lugar com sistema de equações diferenciais

4.4.1 Elementos do modelo do contínuo

Nesta sessão os sistemas de equações diferenciais associados a lugares serão denominados de subsistemas, já que na modelagem deles é usando o bloco funcional *SubSystem*.

1. Variáveis contínuas

As variáveis contínuas são armazenadas de forma análoga às marcações da Rede de Petri. Utiliza-se o bloco *Data Store Memory* com um valor inicial, o bloco *Data Store Read* para acesso ao valor, e o bloco *Data Store Write* para a atualização do valor das variáveis.

2. Habilitação do sistema

Para a habilitação do sistema, deverá ser utilizado o bloco *Enable* (Biblioteca Signals & Systems). Este bloco deve ser adicionado dentro do bloco *SubSystem*.

Toda vez que habilitar o subsistema os valores das variáveis podem ser "*resetas*" ou mantidas. Isto dependerá do modelo simulado.

3. Entrada e saída do subsistema

Para que o subsistema tenham arcos de entrada e saída deverão introduzir os blocos *In* e *Out* (Biblioteca Signasl & Systems). Cada bloco *In* introduzido representa uma entrada do subsistema, e cada bloco *Out* do subsistema, isto é se o subsistema tiver 5 blocos *In* e 4 blocos *Out* então ele terá 5 entradas e 4 saídas.

4. Funções de habilitação

Para implementar as funções de habilitação será utilizado o bloco *Switch* (Biblioteca Nonlinear).

Os valores das variáveis simuladas serão lidas e avaliadas ou pelo bloco *Function* (Biblioteca Functions & Tables) encapsulando a função pré definida de habilitação ou por outros blocos funcionais tais como *Logial Operator* (Math) ou *Relation* Operator (Biblioteca Math).

As saídas desses blocos serão conectadas ao bloco *Switch* e este selecionará o valor (0 ou 1) a ser devolvido pelo subsistema para o simulador discreto de Redes de Petri como sinal externo.

A Figura 44 abaixo ilustra a metodologia acima apresentada.

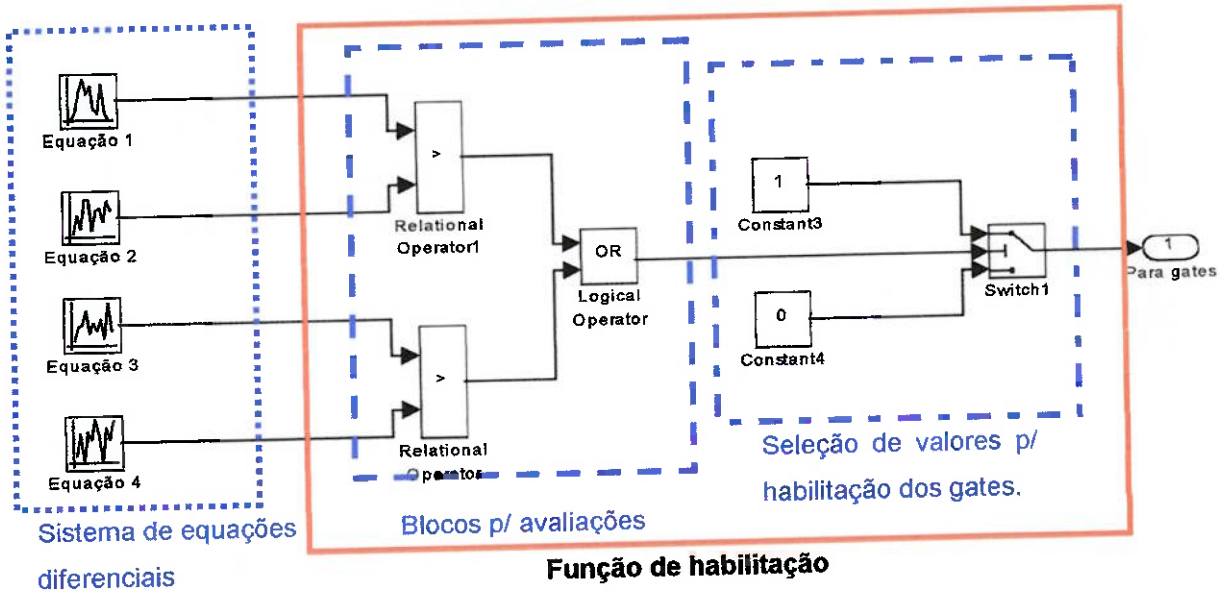


Figura 44 – Modelagem da função de habilitação

5. Sistema de equações diferenciais.

Não há uma metodologia específica para a implementação dos sistemas de equações diferencias. Os blocos funcionais utilizados na montagem de um sistema de equações variam de modelo para modelo Recomenda-se a leitura do [Matsumoto, 2001]. Para citar um exemplo será adotado o seguinte sistema de equações diferenciais:

$$\dot{x}_1 = -1$$
$$\dot{x}_2 = -0.1 * x_1 + 1$$

As condições iniciais para x1 é 0 e para x2 é 1.

Para montagem deste sistema serão utilizados os seguintes blocos:

- *Constant*: valores constantes e valores iniciais
- *Function*: definição da operação $-0.1*x_1+1$;
- *Sum:* Bloco somador para efetuar somas.
- *Integrator*: Integrador para fornecer valor de x_1 e x_2

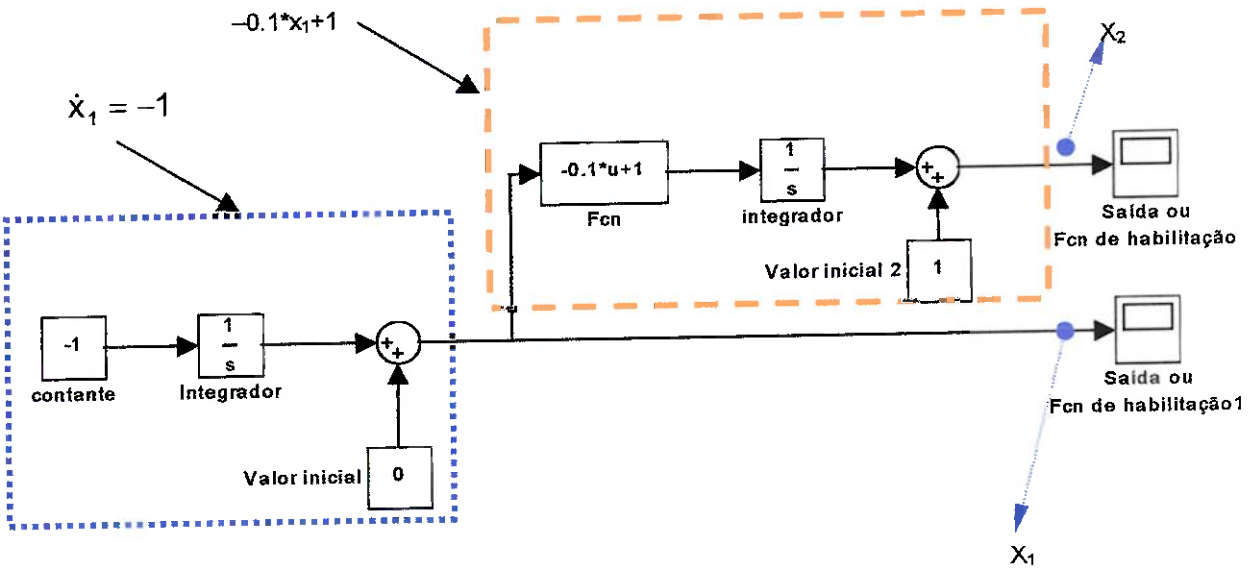


Figura 45 – Sistema de equações diferenciais

6. Função de junção

A função de junção atualiza os valores das variáveis quando a transição contínua habilitada dispara.

Para a modelagem da função de junção utiliza-se dois blocos de subsistemas. O primeiro dele atualiza os valores e o segundo guarda os valores atualizados e posteriormente fornece-os para o subsistema de equações diferenciais.

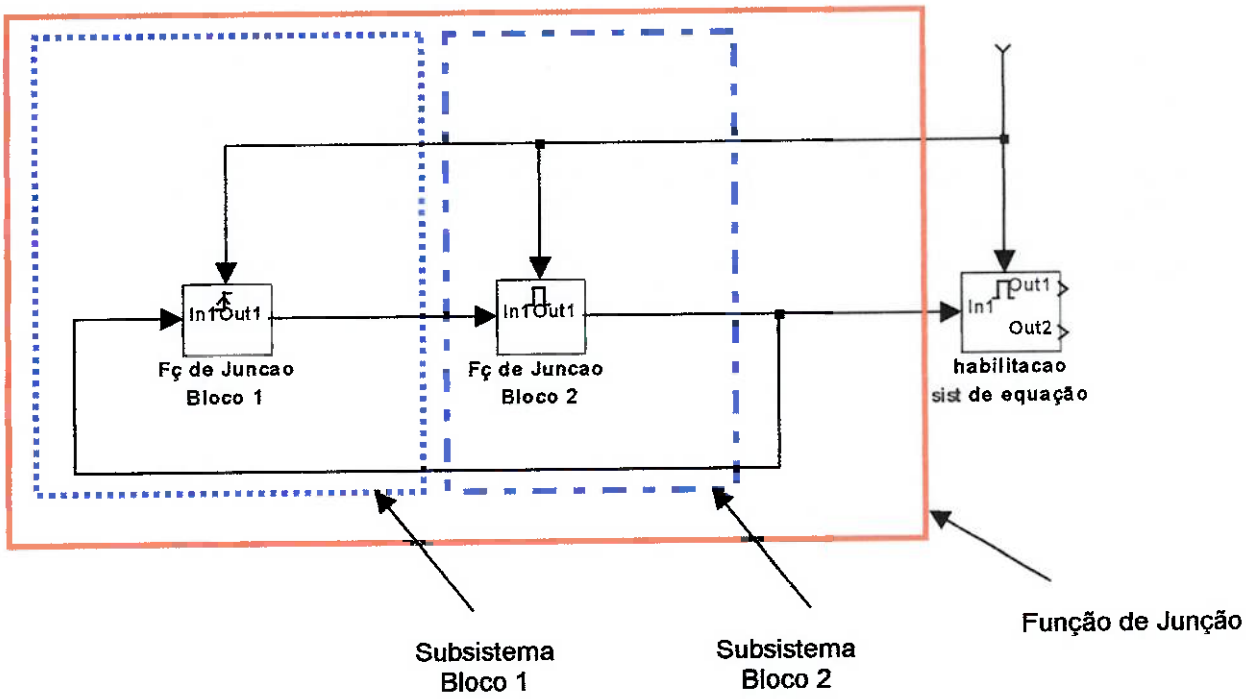


Figura 46 – Função de Junção

No primeiro subsistema a função de junção é habilitada pela mudança dos sinais das marcações. Quando o sinal do lugar passa de 0 para 1, isto representa que a transição da função de junção foi disparada, conseqüentemente as variáveis contínuas devem ser atualizadas. Para fazer o subsistema disparar com a mudança de sinal usa-se o bloco habilitador *Trigger* (Biblioteca Signals & Systems). Com o bloco *Trigger* o bloco modelo da função de junção é ativado por borda de subida (quando o sinal passa de 0 para 1).

Dentro do 1° subsistema da função de junção devem estar as funções para atualizar as variáveis.

As entradas do primeiro bloco de subsistema são as saídas do bloco do subsistema de equações diferenciais que modela o lugar à entrada da transição com função de junção. Caso a função de junção não realize operação alguma com as variáveis, não é necessário definir uma entrada para o primeiro bloco.

O bloco 2 tem a função de guardar os valores das variáveis atualizadas e posteriormente fornecê-las para o subsistema das equações diferenciais e o bloco 1 da função de junção.

4.5 Exemplo

Para ilustrar a metodologia, será considerada a Rede de Predicado/Transição Diferencial abaixo.

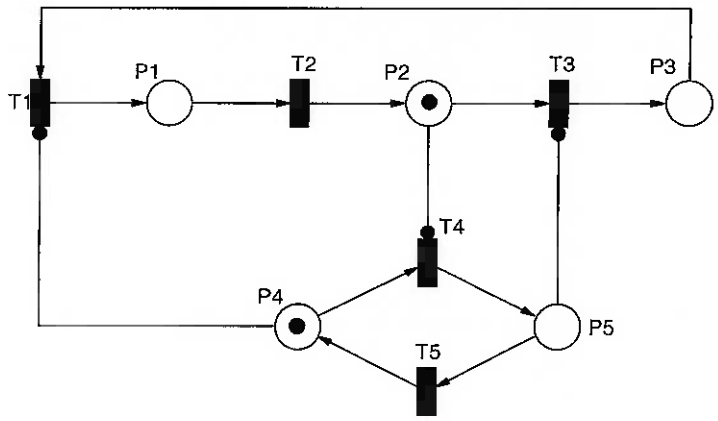


Figura 47 –Exemplo de Rede Predicado/Transição Diferencial

Variáveis contínuas para a rede: x_1 e x_2

Condições Iniciais: $x_1 = 0$, $x_2=1$;

Sistema de Equações de P4:

$$\dot{x}_1 = -1$$
$$\dot{x}_2 = -0.1 * x_1 + 1$$

Função de habilitação de T4:

$x_2 > x_1$

Função de junção de T4:

$x_1 = 0$

$x_2 = 0$

Sistema de Equações de P5:

$$\dot{x}_1 = 2$$
$$\dot{x}_2 = 0.1 * x_1 + 1$$

Função de habilitação de T5:

$x_2 > x_1$

Função de junção de T5:

$x_1 = x_1 + 2$

O modelo implementado é ilustrado abaixo:

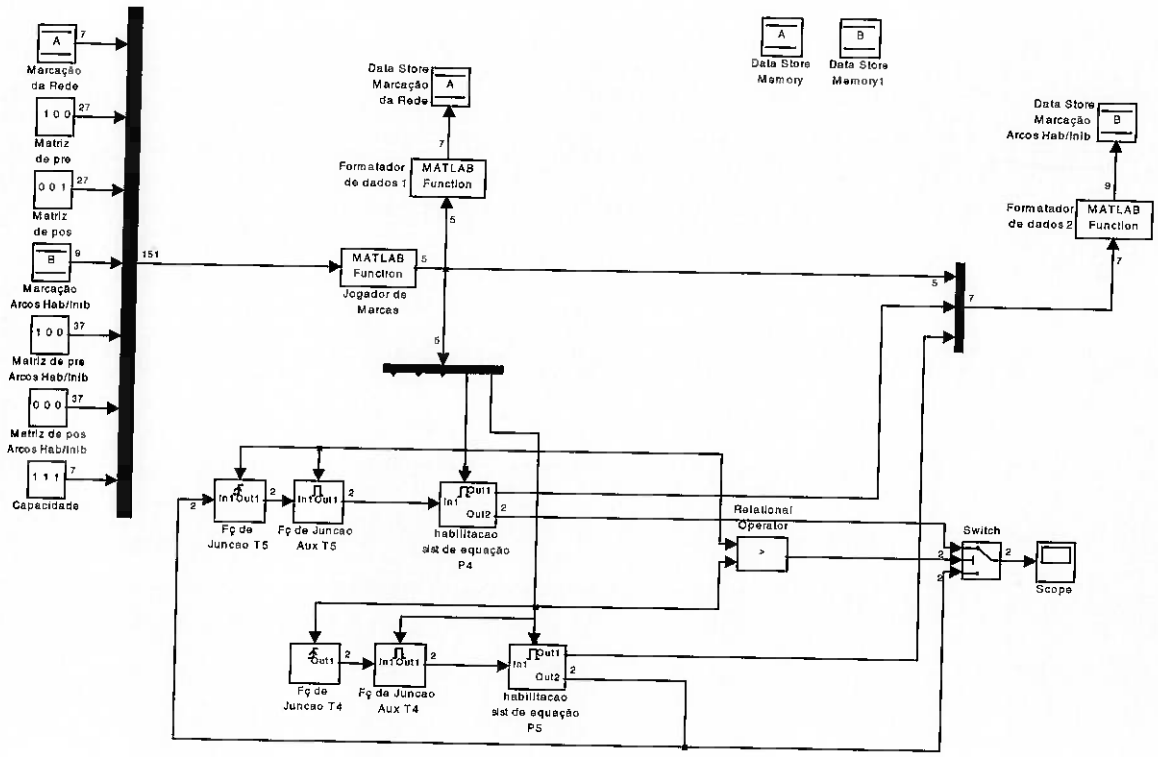


Figura 48 – Modelo implementado de rede da Figura 47

Subsistema bloco 1 da função de junção de T5

$x_1=x_1 + 2$

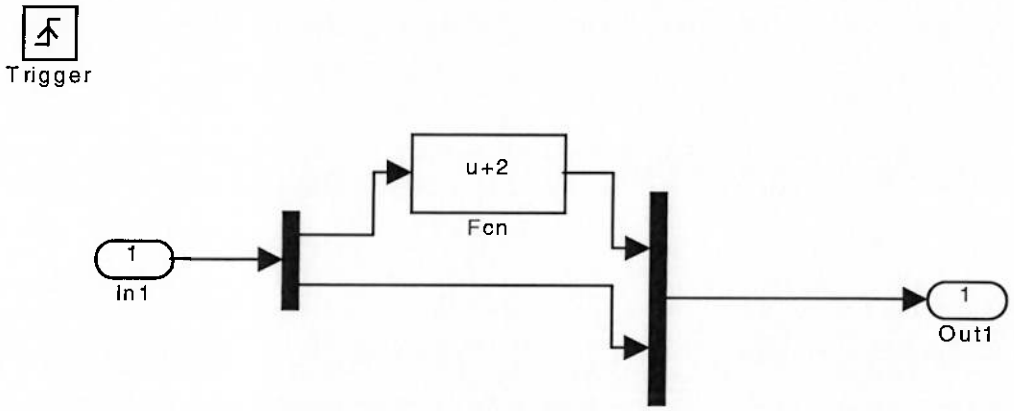


Figura 49 –Subsistem bloco 1 da função de junção de T5

Subsistema bloco 2 da função de junção de T5

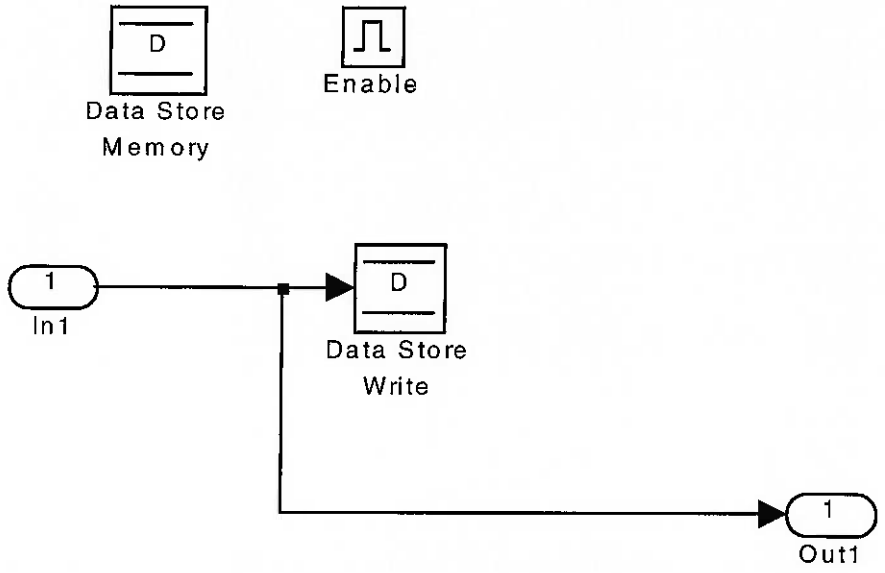


Figura 50- Subsistema bloco 2 da função de junção de T5

Subsistema bloco 1 da função de junção de T4

$x_1 = 0$ e $x_2 = 0$

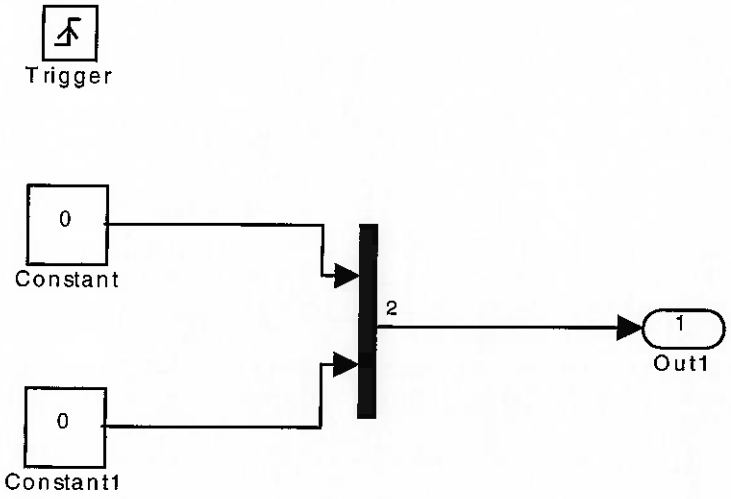


Figura 51 – Subsistema bloco 1 da função de junção de T4

Subsistema bloco 2 da função de junção de T4

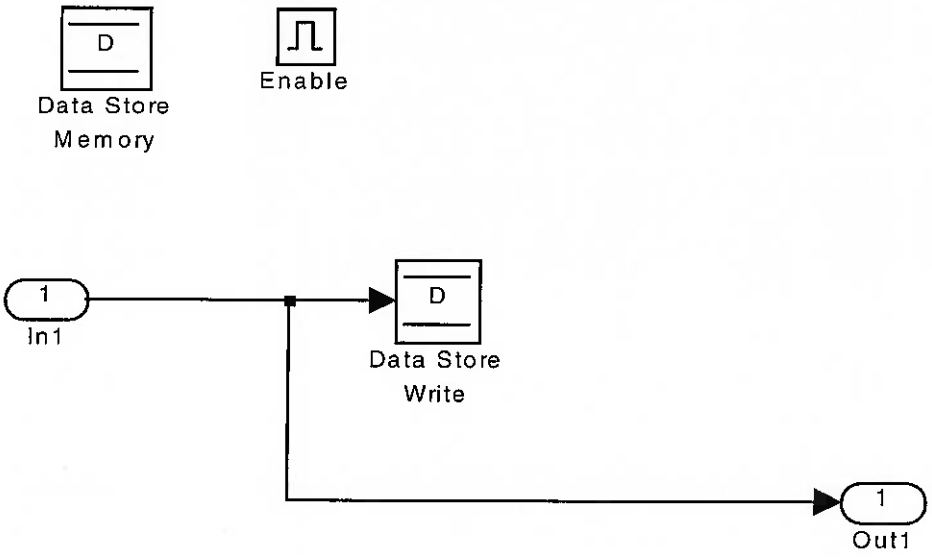


Figura 52 – Subsistema bloco 2 da função de junção de T4

Sistema de equações diferencias de P4 e a função de habilitação

Sistema de equações diferenciais: $\dot{x}_1 = -1$
$\dot{x}_2 = -0.1 * x_1 + 1$

Função de habilitação: $x_2 > x_1$

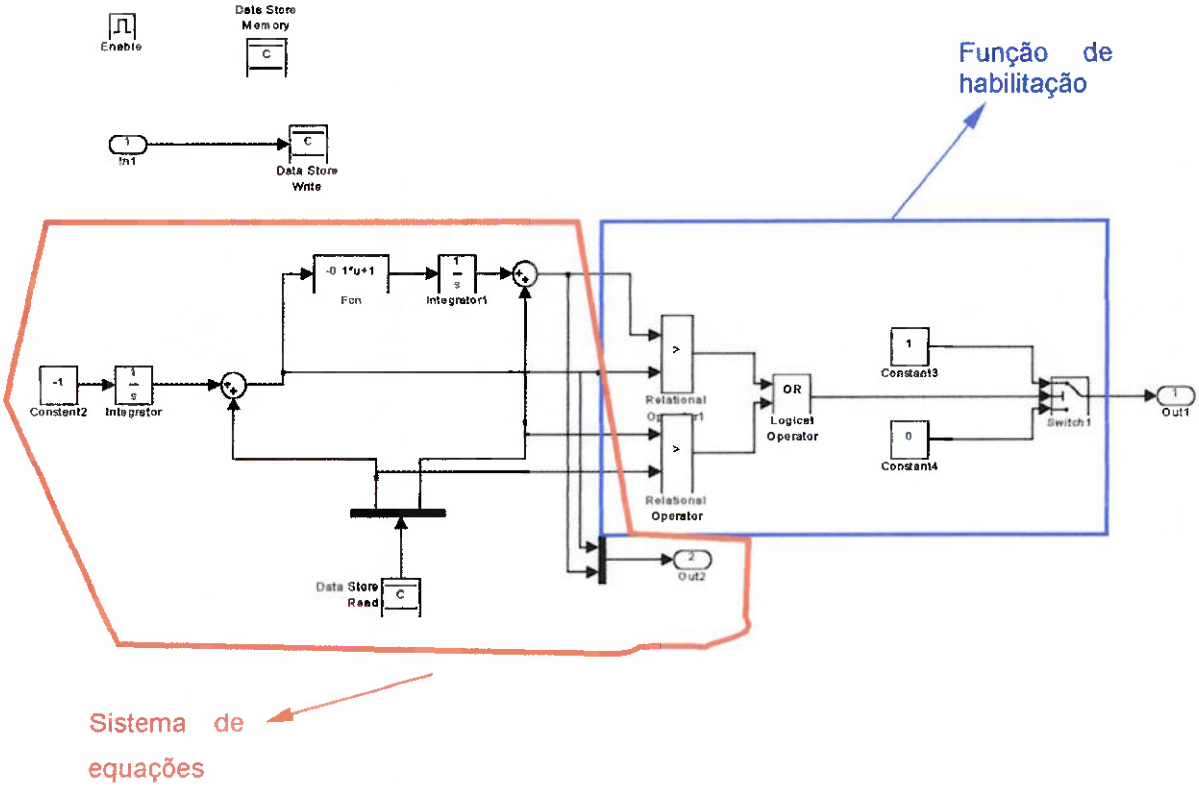


Figura 53 – Sistema de equações diferenciais de P4 e a função de habilitação de T4

Sistema de equações diferenciais de P5 e a função de habilitação

Sistema de equações diferenciais: $\dot{x}_1 = 2$
$\dot{x}_2 = 0.1 * x_1 + 1$

Função de habilitação: $x_2 > x_1$

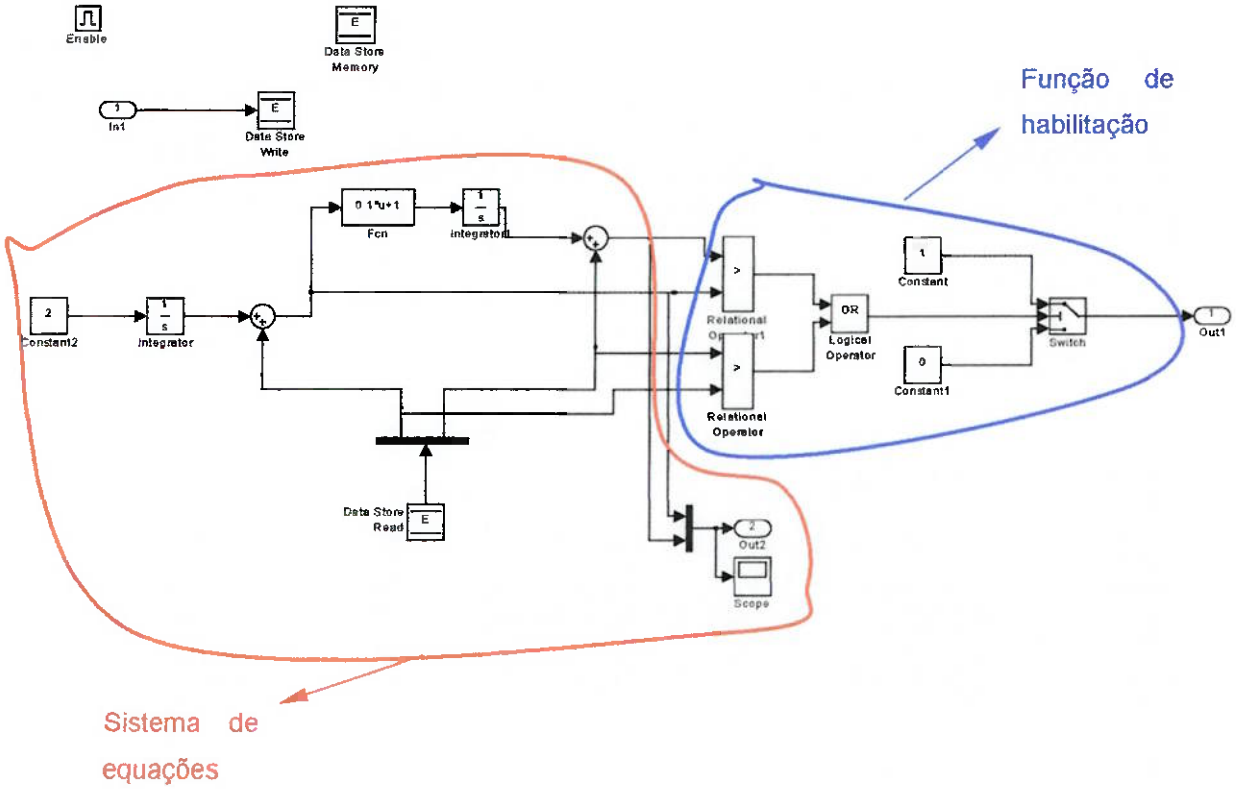


Figura 54 – Sistema de equações diferenciais de P5 e a função de habilitação de T5

5 Procedimento para simulação de sistemas híbridos

O procedimento considerado para a simulação de sistemas híbridos está esquematizada no fluxograma [Villani, 2000] da Figura 55:

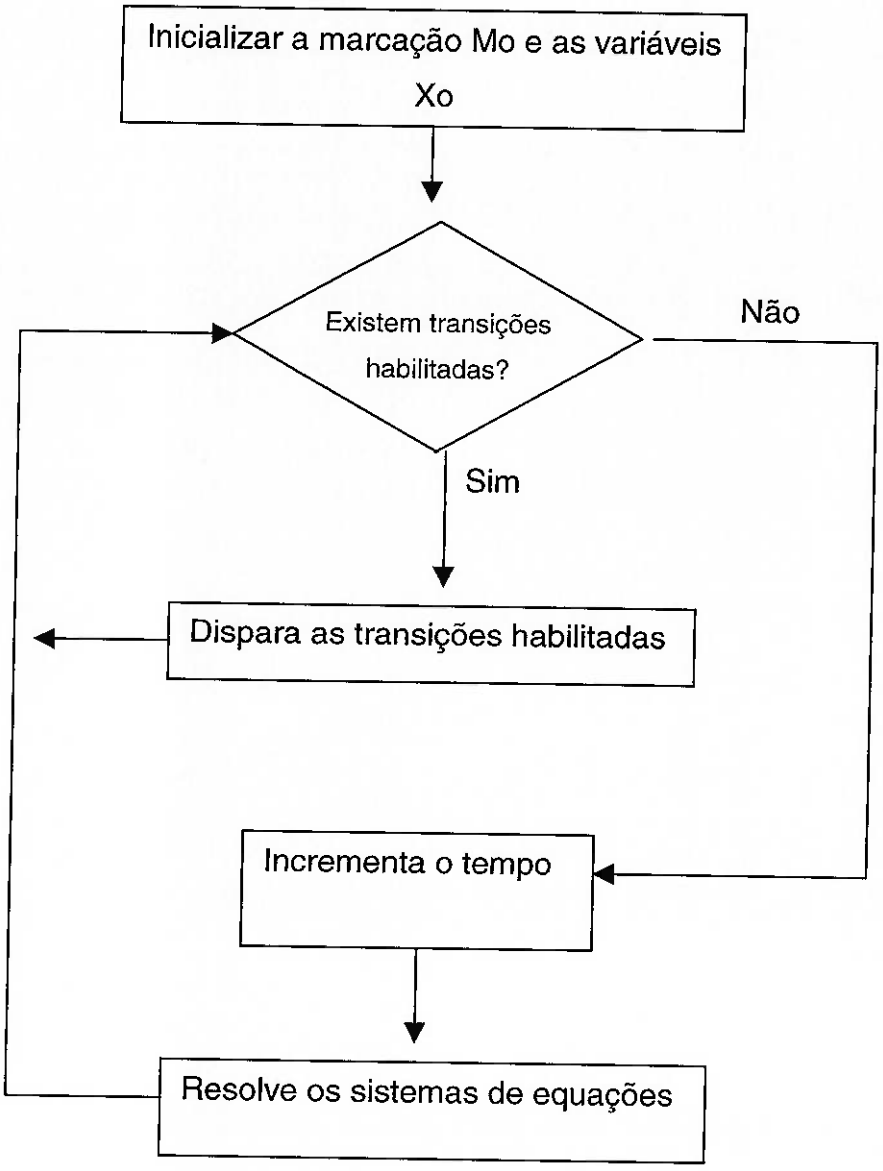


Figura 55 – Fluxograma para simulação de um sistema híbrido

A atualização dos estados dos arcos habilitadores e inibidores é feita pelo SIMULINK através da manipulação dos blocos funcionais.

Simulando o exemplo da sessão 4.5, os resultados são:

1-Disparo de T4,

2-Disparo de T3,

3-Simulação do sistema de P5 até T=12.6 (aproximadamente),

4-Disparo de T5,

5-Disparo de T1,

6-Disparo de T2,

7-Evolução do sistema de P4 até T=12.6+12.9 (aproximadamente)

8-Disparo de T4,

E assim por diante, repetindo a mesma seqüência.

O gráfico da Figura 56 representa a variação das variáveis x_1 e x_2 ao longo da simulação.

O x_1 está representada pela curva em azul e o x_2 em verde.

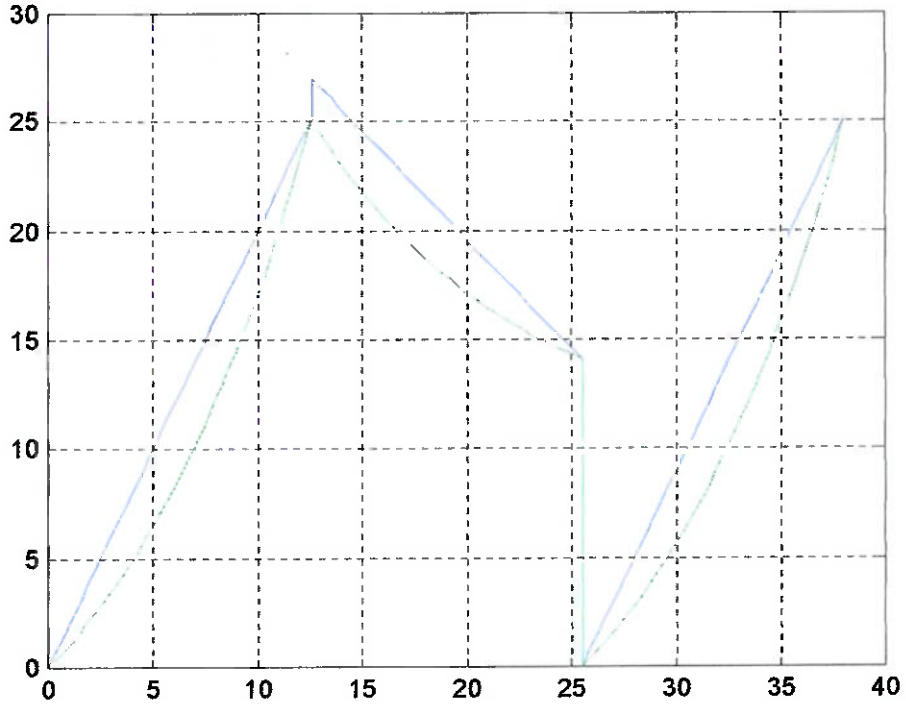


Figura 56 – Variação das variáveis ao longo do tempo

6 Aplicação do simulador em um estudo de caso

Como foi definido no Capítulo 0, a segunda parte do trabalho é a aplicação do simulador implementado a um estudo de caso.

O estudo de caso considerado é baseado no sistema de ar condicionado do Prédio dos Ambulatório (PAMB) do Hospital das Clínicas da Faculdade de Medicina da Universidade de São Paulo (HC – FMUSP), cujos modelos em Redes de Petri e em Redes de Petri Predicado-Transição Diferencial se encontram detalhados em [Villani, 2000]. Para facilitar a compreensão dos modelos desenvolvidos, utiliza-se uma versão simplificada do sistema de ar condicionado do PAMB.

6.1 O sistema original

O sistema de ar condicionado na PAMB apresenta funções tanto de aquecimento como de resfriamento. A água quente e fria são produzidas centralmente e distribuídas pelo edifício para o condicionamento do ar em cada zona. O PAMB possui cerca de 300 zonas. Para produção de água gelada o sistema possui 4 "chillers" e 8 torres de resfriamento. A água quente é produzida por 2 aquecedores. O transporte da água pelo edifício é realizado por 8 bombas.

Em cada zona o condicionamento do ar é realizado em "fan-coils". O ar é retirado do ambiente por ventiladores, é misturado com ar exterior, passa através da serpentina de resfriamento/aquecimento e é novamente insuflado no

ambiente. A obtenção da temperatura desejada no ambiente é realizada através da modificação da temperatura do ar que é insuflado no ambiente. Para tanto, uma válvula de 3 vias controla a passagem de água quente ou fria pela serpentina. Esta válvula é posicionada por um controlador PI (proporcional-integral).

Além do sistema de controle local das serpentinas, o sistema de controle proposto para o PAMB em [Villani, 2000] é formado também por um sistema de gerenciamento que é composto por diversas estratégias de controle acionadas de acordo com a necessidade. Algumas destas estratégias são aplicadas para controle dos equipamento de uma zona e outras para o controle dos equipamentos de água gelada ou quente.

6.2 O sistema adotado

Como foi explicado anteriormente o presente estudo de caso é uma simplificação do sistema original do PAMB. O sistema considerado possui as seguintes características:

- A produção de água gelada é realizada em 1 "chiller". O "chiller" troca de calor diretamente com o meio ambiente.
- Há uma bomba para transporte de água gelada do "chiller" para as serpentinas,
- O sistema não permite o aquecimento do ar.
- O edifício é dividido em 3 zonas.

Cada zona possui:

- Uma caixa de mistura;
- Um ventilador de retorno;
- Um ventilador de insuflamento;
- Uma serpentina;

O condicionamento do ambiente é realizado de forma análoga ao descrito para o sistema original.

Um esquema do sistema de ar condicionado do estudo de caso é apresentado na Figura 57

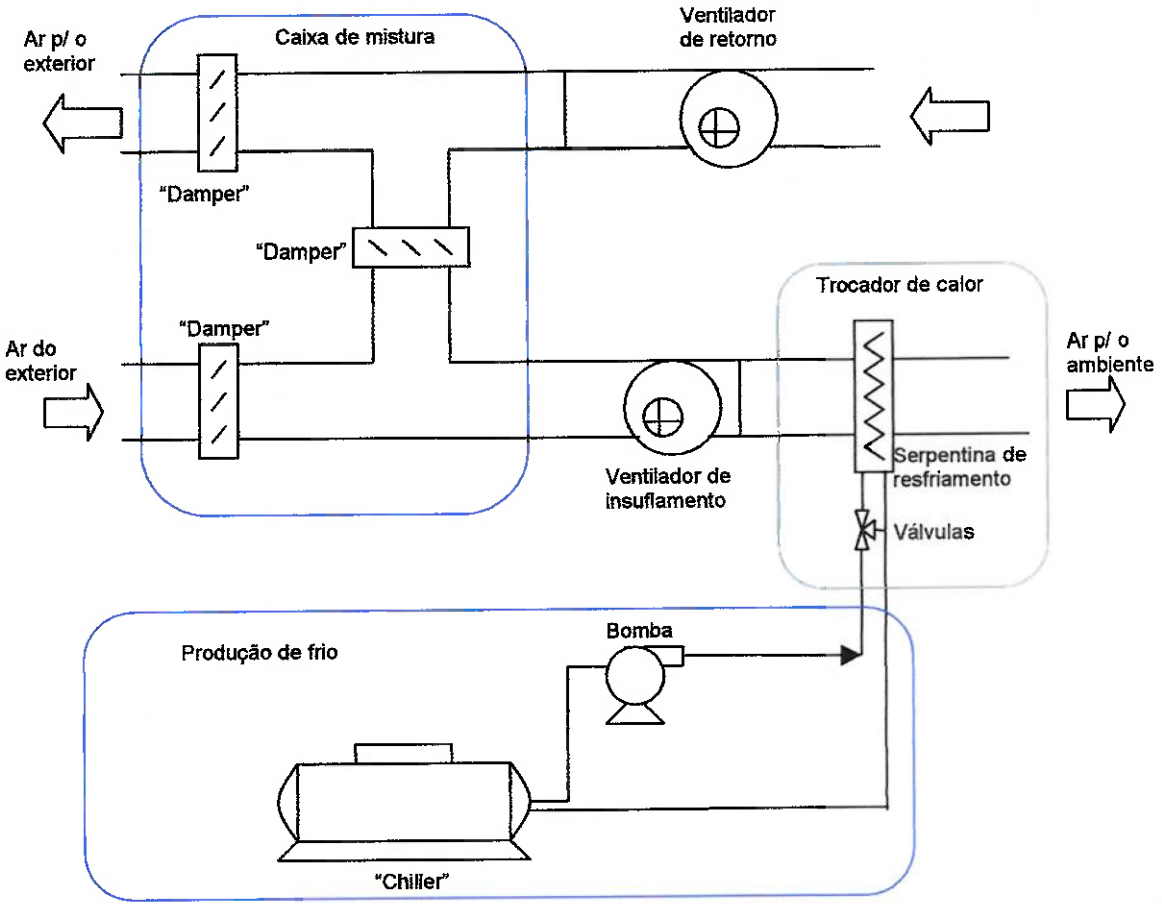


Figura 57 - Esquema do sistema de ar condicionado do estudo de caso

O sistema de gerenciamento adotado para o estudo de caso é formado pelas seguintes estratégias de controle:

- Estratégia em caso de Incêndio – adotada para cada zona em caso de detecção de incêndio na zona. Neste caso deve-se renovar 100% do ar retirado do ambiente e diminuir a velocidade do ventilador de insuflamento de forma a diminuir a pressão na zona com incêndio evitando assim que a fumaça se espalhe por outras zonas.
- Estratégias de área não utilizada – adotada quando não existem pessoa ou equipamentos ligados dentro de uma zona. Neste caso desligam-se os ventiladores, posiciona-se a caixa de mistura para renovação de 0% e desliga-se o controlador da válvula da serpentina.
- Estratégias de área utilizada – adotada quando existem pessoas ou equipamentos ligados dentro de uma zona. As medidas tomadas são: posicionar a renovação de ar para parcial, ventiladores de insuflamento e retorno são ligadas a uma velocidade média e o condicionamento do ambiente é ligado.

6.3 O Modelo do Sistema

Por questão de espaço e legibilidade do grafo, o modelo detalhado do estudo de caso encontra-se no Anexo II. As Figura 58 e Figura 59 são apenas o esquemático do modelo.

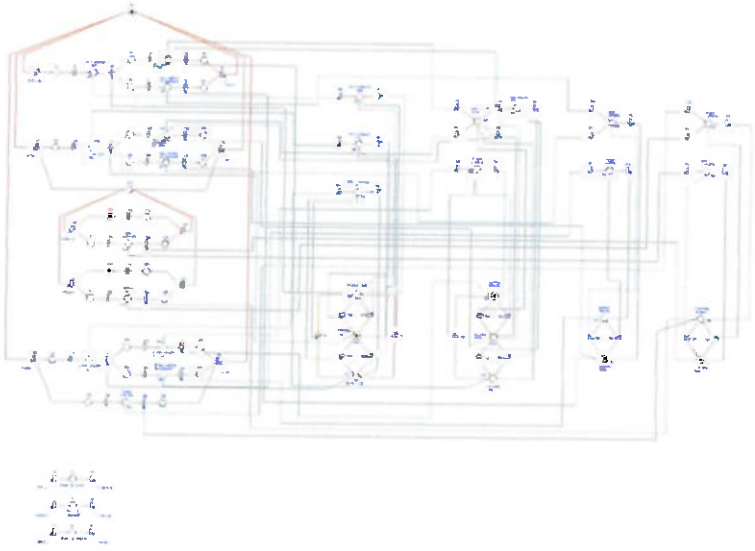


Figura 58 – Modelo em Rede de Petri do estudo de caso

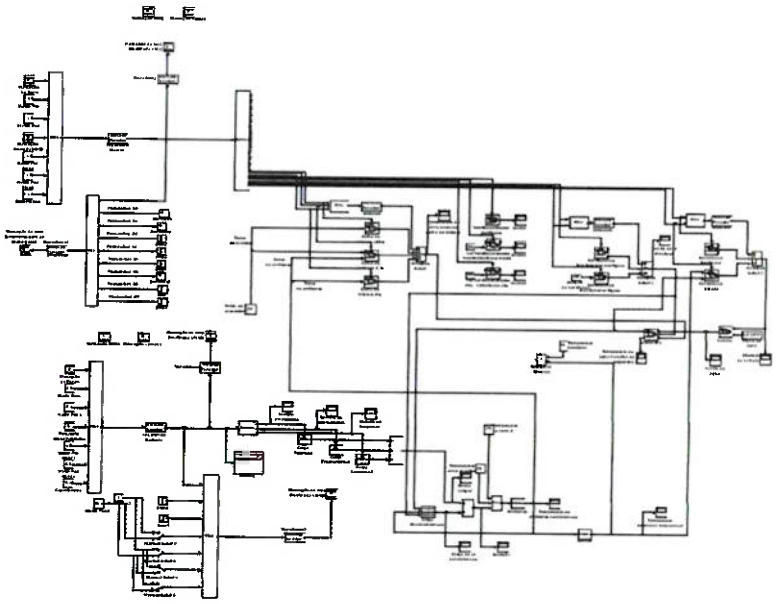


Figura 59 – Modelo do estudo de caso implementado no simulador

A parte discreta do sistema é adaptado do sistema original que encontra-se em [Villani, 2000] para o sistema adotado.

A parte contínua do sistema, os sistemas de equações diferenciais adotados e as constantes são extraídos de [Villani, 2000].

O modelo do sistema não considera variações de pressão do ambiente condicionado.

6.4 Codificação do modelo

Devido ao mesmo fator de espaço e legibilidade das matrizes de incidência, estas encontra-se no Anexo III. A seguir serão apresentadas os esquemáticos das matrizes da codificação do modelo.

$$A(\text{pré} - \text{condição}) = \begin{bmatrix} 1 & 0 & . & . & 0 & 0 \\ 0 & 1 & . & . & 0 & 0 \\ . & . & . & . & . & . \\ . & . & . & . & . & . \\ 0 & 0 & . & . & 1 & 0 \\ 0 & 0 & . & . & 0 & 1 \end{bmatrix}_{59x74}$$

$$B(\text{pós} - \text{condição}) = \begin{bmatrix} 0 & 0 & . & . & 0 & 0 \\ 1 & 0 & . & . & 0 & 0 \\ . & . & . & . & . & . \\ . & . & . & . & . & . \\ 0 & 0 & . & . & 0 & 1 \\ 0 & 0 & . & . & 1 & 0 \end{bmatrix}_{59x74}$$

$$M_0(\text{marcação inicial}) = \begin{bmatrix} 1 \\ 0 \\ . \\ . \\ 0 \\ 1 \end{bmatrix}_{59x1}$$

$$A'(\text{pré - condição consid. arcos}) = \begin{bmatrix} 0 & 0 & . & . & 0 & 0 \\ 0 & 0 & . & . & 0 & 0 \\ . & . & . & . & . & . \\ . & . & . & . & . & . \\ 0 & 0 & . & . & 0 & 0 \\ 0 & 0 & . & . & 0 & 0 \end{bmatrix}_{67x74}$$

B' (pós - condição consid. arcos) = Matriz nula de 67 x 74

$$M'_0 \text{ (marcação inicial + arcos + estados inciais externos)} = \begin{bmatrix} 1 \\ 0 \\ . \\ . \\ 0 \\ 0 \end{bmatrix}_{67x1}$$

6.5 Resultados da simulações

Os resultados das simulações são obtidas através de imposição de eventos que acontecem num determinado tempo ao longos das simulações. A imposição destes eventos é realizada utilizando um bloco com saída degrau. Para efeito mais genérico pode ser empregado um bloco de chaveamento manual para verificar o comportamento do sistema.

6.5.1 Simulação 1

No tempo t=0 a t=5 min entram 20 pessoas na zona, em tempo = 0,8 hs ligam-se as lâmpadas, um equipamento está ligado durante o período da simulação. A estratégia "área utilizada" é acionada, e o controlador é ligado.

No tempo t = 1,5 h ocorre um alarme de incêndio, e as 20 pessoas saem da zona, entretanto as lâmpadas continuam ligadas.

Para esta simulação a temperatura do ambiente condicionado varia da conforme a Figura 60

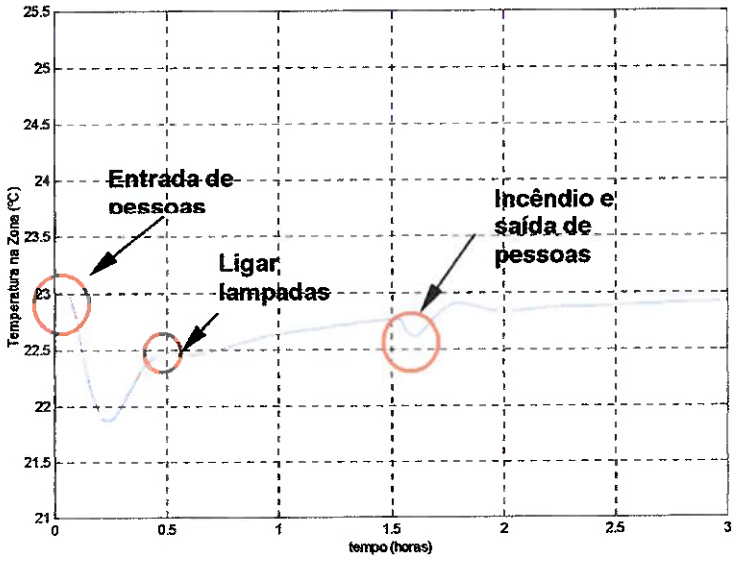


Figura 60 – Evolução da temperatura da zona ao longo do tempo

Através do modelo implementado, pode-se obter várias informações como as referentes às características do fluxo de ar e água, equipamento do ar condicionado além das informações como quantidades pessoas e equipamentos ligados.

Como exemplo as Figura 61, Figura 62, e Figura 63, apresentam respectivamente, a evolução no tempo da quantidade de pessoas, temperatura da água na saída da serpentina e posição da válvula de três vias.

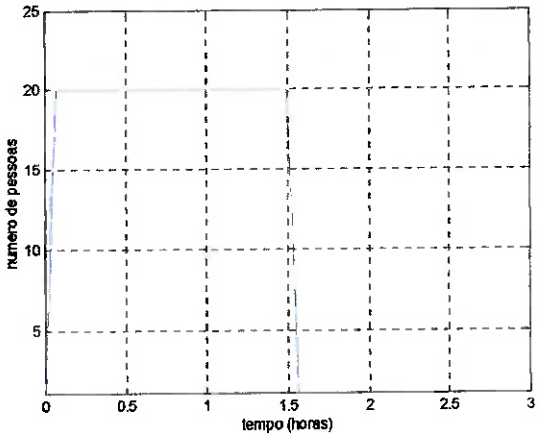


Figura 61 – Evolução da quantidade de pessoas ao longo do tempo

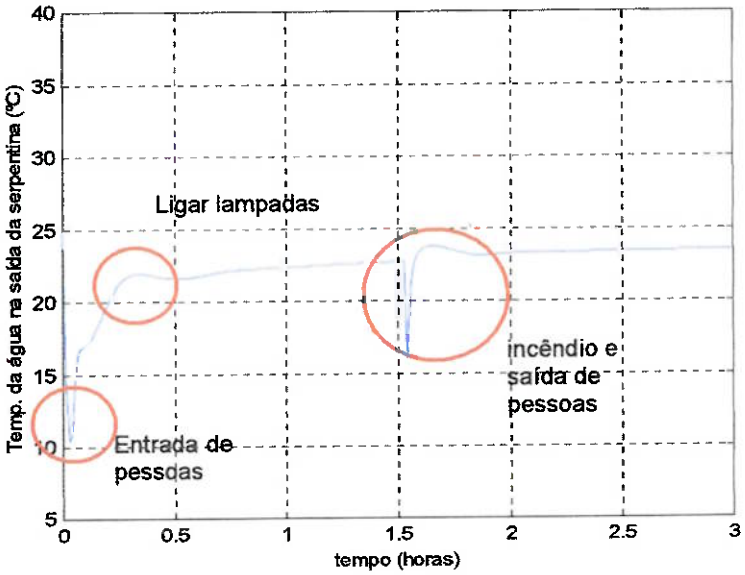


Figura 62 – Evolução da temperatura da água na saída da serpentina

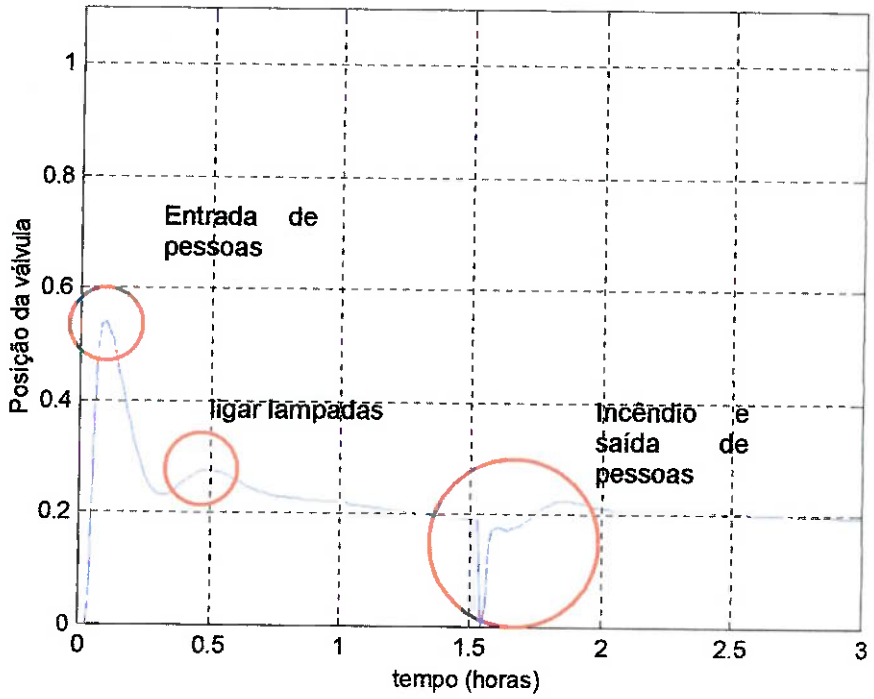


Figura 63 – Evolução da posição da válvula ao longo do tempo

6.5.2 *Simulação 2*

Os eventos que acontecem na segunda simulação são:

- Entrada de pessoas entre tempo t=0 e t=5 min.
- Ligar a estratégia da "área utilizada";
- Acender as lâmpadas;
- Saída das pessoas no tempo aproximadamente t= 2 h;
- Passar para a estratégia da "área não utilizada" e desligar os controladores;

As Figura 64, Figura 65 e Figura 66 são respectivamente, temperatura do ambiente, temperatura da água na saída da serpentina e posição da válvula de três vias.

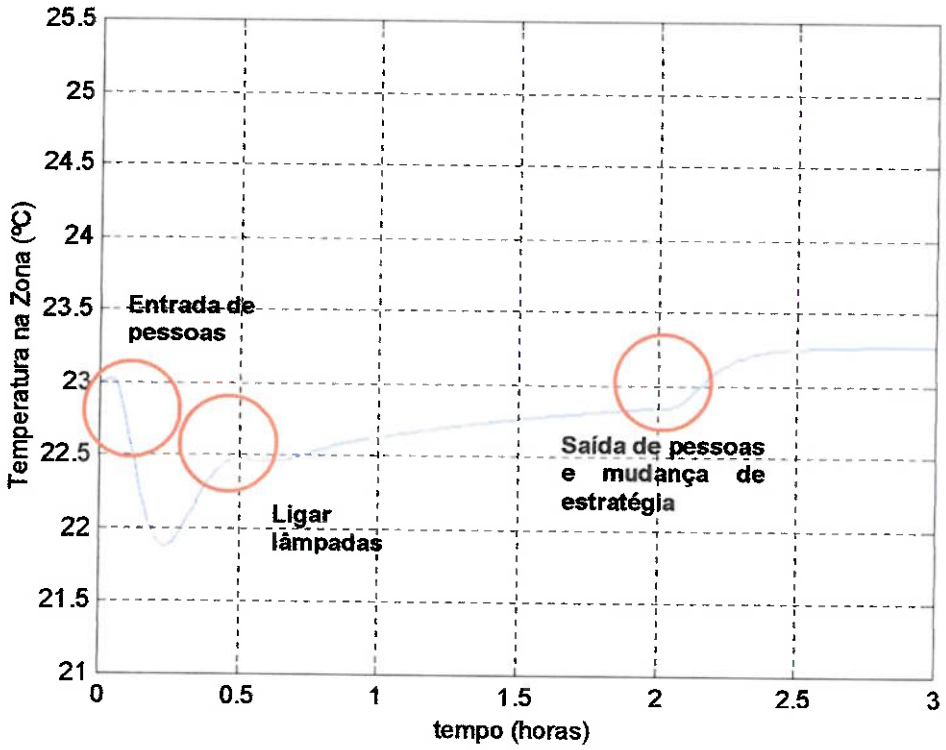


Figura 64 – Evolução da temperatura ambiente com ao longo do tempo

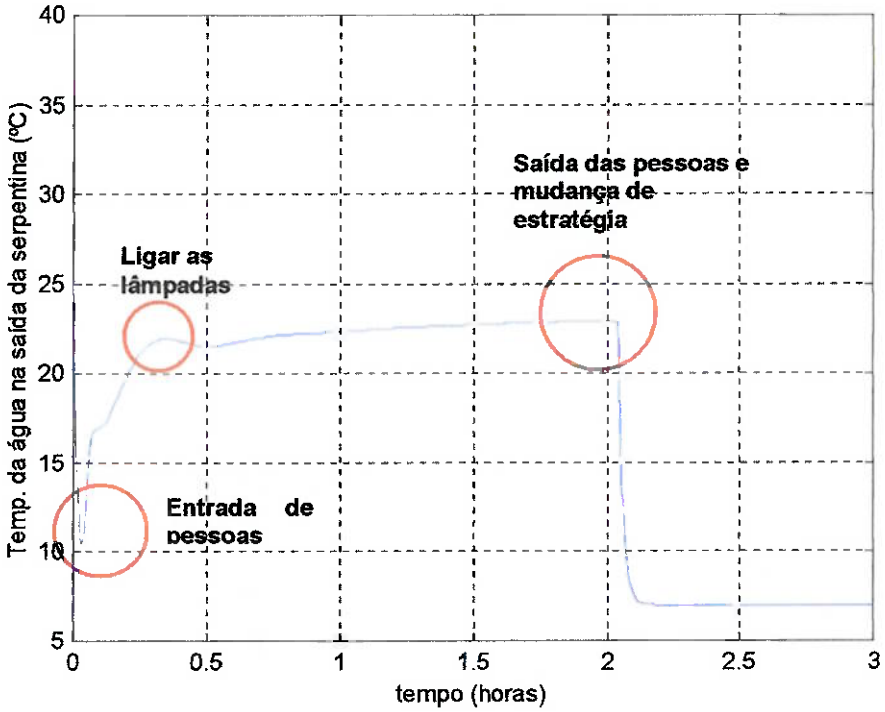


Figura 65 – Evolução da temperatura da água na saída da serpentina

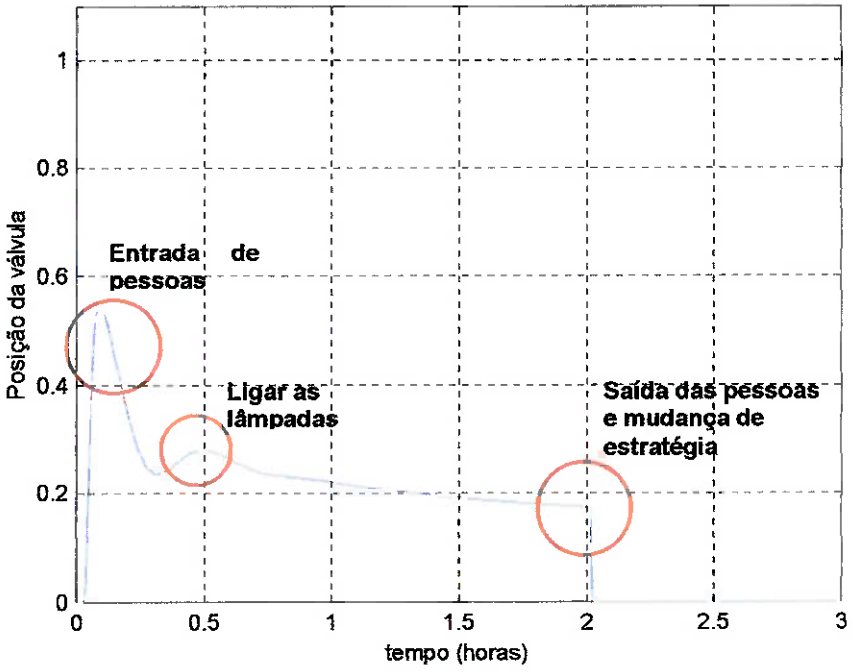


Figura 66 – Evolução da posição da válvula

7 Considerações Finais

Os sistemas híbridos são um tópico de estudo relativamente recente, quando comparado com os sistemas a eventos discretos ou com o sistemas de variáveis contínuas, e devido a carência de técnicas analíticas de caracterização comportamental, as análise destes sistemas baseiam-se principalmente em simulação, isto é, o estudo e projeto desta classe de sistema é baseado na análise da verificação do comportamento dinâmico de modelos. O desenvolvimento de um simulador é portanto de grande utilidade.

O plano de trabalho desenvolvido para conduzir o trabalho foi dividido da seguinte forma:

1. Estudo dos fundamentos teóricos;
2. Estudo das ferramentas;
3. Desenvolvimento dos algoritmos;
4. Implementação do simulador;
5. Aplicação do simulador em um estudo de caso;

A técnica de modelagem de sistemas híbridos baseada em Rede Predicado/Transição Diferencial combina a descrição das características de um sistema à eventos discretos com as características de um sistema de variáveis contínuas em um único modelo, que foi considerada na implementação do simulador.

A metodologia empregada para elaboração do simulador foi baseada na técnica *Top-Down*, isto é, no detalhamento sucessivo das funções do simulador até atingir uma forma com maior facilidade de implementação.

As ferramentas empregadas para a implementação do simulador mostraram-se eficientes para a aplicação, entretanto o Simulink apresentou certas limitações pela dificuldade de trabalhar com dados matriciais.

Quanto ao desempenho do simulador observou-se que:

- O simulador é eficiente para modelos relativamente pequenos, aproximadamente 20 "lugares" e 20 "transições";
- Para modelos maiores (mais de 70 "lugares" e 70 "transições") o simulador mostou-se lento.
- Apesar da funcionalidade do simulador, ele carece de uma melhor interface gráfica com o usário.

Quanto aos resultados das simulações realizadas observou-se, por exemplo que:

- A temperatura, quando liga a estratégia de "área utilizada", sempre tendo à temperatura do *setup* de 23°;
- A temperatura da zona controlada é influenciada a cada introdução de evento na zona;
- O resultado mostrou-se coerente com os resultados previstos em [Villani, 2000].

Esta análise de resultados ilustram a utilidade da ferramenta desenvolvida

Tomando-se como base as observações feitas acima, para os trabalhos futuros sugere-se os seguintes pontos:

- Melhoria da interface com o usuário;
- Revisão do algoritmo adotado para acelerar as simulações;

- Aplicação do simulador no mesmo estudo de caso, porém sem as simplificações adotadas;
- Aplicação do simulador em outros estudos de caso, tais como chão de fábrica, controle de processo de soldagem, etc.

8 Bibliografia

- Alla, H. & David, R; "Continuous and Hybrid Petri Nets" Journal of Circuits, Systems and Computers, vol.8, n.1, pp 159-188; 1998.
- Arata, Wilson Munemassa; Analise quantitativa de sistemas de manufatura: abordagem baseada em redes gspn Dissertação de Mestrado, EPUSP, Sao Paulo, 1994.
- Futema, Jorge; Curso Prático de Solda a Ponto; Mafersa S.A; 1997
- Maciel, Paulo Romero Martins; Introdução as Redes de Petri e Aplicacoes; Editora Unicamp, 1996.
- Miyagi, Paulo Eigi; Controle Programável, Fundamentos de Controle de Sistemas a Eventos Discretos; Editora Edgard Blücher LTDA, São Paulo,1996
- Matsusaki, Cristina Toshie Motohashi; Redes F-MFG (Functional Mark Flow Graph) e sua aplicação no projeto de sistemas antropocêntricos. Dissertação de Mestrado, EPUSP, São Paulo, 1998.
- Matsumoto, E.Y; MATLAB 6 – Fundamentos de programação; Editora Érica LTDA, São Paulo, 2001.
- Villani, Emília; Abordagem híbrida para modelagem de sistemas de ar condicionado em edifícios inteligentes; Dissertação de Mestrado, EPUSP, São Paulo, 2000

ANEXO I – CÓDIGO FONTE

Abaixo estão o código fonte da parte discreta do simulador. Este parte é implementada em Matlab e é composta pelas seguintes funções:

- "Função Jogador": Função principal da parte discreta do simulador.
- "Função Simula": Esta função é responsável pela organização e seqüências das etapas a serem executadas durante a simulação.
- "Função transição": Esta função verifica todos as transições e retorna quais delas estão habilitadas.
- "Função conflito": Localiza se há conflito na rede.
- "Função atualiza": Atualiza o estado da rede
- "Função habilita": Após a verificação das transições que estão habilitadas, esta função define quais delas podem ser disparadas. Caso haja conflito, é dentro desta função que será resolvida esse conflito. A transição é selecionada para o disparo através de um processo de escolha aleatória.
- "Função Transforma": Esta função transforma os dados de saída do jogador de marcas no formato válido para a entrado do jogador de marcas.

Função Jogador

```
function vectormat = vectormat(vetor)

vetor_linha = size(vetor,1);   %quantidade de linha de um vetor em simulink%

k = 1;
x = 1;        % variavel auxiliar

while k<=vetor_linha
  linha = vetor(k,1);
  k = k + 1;
  coluna = vetor(k,1);
  k = k + 1;
  matriz = 0;
  for i=1:linha
    for j=1:coluna
      matriz(i,j)=vetor(k,1);
      k=k+1;
    end
  end

  if x==1
    m0 = matriz;
  end
  if x==2
    pre = matriz;
  end
  if x==3
    pos = matriz;
  end
  if x==4
    m0_gate = matriz;
  end
  if x==5
    pre_gate = matriz;
  end
  if x==6
    pos_gate = matriz;
  end
  if x==7
    capacidade = matriz;
  end
    x = x+1;
end

vectormat = simula(m0,pre,pos,m0_gate,pre_gate,pos_gate,capacidade);
```

Função Simula

```
function simula = simula(m0, pre, pos, m0_gate, pre_gate, pos_gate, capacidade)

transicao = habilita(m0, pre, pos, m0_gate, pre_gate, pos_gate, capacidade);
simula = atualiza(m0,pre,pos,transicao);
```

Função Habilita

```
function habilita = habilita(m0, pre, pos, m0_gate, pre_gate, pos_gate, capacidade)

 vetor_transicao_um = trans(pre,pre,pos,m0,1, capacidade);

 vetor_transicao_dois = trans(pre,pre_gate,pos_gate,m0_gate,0, capacidade);

 transicoes = size(pre,2);

 for j=1:transicoes
  vetor_transicao(j,1) = vetor_transicao_um(j,1)*vetor_transicao_dois(j,1);
 end

%resolvendo conflitos

 lugares = size(m0,1);

    conflict = conflito(pre);
 resolvido=0;


for i=1:lugares
 if((m0(i,1)~=0) & (conflict(i)~=0))
  conflitos = 0;
  for j=1:transicoes
   if(pre(i,j)~=0)
    conflitos = conflitos + 1;
   end
  end
  for j=1:transicoes
   if(pre(i,j)==0)
    resolve_conflito(j)=1;
   else
    criterio = round(10*rand);
    if ((conflitos ==0) & (resolvido==0))
     resolve_conflito(j)=1;
    end
    if ((conflitos==0) & (resolvido==1))
       resolve_conflito(j)=0;
    end
    if ((conflitos == 1) & (resolvido==0))
      resolve_conflito(j)=1;
      conflitos = 0;
      resolvido=1;
    end
```

```
    if((criterio >5) & (conflitos>1))
     resolve_conflito(j)=1;
     conflitos = 0;
     resolvido=1;
    end
    if((criterio <=5) & (conflitos>1))
     resolve_conflito(j)=0;
     conflitos = conflitos - 1;
    end
   end
  end
  for j=1:transicoes
   vetor_transicao(j,1)=vetor_transicao(j,1)*resolve_conflito(j);
  end
 end
end
habilita =  vetor_transicao;
```

Função Transição

```
function trans = trans (pre1,pre,pos,m0,normal,capacidade)

c= pos-pre;    %Calcular a matriz de efeitos liquidos

if(normal ==0)
  original = size(capacidade,2);
  gates = size(pre,1);
  for i=original:gates
    capacidade(i+1)=1;
  end
end

lugares = size(pre,1);
transicoes = size(pre,2);

if(lugares > 1)
 for j=1:transicoes
  contador = 0;
  for i=1:lugares
   if(m0(i,1)>=pre(i,j))
     contador = contador +1;
   end
  if(contador == lugares)
   vetor_pre(j,1) = 1;
  else
   vetor_pre(j,1) = 0;
  end
 end
end

 for j=1:transicoes
 contador = 0;
```

```
  for i=1:lugares
   if((m0(i,1)+c(i,j))<=capacidade(i))
    contador = contador +1;
   end
   if(contador == lugares)
    vetor_pos(j,1) = 1;
   else
    vetor_pos(j,1) = 0;
   end
  end
 end
 for i=1:transicoes
  vetor_transicao(i,1) = vetor_pre(i,1)*vetor_pos(i,1);
 end

trans = vetor_transicao;

 else
  transicoes = size(pre1,2);
  for j=1:transicoes
   vetor_transicao(j,1) = 1;
  end
  trans = vetor_transicao;
 end
```

Função Conflito

```
function conflito = conflito(pre)

 lugares = size(pre,1);
 transicoes = size(pre,2);

 for i=1:lugares
  contador = 0;
  for j=1:transicoes
   if (pre(i,j)~=0)
    contador = contador +1;
   end
  end
  if(contador>1)
   lugar_conflito(i) = 1;
  else
   lugar_conflito(i) = 0;
  end
 end

 conflito = lugar_conflito;
```

Função Atualiza

```
function atualiza = atualiza(m0,pre,pos,transicao)

c = pos - pre;

 m0 = m0 + c*transicao;

atualiza = m0;
```

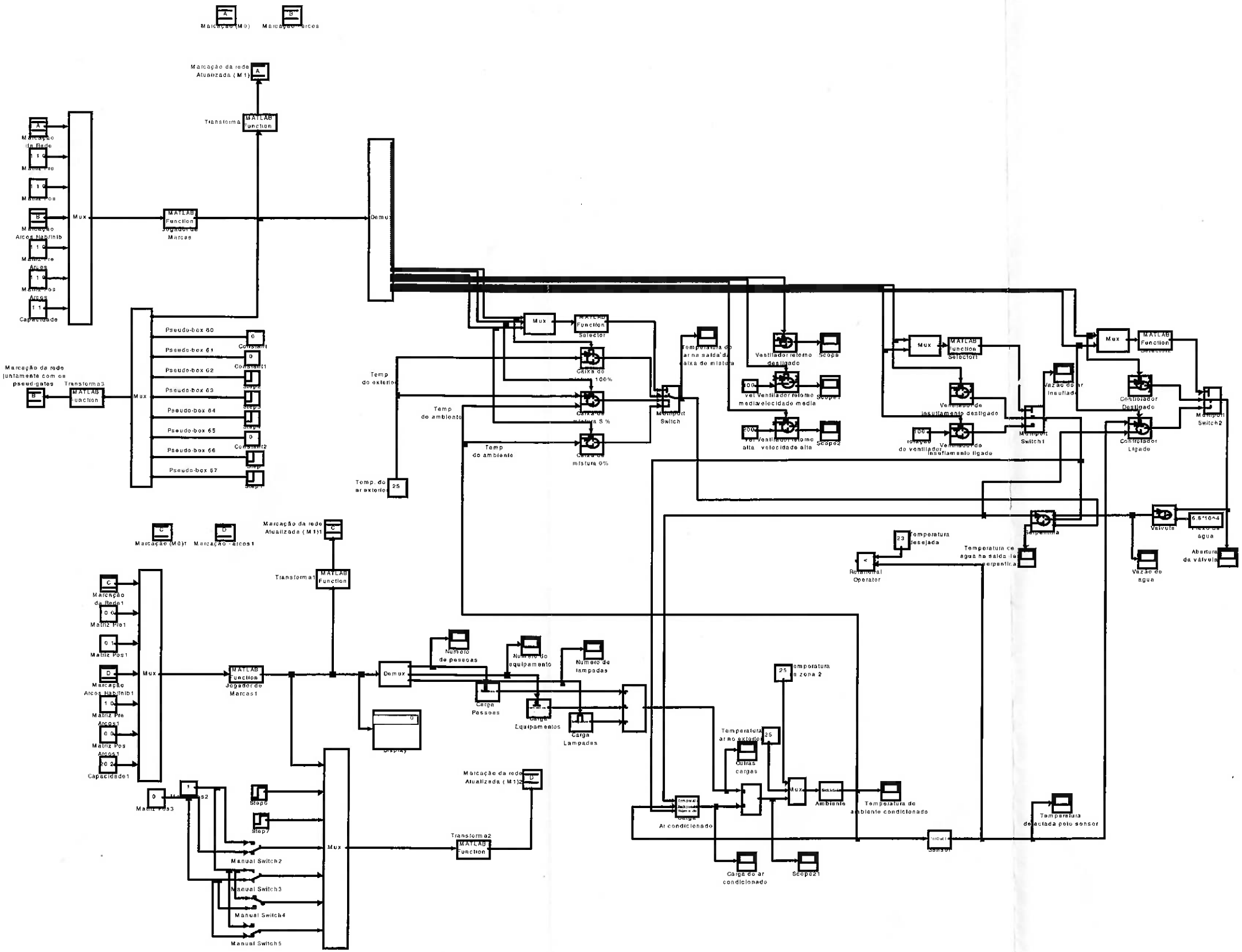
Função Transforma

```
function test2 = test2(x)

linha = size(x,1);
coluna = size(x,2);
resultado(1,1)=linha;
resultado(2,1)=coluna;

for i=1:linha
   resultado(i+2,1) = x(i,1);
end

test2=resultado;
```


Marcação (M0)
Marcação Arcos
Marcação da rede Atualizada (M1)
Transforma
MATLAB Function
Marcação da Rede
Matriz Pre
Matriz Pos
Marcação Arcos Habilit
Matriz Pre Arcos
Matriz Pos Arcos
Capacidade
Mux
MATLAB Function Jogador de Marcas
Demux
Pseudo-box 60
Pseudo-box 61
Pseudo-box 62
Pseudo-box 63
Pseudo-box 64
Pseudo-box 65
Pseudo-box 66
Pseudo-box 67
Marcação da rede juntamente com os pseudo-gates
Transforma3
Temp do exterior
Temp do ambiente
Temp. do ar exterior
25
Caixa de mistura 100%
Caixa de mistura 5 %
Caixa de mistura 0%
Manual Switch
Temperatura do ar na saída da caixa de mistura
Ventilador retorno desligado
Scope
100% vel Ventilador retorno media velocidade media
Scope1
20% Ventilador retorno alta velocidade alta
Scope2
Selector
Vazão de ar insuflado
Ventilador de insuflamento desligado
100% rotação do ventilador
Ventilador de insuflamento ligado
Manual Switch1
Controlador Desligado
Controlador Ligado
Manual Switch2
Temperatura desejada
Relational Operator
Temperatura de agua na saída do serpentina
Serpentina
Valvula
6.6*10^-4
Fluxo de água
Vazão de água
Abertura da válvula
Marcação (M0)1
Marcação Arcos1
Marcação da rede Atualizada (M1)1
Transforma1
Marcação da Rede1
Matriz Pos1
Matriz Pre1
Marcação Arcos Habilit1
Matriz Pre Arcos1
Matriz Pos Arcos1
Capacidade1
Matriz Pos3
Manual Switch3
Manual Switch4
Manual Switch5
Step6
Step7
Numero de pessoas
Numero de equipamento
Numero de lampadas
Carga Pessoas
Carga Equipamentos
Carga Lampadas
Display
Marcação da rede Atualizada (M1)2
Transforma2
Temperatura da zona 2
Temperatura ar no exterior
Outras cargas
Ar condicionado
Carga do ar condicionado
Scope21
Ambiente
Temperatura do ambiente condicionado
Sensor
Temperatura detectada pelo sensor

ANEXO III – CODIFICACAO DO MODELO

a. Matriz de pré condição

1 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

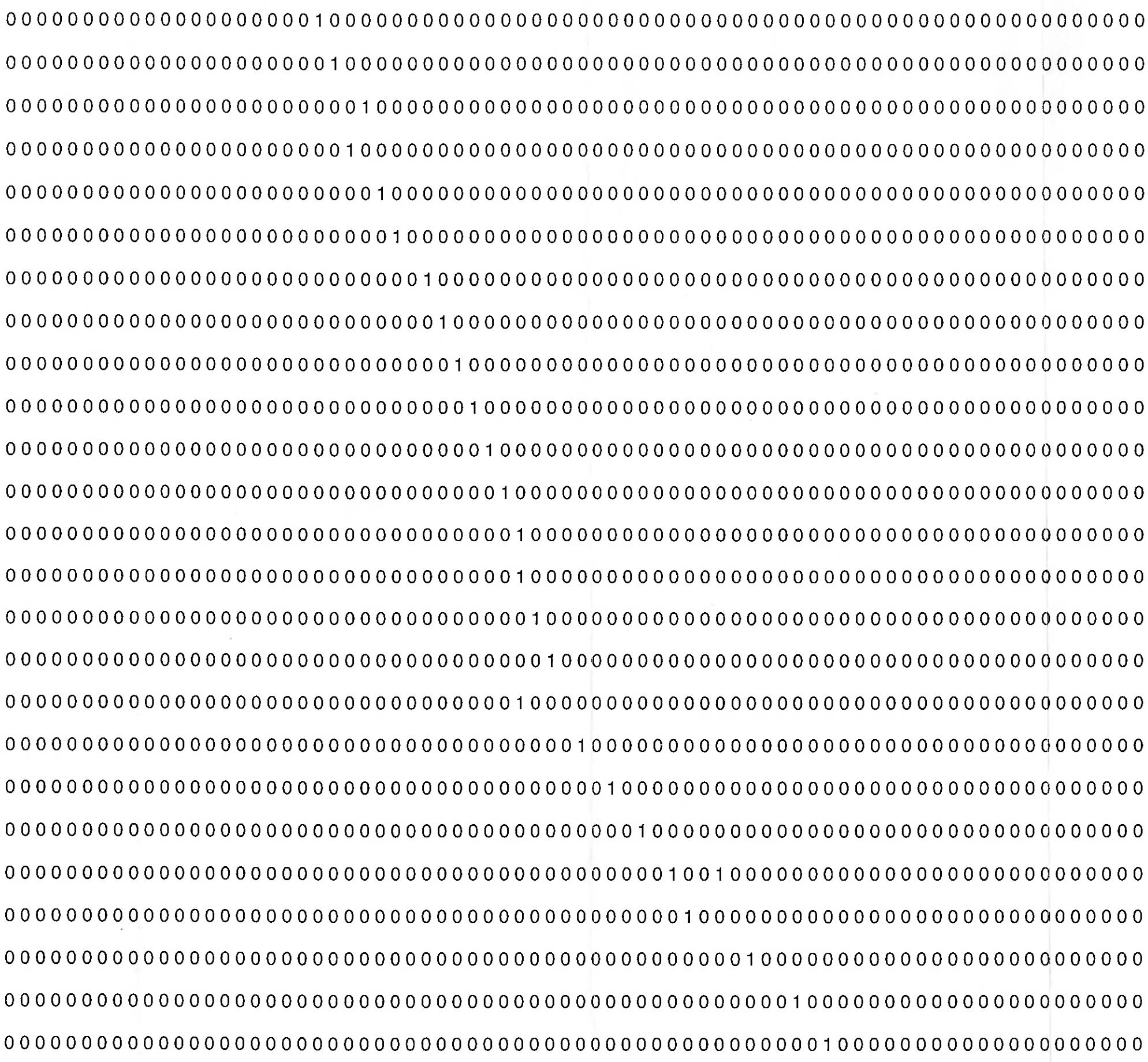
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

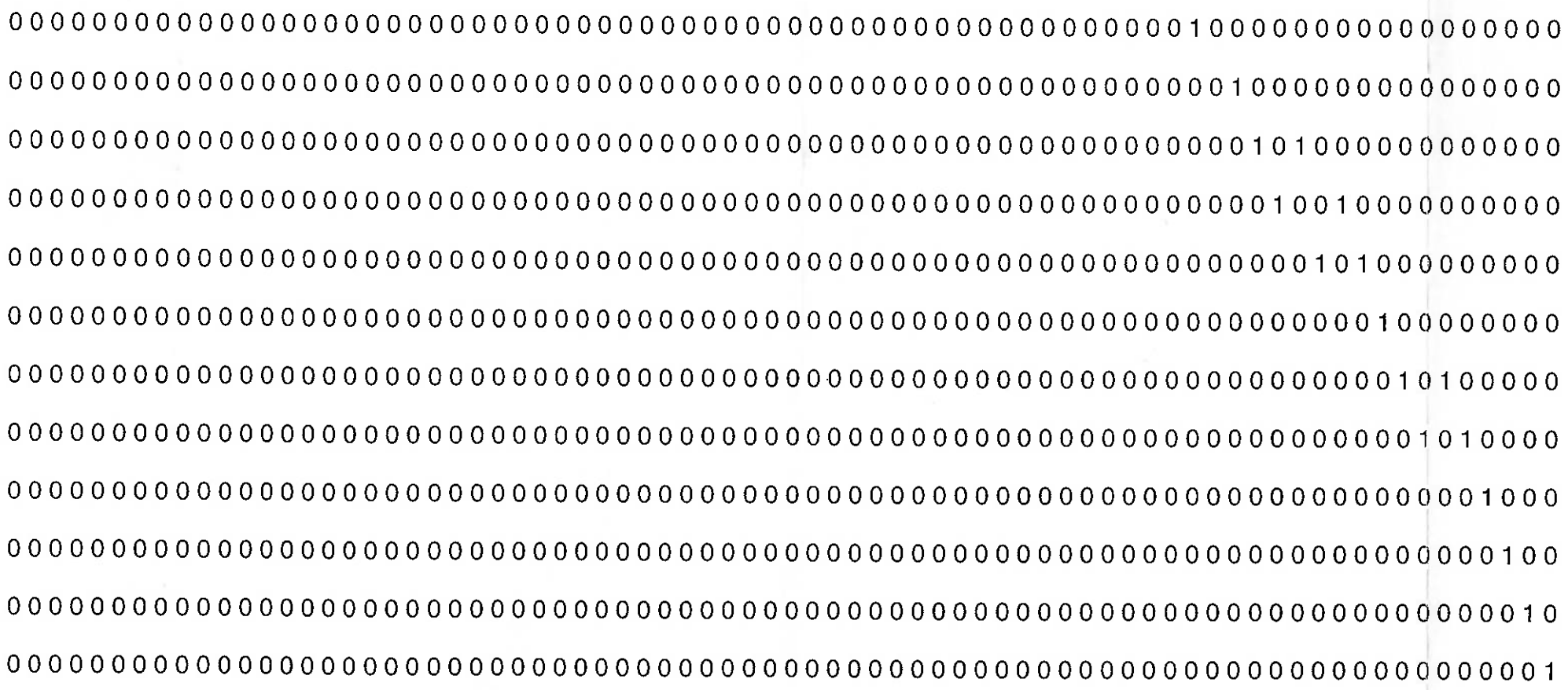
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

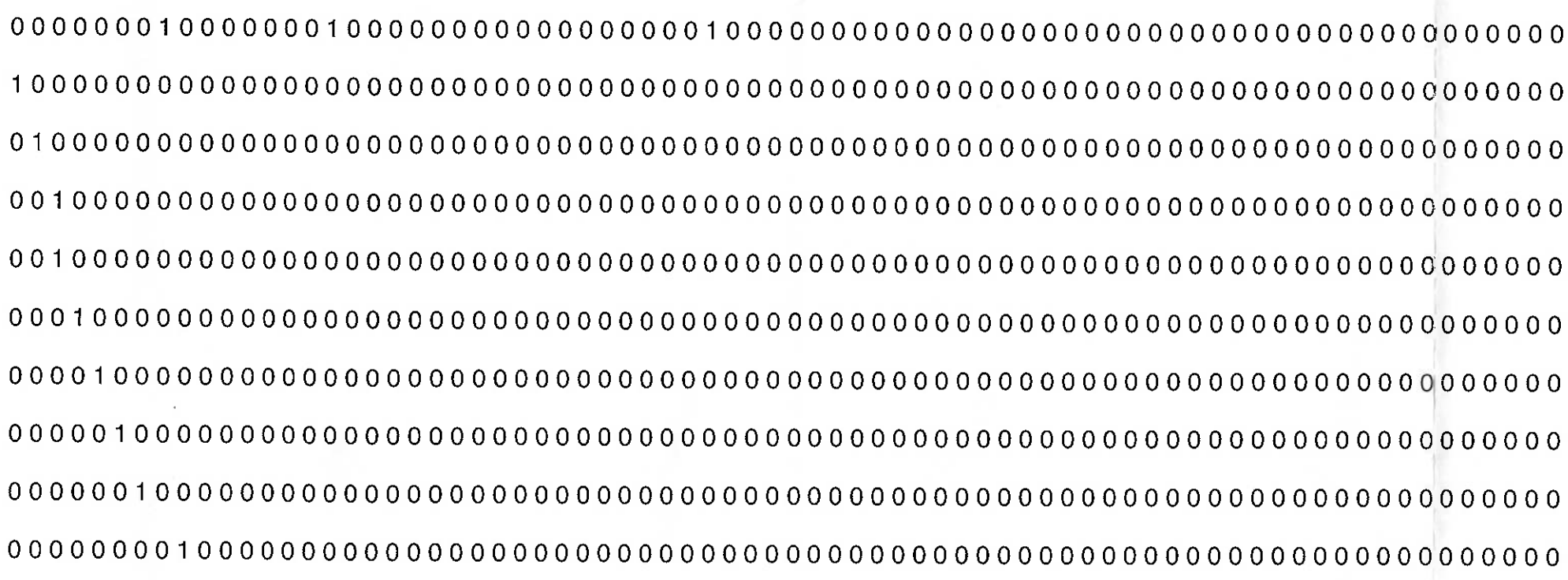
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

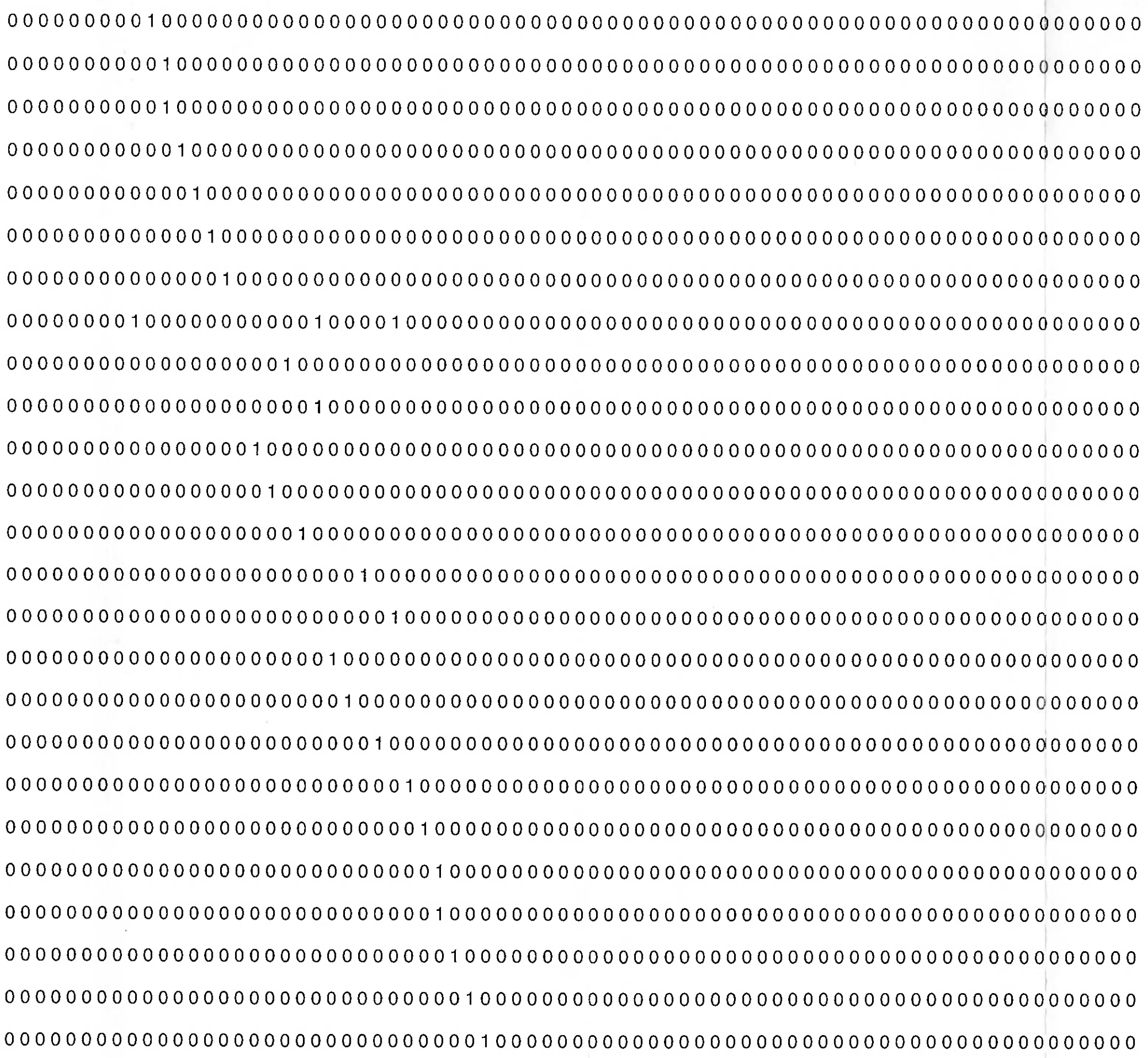
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

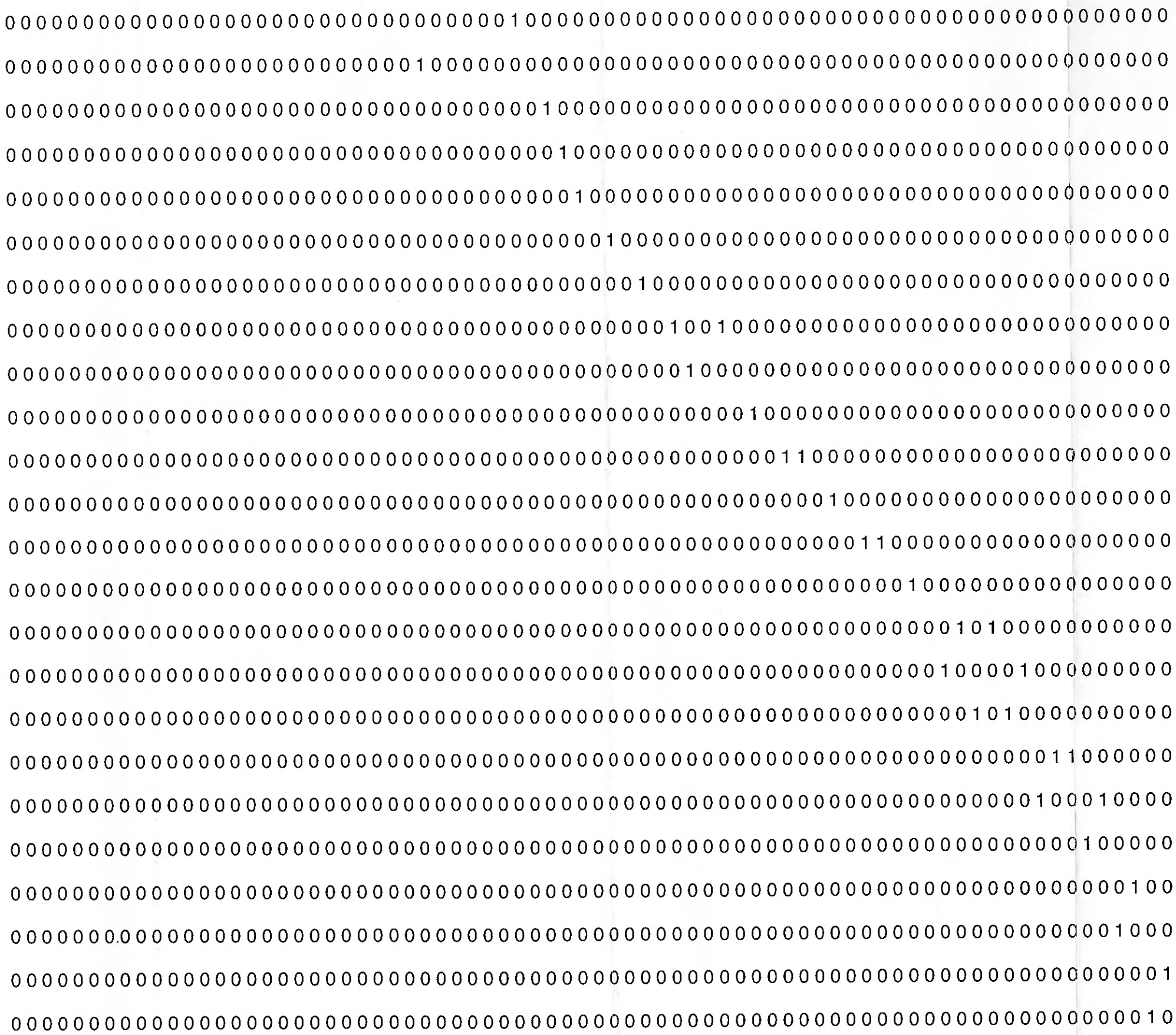




b. Matriz de pós condição







c. Matriz de pré condição incluindo os arcos e sinais externos

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0

0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 1 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

d. Matriz de pós condição incluindo os arcos e sinais externos

MATRIZ NULA DE DIMENSÃO 67 X 74