

UNIVERSIDADE DE SÃO PAULO - ESCOLA POLITÉCNICA  
DEPARTAMENTO DE ENGENHARIA MECATRÔNICA E SISTEMAS MECÂNICOS

Note final  
9,9 (nove e nove)  


PROJETO DE CONCLUSÃO DE CURSO

LUÍS FELIPE DE ALMEIDA ZANINI  
MARCELO STEFFEN

Ferramenta computacional para designação de controladores programáveis em  
arquiteturas de controle distribuídas

São Paulo  
2005

LUÍS FELIPE DE ALMEIDA ZANINI  
MARCELO STEFFEN

**FERRAMENTA COMPUTACIONAL PARA  
DESIGNAÇÃO DE CP'S EM ARQUITETURAS DE  
CONTROLE DISTRIBUÍDAS**

Dissertação apresentada à  
Escola Politécnica da  
Universidade de São Paulo  
para obtenção do título de  
Engenheiro Mecânico –  
Ênfase Automação e Sistemas

Orientador: Prof. Dr. Diolino José dos Santos Filho

São Paulo  
2005

DEDALUS - Acervo - EPMN



31600011855

1493286

**FICHA CATALOGRÁFICA**

Zanini, Luís Felipe de Almeida  
Ferramenta computacional para designação de  
controladores  
programáveis em arquiteturas de controle distribuídas / L.F. de  
A. Zanini, M. Steffen. -- São Paulo, 2005.  
60 p.

Trabalho de Formatura - Escola Politécnica da Universidade  
de São Paulo. Departamento de Engenharia Mecatrônica e de  
Sistemas Mecânicos.

1. Sistemas discretos 2. Controladores programáveis  
I. Steffen,  
Marcelo II. Universidade de São Paulo. Escola Politécnica.  
Departamento de Engenharia Mecatrônica e de Sistemas  
Mecânicos III.t.

## **DEDICATÓRIA**

Dedicamos este trabalho ao Prof. Dr. Diolino que nos guiou ao longo deste projeto, e que, com sua visão, levou-nos a um trabalho de continuidade potencial.

## AGRADECIMENTOS

Agradeço a Deus pela companhia constante durante todo este processo muitas vezes difícil. Aos meus pais pelo apoio e incentivo constante sem os quais não conseguiria completar mais esta etapa. Agradeço à Vanessa, meu amor, por todo apoio, compreensão e cumplicidade dedicados durante esta jornada.

Luis Felipe de Almeida Zanini

Agradeço a meus pais que sempre me apoiaram e me incentivaram com seus conselhos, experiências, ensinamentos e suporte incondicional aos meus estudos. Sem eles não chegaria onde estou, e sem eles não teria as oportunidades que se abrem e aquelas as quais vislumbro.

Agradeço também a minha eterna companheira por toda a motivação, descontração e alegria que me proporcionou tanto nos momentos felizes como nos momentos difíceis e de concentração.

Aos meus amigos por compreenderem minha ausência nestes anos de estudo intenso.  
Àqueles que inspiraram todo um sonho de Engenharia.  
E aos professores que mostraram mundos novos de possibilidades.

Marcelo Steffen

## RESUMO

Observando o mercado atual com constantes mudanças em tecnologia e demanda, percebemos a necessidade de dinamismo e agilidade nos sistemas produtivos. O controle da produção deve estar condizente à necessidade de mudanças rápidas no processo ou em partes deste.

Dessa maneira, várias pesquisas na área de controle foram desenvolvidas chegando a teorias para o projeto e implementação de sistemas de controle modulares e distribuídos, em que o sistema de controle deixa de ser centralizado e passa a atuar em módulos dentro da planta, tal que quanto maior o número de módulos mais eficaz é este tipo de controle.

A finalidade deste trabalho é o desenvolvimento e a implementação de uma ferramenta de software que trate do problema de designação de controladores programáveis – CPs - em uma arquitetura física, e realize a distribuição dos códigos de controle do processo nestes controladores a partir dos algoritmos de controle existentes, possibilitando maior agilidade na manutenção e atualização do processo que atua na planta.

Esta abordagem mostra-se como uma alternativa aos procedimentos que são realizados atualmente que depende intrinsecamente do conhecimento empírico dos projetistas e pode provocar custos maiores de implementação do sistema de controle, além de dificultar ações rápidas de manutenção/atualização rápida. Com a ferramenta proposta, o tempo de implementação e o custo poderão ser reduzidos/otimizados permitindo mudanças e atualizações freqüentes nos sistemas produtivos.

## ABSTRACT

Observing the current market scene with constants changes in technology and demand, it's visible the necessity of dynamism and agility in the productive systems. So, the production control must follow this necessity of fast changes in the processes or in some of its part.

In this way, many researches in this area have been developed leading to theories for the project and implementation of modular and distributed control systems, where the control system leaves of being centered and starts to act inside modules of the plant, such that the bigger the number of modules, more efficient is this type of control.

The purpose of this work is the development and the implementation in software of a tool to distribute controllers for the modules of a physical architecture, and the distribution of control codes of the process within these controllers using the existent algorithms, making possible faster agility to change the process that acts in this plant.

This approach is an alternative in contrast to as it is made currently: modifications with empirical procedure made by the designers that increase the cost of implementation and disables fast implementation in real time. With this proposed tool, the period of implementation and the cost could be reduced allowing frequent changes and updates in the productive systems.

# SUMÁRIO

<b>1 INTRODUÇÃO .....</b>	<b>8</b>
1.1 Objetivo deste documento .....	8
1.2 Motivação.....	8
<b>2 ENTENDIMENTOS .....</b>	<b>9</b>
2.1 Definição do problema .....	9
2.2 Unidade de Trabalho.....	9
2.3 Restrições para distribuição .....	10
<b>3 CONCEITOS BÁSICOS .....</b>	<b>11</b>
3.1 Classificação dos sistemas de controle .....	11
3.2 Componentes básicos de um sistema de controle.....	11
3.3 Classificação dos sistemas de controle em função da topologia adotada .....	12
3.4 Organização dos controladores em sistemas de controle distribuídos .....	13
3.5 Módulos de controle e controle modular.....	13
3.6 Ferramentas empregadas na modelagem de SED.....	13
3.6.1 Redes de Petri e representação de SED por Redes de Petri.....	14
3.6.1.1 Vantagens das Redes de Petri .....	14
3.6.1.2 Composição das Redes de Petri.....	14
3.6.2 PFS/MFG .....	15
3.6.2.1 PFS - Production Flow Shema.....	15
3.6.2.1.1 Elementos estruturais do PFS.....	15
3.6.2.2 MFG – Mark Flow Graph.....	16
3.6.2.2.1 Características do MFG.....	16
3.6.2.2.2 Elementos estruturais do MFG .....	17
3.6.2.2.3 Dinâmica do MFG .....	18
3.6.2.2.4 Transição habilitada.....	18
3.6.2.2.5 Regras para disparo da transição .....	18
3.6.2.2.6 Extensão para conceito de tempo .....	19
<b>4 METODOLOGIA .....</b>	<b>21</b>
4.1 Entradas necessárias para o algoritmo.....	21
4.1.1 Entradas referentes ao algoritmo de alocação física dos CPs .....	21
4.1.2 Entradas referentes ao algoritmo de alocação de blocos funcionais .....	21
4.1.3 Estrutura de árvores .....	21
4.2 Critérios envolvidos no desenvolvimento do algoritmo de alocação de blocos funcionais .....	26
4.3 Alocação dos blocos funcionais de um sistema produtivo .....	27
4.4 Uma descrição geral do algoritmo a ser implementado.....	28
4.4.1 Extração das informações referentes aos blocos funcionais .....	29
4.4.2 Definição dos tipos de CPs e suas caracterísitcas.....	29
4.4.3 Definição dos CPs .....	30
4.4.4 Distribuição pela arquitetura física da planta.....	30
4.4.5 Alocação (designação) dos blocos funcionais dos CPs .....	30
4.4.6 Verificação final.....	30
4.5 Descrição do algoritmo de alocação física dos CPs e designação de blocos funcionais em CPs .....	30

4.5.1 Alocação física dos CPs .....	30
4.5.1.1 Conceitos e funções envolvidas na alocação física dos CPs.....	31
4.5.2 Alocação dos blocos funcionais .....	37
4.5.2.1 Modelo do bloco funcional .....	38
4.5.2.2 Modelo dos CPs.....	38
4.5.2.3 Custo computacional dos blocos funcionais.....	39
4.5.2.4 Consumo de recursos dos CPs.....	40
<b>5 IMPLEMENTAÇÃO .....</b>	<b>44</b>
5.1 Considerações sobre a implementação.....	44
5.2 A implementação em software.....	44
5.2.1 Recursos utilizados:.....	44
5.2.2 Implementação das Entradas e Saídas .....	45
5.2.2.1 Processos Estruturados em Árvores .....	45
5.2.2.2 Base de Dados de CP's .....	45
5.2.3 Algoritmo de Alocação Funcional .....	46
5.2.4 Algoritmo de Alocação Física .....	46
5.2.5 Demonstração dos resultados e posicionamento .....	47
<b>6 TUTORIAL PARA UTILIZAÇÃO DO SOFTWARE.....</b>	<b>48</b>
6.1 A simplicidade de interação .....	48
6.2 Passo a passo.....	48
6.2.1 'Splash Screen' .....	48
6.2.2 Selecionando arquivos com blocos estruturais estruturados .....	48
<b>7 CONSIDERAÇÕES FINAIS .....</b>	<b>59</b>
<b>8 REFERÊNCIA BIBLIOGRÁFICA LEVANTADA.....</b>	<b>60</b>

## 1. INTRODUÇÃO

### 1.1 Objetivo deste documento

Esta dissertação é referente ao Projeto de Conclusão de Curso, sua motivação, pesquisa, entendimentos, necessidades, hipóteses e desenvolvimento final.

### 1.2 Motivação

Observando o mercado atual com constantes mudanças em tecnologia e demanda, percebemos a necessidade de dinamismo e agilidade nos sistemas produtivos. A produção precisa possibilitar mudanças rápidas no processo ou em partes deste (ZHANG, 1999).

Dessa maneira, várias pesquisas na área de controle foram desenvolvidas chegando a teorias para o projeto e implementação de sistemas de controle modulares e distribuídos, em que o sistema de controle deixa de ser centralizado e passa a atuar em módulos dentro da planta, tal que quanto maior o número de módulos mais eficaz é este tipo de controle (YOUNG, 1995) ; (YUSUF, 1999).

Outro aspecto importante norteado por essas necessidades é o requisito de se ter uma ferramenta de modelagem que facilite o entendimento do modelo do processo em questão. Como resultado foram desenvolvidas ferramentas tais como Redes de Petri e linguagens derivadas, tais como MFG. Entretanto, há uma diferença significativa entre o modo que a linguagem do modelo é escrita e o modo que a linguagem é interpretada na arquitetura física. Ou seja, a linguagem de programação em seu destino final tem uma representação diferente da linguagem que descreve o modelo.

Visando a complementação desta teoria, a finalidade deste trabalho é o desenvolvimento e a implementação em software de conceitos de distribuição de controladores pelos módulos de uma arquitetura física, e distribuição dos códigos de controle do processo nestes controladores a partir de sua descrição formal, possibilitando maior agilidade na modificação do processo que atua na planta.

Isso permite, ao contrário de como é feito atualmente, alterar o procedimento empírico feito pelos projetistas que aumenta muito o custo da implementação do sistema de controle e impossibilita uma implementação rápida em tempo real. Com esta metodologia, o tempo de implementação e o custo poderão ser reduzidos permitindo mudanças mais freqüentes nos sistemas produtivos.

## 2. ENTENDIMENTOS

### 2.1 Definição do problema

O primeiro passo para o início do desenvolvimento do trabalho é o entendimento e definição do problema, que seria definir o que é essa “metodologia de distribuição de códigos de controle”.

Após estudos e discussões, identificou-se que, do modelo do projeto de controle de um processo para sua atuação na produção temos a seguinte configuração de acordo com a Fig. 1.

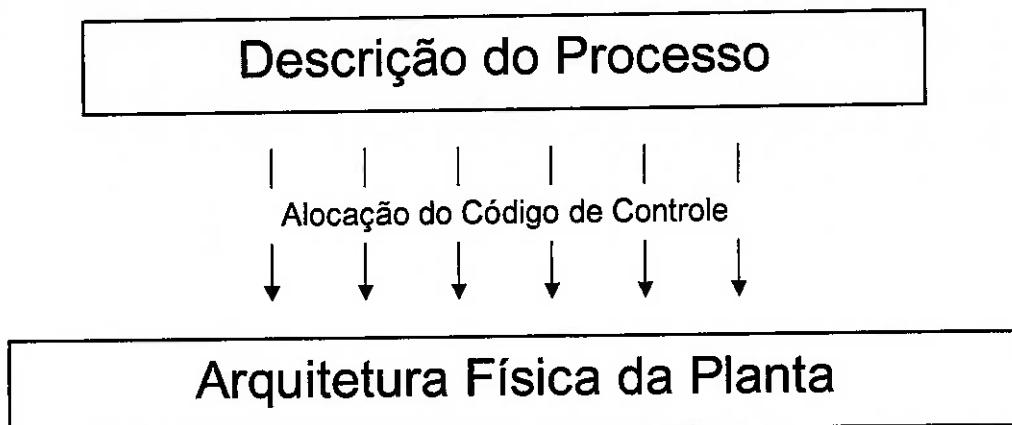


Figura 1. Definição esquemática do problema.

De posse da descrição do processo e da descrição da arquitetura da planta na qual se deseja realizar o controle do processo, é necessária uma distribuição inteligente dos códigos de controle entre os controladores disponíveis. Essa etapa de distribuição é a abordagem do trabalho, onde temos por objetivos implementar uma maneira mais rápida e inteligente de como é feito hoje para se realizar esta etapa. Isso será feito através da implementação em software de critérios existentes na literatura.

### 2.2 Unidade de Trabalho

Após algumas discussões e orientações, percebeu-se a necessidade de se definir uma “unidade de trabalho”, que é a menor divisão em que se pode distribuir o controle em uma planta.

Chegamos à conclusão que para a implementação do algoritmo, a menor unidade de trabalho a ser usada será um processo, isto é, um conjunto de atividades que descrevem um sequenciamento de ações que resultam em um produto final.

O motivo de tomar-se esta decisão é porque não se deve quebrar a estruturação do modelo de controle podendo causar travamento do sistema, uma vez que as partes de vários processos quando executadas podem dar origem a outro processo que pode comprometer a alocação dos recursos necessários de maneira não previsível.

### 2.3 Restrições para distribuição

A distribuição do controle em uma planta envolve inúmeras variáveis. Para realizarmos a distribuição do controle, é necessário definir qual variável e abordagem utilizaremos.

No projeto de sistemas de controle seqüenciais os requisitos em termos de velocidade de processamento de hardware de CPs para a realização do controle a partir de blocos funcionais, considera a constante de tempo de cada um destes blocos e os custos de processamento das instruções no controlador programável (CP) que for responsável pelo processamento.

### **3. CONCEITOS BÁSICOS**

Conceitos necessários para o entendimento do trabalho.

#### **3.1 Classificação dos sistemas de controle**

Os sistemas de controle, com relação ao tipo de variável controlada, podem ter três classificações diferentes (MIYAGI, 1996); (MORAES; CASTRUCCI, 2000): Sistemas de Controle de Variáveis Contínuas, que trabalha com variáveis quantitativas que podem assumir infinitos valores dentro de certos intervalos determinados. Sistemas de Controle de Variáveis Discretas, que trabalha com variáveis qualitativas que assumem valores determinados que não variam continuamente com o tempo. Sistemas de Controle Híbridos.

Para fins de síntese da lógica de controle, os sistemas de manufatura podem ser classificados como Sistemas a Eventos Discretos (SEDs), que são sistemas dinâmicos cuja mudança de estado ocorre em pontos discretos do tempo, em decorrência de eventos isolados, como por exemplo comandos para operação de máquinas ou sinais de ativação de sensores. Em contrapartida aos sistemas dirigidos pelo tempo, que são classicamente modelados por equações diferenciais e a diferenças, o comportamento dessa classe de sistemas pode ser matematicamente modelado por linguagens, que são conjuntos de cadeias finitas de símbolos representando todas as seqüências de eventos admitidas pelo sistema (HOPCROFT e ULLMAN, 1979).

#### **3.2 Componentes básicos de um sistema de controle**

Os sistemas, que podem ser de manufatura, transporte, comunicação, etc., podem ser caracterizados por uma dinâmica de ocorrência de eventos e controlados por dispositivos programáveis, sendo que a ocorrência de eventos pode alterar o estado do sistema dependendo unicamente das condições de transição serem satisfeitas (ou seja, conforme um procedimento pré-estabelecido seja atendido).

Um sistema de controle é dividido em seis componentes básicos (MIYAGI, 1996):

- Dispositivo de comando, onde o operador entra com comandos e ajustes de variáveis;
- Dispositivo de atuação, onde o sistema envia comandos para os atuadores;
- Dispositivo de detecção onde são recebidas informações das variáveis do processo;
- Dispositivo de monitoração, onde se tem uma visão geral das variáveis do processo;
- Dispositivos de controle onde é realizada a lógica de controle do processo.
- Dispositivo de realização do controle

A Fig. 2 a seguir ilustra o modelo formado por estes vários dispositivos.

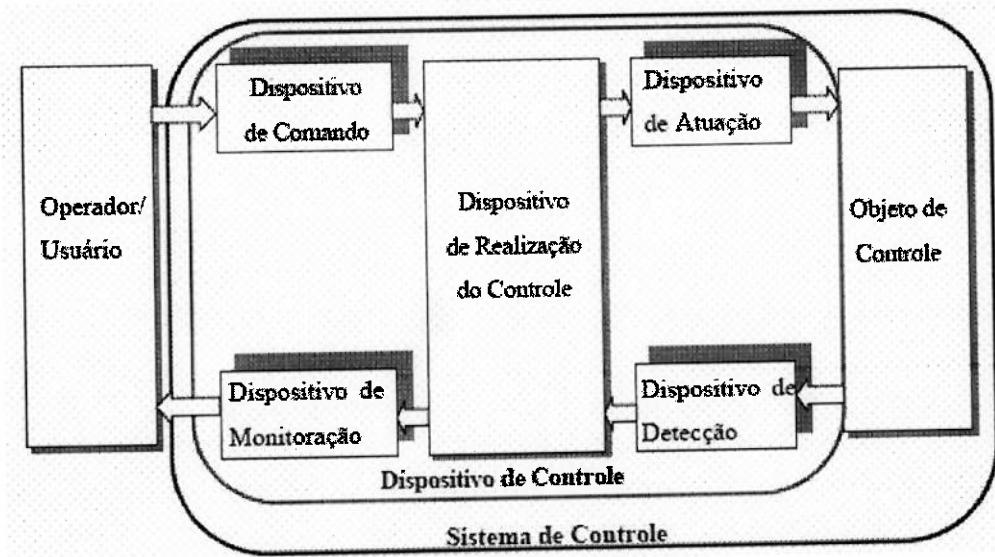


Figura 2. Diagrama conceitual básico do sistema de controle do SED.

Percebe-se uma divisão entre as funções de monitoração (operador) e de atuação do controle (objeto de controle).

Assim, podemos dizer que um controle em SED é o controle que executa ordenadamente cada estágio de controle baseado num procedimento pré-estabelecido ou numa lógica fixa que estabelece um procedimento.

### 3.3 Classificação dos sistemas de controle em função da topologia adotada

Em função da topologia adotada, o controle de sistemas pode ser classificado em (SANTOS FILHO, 1998):

- Sistema de Controle Centralizado, que são sistemas que possuem um único controlador e toda sua lógica é processada nele;
- Sistema de Controle distribuído, onde há dois ou mais controladores envolvidos na execução do controle.

### 3.4 Organização dos controladores em sistemas de controle distribuídos

Os controladores podem estar organizados nas seguintes formas básicas (Fig. 3):

- Arquitetura centralizada;
- Arquitetura hierárquica;
- Hierárquica modificada.

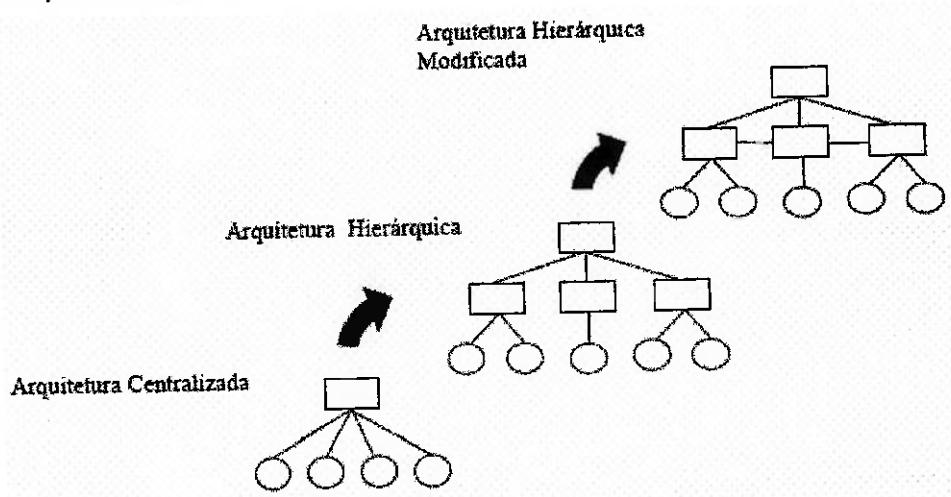


Figura 3. Evolução dos sistemas de controle.

### 3.5 Módulos de controle e controle modular

Como o projeto de sistemas de manufatura envolve uma grande quantidade de especificações, a abordagem de controle modular é freqüentemente usada para a síntese de controladores. Ao invés de se projetar um único supervisor monolítico que satisfaça todas as especificações, procura-se construir um supervisor para cada especificação (WONHAM e RAMADGE, 1988). Neste caso, deseja-se que os supervisores resultantes sejam modulares, isto é, que a ação conjunta dos supervisores modulares tenha o mesmo desempenho que a do supervisor monolítico. Quando essa propriedade é verificada, a abordagem de controle modular é bastante vantajosa no sentido de promover maior flexibilidade, maior eficiência computacional e segurança na aplicação do controle.

O módulo de controle é a divisão das tarefas envolvidas no controle de um SP em várias sub-tarefas (CAVALHEIRO, 2004). A síntese modular permite que problemas complexos possam ser decompostos em módulos mais simples, de forma a atribuir maior flexibilidade ao controle resultante (CURY, 2000). Além de ser mais facilmente construído, um controle modular costuma ser mais facilmente modificado, atualizado e corrigido (MIYAGI, 1996).

Quanto maior o número de módulos envolvidos na solução do problema, mais vantajosa será a abordagem modular em relação à monolítica (CURY, 2000).

### 3.6 Ferramentas empregadas na modelagem de SED

Modelos baseados em sistemas a eventos discretos são intensamente utilizados para descrever, analisar e controlar processos em ambientes de redes de computadores e manufatura. O que caracteriza este processo é o fato de sua dinâmica ser governada pela ocorrência de eventos discretos. Logo, a evolução de estados destes sistemas é baseado em regras que definem as condições para a ocorrência de eventos e o resultado destes eventos (CASSANDRAS & STRICKLAND, 1992). Assim, ferramentas como Redes de Petri e seus derivados como PFS/MFG podem ser muito bem utilizadas para modelar os sistemas em questão.

### 3.6.1 Redes de Petri e representação de SED por Redes de Petri

Rede de Petri é uma ferramenta gráfica e matemática de modelagem que pode ser aplicada a diversos tipos de sistemas, apresentando um bom nível de abstração em relação a outros modelos gráficos. Permite a modelagem de sistemas dinâmicos discretos com grande poder de expressividade, possibilitando representar com facilidade todas as relações de causalidade entre processos em situações de seqüencialidade, conflito, concorrência e sincronização. Além disso, as Redes de Petri possibilitam a verificação do correto funcionamento do sistema especificado (MIYAGI, 1996).

#### 3.6.1.1 Vantagens das Redes de Petri

As vantagens das Redes de Petri são:

- permite modelar sistemas com um nível de complexidade maior;
- gera modelos mais compactos;
- extensões que incorporam parâmetros temporais.

Seu formalismo matemático permite a análise precisa do modelo, além da visualização dos processos e a comunicação entre eles (MIYAGI, 1996).

#### 3.6.1.2 Composição das Redes de Petri

As Redes de Petri são formadas por dois elementos, um ativo denominado de transição, e outro passivo denominado lugar. Os lugares correspondem às variáveis de estado e as transições às ações realizadas pelo sistema. Os arcos que interligam lugares às transições correspondem à relação entre as condições verdadeiras, que em um dado momento, possibilitam as execuções das ações, enquanto os arcos que interligam transições aos lugares representam a relação entre as ações e as condições que se tornam verdadeiras com a execução das ações. É possível observar um exemplo simplificado de uma rede de Petri e seus elementos na Fig. 4.



Figura 4. Elementos de uma Rede de Petri.

### 3.6.2 PFS/MFG

A metodologia PFS/MFG está baseada no procedimento de refinamento gradativo do PFS, substituindo-se uma atividade ou um distribuidor por uma rede.

#### 3.6.2.1 PFS - Production Flow Schema

As Redes de Petri são muito úteis para representar sistemas de pequeno porte com facilidade mostrando o relacionamento entre as partes.

Entretanto para sistemas maiores e mais complexos, as relações entre os componentes deixam de ser tão visíveis. De modo a sanar essa dificuldade, foi proposto por Miyagi [1996] uma metodologia denominada PFS (Production Flow Schema), que ao invés de representar de imediato o modelo em nível detalhado, adota-se primeiro a modelagem do sistema em um nível conceitual para em seguida detalhar cada elemento gradativamente até chegar em um nível de detalhamento que se possa representar o sistema utilizando-se grafos específicos para o controle como MFG.

Assim, as redes PFS têm como objetivo principal mostrar explicitamente os componentes que formam o sistema e quais relações existem entre cada um deles, deixando marcas ou marcações para outros níveis de detalhamento. As inscrições em seus elementos indicam apenas quando eles operam e como operam (MIYAGI, 1996).

##### 3.6.2.1.1 Elementos estruturais do PFS

O PFS pode ser representado por 3 elementos básicos:

- elementos ativos, ou atividades. Realizam ações, modificações nos itens; representadas por um bloco delimitado por dois colchetes ( [ ] );
- elementos passivos, ou distribuidores. Não realizam transformação, mas são capazes de armazenar e distribuir os itens; representados por círculos ( O )
- arcos, que representam as relações entre os elementos ativos e passivos e indicam o sentido do fluxo de itens. Os arcos não representam necessariamente um elemento real do sistema, mas descreve uma relação lógica, abstrata entre os componentes. Pode-se ter:
  - Arcos orientados das atividades aos distribuidores ( [ ] → O );
  - Arcos distribuidores das atividades às atividades ( O → [ ] ).

O grafo (diagrama) resultante da representação do fluxo de itens de um sistema produtivo composto pelos elementos acima é chamado PFS.

As interpretações (inscrições na rede) são simples descrições dos elementos do sistema. As inscrições nos distribuidores descrevem quais itens são armazenados nestes elementos. As inscrições nas atividades indicam quando e como ocorrem estas alterações nos itens.

Assim, as redes PFS mostram explicitamente os componentes que formam o sistema e quais relações existem entre cada uma delas, exemplificado pela Fig. 5.

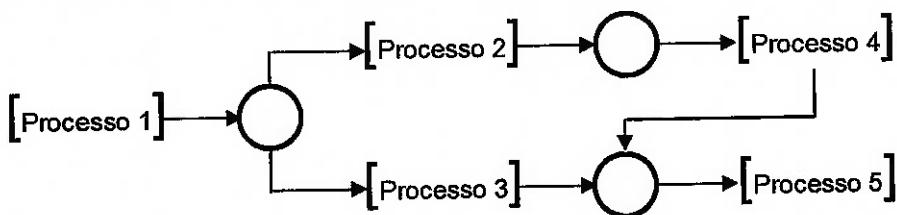


Figura 5. Exemplo de um grafo FPS.

### 3.6.2.2 MFG – Mark Flow Graph

O Mark Flow Graph (MFG) é um grafo derivado da Rede de Petri, onde as funções de entrada e de saída são devidamente consideradas, visando a modelagem e o controle dos SED de maneira mais simples e eficaz (MIYAGI, 1996). Um exemplo de MFG é apresentado na Fig. 6.

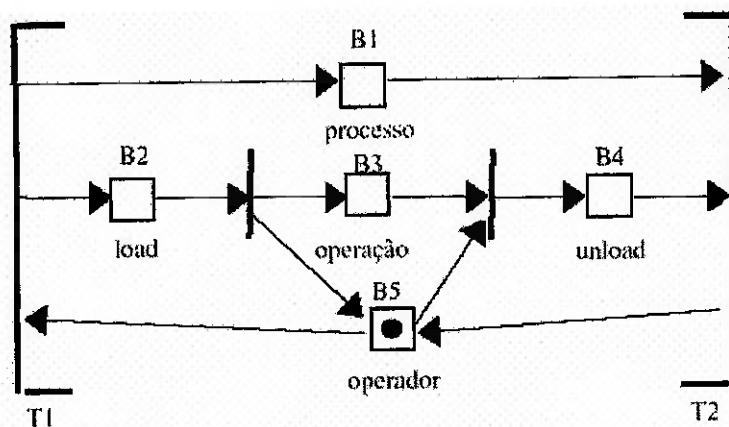


Figura 6. Exemplo de um modelo em MFG.

#### 3.6.2.2.1 Características do MFG

Segundo Santos Filho (1993), o MFG possui as seguintes características:

- a capacidade de descrição do comportamento dinâmico do sistema de forma consistente;
- a capacidade de realização de uma descrição funcional para uma estruturação do processo de modelagem;
- disponibilidade de modelos matemáticos eficientes para a análise do sistema e seu projeto;
- capacidade de realização adequada e eficiente de interfaces homem-máquina e máquina-máquina nos diversos níveis.

### 3.6.2.2.2 Elementos estruturais do MFG

Elementos estruturais do MFG (MIYAGI, 1996; SANTOS FILHO, 1993):

- box: indica uma condição e é representado por um bloco quadrado;  
- transição: indica um evento e é representada por uma barra vertical;  
- arco orientado: conecta boxes e condições para indicar a relação entre uma condição e os pré e pós-eventos que o definem. Não existe limite para o número de arcos que entram ou saem de um box ou de uma transição, apenas deve-se observar que entre um par box-transição deve existir apenas um arco que os conecte.

- marca: indica a manutenção de uma condição e é representada por um ponto negro no interior do box correspondente a esta condição;

- porta: habilita ou inibe a ocorrência dos eventos correspondentes às transições, podendo ser portas habilitadoras (um arco que possui um círculo negro na extremidade conectada à transição), ou portas inibidoras (um arco que possui um círculo branco na extremidade conectada à transição), conforme sua natureza. Estas, por sua vez podem ser classificadas em porta externa ou porta interna de acordo com a origem do sinal. A origem do sinal de uma porta interna é um box e a origem do sinal de uma porta externa é de alguma entidade que não faz parte do grafo, ou seja, ela indica a entrada de um sinal binário gerado por algum dispositivo externo. A porta inibidora habilita a transição à qual está conectada quando seu sinal for “1”. Quando o sinal da porta inibidora for “0”, a transição conectada a esta porta é inibida;

- arco de sinal de saída: este arco envia um sinal binário do box para um dispositivo externo ao grafo e é representado por uma linha que conecta esses dois elementos. Quando houver uma marca no box, o sinal é “1”, do contrário é “0”.

As portas e arcos de sinal de saída são formas de representação explícitas das condições de controle.

Cada um destes elementos está representado na Fig. 7.

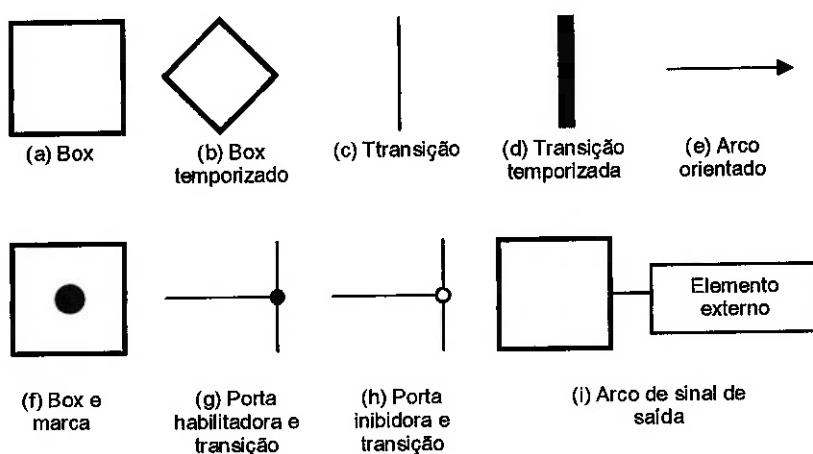


Figura 7. Elementos estruturais do MFG.

### **3.6.2.2.3    Dinâmica do MFG**

De acordo com (MIYAGI, 1996), os conceitos de disparo e de regras de marcação são basicamente os mesmos das Redes de Petri. De fato o MFG pode ser considerado uma interpretação especial de uma classe de Rede de Petri do tipo condição-evento.

A disposição das marcas no grafo define uma marcação. A marcação atual do grafo representa o estado em que o sistema se encontra.

A marcação inicial é o arranjo inicial das marcas. Na marcação inicial é permitido no máximo uma marca em cada box. Se a marcação inicial não contiver nenhuma marca, é denominada de marcação inicial zero ou nula.

A evolução dinâmica do sistema é caracterizada pela alteração nos estados, causada pela ocorrência de eventos. Para que isso seja representado no MFG, são definidas regras de habilitação e de disparo das transições.

### **3.6.2.2.4    Transição habilitada**

Uma transição está habilitada para disparo se as seguintes condições forem todas satisfeitas:

- Não existe box com marcas no lado da saída;
- Não existe box sem marcas no lado da entrada;
- Não existe porta habilitadora interna que esteja no estado de desabilitação;
- Não existe porta inibidora interna que esteja no estado de inibição.

Uma transição que está habilitada para disparo é chamada transição habilitada, caso contrário é chamada de transição desabilitada.

### **3.6.2.2.5    Regras para disparo da transição**

Para que ocorra disparo da transição habilitada é necessário que duas condições sejam satisfeitas, caracterizando-a como condição disparável

- Todas as portas habilitadoras externas estão no lado habilitado;
- Todas as portas inibidoras externas estão no estado desabilitado (ou seja, não estão inibindo).

Se uma transição é disparável, então ocorre seu disparo de forma imediata e instantânea (supondo que não exista concorrência ou atraso de tempo).

O disparo de uma transição retira as marcas de todos os boxes de entrada e coloca marcas em todos os boxes de saída.

### 3.6.2.2.6 Extensão para conceito de tempo

A extensão que considera a inclusão do conceito de tempo no MFG envolve os seguintes elementos:

- box temporizado: quando uma marca aparece nesse tipo de box, ela deve necessariamente permanecer neste box um intervalo de tempo;

- transição temporizada: uma vez que todas as condições de disparo estejam satisfeitas, esta transição só dispara após o decorrer de um intervalo de tempo. Se durante esse tempo umas das condições deixa de ser satisfeita, a contagem do tempo é anulada. Será reiniciada somente após todas as condições estarem novamente satisfeitas.

Com base nos elementos acima descritos, é possível gerar macro elementos, isto é, uma modularização do MFG efetiva para modelagem, análise e controle de sistemas produtivos (Fig. 8).

Em seguida apresentamos alguns macro-elementos considerados:

- box capacidade: possui um símbolo N associado que indica sua capacidade de alojar em seu interior N marcas;
- box agrupador: tem a função equivalente a uma montagem em que N marcas entram e apenas uma sai;
- box dispersor: tem a função similar a de uma desmontagem, em que uma marca entra e N marcas saem.

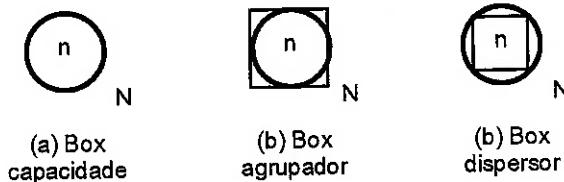


Figura 8. Macro-elementos do MFG (*n* indica o número atual de marcas dentro do lugar)

Uma das principais características do MFG que o torna uma classe de Redes de Petri mais efetiva em aplicações práticas do que a original é o fato de ser desenvolvido especificamente para a modelagem e realização de controle, considerando, por exemplo, as seguintes características desejáveis nos sistemas de controle:

- “safeness” no controle de SED é uma característica indispensável, e através da aplicação de MFG, as condições que garantem o “safeness” são intrinsecamente consideradas durante o projeto e análise do sistema;

- no MFG existe um controle detalhado das marcas no interior de cada box, e com isto o modelo pode ser tratado também por uma lógica binária. O MFG convertido para esta lógica binária pode ser diretamente utilizado para programar um CP, ou através de uma interpretação adequada para organizar sistemas mais complexos;

- com o MFG, o controle de SED evolui conforme a troca de sinais entre o controlador (grafo MFG) e o objeto de controle, isto é, o controlador envia sinais (através dos arcos de saída) para os dispositivos que fazem com que uma tarefa seja realizada, e ao receber o sinal de confirmação do término da tarefa gerado pelos dispositivos (através de portas externas), passa para a etapa seguinte do controle.

No MFG as tarefas são descritas pelos boxes que podem enviar sinais de estado pelos arcos de saída, o início e o término das tarefas são descritas pelas transições, condicionados aos sinais das entidades externas. Desta forma, o MFG pode representar adequadamente não só o sistema de controle de SED, mas também as conexões entre o controlador e o objeto de controle (MIYAGI, 1996).

## 4. METODOLOGIA

### 4.1 Entradas necessárias para o algoritmo

#### 4.1.2 Entradas referentes ao algoritmo de alocação física dos CP's

Para entrada do algoritmo de alocação física dos CP's o usuário deverá entrar com as coordenadas da posição dos componentes da instalação que trocam informações com o sistema de controle.

Deverá entrar também com os custos variáveis (por metro) e fixos relativos à transmissão dos sinais, como custo de cabos, blindagem, painéis, etc.

Com estas informações o algoritmo pode processá-las e gerar a divisão do sistema na área de controle.

#### 4.1.3 Entradas referentes ao algoritmo de alocação de blocos funcionais

Para o algoritmo de alocação de blocos funcionais serão necessárias informações referentes aos CP's e aos Blocos Funcionais.

As características do CP escolhido serão obtidas a partir de uma base de dados onde serão cadastrados tantos CP's quanto desejado. Quando o usuário escolher o CP o próprio software buscará na base as informações que desejar.

Quanto aos blocos funcionais estes já estarão estruturados segundo a norma IEC 61499 e as operações serão descritas em uma linguagem composta por estrutura de árvores:

#### 4.1.4 Estrutura de árvores

A descrição do processo estruturado (blocos funcionais) que será uma das entradas do algoritmo terá suas instruções descritas na forma de árvores, onde cada árvore representará um "rung" de ladder.

Os nós das árvores representam operações ou variáveis segundo a codificação seguinte:

#### OPERADORES DE ATRIBUIÇÃO

SET = 0x0201  
RESET = 0x0202  
STORE = 0x0203  
INC = 0x0204  
DEC = 0x0205

#### OPERADORES LÓGICO-ARITMETICO:

AND = 0x8201  
OR = 0x8202  
LESS = 0x8203  
MORE = 0x8204  
EQUAL = 0x8205  
DIFFER = 0x8206  
LEQUAL = 0x8207  
MEQUAL = 0x8208  
NOT = 0x8100

#### OPERANDOS VARIÁVEIS

FLAG = 0x9400  
INPUT = 0x9800

OUTPUT = 0x9C00

#### OPERANDOS CONSTANTES

CONSTANT = 0x9000

Como, por exemplo, o "rung" abaixo, Fig. 8.

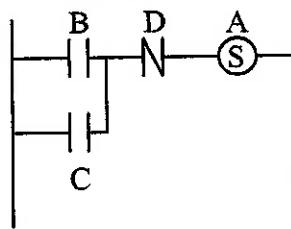


Figura 8. "Rung" de código Ladder.

E pode ser representado pela árvore, Fig. 9.

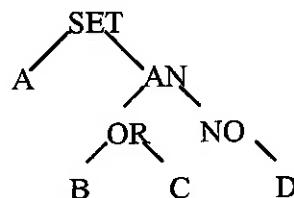


Figura 9. Árvore estruturada descrevendo "rung" de Ladder.

Portanto a entrada da descrição do bloco funcional é uma estrutura de dados de acordo com a descrição acima.

O código para geração da árvore não faz parte do escopo deste trabalho, sendo gerado em outros estudos, e trata-se apenas de uma descrição semântica do código ladder, o que é suficiente para obtenção das informações para os cálculos necessários no algoritmo de alocação dos blocos funcionais. Mas a seguir demonstraremos como será descrita a rede de petri que dá origem a esta árvore.

A descrição é feita da seguinte maneira:

- Com os lugares e transições nomeados de modo que o CLP identifique, é criado para cada transição um “rung” equivalente à expressão (usando notação prefixa de operadores):

```
STORE(TRANSIÇÃO,AND(AND(PRÉ-CONDIÇÕES),
AND(NOT(PÓS-CONDIÇÕES)),
AND(CONDIÇÕES-HABILITADORAS),
AND(NOT(CONDIÇÕES-INIBIDORAS))))
```

Obs. As condições habilitadoras são quaisquer condições ou sinais ligados à transição por arcos habilitadores. Usando-se a mesma analogia para condições inibidoras. A linha atribui os valores lógicos TRUE ou FALSE para a transição caso ela esteja habilitada ou não.

- Em seguida é gerada uma lista de conflitos para cada conflito existente na rede onde é gerada uma lista com todas as transições contidas neles.

Temos a seguir uma rede exemplo e o seu código que após um processamento será representado como árvores, Fig. 10.

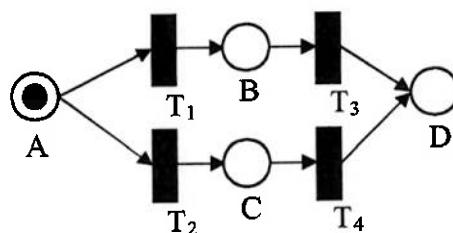
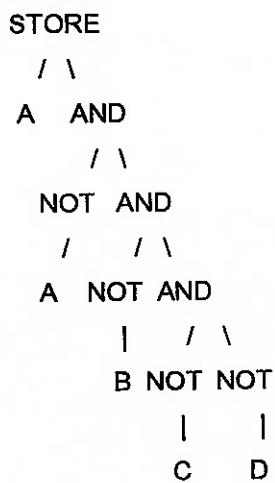


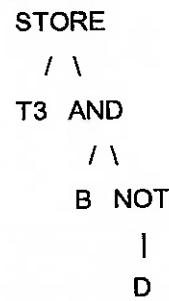
Figura 10. Rede de Petri que será representada por diagramas em árvore.

```
1  LOOP_INFINITO {
2      LÊ_SINAIS_ENTRADA()
3      IF A == FALSE AND
4          B == FALSE AND
5          C == FALSE AND
6          D == FALSE THEN
7          A = TRUE;
8
9          T1 = A AND NOT(B)
10         T2 = A AND NOT(C)
11         T3 = B AND NOT(D)
12         T4 = C AND NOT(D)
13
14         IF T1 == TRUE THEN
15             RESET A
16             SET B
17             RESET T2
18             ENDIF
19             IF T2 == TRUE THEN
20                 RESET A
21                 SET C
22                 RESET T1
23                 ENDIF
24                 IF T3 == TRUE THEN
25                     RESET B
26                     SET D
27                     RESET T4
28                     ENDIF
29                     IF T4 == TRUE THEN
30                         RESET C
31                         SET D
32                         RESET T3
33                         ENDIF
34
35             ESCREVE_SINAIS_SAIDA()
36 }
```

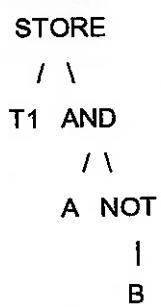
Árvore 1



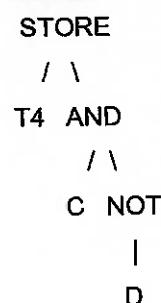
Árvore 4



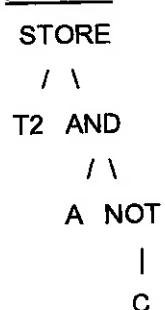
Árvore 2



Árvore 5



Árvore 3



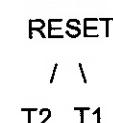
Árvore 6



Árvore 7



Árvore 8



Árvore 9

RESET

/ \

A T2

Árvore 13

SET

/ \

D T3

Árvore 10

SET

/ \

C T2

Árvore 14

RESET

/ \

T4 T3

Árvore 11

RESET

/ \

T1 T2

Árvore 15

RESET

/ \

C T4

Árvore 12

RESET

/ \

B T3

Árvore 16

SET

/ \

D T4

Árvore 17

RESET

/ \

T3 T4

#### **4.2 Critérios envolvidos no desenvolvimento do algoritmo de alocação de blocos funcionais**

Para realização da alocação dos blocos funcionais em uma arquitetura física existem inúmeras variáveis complexas envolvidas.

Devido a isso há a necessidade do projetista definir as variáveis que determinam esta alocação da melhor maneira, aplicando os conceitos de sistemas antropocêntricos (CAVALHEIRO, 2004).

Portanto, o algoritmo tratará da automatização da aplicação das regras e procedimentos que auxiliarão o projetista (usuário) a realizar a melhor alocação dos blocos funcionais.

A técnica PFS/MFG utilizada para modelar sistemas de forma estruturada em blocos funcionais não aborda o problema da alocação de blocos funcionais nos equipamentos que realizam o controle do sistema produtivo. Torna-se necessário a utilização de outras técnicas que processem essas informações para solucionar o problema de maneira rápida e adequada (CAVALHEIRO, 2004) e que neste trabalho serão implementadas em software.

Para garantir esta alocação adequada antes de elas serem executadas, são necessárias três abstrações principais para modelar o sistema (SANTOS FILHO, 2001); (NORONHA, 2001).

- Definição das tarefas (blocos funcionais) do sistema, que no caso serão tratados entre as variáveis de entrada do algoritmo.
- Reserva de recursos de CPU para garantir o processamento das tarefas, que será realizada pelo algoritmo implementando no software com interação com o usuário.
- Consideração das restrições temporais de processamento, implementada no algoritmo.

Com a aplicação das abstrações citadas o algoritmo visa auxiliar na alocação ótima dos blocos funcionais, garantindo a qualidade do projeto de automação do sistema produtivo. Mas para isso ser realizado, torna-se necessário a definição de uma arquitetura de hardware para processá-los (CAVALHEIRO, 2004), que será também implementada no algoritmo.

#### **4.3 Alocação dos blocos funcionais de um sistema produtivo**

A seguir segue uma descrição das considerações e características referentes a alocação dos blocos funcionais proposto em (CAVALHEIRO, 2004).

Para realização das tarefas de um sistema, deve-se levar em conta o tempo de resposta que garanta a execução do controle. Este tempo é definido como sua constante de tempo (LEWIS, 1989) e se trata do período em que a lógica do sistema pode ser processada sem que sejam comprometidos os critérios temporais referentes à execução do sistema produtivo.

Portanto, o tempo que um CP leva para processar blocos funcionais, deve atender às necessidades de controle impostas por um sistema produtivo, considerando as constantes de tempo que regem o comportamento dinâmico do sistema que está sendo controlado. Devem ser levadas em conta as exigências das constantes de tempo dos diversos processos que são envolvidos em um SP sendo possível que todos os blocos funcionais sejam processados em tempo hábil (CAVALHEIRO, 2004).

Existem algumas características que devem ser levadas em consideração na definição da arquitetura de controle para alocação de blocos funcionais em CP's (CAVALHEIRO, 2004):

- Indeterminismo em relação ao tempo de execução das tarefas; que se trata do fato de que o tempo de processamento de um bloco funcional depende diretamente do CP que irá processá-lo, onde somente se conhece a constante de tempo associada ao bloco funcional.

- Indeterminismo em relação a seqüência de processamento dos blocos funcionais. Não existe sincronismo entre os processos a serem executados que permita saber a seqüência de requisições do CP, sabendo-se somente que todos os blocos funcionais serão processados e atualizados em tempo hábil para efetuar o controle.

O sistema de distribuição deve considerar o sequenciamento de blocos funcionais imposto pela lógica de controle do SP e possuir flexibilidade para definir a melhor designação destes em CP's disponíveis.

O sistema de distribuição deve ser hábil para alocar blocos funcionais segundo um dado critério, no caso o custo de processamento, resultando em uma distribuição homogênea em função da demanda de processamento exigida por estes e dos recursos de cada bloco funcional em função do CP eleito para processá-lo.

Deve ser considerado o fluxo de informações na rede de comunicação, para evitar que ocorra saturação nesta.

#### **4.4 Uma descrição geral do algoritmo a ser implementado**

O algoritmo a ser implementado visa complementar a fase final do projeto de um sistema de controle de um sistema produtivo.

Após a fase de modelagem do sistema, é feita a criação e modelagem do algoritmo de controle do sistema, que seguindo a metodologia apresentada, é feita utilizando a técnica PFS/MFG já apresentada. Em seguida este modelo do algoritmo de controle é estruturado e padronizado conforme a norma IEC 61499.

Em seguida, este modelo padronizado e descrito computacionalmente é traduzido para uma linguagem de representação de Redes de Petri. Essa representação será um dos pontos de partida (entrada) para o algoritmo a ser implementado neste trabalho. A outra entrada será a base de dados com as características dos CP's onde o próprio algoritmo ou o usuário os selecionará.

A partir disto, segue uma seqüência de atividades a serem realizadas pelo algoritmo em interação com o usuário (CAVALHEIRO, 2004) adaptada para a implementação do software, Fig. 11.

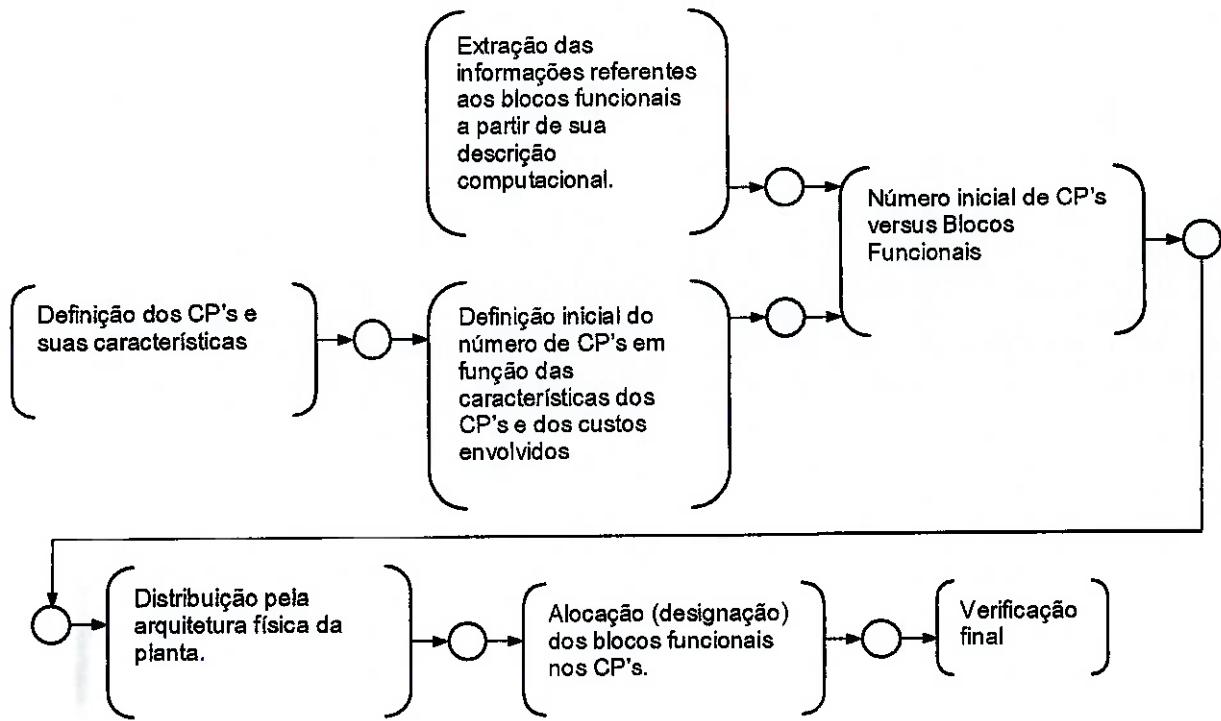


Figura 11. Esquematização de algoritmo de designação.

Em seguida é apresentada uma breve descrição das atividades e, de maneira geral, como serão implementadas:

#### 4.4.1 Extração das informações referentes aos blocos funcionais

Trata-se da leitura do modelo computacional da lógica do bloco funcional possibilitando a extração pelo software dos dados necessários dos blocos funcionais (conforme o modelo a ser apresentado adiante) para o algoritmo realizar sua alocação no CP's.

#### 4.4.2 Definição dos tipos de CP's e suas características

É a definição dos CP's a serem utilizados para realizar o controle. Isto será feito pelo usuário que poderá selecionar o CP desejado selecionando empiricamente baseando-se no custo e outras variáveis. Esta seleção poderá ser feita através de uma base de dados onde será possível o cadastro de CP's de vários modelos e marcas e de suas características.

#### **4.4.3 Definição dos CP's**

Define-se o modelo do CP a ser utilizado, considerando as características de hardware envolvidas no sistema de controle onde será levado em conta os instrumentos do sistema produtivo (como sensores, atuadores, etc.), quantidade de sinais de entrada e saída, bem como custo do CP e de sua alocação física. Esta seleção será feita pelo usuário que poderá selecionar entre vários CP's a partir de uma base de dados que contenha informações com as características dos CP's

#### **4.4.4 Distribuição na arquitetura física da planta**

Etapa onde o software realiza a distribuição dos CP's na arquitetura física da planta. Consiste na divisão do sistema produtivo em áreas de controle cuja quantidade definirá o número de CP's a serem utilizados.

Nesta etapa, é realizado também o posicionamento dos CP's de modo que o custo de envio e recebimento de sinais seja o menor possível.

#### **4.4.5 Alocação (designação) dos blocos funcionais dos CP's**

Consiste na alocação ótima dos Blocos Funcionais, considerando seus custos de processamento, nos CP's escolhidos. Isto é possível através da implementação em algoritmo do método de designação proposto por (NOVAES, 1978); (BRONSON, 1985); (PUCCINI; PIZZOLATO, 1987).

#### **4.4.6 Verificação final**

Nesta etapa é apresentado o resultado final do algoritmo que implementa o método de designação permitido ao usuário reallocar blocos funcionais nos CP's de maneira que seja mais conveniente ao controle do sistema produtivo do que a alocação proposta. Isso é permitido pois lembrando que as atividades implementadas no algoritmo apresentado e implementado envolvem inúmeras variáveis e abordagens, sendo a abordagem apresentada aqui levando em conta o custo computacional é a mais comum mas nem sempre gera os melhores resultados em termos funcionais para o sistema produtivo, cabe ao projetista (usuário) uma análise crítica e decisão final.

### **4.5 Descrição do algoritmo de alocação física dos CP's e designação de blocos funcionais em CP's**

#### **4.5.1 Alocação física dos CP's**

Primeiramente trataremos do problema de alocação física dos CP's.

Isso é feito visando a definição do número de CP's que realizarão o controle. Para isto é necessário considerar que todas as exigências do sistema de controle sejam atendidas buscando o menor custo possível.

Para se efetuar esta tarefa o algoritmo a ser implementado considera o método SLP (Systematic Layout Planning) (MUTHER, 1978) que visa buscar o menor custo em função do arranjo físico do sistema. Para isso, considera-se o modelo físico do sistema, considerando a disposição física dos instrumentos no sistema produtivo, e todos os custos envolvidos para levar os sinais gerados por estes, até o(s) controlador(es) que vão efetuar o controle, levando em consideração o custo atribuído a esses equipamentos. Trata-se de uma solução voltada para o problema de "layout" (CAVALHEIRO, 2004).

O algoritmo proposto por (CAVALHEIRO, 2004) baseia-se na hipótese de que, partindo do arranjo físico já existente, a divisão da planta em áreas de controle é realizada, de tal modo que todos os elementos dentro de determinada área sejam controlados por um único CP. A planta é dividida em tantas áreas quanto forem necessárias, levando-se em consideração o custo para a transmissão dos sinais e o custo do CP. Pelo número de áreas de controle obtido obtém-se uma aproximação inicial do número de CP's a se utilizar, que será posteriormente confirmada pela verificação dos recursos de processamentos disponíveis nesta configuração pelo algoritmo de alocação de blocos funcionais.

Essa abordagem é muito interessante na minimização de custos no projeto de sistemas de controle de grande porte.

#### 4.5.2.1 Conceitos e funções envolvidas na alocação física dos CP's

A seguir segue uma apresentação de conceitos propostos em (CAVALHEIRO, 2004) que foram utilizados no desenvolvimento do algoritmo de alocação física e também serão utilizados em sua implementação.

Para iniciar, são considerados N instrumentos ligados a um único CP e a partir disto objetiva-se definir o local de posicionamento do CP de maneira que o custo de transmissão de informações (cabeamento) seja o menor possível.

A princípio posiciona-se o CP de modo que este fique mais próximo dos instrumentos que possuem maior custo de transmissões de sinais.

Apresenta-se a seguir a proposta de uma equação para realizar o cálculo das coordenadas do ponto de fixação do painel (CP):

$$E_g = O + \sum_{i=1}^N \left( \frac{D_{mi}}{\sum_{j=1}^N D_{mj}} \right) \cdot (P_i - O) \quad (4.1)$$

Isso é feito visando a definição do número de CP's que realizarão o controle. Para isto é necessário considerar que todas as exigências do sistema de controle sejam atendidas buscando o menor custo possível.

Para se efetuar esta tarefa o algoritmo a ser implementado considera o método SLP (Systematic Layout Planning) (MUTHER, 1978) que visa buscar o menor custo em função do arranjo físico do sistema. Para isso, considera-se o modelo físico do sistema, considerando a disposição física dos instrumentos no sistema produtivo, e todos os custos envolvidos para levar os sinais gerados por estes, até o(s) controlador(es) que vão efetuar o controle, levando em consideração o custo atribuído a esses equipamentos. Trata-se de uma solução voltada para o problema de "layout" (CAVALHEIRO, 2004).

O algoritmo proposto por (CAVALHEIRO, 2004) baseia-se na hipótese de que, partindo do arranjo físico já existente, a divisão da planta em áreas de controle é realizada, de tal modo que todos os elementos dentro de determinada área sejam controlados por um único CP. A planta é dividida em tantas áreas quanto forem necessárias, levando-se em consideração o custo para a transmissão dos sinais e o custo do CP. Pelo número de áreas de controle obtido obtém-se uma aproximação inicial do número de CP's a se utilizar, que será posteriormente confirmada pela verificação dos recursos de processamentos disponíveis nesta configuração pelo algoritmo de alocação de blocos funcionais.

Essa abordagem é muito interessante na minimização de custos no projeto de sistemas de controle de grande porte.

#### 4.5.2.1 Conceitos e funções envolvidas na alocação física dos CP's

A seguir segue uma apresentação de conceitos propostos em (CAVALHEIRO, 2004) que foram utilizados no desenvolvimento do algoritmo de alocação física e também serão utilizados em sua implementação.

Para iniciar, são considerados N instrumentos ligados a um único CP e a partir disto objetiva-se definir o local de posicionamento do CP de maneira que o custo de transmissão de informações (cabeamento) seja o menor possível.

A princípio posiciona-se o CP de modo que este fique mais próximo dos instrumentos que possuem maior custo de transmissões de sinais.

Apresenta-se a seguir a proposta de uma equação para realizar o cálculo das coordenadas do ponto de fixação do painel (CP):

$$E_g = O + \sum_{i=1}^N \left( \frac{D_{mi}}{\sum_{j=1}^N D_{mj}} \right) \cdot (P_i - O) \quad (4.1)$$

Tal que:

$N$  é o número de instrumentos do sistema produtivo.

$O$  é a origem do sistema do plano  $(\vec{O}, \vec{e_1}, \vec{e_2})$

$E_g$  é a posição do painel de controle  $(\vec{O}, \vec{e_1}, \vec{e_2})$

$D_{mi}$  é o custo do cabo (material gasto) para a transmissão do sinal do instrumento  $E_i$

$D_{mj}$  é o custo do cabo (ou material gasto) para a transmissão do sinal do instrumento  $E_j$

$P_i$  é a posição do instrumento  $E_i$  em  $(\vec{O}, \vec{e_1}, \vec{e_2})$  no sistema produtivo.

Esta equação (4.1) caracteriza o custo das distâncias  $Op_i$ , onde se utiliza a abordagem que consiste em relacionar o conceito “consumo de material” em cada local que caracteriza a localização de um instrumento. A partir disso consegue-se definir uma posição estratégica em que os custos são mínimos. Essa posição é denominada “Centro Virtual do Custo Global” (CVCG).

Estendendo-se o conceito para  $M$  painéis de controle, temos o Centro Virtual do Custo Regional que introduz o procedimento de divisão do sistema em áreas de controle onde cada CP será alocado de maneira a gerar o menor custo de transmissão de sinais.

Para se obter uma diretriz para a partição em áreas de controle pode-se utilizar critérios como a taxa de ocupação do CP em relação a quantidade de entradas e saídas, entre outras.

O custo para o painel da área de controle é definido por:

$$D_{MR} = \frac{\sum_{j=1}^N D_{mj}}{T} \quad (4.2)$$

Tal que:

$T$  corresponde ao número total de painéis de controle.

Com relação ao problema de quantificação do custo de cabeamento e o custo relacionado ao painel de controle, pratica-se a expressão abaixo (4.3) (CAVALHEIRO, 2004), baseada na proposta de Muther (1978), que é utilizada em casos onde o objetivo é fazer uma análise dos problemas de arranjo físico baseado nos conceitos fundamentais:

- Inter-relação: grau relativo de dependência entre as partes;
- Espaço: quantidade, tipo e forma dos itens a serem posicionados;

- Ajuste: arranjo das áreas e equipamentos da melhor maneira possível.

$$ITT_i = \sum_{j=1}^{M_i} D_{mj} \cdot d_{ij} \quad (4.3)$$

Tal que:

$i = 1, 2, \dots, N$ ; tal que  $N$  é o número de painéis.

$ITT_i$  é o índice associado a um painel de controle para delimitar uma área de controle em função do custo de um painel  $E_i$ .

$M_i$  é o número de instrumentos  $E_i$  associados ao painel de controle, que não podem ultrapassar o número máximo de E/S do CP independente se o custo  $ITT$  foi alcançado ou não.

$D_{mj}$  é o custo do cabo do painel  $E_i$  para o instrumento  $E_j$ .

$d_{ij}$  é a distância entre o painel  $E_i$  (origem) e o instrumento  $E_j$  (destino).

O procedimento apontado para a divisão do sistema produtivo em  $M$  regiões composta por  $M$  painéis de controle baseia-se nos critérios a seguir (CAVALHEIRO, 2004):

- Deve haver homogeneidade quanto aos índices de alocação de instrumentos na definição das áreas de controle.
- Cada área de controle deve ter um local estratégico para abrigar seu respectivo painel.

Com relação às hipóteses levantadas para o desenvolvimento da conduta para a divisão do sistema produtivo em áreas de controle têm-se (CAVALHEIRO, 2004):

- Não existe distinção em relação a qual instrumento pode ser alocado em um painel de controle (CP). Todos os CP's são considerados, operacionalmente, equivalentes.
- O critério deve ser único para delinear-se uma área de controle: custos locais de cabeamento/painéis de controle e capacidade de alocação de sinais E/S nos CP's.
- A probabilidade de existir troca de informações entre CP's via rede de comunicação não deve influenciar os critérios que definem os limites de cada local.

Apresentando os conjuntos de procedimentos sugeridos em (CAVALHEIRO, 2004) para definir as áreas de controle de um sistema e que serão implementados no software:

1. Primeiramente, determina-se o CVCG do sistema global, de acordo com a expressão 4.1.

2. Estima-se o valor do cabeamento por área de controle  $ITT_{mr}$  para cada tipo de CP em função de seu custo e de seu painel de acordo com a expressão abaixo, derivada da expressão 4.3.

$$ITT_{mr} = \frac{\left( \sum_{i=1}^N \left( \sum_{j=1}^{N_k} D_{mj} \cdot (|x_i - x_j| + |y_i - y_j|) \right) \right)}{N} \quad (4.4)$$

Onde:

$N$  é o número de CP's  $E_i$  do sistema produtivo;

$N_k$  é o número de instrumentos  $E_j$  alocados no CP em função do número máximo de sinais de E/S suportado pelo CP  $E_i$ .

$D_{mj}$  é o custo do cabo para interligar o instrumento  $E_i$  ao painel  $E_j$ .

$(|x_i - x_j| + |y_i - y_j|)$  é a distância entre os instrumentos  $E_i$  (origem) e os painéis  $E_j$  (destino).

3. Transfere-se o sistema de referência  $(O, \vec{e}_1, \vec{e}_2)$  para o sistema  $(O', \vec{e}'_1, \vec{e}'_2)$  de acordo com a expressão:

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \cdot \begin{bmatrix} x' \\ y' \end{bmatrix} + \begin{bmatrix} x_{cg} \\ y_{cg} \end{bmatrix} \quad (4.5)$$

Tal que

$\begin{bmatrix} x \\ y \end{bmatrix}$  são as coordenadas do ponto em  $(O, \vec{e}_1, \vec{e}_2)$

$\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}$  é a matriz mudança de base de um sistema de coordenadas para o outro.

$\begin{bmatrix} x' \\ y' \end{bmatrix}$  são as coordenadas do ponto em  $(O', \vec{e}'_1, \vec{e}'_2)$

$\begin{bmatrix} x_{cg} \\ y_{cg} \end{bmatrix}$  são as coordenadas do ponto do CVCG em  $(O, \vec{e}_1, \vec{e}_2)$  calculado em (1).

4. Troca-se o sistema de coordenadas cartesianas  $(O', \vec{e}'_1, \vec{e}'_2)$  para coordenadas polares de acordo com a expressão:

$$\left. \begin{array}{l} x' = \rho \cos \theta \\ y' = \rho \sin \theta \\ \tan \theta = y' / x' \end{array} \right\}$$

Variando-se  $\theta$ , inicia-se a partição do sistema em áreas de controle utilizando-se o procedimento da Figura x.1.

Para cada um dos locais definidos a partir do procedimento da figura 4.1 calcula-se o ponto estratégico para a colocação do painel de controle. Para isto, utiliza-se a expressão 4.6 (abaixo) que vem da expressão 4.1 em que  $N_k$  é o número de instrumentos da região k limitado pelo valor máximo de sinais de E/S suportado pelo CP.

$$E_{Gk} = O' + \sum_{i=1}^{N_k} \left( \frac{D_{mi}}{\sum_{j=1}^{N_k} D_{mj}} \right) + (P_i - O')$$

Onde:

$O'$  é a origem do sistema do plano  $(O', \vec{e}'_1, \vec{e}'_2)$

$E_{Gk}$  é a posição do painel de controle em  $(O', \vec{e}'_1, \vec{e}'_2)$  na área de controle k.

$D_{mj}$  é o custo do material gasto para transmitir o sinal gerado pelo instrumento  $E_j$ .

$P_i$  é a posição do instrumento  $E_i$  em  $(O', \vec{e}'_1, \vec{e}'_2)$  no sistema.

$N_k$  é o número de instrumentos que podem ser alocados no CP da área de controle k.

Utilizando-se este conjunto de procedimentos implementados no software proporcionará a este a obtenção de um mapeamento do sistema produtivo em áreas de controle. O ponto estratégico para a fixação do painel de controle (CP) em cada área será o CVCR.

O algoritmo é apresentado na Fig. 12.

<p>Inicializa <math>ITT_M = ITT_{MR}</math> ( <math>ITT</math> para a região)</p> <p>e <math>ITT_{MR} = 0</math> (correspondente ao <math>ITT_{MR}</math> calculado)</p> <p>e <math>i = 0</math> e <math>j = 0</math></p>																			
<p>Abre o arquivo de dados com as posições dos instrumentos <math>E_i</math> em coordenadas polares em ordem crescente de <math>\theta</math></p>																			
<p>Enquanto houver instrumentos na lista</p>																			
<table border="1"> <tr> <td>Inicializa uma nova região <math>\rightarrow j = j + 1</math></td> <td>e <math>ITT_{MRC} = 0</math></td> </tr> <tr> <td colspan="2">Enquanto <math>ITT_{MRC} &lt; ITT_M</math></td> </tr> <tr> <td colspan="2"> <math>i = i + 1 \rightarrow</math> Insere instrumento (<math>i</math>) na Área de Controle (<math>j</math>)</td> </tr> <tr> <td colspan="2">Atualiza o índice de ocupação do painel</td> </tr> <tr> <td colspan="2"><math>ITT_{MRC} = ITT_{MRC} + ITT_M(i)</math></td> </tr> <tr> <td colspan="2"> <math>ITT_{MRC} = ITT_M</math>  ?? </td> </tr> <tr> <td colspan="2"> <math>ITT_{MRC} - ITT_M</math> é maior que  <math>ITT_M - (ITT_{MRC} - ITT_M(i))</math>  ?? </td> </tr> <tr> <td> Atualiza <math>ITT_M</math> para próximo painel  <math>ITT_M = ITT_{MR}</math> </td><td> Atualiza <math>ITT_M</math> para próxima Área  <math>ITT_M = ITT_{MR} - (ITT_{MRC} - ITT_{MR})</math> </td> </tr> <tr> <td> Elimina o último Instrumento da Área (<math>j</math>)  <math>\downarrow</math>  <math>i = i - 1</math> </td><td> Atualiza <math>ITT_M</math> para próxima Área  <math>ITT_M = ITT_{MR} + (ITT_{MR} - (ITT_{MRC} - ITT_M))</math> </td> </tr> </table>		Inicializa uma nova região $\rightarrow j = j + 1$	e $ITT_{MRC} = 0$	Enquanto $ITT_{MRC} < ITT_M$		$i = i + 1 \rightarrow$ Insere instrumento ( $i$ ) na Área de Controle ( $j$ )		Atualiza o índice de ocupação do painel		$ITT_{MRC} = ITT_{MRC} + ITT_M(i)$		$ITT_{MRC} = ITT_M$ ??		$ITT_{MRC} - ITT_M$ é maior que $ITT_M - (ITT_{MRC} - ITT_M(i))$ ??		Atualiza $ITT_M$ para próximo painel $ITT_M = ITT_{MR}$	Atualiza $ITT_M$ para próxima Área $ITT_M = ITT_{MR} - (ITT_{MRC} - ITT_{MR})$	Elimina o último Instrumento da Área ( $j$ ) $\downarrow$ $i = i - 1$	Atualiza $ITT_M$ para próxima Área $ITT_M = ITT_{MR} + (ITT_{MR} - (ITT_{MRC} - ITT_M))$
Inicializa uma nova região $\rightarrow j = j + 1$	e $ITT_{MRC} = 0$																		
Enquanto $ITT_{MRC} < ITT_M$																			
$i = i + 1 \rightarrow$ Insere instrumento ( $i$ ) na Área de Controle ( $j$ )																			
Atualiza o índice de ocupação do painel																			
$ITT_{MRC} = ITT_{MRC} + ITT_M(i)$																			
$ITT_{MRC} = ITT_M$ ??																			
$ITT_{MRC} - ITT_M$ é maior que $ITT_M - (ITT_{MRC} - ITT_M(i))$ ??																			
Atualiza $ITT_M$ para próximo painel $ITT_M = ITT_{MR}$	Atualiza $ITT_M$ para próxima Área $ITT_M = ITT_{MR} - (ITT_{MRC} - ITT_{MR})$																		
Elimina o último Instrumento da Área ( $j$ ) $\downarrow$ $i = i - 1$	Atualiza $ITT_M$ para próxima Área $ITT_M = ITT_{MR} + (ITT_{MR} - (ITT_{MRC} - ITT_M))$																		

Figura 12. Representação em tabela do algoritmo de alocação física.

Tal que:

- $i \in \{1, \dots, N\}$ , sendo  $N$  o número de instrumentos;
- $j \in \{1, \dots, M\}$ , sendo  $M$  o número de Áreas de Controle ou painéis;
- $ITT_{MRC}$  é o  $ITT_{MR}$  calculado para delinear uma Área de Controle;
- $ITT_M(i)$  é o índice de ocupação do controlador  $i$ ;
- $ITT_M$  é o índice de ocupação esperado para delinear uma Área de Controle de acordo com o custo do cabeamento dos Instrumentos.

#### 4.5.2 Alocação dos blocos funcionais

Após a determinação das áreas de controle e consequentemente a determinação do número de CP's, devemos partir para a alocação ótima dos blocos funcionais nos CP's.

O problema de alocação blocos funcionais envolve inúmeras variáveis complexas que implicam em aumento da complexidade do procedimento tornando-o inviável. Para ser possível obter um algoritmo de distribuição que possa realizar a designação dos blocos funcionais em CP's de maneira ótima, torna-se necessário que os blocos funcionais sejam estruturados, atômicos e trabalhem de maneira independente (CAVALHEIRO, 2004).

Com os CP's e o número de CP's especificados e a premissa acima verificada, a solução do problema de alocação de blocos funcionais em sistemas de controle tem como base as seguintes etapas (CAVALHEIRO, 2004):

- Definição das características dos CP's que realizam o controle do sistema produtivo e suas peculiaridades, que é feita no início do processo quando o tipo de CP é escolhido, bastando acessar a base de dados e obter as características desejadas.
- Determinação da quantidade de blocos funcionais que compõem a lógica de controle do sistema produtivo, informação que vem na entrada do software junto com a descrição do processo em MFG já estruturado nestes blocos funcionais;
- Determinação das características de cada bloco funcional do sistema. Isso será implementado associando as informações dos CP's na base de dados à descrição do processo no formato de entrada e calculando seus custos.
- Especificação de algoritmos para a alocação dos blocos funcionais de acordo com as características dos CP's, algoritmo de designação.

A seguir apresentaremos a modelagem dos componentes envolvidos na alocação dos blocos funcionais propostos por (CAVALHEIRO, 2004):

#### 4.5.2.1 Modelo do bloco funcional

Para execução do algoritmo o Bloco Funcional será representado por (VALDEHI; KRISHNAN, 1997):

- Peso  $\{s_i\}$ , onde  $s_i$  representa o número de instruções do bloco funcional  $i$ ;
- Constante de tempo  $\{d_i\}$ , onde  $d_i$  representa o "scatime" mínimo em que o bloco funcional  $i$  poder ser processado;
- Recursos consumidos  $\{res_i\}$ , onde  $res_i$  representa a quantidade de recursos consumidos pelo bloco funcional  $i$ ;
- Pesos  $\{v_{ij}\}$ , onde  $v_{ij}$  representa a carga computacional associada ao volume de informações trocadas entre os blocos funcionais  $i$  e  $j$ .

Para utilização do algoritmo o importante é o cálculo da carga computacional consumida, ou seja, o tempo de processamento gasto para executar o algoritmo elaborado para ser processado no equipamento.

#### 4.5.2.2 Modelo dos CP's

Uma arquitetura de controle em sistemas de controle com processamento paralelo e distribuído, segundo (VALDEHI; KRISHNAN, 1997); (ABB, 2001a); (ABB, 2001b); (ROCKWELL, 2003), pode ser representada por:

- Pesos  $\{\mu_q\}$ , onde  $\mu_q$  representa a taxa de serviço do processador  $q$ ;
- Período do ciclo de processamento  $\{win_q\}$ , onde  $win_q$  representa o período utilizado pelo processador  $q$  para executar os blocos funcionais a ele designados;
- Recursos fornecidos  $\{res_q\}$ , onde  $res_q$  representa os recursos disponíveis no CP  $q$ ;
- Links de comunicação entre os CP's são associados peso  $\{c_{pq}\}$  que representam a capacidade de comunicação dos links entre os CP's  $p$  e  $q$ .

Com os modelos definidos podemos estabelecer o relacionamento do bloco funcional x CP conforme os parâmetros apresentados na seqüência (CAVALHEIRO, 2004). Para os cálculos destes parâmetros, é necessária a associação do bloco funcional ao CP.

#### 4.5.2.3 Custo computacional dos blocos funcionais

Trata-se do período de tempo em que o CP está alocado para executar o bloco funcional em relação a velocidade de processamento do CP, demonstrado na equação 4.7:

$$C_i^q = \left[ \frac{\sum_{i=1}^n S_i + \sum_{j=1}^n v_{ij}}{\mu_q} \right] \quad (4.7)$$

Tal que:

$C_i^q$  é o tempo de execução (custo computacional) do bloco funcional  $i$  no CP  $q$

$S_i$  é o número de instruções necessárias para calcular o bloco funcional  $i$  no CP  $q$

$\mu_q$  é a velocidade de processamento do CP  $q$

$v_{ij}$  é o número de instruções associada ao volume de informações solicitada pelo bloco funcional  $i$  enviada (ou recebidas) através dos blocos de interface;

Carga de processamento dos CP's:

Os CP's trabalham em ciclos de leitura (scantime) que detectam as mudanças nos estados a cada leitura. Como os Blocos Funcionais apresentam uma constante de tempo ( $d_i$ ), estes ciclos de leitura devem ser feitos a uma velocidade ( $\mu_q$ ) e scantime ( $win_q$ ) compatíveis com esta constante. Assim tempos a carga de processamento na equação 4.8:

$$Lc_q = \left[ \frac{\sum_{i=1}^n S_i + \sum_{j=1}^n v_{ij}}{\mu_q} \right] \quad (4.8)$$

Tal que:

$win_q$  do CP deverá ser menor ou no máximo igual ao menor  $d_i$  associados ao bloco funcional que será executado para atender os requisitos temporais do bloco funcional.

Quanto  $Lc_q > 1$  significa que a carga computacional do processador  $q$  excedeu os 100% da capacidade máxima do CP. Portanto, este CP não cumprirá os requisitos temporais exigidos pelo bloco funcional, podendo assim causar distúrbios no controle do sistema produtivo.

#### 4.5.2.4 Consumo de recursos dos CP's

É a disponibilidade de recursos do CP  $q$  à medida que blocos funcionais  $i$  são alocados neste CP e vão diminuindo a quantidade de recursos disponíveis.

$$Lres_q = \frac{\sum_{i=1}^n res_i}{res_q} \quad (4.9)$$

Se  $Lres_q > 1$  significa que os recursos computacionais  $q$  do CP excederam os 100% da capacidade máxima de alocação de blocos funcionais do CP. Portanto, não pode ser alocado mais nenhum bloco funcional neste CP.

Para definir um procedimento (algoritmo) para alocação dos CP's, considera-se um problema de alocação  $N_t$  blocos funcionais em  $N_p$  CP's disponíveis com o objetivo de determinar o que deve ser processado e quem deve processar, onde os blocos funcionais que exigem maior rapidez para a entrega do resultado, exige uma demanda maior de processamento em relação a um bloco funcional com o mesmo número de elementos e a definição de quem deve processar cada bloco funcional, implica em avaliar-se o custo envolvido entre os CP's (CAVALHEIRO, 2004). Logo teremos duas premissas básicas como diretrizes para realização da alocação dos blocos funcionais (CAVALHEIRO, 2004):

1. Considerando-se um conjunto de  $M$  CP's para o processamento dos blocos funcionais que efetuam o controle e a monitoração do processo em um sistema produtivo dividido em  $M$  áreas de controle, a uniformidade quanto as cargas de processamento de cada área deve ser uma condição necessária para realização da alocação destes blocos funcionais.
2. Considerando-se a carga de processamento de um certo CP em uma determinada área de controle, a ociosidade do CP deve ser uma condição básica para realização da alocação de blocos funcionais neste CP.

Tendo todas os parâmetros, condições e premissas para a alocação dos blocos funcionais sido estabelecidos, podemos partir para a apresentação dos procedimentos a serem implementados no software para realização desta alocação:

Trata-se de um problema onde temos um número de determinado de CP's e outro de blocos funcionais e o custo  $C$  associado ao processamento do bloco funcional em determinado CP.

Então o problema consiste em estabelecer a matriz de custos ( $C_{ij}$ ) e a partir desta aplicar o algoritmo de designação.

A matriz de custos trata-se de uma matriz com os custos  $C_{ij}$  do processamento do bloco funcional  $i$  no CP  $j$ . Como na maioria dos sistemas de controle o número de blocos funcionais é maior que o de CP's para tornar a matriz quadrada, são criados CP's fictícios com custos da ordem de e10 o que não compromete a solução.

Os custos  $C_{ij}$  podem ser calculados de duas maneiras (CAVALHEIRO, 2004), através da expressão 4.10 derivada da expressão 4.7

$$C_i^q = \frac{s_i + v_{ij}}{\mu_q} \quad (4.10)$$

Tal que:

$s_i$  é a carga computacional exigida pelo bloco funcional  $i$ , e pode ser quantificada pelo número de instruções que o bloco funcional possui;

$v_{ij}$  é a carga computacional exigida pelo bloco funcional  $i$  para utilizar blocos de interface para trocar informações com o bloco funcional  $j$  para leitura e escrita dos dados de E/S, e pode ser quantificada pelo número de instruções que o(s) bloco(s) funcional (ais) de interface possui (em);

$\mu_q$  é a velocidade com que o CP  $q$  calcula o bloco funcional  $i$ , e pode ser quantificada pelo número de instruções que o CP processa em um determinado período;

$C_i^q$  é o custo computacional associado ao bloco funcional  $i$  em relação ao CP  $q$  e pode ser quantificado pelo tempo de processamento gasto pelo CP para processar um bloco funcional.

Ou através da expressão 4.11:

$$C_i^q = \sum_{j=1}^n N_j \times T_j \quad (4x.11)$$

Tal que:

$N_j$  é o número de funções do tipo  $j$  do bloco funcional  $i$ ;

$T_j$  é o tempo de execução da função  $j$  no CP  $q$ ;

$C_i^q$  é o custo computacional associado ao bloco funcional  $i$  em relação ao CP  $q$  e pode ser quantificado pelo tempo de processamento gasto pelo CP para processar um bloco funcional.

A princípio utilizaremos a expressão 4.11, pois se trata de um cálculo mais simples porém não menos realista do custo de processamento.

Definida a matriz de custos, segue a sistematização do conjunto de procedimentos proposto em (CAVALHEIRO, 2004) para definir a alocação dos blocos funcionais nos CP's.

1. Alocar os blocos funcionais com CP's pré-determinados. Esta etapa a princípio não será implementada, pois trata-se de casos particulares onde o projetista deseja que um determinado bloco funcional seja alocado em um determinado CP.
2. Verificar quais blocos funcionais que não possuem CP's designados. Que, em decorrência da não implementação da etapa 1, serão todos os blocos funcionais.
3. Verificar quais CP's são capazes de processar os blocos funcionais utilizando a equação 4.11.
4. Verificar quais CP's são capazes de armazenar os blocos funcionais utilizando a equação 4.10.
5. Montar a matriz de custo dos  $i$  blocos funcionais versus os  $j$  CP's capazes de armazená-los e processá-los. Se em um CP não for possível armazenar ou processar um bloco funcional a este elemento da matriz será atribuído um custo da ordem de  $e^{10}$ .
6. Calcular os custos para cada par  $C_{ij}$  (bloco funcional x CP) da matriz de custo  $C$ , utilizando a expressão 4.11.
7. Utilizar o método de designação (manipulação da matriz C) para alocar os blocos funcionais nos CP's.

8. Mostra o quanto houve de sobra (ou falta) de carga de processamento, consumo de recursos dos CP's do sistema e a lista de alocação dos blocos funcionais x CP's.

Ao término da execução destes procedimentos obteremos a alocação ótima dos blocos funcionais nos CP's quanto ao custo de processamento destes.

## **5. IMPLEMENTAÇÃO**

### **5.1 Considerações sobre a implementação**

Para possibilitar uma implementação computacional dos conceitos apresentados no decorrer do texto baseados em (CAVALHEIRO), tornam-se necessárias algumas considerações e modificações para que a metodologia seja implementada de maneira coerente com as hipóteses adotadas.

A alocação física deve ser realizada levando em conta os componentes do sistema de controle, associados a um bloco funcional. De acordo com o algoritmo para alocação física, proposto em (CAVALHEIRO), o sistema é dividido em áreas de controle observando os componentes do sistema de controle isoladamente. Isso desrespeita a hipótese de atomicidade, permitindo a alocação de entradas e saídas de um mesmo bloco funcional em CP's diferentes. Isso viola a hipótese de atomicidade, o que não é desejado para um sistema de controle. Portanto no momento de alocação física deve-se considerar além de seus componentes, que no caso serão os sinais entradas e saídas dos CP's, a qual bloco funcional recebe ou envia sinais ao componente.

Também para garantir a atomicidade, considerando que normalmente em um sistema de controle é utilizado um único modelo de CP - por questões de compatibilidade e padronização da linguagem dos processos - só é permitido utilizar na alocação modelos de CP's que contenham número de entradas e saídas que garantam a alocação do maior bloco funcional do processo em termos de entradas e saídas.

Pelo fato da alocação física ser feita associada aos blocos funcionais, ao fim da alocação física, a alocação dos blocos funcionais já é determinada não sendo necessária a implementação do algoritmo de designação como proposto em (CAVALHEIRO). A única ação com relação à alocação funcional necessária é garantir, no momento da alocação física, que o tempo de processamento não ultrapasse a menor constante de tempo do bloco funcional alocado em determinado CP, e a memória ocupada pelo código dos blocos funcionais não exceda a memória disponível no CP.

### **5.2 A implementação em software**

#### **5.2.1 Recursos utilizados:**

Para a implementação do algoritmo desenvolvido em software identificamos a necessidade de alguns recursos. Entre eles, uma interface gráfica de fácil programação, estrutura para implementação de algoritmos e uma estrutura para implementação de uma base de dados.

Para tanto identificamos que o Visual Basic associado ao Microsoft Excel seria uma ferramenta apropriada, pois apresenta grande praticidade e agilidade na implementação dos

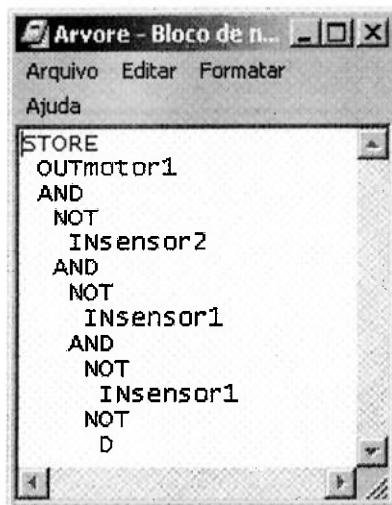
recursos desejados. Trata-se de uma consideração bastante importante, pois proporcionou certa facilidade ao permitir uma maior concentração nos conceitos acerca do desenvolvimento do algoritmo.

### 5.2.2 Implementação das Entradas e Saídas

Determinadas entradas do algoritmo necessitaram de um formato definido para serem implementadas. A seguir uma descrição destes formatos.

#### 5.2.2.1 Processos Estruturados em Árvores

A descrição dos processos estruturados, como foi apresentado na metodologia, foi implementada em arquivos de texto, com os níveis das árvores apresentados como indentações. Cada bloco funcional é apresentado em um arquivo que apresenta o formato como mostra a Fig. 13.



```
STORE
OUTmotor1
AND
NOT
INsensor2
AND
NOT
INsensor1
AND
NOT
INsensor1
NOT
D
```

Figura 13. Arquivo de texto com bloco funcional estruturado.

#### 5.2.2.2 Base de Dados de CP's

A base de dados de CP's é considerada entre as entradas dos algoritmos pois deve ser configurada pelo usuário. Esta foi implementada através de uma planilha com a marca, modelo tempos de processamento de cada operador (and, or, set, reset, etc) no CP, espaço em memória ocupado por cada operador e seu custo.

### **5.2.3 Algoritmo de Alocação Funcional**

Na implementação do algoritmo de alocação funcional tornou-se necessário uma integração dos conceitos dos dois algoritmos apresentados em (CAVALHEIRO): o algoritmo de alocação física e o algoritmo de alocação funcional.

Isso se tornou necessário devido a hipótese de atomicidade, como foi apresentado nas considerações para a implementação.

A alocação funcional é realizada levando-se em conta o número de entradas e saídas associadas a cada bloco funcional, garantido assim a atomicidade do sistema e consequentemente sua integridade.

A alocação funcional é implementada da seguinte maneira:

Para cada modelo de CP possível de ser utilizado (que garanta a alocação do maior bloco funcional), é realizada a alocação dos blocos funcionais seguindo três critérios:

- 1 - Número de entradas e saídas
- 2 - Tempo de processamento
- 3 - Memória de armazenamento

Os blocos funcionais vão sendo alocados no CP até que algum dos critérios seja violado, então, é selecionada mais uma unidade do CP para alocar os blocos funcionais restantes.

Esse processo é repetido até que todos os blocos funcionais sejam alocados em CP's.

Portanto os blocos funcionais são alocados visando a máxima utilização possível dos recursos do CP.

### **5.2.4 Algoritmo de Alocação Física**

A segunda parte do algoritmo denominada de "alocação física" trata do algoritmo de posicionamento dos CP's após a alocação dos blocos funcionais.

Para se determinar o posicionamento dos CP's - primeiramente de acordo com o tamanho da planta - cria-se um intervalo discretizado correspondente à área da planta.

Em seguida identifica-se que componentes trocam sinais com o CP (entradas e saídas) e suas posições e, por fim, é calculada a posição de menor distância de todos os componentes até o CP.

A resposta deste algoritmo é uma sugestão de alocação considerando o intervalo discretizado da planta.

### **5.2.5 Demonstração dos resultados e posicionamento**

Os resultados dos algoritmos são apresentados no decorrer da execução do software e apresenta ao usuário todas as soluções possíveis para os modelos CP's – que atendam à hipótese de atomicidade - cadastrados na base de dados, apresentando:

- Entradas e saídas utilizadas de cada CP
- Tempos de processamento dos BF's associado a cada CP
- Memória utilizada de cada CP
- Sugestão de posicionamento dos CP's na planta
- Valores porcentuais de utilização dos recursos do CP
- Custo (\$) de cada CP

A partir das soluções possíveis são apresentadas através de escolhas dentre os diversos modelos de CP's e sua alocação dos blocos funcionais, cabendo ao usuário a decisão sobre a escolha que melhor lhe convém.

Para cada escolha definida, através de seleção pelo usuário, o software gera o layout gráfico da planta com a sugestão de posicionamento dos CP's e com o posicionamento dos componentes do sistema de controle.

## 6. TUTORIAL PARA UTILIZAÇÃO DO SOFTWARE

### 6.1 A simplicidade de interação

O software foi desenvolvido para que o usuário tenha o menor trabalho possível ao implementar seu teste e obter as respostas desejadas. Dessa maneira, os passos que requerem a interação do usuário são poucos.

Deve-se lembrar que há uma dualidade entre essa mínima interação do usuário com a redução de opções dadas ao usuário – que seria interessante para uma combinação maior de possibilidades de testes. Porém, como uma maior gama de interações resultaria em uma necessidade de robustez cada vez maior do software, assim como maior necessidade de implementar interfaces gráficas, o que não incrementaria muita coisa no entendimento do trabalho de alocação funcional. Assim, optamos definitivamente por essa simplificação.

### 6.2 Passo a passo

#### 6.2.1 'Splash Screen'

Ao iniciar o programa, visualiza-se inicialmente um 'Splash Screen' identificando o Software e seus autores, como na Fig. 14.



Figura 14. 'Splash Screen' identificando o software e autores.

#### 6.2.2 Selecionando arquivos com blocos estruturais estruturados

Na primeira tela do software – Inserção de Blocos Funcionais, Fig. 15 – é onde se carregam os arquivos de extensão .txt com os blocos funcionais estruturados em árvore como se verifica na Fig. 16.

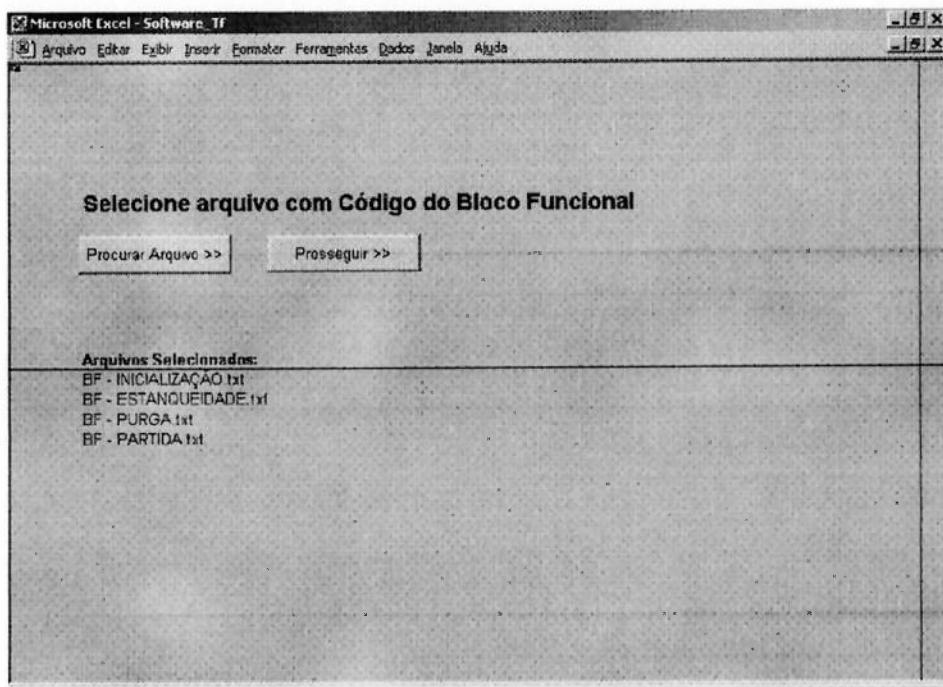


Figura 15. Inserção de Blocos Funcionais, tela inicial de seleção de arquivos com blocos funcionais.

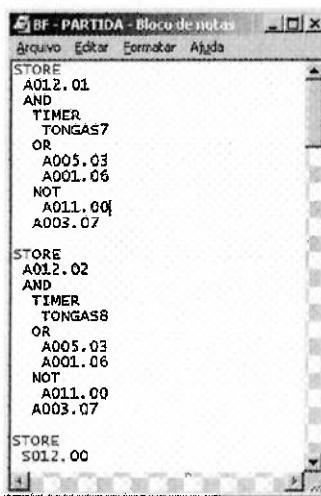


Figura 16. Arquivo txt com bloco funcional estruturado.

Através do botão **Procurar Arquivo >>**, pode-se escolher adicionar quantos blocos funcionais se desejar. O botão leva à janela de seleção de arquivos já com seleção designada para tipos de texto, Fig. 17.

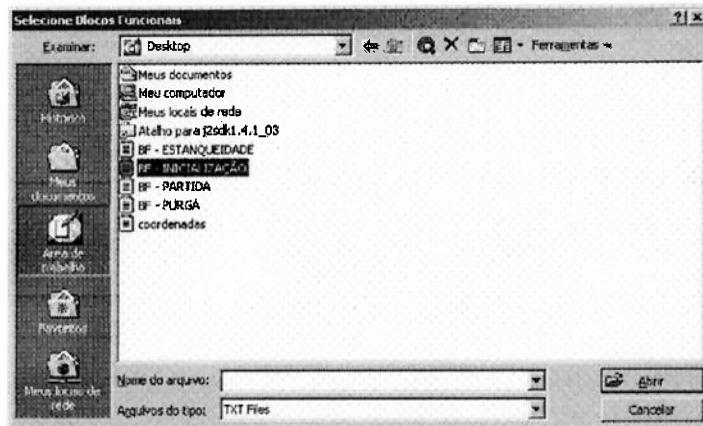


Figura 17. Janela para seleção de arquivos com blocos funcionais.

Neste momento quando o arquivo é selecionado e o usuário clica em abrir, o software importa o arquivo de texto organizando suas operações e variáveis em células e realiza a varredura do código realizando a contagem dos diversos operadores, flags, entradas e saídas.

Essas informações são armazenadas em duas bases de dados.

A primeira base de dados é o sumário de blocos funcionais, onde é inserido uma linha para cada bloco funcional (arquivo) armazenando nas colunas a quantidade de cada operador do código, quantidade de flags, de entradas e saídas. Esta base será sempre consultada quando se desejar saber a quantidade de operações no BF.

A segunda é uma base de dados onde são armazenados o nome de cada variável de entrada e saída, associadas ao bloco funcional, que será utilizada para gerar a interface gráfica onde o usuário irá cadastrar as coordenadas dos componentes do sistema de controle.

Para cada arquivo selecionado, o software pergunta qual a constante de tempo de associada àquele bloco funcional, vide Fig. 18.

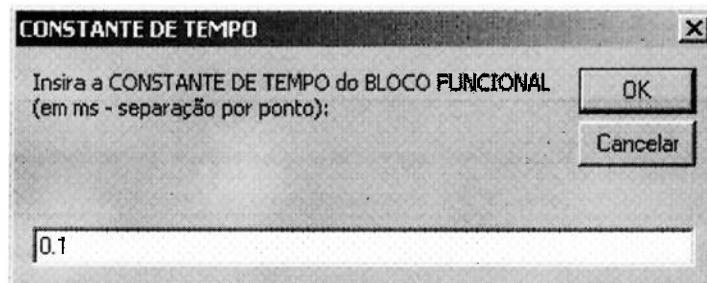


Figura 18. Entrada da constante de tempo associado ao bloco funcional selecionado.

Este valor é guardado na base de dados que contém o sumário dos blocos funcionais mencionada acima.

Para cada arquivo selecionado é registrado logo abaixo seu nome para identificação, Fig. 19.

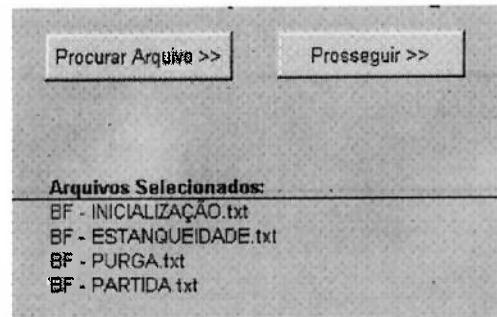


Figura 19. Detalhe dos nomes dos arquivos que foram selecionados.

Quando aperta-se o botão **Prosseguir >>**, o software leva à próxima tela – Inserção de Dimensões e Posições, Fig. 21 – onde o usuário deve entrar com dados de dimensão da planta e posicionamento dos elementos de entrada e saída associados aos blocos funcionais.

O próprio software avisa através de uma janela o que deve ser feito nesta etapa, Fig. 20.

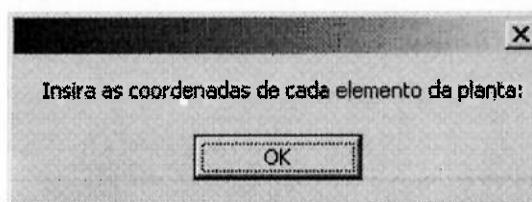


Figura 20. Aviso sobre a tarefa a ser executada pelo usuário.

Observações importantes:

- Deve-se ressaltar que para o correto funcionamento do software, é extremamente importante que o software registre apenas valores válidos e possíveis para as dimensões da planta e as coordenadas dos elementos.
- A planta não deve ter dimensão nula e nem aceitar um caractere que não seja numérico. Já as coordenadas dos elementos não devem permitir caracteres não-numéricos, dimensão nula ou coordenadas que ultrapassem as dimensões da planta.
- Assim, ao invés de contar com o usuário para que os dados estejam corretos, o software apresenta robustez tal que em todos estes casos é dado um aviso para que a correção seja feita, Fig. 22.

**DIMENSÕES DA PLANTA E POSICIONAMENTO DOS ELEMENTOS DE ENTRADA E SAÍDA**

**Dimensões da Planta:**

X:	75	(metros)
Y:	40	

**Posição das Entradas na Planta:**

	x	y	(metros)
BE01	45	24,9	
FSH02	69,77	35,16	
LSL01	67,77	35,16	
LSL02	67	35,16	
M001-L	0	26,52	
PSH01	9,34	27,47	
PSH01	73,07	25,82	
PSL01	10,52	27,47	
PSL02	63,73	27,47	
PSL03	15,93	27,47	
PSL04	57,24	27,47	
SM01-A	53	7	
SM01-F	53,5	7	
SM02	50	30	
SV01-A	3,86	26,57	
SV01-F	3,86	26,57	
SV02-A	12,64	26,57	
SV02-F	12,64	26,57	

**Posição das Saídas na Planta:**

	x	y	(metros)
M001	41	24,9	
SM02-A	69,67	35,16	
SM02-F	67	35,16	
SV01	67	35,16	
SV02	3,66	26,57	
SV05	12,64	26,57	
TR01	3,7	35,16	

Figura 21. Inserção de Dimensões da Planta e Posições dos Elementos IN/OUT.

Para a apresentação da tela acima o software consulta a base de entradas e saídas geradas a partir da leitura do código do processo e as apresenta para o usuário.

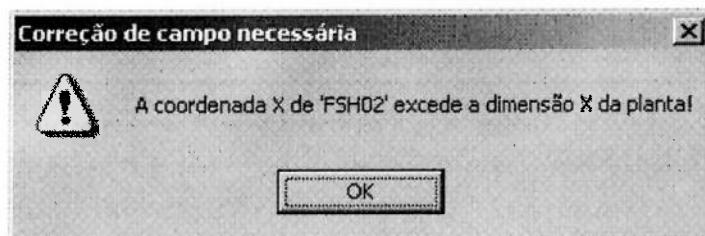


Figura 22. Robustez - aviso para correção de dado incorreto.

Mais uma vez ao escolher **Prosseguir >>**, o usuário já pode visualizar todas as escolhas possíveis entre os controladores cadastrados na base de dados de controladores, tela Possibilidades de Solução, Fig. 23.

Quando o botão é acionado uma rotina no software é disparada. Esta rotina realiza a seleção dos CP's que podem ser utilizados, escolhendo os CP's que tenham número de entradas e saídas suficientes para alocar o maior bloco funcional cadastrado.

Em seguida ordena os CP's em ordem do maior para o menor tamanho e para cada modelo escolhido realiza uma verificação do número de entradas e saídas para ver se ainda há espaço suficiente para alocar o bloco funcional. Continuando, calcula o tempo de processamento do bloco funcional em questão, soma ao tempo do bloco funcional previamente alocado pelo algoritmo (a partir da segunda iteração) e verifica se resultado é menor que um décimo da menor das constantes de tempo dos blocos funcionais alocados e do bloco funcional em questão.

Por último a rotina calcula a quantidade de memória ocupada pelo código do BF e verifica se o espaço em memória livre é suficiente para armazená-lo.

Se todas as condições forem atendidas a rotina aloca o BF no CP e parte para tentar alocar o próximo BF no CP. Ela faz isso para todos os BF's do maior para o menor para ver se há mais algum BF que possa ser alocado no CP, se não for possível alocar mais nenhum dos BF's no CP a rotina seleciona mais uma unidade do CP e repete todo o processo para os BF's restantes.

O processo é repetido até que todos os BF's sejam alocados naquele modelo.

Após isso, a rotina parte para a próxima escolha que representa a alocação para o próximo modelo da base de dados, e assim continua até que todos os modelos possíveis sejam analisados.

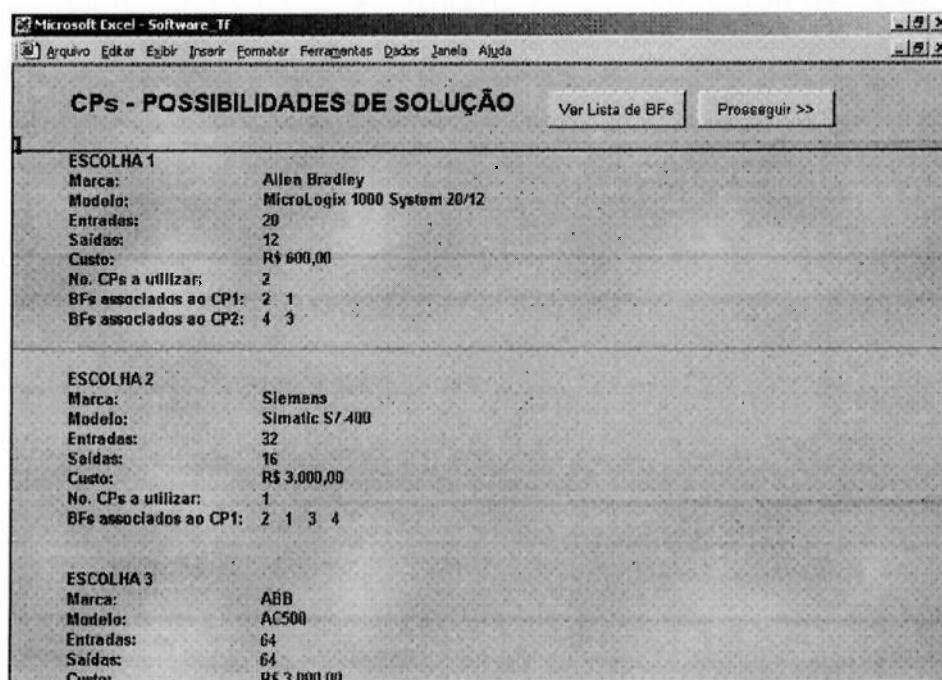


Figura 23. Possibilidade de Solução de controladores e suas características.

Em cada possibilidade de solução, são apresentados: a marca do controlador, o modelo, número de entradas e de saídas, custo, sugestão de número de controladores a ser utilizado para essa combinação, e os blocos funcionais associados a cada controlador.

Como a associação do bloco funcional ao controlador é feita através da chave primária que o identifica na base de dados de blocos funcionais, para saber qual é de fato o arquivo associado ao bloco, há o botão **Ver Lista de BFs** que possibilita verificar pela relação da chave primária qual o nome do arquivo associado e sua constante de tempo, Fig. 24.

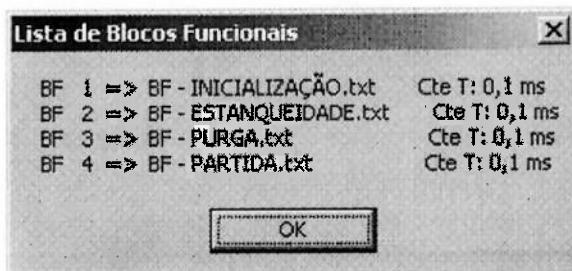


Figura 24. Janela auxiliar que identifica os blocos funcionais e chaves primárias associadas, além da constante de tempo de cada bloco.

De modo a fundamentar a escolha do controlador, em **Proseguir >>** é possível visualizar o resumo de ocupação do controlador pelos blocos funcionais, na tela Resumo de Utilização do Controlador por Escolha – Valores Absolutos, Fig. 25.

Neste processo o software dá o "display" dos valores calculados na rotina logo acima.

	MEMÓRIA	ENTRADA	SAÍDA	TEMPO DE PROCESSAMENTO	CUSTO
ESCOLHA 1 CP1	179	15	3	0,000422 ms	R\$ 600,00
	CP2	320	17	0,000800 ms	R\$ 600,00
ESCOLHA 2 CP1	507	32	10	0,001136 ms	R\$ 3.000,00
ESCOLHA 3 CP1	507	32	10	0,001184 ms	R\$ 3.000,00

Para visualizar o posicionamento de menor custo dos CPS na planta, determine a ESCOLHA a ser utilizada:

Selecionar ESCOLHA

Visualizar Planta

Printar los CPS

Figura 25. Resumo de Utilização do Controlador por Escolha – Valores Absolutos.

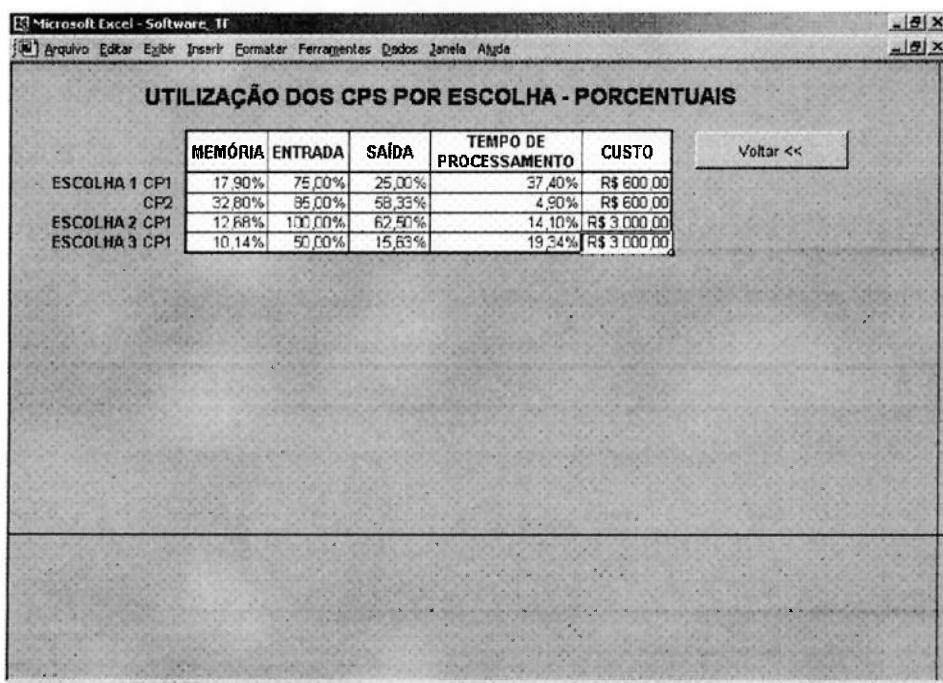
Nesta tela verifica-se quanto foi ocupado por critério em cada controlador das escolhas. Por exemplo, o campo 'Memória' apresenta o total de memória ocupada por todos os blocos funcionais associados àquele controlador. No campo 'Entradas' apresenta-se o número de entradas ocupadas naquele controlador, e o mesmo para 'Saídas'.

Por fim, em 'Tempo de Processamento', apresenta-se o tempo total de processamento que os blocos funcionais associados àquele controlador levam para serem executados. Vemos o detalhe do resumo na Fig. 26.

	MEMÓRIA	ENTRADA	SAÍDA	TEMPO DE PROCESSAMENTO	CUSTO
ESCOLHA 1 CP1	179	15	3	0,000422 ms	R\$ 600,00
	CP2	328	17	0,000800 ms	R\$ 600,00
ESCOLHA 2 CP1	507	32	10	0,001136 ms	R\$ 3.000,00
ESCOLHA 3 CP1	507	32	10	0,001184 ms	R\$ 3.000,00

Figura 26. Detalhe do resumo de ocupação dos controladores.

Entretanto, como os valores absolutos podem não levar a uma conclusão muito satisfatória de ocupação dos critérios apresentados, é possível visualizar os mesmos valores, porém com referência a seus valores limites, dando um porcentual de ocupação. Para isso, basta o usuário pressionar o botão **% de Utilização**. Neste momento passa-se à tela Resumo de Utilização do Controlador por Escolha – Valores Porcentuais, Fig. 27.



	MEMÓRIA	ENTRADA	SAÍDA	TEMPO DE PROCESSAMENTO	CUSTO
ESCOLHA 1 CP1	17,90%	75,00%	25,00%	37,40%	R\$ 600,00
	CP2	32,80%	55,00%	58,33%	4,90% R\$ 600,00
ESCOLHA 2 CP1	12,68%	100,00%	62,50%	14,10%	R\$ 3.000,00
ESCOLHA 3 CP1	10,14%	50,00%	15,63%	19,54%	R\$ 3.000,00

Figura 27. Resumo de Utilização do Controlador por Escolha – Valores Porcentuais.

Estas percentagens são calculadas a partir dos valores da tela anterior, tomando como absolutos para memória, entradas e saídas, os dados do CP's. E para o tempo de processamento o valor absoluto é um décimo da menor constante de tempo do bloco funcional alocado naquele CP.

Através desta tela, é possível entender melhor a ocupação que foi feita entre os critérios dos controladores em diferentes escolhas. No detalhe do resumo, Fig. 28, vemos a ocupação por cada critério.

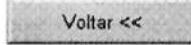
Com esses dados, obtemos informações de ocupação que nos possibilitam pensar em alterações do projeto e da alocação dos blocos funcionais, ou seja, voltando à nossa motivação inicial, caso nossos processos precisem ser modificados levando a consequentes modificações dos códigos de controle e seus blocos funcionais estruturados, podemos saber se aquele controlador possibilita alocar mais recursos ou não dada sua alocação atual.

Exemplo: dada a ocupação do resumo em Fig. 28, queremos modificar o bloco funcional 3 (Purga) devido a alterações em seu processo.

Na Escolha 1 (CP de 20 entradas), o bloco funcional 3 ocupa o CP2. Caso as alterações incluam adição de novas entradas, somente 3 novas entradas poderão ser acrescentadas caso se deseje continuar com a possibilidade de manter o CP da Escolha 1 dentre as escolhas possíveis, uma vez que já estão sendo utilizadas 17 entradas.

	MEMÓRIA	ENTRADA	SAÍDA	TEMPO DE PROCESSAMENTO	CUSTO
ESCOLHA 1 CP1	17,90%	75,00%	25,00%	37,40%	R\$ 600,00
	32,80%	85,00%	58,33%	4,90%	R\$ 600,00
ESCOLHA 2 CP1	12,68%	100,00%	62,50%	14,10%	R\$ 3.000,00
ESCOLHA 3 CP1	10,14%	50,00%	15,63%	19,34%	R\$ 3.000,00

Figura 28: Detalhe da ocupação porcentual dos controladores.

Ao pressionar  Voltar <<, volta-se à tela Resumo de Utilização do Controlador por Escolha – Valores Absolutos. Do mesmo nesta tela, ao pressionar  Voltar <<, volta-se à tela Possibilidade de Solução.

Ainda na tela Resumo de Utilização do Controlador por Escolha – Valores Absolutos, pode-se para cada solução possível escolher no DropBox a Escolha de modo a visualizar a planta e o posicionamento dos elementos de entrada e de saída. Também visualiza-se o posicionamento do(s) CP(s) de acordo com o critério de mínimo custo.

Selecionar ESCOLHA
ESCOLHA 1
ESCOLHA 2
ESCOLHA 3

Após selecionar a Escolha, pode-se visualizar a planta em

**Visualizar Planta**, ou ainda verificar as coordenadas de posição do(s) CP(s) em **Posição dos CPs**.

Estes botões somente são ativados depois que é selecionada uma escolha.

Escolhendo **Visualizar Planta**, observa-se a tela Planta, Fig. 29.

No momento em que se seleciona “visualizar planta” o software dispara uma rotina que verifica a escolha selecionada, identifica a quantidade de CP's utilizada bem como os elementos associados a cada CP. Em seguida a partir de uma discretização das dimensões da planta, gera uma área equivalente à área da planta e posiciona nela os componentes do sistema de controle.

A seguir, realiza-se para cada CP e seus componentes associados um cálculo exaustivo para cada posição (discretizada) da planta a distância a cada componente associado (pela hipotenusa das coordenadas) e soma, obtendo a distância total.

Por fim, determina-se o posicionamento do CP para o local cuja distância calculada é a menor e o posiciona na planta.

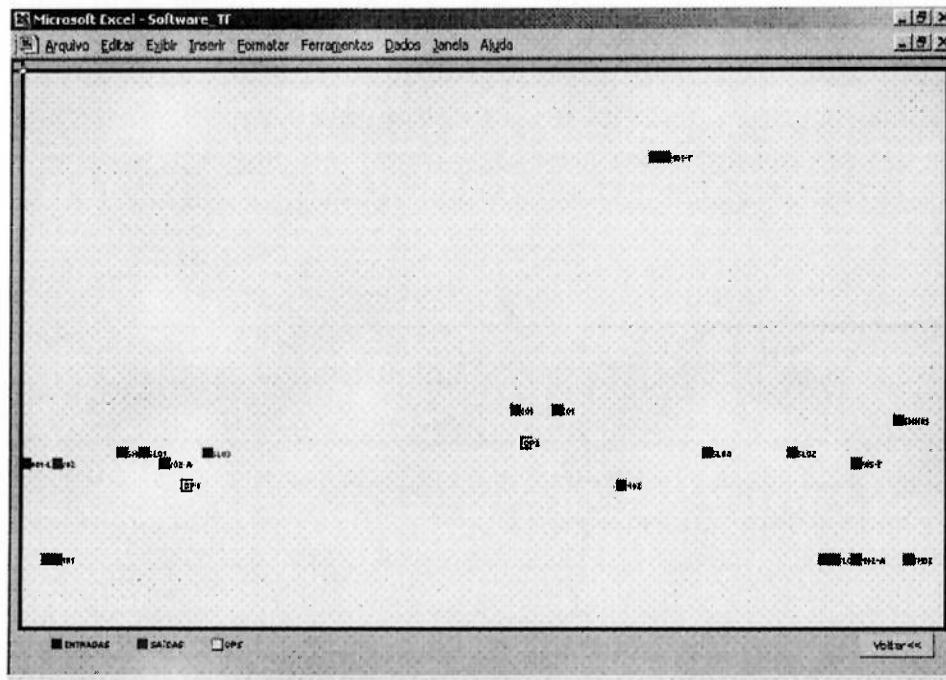


Figura 29. Planta com elementos de entrada em azul, elementos de saída em vermelho, e controladores em amarelo.

Escolhendo **Posição dos CPs**, observa-se a tela Posição do Controladores na Planta para Escolha Selecionada, Fig. 30.

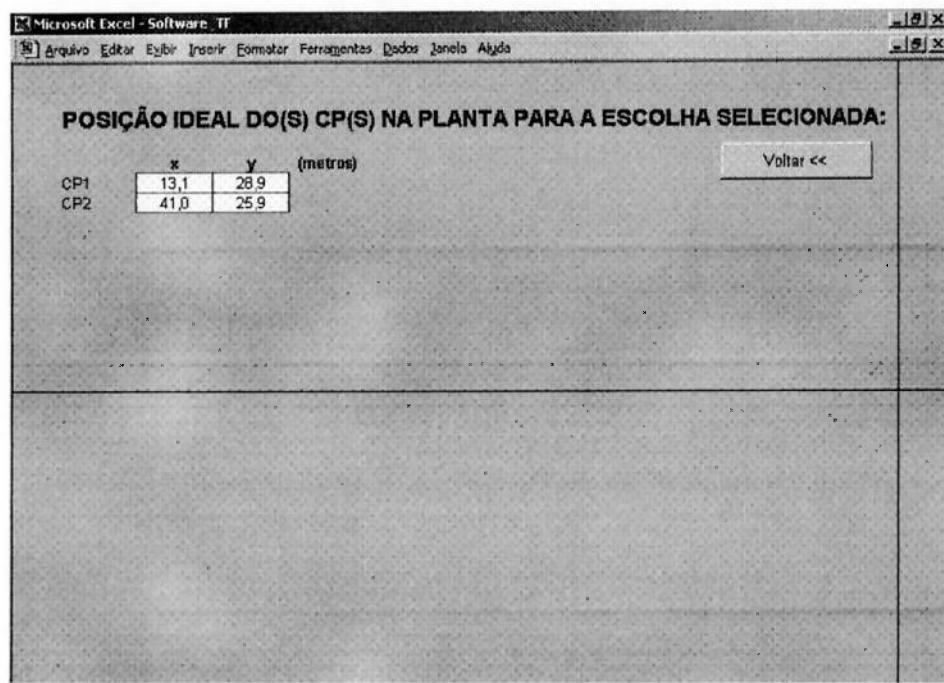


Figura 30. Posição dos Controladores na Planta para Escolha Selecionada.

## 7. CONSIDERAÇÕES FINAIS

Como foi mencionado em várias partes do texto, o problema de alocação de blocos funcionais em CP's envolve um número grande de variáveis complexas o que torna o problema muito difícil de se resolver considerando todos os aspectos necessários. Por exemplo, a evolução tecnológica e das normas IEC já tornam mais complexo o processo de tomada de decisão.

Para se poder implementar uma solução automatizada para o problema, que é o foco deste trabalho, tornou-se necessário a definição de um aspecto a ser abordado, no caso o custo computacional, e a consideração de várias premissas e limitações o que torna a solução um tanto quanto limitada.

Porém, apesar de limitada, a solução é bastante válida considerando que o aspecto abordado é o mais importante na alocação dos blocos funcionais:

- verificação do tempo de processamento de acordo com a constante de tempo do bloco funcional;
- e o consumo de recursos do bloco funcional no CP.

Observa-se também que os conceitos apresentados e o software que os implementará não devem ser considerados respostas finais para o problema, destacando a importância de que a última decisão é ainda por parte do projetista. Assim, este trabalho é muito importante para a agilidade e auxílio do projetista na tomada de decisão, entrando no conceito de manufatura ágil que foi tomada como inspiração para o trabalho, uma vez que isto permitirá a implementação de novos processos muito mais rapidamente.

Portanto, esta ferramenta computacional é uma proposta de Sistema de Apoio a Decisão para o projetista de arquiteturas físicas de controle que necessita conviver com o avanço tecnológico que permite lidar com sistemas cada vez mais complexos, além da evolução das normas que regem o projeto de sistemas de controle produtivos baseado em CPs.

## 8. REFERÊNCIAS BIBLIOGRÁFICAS

ABB. **Advant controller 410 Version 1.5/2.** Product Guide – 3BSE 015 966R201 Rev B. [www.abb.com](http://www.abb.com), 2001a.

ABB. **Advant controller 450 Version 2.3/1.** Product Guide – 3BSE 015 953R201 Rev B. [www.abb.com](http://www.abb.com), 2001b.

BRONSON, R. **Pesquisa operacional.** 1.ed. McGraw-Hill do Brasil, São Paulo, 1985.

CASSANDRAS, C.G. ; STRICKLAND, S.G. **Sample path properties of timed discrete event systems in discrete event dynamic systems – analizing complexity he visualization of control logic and physical machine elements within an integrated machine design and control environment.** Mechatronics, 10, p. 669–698, 2000.

CAVALHEIRO, A.C.M. **Projeto de sistemas de controle modulares e distribuídos.** 2004. 156p. Dissertação (Mestrado). Departamento de Engenharia Mecatrônica e de Sistemas Mecânicos, Escola Politécnica da Universidade de São Paulo, São Paulo. 2004.

CHOI, D.Y. **Tradutor E-MFG/SFC.** 2000. 62p. Trabalho de formatura do curso de graduação. Departamento de Engenharia Mecatrônica e de Sistemas Mecânicos, Escola Politécnica da Universidade de São Paulo, São Paulo. 2000.

CURY, J.E.R. ; QUEIROZ, M.H. **Modular supervisory control of large scale discrete-event systems.** Discrete event systems: analysis and control. Kluwer Academic Publishers, pp. 103-110; Ghent, Belgium, 2000.

KUSIAK, A. ; HE, D.W. **Design for agility: a scheduling perspective.** Robotics and Computer-Integrated Manufacturing, 14, p. 415–427, 1998.

LEWIS, R.W. **Programming industrial control systems using IEC 1131-3 (Revised Edition).** The Institution of Electrical Engineer, London, United Kingdom, 1998.

MIYAGI, P.E. **Controle programável: fundamentos do controle de sistemas a eventos discretos.** 1.ed. São Paulo: Editora Edgard Blücher, 1996.

MOORE, P.R. ; PU, J. ; NG, H.C. ; WONG, C.B. ; CHONG, S.K. ; CHEN, X. ; ADOLFSSON, J. ; OLOFGARD, P. ; LUNDGREN, J.O. **Virtual enginnering: an integrated approach to agile manufacturing machinery design and control.** Mechatronics, 13, p. 1105–1121, 2003.

MORAES, C.C. ; CASTRUCCI, P.L. **Engenharia de automação industrial.** LTC – livros Técnicos e Científicos Editora, Rio de Janeiro. 2001.

MUTHER, R. **Planejamento de layout – sistema SLP.** Edgard Blucher Ltda, São Paulo, 1978.

NOVAES, A.G. **Métodos de otimização.** Edgard Blucher Ltda, São Paulo, 1978.

PAPAIOANNOU, T. ; EDWARDS, J. **Using mobile agents to improve the alignment between manufacturing and its IT support systems.** Robotics and Autonomous Systems, 27, p. 45–57, 1999.

ROCKWELL. **Logix5000 controllers execution time and memory use reference manual.** 1756 ControlLogix®, 1769 CompactLogix, 1789 SoftLogix, 1794 FlexLogix, PowerFlex® 700S with DriveLogix 2000, Publication 1756-RM087B-EN-E, September 2003.

SANTOS FILHO, D.J. **Proposta de Mark Flow Graph estendido para a modelagem de sistemas integrados de manufatura.** Dissertação de mestrado, Escola Politécnica da Universidade de São Paulo, São Paulo, 1993.

SANTOS FILHO, D.J. **Controle de sistemas antropocêntricos de produção baseado em Redes de Petri interpretadas.** Tese de doutorado, Escola Politécnica da Universidade de São Paulo, São Carlos, 2001.

UNIVERSIDADE DE SÃO PAULO. ESCOLA POLITÉCNICA. SERVIÇO DE BIBLIOTECAS. **Diretrizes para apresentações de dissertações e teses.** São Paulo, Serviço de Bibliotecas da EPUSP. 2ed. – São Paulo, 2001. 39p.

VALDEHI, V. ; KRISHNAN, C.N. **Data token heuristic scheduling of Kalman algorithm onto a message passing multiprocessor system.** Control Eng. Practice, Vol. 5, no. 12, pp 1691-1703, 1997.

WEST, A.A. ; HARRISON, R. ; WRIGHT, C.D. ; CARROTT, A.J. **The visualization of control logic and physical machine elements within an integrated machine design and control environment.** Mechatronics, 10, p. 669–698, 2000.

WOHMAN, W.M. ; RAMADGE, P.J.G. **The control of discrete event systems.** Proceedings of the IEEE, 77, 81-98, 1989.

YOUNG, S.L. **Technology... the enabler for tomorrow's agile enterprise.** ISA Transactions, 34, p. 335–341, 1995.

YUSUF, Y.Y. ; SARHADI, M. ; GUNASEKARAN, A. **Agile manufacturing: the drivers, concepts and attributes.** International Journal of Production Economics, 62, p. 33–43, 1999.

ZHANG, J. ; GU, J. ; LI, P. ; DUAN, Z. **Object-oriented modeling of control system for agile manufacturing cells.** International Journal of Production Economics, 62, p. 145–153, 1999.