Integração de um Sistema de Planejamento de Atividades numa Célula de Montagem Robotizada

Reinaldo A. C. Bianchi, Marcus V. T. dos Santos, Márcio Rillo

Laboratório de Sistemas Integráveis (DAIA /LSI)
Escola Politécnica da Universidade de São Paulo
rbianchi-, marcus-, rillo@lsi.usp.br

Resumo

Este artigo descreve a integração de um sistema de planejamento de atividades e uma célula de montagem robotizada. Este sistema de planejamento aumenta o grau de autonomia da célula na medida em que ele determina a sequência de ações que devem ser realizadas para atigingir os objetivos propostos. São apresentadas uma descrição da célula, alguns conceitos básicos sobre sistemas inteligentes de planejamento, as características básicas do sistema utilizado e a sua integração na célula de montagem.

1. INTRODUÇÃO

Atualmente encontra-se em desenvolvimento na Divisão de Automação e Inteligência Artificial do Laboratório de Sistemas Integráveis da Escola Politécnica da Universidade de São Paulo (DAIA/LSI), uma célula de montagem flexível com 2 robôs e dois sistemas de visão, cujo propósito é realizar pequenas montagens industriais.

Com o objetivo de aumentar a autonomia da célula de montagem, no sentido de dar aos agentes e à célula como um todo uma maior capacidade de decidir autonomamente quais ações deverão ser realizadas no futuro, vários sistemas de planejamento estão em estudo na DAIA. Como parte desses estudos, o sistema NONLIN, implementado pelo do grupo do Prof. J. Hendler da Universidade de Maryland - USA, foi integrado à célula.

2. A CÉLULA DE MONTAGEM

A Célula de Montagem do LSI [Rillo, M. 1994] começou a ser construída em 1989, num projeto conjunto com o "Institut fuer Prozessrechentechnik und Robotik" (IPR) da Universidade de Karlsruhe - Alemanha, com o apoio financeiro da Fundação Volkswagen da República Federal da Alemanha e da FINEP.

Atualmente a célula possui a configuração apresentada na figura 1, sendo constituída por:

- uma estação de trabalho, onde se encontra o sistema de controle;
- duas estações de trabalho, uma das quais com uma placa digitalizadora e microcâmera, destinadas ao processamento do sistema de visão.
- dois robôs Mitsubishi RM de 5 graus de liberdade, cada um controlado por um microcomputador;
- uma câmera CCD, placa digitalizadora e um microcomputador, onde se encontra o sistema de visão bidimensional (2D).

Todos os computadores que formam a célula são interligados por uma rede ethernet local.

Pelas suas características, esta célula poderá ser aplicada na montagem de pequenos produtos industriais, tais como: pequenas peças mecânicas, brinquedos de pequeno porte, etc. É importante lembrar que a célula pode ser inserida numa linha de montagem e realizar a montagem de uma ou mais partes de um produto complexo, não necessariamente efetuando a operação de montagem do produto inteiro.

2.1 O sistema de controle

O sistema de controle da célula atualmente implementado é baseado em redes de Petri [Rillo, M. 1988]. Este sistema permite a programação das tarefas a serem realizadas pela célula através de diferentes tipos de redes, tais como: redes a condição e eventos, grafos de comando etapa transição, redes de lugares e transições (place-transition nets) e redes com marcas individuais. Este sistema está implementado em Common Lisp em uma estação e é denominado nó central da célula.

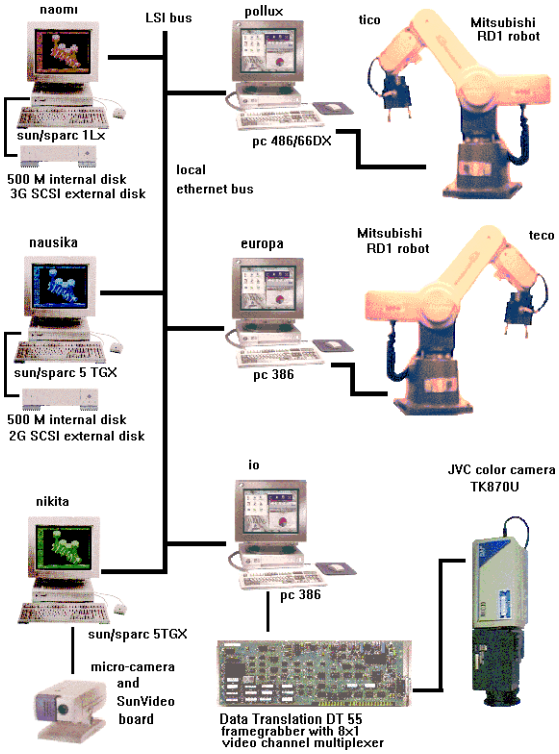

Figura 1 - Configuração atual da Célula de Montagem da DAIA/LSI.

Os sistemas de controle dos robôs são baseados em um sistema proposto por [Rillo, M. 1983], semelhante ao do nó central. A cada robô, seu controlador e microcomputador em que se encontra o sistema, dá-se o nome de nó local. Este sistema trabalha apenas com um tipo de rede de Petri, os grafos de comando etapa transição (GRAFCET), pois é suficiente para comandar as operações dos robôs. A introdução de outros tipos de redes mais poderosas pode ser feita de modo bastante simples seguindo o padrão utilizado para o nó central.

2.2 O sistema de visão

O sistema de visão 2D [Rillo, A. 1989] está implementado em um microcomputador com placa digitalizadora e uma câmera CCD. Este sistema, dotado de um sistema de controle, formam um outro nó local, que pode reconhecer vários objetos que estejam em seu campo de observação, determinando a identidade, localização e orientação das peças e transmitindo esses parâmetros para um robô manipulador. Este é capaz de pegar a peça reconhecida e manipulá-la, segundo um procedimento pré-especificado.

Uma câmera é montada verticalmente acima da cena composta pelas peças, capturando uma imagem em níveis de cinza. Esta imagem de entrada é processada por um processador digital de sinais dedicado, permitindo, desse modo, que se tenha facilidades de processamento em tempo real para os primeiros estágios do sistema de visão.

O sistema de visão 3D [Rillo, A. 1994], reconhece objetos tridimensionais rígidos em oclusão parcial, a partir de uma única imagem bidimensional.

Os programas do sistema de controle, dos nós locais que controlam os robôs e do nó local que controla o sistema 2D já estão totalmente implementados e integrados à versão atual da Célula de Montagem. Atualmente, está em fase de integração à célula o sistema de visão tridimensional.

3. SISTEMAS DE PLANEJAMENTO

Este ítem apresenta uma breve introdução à área de “Artificial Intelligent Planning” (AIP).

O que se pretende com a introdução de sistemas de planejamento na célula é aumentar o seu grau de autonomia, no sentido de deslocar o nível de programação inicial, bastante rígido, para um nível de programação de mais alto nível, próximo à descrição de objetivos a serem alcançados.

3.1 Introdução

Os primeiros sistemas desenvolvidos na área de Inteligência Artificial (IA) atendiam a uma classe de problemas que se caracterizavam, entre outros aspectos, por uma pequena interação com o ambiente onde eles eram utilizados e por não abordarem o tempo como um aspecto inerente aos problemas estudados.

Atualmente, existe uma sub-área de pesquisa em IA que se preocupa em desenvolver sistemas que possuam, entre outras características, a capacidade de desenvolver suas atividades com alto grau de autonomia, imersos em ambientes reais, com os quais eles podem interagir fortemente, e executando atividades nas quais o aspecto tempo é importante, ou seja, podem existir restrições de tempo dentro dais quais o sistema deve desenvolver suas atividades.

O desenvolvimento deste tipo de sistema, sobretudo no estudo de sistemas que possuem agentes que realizam ações num ambiente dinâmico, agregou, em torno de certos objetivos e propostas, um conjunto de pesquisadores que se dizem pesquisadores da sub-área de IA chamada “Artificial Intelligent Planning” (AIP). Esta sub-área tem crescido substancialmente nos principais encontros da área de IA.

Uma boa amostra do que vem a ser a área de AI Planning pode ser vista em [Allen; Hendler; Tate, 1990], [AIPS, 92], [AIPS, 94].

3.2 Conceitos Básicos

Em virtude deste crescimento recente, vários conceitos ainda não são universalmente aceitos e

nem mesmo a sub-área pode ser claramente delimitada.

Os primeiros sistemas desenvolvidos, a partir do início da década de 70, são chamados por vários autores como "sistemas clássicos de planejamento", embora não exista nenhum critério bem estabelecido que distingua sistemas clássicos dos demais. Esta separação tem apenas um caráter histórico e procura colocar nesta classificação a maioria dos sistemas desenvolvidos durante a década de 70 e início da década de 80 .

Na era dos sistemas clássicos, planejamento inteligente de atividades era entendido como sendo a determinação de uma seqüência de ações que leva o sistema do estado inicial ao estado final desejado *("goal state").*

Logo no início da década de 80, esta definição já não atendia mais às ambições dos sistemas que estavam sendo propostos na comunidade de AI Planning e, particularmente depois da metade da década, começaram a surgir novas definições.

3.3 Representação do mundo em estados

De modo geral, os planejadores planejam as atividades de determinados agentes, como por exemplo robôs. Estes agentes estão imersos num ambiente (algumas vezes chamado de *mundo*), e muitos planejadores possuem internamente uma representação deste ambiente na forma de estados, descritos através de alguma linguagem formal, como por exemplo lógica de predicados de primeira ordem. Esta descrição pode ser um conjunto de sentenças nesta linguagem que descreve o que é verdadeiro no instante considerado.

3.4 Ações como mudanças de estados

Uma maneira de se referir aos sistemas de planejamento é o de "sistemas que raciocinam sobre ações". Isto evidencia a importância do conceito do que seja uma ação realizada por um ou vários agentes, agentes estes que freqüentemente atuam sobre o mundo real em função de objetivos, estímulos, intenções, etc.

Muitos sistemas, que utilizam o conceito de representação do mundo por estados, definem as ações como sendo funções de um estado para outro estado.

Este tipo de interpretação de ações possui implicações importantes, uma vez que os sistemas que utilizam este conceito não consideram o que pode acontecer durante a realização da ação, pois ela apenas define qual será o estado após a sua aplicação, impedindo assim que o planejador raciocine sobre possíveis estados intermediários.

As ações são normalmente derivadas a partir de operadores. Estes operadores contêm, normalmente, um conjunto de variáveis que, uma vez instanciadas, descreverão ações específicas e individuais, caracterizando assim uma classe de possíveis ações.

3.5 Linearidade e não-linearidade

A preocupação com o conceito de linearidade e a introdução de sistemas não-lineares se verificou logo nos primeiros planejadores. Um planejador não linear é aquele que tem a habilidade de tratar metas que interagem. Desde então, tem sido utilizada em diversos sistemas, sendo citada por muitos autores como uma propriedade necessária dos sistemas de planejamento de atividades.

É importante observar que a não-linearidade é a propriedade do processo de busca usada na construção dos planos e não uma propriedade da representação do mundo. Assim, esta propriedade é importante porque pode facilitar muito o processo de planejamento, uma vez que a decisão sobre o ordenamento de certas ações pode ser postergado até a execução do plano, ou, ao menos, até o ponto onde alguma informação determine em que ordem certas ações, ainda não ordenadas entre si, devam ser ordenadas entre si.

3.6 Dependência e independência de domínio

Sistemas de planejamento ditos independentes de domínio são aqueles que possuem modos de representação de conhecimento e algoritmos que podem ser empregados numa ampla gama de aplicações. Sistema de planejamento dependentes de domínio são sistemas que usam formas de representação de conhecimento ou algum tipo de heurística específica do domínio de aplicação, o que impede sua aplicação em outro domínio que não aquele para o qual foi orientado.

3.7 Alguns sistemas clássicos

O STRIPS (**ST**anford **R**esearch **I**nstitut **P**roblem **S**olver) [Fikes & Nilsson, 1971] foi proposto em 1972 e influenciou quase todos os sistemas propostos até a metade da década de 80. Embora tenha sido aplicado na área de robôs móveis, o STRIPS é um planejador independente de domínio.

No STRIPS um estado do mundo é representado através de "fórmulas bem formadas" ("wff's") em lógica de predicados de primeira ordem (LPPO). Assim, o processo de elaboração do plano significa procurar, a partir de um estado inicial, qual a seqüência de ações que pode levar o mundo deste estado inicial a um estado final desejado, chamado estado meta ("goal").

As ações são calculadas através da instanciação de variáveis em operadores. Um operador possui um nome e pode possuir precondições, uma lista de remoção e uma lista de adição.

As precondições descrevem as condições que devem existir no mundo para que a ação instanciada possa ser aplicada. As listas chamadas de lista de adição ("add list") e lista de remoção ("delete list") descrevem os efeitos no modelo do mundo quando a ação é aplicada. A lista de adição representa as sentenças a serem inseridas no modelo do mundo existente antes da ação ser aplicada e a lista de remoção descreve as sentenças que devem ser removidas do modelo do mundo existente antes da ação ser aplicada. A lista de adição e a lista de remoção definem a transição entre dois estados. Assim, as ações podem ser descritas como funções de mudança de estados. A estratégia de busca do STRIPS utiliza de modo conveniente métodos de prova de teoremas e um método de busca no espaço de estados do mundo. Para efetuar a busca no espaço de estados do mundo o STRIPS utiliza a estratégia conhecida como análise meios fim ("means-end-analysis").

O Sistema NOAH ("**N**ets **O**f **A**ction **H**ierarchies") foi proposto também no SRI, em 1975 [Sacerdoti,1975]. Este sistema utiliza uma estrutura de dados chamada redes procedimentais ("procedural nets") para representar um plano atráves de uma seqüência de ações no tempo. Estas redes possuem alguns tipos de nós, sendo que cada um deles pode possuir ligações .

A cada nó estão associadas uma lista de adição e uma lista de remoção, que definem como determinado nó atua sobre o modelo do mundo.

Os relacionamentos entre as precondições e as submetas são inferidos pelo sistema através de apontadores indicando os nós que representam expansões de outros.

O NOAH contrói inicialmente uma rede com um único nó, cuja ação correspondente é a meta a ser atingida pelo nó. Em seguida, este nó é expandido, ordenando os nós filhos na ordem necessária para a realização da meta descrita no nó inicial. Esta expansão dá origem a uma rede, cujos nós serão novamente expandidos nos seus filhos e assim sucessivamente, até que se chegue a um nível de planejamento que possa ser executado.

Um problema importante é que o NOAH não realiza "backtracking". Isto é, uma vez feita uma escolha de decomposição, não existe possibilidade de retornar e refazer a escolha.

Um outro sistema, o NONLIN, foi desenvolvido em 1977 [Tate, 1977] e se baseou em dois sistemas anteriores: o STRIPS e o NOAH. O NONLIN é um planejador não-linear, hierárquico independente do domínio. Um plano é gerado para se atingir uma meta, podendo esta ser composta de metas múltiplas que interagem em um domínio específico.O NONLIN introduziu a possibilidade de que um sistema basicamente igual ao NOAH realizasse "backtracking".

Este planejador utiliza o chamado "Task Formalism", um mecanismo criado por Tate [1977] para especificar um domínio. Neste formalismo um domínio é especificado pelas metas, acões, operadores e predicados que o compõe.

Um outro avanço importante realizado no NONLIN foi a introdução de críticas mais poderosas. Estas analisavam o porquê de cada meta introduzida durante o processo de planejamento. Essa análise permitia uma maior possibilidade de determinação e de solução de conflitos e, portanto, de raciocínio, por exemplo, se uma meta que está causando conflito não poderia ser retirada através da alteração do operador que lhe deu origem.

Existe, na DAIA, uma cópia do sistema NONLIN que foi ampliada pelo grupo do Prof. J. Hendler da Universidade de Maryland - USA. Este sistema foi implementado em Common Lisp e baseado no NONLIN original

O NONLIN é executado a partir do interpretador do Common Lisp, tendo como entrada a definição dos operadores e ações do domínio, pedindo interativamente 3 listas: a primeira com os fatos sempre verdadeiros; a segunda, descrevendo o estado inicial do mundo; a terceira, com as metas a serem atingidas. A saída é uma lista com o plano gerado.

4. A IMPLEMENTAÇÃO ATUAL

O NONLIN foi implementado na Célula utilizando a World Wide Web como interface com o usuário. Através dela, o usuário fornece os dados iniciais e as metas, recebendo o plano gerado.

O domínio usado nesta aplicação é o do mundo de blocos ("blocks world"). Este consiste em blocos de madeira que estão sobre uma mesa e são identificados por letras . Os blocos podem ser movimentados sobre a mesa, empilhados e desempilhados. Um dos motivos da escolha deste domínio é que ele possui algumas características que facilitam a aplicação de braços robóticos na movimentação de objetos. Além disto, este domínio serveria como protótipo para o estudo de aplicações mais sofisticadas dos braços robóticos.

Os operadores usados neste domínio são: *cleartop b* e *on a b*. Eles descrevem, respectivamente, quando o bloco *b* não possui nenhum bloco em seu topo e quando o bloco *a* está sobre *b*, *b* podendo ser um bloco ou a mesa.

A seguir, um exemplo de descrição de um problema:

fatos sempre válidos: *((cleartop table))*

modelo inicial do mundo:

((on b table) (on a b) (on c a) (cleartop c))

lista de metas:

((on a b) (on b c) (on c table))

O plano que o Nonlin gera para a solução deste problema é o seguinte:

puton c table

puton a table

puton b c

puton a b

O estado inicial do mundo está descrito em dois arquivos. O primeiro contém as informações sobre o posicionamento relativo dos blocos (um bloco em relação ao outro) e tem a seguinte sintaxe:

(on <obj1> <obj2>).

onde *<obj1>* e *<obj2>* são blocos.

O outro arquivo contém o posicionamento absoluto dos blocos (em relação ao sistema de coordenadas do robô), e possui a seguinte sintaxe:

(at <obj> <posição> <altura>), onde:

<obj> é o mesmo símbolo *x* de *puton x y* e representa o objeto;

<posição> é a posição em que se encontra inicialmente o objeto;

<altura> é a altura em que se encontra inicialmente o objeto.

As posições absolutas dos blocos constituem uma representação interna da aplicação, sendo invisível ao usuário.

Ao ser requisitada a página que controla o planejador, o servidor da WWW verifica se existem os arquivos com o estado inicial do mundo. Se os arquivos existem, o servidor apresenta uma página com o estado do mundo e que pede ao usuário as metas. Se estes arquivos não existem, o servidor envia uma página que permite ao usuário acessar o planejador, mas que não realiza os movimentos na célula. Isto é feito porque a célula não está disponível para o uso desse sistema em tempo integral, mas tem sua utilização compartilhada por vários projetos.

Após o fornecimento das metas pelo usuário, o cliente as envia para o servidor, que cria um arquivo com as metas e a descrição do mundo relevante ao planejamento (somente as posições relativas). O servidor executa um programa que controla a execução em "batch" do NONLIN dentro do interpretador Common Lisp. Esta chamada é feita usando a função "system" da linguagem C. Para poder ser executado em "batch", pequenas modificações foram realizadas no NONLIN, permitindo que as entradas de dados e a saida do plano sejam realizadas via arquivo.

Ao final do planejamento, um programa de pós-processamento, escrito em Common Lisp, é executado automaticamente para criar um arquivo que possa ser executado pelos robôs. Esse arquivo é criado a partir do plano gerado pelo NONLIN e da descrição do estado inicial do mundo.

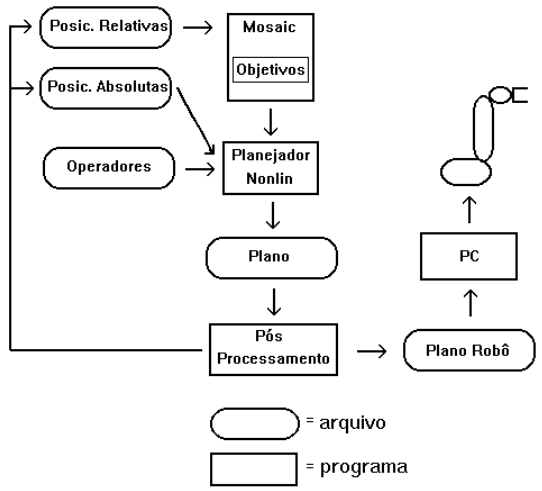

Figura 2: Implementação atual do sistema.

Antes da geração do plano o programa lê os arquivos com as posições relativas e absolutas dos blocos. Monta-se então uma estrutura de dados do tipo lista que representará o modelo do mundo durante a execução das operações com os blocos. Após o NONLIN gerar o plano, o programa LISP tranforma cada operação *puton x y* em duas outras operações *goto px hx*, *moveto py hy*, que são gravadas em um outro arquivo. Cada operação *puton x y* também altera a estrutura de dados que representa o modelo do mundo, pois o objeto *x* após esta operação passará a ocupar a posição do objeto *y* e estará em uma altura igual a *<altura de x> + <altura de y>*.

A operação *goto px py* indica que o braço robótico deve deslocar se com a garra aberta até a posição *(px,hx)* e em seguida fechá-la. A operação *moveto px hx* indica que o braço robótico deve se deslocar com a garra fechada até a posição *(px, hx)* e em seguida abri-la.

Outro objetivo deste pós-processamento é criar um arquivo com o estado final do mundo, que será o estado inicial na próxima execução do planejador.

Um programa é executado ininterruptamente no microcomputador que controla o robô, esperando a existência de um arquivo com um plano para o robô. Cada vez que esse arquivo existe, é lido pelo programa que realiza o plano fisicamente.

O servidor da WWW também fica esperando o final do planejamento pelo NONLIN. Quando isto ocorre, ele lê o arquivo com o plano e o apresenta em uma página. No futuro, deseja-se apresentar o video do plano sendo realizado pelo robô nesta página.

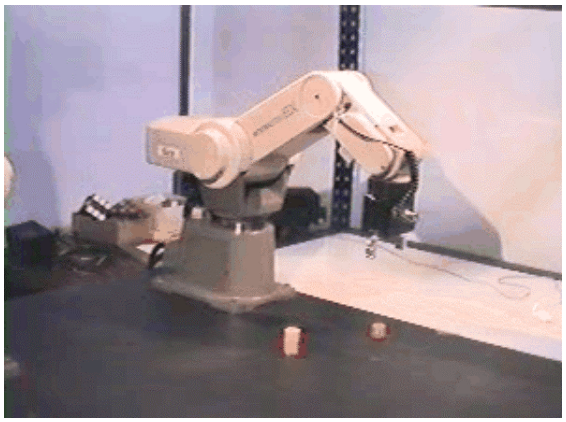

Figura 3: O robô e os blocos.

Para simplificar o domínio de trabalho, visto que trabalha-se sem o sistema de visão e sem perda de generalidade, foram feitas algumas modificações na descrição da célula. Inicialmente, as posições absolutas da área de trabalho da célula foram discretizadas em uma matriz de seis posições, cada posição com seis alturas. Além disso, estas posições estão dispostas cilindricamente em torno do robô, o que evita necessidade do controle da posição do punho do robô. Uma última simplificação é a utilização de somente um dos braços robóticos da célula, pois não são necessários os dois braços existentes para esta implementação.

Finalmente, o cliente da WWW usado é o Netscape: uma ferramenta parecida com o Mosaic, mas que possui mais recursos gráficos.

A página hipertexto da www inicial é acessível no endereço http://www.lsi.usp.br/~daia/celula/NITC. É uma página que explica o projeto, possuindo videos, fotos e um hiperlink para outra página que controla o planejador.

Esta implementação foi escrita em Common Lisp, linguagem C e HTML.

5. CONSIDERAÇÕES FINAIS

Diversos sistemas de planejamento estão sendo desenvolvidos na DAIA/LSI, sendo um baseado em casos (“case based planning”), um baseado em lógica temporal, outro baseado em lógica de transações e um sistema fortemente reativo. A troca de informações entre os pesquisadores envolvidos no desenvolvimento destes sistemas foi de grande importância para a implementação do sistema descrito neste artigo.

Quanto ao programa que faz o pós-processamento do plano gerado pelo NONLIN, não haveria a necessidade de sua utilização se os operadores do mundo fornecidos pelo planejador fossem alterados no sentido de considerarem as posições e alturas dos blocos.

Sabe-se que esta solução é a mais elegante. No entanto, escolheu-se o pós-processamento, pois a integração do NONLIN à célula, o objetivo primordial do trabalho, ocorreria de forma mais rápida. Além disso, o pós-processamento possibilita a utilização do controle com o GRAFCET, uma maior flexibilidade em relação ao planejador usado e facilita a conexão com o sistema de visão .

A introdução de um sistema clássico já pronto como o NONLIN, tem o objetivo de facilitar o aprendizado da utilização do planejamento em uma aplicação real. Os resultados obtidos deste estudo foram importantes para o desenvolvimento dos outros sistemas que estão sendo implementados.

6. REFERÊNCIAS BIBLIOGRÁFICAS

AIPS. THE FIRST INTERNATIONAL CONFERENCE ON ARTIFICIAL INTELLIGENCE, **Proceedings**, James Hendler, jun 1992.

AIPS. THE SECOND INTERNATIONAL CONFERENCE ON ARTIFICIAL INTELLIGENCE PLANNING SYSTEMS, **Proceedings**, Kristian Hammond, jun 1994.

ALLEN, J.F.; HENDLER, J.A.; TATE, A. (editores) **Readings in planning.** San Mateo, Morgan Kaufmann, 1990.

FIKES,R.E.; NILSSON, N.J. “STRIPS : A new approach to the application of theorem proving to problem solving” in **Artificial Intelligence**, v.2, p.189-208, 1971.

RILLO, A. **Sistema de visão binária para peças parcialmente oclusas.** (Mestrado) São Bernardo do Campo, Faculdade de Engenharia Industrial - Fundação de Ciências Aplicadas (FEI - FCA), 1989.

RILLO, A.. **RECTRI: um sistema de reconhecimento tridimensional a partir de uma única imagem de intensidade luminosa**. (Doutorado) São Paulo, Escola Politécnica, Universidade de São Paulo, 1994.

RILLO, M. **Controlador lógico programável usando grafos de comando etapa transição**. (Mestrado) São Paulo, Escola Politécnica, Universidade de São Paulo, 1983.

RILLO, M. **Aplicações de redes de Petri em sistemas de manufatura**. (Doutorado) São Paulo, Escola Politécnica, Universidade de São Paulo, 1988.

RILLO, M. **Uma célula robotizada de montagem com alto grau de autonomia**. (Livre Docência) São Paulo, Escola Politécnica, Universidade de São Paulo, 1994.

SACERDOTI, E.D. “The nonlinear nature of plans”. In: INTERNATIONAL JOINT CONFERENCE ON ARTIFICIAL

INTELLIGENCE, 2, 1975. **Proceedings**. p.206-214.

TATE, A. "Generating project networks". In : INTERNATIONAL JOINT CONFERENCE ON ARTIFICIAL INTELLIGENCE, 3, 1977. **Proceedings**. p.888-889.