

Geração e Representação de Planos para o Planeamento do Processo

Neste artigo vamos descrever um sistema designado TPMS (*Task Planning for Manufacturing Systems*) que tem por finalidade efectuar a geração automática de planos simbólicos para tarefas de fabrico. Este sistema é especialmente útil como auxiliar do Planeamento do Processo, podendo ser também utilizado para estimar o plano óptimo, auxiliando desse modo a fase de Planeamento da Produção, embora não seja concebido para o planeamento operacional de tarefas industriais.

O sistema TPMS é capaz de gerar todos os planos, mas só o faz quando o problema não conduz à explosão combinacional. Nestes casos é gerado apenas o melhor plano, atendendo ao critério de minimização do tempo de execução.

Palavras-chave: Planeamento de Tarefas, Sistemas Flexíveis de Fabrico, Restrições

1. Estado da Arte

Várias abordagens têm sido utilizadas para o Planeamento e Execução Automática de Tarefas [1 - 15]. Uma classificação dessas abordagens pode ser encontrada na referência [16]. No entanto, a maior parte apenas considera o planeamento simbólico a alto nível, não se preocupando com a programação das máquinas ou robôs.

Bourjault [4] descreveu um sistema interactivo para geração da sequência de montagem de um produto. É criado um grafo representando as conexões entre os diversos componentes do produto. Uma vez que o sistema não dispõe de capacidades de inferência sobre raciocínio geométrico relativo à montagem, este conhecimento é obtido através da apresentação de perguntas ao utilizador.

De Fazio [5] melhorou esta metodologia, reduzindo o número de perguntas colocadas ao utilizador permitindo-lhe escrever fórmulas lógicas representando a sequência de operações. É, também, gerado um grafo representando as conexões entre os diversos componentes do produto.

Homem de Mello [6] usou grafos E/OU para representar montagens, originando uma redução drástica no número de nós. No entanto, o problema da explosão combinacional pode surgir. É necessária a realização de alguns testes para verificação da viabilidade de execução do plano.

Wilson [7] melhorou a eficiência da geração de planos a partir de grafos E/OU, conjugando a geração de uma sequência de montagem com o raciocínio geométrico que este tipo de problemas deve envolver, sendo geradas apenas as sequências factíveis.

Sukhan Lee [8] desenvolveu um sistema caracterizado por uma aproximação distribuída e cooperativa ao planeamento utilizando o paradigma "Blackboard". Para o processo

de geração do plano é usado o planeamento de montagem em ordem inversa.

O sistema de Huang e Lee [9, 10] aceita como entrada a descrição de um produto efectuada por um sistema CAD e gera um plano de montagem, sujeito às restrições impostas pelos recursos existentes na célula de montagem. Existe uma base de dados do "mundo", representando os componentes, as relações entre eles, operações industriais e as máquinas e ferramentas da célula de montagem. O gerador de planos efectua inferências sobre um conjunto de regras de produção do tipo "SE <pré-condições> ENTÃO <acções>", em que as acções correspondem à criação de uma lista de junção e de uma lista de remoção. O plano é obtido pela inversão da sequência de desmontagem.

Os sistemas anteriores não consideram a existência de um plano de execução, i. e., como programar as máquinas e/ou os robôs para executarem o plano simbólico de alto nível gerado. Contudo, alguns sistemas tratam este problema.

O grupo de Lozano-Perez [11] implementou um sistema para realizar operações de agarrar e largar no âmbito de problemas do tipo pegar e colocar, admitindo que o plano para a tarefa já se encontra disponível. Contudo este sistema tem algumas limitações no que se refere à capacidade de raciocínio acerca das acções a realizar.

Hutchinson [12] apresentou outra aproximação interessante a este problema utilizando uma hierarquia de planeamento a três níveis: nível operacional, nível geométrico e nível de redução de incertezas. Contudo o problema da análise de colisões não é considerado neste sistema.

TPMS utiliza algum do trabalho desenvolvido no CIARC (*Cooperative Intelligent Assembly Robotic System*). CIARC foi implementado para a execução e planeamento de tarefas de manipulação e montagem, utilizando um braço de robô programado através de uma linguagem textual [13, 14, 15]. Embora incluindo conceitos de inteligência e flexibilidade, CIARC foi desenvolvido para a montagem e manipulação robótica, controlando a operação de um único robô.

TPMS é a extensão de CIARC para sistemas de fabrico. O objectivo final é a geração automática de programas ou conjuntos de comandos para robôs, AGV's, máquinas de controlo numérico e outras componentes dos sistemas, sem a necessidade de gastar muito tempo na programação dos diversos componentes do sistema de fabrico. Pretende-se, pois, desenvolver um sistema capaz de substituir ou ajudar o programador da célula de montagem, i.e. obter automaticamente um plano para a tarefa industrial e gerar o código de controlo (programa, instruções) de cada componente (máquina, robô, ...) da célula de montagem. Neste artigo apenas iremos tratar do planeamento de alto nível, não nos debruçando sobre a geração do código para o controlo das máquinas.

2. Planeamento de Alto Nível

2.1. Introdução

O Planeamento e Execução de Tarefas envolve duas fases: em primeiro lugar é gerado e representado automaticamente um plano de alto nível composto por operadores simbólicos (e.g. soldar A com B, inserir B em C); em seguida são gerados os programas para controlar os diversos componentes do sistema de fabrico (e.g. um programa de controlo de um robô para inserir um objecto no buraco de outro objecto).

O planeamento de alto nível permite a geração de planos considerando todas as restrições de processamento, de factibilidade e geométricas. É também possível gerar não apenas um, mas vários planos que podem ser activados dinamicamente, de acordo com a situação corrente.

2.2 Conceitos Básicos

O planeamento é uma área de investigação, na qual se procura responder à questão de "como resolver um problema", ou por outras palavras, de "como atingir um objectivo ou um conjunto de objectivos". Para resolver este problema é necessário passar de um estado inicial para o estado final desejado. Vamos considerar algumas definições básicas de conceitos utilizados no Planeamento de Tarefas.

Estado - um estado é uma representação que descreve a situação em que se encontra um dado "mundo" relativo ao domínio para o qual estamos a planear. Por exemplo, num estado de uma montagem mecânica estão representadas as posições (X, Y, Z) e orientações (O) das partes (objectos) que integram a montagem, podendo inclusive serem obtidas automaticamente as relações simbólicas entre partes (por exemplo, relações do tipo "inserido em", "em cima de") [13 - 15].

Operador - um operador é uma entidade que pode ser utilizada para alterar o "mundo" relacionado com o nosso domínio. Em termos industriais podemos identificar vários operadores como por exemplo: furar (Ob), inserir (Ob1,Ob2), soldar (Lista_Ob), transportar (Ob), decapar (Fio, Extremidade), etc. Os argumentos dos operadores devem ser considerados como variáveis não instanciadas.

Operação - uma operação corresponde à instanciação de um operador, ou seja, os argumentos dos operadores tomam o valor de objectos concretos (inserir (saleiro, base), furar (tampa),...).

Plano - um plano é um conjunto de operações. Note-se que usamos o termo conjunto ao invés de sequência pois nada diz que as operações do plano tenham que ser efectuadas sequencialmente (como no "job shop"). De facto poderemos ter paralelismo na execução de operações para a mesma tarefa.

Restrições - as restrições correspondem a conhecimento acerca das operações de uma dada tarefa que indica que uma dada operação tem que ser efectuada antes de outra. No fundo, são as restrições que irão definir as precedências no Grafo de Restrições da Tarefa (ver exemplo da figura 2). Do ponto de vista industrial foram classificados três tipos de restrições: restrições de factibilidade, restrições geométricas e restrições de processamento [1 -3].

3. Um Exemplo

Vamos considerar um exemplo de montagem de uma cablagem eléctrica [17] (figura 1). Neste exemplo utilizamos 16 componentes: 4 fios (W1, W2, W3, W4) com comprimentos diferentes, 3 terminais normais (L1, L2, L3), 3 terminais circulares (R11, R12, R13), 2 termi-

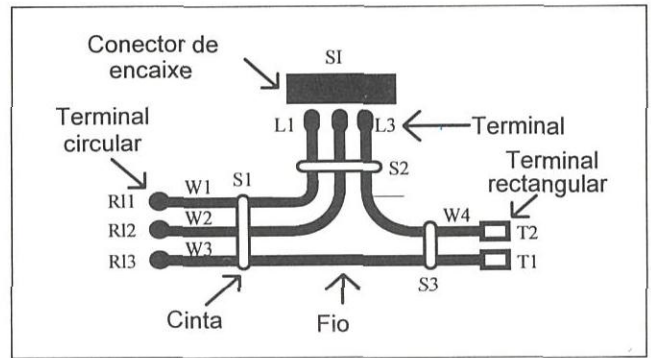


Figura 1 - Exemplo de montagem de uma cablagem eléctrica

nais rectangulares (T1, T2), 1 conector de encaixe (SI) e 3 cintas (S1, S2, S3).

Os 4 fios necessitam ser cortados e decapados por uma máquina de corte e as suas extremidades devem levar conectores normais, conectores circulares e terminais, sendo a união estabelecida através de máquinas de soldar. De seguida deve ser inserido um conector de encaixe. A cablagem é então montada num painel. Uma vez que os fios tenham sido correctamente moldados, os fios da cablagem podem ser agrupados por cintas.

À priori, a ordem através da qual as partes vão ser processadas e montadas não é conhecida. A sequência de operações do plano vai ser obtida tendo em atenção as restrições de factibilidade e geométricas entre partes e ferramentas, sendo necessário considerar também a existência de restrições de processamento.

Para a tarefa descrita neste exemplo os recursos a utilizar são os seguintes:

- uma máquina para pôr cintas
- uma máquina de corte
- uma máquina de soldadura
- um robô

As operações a considerar são as seguintes:

- cortar e decapar W1 na extremidade 1 → *cut* (W1, 1)
- cortar e decapar W1 na extremidade 2 → *cut* (W1, 2)
- cortar e decapar W2 na extremidade 1 → *cut* (W2, 1)
- cortar e decapar W2 na extremidade 2 → *cut* (W2, 2)
- cortar e decapar W3 na extremidade 1 → *cut* (W3, 1)
- cortar e decapar W3 na extremidade 2 → *cut* (W3, 2)
- cortar e decapar W4 na extremidade 1 → *cut* (W4, 1)
- cortar e decapar W4 na extremidade 2 → *cut* (W4, 2)
- soldar R11 e W1 → *solder* (R11, W1)
- soldar R12 e W2 → *solder* (R12, W2)
- soldar R13 e W3 → *solder* (R13, W3)
- soldar L1 e W1 → *solder* (L1, W1)
- soldar L2 e W2 → *solder* (L2, W2)
- soldar L3 e W4 → *solder* (L3, W4)
- soldar T1 e W3 → *solder* (T1, W3)
- soldar T2 e W4 → *solder* (T2, W4)
- inserir SI em [W1, W2, W4] → *insert* (SI, [W1, W2, W4])
- moldar W1 → *lay* (W1)
- moldar W2 → *lay* (W2)
- moldar W3 → *lay* (W3)
- moldar W4 → *lay* (W4)
- cintar S1 em [W1, W2, W3] → *strap* (S1, [W1, W2, W3])
- cintar S2 em [W1, W2, W4] → *strap* (S2, [W1, W2, W4])
- cintar S3 em [W3, W4] → *strap* (S3, [W3, W4])

Contudo, não sabemos a ordem através da qual essas operações serão executadas. Essa decisão só poderá ser tomada depois da análise de um grafo que represente as restrições. As restrições a considerar neste caso são:

restrições de processamento:

- neste tipo de montagens devemos inserir os fios no conector de encaixe antes de os moldar, ou seja:

- *insert* (SI, [W1, W2, W4]) antes de *lay* (W1)
- *insert* (SI, [W1, W2, W4]) antes de *lay* (W2)
- *insert* (SI, [W1, W2, W4]) antes de *lay* (W4)

restrições de factibilidade:

- só faz sentido soldar os fios aos terminais se estes estiverem cortados e decapados:

- *cut* (W1, 1) antes de *solder* (R11, W1)
- *cut* (W1, 2) antes de *solder* (L1, W1)
- *cut* (W2, 1) antes de *solder* (R12, W2)
- *cut* (W2, 2) antes de *solder* (L2, W2)
- *cut* (W3, 1) antes de *solder* (R13, W3)
- *cut* (W3, 2) antes de *solder* (T1, W3)
- *cut* (W4, 1) antes de *solder* (L3, W4)
- *cut* (W4, 2) antes de *solder* (T2, W4)

- ao colocar o conector de encaixe os fios já devem estar soldados aos terminais:

- *solder* (L1, W1) antes de *insert* (SI, [W1, W2, W4])
- *solder* (L2, W2) antes de *insert* (SI, [W1, W2, W4])
- *solder* (L3, W4) antes de *insert* (SI, [W1, W2, W4])

restrições geométricas:

- quanto à moldagem se as cintas fossem colocadas antes de moldar veríamos que não era possível obter a forma desejada:

- *lay* (W1) antes de *strap* (S1, [W1, W2, W3])
- *lay* (W2) antes de *strap* (S1, [W1, W2, W3])
- *lay* (W3) antes de *strap* (S1, [W1, W2, W3])
- *lay* (W1) antes de *strap* (S2, [W1, W2, W4])
- *lay* (W2) antes de *strap* (S2, [W1, W2, W4])
- *lay* (W4) antes de *strap* (S2, [W1, W2, W4])
- *lay* (W3) antes de *strap* (S3, [W3, W4])
- *lay* (W4) antes de *strap* (S3, [W3, W4])

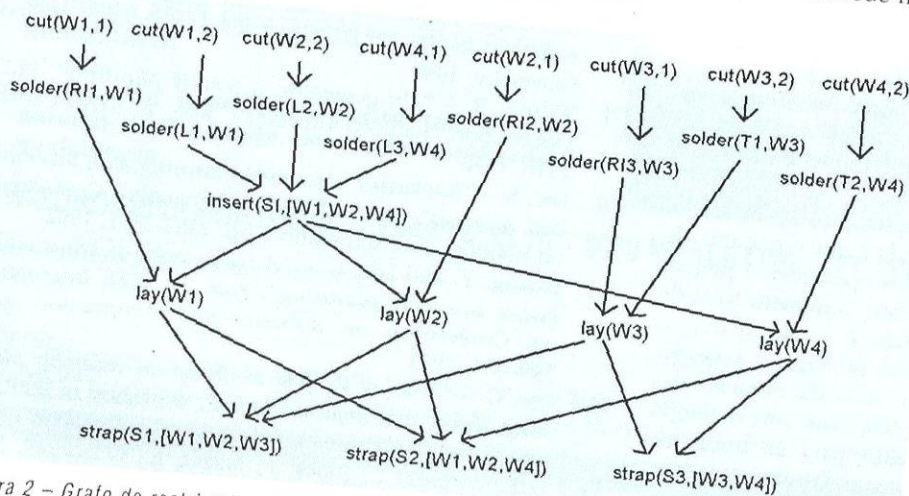


Figura 2 - Grafo de restrições

Com base nas restrições apresentadas, o sistema TPMS começa por construir automaticamente um grafo de restrições (figura 2). Um plano pode ser obtido directamente percorrendo esse grafo.

O algoritmo para gerar o plano é o seguinte:

```

|
| repetir
| {
|   escolher uma operação sem precedências que se dirija
|   ao correspondente nó N
|   remover N e todas as precedências que se iniciem em N
| }
| até que número de nós = 0
|

```

Se, no entanto, pretendêssemos gerar todos os planos possíveis, poderíamos utilizar um algoritmo similar que inclua *backtracking* (neste caso, nós e precedências seriam marcados em vez de removidos).

No entanto poderíamos ter o problema da "explosão combinacional", o que acontece especialmente em grafos fortemente conectados com muitos ramos paralelos. No exemplo que consideramos, apenas tendo em atenção as 8 operações de corte, que podem ser feitas em qualquer ordem (notar o paralelismo do grafo da figura 2) teremos 8! combinações nos planos que começam com 8 operações de corte. Logo, a geração de todos os planos é intratável do ponto de vista computacional.

O sistema TPMS dispõe de uma rotina capaz de gerar todos os planos, mas que só é usada quando o problema não conduz à explosão combinacional. Nos casos mais complexos o sistema TPMS dispõe de um algoritmo que permite a geração do melhor plano atendendo ao critério da minimização do tempo de execução do plano (no fundo o principal critério usado na selecção do plano).

Esse algoritmo toma em consideração o tempo de processamento das operações (inerente a cada operação) e o tempo de atraso originado pela realização de uma operação antes ou depois de outra (por exemplo o tempo necessário para uma mudança de ferramenta).

A cada nó (que corresponde a uma operação) é associado um "custo", que é obtido pela adição do "custo" dos dois componentes (tempo de processamento e tempo de atraso) anteriormente descritos, com o "custo" do nó "pai" (nó que o antecede no grafo).

O custo do nó inicial (nó zero) é zero.

É também necessário estimar o "mérito" de cada nó que é gerado, de forma a tornar possível a selecção, em primeiro lugar, dos caminhos mais promissores. Essa estimativa é efectuada somando os tempos de todas as operações de mudança de ferramentas necessárias para obter o plano. Uma vez que esta é uma estimativa por defeito do tempo total necessário para a execução do plano, é garantido que o algoritmo encontra um plano óptimo (tal como definido anteriormente). O algoritmo é o seguinte:

```

{
  o primeiro nó tem o número 0 (considerando todas as
  operações e restrições)
  o nó 0 é um nó terminal
  i = 0
  repetir
  {
    expandir nó i
    escolher o próximo nó para expandir
  }
  até que não exista nenhum não terminal com custo
  inferior ao do nó terminal
}

```

Para expandir um nó:

```

{
  repetir
  {
    seleccionar uma operação i sem precedências que cheguem ao correspondente nó N
    remover a operação i da lista de operações
    remover todas as precedências que se iniciem no nó N da lista de precedências
    gerar um novo nó com nova operação e lista de precedências
    marcar o novo nó como nó terminal
  }
  até que não exista nenhuma operação com precedências chegando ao correspondente nó N
}

```

Para escolher um nó:

```

{
  seleccionar todos os nós terminais
  calcular o custo de todos os nós terminais
  escolher o nó com menor custo
}

```

```

cut(w1, 1)
cut(w1, 2)
cut(w2, 1)
cut(w2, 2)
cut(w3, 1)
cut(w3, 2)
cut(w4, 1)
cut(w4, 1)
solder(r1, w1)
solder(l1, w1)
solder(l2, w2)
solder(l3, w4)
solder(r12, w2)
solder(r13, w3)
solder(t1, w3)
solder(t2, w4)
insert(si, [w1, w2, w4])
lay(w1)
lay(w2)
lay(w3)
lay(w4)
strap(s1, [w1, w2, w3])
strap(s2, [w1, w2, w4])
strap(s3, [w3, w4])

```

Figura 3
plano óptimo

Neste algoritmo é considerada a configuração do sistema de fabrico, uma vez que a sequência de operações é importante devido à inclusão do tempo necessário para mudar de uma operação para a seguinte.

Utilizando a metodologia descrita é possível gerar um plano óptimo (figura 3), uma vez que todos os tempos envolvidos nas operações e nas trocas de ferramentas são considerados.

No exemplo, apresentado, atendendo a que o número de diferentes operações é pequeno e a que o número total de operações é muito maior, o problema principal a considerar não é o da duração das operações, mas sim o tempo necessário para as trocas de ferramentas. É por este motivo que o melhor plano é aquele que minimiza o número de trocas de ferramentas.

4. Conclusões e Trabalho Futuro

Este artigo apresentou um sistema de apoio ao Planeamento do Processo que permite a geração automática de planos constituídos por operações simbólicas, com a possibilidade de geração do plano óptimo. O sistema TPMS apropriado para processos industriais não sequenciais, com alto grau de paralelismo de operações, sendo mais indicado para processos de produtos com ciclos de vida reduzido com bastantes variantes (por exemplo, cablagens eléctricas de automóveis).

Um dos passos a médio prazo no desenvolvimento deste sistema é a conversão automática de operações simbólicas em código de controlo de equipamento (por exemplo em programas para robôs, PLC's ou máquinas de Controlo Numérico). Outro desenvolvimento deste sistema será a sua integração com um Sistema de Apoio à Decisão para Escalonamento Dinâmico atendendo a Prazos de Entrega [18].

Referências

- [1] Rocha, J. and Ramos, C. – "Task and Execution Planning for Flexible Manufacturing Systems", Proc. of the Second International Conference on Integrate Logistics and Concurrent Engineering - ILCE'94, pp. 131-139, Montpellier France, 1994
- [2] Rocha, J. and Ramos, C. – "Generating and Converting Plans for Intelligent Manufacturing Systems", Proc. of IEEE International Symposium on Industrial Electronic - ISIE'94, pp. 304-309, Santiago, Chile, 1994
- [3] Rocha, J. and Ramos, C. – "Task Planning for Flexible and Agile Manufacturing Systems", Proc. of IEEE International Conference on Intelligent Robots and Systems - IROS'94, pp. 105-112, Munich, Germany, 1994
- [4] Bourjault, A. – "Contribution a une approche methodologique de l'assemblage automatisé: Elaboration automatique des séquences opératoires", Thèse d'État, Université de Franche-Comté, Besançon, France, 1984
- [5] De Fazio T. and Whitney, D. – "Simplified generation of all mechanical assembly sequences", IEEE Journal of Robotics and Automation, vol. 3, n. 6, pp. 640-658, December 1987
- [6] Homem de Mello, L. – "Task sequence planning for robotic assembly", PhD Dissertation, Carnegie Mellon University, 1989
- [7] Wilson, R. – "On geometric assembly planning", PhD Dissertation, Stanford University, Rep. n. STAN-CS-9:1416, 1992
- [8] Lee, S. – "Backward Assembly Planning with assembly cost analysis", Proc. of IEEE International Conference on Robotics and Automation, pp. 2382-2391, 1992
- [9] Huang, Y. and Lee, C. – "A framework of knowledge based assembly planning", Proc. of IEEE International Conference on Robotics and Automation, pp. 599-604, 1991
- [10] Lee, C. – "Proc. of the first workshop on assembly planning: theory and implementation", Workshop of IEEE International Conference on Robotics and Automation 1992
- [11] Lozano-Pérez, T.; Jones, J.; Mazer, E.; O'Donnel, P. – "Task level planning of pick-and-place robot motions", IEEE Computer, vol. 22, n. 3, pp. 21-29, March 1989

- [12] Hutchinson, S. and Kak, A. - "SPAR: A planner that satisfies operational and geometric goals in uncertain environments", AI Magazine, Spring 1990, pp. 30-61, 1990
- [13] Ramos, C. and Oliveira, E. - "Closing the loop of Task Planning, Action and Sensing", Proc. of IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 909-916, Raleigh, NC, USA, 1992
- [14] Ramos, C. and Oliveira, E. - "Sensor-based Reactive Planning and Execution for Robotics Assembly Tasks", Proc. of IEEE International Conference on Systems Engineering, pp. 135-138, Kobe, Japan, 1992
- [15] Ramos, C. and Oliveira, E. - "CIARC: A Multi-Agent Community for Intelligent Assembly Robotics Systems with Real-Time Capabilities", Proc. of IEEE International Conference on Systems, Man and Cybernetics, Le Touquet, France, 1993
- [16] Gottschlich, S; Ramos, C.; Lyons, D. - "Assembly and Task Planning - a Taxonomy", IEEE R&A Magazine, n. 3, pp. 4-12, September 94
- [17] Aguirre, E and Raucant, B - "Performances of Wire Harness Assembly Systems", Proc. of IEEE International Symposium on Industrial Electronics - ISIE'94, pp. 292-297, Santiago, Chile, 1994
- [18] Ramos, C; Almeida, A. and Vale, Z. - "Um Sistema de Apoio à Decisão para Escalonamento Dinâmico com Prazos de Entrega", 4th Jornadas Luso-Espanholas de Engenharia Electrotécnica, Junho 1995

* Departamento de Engenharia Informática ISEP

** Departamento de Engenharia Informática ISEP

*** Departamento de Engenharia Electrotécnica FEUP

Sistemas de Accionamento

A Qualidade está ao alcance dos fabricantes e utilizadores portugueses de bens de equipamento. Escolher a SEW-EURODRIVE é apostar na competitividade e profissionalismo. A SEW-EURODRIVE investiu em Portugal.

- Moto-redutores e moto-variadores-redutores com e sem freio. Sistema modular.
- Conversores de frequência MOVITRAC®. Cartas opcionais para regulação de velocidade com precisão e sincronização da velocidade angular.
- Servo-motores e servo-motores-redutores trifásicos. Variadores electrónicos MOVI-DYN® para servo-motores.
- Software SEW para sistemas de accionamento SEW.
- Centro de produção, apoio técnico e serviço, na Mealhada. Uma rede mundial de assistência.



SEW-EURODRIVE PORTUGAL - APARTADO 15 - 3050 MEALHADA - TEL (031) 23 684 - FAX (031) 23 685

PEDIDO DE ENVIO DE DOCUMENTAÇÃO <input type="checkbox"/>	VISITA <input type="checkbox"/>
EMPRESA _____	DEP. _____
NOME: _____	FUNÇÃO: _____
ENDEREÇO _____	
COD.POSTAL _____	TEL _____ FAX _____