

FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO



FEUP

Logística Flexível Baseada em AGVs

Luís André Freitas da Rocha

Dissertação realizada no âmbito do
Mestrado Integrado em Engenharia Electrotécnica e de Computadores
Major Automação

Orientador: Prof. Dr. António Paulo Gomes Mendes Moreira

Co-orientador: Prof. Dr. Américo Lopes de Azevedo

Julho de 2010

A Dissertação intitulada

“LOGÍSTICA FLEXÍVEL BASEADA EM AGVs”

foi aprovada em provas realizadas em 23/Julho/2010

o júri



Presidente Professor Doutor José António Rodrigues Pereira de Faria
Professor Auxiliar do Departamento de Engenharia Industrial e Gestão da
Faculdade de Engenharia da Universidade do Porto



Professor Doutor Eusébio Manuel Pinto Nunes
Professor Auxiliar do Departamento de Produção e Sistemas da
Universidade do Minho



Professor Doutor António Paulo Gomes Mendes Moreira
Professor Auxiliar do Departamento de Engenharia Electrotécnica e de Computadores da
Faculdade de Engenharia da Universidade do Porto



Professor Doutor Américo Lopes de Azevedo
Professor Associado do Departamento de Engenharia Industrial e Gestão
da Faculdade de Engenharia da Universidade do Porto (Co-Orientador)

O autor declara que a presente dissertação (ou relatório de projecto) é da sua exclusiva autoria e foi escrita sem qualquer apoio externo não explicitamente autorizado. Os resultados, ideias, parágrafos, ou outros extractos tomados de ou inspirados em trabalhos de outros autores, e demais referências bibliográficas usadas, são correctamente citados.



Autor - Luis André Freitas da Rocha

Faculdade de Engenharia da Universidade do Porto

Resumo

Automated Guided Vehicles (AGVs) são veículos autónomos usados no transporte de materiais entre postos de trabalho, sem intervenção de um operador, mas também, e mais recentemente, no desempenho de operações de vigilância e exploração. São largamente utilizados em sistemas de manuseamento de materiais bem como sistemas de manufactura flexíveis, em que uma constante alteração da produção é expectável de ocorrer.

Hoje em dia, e muito devido ao desenvolvimento da tecnologia, máquinas mais sofisticadas estão disponíveis no mercado, o que permite diminuir os tempos de *setup* bem como os tempos de cada operação. Este desenvolvimento veio incentivar mais as empresas a deixarem a produção em massa passando a produzir pequenas quantidades de diferentes produtos o que implica muitas vezes a alteração da sequência de operações bem como o *layout* fabril.

Posto isto, uma grande percentagem do *lead time* de um produto passa então a corresponder a tempos de espera e de transporte entre locais de trabalho.

Com a introdução dos *AGVs* pretende-se aumentar flexibilidade, algo que até à pouco tempo era bastante reduzido devido ao uso de transportadores rígidos, diminuindo com isso o tempo de transporte e os respectivos custos.

Com este trabalho pretende-se comparar, para uma aplicação industrial (revestimento de peças por pó), um sistema de transporte baseado em transportadores (ex: tapetes ou correntes de transporte) com um sistema baseado em *AGVs*, verificando o seu impacto na eficiência do sistema de produção. No final deste estudo é pretendido que um sistema baseado em *AGVs* esteja completamente dimensionado, isto é, o número de *AGVs*, a sua velocidade e respectivo número de peças que transportam em simultâneo esteja definido. Ao mesmo tempo serão avaliadas as potencialidades de alguns simuladores disponíveis para o desenho deste tipo de sistemas.

Por último será utilizado um simulador, no qual será desenvolvida uma aplicação virtual de um sistema de produção, no âmbito de um concurso que o núcleo da robótica da Faculdade de Engenharia pretende realizar. Aqui será dimensionado o *layout* fabril, as estações de trabalho, bem como a construção e controlo dos veículos responsáveis pelo transporte das peças no chão de fábrica.

O estado actual do mercado mundial, e a procura cada vez mais por parte das empresas de soluções vantajosas mas ao mesmo tempo de baixo custo, torna o desenvolvimento deste trabalho motivante, uma vez que o que se pretende será desenvolver uma solução competitiva recorrendo ao menor número de recursos possíveis.

Abstract

Automated Guided Vehicles (AGVs) are self-driven vehicles used to transport material between workstations in the shop floor without the help of an operator, although they can also be applied in security and exploration. They are widely used in material handling systems and flexible manufacturing systems, where production orders are constantly changing.

Today, and due to the constant development of technology, sophisticated machinery is increasingly available, thus enabling manufacturing firms to achieve significant process and setup time reductions. With this development, enterprises are encouraged to leave mass production approaches and start adopting small productions lot sizes, leading to constants changes in the production operation's sequences as well as changes in the factory layout. As a consequence of the development of technology, products started to spend a big percentage of time in the queue line or being transported from one workstation/storage to another.

With the introduction of AGVs production process flexibility may increase, which, in many productions processes, is still below the expectations due to the used transportation system (ex: conveyors). At the same time, with the AGVs it is possible, to decrease transportations times and costs.

In this work, we will study by means of simulation, the impact of the use of an AGV transportation based system in an industrial coating application. The AGV will be responsible for transporting the parts from the system's entrance to the workstations. With this, flexibility in the production process will increase, which will be reflected in system's productivity. At the end of this study is intended that a AGVs based system is fully dimensioned, i.e., the number of AGVs, its speed and the number of pieces that they carry simultaneously are defined. At the same time will be assessed the potential of some simulators for the design of such systems.

Per last, and using a framework, it will be developed a virtual production process, in a contest scope organized by the robotic core of Faculdade de Engenharia Universidade do Porto. Here the factory layout dimensions, the workstations as well as the constructions and control of the vehicles responsible of transporting parts in the shopfloor will be defined.

The actual situation of the world economy, and the looking for new solutions by companies, which introduce gains but at the same time with limited budget, makes this work interesting, since we want to develop a competitive solution using the fewest possible resources.

Agradecimentos

Ao Prof. Dr . António Paulo Moreira e Prof. Dr. Américo Azevedo pelo apoio e suporte na realização desta dissertação, como também na realização dos dois artigos que tiveram origem no presente trabalho.

Gostaria de agradecer aos meus colegas e amigos Ana Cristina, Andry Maykol, David Lima, João Ferreira e José Braga pela amizade, conselhos e ajuda prestados ao longo de todo o percurso que culmina nesta dissertação.

Agradeço, de forma muito especial à Inês Isabel e seus familiares pelo acreditar, pela força e apoio demonstrado.

Agradeço, de forma muito especial aos meus pais pela oportunidade, pela dedicação, pela compreensão e incentivo em todos os momentos.

Um Muito Obrigado a todos

Luís F. Rocha

Conteúdo

1	Introdução	1
1.1	Enquadramento Histórico	1
1.2	Objectivos	3
1.3	Organização do Documento	3
2	Estado da Arte	5
2.1	Estado actual da indústria	5
2.2	Soluções Tecnológicas <i>AGVs</i>	6
2.2.1	Topologias de <i>AGVs</i> disponíveis no mercado	6
2.2.2	Gestão das baterias dos <i>AGVs</i> - Soluções no mercado	7
2.2.3	Segurança	9
2.2.4	Rede de trajectórias dos <i>AGVs</i>	11
2.3	Ferramentas de simulação	13
2.3.1	Breves Definições	14
2.3.2	Tipos de modelos	14
2.3.3	Porque recorrer à simulação?	15
2.3.4	Vantagens e Desvantagens da simulação	16
2.3.5	Simulação de sistemas baseados em <i>AGVs</i> - Simuladores no Mercado	17
3	Simulador ARENA	21
3.1	História Software ARENA	21
3.2	ARENA - Introdução	22
3.3	O VBA no ARENA	24
3.4	Exemplo desenvolvido para familiarização com o software Arena	25
4	Caso de Estudo	29
4.1	Descrição da Aplicação Industrial em Estudo	29
4.2	Layout Fabril	30
4.3	Soluções Propostas	32
4.3.1	Primeira solução	32
4.3.2	Segunda solução Proposta	32
4.4	Abordagem por Simulação	33
4.4.1	Sistema de Produção Actual	33
4.4.2	Primeira Solução Proposta - Duplicação do Sistema de Produção Actual	34
4.4.3	Segunda Solução Proposta - Sistema de transporte baseado em <i>AGVs</i>	34
4.5	Resultados e sua Análise	37
4.5.1	Resultados - Produção de 150 peças	37
4.5.2	Resultados - Produção de n em n minutos	46

4.6	Conclusão final	50
5	Robot Factory	53
5.1	Apresentação do ambiente industrial desenvolvido	53
5.2	Ambiente de desenvolvimento SimTwo	54
5.2.1	Descrição geral	54
5.2.2	Construção do ambiente de simulação	55
5.3	Métodos de localização dos AGVs no referencial mundo	59
5.3.1	Método de localização relativa - Odometria	60
5.3.2	Calibração odometria	63
5.3.3	Modelo estocástico do erro da odometria	63
5.3.4	Método de Localização Absoluta	66
5.3.5	Fusão Sensorial	67
5.3.6	Resultados obtidos - Métodos de Localização	69
5.4	Aplicação externa de controlo	71
5.4.1	Protocolo de comunicação UDP	73
5.4.2	Interacção entre Aplicações	74
5.5	Funções básicas de movimento implementadas	75
5.5.1	GoToXY	76
5.5.2	Follow Line	76
5.5.3	Follow Circle	77
5.5.4	Stop	78
5.6	Planeamento de trajetórias - Ferramentas de Apoio	78
5.6.1	Algoritmo BFS	79
5.6.2	Algoritmo de Pesquisa A*	80
5.6.3	Comparação entre o algoritmo BFS e A*	82
5.7	Robô em desenvolvimento	83
5.7.1	Motores e <i>Encoder</i>	83
5.7.2	Driver MD25	84
5.7.3	Rodas	85
5.7.4	Alimentação	85
6	Conclusões e Trabalho Futuro	87
6.1	Conclusões	87
6.2	Trabalho Futuro	88
A	Esquemático Layout Fabril Robot Factory	89
	Referências	91

Lista de Figuras

1.1	Porto Marítimo de Roterdão [1]	2
2.1	Substituição manual de baterias	7
2.2	Recarregamento automático de baterias	8
2.3	Substituição automática de baterias	8
2.4	Recarregamento de baterias durante a produção	8
2.5	Segurança -Estrutura AGVs	9
2.6	Trajectoria livre - AGVs (adaptado de [2])	11
2.7	Trajectoria restrita - AGVs (adaptado de [3])	11
2.8	Direcções Trajectórias AGVs	12
2.9	Tipos de modelos [4]	14
3.1	Fluxograma Exemplo ARENA	23
4.1	<i>Layout</i> Fabril Actual do Processo de Produção	30
4.2	<i>Layout</i> Fabril do Processo de Produção - Solução Proposta 1	32
4.3	<i>Layout</i> Fabril do Processo de Produção - Solução Proposta 2	33
4.4	Distâncias <i>layout</i> actual	34
4.5	Distancias <i>layout</i> sistema baseado em AGVs	35
4.6	Gráfico Comparativo soluções baseadas em transportadores - Tempos de produção de 150 peças	38
5.1	<i>Solids</i> - Exemplo SimTwo	56
5.2	Ambiente industrial modelado em SimTwo	58
5.3	AGV diferencial construído em SimTwo	58
5.4	Processo de carregamento da peça no AGV - Entrada do sistema	59
5.5	Processo de descarga da peça na estação de trabalho.	59
5.6	Região de incerteza quando utilizada para a estimativa da posição a adometria	61
5.7	Robô Diferencial	62
5.8	Esquema representativo da medição da distância do robô a uma parede lateral	66
5.9	Percurso definido para o AGV	69
5.10	Trajectoria do robô - localização recorrendo apenas ao sistema de odometria	70
5.11	Trajectoria do robô recorrendo apenas ao sistema de odometria Posição real	70
5.12	Trajectoria do robô - Sistema de odometria mais sensores infravermelhos	71
5.13	Trajectoria do robô com a utilização do sistema odometria mais sensores infravermelhos- Posição real	71
5.14	Aplicação de controlo remoto	72
5.15	Arquitectura do sistema desenvolvido	72
5.16	Aplicação local	73

5.17	Cabeçalho da mensagem Protocolo UDP	74
5.18	Troca de dados entre a aplicação de controlo remoto e SimTwo	75
5.19	Troca de dados entre a aplicação Controlo Remoto e Controlo Local	75
5.20	Máquina de estados da rotina GoToXY	76
5.21	Máquina de estados da rotina Follow Line	77
5.22	Máquina de estado Follow Circle	78
5.23	Algoritmo <i>Breadth-First Search</i> (BFS) - Exemplo	79
5.24	Grafo Layout SimTwo	80
5.25	Exemplo - Algoritmo BFS implementado	81
5.26	Exemplo - Algoritmo A*	82
5.27	Robô desenvolvido	83
5.28	Sistema: Motor,Driver e rodas	83
5.29	Sistema EMG30 - Motor, caixa redutor e <i>encoder</i>	84
5.30	Driver Motor DC - Devantech MD25 Dual 12V 2.8A H-Bridge	84
5.31	Roda do robô	85
5.32	Alimentação robô	85

Lista de Tabelas

3.1	Resumo operações principais	26
3.2	Distâncias entre postos de trabalho consideradas - Exemplo	26
3.3	Sequência de operações para cada tipo de peça - Exemplo	27
4.1	Sequência de Operações	30
4.2	Tempos Mínimos no Processo de Produção	30
4.3	Velocidade do transportador para o Processo de Cozedura	31
4.4	Velocidade do transportador para o Processo de Pintura	31
4.5	Distâncias entre postos de trabalho consideradas	36
4.6	Tempos de Produção das 150 peças - Soluções baseada exclusivamente em transportadores	37
4.7	Resultados Tempos de Produção de 150 peças - Solução AGVs 1m/s - Prioridades de transporte iguais	40
4.8	Resultados Tempos de Produção de 150 peças - Solução AGVs 1m/s - Prioridades de transporte diferentes	40
4.9	Resultados Tempos de Produção de 150 peças - Solução AGVs 2m/s - Prioridades de transporte iguais	41
4.10	Resultados Tempos de Produção de 150 peças - Solução AGVs 2m/s - Prioridades de transporte diferentes	41
4.11	Resultados Tempos de Produção de 150 peças - Solução AGVs 3m/s - Prioridades de transporte iguais	42
4.12	Resultados Tempos de Produção de 150 peças - Solução AGVs 3m/s - Prioridades de transporte diferentes	42
4.13	Resultados Tempos de Produção de 150 peças - Solução AGVs 1m/s e transporte de 2 peças em simultâneo- Prioridades de transporte iguais	43
4.14	Resultados Tempos de Produção de 150 peças - Solução AGVs 1m/s e transporte de 2 peças em simultâneo- Prioridades de transporte diferentes	44
4.15	Resultados Tempos de Produção de 150 peças - Solução AGVs 2m/s e transporte de 2 peças em simultâneo- Prioridades de transporte iguais	44
4.16	Resultados Tempos de Produção de 150 peças - Solução AGVs 2m/s - Prioridades de transporte diferentes	45
4.17	Resultados Tempos de Produção de 150 peças - Solução AGVs 3m/s e transporte de 2 peças em simultâneo - Prioridades de transporte iguais	45
4.18	Resultados Tempos de Produção de 150 peças - Solução AGVs 3m/s e transporte de 2 peças em simultâneo- Prioridades de transporte diferentes	46
4.19	Comparação entre os melhores resultados obtidos para a produção em massa de 150 peças (paradigma custo/performance)	46

4.20	Resultados Tempos de Produção para a chegada de novas ordens de fabrico de 1 em 1 minuto - Solução AGVs 1,2 e 3 m/s - Prioridades de transporte diferentes . .	47
4.21	Resultados Tempos de Produção para a chegada de novas ordens de fabrico de 3 em 3 minuto - Solução AGVs 1,2 e 3 m/s - Prioridades de transporte diferentes . .	48
4.22	Resultados Tempos de Produção para a chegada de novas ordens de fabrico de 1 em 1 minuto - Solução AGVs 1,2 e 3 m/s com transporte de 2 peças em simultâneo - Prioridades de transporte diferentes	48
4.23	Resultados Tempos de Produção para a chegada de novas ordens de fabrico de 3 em 3 minuto - Solução AGVs 1,2 e 3 m/s com transporte de 2 peças em simultâneo - Prioridades de transporte diferentes	49
4.24	Melhores resultados obtidos considerando a produção de 1 em 1 minuto	50
4.25	Melhores resultados obtidos considerando a produção de 3 em 3 minuto	50
5.1	Secções abrangidas pelo simulador SimTwo	55
5.2	Objectos básicos disponíveis no SimTwo	55
5.3	Propriedades dos objectos básicos	56
5.4	Objectos de complexidade mais elevada disponíveis no SimTwo	56
5.5	Propriedades dos objectos de complexidade mais elevada disponíveis no SimTwo	57
5.6	Valor dos Parâmetros obtidos - Calibração odometria	63
5.7	Valor dos Parâmetros obtidos - Modelo estocástico do erro da odometria	65

Abreviaturas e Símbolos

73/23/EEC	<i>Low Voltage Directive</i>
98/37/EC	<i>Machine Directive</i>
A*	<i>Astar</i>
AGV	<i>Automated Guided Vehicle</i>
BFS	<i>Breadth-First Search</i>
CAD	<i>Computer Aided Design</i>
EKF	<i>Extended Kalman Filter</i>
EMC 89/336/EEC	<i>Electromagnetic Compatibility</i>
EN 418 Standard	<i>European Standard for Safety machinery</i>
EN 954-1 Standard	<i>Safety of Machinery, Related Parts of Control Systems - Design</i>
EN 1525 Standard	<i>Safety of industrial trucks Standard</i>
FIFO	<i>First In First Out</i>
GPSS	<i>General Purpose Simulation System</i>
ODE	<i>Open Dynamics Engine</i>
SIMAN	<i>SIMmulation ANalysis language</i>
UDP	<i>User Datagram Protocol</i>
WIP	<i>Work-in-Process</i>
VBA	<i>Visual Basic for Applications</i>
WWW	<i>World Wide Web</i>

Capítulo 1

Introdução

Até há alguns anos atrás, altura em que a robótica móvel ainda não se encontrava suficientemente desenvolvida para ser considerada pelas empresas como uma opção válida, todo o transporte de materiais entre postos de trabalho e armazéns era efectuado essencialmente recorrendo a recursos humanos ou a soluções tecnológicas pouco flexíveis, como por exemplo os tapetes de transporte.

Com a introdução da robótica móvel no ambiente industrial, mais propriamente os *AGVs*, conseguiu-se reduzir o número de acidentes de trabalho, aumentar a flexibilidade do processo de fabrico (facilidade no aumento do número de *AGVs* e respectivas trajectórias bem como na adição de pontos de (des)carga) sendo estes facilmente integrados com outros sistemas de automação, que aliás é um dos grandes problemas indústrias dos nossos dias [5].

Estes sistemas robóticos por norma necessitam de um grande investimento inicial, o que com o passar dos anos demonstra ser uma mais-valia devido à sua robustez, precisão, baixo custo de manutenção e de operação [6].

Com este trabalho pretende-se desenvolver um sistema baseado em *AGVs* para uma aplicação industrial, especializada no revestimento de peças, no qual será comparada a solução actual baseada em transportadores, com uma logística baseada em *AGVs*, que se encontra em grande desenvolvimento no mercado actual.

1.1 Enquadramento Histórico

A origem dos *AGVs* data de 1953, pelas mãos da *Barrett Electronics de Northbrook, Illinois*, com o objectivo de automatizar o veículo de reboque que era utilizado por inúmeras empresas naquela época [7, 8].

Este veículo era guiado por fios eléctricos embutidos no pavimento, que criavam um campo magnético utilizado pelos sensores colocados na parte inferior do veículo para se localizar.

Este nível de tecnologia vingou no mercado até há década de 70, onde se verificou um grande avanço a nível dos controladores permitindo ao sistema evoluir em capacidade e flexibilidade. Como resultado verificou-se uma explosão nas áreas de aplicação dos AGVs.

Actualmente esta tecnologia encontra-se bastante desenvolvida, desde o planeamento de trajectórias aos sistemas de localização e segurança, mas com certeza mais avanços irão ser realizados. A solução de fios embutidos no pavimento ainda se encontra no mercado, estando no entanto em vias de se tornar obsoleta, uma vez que existem neste momento várias outras tecnologias que aumentam a flexibilidade do sistema, nomeadamente sistemas baseados em marcos, visão, correspondência de mapas entre outros.

Um segundo factor importante que muito contribui para o cada vez maior recurso a sistemas robóticos autónomos, dos quais os AGVs fazem parte, é a grande competitividade actual do mercado. No passado as empresas conseguiam facilmente vender o seu produto num mercado menos exigente. Agora as empresas são obrigadas a reduzir o *lead time* dos seus produtos bem como os seus ciclos de vida, oferecendo produtos e serviços de maior qualidade [9].

É necessário então às empresas obter soluções tecnológicas que lhes permitam ter a flexibilidade necessária para responder o mais rapidamente possível às exigências dos seus clientes e às expectativas do mercado.

Estes dois factores, a evolução na robótica móvel e a necessidade de flexibilidade, a produção de produtos/serviços de maior qualidade mas ao mesmo tempo de menor custo por parte das empresas, levou a que o recurso a AGVs seja cada vez mais utilizado no ambiente industrial de hoje.

Um exemplo, dos mais conhecidos, é o caso do Porto Marítimo de Roterdão o qual possui mais de 100 AGVs no transporte de contentores desde os navios até ao respectivo local de armazenagem (figura 1.1) [7]. Este caso é aliás ainda hoje muito estudado quer a nível de trajectórias quer a nível de métodos de controlo de alto nível [10, 11].



Figura 1.1: Porto Marítimo de Roterdão [1]

1.2 Objectivos

A presente dissertação tem como objectivo principal realizar uma comparação, para um caso de estudo em concreto, entre sistemas de transporte usualmente utilizados (ex:tapetes e correntes de transporte) e um sistema baseado em *AGVs* emergente no mercado actual. Esta comparação terá como foco principal a avaliação da flexibilidade introduzida por ambos os sistemas de transporte e o seu impacto no *throughput* do sistema de produção. Um segundo objectivo consiste na construção em ambiente de simulação de um processo de fabrico, no âmbito de um concurso que o núcleo de robótica da Faculdade de Engenharia pretende realizar. Este ambiente modelado em simulação será mais tarde construído à escala em ambiente real.

Assim para a concretização destes objectivos esta dissertação será enformada nas seguintes etapas:

- Levamento das soluções tecnológicas disponíveis no mercado que afectam o funcionamento dos *AGVs*;
- Estudo do estado actual da indústria e potenciais vantagens da introdução dos *AGVs* nesse ambiente;
- Análise do caso de estudo considerado (suas limitações, apresentação de possíveis soluções e comparação de resultados);
- Estudo de diferentes ferramentas de simulação, possíveis de serem utilizadas para o desenvolvimento da dissertação;
- Construção, em ambiente de simulação, de um sistema de produção e respectivo controlo;

1.3 Organização do Documento

Esta secção tem como objectivo apresentar a estrutura e organização do documento.

Neste contexto, e no capítulo 2, será realizada uma análise do estado da arte, focando-nos nas soluções actuais baseadas em *AGVs*, no estado actual da indústria, bem como nalguns simuladores que permitem simular sistemas baseados em *AGVs*.

No capítulo 3 será apresentado o simulador seleccionado para realizar o primeiro objectivo proposta nesta dissertação. No capítulo 4 é descrito o caso de estudo considerado, sendo analisadas as suas limitações e possíveis soluções que poderão ser tomadas para as ultrapassar.

No capítulo 5 é descrito o sistema de produção construído, bem como os diferentes algoritmos de controlo e decisão implementados.

Finalmente no capítulo 6 são apresentadas as conclusões retiradas no desenvolvimento desta dissertação, bem como algumas proposta de trabalho a serem realizados no futuro.

Capítulo 2

Estado da Arte

A eficiência e eficácia dos sistemas de produção, nos dias de hoje, estão directamente relacionados com o desempenho no transporte de materiais ao longo do chão de fábrica, sendo necessário integrar este transporte com os recursos já presentes no local. Neste domínio os veículos guiados autonomamente têm ganho algum terreno aos restantes meios já existentes à algum tempo no mercado. Neste contexto, o desempenho de um sistema automático de transporte é essencial para se conseguir um bom desempenho geral do sistema de produção.

Será então importante iniciar o trabalho pela análise do estado actual da indústria e encontrar respostas para a necessidade de flexibilidade actual nas empresas, bem como o motivo pelo qual estes veículos são cada vez mais utilizados.

2.1 Estado actual da indústria

Num altura em que as empresas, com a conjuntura do mercado que se verifica, se deparam com uma grande concorrência externa é necessário desenvolver produtos com cada vez maior qualidade, mas ao mesmo tempo com custos reduzidos e com limites temporais curtos.

A juntar a estes factores junta-se ainda a necessidade das empresas irem ao encontro das expectativas dos seus clientes, isto é, os produtos desenvolvidos são cada vez mais costumizados, provocando a diminuição do tamanho dos lotes bem como a necessidade de a empresa conseguir adaptar rapidamente todos as entidades relacionados ao processo de manufactura aos novos pedidos de produção, nomeadamente sistemas de informação e recursos utilizados no processo de fabrico, entre outros.

Para isso, um ponto-chave, é a necessidade de que a logística interna de produção esteja fortemente coordenada com os recursos de manufactura, uma vez que se encontram integrados num ambiente em que a complexidade, variabilidade e volumes dos produtos é uma realidade em constante mudança. Entenda-se logística interna de produção como um segmento da indústria

automatizada, responsável pela gestão e controlo da mão-de-obra, recursos e informação referentes ao processo de fabrico.

Hoje em dia as empresas de manufactura estão focadas na eliminação de desperdício e na eliminação de actividades que não acrescentem valor ao produto final. Esta abordagem tem como consequência alterações ao nível do chão de fábrica, destacando-se alterações no *layout* de produção bem como no transporte de materiais entre postos de trabalho, o qual irá permitir um aumento da eficiência do sistema de produção.

Assim um dos aspectos críticos num processo de produção é a forma como os materiais são transportados, bem como as operações de carga e descarga são efectuadas. Actualmente alguns possíveis meios de transporte englobam *AGVs*, tapetes de transporte e outros meios que usualmente necessitam da intervenção de um operador.

Os *AGVs* são no entanto cada vez mais os utilizados para estas operações, devido à sua robustez, flexibilidade na adaptação do fluxo de materiais tendo reflexo directo na produtividade do sistema.

Além destas vantagens, o uso de *AGVs* oferece a possibilidade de controlo do fluxo de materiais em tempo-real, permitindo tomar o conhecimento do percurso efectuado por cada tipo de peça bem como o veículo responsável por esse transporte, permitindo reduzir o *WIP* (*Work-in-Process*) e introduzindo uma melhor resposta às ordens de fabrico [12].

Estudos mais recentes têm vindo a analisar a possibilidade de juntar as vantagens do uso de tapetes de transporte com as de uso de sistemas baseados em *AGVs*, sendo no entanto uma área ainda pouco desenvolvida [13].

Na próxima secção são apresentadas algumas características dos *AGVs*, os quais tornam estes veículos cada vez mais apetecíveis no ambiente industrial.

2.2 Soluções Tecnológicas *AGVs*

Como referido anteriormente, os sistemas baseados em *AGVs* têm verificado nos últimos anos um grande desenvolvimento tecnológico. Será interessante por isso avaliar um pouco o estado actual do mercado. Este estudo centrar-se-á em alguns aspectos estruturais deste tipo de veículos bem como algoritmos de afectação de trajectórias, começando por analisar os variados tipos de *AGVs* que existem no mercado, passando pela segurança e métodos de carregamento das respectivas baterias.

2.2.1 Topologias de *AGVs* disponíveis no mercado

O primeiro *AGV* disponível no mercado tinha como função o reboque de outros materiais. Ora com o crescimento das áreas de aplicação dos *AGVs* foi necessário redesenhar estes veículos de forma a poderem desempenhar as suas funções com a máxima autonomia e flexibilidade.

Actualmente os veículos guiados autonomamente podem ser divididos em variados tipos, desde empilhadores, rebocadores, carregadores os quais diferem nas suas propriedades mecânicas,

cada um especializado numa determinada área de aplicação [7, 14]. Estas diferentes tipologias têm influência directa na eficiência com que executam o seu trabalho, sendo essencial uma escolha adequada para cada problema. Grande parte dos produtores deste veículos permitem soluções customizadas, que vão de encontro às necessidades de cada cliente.

Abordados os diferentes tipos de veículos disponíveis, serão introduzidos os métodos de carregamento das baterias destes veículos. A escolha do método para a execução desta operação é crítico, uma vez que se pretende que esta operação condicione o menos possível o processo de produção.

2.2.2 Gestão das baterias dos AGVs - Soluções no mercado

Assim e quando um empresa pretende instalar uma sistema baseado em *AGVs*, depara-se com o problema de ter de reservar um local que apenas será utilizado para o carregamento das respectivas baterias, indispensáveis para que estes possam desempenhar as funções para as quais foram desenhados. A escolha de cada uma das soluções depende de problema para problema não havendo, à partida uma solução tipo. É necessário avaliar cada sistema de produção e verificar qual a solução que mais se adequa.

Os fabricante deste tipo de sistema têm assim um leque de opções para oferecer aos seus clientes, dos quais se destacam:

Substituição Manual: Operação na qual um operador com formação tem de estar presente de forma a retirar a bateria descarregada do *AGV* (figura 2.1), colocando no seu lugar uma bateria completamente recarregada [15].



Figura 2.1: Substituição manual de baterias

Carregamento Automático ou por Oportunidade: Esta operação consiste na deslocação automática do *AGV* para um local onde as baterias são carregadas de forma automática, onde permanecerá alguns minutos (figura 2.2). Só ocorre em situações em que o veículo não esteja a ser solicitado no sistema [15] ou no caso de o desgaste da respectiva bateria atingir um nível previamente definido [15, 16, 17, 18]. Neste último caso o veículo termina o trabalho que lhe foi atribuído dirigindo-se de seguido para o local de recarregamento.

Substituição Automática: Neste caso é necessária a aquisição de um equipamento extra adequado para esta operação. Nesta solução o veículo quando detecta que o seu nível de bateria

está abaixo de um determinado valor desloca-se para uma determinada área onde uma máquina automatizada substitui a bateria por uma completamente carregada (figura 2.3). Esta máquina terá ainda a função de colocar a bateria desgastada num local de carregamento, garantindo que apenas baterias completamente carregadas são colocadas nos AGVs [15, 19].



Figura 2.2: Recarregamento automático de baterias



Figura 2.3: Substituição automática de baterias

Carregamento de baterias durante o processo de produção: Neste caso a operação de recarregamento de baterias é executado com o AGV a realizar tarefas no processo de produção (figura 2.4). São criados vários pontos de recarga, lateralmente ou no pavimento, ao longo do percurso do AGV permitindo assim que nas operações de (des)carga e durante o transporte as baterias possam ser recarregadas sem necessitar de deslocar o veículo para locais fora do processo de produção [20, 21].



Figura 2.4: Recarregamento de baterias durante a produção

Para realizar a escolha da solução mais adequada para o sistema de produção é necessário ter em conta alguns aspectos importantes. Nomeadamente o tempo em que os veículos estão a aguardar alocação de uma nova tarefa, para o caso de se recorrer a um sistema automático de carregamento de baterias. Neste caso alguns fornecedores aconselham que o tempo estimado para o carregamento num turno de trabalho seja igual ao tempo que o veículo está em operação durante o mesmo [21]. No entanto este tempo estará sempre dependente do tipo de bateria utilizado (chumbo, Ni-MH, Lítio, etc).

Para um *AGV* se deslocar ao nível do chão de fábrica não basta apenas garantir o seu funcionamento ininterrupto. É necessário integrar este veículo com o ambiente industrial e todas as adversidades que este coloca ao seu funcionamento. A temática da segurança de funcionamento destes veículos irá ser abordada no capítulo que se sucede.

2.2.3 Segurança

Os *AGVs* são veículos desenhados com a função principal de transporte de materiais, por vezes frágeis, entre postos de trabalho num ambiente que possui diversos obstáculos, alguns deles móveis como é o caso de operadores.

Assim sendo, uma das principais preocupações entre todos os fabricantes de *AGVs* prende-se com a segurança. É necessário “blindar” o veículo com sistemas adequados que permitam o normal desempenho das suas funções sem por em causa a integridade do ambiente que o rodeia bem como o material que transportam.

Após algum estudo do estado da arte pôde-se auferir que as soluções ao nível de segurança não variam muito de fabricante para fabricante.

Uma das soluções tipo encontrada (ver figura 2.5), refere-se ao fabricante sueco *AGVElectronics* [22], que propõe a seguinte solução:

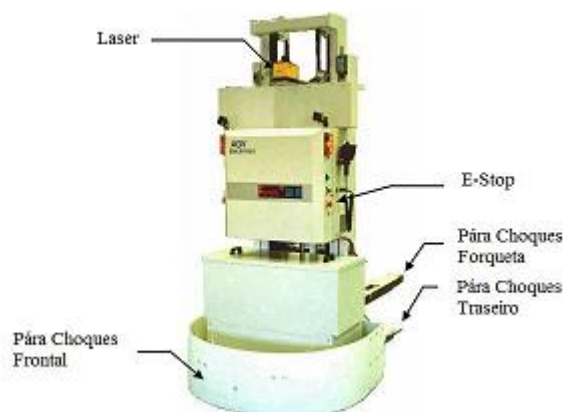


Figura 2.5: Segurança -Estrutura AGVs

Pára-choques dianteiro - Peça de protecção frontal e lateral que protege o corpo do *AGV* de impactos indesejáveis. Esta protecção tem de respeitar a norma europeia EN954-1 categoria 3.

Pára-choques traseiro - Protege o veículo durante os seus movimentos de marcha atrás. Pode conter sensores ópticos que protege o veículo não só lateralmente mas também a sua zona de carga.

Pára-choques forqueta - Localizados em cada extremidade da forqueta, são usados para prevenir impactos com as cargas ou com objectos estacionários se as cargas ou a altura da forqueta não estiver perfeitamente alinhada. É uma protecção opcional uma vez que não é necessária nem exigida por qualquer norma existente.

Scanner a laser - Usado para detectar objectos à frente do robot permitindo a este reduzir a sua velocidade ou até mesmo parar atempadamente.

E-stop - Botão de emergência colocado em cada lado da cabine de controlo do *AGV*. Este botão poderá variar tanto em número como em localização dependendo da estrutura do veículo. O botão de emergência terá de estar em conformidade com a norma EN418 categoria 0 e tem de ser construído segundo a norma EN954-1 categoria 3.

Luzes e sons de aviso - As luzes de aviso estão montadas directamente no *AGV*, podendo variar a sua frequência de comutação indicando diferentes modos de operação. Os sons de aviso são accionados quando o veículo está em movimento, podendo ser programado para outras situações.

Sensores para evitar colisões - Sensores anti-colisão, infravermelhos, ultra-sons podem ser adicionados ao *AGV* permitindo diminuir a sua velocidade para o caso de obstáculos serem detectados suavizando assim a paragem do veículo.

Chave de segurança - Chave usada para ligar o *AGV* e para alternar entre o modo automático e manual. Para activar este último modo a chave tem de estar colocada no veículo evitando assim que pessoal não autorizado manobre o *AGV* em modo manual.

Para além destes sistemas de segurança a construção do *AGV* terá de estar em conformidade com o seguinte conjunto de normas europeias:

Directiva máquinas 98/37/EC, compatibilidade electromagnética (Directiva EMC 89/336/EEC), Directiva de baixa voltagem 73/23/EEC se a tensão for acima de 75V, o que não é normal num *AGV* e finalmente uma norma do tipo C que normalmente é utilizada, EN1525-segurança em veículos industriais e seus sistemas.

2.2.4 Rede de trajetórias dos AGVs

Tendo sido referido um pouco sobre os aspectos estruturais dos veículos guiados autonomamente, serão agora apresentados temas relacionados com o seu controlo.

Ao instalar um sistema baseado em AGVs no chão de fábrica de uma empresa, um dos aspectos importantes que é necessário definir é a trajetória que o veículo terá de executar para transportar os materiais de um determinado local para a respectiva estação de trabalho. A escolha desta trajetória tem duas principais consequências no processo de produção: a primeira refere-se ao espaço necessário para a circulação destes veículos, a segunda refere-se aos tempos de atraso que o transporte irá introduzir no sistema.

Como podemos ver a selecção das trajetórias dos veículos é um processo crítico que tem influência directa na eficiência com que o sistema de produção opera.

Com os sistemas de localização disponíveis no mercado podem ser definidas dois tipos de trajetórias: livres (figura 2.6) e restritas (figura 2.7) [23].

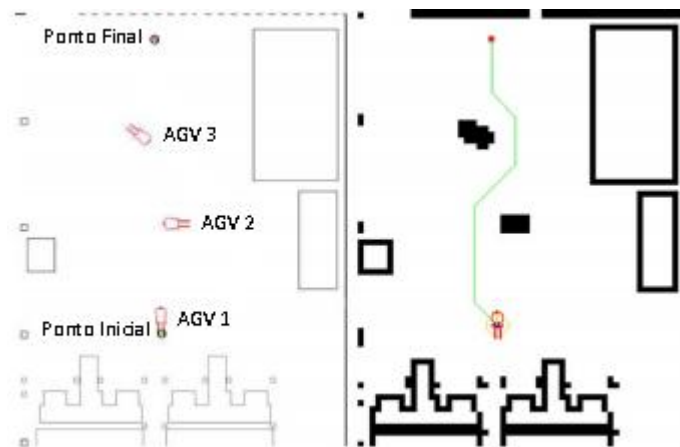


Figura 2.6: Trajetória livre - AGVs (adaptado de [2])

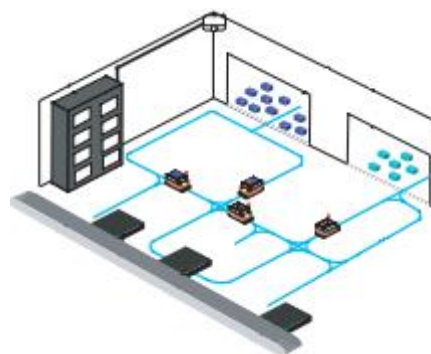


Figura 2.7: Trajetória restrita - AGVs (adaptado de [3])

Nas trajetórias livres os AGVs são guiados recorrendo a lasers ou infravermelhos que são reflectidos por “espelhos” colocados ao longo do trajecto, através de marcadores, ou por correspondência de mapas para determinar a sua posição, sendo a sua trajetória definida

recorrendo a software. Nas trajectórias restritas os *AGVs* seguem uma trajectória definida por fios embutidos no pavimento, ou por uma pista pintada no chão de fábrica.

Rapidamente pode-se auferir que a primeira opção é bastante mais flexível, uma vez que caso seja necessário alterar a respectiva trajectória é bastante mais económico recorrer a software não sendo necessário parar o sistema de produção nem efectuar grandes mudanças ao nível do chão de fabrica, para além de permitir trajectórias mais curtas entre postos de trabalho, reduzindo assim os tempos de transporte.

Quando se pretende definir as trajectórias dos veículos guiados autonomamente, é necessário ter em atenção uma outra variável, as direcções que estes podem tomar nessas trajectória. As trajectórias podem ser classificadas em 3 categorias: unidireccionais, bidireccionais e mistas. Esta definição fará maior sentido aplicar apenas em casos em que as trajectórias dos veículos são do tipo restritas, uma vez que nas livres é apresentada uma flexibilidade em tempo real elevada [2], permitindo aos veiculos movimentarem-se “livremente” numa determinada área.

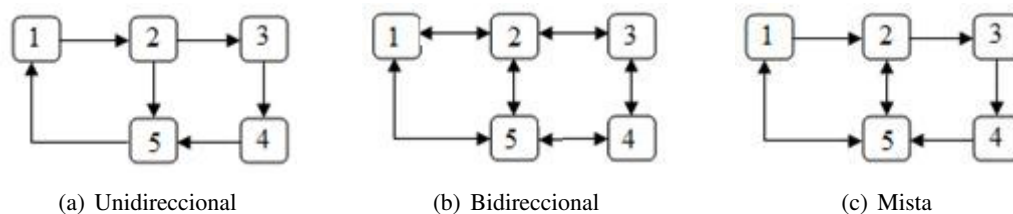


Figura 2.8: Direcções Trajectórias *AGVs*

Ora, analisando todas as possibilidades, concluí-se que em termos de controlo o modelo unidireccional torna-se mais simples, ou seja, os veículos apenas podem deslocar-se numa direcção, o que impossibilita a ocorrência de colisões durante todo trajecto entre diferentes postos de trabalho. Mas pelo contrário, esta solução torna-se bastante menos eficiente na medida em que, os veículos para executarem as tarefas que lhes foram alocadas necessitam muitas vezes de realizar percursos mais longos do que numa rede bidireccional, o que aumenta os tempos de transporte afectando a produtividade do processo de fabrico.

Como alternativa a esta abordagem, pode-se recorrer ao modelo bidireccional onde é possível desenhar as trajectórias de duas formas: ou recorrendo a duas trajectórias paralelas unidireccionais com sentidos contrários a interligar dois postos de trabalho, ideia em tudo semelhante ao que se utiliza nas auto-estradas, o que implica a existência de uma maior área alocada ao deslocamento dos *AGVs*, ou então utilizar apenas uma trajectória bidireccional em que é necessário lidar com possíveis colisões entre veículos sendo necessária a criação de *buffers* durante a sua trajectória.

Esta última solução apesar de necessitar de um controlo sobre os veículos mais complexos por comparação com o modelo unidireccional, permite reduzir o tamanho da frota de *AGVs* e aumentar a eficiência do sistema de produção [24].

Nesta área têm sido ao longo dos anos desenvolvidos alguns algoritmos de definição das trajectórias permitindo avaliar ao mesmo tempo se o modelo mais indicado para cada problema

é um trajectória unidireccional, bidireccional ou mista. Em [3, 25, 26], podem ser encontrados alguns destes algoritmos.

Após ter sido abordada a temática de definição das trajectórias que irão existir no chão de fábrica surge uma nova questão:

De que forma os veículos guiados autonomamente são alocados às trajectórias definidas?

Nomeadamente a resposta depende dos materiais que o veículo irá transportar e entre que locais irá executar esse transporte. Ora para resolver este problema de optimização duas abordagens poderão ser tomadas: recorrer a um controlo centralizado ou descentralizado.

Na abordagem por controlo centralizado toda a rede de trajectórias é decidida apenas por um “decisor”, o qual tem acesso a toda a informação do sistema. Este tipo de controlo é bastante eficaz para obter a solução óptima pois temos em consideração todo o comportamento do sistema. No entanto no caso de ocorrerem situações inesperadas tornam-se difíceis de resolver com esta arquitectura, uma vez que requer muito tempo computacional para obter o plano de trajectórias para todos os veículos. Como alternativa será interessante recorrer a um controlo descentralizado, caso em que cada veículo colecta a informação de que necessita e é responsável pela selecção da sua própria trajectória. Para isso cada veículo deverá ter em conta a trajectória dos restantes, caso contrário essa torna-se impraticável, assegurando assim uma trajectória livre de colisões e bloqueios.

Hoje em dia é utilizado o controlo centralizado para definir quais as tarefas alocadas a cada veículo tendo em conta a performance global do sistema e a arquitectura descentralizada para desenvolver uma trajectória livre de colisões e bloqueios.

Nesta área de controlo as empresas actualmente implantadas no mercado, não divulgam os seus métodos de controlo e afectação de trajectórias, pois trata-se de uma área ainda em grande desenvolvimento, em que uma nova solução poderá significar um avanço em relação à concorrência.

2.3 Ferramentas de simulação

No desenho destes tipos de sistemas, e tendo em conta todas as variáveis com que é necessário lidar num sistema de produção e respectivo sistema automático de transporte, torna-se impraticável a sua análise a não ser recorrendo a métodos de simulação.

A modelação e simulação de sistemas de produção têm vindo a crescer quase desde o nascimento dos computadores. Vários trabalhos de investigação mostram que a simulação de sistemas de produção tem-se revelado um importante passo no seu desenvolvimento e optimização. A abordagem por simulação tem também provado ser uma ferramenta de apoio à decisão amplamente aceite. Tem sido usada para testar diferentes cenários possíveis, construindo um modelo “as-is” e introduzindo vários cenários “what-if”, permitindo no final retirar conclusões sobre os diferentes cenários considerados. Esta abordagem pode ser utilizada para avaliar vários

tipos de situações, como a reconfiguração dos layouts, a introdução de novos produtos ou máquinas, ou adopção de diferentes políticas de produção.

Será então importante familiarizar o leitor para os principais conceitos de modelação e simulação, de forma a introduzir alguns assuntos que vão ser tratados ao longo deste trabalho.

2.3.1 Breves Definições

Quando se inicia um novo assunto é importante começar por clarificar os conceitos que vão ser tratados. Em particular e durante este capítulo, três conceitos vão ser constantemente referidos, modelo, sistema e simulação.

Um sistema é um conjunto de objectos que estão interligados entre si com um objectivo, sobre os quais se pretende realizar um estudo [27].

Modelo é uma representação aproximada do objecto sobre o qual se pretende concentrar as atenções.

Simulação é desenvolver um modelo representativo do sistema em fase de projecto ou já desenvolvido, com o objectivo de avaliar o seu comportamento sob diferentes condicionantes sendo possível retirar conclusões importantes que poderão condicionar o seu desenho ou aplicação.

2.3.2 Tipos de modelos

Quando se pretende modelar um processo real de produção pode-se recorrer a vários tipos de modelos. Segundo [4], existem os seguintes tipos ilustrados na figura 2.9.

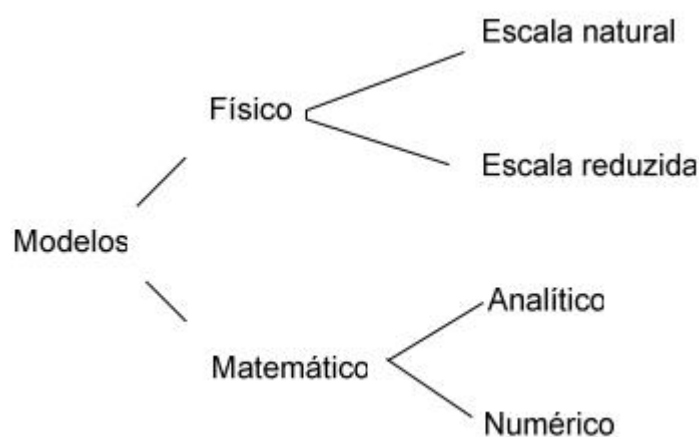


Figura 2.9: Tipos de modelos [4]

Os modelos podem então ser classificados de duas formas: modelo matemático ou modelo físico. Nos modelos matemáticos gostaria de chamar a atenção para o facto de se poder dividir

em analíticos ou numéricos. Nos analíticos o modelo é baseado em equações matemáticas, sendo utilizados quando a complexidade do sistema a modelar não é muito elevada, permitindo assim obter resultados mais precisos. Contudo quando o grau de complexidade do sistema aumenta, os métodos analíticos tornam-se inviáveis de utilizar. Nestes casos recorre-se à simulação (modelo numérico), onde a modelação é realizada com recurso a equações sem significado físico (eventualmente estatísticas) que tentam aproximar o comportamento real do sistema, uma vez que as verdadeiras equações do mesmo não são conhecidas ou demasiado complexas.

2.3.3 Porque recorrer à simulação?

As ferramentas de simulação são cada vez mais importantes, uma vez que ajudam no planeamento da produção e de processos cada vez mais complexos, assegurando que um sistema bem sucedido seja projectado no menor período de tempo.

A simulação tornou-se assim um procedimento padrão para muitas empresas quando uma alteração na produção está a ser investigada ou um sistema novo está a ser desenvolvido. Esta permite que muitas das dificuldades encontradas no início de um novo sistema sejam ultrapassadas, permitindo visualizar em curtos espaços de tempo a capacidade real do sistema para atingir os objectivos de desempenho para os quais foi desenhado.

Executando um paralelismo e pensando num simulador de voo, é possível a um piloto de reduzida experiência, desenvolver as suas capacidades sem riscos, tempo e custo associados ao seu treino quando são usados equipamentos reais. Este poderá cometer erros durante a simulação e aprender com esses mesmos erros sem sofrer as consequências das suas acções. Tanto as competências como a habilidade de tomar decisão são melhorados, resultando em segurança e eficiência ainda maiores.

Este exemplo é em tudo idêntico a uma situação real industrial.

Assim a simulação de um sistema é realizada com o intuito de melhorar a compreensão de como um sistema opera não impondo custos tão elevados como se estivessemos a operar sobre o sistema real.

Apesar destas semelhanças, a simulação de um processo difere da simulação de voo principalmente na metodologia. Num simulador de voo o piloto toma todas as decisões em tempo real, enquanto numa simulação de um processo o utilizador terá de ter as decisões previamente definidas sob a forma de regras de decisão e variáveis de operação. Além disso a simulação de processos é executada em tempo comprimido, permitindo ao utilizador executar a simulação de um sistema inúmeras vezes, executando diferentes experiências de uma forma rápida, melhorando a sua compreensão do sistema a um ritmo exponencial [28].

Contudo a simulação não é por si só uma técnica de optimização. É uma técnica de avaliação do desempenho do sistema. Pode ser vista como uma ferramenta de decisão, pois permite uma análise e avaliação das consequências da escolha de diferentes opções no sistema estudado [29].

2.3.4 Vantagens e Desvantagens da simulação

Nesta secção irão ser apresentadas as vantagens e desvantagens de adoptar uma abordagem de simulação, na análise do comportamento de um sistema real. Irão ainda ser apresentadas possíveis causas de insucesso deste tipo de abordagem [30, 31, 32].

2.3.4.1 Vantagens

- A simulação pode ser utilizada em sistemas complexos com elementos estocásticos que não conseguem ser descritos perfeitamente com os modelos matemáticos determinísticos.
- A modelação de sistemas obriga a realizar um levantamento de todos os processos e respectivas variáveis do sistema e sua interacção.
- O desenvolvimento deste modelo ajuda a levantar as variáveis controláveis e não controláveis, permitindo a análise da influência da sua variação no sistema.
- Permite a identificação de gargalos resultantes de especificações mal concebidas, tendo como consequência directa alterações no sistema real ou projecto, impedindo desta forma investimentos avultados e condenados ao fracasso.
- Possibilita um controlo temporal sobre o sistema, podendo simular em curtos intervalos de tempo grandes durações temporais reais.
- Permite a análise de sensibilidade do tipo “Que aconteceria se...”. Diferentes cenários poderão ser testados e comparados rapidamente podendo ser tomadas diferentes soluções em conformidade com o analisado.
- O resultado da simulação permite que a organização avalie os recursos actuais e a forma como estão alocados, podendo levar a possíveis alterações.
- Apresenta grandes vantagens na análise de sistemas de grande complexidade, sistema onde a análise por meios analíticos é intolerável.

2.3.4.2 Desvantagens

- A modelização de sistemas complexos pode impor elevados custos e meses de desenvolvimento, principalmente em casos em que existe dificuldade na obtenção de dados. A simulação não gera resultados fiáveis sem *inputs* apropriados.
- A execução de simulações estocástica apenas produz estimativas dos parâmetros analisados. Apesar de programas de simulação recentes já possuírem ferramentas mais desenvolvidas, o que permite a obtenção de melhores resultados, não garantem que a solução encontrada seja a solução óptima.
- A construção do modelo requer um trabalho árduo, com método e criterioso

- Os modelos desenvolvidos não são reutilizáveis para outros sistemas, o que pode prejudicar a possibilidade de ganhos de tempo consideráveis.
- Pode ser muito difícil de garantir que o modelo desenvolvido represente fidedignamente o sistema real. O que por vezes pode ser mascarado pelo facto dos dados de saída se encontrarem dentro do esperado.

2.3.4.3 Causas de insucesso no desenvolvimento da simulação

Seguidamente são apresentadas algumas causas de insucesso da simulação, em [32] poderá ser encontrada uma lista mais detalhada.

- Detalhe do sistema inadequado ao pretendido;
- Má interpretação dos resultados obtidos na simulação;
- Falha na obtenção de dados representativos do comportamento do sistema;
- Software de simulação inadequado;
- Executar uma única vez a simulação e considerar os dados obtidos como a resposta verdadeira (tempo de execução inadequado);
- Utilizar medidas de desempenho inadequadas;

Em suma a modelação de um sistema deverá ser executada por um programador que possua alguma experiência de forma a poder simplificar o máximo possível o modelo sem perder precisão em relação ao processo real.

2.3.5 Simulação de sistemas baseados em AGVs - Simuladores no Mercado

Uma fase importante para desenvolver uma simulação, corresponde à escolha do simulador. A melhor ou pior escolha do simulador, bem como a linguagem de programação utilizado pode influenciar de forma crítica o sucesso da respectiva experiência.

De seguida são apresentados alguns dos simuladores actualmente presentes no mercado, de forma sucinta, e que segundo os seus fabricantes se adequam à simulação de sistemas de transporte de materiais utilizados actualmente num processo de produção.

ShowFlow: Software desenhado para a modelação, simulação, animação e análise de processos de logística, manufactura e transporte de materiais. Este software permite obter informação em relação ao *throughput* do sistema de produção, a identificação de gargalos, *lead times* e taxas de utilização dos recursos.

Pode ser usado para suportar decisões de investimento, verificação de sistemas de manufactura em fase de projecto, testar diferentes estratégias de manufactura e a realização de testes de

performance do nosso sistema.

Whitniss: É uma ferramenta de simulação de processos dinâmicos que permite validar um sistema de produção. Permite verificar se o sistema atinge a performance esperada apoiando uma organização na melhoria contínua do seu processo de fabrico.

Estão ainda disponíveis no mercado muitos outros simuladores, tais como: Promodel, Extend, GoldSim, Mast, SimCad, os quais possibilitam a simulação de sistemas de produção.

Dentro desta área de simuladores, a Faculdade de Engenharia Universidade do Porto adquiriu licenças de utilização referentes a dois softwares: o Anylogic e o Rockwell ARENA.

O Anylogic é uma ferramenta de simulação que suporta todas as metodologias de simulação mais comuns nos nossos dias: desde sistemas dinâmico, baseado em agentes, ou orientada a eventos discretos. Esta flexibilidade única, permitida pelas diferentes linguagens de modelação, possibilita ao utilizador “absorver” a complexidade e heterogeneidade dos sistemas (negócios, económicos, sociais) para qualquer nível de detalhe pretendido. Esta ferramenta de simulação possui ainda uma interface gráfica e livrarias de objectos os quais permitem rapidamente ao utilizador modelar diferentes áreas, desde logística e manufactura, negócios, recursos humanos, comportamento de consumidores e pacientes. No software Anylogic para a definição de estruturas de dados complexas, algoritmos e conectividade externa é utilizada a linguagem de programação Java.

Este software permite ainda a análise de dados, disponibilizando ao utilizador diferentes gráficos de negócios tais como gráficos de barras e histogramas. Estes são projectados com o objectivo de permitir ao utilizador a visualização e análise eficiente da variação de dados durante a simulação.

O software ARENA foi desenvolvido pela empresa Rockwell ARENA com o objectivo de se tornar uma ferramenta de ajuda na demonstração, previsão e medição das consequências da tomada de determinadas escolhas estratégias (off-line) permitindo aumentar a eficiência e eficácia, ou seja otimizar a performance do sistema. Esta ferramenta contém recursos para modelagem de processos, design de animações, análise estatística e análise de resultados. Para isso basta recorrer a livrarias já integradas na ferramenta, tornando-a numa ferramenta bastante eficiente para a área de reengenharia, manufactura, logística.

A modelação dos sistemas é realizada construindo um modelo lógico-matemático que representa a dinâmica do sistema em estudo. Esta modelação é executada recorrendo a objectos orientados à simulação não sendo necessário ao utilizador o desenvolvimento de código de programação.

Tal como o Anylogic possui ferramentas de apoio à análise de dados, devolvendo ao utilizador um registo geral do comportamento do sistema, no final de cada modelação.

Um pequeno exemplo será implementado em ambos os simuladores (abordado no final no final do capítulo 3) o que permitirá ter um primeiro contacto com estas ferramentas bem como uma avaliação das suas potencialidades para a execução do estudo pretendido. No entanto poderá

já ser adiantado que o software seleccionado para o apoio na realização do estudo foi o software ARENA, devido à sua facilidade de uso bem como devido à existência de um maior número de referências bibliografias que apoiam na modelação dos sistemas reais, sendo esta decisão tomada também apoiada pelo exemplo desenvolvido em ambos os simuladores.

Capítulo 3

Simulador ARENA

Neste capítulo irá ser introduzida a ferramenta de simulação ARENA. Será efectuada uma pequena introdução histórica, baseada em [32], sendo de seguida apresentadas algumas das suas potencialidades, na análise e modelação de sistemas de produção.

3.1 História Software ARENA

A linguagem de simulação SIMAN teve a sua origem em 1982 pela *Systems Modelyng Corperation(EUA)*, tendo como base a linguagem GPSS utilizada até então em computadores.

Nos anos que se sucederam, esta linguagem sofreu uma grande inovação, tornando-se na primeira aplicação a fazer uso do rato na sua interface, numa época em que o sistema operativo Windows ainda não atingira o mercado. Em 1990 foi introduzida uma representação animada e a cores do funcionamento dos sistemas simulados, através do lançamento do pacote CINEMA no qual estava integrado a linguagem SIMAN. Mais uma vez a primeira interface do género disponível em simuladores.

O ARENA nasceu então no ano de 1993, tendo como base estes dois recursos anteriormente referidos. A linguagem SIMAN passou então a ter um suporte gráfico tornando-se muito mais intuitiva. Em 1995 foi lançado o ARENA para o sistema operativo Windows 95, tendo no ano seguinte recebido a certificação *Microsoft Windows Compatible*, a primeira ferramenta de simulação com esta certificação, estando integrada a linguagem VBA a qual permite comunicar com aplicativos do microsoft office.

Mais tarde, no ano 2000, a *Rockwell* adquiriu a *Systems Modelling*, a qual apostou no desenvolvimento da ferramenta ARENA tendo sido lançadas novas versões deste software com grandes melhorias. O ARENA passou assim a fazer parte da suíte *RS BIZWARE* especializada na área de planeamento e gestão de recursos no chão de fábrica, passando a representar um papel importante no segmento *Manufacturing Execution Systems* dentro do grupo *Rockwell*.

3.2 ARENA - Introdução

A concepção e implementação de fornecedores, a fabricação, o cliente e os processos internos organizacionais têm um impacto crucial na sustentabilidade de uma empresa bem como a sua habilidade para se manter competitiva, tendo sempre implementada uma filosofia de melhoria contínua. Se estas implementações forem mal planeadas podem ter efeitos desastrosos, resultando em frustração de clientes, negócios perdidos, e diminuição dos lucros.

O software ARENA possibilita a simulação de sistemas, prevendo o impacto das novas ideias de negócio, regras e estratégias antes da execução - offline, sem causar interrupções no serviço. É uma ferramenta de simulação que reúne os recursos da linguagem de programação com a facilidade de uso dos simuladores, integrado num ambiente gráfico. Pode ser usado na modelação de aplicações no contexto de sistemas de manufactura, nomeadamente design/configuração, estratégias de escalonamento e de afectação de tarefas, mas também noutros contextos como a indústria da saúde, aeroportos e militar. Contém recursos para a modelação de sistemas, análise estatística e de resultados.

É compatível com o sistema operativo windows, permitindo desenvolver modelos animados para sistemas discretos e contínuos. O ARENA permite ainda a interacção com outros aplicativos nomeadamente o *Microsoft Office*.

Esta tecnologia tem como base a linguagem SIMAN, como já foi referido, a qual está organizada em duas classes de objectos: blocos e elementos. Mais especificamente os blocos são construções lógicas que representam operações, enquanto que os elementos representam recursos, filas de espera e outros, permitindo a modelação de sistemas sem recorrer a uma única linha de código. A construção do modelo baseia-se assim na utilização de um ambiente gráfico e visual, não obrigando o programador a conhecer a linguagem de simulação SIMAN.

Para a animação dos modelos o software possibilita ao programador a implementação de animações 2D e 3D (necessário a aquisição de um pacote) em tempo real possibilitando ao mesmo tempo a importação de desenhos CAD.

O ARENA disponibiliza ao utilizador variadas bibliotecas os quais apresentam vários módulos que o apoiam na modulação de um sistema. Estes módulos estão organizados num painel com diversas categorias, podendo ser utilizados módulos de categorias diferentes na mesma modelação. Com a versão *Standard* são fornecidos três categorias genéricas - "*Basic Process*", "*Advanced Process*" e "*Advanced Transfer*". Todos juntos são designados por "*ARENA Template*" os quais são constituídos por blocos e elementos designados por "*SIMAN Template*". Este software permite ainda a criação de *templates* customizados, específicos para cada aplicação.

Em suma, a ferramenta de simulação recorre primordialmente a uma programação visual para a construção de modelos de sistemas. Os modelos são construídos sobre a forma de fluxogramas, que modelam a ocorrência de eventos a um cliente genérico, denominado entidades, o qual flui pelo sistema. A cada evento corresponde um módulo. Vários módulos ligados entre si numa determinada sequência formam o fluxograma do sistema (ver figura 3.1).

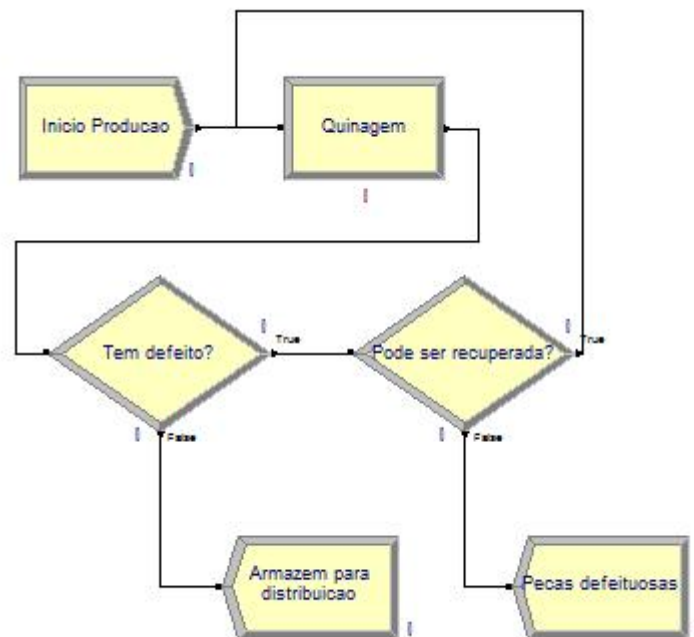


Figura 3.1: Fluxograma Exemplo ARENA

Actualmente o software ARENA é constituído por um grupo de 5 produtos: "ARENA *Basic*", "ARENA *Plus*", "ARENA *Standard*", "ARENA *Professional*", "ARENA *Enterprise*". São disponibilizadas ainda algumas opções que acompanham os produtos anteriormente referidos:

- *Packaging* ("Template");
- *Contact Center* ("Template");
- *ARENA 3D Player* ("3D Animator");
- *OptQuest for Arena* ("Optimization Tool");
- *Run Time* ("Simulation Player");

De seguida é apresentada uma breve descrição de cada uns dos produtos, onde são referidas as "templates" usadas, baseado na informação que a empresa Rockwell disponibiliza ao seus clientes [33].

- *ARENA Basic* - Mais eficaz quando se pretende modelar e analisar negócios, serviços ou fluxos de produção. Este software contém então os blocos básicos para a modelação de sistemas mais simples, em que as distâncias entre os processos, ou seja, operações que não acrescentam valor ao produto final são menosprezadas. É constituído pelo *template* "Basic Process".
- *ARENA Plus* - Permite melhorar o modelo, na medida que faculta a animação de pessoas, objectos e/ou manuseamento de materiais, equipamentos como esteiras, transportadores, ou

AGVs. Permite a animação de entidades em desenhos provenientes do Autocad, mapas ou imagens. É constituído pelo template “Basic Process” e “Advanced Transfer”.

- *ARENA Standard*- Arena Standard Edition é projectado para uso em toda a empresa, apoiando ambas as análises em profundidade de uma determinada área (por exemplo, manufactura, logística, atendimento ao cliente, cuidados de saúde) e análise de processos que abrangem diversas áreas funcionais. Inclui os *templates* “Basic Process”, “Advanced Transfer” e “Advanced Process”.
- *ARENA Professional* - Este software é mais eficaz na análise complexa de médios e grandes projectos que envolvam alterações críticas no fornecimento, manufactura, processos, logística, distribuição, armazenamento, sistemas e serviços. Permite ainda a customização de templates, que são nada mais nada menos que bibliotecas de objectos personalizados que tornam muito mais fácil e mais rápido o desenvolvimento de modelos que requerem lógica de repetição. Inclui os *templates* “Basic Process”, “Advanced Transfer”, “Advanced Process”, “Flow Process”, “Blocks” e “Elements” e as ferramentas “OptQuet”.
- *ARENA Enterprise* - O *Arena Enterprise Suite* oferece um pacote de produtos convenientes para a organização com uma vasta gama de problemas de modelagem. Inclui todas as características de fábrica *Arena Analyzer*, além de *Contact Center Arena* e *Arena 3DPlayer*.

3.3 O VBA no ARENA

O ARENA incorpora duas tecnologias que são utilizadas para melhorar aplicações desktop: *Active X Automation* que oferece facilidades de controlo para aplicativos através de uma plataforma estruturada que é acessível através de uma linguagem de programação e o *Visual Basic for Applications* ou VBA que é uma linguagem de programação, disponível no ambiente ARENA, que foi desenvolvida pela *Microsoft Visual Basic* e utilizado em todas as aplicações Microsoft Office e outras aplicações como Visio, AutoCad e Msword, que são suportados pelo *Active X Automation*, possibilitando assim a integração destes software durante uma aplicação. É utilizado no ARENA como complemento às funcionalidades visuais que disponibiliza, permitindo o desenvolvimento de algoritmos mais eficientes do que a programação visual, mas mais difíceis de implementar.

O ambiente de programação VBA pode ser acedido no software Arena a partir do menu ferramentas seleccionando a opção Macro e de seguida a opção mostrar Visual Basic Editor. Esta acção abre uma janela separada que hospeda o código VBA, formulários, interface de debug, e acesso à ajuda.

Aqui, o utilizador pode gravar diversos tipos de código VBA:

1. Código que pode ser processado como resposta à ocorrência um evento no sistema;

2. Código desenvolvido em blocos VBA, disponibilizados no *template* “*Blocks*”.O código aqui descrito é processado quando uma entidade entra no bloco. Após a sua execução, a entidade percorre o resto do modelo de forma usual.
3. Finalmente, o utilizador pode escrever código que responde à ocorrência de eventos externos, que podem ocorrer sobre a forma de mensagens em tempo-real , que chegam ao simulador ARENA através de uma rede de comunicações.

Como o código VBA é independente do ARENA, o utilizador poderá assim definir variáveis que não estejam a ser utilizadas no ARENA, permitindo o desenvolvimento de controlos com maior complexidade.

3.4 Exemplo desenvolvido para familiarização com o software Arena

Com o objectivo de fazer uma introdução ao software, avaliando ao mesmo tempo as suas capacidades para a simulação de sistemas de transporte baseados em *AGVs*, foi desenvolvido uma pequena aplicação real industrial , exemplo baseado em [34]. Num sistema de manufactura o transporte de materiais pode ser empregue de dois modos: para transporte de cargas discretas (por exemplo, camiões, empilhadoras, *AGVs*), e transportadores para transportar um fluxo contínuo de material ou produto (por exemplo, tapetes de transporte). No ARENA a modelação destes dois tipos de transportares é análoga. Consequentemente no caso exemplo desenvolvido a atenção será virada para o transporte de materiais entre postos de trabalho, onde o sequenciamento de operações é um ponto chave. Neste exemplo será considerado um sistema de manufactura do tipo *job shop*¹ , em que são produzidas três tipos de peças diferentes, G1, G2 e G3, em que o layout fabril é composto pelos seguintes locais:

- Um cais de chegada de novas ordens de fabrico;
- Uma estação de fresagem com quatro fresadoras;
- Uma estação de perfuração com três máquinas de perfuração;
- Uma oficina de pintura com duas cabines de pintura;
- Uma área de polimento com apenas um trabalhador;
- Ponto de saída do sistema

A tabela que se segue é resumido toda a informação do sistema de manufactura até agora adquirido.

¹O termo *job shop* é utilizado para designar o tipo de processo onde variados produtos, customizados pelos clientes são produzidos em pequenas quantidades. Este produtos normalmente possuem diferentes sequências de operações e de tempos de processamento. Um dos grandes problemas deste tipo de manufactura é que apesar de se conhecer os tempos de processamento de cada uma das peças constituintes do produto final, bem como a sequência de operações, é bastante complicado prever qual o estado do ocupação do shop floor em geral, traduzindo-se assim num grande problema de planeamento e controlo das operações.

Tabela 3.1: Resumo operações principais

Operação (ID)	Posto de Trabalho (ID)	Capacidade (em peças)
Fresagem	1	4
Perfuração	2	3
Pintura	3	2
Polimento	4	1

Para o transporte de peças entre postos de trabalho irá ser utilizado um sistema baseada em *AGVs*, constituído por 2 veículos, sendo então necessário definir distâncias entre postos de trabalho e a respectiva velocidade com que os veículos se deslocam. Na tabela 3.2, poderá ser visualizado as distâncias entre postos de trabalho consideradas.

Tabela 3.2: Distâncias entre postos de trabalho consideradas - Exemplo

De	Para (ID)	Distância (m)
Cais de chegada	Estação de fresagem	100
Cais de chegada	Estação de perfuração	100
Estação de fresagem	Estação de perfuração	300
Estação de fresagem	Oficina de pintura	400
Estação de fresagem	Área de polimento	150
Oficina de pintura	Área de polimento	300
Estação de perfuração	Oficina de pintura	150
Estação de perfuração	Área de polimento	400
Oficina de pintura	Cais de chegada	250
Área de polimento	Cais de chegada	250
Área de polimento	Ponto de saída	200
Ponto de saída	Cais de chegada	550
Ponto de saída	Estação de perfuração	500
Ponto de saída	Estação de fresagem	300
Ponto de saída	Oficina de pintura	400
Ponto de saída	Área de polimento	200

A velocidade de deslocamento dos *AGVs* considerada foi de 1 m/s, tendo capacidade para transportar uma única peça de cada vez. As ordens de fabrico chegam em lotes de 10 unidades, e os respectivos tempos de chegada seguem uma distribuição uniforme entre os 400 e os 600 minutos. Os lotes são constituídos por um único tipo de peça, sendo a probabilidade de cada tipo: 50 por cento são do tipo G1, 30 por cento são do tipo G2, e 20 por cento são do tipo G3.

É considerado que cada tipo de peça possui uma sequência de operações diferente, apresentadas na tabela 3.3

Neste contexto, aquando da chegada de uma nova peça à entrada do sistema (doca de chegada), as peças aguardam a chegada de um *AGV* que consoante a sequência de operações do tipo de peça que transporta a irá encaminhar para a estação correcta. À saída de cada processo existe um *buffer* onde as peças aguardam a chegada de um *AGV*, de forma a não criar atrasos nos processo de maquinagem. Quando uma peça é transportada para o processo seguinte ela é colocada num

Tabela 3.3: Sequência de operações para cada tipo de peça - Exemplo

Tipo de Peça	Sequência de Operações	Tempo de maquinagem (minutos)
G1	Fresagem	35
	Perfuração	20
	Pintura	55
	Polimento	15
G2	Fresagem	25
	Pintura	35
	Polimento	15
G3	Perfuração	18
	Pintura	35
	Polimento	15

buffer de entrada do tipo FIFO (first in first out). Finalmente quando a operação de polimento é terminada as peças são transportadas para a saída do sistema.

Tal como já foi referido, o exemplo aqui apresentado, foi implementado nos simuladores ARENA e Anylogic, servindo também como suporte de decisão para a escolha do simulador a utilizar para o caso de estudo aqui apresentado.

A nível de tratamento e análise de dados ambos os simuladores apresentam ferramentas de suporte (diferentes tipos de gráficos e análise estatística), sendo permitido em ambos retirar conclusões acerca da taxa de utilização do veículos, taxa de produtividade e tempos de processamento médio de cada peça. Para a modelação das operações bem como para a modelação dos tapetes de transporte ambos os sistemas são bastante idênticos. Apesar desta grande similaridade, o ARENA para a modelação de veículos responsáveis pelo transporte de materiais entre postos de trabalho demonstra ser mais eficiente uma vez que possui já na sua livraria modelos desenvolvidos com este objectivo, ao passo que no software Anylogic estes blocos teriam de ser programados. Além desta vantagem o ARENA permite ainda a definição de distâncias entre postos de trabalho, o que pelo contrário no software Anylogic estas distâncias não são possíveis de modelar, sendo necessário lidar com os tempos de transporte, implementados através da ligação a uma base de dados externa (por exemplo desenvolvida em excel), sendo esta ligação programada em Java. Finalmente um terceiro ponto favorável ao software ARENA é o desenvolvimento de animações 2D (ambos os simuladores permitem animações 3D sendo no entanto necessário a aquisição de um pacote extra, o qual a Faculdade de Engenharia não possui), utilizado de certa forma como validação do modelo desenvolvido. Aqui o software ARENA mostrou-se mais forte permitindo desenvolver um animação mais realista e bastante rica possuindo alguns objectos em base de dados.

Capítulo 4

Caso de Estudo

Neste capítulo o sistema de produção real, alvo do estudo deste trabalho, será apresentado. Este é caracterizado por recorrer à utilização de um sistema de transporte rígido (transportadores), facto que introduz algumas restrições no processo de produção. Neste contexto, a solução baseada em transportadores será comparada com uma solução baseada em *AGVs*, sendo analisado o seu impacto no *troughtput* do sistema de produção.

Na secção 4.1 o sistema de produção considerado será apresentado. Na secção 4.2 o *layout* actualmente utilizado e respectivas limitações serão apresentados. Na secção 4.3 serão apresentadas algumas soluções propostas, de forma a ultrapassar as limitações do sistema actual. Na secção 4.4 parâmetros importantes de cada uma das soluções são definidos. Na secção 4.5 os resultados obtidos a partir de simulação serão apresentados e discutidos.

Finalmente na secção 4.6 serão apresentadas algumas conclusões.

4.1 Descrição da Aplicação Industrial em Estudo

O caso apresentado neste trabalho é baseado numa aplicação real industrial a qual consiste na aplicação de revestimentos de fluorcarbonetos e outros polímeros nobres sobre a superfície de peças construídas em metal, cerâmica, vidro e outros materiais naturais e sintéticos, conhecido como *Powder coating*. O revestimento por pó é um processo onde partículas finas de pigmentos e resina são pulverizados para o produto a ser revestido. Este revestimento transmite à superfície das peças as características dos polímeros que a constituem, nomeadamente: anti-aderência, resistência térmica, auto-lubrificação, baixo atrito, resistência ao uso e resistência química.

Para se obter o produto final, as peças terão de passar por duas operações principais: pintura, onde é necessária a presença de um operador, uma vez que a pulverização das peças é realizada de forma manual, e um processo de cozedura. Neste caso de estudo, e conectando estas duas operações, existe um transportador responsável por transportar as peças durante todo o processo produtivo (de notar que as peças nunca deixam o transportador, mesmo quando as duas operações

principais estão a ser executadas). As peças são introduzidas no transportador recorrendo a mão de obra humana, facto que adiciona tempos de atraso no sistema que serão considerados mais tarde durante a simulação.

4.2 Layout Fabril

Tal como referido na secção anterior, este processo de revestimento é constituído por duas operações principais, estando estas interligadas por um transportador (figura 4.1).

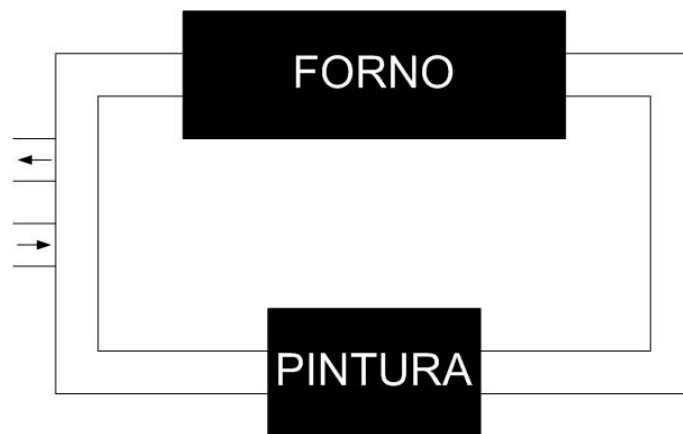


Figura 4.1: *Layout* Fabril Actual do Processo de Produção

Da perspectiva do sistema de produção implementado, será considerada a produção de três tipos de peças, que diferem entre elas na sequência de operações e nos tempos de processamento (tabela 4.1 e 4.2). Tendo então em conta que as peças nunca deixam o transportador, a única forma de variar os tempos de processamento de cada tipo de peça é variando a velocidade do transportador.

Tabela 4.1: Sequência de Operações

Tipo Peça	Sequência Operações
1	P-C-P-C-P-C-E
2	P-C-P-C-E
3	P-C-E

P-Processo de Pintura
C-Processo de Cozedura
E-Exit

Tabela 4.2: Tempos Mínimos no Processo de Produção

Tipo Peça	Tempo Pintura(segundos)	Tempo Cozedura(segundos)
1	27	120
2	27	86
3	27	86

Assim e sabendo que o forno tem um comprimento de 6 metros e a pintura um comprimento de 4 metros a velocidade dos transportadores poderá ser obtida (tabela 4.3 e 4.4).

Tabela 4.3: Velocidade do transportador para o Processo de Cozedura

	Tipo Peça 1	Tipo Peça 2	Tipo Peça 3
Velocidade do transportador (m/s)	0,05	0,07	0,07

Tabela 4.4: Velocidade do transportador para o Processo de Pintura

	Tipo Peça 1	Tipo Peça 2	Tipo Peça 3
Velocidade do transportador (m/s)	0,15	0,15	0,15

No estudo aqui apresentado, será interessante analisar um cenário de produção em que a produção dos diferentes tipos de peças são considerados (flexibilidade do processo). Assim sendo, pode ser considerado que num determinado instante existem diferentes tipos de peças no transportador para serem produzidos. Neste contexto, não será então possível variar a velocidade do transportador tendo em conta o tipo de peça, sob pena de ser posto em risco os tempos de processamento mínimo das diferentes operações.

Um outro caso consiste em considerar que apenas é colocado um tipo de peça em produção. Assim poderá ocorrer a situação de existir peças simultaneamente na operação de cozedura e pintura. Mais uma vez a velocidade do transportador não pode ser definida para a operação pintura, uma vez que essa alteração iria causar que o tempo mínimo da operação cozedura não seria respeitado.

Com este *layout* e para resolver o problema, poderão ser tomadas duas abordagens distintas. A primeira consiste em considerar que apenas é possível produzir um tipo de peça de cada vez, alterando a velocidade de acordo com o tipo de peça a ser produzido. Mas tendo sempre em mente que apenas se poderá alternar entre velocidades correspondentes aos tempos de cozedura, uma vez que garantindo que o tempo de cozedura é cumprido o tempo de pintura também o será. Esta é actualmente a solução utilizada.

Contudo esta solução apresenta a desvantagem de que é necessário esperar que toda a sequência de operações de um determinado tipo de peça seja concluído para introduzir um novo tipo no sistema de produção, impondo assim grandes atrasos na produção.

A segunda abordagem consiste em manter a velocidade do transportador constante e igual à velocidade que garante que o tempo de todas as operações são cumpridas. Esta solução apresenta também um problema, uma vez que o tipo de peça 1 é aquele que necessita de mais tempo de operação na cozedura, o *lead time* dos tipos 2 e 3 vai ser incrementado para além daquilo que seria o necessário.

Como se pode verificar a flexibilidade do *layout* fabril fica aquém daquilo que é necessário, tendo consequências directas na produtividade do sistema de produção.

Neste contexto, o principal objectivo deste estudo será aumentar a produtividade e flexibilidade do sistema tendo em conta as restrições do processo de produção.

Duas possíveis soluções para atingir este objectivo serão descritas na próxima secção.

4.3 Soluções Propostas

4.3.1 Primeira solução

A primeira solução que foi alvo de estudo, consiste na duplicação do sistema de produção actual (figura 4.2). Duplicando o sistema duplicamos a sua capacidade.

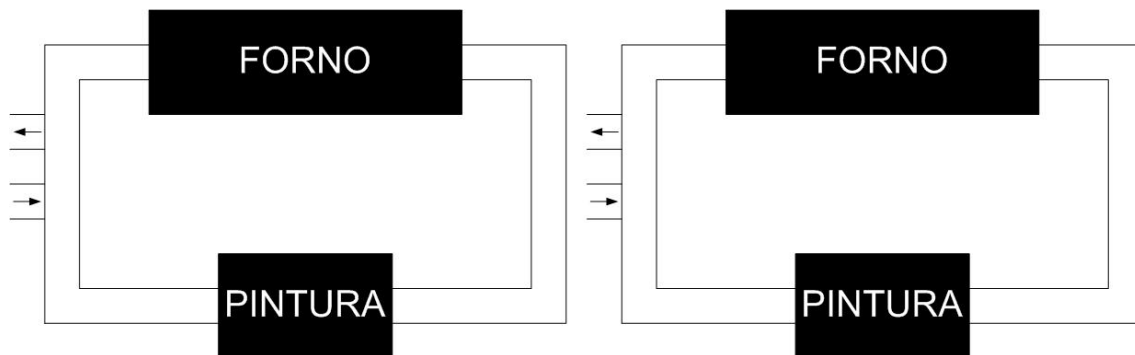


Figura 4.2: *Layout* Fabril do Processo de Produção - Solução Proposta 1

No contexto deste *layout*, dois transportadores estão disponíveis, permitindo alocar os diferentes tipos de peça de acordo com a abordagem que trará maiores vantagens para o *layout* em estudo. Assim e relembrando as velocidades dos transportadores para cada um dos tipos de peça (tabelas 4.1, 4.2 e 4.3), pode-se concluir que a eficiência do sistema é aumentada se o tipo de peça 1 for produzido em separado do tipo de peça 2 e 3.

4.3.2 Segunda solução Proposta

A segunda solução proposta consiste na adopção de uma abordagem inteiramente distinta, o uso de um sistema de transporte baseado em *AGVs*, uma solução emergente na actualidade. Neste contexto e relembrando os tempos de processamento para cada tipo de peça, pode-se verificar que o tempo necessário para a operação de pintura representa menos de metade do tempo total de processamento, o que significa que apenas será necessário no nosso sistema a existência de duas estações de cozedura e uma de pintura. O *layout* fabril seleccionado para esta solução é apresentado na figura 4.3.

Neste *layout*, e tal como já foi referido, a presença de transportadores é necessário devido ao facto de estes serem responsáveis pelo transporte das peças desde a entrada até à saída de cada estação de trabalho. No entanto para esta solução, estes permitirão disponibilizar ao mesmo tempo *buffers* nas suas extremidades, onde os *AGVs* poderão colocar ou levantar as peças. A existência destes *buffers* permite: à entrada dos transportadores criar um espaço onde os *AGVs* poderão deixar as peças sem haver a necessidade de que o transportador esteja livre; à saída criar um espaço em que as peças a serem transportadas para a estação seguinte, aguardam a chegada

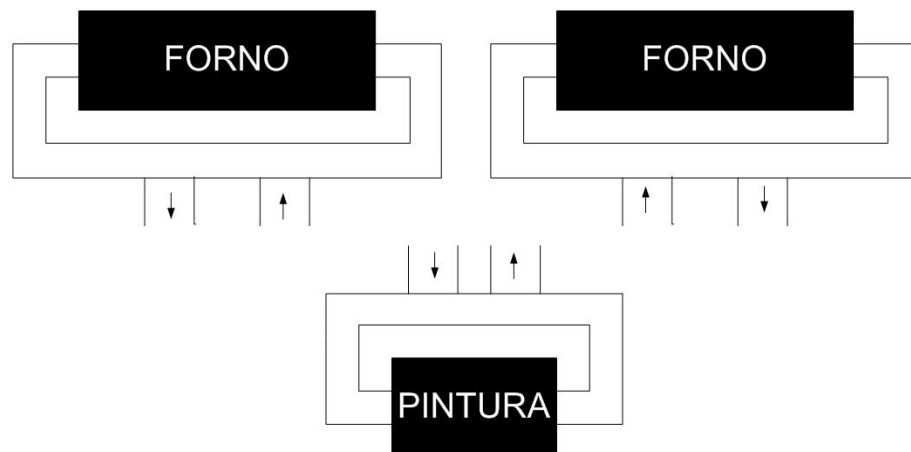


Figura 4.3: *Layout* Fabril do Processo de Produção - Solução Proposta 2

de um *AGV* sem haver a necessidade de parar o transportador. A existência de transportadores independentes possibilita ainda a definição de velocidades diferentes.

Neste *layout* fabril os *AGVs* serão responsáveis por recolher as peças na entrada do sistema e tendo a sequência de operações definida, que pode ser alocado por um controlo centralizado [23, 35], enviarão a peça para a estação de trabalho seguinte.

Como se pode verificar, com esta solução a flexibilidade do sistema é incrementada, uma vez que as sequências de operações de um tipo de peça, poderá ser rapidamente alterada, sem haver a necessidade de alterar o *layout* fabril, tal como acontece nos *layouts* anteriormente abordados. Nesta solução, e quando comparada com a primeira solução proposta, é também evitado a necessidade de aquisição de uma nova estação de pintura e conseqüente necessidade de contratação de um operador para o local, diminuindo ao mesmo tempo o espaço utilizado no chão de fábrica.

Além destes aspectos, a modularidade do *layout* permite ainda a adição e remoção de estações de trabalho de forma fácil, não tendo para isso de parar a produção.

Para avaliar as soluções propostas, foi seleccionada e tal como referido anteriormente, uma abordagem por simulação. Antes de mais, será necessário definir os diferentes parâmetros de cada um dos *layouts* abordados.

4.4 Abordagem por Simulação

Neste capítulo, e começando pelo *layout* actual, a definição dos parâmetros tanto para as soluções propostas como também para o sistema actual serão definidas, de forma a realizar a modelação do sistema no software ARENA.

4.4.1 Sistema de Produção Actual

Na figura 4.4 é apresentado o *layout* fabril actual com as distâncias consideradas.

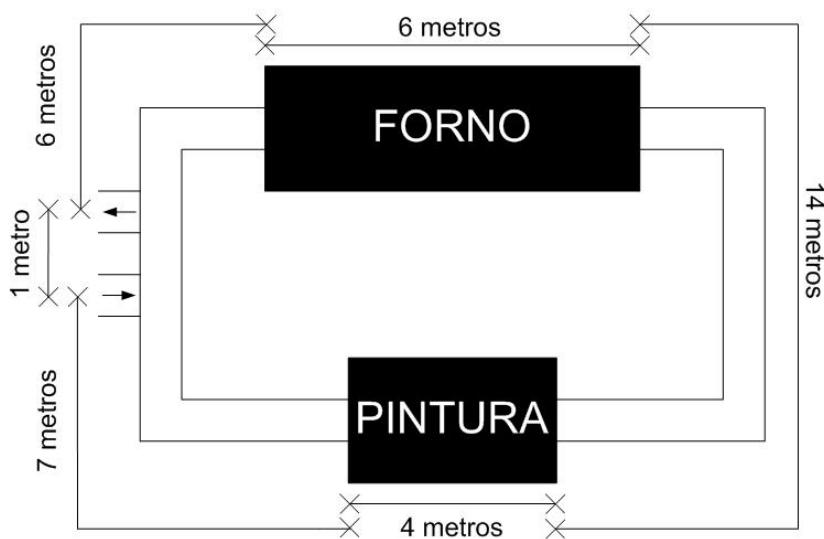


Figura 4.4: Distâncias *layout* actual

Tendo já definido anteriormente a sequência de operações para os diferentes tipos de peça, resta agora definir a velocidade do transportador e os tempos de atraso que serão considerados no sistema.

Como referido anteriormente serão seguidas duas abordagens: para a primeira a velocidade do transportador irá variar consoante o tipo de peça a ser produzido. Na segunda abordagem a velocidade do transportador irá tomar um valor constante.

Assim e analisando as tabelas 4.1, 4.2 e 4.3, na primeira abordagem a velocidade do transportador poderá variar entre a velocidade 0,05 m/s e 0,07 m/s. Para a segunda abordagem a velocidade do transportador será constante e igual a 0,05 m/s.

Os tempos de atraso considerados, correspondem ao tempo de remoção/inserção de peças no transportador. O tempo considerado necessário para a realização destas operações foi de 20 segundos.

4.4.2 Primeira Solução Proposta - Duplicação do Sistema de Produção Actual

Para esta primeira solução proposta irão ser consideradas as mesmas dimensões do *layout* fabril actual, bem como os tempos de atraso. A diferença apenas reside no facto de que agora existem dois transportadores, logo será necessário definir as suas velocidades. Tendo então em conta a ideia inicialmente descrita, o primeiro sistema irá produzir o tipo de peça 1 e o segundo o tipo de peça 2 e 3. Analisando mais uma vez as tabelas 4.1, 4.2 e 4.3, a velocidade do transportador A será igual a 0,05 m/s e a do transportador B igual a 0,07 m/s.

4.4.3 Segunda Solução Proposta - Sistema de transporte baseado em AGVs

Nesta solução, um novo sistema de transporte é introduzido no processo de produção. Logo a definição de novas distâncias, atrasos e velocidades terá de ser realizada.

Começando pela definição das distâncias do *layout* fabril, figura 4.5.

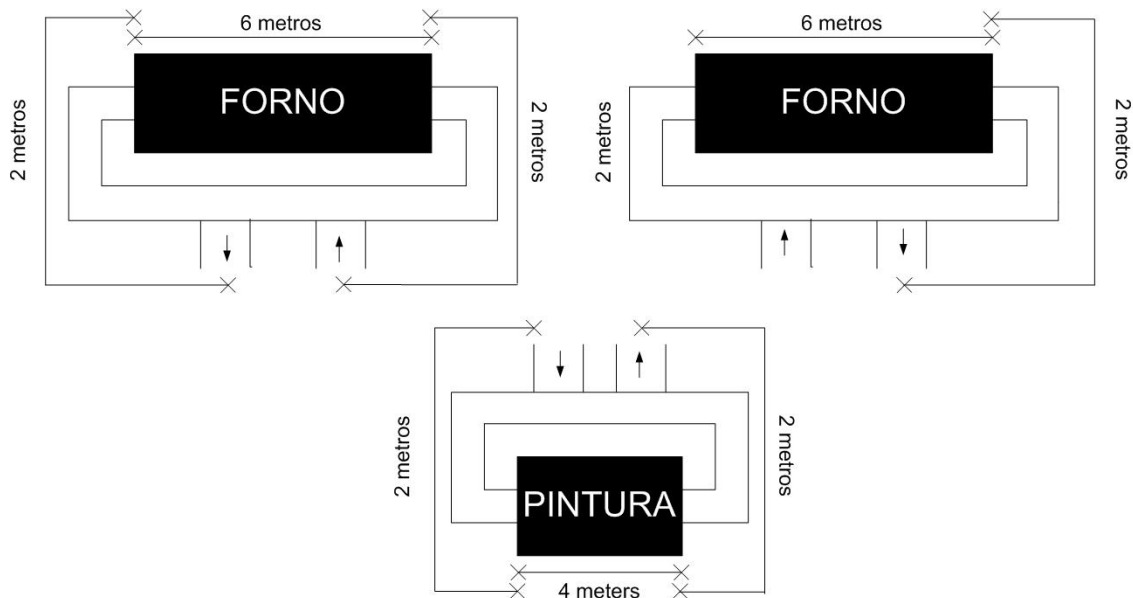


Figura 4.5: Distancias *layout* sistema baseado em AGVs

Definido o comprimento dos transportadores, é necessário definir as trajetórias dos AGVs. Para isso é necessário relembrar dois tipos de trajetórias que referi no capítulo 2: livres e restritas. Nas trajetórias livres é gozada uma maior liberdade de controlo do AGV, o que se traduz num aumento da flexibilidade, numa maior facilidade de controlo evitando colisões e numa diminuição dos tempo de transporte, sendo esta a escolhida para a nossa implementação.

Devido ao facto de o software ARENA não suportar o desenvolvimento de controlos de baixo nível, sendo que apenas possui nas suas opções trajetórias do tipo restritas com fiabilidade reduzida [12], será apenas considerado distâncias médias entre os postos de trabalho. Estas distâncias são apresentadas na tabela 4.5.

Com todas as distâncias definidas resta definir a velocidade dos transportadores. Observando o *layout* para esta solução, é possível auferir que todos os transportadores são independentes uns dos outros. Esse facto permite definir a velocidade dos transportadores de forma independente.

Assim, e seguindo a mesma abordagem que foi realizada na primeira solução em termos de produção dos tipos de peças, e analisando pela ultima vez as tabelas 4.1, 4.2, 4.3 e 4.4, conclui-se que as velocidades dos transportadores para quais se tira o máximo partido do *layout*, será definir a velocidade do transportador A igual a 0,05 m/s, a do transportador B igual a 0,07 m/s e a do transportador da pintura igual a 0,15 m/s.

Em relação ao sistema baseado em AGVs, as velocidades de cada veículo que serão consideradas na simulação são: 1, 2 e 3 m/s.

Para os tempos de atraso, neste caso como não existe nenhuma intervenção humana durante o processo de produção, vão corresponder ao tempos de carga e descarga das peças por parte dos AGVs. Os tempos de carga e descarga considerados foram de 10 segundos.

Tabela 4.5: Distâncias entre postos de trabalho consideradas

Origem	Destino	Distância (metros)
Entrada do sistema	Entrada transportador Pintura	11
Entrada do sistema	Saída transportador Pintura	14
Entrada do sistema	Entrada transportador Cozedura A	9
Entrada do sistema	Saída transportador Cozedura A	4
Entrada do sistema	Entrada transportador Cozedura B	21
Entrada do sistema	Saída transportador Cozedura B	16
Entrada do sistema	Saída do Sistema	20
Entrada transportador Pintura	Saída transportador Pintura	7
Entrada transportador Pintura	Entrada transportador Cozedura A	6
Entrada transportador Pintura	Saída transportador Cozedura A	11
Entrada transportador Pintura	Entrada transportador Cozedura B	14
Entrada transportador Pintura	Saída transportador Cozedura B	9
Entrada transportador Pintura	Saída do Sistema	13
Saída transportador Pintura	Entrada transportador Cozedura A	9
Saída transportador Pintura	Saída transportador Cozedura A	14
Saída transportador Pintura	Entrada transportador Cozedura B	11
Saída transportador Pintura	Saída transportador Cozedura B	7
Saída transportador Pintura	Saída do Sistema	11
Entrada transportador Cozedura A	Saída transportador Cozedura A	9
Entrada transportador Cozedura A	Entrada transportador Cozedura B	16
Entrada transportador Cozedura A	Saída transportador Cozedura B	11
Entrada transportador Cozedura A	Saída do Sistema	15
Saída transportador Cozedura A	Entrada transportador Cozedura B	21
Saída transportador Cozedura A	Saída transportador Cozedura B	16
Saída transportador Cozedura A	Saída do Sistema	20
Entrada transportador Cozedura B	Saída transportador Cozedura B	9
Entrada transportador Cozedura B	Saída do Sistema	3
Saída transportador Cozedura B	Saída do Sistema	8
Estação central	Entrada transportador Pintura	5
Estação central	Entrada transportador Cozedura A	7
Estação central	Entrada do sistema	12
Estação central	Saída transportador Pintura	5
Estação central	Saída transportador Cozedura B	7
Estação central	Saída transportador Cozedura A	12
Estação central	Entrada transportador Cozedura B	12
Estação central	Saída do Sistema	12

Três aspectos importantes ainda não referidos. Primeiro a existência de uma estação central entre o processo de cozedura A e o B para onde os veículos em repouso se deverão dirigir. Foi implementado com o objectivo de colocar os veículos numa zona central no sistema, permitindo que estes respondam mais rapidamente a novos pedidos de transporte. Segundo, a selecção do AGV para o transporte de cada peça é feito tendo em conta a sua disponibilidade e a sua proximidade com a estação onde o pedido foi realizado. Finalmente e terceiro, tendo em conta a distância da estação central ao *buffer* de saída de cada uma das estações de trabalho, o pedido para o transporte de uma peça por parte dos AGVs é feito segundos antes de a peça chegar ao *buffer* de saída do transportador (Controlo Preditivo), tentando com esta estratégia diminuir os tempos de transporte.

4.5 Resultados e sua Análise

Neste capítulo os resultados obtidos através de simulação e respectiva análise vão ser apresentados. Para cada uma das soluções duas hipóteses de produção foram consideradas. A primeira corresponde a uma produção em massa de 150 unidades, tendo esta simulação o objectivo de analisar os tempos mínimos necessários para a produção das 150 peças por parte de cada um dos *layouts* já apresentados. Esta tipo de produção é o utilizado actualmente na indústria alvo do nosso estudo.

A segunda abordagem de produção considerada consiste na chegada de novas ordens de fabrico em intervalos de tempo de n minutos, pretendendo neste caso avaliar a capacidade de resposta do sistema de produção às ordens de fabrico. Esta última apesar de não ser considerada para a aplicação (revestimentos de peças por pó) nos dias de hoje, será interessante analisar uma vez que poderá servir propósitos de outras aplicações em que um *layout* fabril similar esteja implementado.

Será importante referir que os resultados de seguida apresentados foram obtidos a partir da corrida de 20 replicações no simulador, sendo apresentado o respectivo valor médio.

4.5.1 Resultados - Produção de 150 peças

Para a simulação apresentada nesta secção, foi considerada a existência de um *buffer* de 150 peças na entrada do sistema. Aqui e com o objectivo de aproximar o mais possível as condições iniciais de simulação de cada um dos *layouts* apresentados, e por forma a que os resultados não sejam afectados por outras variáveis a não ser pelo sistema de transporte considerado, foi definido que das 150 peças iniciais a serem produzidas, 70 são do tipo 1, 40 do tipo 2 e 50 do tipo 3.

Os resultados da simulação são de seguida apresentados, começando pela *layout* actual e primeira solução proposta, sendo posteriormente apresentados os resultados para a solução baseada em AGVs.

Na tabela 4.6 é apresentado o tempo necessário da produção dessas 150 peças para a solução um e actual.

Tabela 4.6: Tempos de Produção das 150 peças - Soluções baseada exclusivamente em transportadores

Solução	Tempo de Processamento (minutos)
Actual - Velocidade do transportador de Transporte Constante	188
Actual - Velocidade do transportador Variável	221
Primeira Solução Proposta	122

Como se pode verificar por via da análise da tabela 4.6 e do gráfico 4.6 nos quais são apresentados os resultados para as soluções exclusivamente baseadas em transportadores, a primeira solução proposta é a que apresenta melhores resultados. No entanto é necessário ter em conta que esta solução é baseada na duplicação do sistema actual, o qual suscita a necessidade de realizar um elevado investimento.

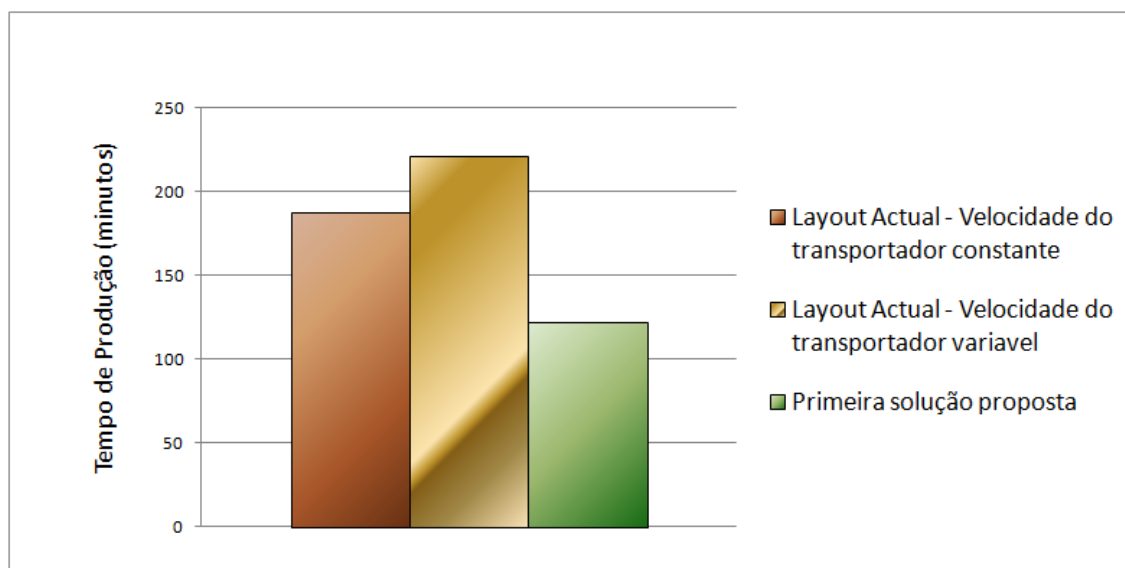


Figura 4.6: Gráfico Comparativo soluções baseadas em transportadores - Tempos de produção de 150 peças

Pode-se ainda verificar uma grande diferença de tempos de produção entre a solução actual com a velocidade do transportador constante e a solução actual com a velocidade do transportador variável consoante o tipo de peça em produção. Esta diferença deve-se ao facto de na segunda ser necessário esperar que todas as sequências de operações de um determinado tipo de peça estejam concluídas para poder introduzir um tipo de peça novo, diferença de tempo que terá no entanto tendência para diminuir e até mesmo inverter os resultados com o aumento do número de peças a serem produzidos.

No que diz respeito à solução baseada em AGVs, foram implementadas duas estratégias de controlo. Uma consiste na definição de prioridades de transporte. Aqui duas simulações serão realizadas, uma considerando que a prioridade de transporte de peças das estações de trabalho e da entrada do sistema são iguais. Uma segunda em que é dada maior prioridade ao transporte de peças das estações de trabalho em relação à entrada do sistema, ou seja, se houver o pedido de transporte de uma peça na entrada do sistema, mas ao mesmo tempo existir um pedido de transporte de peças na saída de uma das operações, esta última é atendida em primeiro lugar.

A segunda estratégia considerada prende-se com a possibilidade de cada AGV poder transportar mais do que uma peça. Neste sentido foram realizadas duas simulações uma considerando que os AGVs transportariam apenas uma peça e uma segunda em que poderão transportar duas peças do mesmo tipo em simultâneo.

Tal como já foi referido, todas estas simulações foram realizadas para as velocidades dos AGVs correspondente a 1, 2 e 3 m/s.

Após alguma recolha de resultados iniciais pode-se verificar que os *buffers* de entrada/saída dos transportadores nas diferentes estações de trabalho tomavam valores elevados, na ordem das 35 peças, algo irrealista para o nosso processo de produção. Foi então necessário tomar medidas

para impedir que houvesse uma acumulação de peças tão elevadas.

Assim com o objectivo de limitar o número de peças nos *buffers* intermédios duas abordagens foram consideradas. A primeira consistia em saber o número de peças na entrada e saída de cada uma das operações e bloquear o transporte (por parte dos *AGVs*) caso o *buffer* da estação seguinte, para onde a peça será transportada, exceda determinadas dimensões. No entanto esta abordagem não foi possível de implementar devido ao facto de não ser possível no software ARENA identificar qual o tipo de peça que cada *AGV* transporta. Além deste aspecto como a operação pintura é um ponto de convergência, em que há sua entrada poderão chegar peças oriundas da estação de cozedura A, cozedura B e da entrada do sistema tornou-se bastante complicado controlar de forma eficiente (tendo em conta as diferentes ferramentas disponibilizadas pelo software ARENA) o fluxo de peças no sistema.

Uma segunda abordagem, e a qual foi implementada, consiste na limitação do número de peças em produção de cada um dos tipos. Limitando o número de peças de cada tipo em produção, ou seja, limitando o fluxo conseguimos controlar o tamanho dos *buffers* do sistema. Assim, e neste contexto, o limite de produção simultânea de peças considerado foi: 10 peças do tipo 1, 10 peças do tipo 2 e 10 peças do tipo 3, ou seja, inicialmente o sistema admite 10 peças de cada tipo deixando de seguida entrar um nova peça de um determinado tipo sempre que uma peça desse mesmo tipo deixa o sistema. Este valor foi escolhido, pois pretendia-se que o limite máximo dos *buffers* não ultrapassa-se em demasia os valores de 10 peças (valor considerado razoável para a aplicação em questão), tendo-se já verificado para esta estratégia (10 peças de cada tipo) alguns resultados acima do limite. Do ponto de vista dos tempos de produção, uma diminuição nos tempos será sinónimo de uma forma geral de aumento dos tamanhos dos *buffers*, cabendo a cada empresa interessada delimitar este tamanho consoante os seus interesses. No entanto simulações tendo em conta o limite de peças em produção de cada tipo de 11, 12 foi considerado, tendo-se chegado à conclusão que tanto o tamanho dos *buffers* como os tempos de produção não sofreram alterações assinaláveis.

Ao mesmo tempo nos resultados apresentados a taxa de utilização do *AGVs* será apresentada, permitindo com isto avaliar se o sistema de transporte atingiu o seu limite crítico, ou se por outro lado terá capacidade de resposta no caso de um aumento do fluxo de peças em produção ser considerado.

Os resultados de seguida apresentados (tabelas 4.7, 4.8, 4.9, 4.10, 4.11 e 4.12) referem-se à simulação considerando a velocidade dos *AGV's* igual a 1 m/s, 2 m/s e 3 m/s, tendo este apenas a capacidade de transporte de apenas uma peça. De notar que os valores apresentados para o tamanho dos *buffers* corresponde ao valor máximo obtido nas 20 replicações efectuadas.

Começando então por analisar os resultados referentes à velocidade dos *AGVs* correspondentes a 1 m/s podemos concluir que de uma forma geral os valores dos tamanhos dos *buffers* estão dentro de valores já bastante aceitáveis para o sistema de produção aqui apresentado. Verificada esta condicionante podemos agora escolher qual a melhor solução para esta velocidade que satisfaz os nossos requisitos, mas tendo sempre em mente o paradigma custo/performance. Ora tendo em conta o tempo de produção da solução actual que se situa nos 188 minutos podemos verificar que

Tabela 4.7: Resultados Tempos de Produção de 150 peças - Solução AGVs 1m/s - Prioridades de transporte iguais

Número de AGV's	Tamanho dos Buffers em Peças						Tempo de Produção (minutos)	Utilização (%)
	Entrada Cozedura A	Entrada Cozedura B	Entrada Pintura	Saída Cozedura A	Saída Cozedura B	Saída Pintura		
1	1	1	1	10	2	2	563	100
2	2	2	2	10	3	4	281	100
3	3	3	3	10	5	6	207	93
4	2	6	4	10	5	8	175	85
5	3	9	5	10	6	9	159	75
6	5	10	6	10	6	6	149	67
7	4	13	8	9	2	6	141	61
8	6	13	11	7	2	6	134	56
9	6	13	14	4	2	6	130	52

Tabela 4.8: Resultados Tempos de Produção de 150 peças - Solução AGVs 1m/s - Prioridades de transporte diferentes

Número de AGV's	Tamanho dos Buffers em Peças						Tempo de Produção (minutos)	Utilização (%)
	Entrada Cozedura A	Entrada Cozedura B	Entrada Pintura	Saída Cozedura A	Saída Cozedura B	Saída Pintura		
1	1	1	1	3	2	3	572	99
2	2	2	2	4	3	4	291	98
3	3	3	3	5	5	6	201	96
4	2	6	4	4	5	8	166	85
5	3	9	5	4	5	9	150	79
6	5	10	6	6	6	6	139	72
7	4	13	8	2	2	3	133	64
8	6	13	11	3	2	4	131	58
9	6	13	14	2	2	6	129	52

independentemente da estratégia escolhida (com prioridades iguais ou diferentes) a melhor solução seria a de 4 AGVs, uma vez que seria aquela que com o menor investimento traria para o sistema a flexibilidade tanto desejada bem como uma melhoria nos tempos de produção. Apesar das diferença entre as duas estratégias de controlo não serem muito significativas podemos verificar que neste caso a melhor abordagem seria a de definir uma prioridade superior ao transporte de peças das estações de trabalho do que ao transporte na entrada do sistema, já que além de se verificar para este controlo uma ligeira diminuição do tamanho dos *buffers* também se verifica uma pequena descida nos tempos de produção para valores abaixo dos 188 minutos (tempo de produção da solução actual).

Analisando agora os resultados para a velocidade de 2 m/s podemos verificar e tal como esperado um queda nos tempos de produção. Uma conclusão que ainda podemos retirar e esta

sim não tão intuitiva é o facto de os *buffers* do sistema manterem-se em valores baixos, próximo dos conseguidos para a velocidade correspondente a 1 m/s. Mais um vez a definição de prioridades diferentes entre as estações provou ser mais eficaz conseguindo com isso aumentar ligeiramente a performance do sistema, mantendo os *buffers* do sistema baixos.

Comparando então os resultados obtidos para a velocidade dos *AGVs* correspondentes a 2 m/s com o tempo de produção do sistema actual podemos concluir que a melhor escolha a ser realizada será optar por 3 *AGVs* com prioridades entre estações diferentes.

Tabela 4.9: Resultados Tempos de Produção de 150 peças - Solução *AGVs* 2m/s - Prioridades de transporte iguais

Número de <i>AGV</i> 's	Tamanho dos Buffers em Peças						Tempo de Produção (minutos)	Utilização (%)
	Entrada Cozedura A	Entrada Cozedura B	Entrada Pintura	Saída Cozedura A	Saída Cozedura B	Saída Pintura		
1	1	1	1	10	2	2	419	100
2	2	2	2	10	4	4	218	97
3	3	4	3	10	4	4	173	84
4	3	7	4	10	5	7	152	72
5	4	10	5	10	4	6	141	63
6	4	12	8	9	2	5	133	56
7	7	12	11	5	1	5	127	50
8	7	13	15	3	1	5	124	45
9	7	12	17	3	1	2	123	41

Tabela 4.10: Resultados Tempos de Produção de 150 peças - Solução *AGVs* 2m/s - Prioridades de transporte diferentes

Número de <i>AGV</i> 's	Tamanho dos Buffers em Peças						Tempo de Produção (minutos)	Utilização (%)
	Entrada Cozedura A	Entrada Cozedura B	Entrada Pintura	Saída Cozedura A	Saída Cozedura B	Saída Pintura		
1	1	1	1	3	2	2	427	99
2	3	2	2	4	4	4	217	98
3	4	4	3	5	5	4	167	86
4	3	7	4	5	4	4	145	75
5	4	10	5	7	4	5	134	64
6	3	12	8	4	2	5	127	58
7	7	12	11	2	1	5	124	51
8	7	13	15	2	1	5	124	45
9	7	12	17	2	1	2	123	41

Finalmente iremos analisar os resultados para a velocidade de 3 m/s. Aqui mais uma vez podemos verificar que os *buffers* mantêm-se dentro de valores aceitáveis para um número de *AGVs* mais baixo juntamente com uma diminuição ligeira dos tempos de produção. Mais uma

vez podemos verificar que a melhor solução a ser tomada será a de 3 AGVs com a definição de prioridades diferentes. Analisando os resultados até agora obtidos podemos verificar que existe um ganho mais elevado no tempo de produção quando comparada a solução a 1 m/s com a de 2 m/s do que a de 2m/s para a de 3 m/s.

Tabela 4.11: Resultados Tempos de Produção de 150 peças - Solução AGVs 3m/s - Prioridades de transporte iguais

Número de AGV's	Tamanho dos Buffers em Peças						Tempo de Produção (minutos)	Utilização (%)
	Entrada Cozedura A	Entrada Cozedura B	Entrada Pintura	Saída Cozedura A	Saída Cozedura B	Saída Pintura		
1	1	1	1	10	2	2	370	100
2	3	2	2	10	2	4	201	93
3	3	4	3	10	3	4	163	78
4	4	9	4	10	7	7	144	67
5	4	11	6	10	2	4	135	57
6	6	11	10	6	1	4	126	51
7	6	12	14	3	2	4	123	45
8	6	11	17	4	1	3	121	40
9	6	11	18	1	1	2	120	36

Tabela 4.12: Resultados Tempos de Produção de 150 peças - Solução AGVs 3m/s - Prioridades de transporte diferentes

Número de AGV's	Tamanho dos Buffers em Peças						Tempo de Produção (minutos)	Utilização (%)
	Entrada Cozedura A	Entrada Cozedura B	Entrada Pintura	Saída Cozedura A	Saída Cozedura B	Saída Pintura		
1	1	1	1	3	2	2	377	99
2	2	2	2	4	2	3	197	95
3	4	4	3	5	4	4	155	82
4	5	9	4	4	6	7	135	71
5	4	11	6	4	2	4	129	60
6	6	11	10	2	1	4	125	53
7	6	12	14	1	2	4	122	46
8	6	11	17	1	1	3	121	40
9	6	11	18	1	1	2	120	36

Depois de analisados os resultados para velocidades diferentes dos AGVs, foi posta em questão uma estratégia que poderia beneficiar o investimento a ter de ser realizado para a aquisição e instalação destes sistema de transporte. Com este objectivo foi então considerada a possibilidade de os AGVs poderem transportar até duas peças em simultâneo. Este factor permitirá em teoria reduzir o número de veículos necessários para o sistema de produção atingir os resultados

pretendidos, reduzindo assim os custos, uma vez que para adaptar um *AGV* para o transporte de duas peças tornar-se menos custoso que a aquisição de um *AGV* extra.

Ora com a possibilidade de os *AGVs* transportarem duas peças em simultâneo é necessário portanto ajustar a nossa estratégia de produção por forma a tirar maior partido deste facto. Assim sendo, a produção de um máximo de 10 peças de cada tipo em simultâneo continuará a ser considerada, mas agora entrarão duas peças novas do mesmo tipo quando saírem duas do mesmo tipo do sistema. De seguida são apresentados os resultados obtidos.

Tabela 4.13: Resultados Tempos de Produção de 150 peças - Solução *AGVs* 1m/s e transporte de 2 peças em simultâneo- Prioridades de transporte iguais

Número de <i>AGV</i> 's	Tamanho dos Buffers em Peças						Tempo de Produção (minutos)	Utilização (%)
	Entrada Cozedura A	Entrada Cozedura B	Entrada Pintura	Saída Cozedura A	Saída Cozedura B	Saída Pintura		
1	3	2	2	10	4	4	284	99
2	3	6	4	10	6	8	180	82
3	5	9	6	10	6	6	157	64
4	7	12	11	6	2	6	143	53
5	7	13	17	2	2	6	136	46
6	7	11	17	2	2	2	136	38
7	7	11	21	2	2	2	136	33
8	7	11	23	2	2	2	136	29
9	7	11	23	2	2	2	136	26

Analisando as tabelas 4.13 e 4.14 e comparando com resultados anteriormente obtidos para a velocidade de 1 m/s podemos verificar que houve uma redução considerável dos tempos de produção, apesar de se ter verificado uma subida considerável do tamanho dos *buffers*, principalmente para um número de *AGVs* superior a 5. No entanto e analisando só para casos mais realistas para o nosso sistema, em que o número de *AGVs* ideal será entre dois a quatro podemos verificar que o tamanho dos *buffers* se mantém dentro de valores admissíveis. Podemos ainda verificar que a definição de controlo com prioridades diferentes para as estações de trabalho se mantém como aquela em que melhores resultados são alcançados.

Analisando agora a reposta do sistema para a velocidade dos *AGVs* definida para 2 m/s, pode-se concluir que mais uma vez os tempos de produção sofreram uma pequena descida para o número de *AGVs* mais baixo, mantendo-se o tamanho dos *buffers* bastante idênticos aos obtidos para a velocidade 1 m/s.

Para os resultados obtidos para 2 m/s a melhor escolha a ser tomada será a opção por 2 *AGVs* com o controlo com prioridades diferentes definido. Podemos desde já verificar que a opção de transporte de duas peças em simultâneo melhora os tempos de produção poupando-se ao mesmo tempo em custos uma vez que o número de *AGVs* necessário também diminui.

Finalmente para a velocidade de 3 m/s não se verifica uma melhoria significativa em relação aos resultados obtidos para 2 m/s, deixando adivinhar que esta solução será preterida uma vez

Tabela 4.14: Resultados Tempos de Produção de 150 peças - Solução AGVs 1m/s e transporte de 2 peças em simultâneo- Prioridades de transporte diferentes

Número de AGV's	Tamanho dos Buffers em Peças						Tempo de Produção (minutos)	Utilização (%)
	Entrada Cozedura A	Entrada Cozedura B	Entrada Pintura	Saída Cozedura A	Saída Cozedura B	Saída Pintura		
1	3	2	2	6	4	4	286	99
2	4	6	4	4	4	8	171	86
3	5	9	6	6	6	6	148	67
4	7	12	11	4	4	6	140	55
5	7	13	17	2	2	6	136	46
6	7	11	17	2	2	2	136	39
7	7	11	21	2	2	2	136	33
8	7	11	23	2	2	2	136	29
9	7	11	23	2	2	2	136	26

Tabela 4.15: Resultados Tempos de Produção de 150 peças - Solução AGVs 2m/s e transporte de 2 peças em simultâneo- Prioridades de transporte iguais

Número de AGV's	Tamanho dos Buffers em Peças						Tempo de Produção (minutos)	Utilização (%)
	Entrada Cozedura A	Entrada Cozedura B	Entrada Pintura	Saída Cozedura A	Saída Cozedura B	Saída Pintura		
1	2	2	2	10	4	4	224	94
2	4	6	4	10	6	8	159	69
3	4	12	8	8	4	4	134	55
4	7	13	15	4	2	4	131	44
5	7	12	20	2	2	2	130	35
6	7	11	20	2	2	2	130	29
7	7	11	23	2	2	2	129	25
8	7	11	25	2	2	2	129	22
9	7	11	25	2	2	2	129	19

que com o aumento da velocidade a duração das baterias diminui obrigando o veículo a dirigir-se um maior número de vezes aos pontos de recarregamento, diminuindo com isso a capacidade de transporte do sistema, afectando também a longevidade das baterias. Outra factor importante será as preocupações com a segurança, que para o caso da velocidade ser igual a 3 m/s será mais apertada que para 2 m/s apesar de ambos os casos um acesso restrito aos locais onde os AGVs circulam poderá ser considerado.

Apresentados então os resultados obtidos para as estratégias consideradas, resta agora seleccionar a solução que melhor se adequa ao sistema aqui apresentado. Para isso, e na tabela 4.19, são lembradas as escolhas feitas tendo em conta as velocidades diferentes consideradas bem como o número de peças que o veículo transporta. A variável controlo com prioridades iguais ou diferentes foi retirada, uma vez que já foi concluído que esta última trás vantagens tanto

Tabela 4.16: Resultados Tempos de Produção de 150 peças - Solução AGVs 2m/s - Prioridades de transporte diferentes

Número de AGV's	Tamanho dos Buffers em Peças						Tempo de Produção (minutos)	Utilização (%)
	Entrada Cozedura A	Entrada Cozedura B	Entrada Pintura	Saída Cozedura A	Saída Cozedura B	Saída Pintura		
1	3	2	2	4	4	4	220	96
2	3	6	4	4	4	4	150	73
3	5	12	8	4	2	4	133	56
4	7	13	15	2	2	4	130	43
5	7	12	20	2	2	2	130	35
6	7	11	20	2	2	2	129	29
7	7	11	23	2	2	2	129	25
8	7	11	25	2	2	2	129	22
9	7	11	25	2	2	2	129	19

Tabela 4.17: Resultados Tempos de Produção de 150 peças - Solução AGVs 3m/s e transporte de 2 peças em simultâneo - Prioridades de transporte iguais

Número de AGV's	Tamanho dos Buffers em Peças						Tempo de Produção (minutos)	Utilização (%)
	Entrada Cozedura A	Entrada Cozedura B	Entrada Pintura	Saída Cozedura A	Saída Cozedura B	Saída Pintura		
1	3	2	2	10	6	6	205	91
2	7	10	4	10	8	8	149	65
3	6	12	11	4	2	4	132	49
4	7	11	17	2	2	4	128	38
5	7	11	21	2	2	2	128	31
6	7	11	21	2	2	2	128	26
7	7	11	23	2	2	2	128	22
8	7	11	25	2	2	2	128	19
9	7	11	25	2	2	2	128	17

para os tempos de produção bem como para o tamanho dos *buffers*.

Observando então a tabela 4.19, podemos verificar que a dimensão dos *buffers* estão todos dentro de valores aceitáveis não sendo então um factor de decisão entre as soluções finais consideradas. Assim a escolha recairá por aquela em que a relação performance/custo do sistema de transporte seja a melhor. Para isso uma ideia chave é o facto de a adaptação de 1 AGV para o transporte de duas peças torna-se mais barato do que a compra de um AGV extra. Logo a primeira solução é eliminada. Observando então os restantes resultados podemos verificar que as soluções para as quais temos um número de AGVs mais reduzido corresponde às duas ultimas soluções, sendo as restantes hipóteses então eliminadas. Comparando então estas duas ultimas soluções, podemos verificar que os tempos de produção não diferem de forma considerável. Consideramos então outro factor, o desgaste mecânico e das baterias. Assim sendo a nossa escolha recairá pela

Tabela 4.18: Resultados Tempos de Produção de 150 peças - Solução AGVs 3m/s e transporte de 2 peças em simultâneo- Prioridades de transporte diferentes

Número de AGV's	Tamanho dos Buffers em Peças						Tempo de Produção (minutos)	Utilização (%)
	Entrada Cozedura A	Entrada Cozedura B	Entrada Pintura	Saída Cozedura A	Saída Cozedura B	Saída Pintura		
1	4	3	2	4	4	6	203	93
2	7	10	4	4	8	8	142	68
3	6	12	11	2	2	4	131	50
4	7	11	17	2	2	4	128	38
5	7	11	21	2	2	2	128	31
6	7	11	21	2	2	2	128	26
7	7	11	23	2	2	2	128	22
8	7	11	25	2	2	2	128	19
9	7	11	25	2	2	2	128	17

Tabela 4.19: Comparação entre os melhores resultados obtidos para a produção em massa de 150 peças (paradigma custo/performance)

Transporte de peças em Simultâneo	Velocidade (m/s)	Número de AGV's	Tamanho dos Buffers em Peças						Tempo de Produção (minutos)	Utilização (%)
			Entrada Cozedura A	Entrada Cozedura B	Entrada Pintura	Saída Cozedura A	Saída Cozedura B	Saída Pintura		
1	1	4	2	6	4	10	5	8	175	85
1	2	3	4	4	3	5	5	4	167	86
1	3	3	4	4	3	5	4	4	155	82
2	1	3	5	9	6	6	6	6	148	67
2	2	2	3	6	4	4	4	4	150	73
2	3	2	7	10	4	4	8	8	142	68

solução com transporte de duas peças em simultâneo, com a velocidade dos AGVs de 2 m/s, com o controlo com diferentes prioridades implementado, uma vez que com esta solução tanto as baterias como a parte mecânica são resguardadas devido ao uso de uma velocidade mais reduzida.

4.5.2 Resultados - Produção de n em n minutos

Concluída então a análise dos resultados para a produção de 150 peças, irá ser agora apresentada uma segunda estratégia de produção, a produção de n em n minutos. Aqui será considerado um turno de produção de 12 horas com a chegada de ordens de fabrico em intervalos de tempo de 1 minuto e 3 minutos.

De notar que para esta estratégia de produção e para a solução actual baseada em transportadores, não faz sentido realizar a simulação para a velocidade do transportador variável com o tipo de peça em produção, uma vez que a chegada dos tipos de peças é aleatória levando assim a constantes mudanças na velocidade do transportador o que irá aumentar drasticamente o tempo de produção devido aos tempos de *setup* dos equipamentos. Posto isto para a solução

Tabela 4.20: Resultados Tempos de Produção para a chegada de novas ordens de fabrico de 1 em 1 minuto - Solução AGVs 1,2 e 3 m/s - Prioridades de transporte diferentes

Turno de 12 horas			
Chegada de Novas Peças			
1 em 1 minuto	Ordens de Fabrico	Peças Produzidas	
Solução actual VC	715	540	
Solução 1	715	709	
Soluções AGV - 1 m/s			
Numero AGV's	Ordens de Fabrico	Peças Produzidas	Utilização AGV %
1	715	183	100
2	715	369	100
3	715	542	99
4	715	706	97
5	715	715	84
Soluções AGV - 2 m/s			
Numero AGV's	Ordens de Fabrico	Peças Produzidas	Utilização AGV %
1	715	248	100
2	715	488	99
3	715	710	96
4	715	715	77
5	715	715	63
Soluções AGV - 3 m/s			
Numero AGV's	Ordens de Fabrico	Peças Produzidas	Utilização AGV %
1	715	281	100
2	715	548	99
3	715	715	88
4	715	715	68
5	715	715	56

actual será apenas considerada a solução em que a velocidade dos transportadores é mantida constante e igual à velocidade que garante que todos os tempo de processamento dos diferentes tipos de peças seja garantido, preterindo assim pelos motivos já referidos a solução em que a velocidade do transportador varia com o tipo de peça em questão. Serão ainda apresentados apenas os resultados considerando um controlo em que as estações de trabalho apresentam uma prioridade de transporte superior aos pedidos efectuados pela entrada do sistema uma vez que se verificou (comprovada com resultados) que uma ligeira melhoria da capacidade de resposta das soluções com estas estratégias.

As velocidades dos AGVs consideradas são : 1 m/s, 2 m/s e 3 m/s. Mais uma vez foram realizadas simulações, para a solução baseada em AGVs, tendo em conta o transporte de uma peça de cada vez ou duas peças do mesmo tipo em simultâneo.

A probabilidade de cada ordem de fabrico nova ser do tipo 1 é igual a 50 por cento, do tipo 2 é igual a 25 por cento e finalmente do tipo 3 será de 25 por cento.

Nas tabelas 4.20 e 4.21 são apresentados os resultados para a chegada de ordens de fabrico de 1 em 1 minuto e de 3 em 3 minutos respectivamente, considerando que o AGV apenas pode transportar uma peça de cada vez.

Começando então por analisar os resultados para a chegada de ordens de produção de 1 em 1 minuto podemos verificar que para a velocidade de 1 m/s os resultados ficam um pouco abaixo daquilo que são os requisitos do sistema, uma vez que será necessário adquirir 4 AGVs para o sistema de produção responder à chegada das ordens de fabrico, o que requer um grande

Tabela 4.21: Resultados Tempos de Produção para a chegada de novas ordens de fabrico de 3 em 3 minuto - Solução AGVs 1,2 e 3 m/s - Prioridades de transporte diferentes

Turno de 12 horas			
Chegada de Novas Peças			
3 em 3 minutos	Ordens de Fabrico	Peças Produzidas	
Solução actual VC	238	218	
Solução 1	239	218	
Soluções AGV - 1 m/s			
Número AGV's	Ordens de Fabrico	Peças Produzidas	Utilização AGV %
1	238	183	100
2	238	227	66
3	238	228	46
4	238	228	35
5	238	228	28
Soluções AGV - 2 m/s			
Número AGV's	Ordens de Fabrico	Peças Produzidas	Utilização AGV %
1	238	223	92
2	238	228	50
3	238	228	33
4	238	228	25
5	238	228	21
Soluções AGV - 3 m/s			
Número AGV's	Ordens de Fabrico	Peças Produzidas	Utilização AGV %
1	238	225	82
2	238	228	42
3	238	228	29
4	238	228	22
5	238	228	18

Tabela 4.22: Resultados Tempos de Produção para a chegada de novas ordens de fabrico de 1 em 1 minuto - Solução AGVs 1,2 e 3 m/s com transporte de 2 peças em simultâneo - Prioridades de transporte diferentes

Turno de 12 horas			
Chegada de Novas Peças			
1 em 1 minuto	Ordens de Fabrico	Peças Produzidas	
Solução actual VC	715	540	
Solução 1	715	709	
Soluções AGV - 1 m/s			
Número AGV's	Ordens de Fabrico	Peças Produzidas	Utilização AGV %
1	715	352	99
2	715	704	76
3	715	704	70
4	715	706	54
5	715	706	43
Soluções AGV - 2 m/s			
Número AGV's	Ordens de Fabrico	Peças Produzidas	Utilização AGV %
1	715	494	99
2	715	704	74
3	715	706	51
4	715	706	39
5	715	706	31
Soluções AGV - 3 m/s			
Número AGV's	Ordens de Fabrico	Peças Produzidas	Utilização AGV %
1	715	550	99
2	715	706	66
3	715	706	45
4	715	706	33
5	715	706	27

investimento. A solução mais promissora será optar por 3 AGVs com uma velocidade de 2 m/s uma vez que as baterias são resguardadas (quando comparada com uma a velocidade de 3m/s) estando a capacidade de produção ao nível do desejado.

Para a chegada de ordens de fabrico de 3 em 3 minutos a exigência do nível de produção é mais reduzido, facto que se traduz nos resultados obtidos para a velocidade de 1 m/s. Assim sendo a solução mais promissora será optar pela solução de 2 AGVs com a velocidade de 1 m/s.

Tabela 4.23: Resultados Tempos de Produção para a chegada de novas ordens de fabrico de 3 em 3 minutos - Solução AGVs 1,2 e 3 m/s com transporte de 2 peças em simultâneo - Prioridades de transporte diferentes

Turno de 12 horas			
Chegada de Novas Peças			
3 em 3 minutos			
	Ordens de Fabrico	Peças Produzidas	
Solução actual VC	238	218	
Solução 1	239	218	
Soluções AGV - 1 m/s			
Número AGV's	Ordens de Fabrico	Peças Produzidas	Utilização AGV %
1	238	236	68
2	238	236	36
3	238	236	24
4	238	236	18
5	238	236	15
Soluções AGV - 2 m/s			
Número AGV's	Ordens de Fabrico	Peças Produzidas	Utilização AGV %
1	238	236	50
2	238	236	25
3	238	236	17
4	238	236	13
5	238	236	10
Soluções AGV - 3 m/s			
Número AGV's	Ordens de Fabrico	Peças Produzidas	Utilização AGV %
1	238	236	44
2	238	236	22
3	238	236	14
4	238	236	11
5	238	236	8

Consideradas então as soluções baseadas apenas no transporte de uma peça, nas tabelas 4.22 e 4.23 apresentados os resultados considerando a possibilidade de o AGV transportar duas peças em simultâneo.

Mais uma vez iniciando a análise pela chegada de ordens de fabrico de 1 em 1 minuto as soluções para a velocidade correspondente a 1m/s melhoraram significativamente, de tal forma que a melhor solução a ser considerada será 2 AGVs com a velocidade de 1 m/s.

Para a chegada de ordens de fabrico de 3 em 3 minutos podemos mais uma vez verificar uma melhoria nos resultados, sendo a melhor solução tendo em conta o seu custo e reposta do sistema será de 1 AGV a 1 m/s.

Obtidos então os resultados para ambas as produções serão agora comparados, tabelas 4.24 e 4.25, os melhores resultados obtidos tendo em conta o paradigma custo/performance.

Começando então por comparar os melhores resultados para chegada de ordens de fabrico de 1 em 1 minuto podemos verificar que a solução que apresenta melhor resposta é a solução

Tabela 4.24: Melhores resultados obtidos considerando a produção de 1 em 1 minuto

Turno de 12 horas				
Chegada de Novas Peças				
1 em 1 minutos	Ordens de Fabrico	Peças Produzidas		
Solução actual VC	715	540		
Solução 1	715	709		
Solução Agv's -1 m/s			Utilização AGV %	Número de peças transportadas em simultâneo
3	715	710	96	1
2	715	704	76	2

correspondente a 3 AGVs. Contudo a diferença de unidades produzidas entre esta solução e a corresponde a dois AGVs onde são transportadas duas peças em simultâneo não compensa o investimento extra na aquisição de mais 1 AGV (a aquisição de mais um AGV fica mais cara do que ajustar um AGV para o transporte de duas peças). Posto isto a melhor escolha para responder à chegada de ordens de produção de 1 em 1 minuto é a solução de 2 AGVs com velocidade de 1 m/s e transporte de duas peças em simultâneo.

Tabela 4.25: Melhores resultados obtidos considerando a produção de 3 em 3 minuto

Turno de 12 horas				
Chegada de Novas Peças				
3 em 3 minutos	Ordens de Fabrico	Peças Produzidas		
Solução actual VC	343	327		
Solução 1	346	331		
Solução Agv's -1 m/s			Utilização AGV %	Número de peças transportadas em simultâneo
2	238	228	67	1
1	238	236	68	2

Finalmente comparando os melhores resultados para a chegada de ordens de fabrico de 3 em 3 minutos pode-se verificar que a melhor solução corresponde à solução 2 (1 AGV), que além de verificar um número de AGVs mais baixo apresenta ainda uma melhor resposta às ordens de fabrico.

4.6 Conclusão final

Nesta capítulo foi apresentado um sistema de produção em que a flexibilidade era um dos pontos fracos do sistema, facto que se estava a reflectir na produtividade do sistema. Algumas soluções que poderão ser tomadas foram apresentadas, nas quais uma solução baseada em transportadores e AGVs foi comparada.

Através dos resultados obtidos pode-se afirmar que a introdução de um sistema de transporte baseado em AGVs permitiu o aumento da flexibilidade, facto que se reflectiu num aumento da produtividade do sistema e uma redução do espaço ocupado quando comparado com uma duplicação do sistema. Ficou assim demonstrado que os AGVs são uma ferramenta poderosa para

a indústria do presente e futuro, onde a exigência de constantes adaptações aos pedidos dos clientes leva a necessidade de existir uma grande flexibilidade no processo produtivo.

Capítulo 5

Robot Factory

A Faculdade de Engenharia, mais propriamente o núcleo de robótica, pretende realizar um concurso em que será disponibilizado um simulador com um ambiente físico muito próximo do real, no qual um ambiente industrial (postos de trabalho, entrada e saída do sistema) estará implementado. Neste contexto, será proposto aos concorrentes que desenvolvam um controlo eficaz dos veículos tendo como objectivo final a produção do maior número de peças num intervalo de tempo definido. Depois de validado o sistema de controlo, no simulador, os concorrentes passaram para um sistema de produção real construído à escala, que será idêntico ao construído em simulação.

Assim para o trabalho aqui apresentado foi proposta a construção de um ambiente industrial, utilizando para isso um software de simulação, bem como a realização de uma aplicação de demonstração no qual os veículos iriam ao ponto de entrada do sistema e iriam encaminhar as peças para as respectivas estações de trabalho e conseqüente saída.

Neste capítulo será então apresentado o ambiente industrial bem como as funções básicas de movimento dos veículos implementadas.

5.1 Apresentação do ambiente industrial desenvolvido

O *layout* fabril pensado para este concurso consiste, e tal como referido, na definição de postos de trabalho, num ponto de entrada e outro de saída do sistema de produção. Tendo então em conta o nível de dificuldade pretendido para o sistema foi acordado que o sistema de produção contaria com 4 estações de trabalho, dois *AGVs*, um ponto de entrada e outro de saída (ver Anexo A).

Em relação à localização no mundo dos veículos iria ser dada a possibilidade aos concorrentes de escolher entre recorrer a trajectórias definidas no chão de fábrica ou recorrer a outro método de localização [36, 37].

Para a modelação do ambiente fabril e dos respectivos veículos foi seleccionada uma ferramenta desenvolvida na Faculdade de Engenharia que irá ser apresentada no capítulo que segue. Esta ferramenta foi seleccionada devido ao facto de se tratar de um ferramenta que permite

a implementação de ambientes que se aproximam bastante do real e ao mesmo tempo, o facto de ter sido desenvolvida na casa, a sua utilização torna-se uma forma de a promover.

5.2 Ambiente de desenvolvimento SimTwo

Neste sub capítulo a ferramenta de simulação SimTwo utilizada para a modelação do ambiente industrial definido bem como para os veículos responsáveis pelo transporte das peças é apresentado. O SimTwo é uma ferramenta de simulação desenvolvida pelo Prof. Dr. Paulo Costa [38], tendo sido criada com o objectivo de possibilitar a realização de simulações de sistemas robóticos nomeadamente verificar o seu comportamento tendo em conta diferentes estratégias de controlo e de decisão, permitindo reduzir o tempo de implementação dos algoritmos de controlo bem como poupar o veiculo real a acidentes indesejados.

Esta aplicação permite observar o comportamento em tempo real do robô (posição, movimento e trajectória) através de uma interface gráfica 3D, auxiliando assim na detecção e correcção de possíveis erros no algoritmo de decisão e controlo, permitindo com isso aperfeiçoar o controlo desenvolvido.

5.2.1 Descrição geral

A plataforma de simulação SimTwo é um software de simulação realista desenhado com o intuito de testar o comportamento de variados robôs (robôs móveis, manipuladores industriais, humanóides entre outros). Aqui os robôs são construídos tendo em conta a sua dinâmica podendo ser divididos em duas partes: uma referente ao seu realismo quando considerado apenas como um corpo rígido, sendo para tal utilizada a biblioteca ODE (*Open Dynamic Engine*) [39]. Uma segunda parte que se prende com o realismo relativo ao comportamento e dimensionamento dos motores eléctricos e seus controladores.

A biblioteca ODE foi desenvolvida com o objectivo de aproximar o comportamento virtual de uma estrutura articulada com o respectivo comportamento real. Pressupõe dois níveis de simulação, o nível da dinâmica no qual as propriedades dinâmicas das estruturas (massa e velocidade) são tratados, e o nível das colisões que se ocupa com a forma da estrutura.

Para a concepção de estruturas articuladas ou rodas, é necessária a definição de eixos que pressuporem um sistema de accionamento. A cada eixo está associado um motor de corrente continua com caixa redutora, um controlador do tipo PID para posição ou velocidade e um *encoder* óptico. De facto neste software denota-se uma especial preocupação na modelação dos motores uma vez que tem em conta as suas não linearidades, tais como limitação de correntes, atrito de Coulomb entre outros [40].

O SimTwo disponibiliza ao seu utilizadores dois níveis de controlo do robô. Um primeiro sistema de controlo de mais baixo nível, com um tempo de ciclo de 10 ms, responsável pelo controlo dos motores. Estes controladores já se encontram definidos, podendo apenas ser efectuado alguns ajustes de parâmetros. Um segundo nível de controlo, este com um tempo de ciclo por defeito de 40 ms, onde é permitido aos utilizadores desenvolver algoritmos mais

complexos que afectam o comportamento das estruturas definidas. Este controlo de mais alto nível poderá ser implementado de duas formas. Uma recorrendo a uma aplicação externa ao SimTwo que comunicará com este através da porta série ou do protocolo de comunicação UDP (*User Datagram Protocol*) ou recorrendo a uma linguagem de programação baseada na biblioteca Pascal Script [41] editável e compilável no próprio simulador [40].

5.2.2 Construção do ambiente de simulação

Para a construção do ambiente de simulação no SimTwo é necessário recorrer a ficheiros .xml. Nestes ficheiros são introduzidas as classes ou secções, que definem no caso concreto, as estações de trabalho, os robôs (*AGVs*), as paredes e as trajectórias pintadas no pavimento. Estes ficheiros são lidos no instante de arranque da aplicação. De seguida são apresentadas as secções predefinidas no simulador.

Tabela 5.1: Secções abrangidas pelo simulador SimTwo

Classes	Descrição
<i>Robot</i>	Definição da estrutura do robô criada a partir de objectos básicos
<i>Obstacles</i>	Definição das estruturas inamovíveis (Obstáculos)
<i>Things</i>	Definição de corpos dinâmicos
<i>Track</i>	Definição de marcadores no pavimento

Todas as secções acima referidas são construídas a partir de objectos básicos, sendo que apenas a classe robô é controlável na plataforma de simulação. Focando-nos apenas agora nesta classe, e tendo em conta a sua estrutura global mais complexa em relação às restantes, serão apresentadas agora as estruturas disponíveis para a sua configuração. Assim quando desejamos construir um robô real temos de nos preocupar com a sua estrutura física bem como sensores e articulações que permitem a sua interacção com o ambiente que o rodeia. No simulador SimTwo estas estruturas encontram-se definidas em grupos distintos [38].

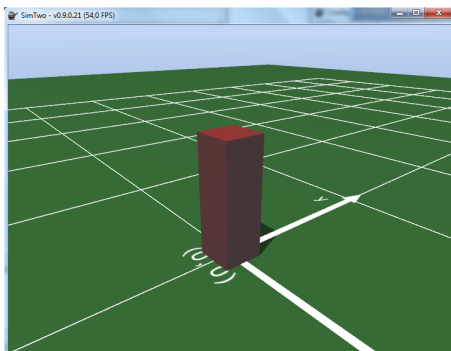
Começando pela estrutura física do robô, e tal como já mencionada, é realizada recorrendo a objectos básicos, denominados no SimTwo de *solids*. Estes objectos, no SimTwo, possuem características ajustáveis, tais como forma, massa, dimensão permitindo assim aproximar a sua dinâmica à dinâmica de um corpo rígido real. Nas tabelas 5.2 e 5.3 é exibida a biblioteca de objectos básicos disponibilizados e suas respectivas propriedades. Na figura 5.1 encontra-se um pequeno exemplo de como esta modelação é realizada.

Tabela 5.2: Objectos básicos disponíveis no SimTwo

Objecto	Descrição
<i>Cuboid</i>	Paralelepípedo
<i>Cylinder</i>	Cilindro
<i>Belt</i>	Tapete Rolante
<i>Sphere</i>	Esfera

Tabela 5.3: Propriedades dos objectos básicos

Propriedade	Descrição
<i>Id</i>	Identificação do objecto
<i>Mass</i>	Definição da massa(Kg) do objecto
<i>Size</i>	Dimensões segundo x,y e z em metros
<i>Pos</i>	Posição no referencial mundo segundo x,y e z
<i>Rot deg</i>	Posição no referencial mundo segundo x,y e z
<i>Color rgb</i>	Posição no referencial mundo segundo x,y e z



```

<solids>
  <cuboid>
    <ID value='robot2' />
    <mass value='100' />
    <size x='0.2' y='0.2' z='0.6' />
    <pos x='0' y='0' z='0.3' />
    <rot_deg x='0' y='0' z='0' />
    <color_rgb r='100' g='0' b='0' />
  </cuboid>
</solids>

```

Figura 5.1: Solids - Exemplo SimTwo

Para a modelação das articulações e sensores, o SimTwo disponibiliza aos seus utilizadores uma biblioteca de objectos mais complexos (tabela 5.4), permitindo assim adicionar um nível de complexidade mais elevada ao robô virtual.

Tabela 5.4: Objectos de complexidade mais elevada disponíveis no SimTwo

Propriedade	Descrição
<i>Sensors</i>	Sensores
<i>Articulations</i>	Articulações
<i>wheels</i>	Rodas
<i>Shells</i>	Superfícies

As articulações e rodas definem eixos articulados entre corpos e permitem ao utilizador actuar sobre o comportamento do robô. As superfícies foram desenvolvidas com o intuito de colmatar uma limitação imposta pela biblioteca ODE, uma vez que esta não implementa colisões entre cilindros. Finalmente são disponibilizados sensores que poderão ser adicionados ao robô, apresentando-se apenas disponíveis sensores de infravermelhos. De seguida são apresentadas as propriedades disponíveis nos objectos mais complexos (tabela 5.5).

Realizada uma pequena introdução à ferramenta de simulação utilizada, nas figuras 5.2 e 5.3 é apresentado, respectivamente, o AGV e o ambiente industrial modelados em SimTwo.

Como se pode observar, figura 5.2, no canto superior direito está colocado um tapete de transporte, que funciona como entrada do sistema. Os AGVs irão receber a peça na entrada do sistema e reencaminhar para uma das estações disponíveis. O processo de selecção da

Tabela 5.5: Propriedades dos objectos de complexidade mais elevada disponíveis no SimTwo

Objecto	Componente	Propriedade	Descrição
<i>Articulations/wheels</i>	Motor	r_i , k_i , v_{max} , i_{max} , active	Parâmetros do motor;
<i>Articulations/wheels</i>	Encoder	ppr, mean, stdev	Parâmetros do codificador óptico;
<i>Articulations/wheels</i>	Friction	B_v , f_c , coulomblimit	Parâmetros de fricção actuadores;
<i>Articulations/wheels</i>	Gear	ratio	Relação da caixa redutora;
<i>Articulations/wheels</i>	Controller	mode, k_p , k_i , k_d , k_r , active	Definição dos Parâmetros do Controlador;
<i>Articulations/wheels</i>	Axis	x,y,z	Parâmetros eixo das rodas;
<i>Articulations/Sensors/Shells</i>	Pos	x,y,z	Posição no referencial mundo;
<i>Articulations</i>	Connect	B1,B2	Parâmetro de ligação entre dois objectos;
<i>Articulations</i>	Type	value	Define o tipo de articulação;
<i>Articulations</i>	Limits	Min,Max	Define limites da articulação;
<i>Sensors</i>	Beam	length, width, initial width, final width	Parâmetros dos sensores;
<i>Shells/Sensors</i>	Rot deg	x,y,z	Rotação em graus
<i>Shells/Sensors</i>	Color rgb	r,g,b	Define a cor na base RGB(8 bits);
<i>whells</i>	-	Omni	Define a roda como omnidireccional;
<i>whells</i>	Tyre	Mass, radius, width, centerdist	Parâmetros da roda;

estação de trabalho ficará a cargo dos correntes, uma vez que diferentes estratégias poderão ser seleccionadas, com implicação directa na eficácia do controlo. Depois de as peças serem recebidas nas estações de trabalho será simulada a ocorrência de uma operação (simulada através do accionamento de um *timer*). Depois da operação concluída a estação de trabalho requisitará um AGV para o transporte da peça para o tapete de transporte de saída (canto inferior esquerdo da figura) ou para outro posto de trabalho. Pode-se ainda verificar a existência de trajectórias definidas no chão de fabrica (a verde na figura 5.2). Os concorrentes poderão utilizar as linhas para percorrer este percurso, sendo que a posição real do robô (disponibilizada pelo SimTwo) está sempre disponível para comparação e aferição do controlador. No entanto estarão restritos a estes percursos, o que introduz atrasos no transporte quando comparado com trajectórias mais flexíveis.

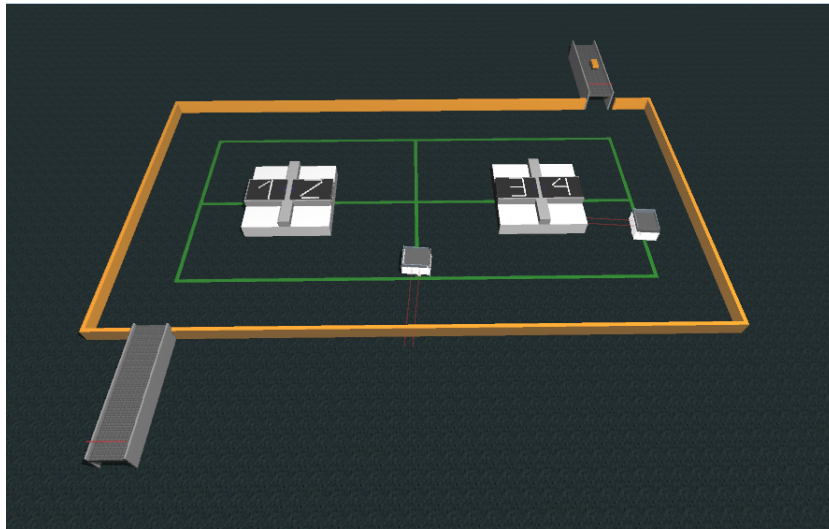


Figura 5.2: Ambiente industrial modelado em SimTwo



Figura 5.3: AGV diferencial construído em SimTwo

Na figura 5.3, e tal como já foi referido, o AGV construído é apresentado. De notar trata-se de um veículo diferencial, tendo sido adicionado 2 sensores laterais de infravermelhos. O número de sensores poderá ser incrementado, sendo a sua integração no algoritmo de controlo, apresentado mais à frente, facilmente realizado. O AGV aqui construído pretende-se que seja o mais próximo do real, em construção, nomeadamente dimensões, parâmetros dos motores, etc.

Para o leitor tomar uma ideia mais real daquilo que se passa no ambiente de produção, as figuras 5.4 e 5.5, ilustram respectivamente o processo de carga de peças no AGV na entrada do sistema e da descarga das peças nas estações de trabalho.

Assim na figura 5.4, o processo de carga de uma peça no AGV é apresentado. O AGV quando solicitado, dirige-se à entrada do sistema para receber uma nova peça.

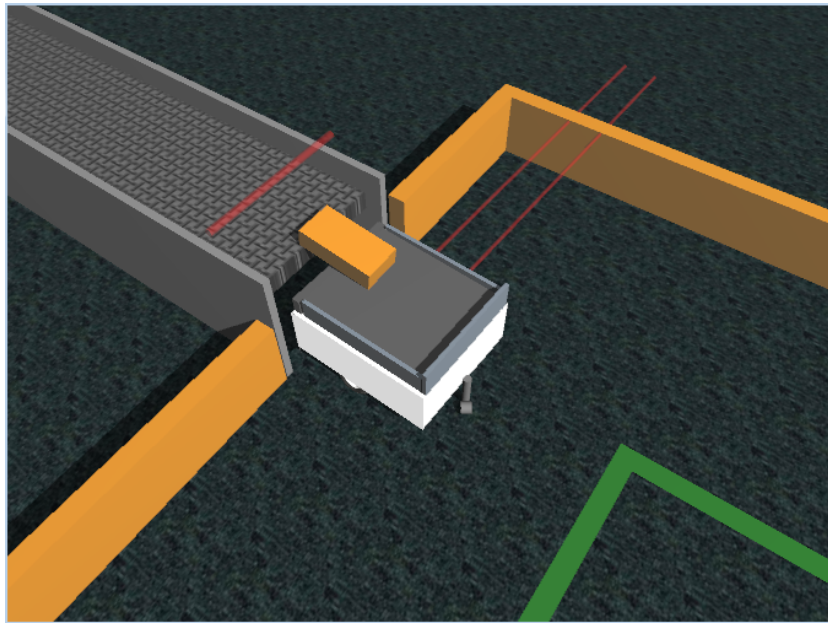


Figura 5.4: Processo de carregamento da peça no AGV - Entrada do sistema

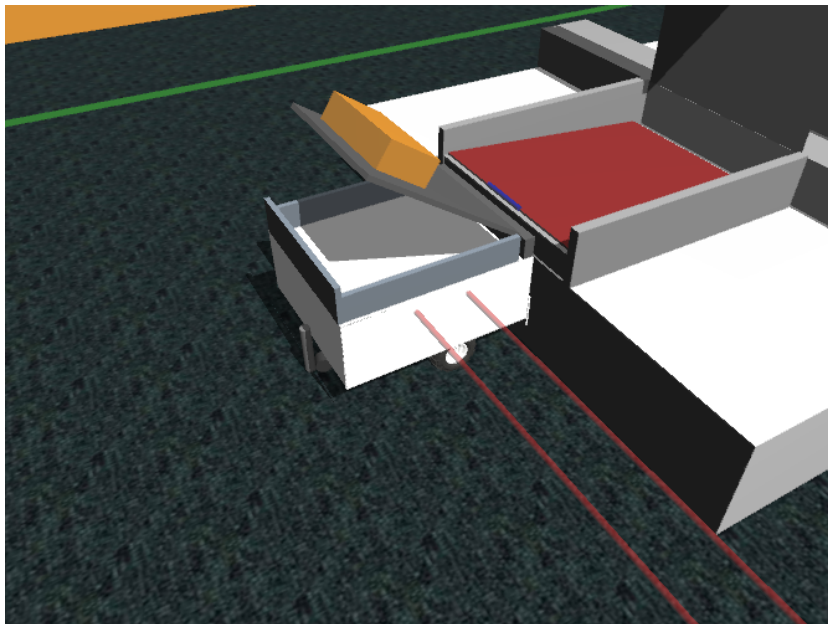


Figura 5.5: Processo de descarga da peça na estação de trabalho.

Na figura 5.5 é apresentado o processo de descarga da peça, por parte do AGV, na estação de trabalho, tendo o processo de produção sido descrito anteriormente.

5.3 Métodos de localização dos AGVs no referencial mundo

Um dos aspectos mais importante para a navegação de robô moveis, refere-se à sua auto-localização. A posição de um veículo é definido pela sua posição cartesiana e orientação

($X=[x,y,\theta]$) segundo um referencial global, usualmente denominado de referencial mundo.

Para obter a sua posição cartesiana e orientação será necessário recorrer a métodos de localização relativa e absoluta [42].

Os métodos de localização relativa asseguram que o veículo se encontra localizado em cada instante. Neste método a estimativa da posição actual é obtida a partir da posição e orientação inicialmente conhecida. Para o caso em que o veículo recorre a rodas para a sua deslocação é comum o uso de encoders acoplados a cada um dos eixos de rotação das rodas, técnica conhecida como odometria. Outros sensores também usualmente utilizados são sensores de dopler, giróscopios e acelerómetros. Nesta secção será apenas abordado o método baseado em odometria.

Já os métodos de localização absoluta permitem obter a posição global do robô sem recorrer a medidas obtidas em instante anteriores. Este método engloba sistema de localização baseado em trilateração ou triangulação com balizas, à utilização de marcadores artificiais e ou naturais, bússola magnética e correspondência de mapas [36, 37, 43]. De salientar que mesmo estas técnicas são imprecisas devido aos ruídos presentes nas medições dos sensores.

De forma a obter a posição do robô e de forma a que o erro seja minimizado, será interessante filtrar as informações obtidas a partir dos sensores, recorrendo para isso ao Filtro de Kalman.

5.3.1 Método de localização relativa - Odometria

Tal como já foi referido, a utilização da odometria como método de localização relativo em veículos é amplamente utilizada. Este método consiste na determinação da posição e orientação do robô a partir da integração dos deslocamentos incrementais das rodas motrizes. Por outras palavras este método recorre à mediação da rotação de cada roda, utilizando essa informação para obter a velocidade, orientação e posição do robô.

Assim existe a necessidade de dotar o veículo com sensores que possibilitem medir a rotação das rodas, entre os quais os mais utilizados são: os encoders, potenciômetros e os resolvers.

No entanto esta técnica está sujeita a erros prejudicando assim a estimativa da posição do robô. Os erros de odometria podem ser classificados em sistemáticos e não-sistemáticos.

Os erros sistemáticos são aqueles causados por incertezas nos parâmetros do modelo cinemático do robô, entre os quais se destacam:

- Rodas com diferentes diâmetros;
- Distância real entre rodas;
- Rodas mal alinhadas;
- Deformação dos pneus com a carga;

Os erros não-sistemáticos têm origem, entre outras causas, num mau contacto entre o piso as rodas do veículo. Fontes de erro não-sistemáticos:

- Resolução finita dos encoders;
- Pavimentos irregulares ou com objectos inesperados;
- Derrapagem das rodas(piso escorregadio, aceleração/travagem excessiva...);
- Discretização e aproximação introduzidas nas equações utilizadas;

Segundo [44] os erros de odometria podem ser representados por uma região elíptica (região de incerteza que irá aumentar com a distância percorrida pelo veículo) em torno da posição actual do robô (ver figura 5.6).

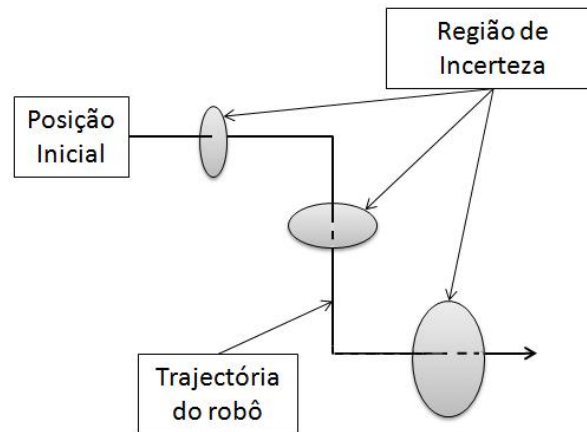


Figura 5.6: Região de incerteza quando utilizada para a estimativa da posição a odometria

Assim a odometria deverá ser calibrada por forma a minimizar os erros sistemáticos, sendo necessário a adição ao sistema de outros métodos de localização, nomeadamente métodos de localização absoluta.

Na figura 5.7 é apresentado o modelo cinemático do veículo de duas rodas com accionamento diferencial. O referencial x e y são as coordenadas base em relação a um referencial cartesiano. Neste contexto, as equações do modelo cinemático do robô diferencial poderão ser retiradas.

$$\frac{d}{dt} \begin{bmatrix} x_c(t) \\ y_c(t) \\ \theta_c(t) \end{bmatrix} = \begin{bmatrix} v(t) * \cos(\theta(t)) \\ v(t) * \sin(\theta(t)) \\ \omega(t) \end{bmatrix} \quad (5.1)$$

com,

$$v(t) = \frac{v_d(t) + v_e(t)}{2} \quad (5.2)$$

e

$$w(t) = \frac{v_d(t) - v_e(t)}{b} \quad (5.3)$$

Em que $v_d(t)$ e $v_e(t)$, corresponde à velocidade de cada roda e b é a distância entre o ponto de contacto de cada roda com o piso.

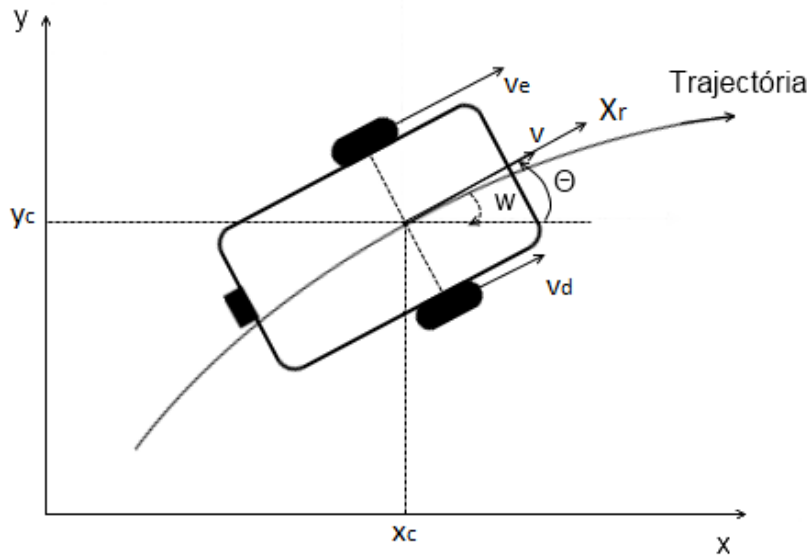


Figura 5.7: Robô Diferencial

O modelo da odometria permite assim estimar através da informação, obtida periodicamente (k) pelos *encoders* de cada uma das rodas, impulsos $N_{d/e}$, a pose e orientação do veículo. Para obter o deslocamento linear de cada roda, $d_d(k)$ e $d_e(k)$, no período de amostragem k , é necessário relacionar esse deslocamento com os impulsos dos *encoders*, $N_{e/d}$. Assim o deslocamento de cada roda é dado por:

$$d_{d/e}(k) = c_m * N_{d/e}(k) \quad (5.4)$$

$$c_m = \frac{\pi D_n}{n C_e} \quad (5.5)$$

Onde c_m é o factor de conversão entre os impulsos dos *encoders* e o deslocamento linear da respectiva roda, D_n é o diâmetro das rodas, C_e é a resolução dos *encoders* e n é a relação da caixa redutora.

Após a discretização do modelo cinemático do robô a partir das diferenças centradas [43], a pose e orientação do robô pode ser obtida a partir da equação:

$$\begin{bmatrix} x_c(k+1) \\ y_c(k+1) \\ \theta(k+1) \end{bmatrix} = \begin{bmatrix} x_c(k) + d(k) * \cos(\theta(k) + \frac{\Delta\theta(k)}{2}) \\ y_c(k) + d(k) * \sin(\theta(k) + \frac{\Delta\theta(k)}{2}) \\ \theta(k) + \Delta\theta(k) \end{bmatrix} \quad (5.6)$$

com,

$$d(k) = \frac{d_d(k) + d_e(k)}{2} \quad (5.7)$$

e,

$$\Delta\theta(k) = \frac{d_d(k) - d_e(k)}{b} \quad (5.8)$$

O facto de se recorrer à discretização do modelo cinemático pelas diferenças centradas, tem por objectivo diminuir o erro introduzido pela discretização na equação 5.4.

Após abordadas as equações que determinam o modelo cinemático do robô em estudo, podemos concluir que o modelo da odometria ficará inteiramente definido após a calibração do raio das rodas e da distância entre rodas.

5.3.2 Calibração odometria

Para a calibração da odometria duas experiências foram realizadas. A primeira consiste em fazer o robô andar em linha recta, considerando neste caso que o diâmetro das rodas é igual, ou seja, foi aplicado o mesmo sinal de velocidade a ambas as rodas, percorrendo uma determinada distância e contar o número de impulsos de cada roda. De seguida recorrendo às equações 5.4, 5.5 e 5.7 poderá ser calculada uma estimativa do diâmetro de cada roda.

A segunda experiência, na qual pretendemos obter a distância entre o ponto de contacto com o chão de cada roda (b), aplicou-se um velocidade igual em módulo, mas de sinais contrários às rodas do robô de forma a que o robô rode sobre si próprio. Depois de o número de impulsos de cada roda ser obtido recorre-se à equação 5.8 e obtém-se o valor de b.

Após efectuadas as experiências acima referidas, obteve-se os seguintes resultados:

Tabela 5.6: Valor dos Parâmetros obtidos - Calibração odometria

Parâmetro	Valor obtido(cm)
D_n	9,64
b	24,7

De salientar que apesar destes parâmetros do sistema de odometria terem sido obtidos através de simulação, foram mais tarde confirmados após algumas experiências realizadas com o AGV real, nomeadamente a realização de rotações e de percursos em linha recta.

5.3.3 Modelo estocástico do erro da odometria

Como referido anteriormente, a odometria está sujeita à perturbação de erros aleatórios ou não sistemáticos, que afectam o sistema de localização relativa do robô. Será então importante quantificar a certeza da pose calculada, daí a necessidade de criar um modelo estocástico do erro da odometria.

Recorreu-se então ao modelo apresentado em [45], que apresenta a seguinte forma:

$$X(k+1) = F(X(k), d, \theta) + W(k) \quad (5.9)$$

Alternativamente,

$$\begin{bmatrix} x_c(k+1) \\ y_c(k+1) \\ \theta(k+1) \end{bmatrix} = \begin{bmatrix} x_c(k) + d(k) * \cos(\theta(k) + \frac{\Delta\theta(k)}{2}) \\ y_c(k) + d(k) * \sin(\theta(k) + \frac{\Delta\theta(k)}{2}) \\ \theta(k) + \Delta\theta(k) \end{bmatrix} + W(k) \quad (5.10)$$

Podemos ver que a equação 5.9, é nada mais que as equação 5.6 adicionada por um ruído $W(t)$ a cada período de amostragem. Este ruído é caracterizado por uma distribuição normal de média nula e uma matriz de covariância, $Q(k)$, definida por:

$$E[W(k)] = 0 \quad (5.11)$$

$$E[W(k) * W(k)^T] = Q(k) \quad (5.12)$$

Em que $E[.]$ representa o valor esperado.

Admitindo que os ruídos de odometria que afectam a pose do robô e respectiva orientação são independentes entre si, a matriz $Q(t)$ é igual:

$$Q(k) = \begin{bmatrix} Q11 & 0 & 0 \\ 0 & Q22 & 0 \\ 0 & 0 & Q33 \end{bmatrix} \quad (5.13)$$

Em que os termos $Q11$, $Q22$ e $Q33$ são aproximados por:

$$Q11 = K_{uu} |d * \cos(\theta(k))| \quad (5.14)$$

$$Q22 = K_{uu} |d * \sin(\theta(k))| \quad (5.15)$$

$$Q33 = K_{u\theta} |d| + K_{\theta\theta} |\Delta\theta| \quad (5.16)$$

Em que K_{uu} é o coeficiente que relaciona o erro de odometria ao longo de x_r em relação a d , $K_{u\theta}$ é o coeficiente que relaciona o erro da odometria segundo x_r em relação a θ , e $K_{\theta\theta}$ é o coeficiente que relaciona o erro da odometria segundo θ em relação a $\Delta\theta$.

A matriz $P(k)$ representa a matriz covariância do erro da pose:

$$P(k) = \begin{bmatrix} Var(x(k)) & Cov(x(k), y(k)) & Cov(x(k), \theta) \\ Cov(x(k), y(k)) & Var(y(k)) & Cov(y(k), \theta) \\ Cov(x(k), \theta(k)) & Cov(y(k), \theta(k)) & Var(\theta(k)) \end{bmatrix} \quad (5.17)$$

Onde Var e Cov significa respectivamente variância e covariância. $P(k+1)$ é obtido a partir da seguinte expressão:

$$P(k+1) = \frac{dF(k)}{dX(k)} * P(k) * \left(\frac{dF(k)}{dX(k)}\right)^T + Q(k) \quad (5.18)$$

Em que $d(F(t))/dX(t)$ corresponde à seguinte matriz:

$$\frac{dF(k)}{dX(k)} = \begin{bmatrix} 1 & 0 & -d * \text{sen}(\theta(k)) + \frac{\Delta\theta(k)}{2} \\ 0 & 1 & d * \text{sen}(\theta(k)) + \frac{\Delta\theta(k)}{2} \\ 0 & 0 & 1 \end{bmatrix} \quad (5.19)$$

A equação 5.18 corresponde à propagação do erro da odometria a partir da linearização de F em relação a X no instante de tempo k, adicionada por uma incerteza em x, y, θ dependente de d e $\Delta\theta$.

Com o objectivo de observar os erros aleatórios no sistema de odometria, e por consequência a determinação dos parâmetros do modelo do erro acima descrito, foram realizadas duas experiências. A primeira consiste em percorrer uma distância razoável segundo o referencial x e registar o erro da odometria obtido, através da comparação entre a posição final real (disponibilizada pelo SimTwo) e a posição obtida através da odometria. Esta experiência foi realizada 20 vezes de forma a observar a dispersão dos resultados. A partir destes resultados, a média das distâncias percorridas ($\text{med}(x_{\text{total}})$), a variância da medida x ($\text{Var}(x)$) e θ ($\text{Var}(\theta)$) poderão ser obtidas.

A segunda experiência consiste em realizar uma rotação e registar o erro de odometria obtido. Mais uma vez 20 ensaios foram realizados.

Assim através dos resultados obtidos da experiência um os valores de K_{uu} e $K_{u\theta}$ poderão ser obtidos. $k_{\theta\theta}$ é obtido através da experiência número dois.

$$K_{uu} = \frac{\text{Var}(x)}{\text{med}(x_{\text{total}})} \quad (5.20)$$

$$K_{u\theta} = \frac{\text{Var}(\theta)}{\text{med}(x_{\text{total}})} \quad (5.21)$$

$$K_{\theta\theta} = \frac{\text{Var}(\theta)}{\text{med}(\theta_{\text{total}})} \quad (5.22)$$

Os valores obtidos para cada um dos parâmetros são apresentados na tabela 5.7.

Tabela 5.7: Valor dos Parâmetros obtidos - Modelo estocástico do erro da odometria

Parâmetro	Valor obtido(cm)
K_{uu}	$4.100E - 05 \text{ m}^2/\text{m}$
$K_{u\theta}$	$0.077 \text{ rad}^2/\text{m}$
$K_{\theta\theta}$	$0.003 \text{ rad}^2/\text{rad}$

Os parâmetros aqui obtidos são característicos do ambiente do robô (nomeadamente o piso, que no ambiente SimTwo é um piso duro sem irregularidades) em que os testes foram realizados. Se as experiências fossem realizadas num piso rugoso, ou escorregadio, os valores de k_{uu} , $k_{u\theta}$ e $k_{\theta\theta}$ seriam maiores, uma vez que estes traduzem a confiança que temos no sistema de odometria.

Para a obtenção destes resultados, foi ainda adicionado ruído à informação obtida a partir dos *encoders*, de forma a aproximar a simulação o mais próximo possível do real.

5.3.4 Método de Localização Absoluta

Como o veículo real ainda se encontra numa fase de construção e controlo, o sistema de localização absoluta foi simulado, em SimTwo, sendo que a sua adaptação para um sistema de localização real se efectua de forma directa.

Como referido anteriormente o SimTwo, neste momento, apenas disponibiliza ao seus utilizadores sensores de infravermelhos, estando no entanto previsto num futuro próximo a disponibilização de sonares, lasers e sensores de linhas brancas no chão. Desta forma o método de localização implementado, em simulação, consiste na medição da distância nas laterais do robô através de dois sensores de infravermelhos. A partir da informação disponibilizada por cada sensor, a distância e orientação em relação à parede/obstáculo pode ser obtida.

De referir que o método aqui apresentado e ilustrado na figura 5.8 apenas permite definir a coordenada perpendicular à parede lateral.

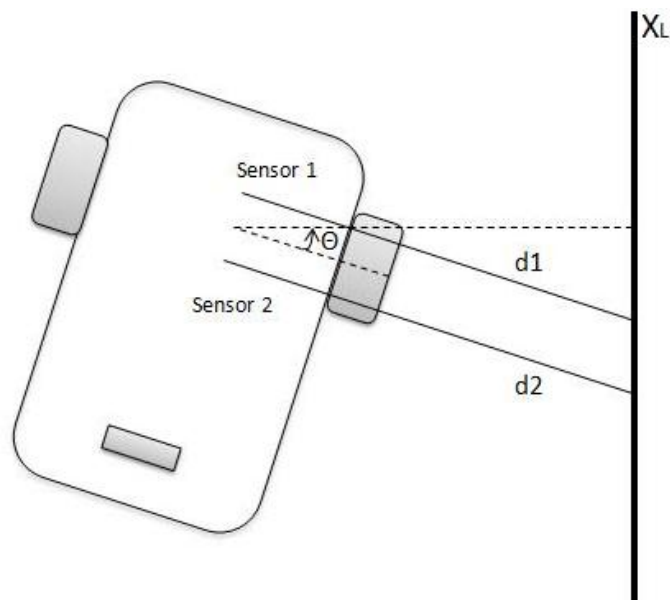


Figura 5.8: Esquema representativo da medição da distância do robô a uma parede lateral

Assim a orientação do robô (θ) em relação à parede é dada por:

$$\theta = \arctan\left(\frac{d_1 - d_2}{2L_s}\right) \quad (5.23)$$

Em que $2L_s$ é a distância entre os sensores. A distância (x) do robô à parede obtém-se a partir de:

$$x = x_L - \frac{d_1 + d_2}{2} * \cos(\theta) \quad (5.24)$$

5.3.5 Fusão Sensorial

Como se pode verificar, neste momento, dois métodos de localização (um relativo e outro absoluto) foram implementados. No método de localização relativa (odometria) e tal como já referido, as medidas da posição apresentam um erro cumulativo, logo será necessário introduzir informação no sistema proveniente de sensores que recolham informação sobre o ambiente (método de localização absoluta) em que o robô se encontra para compensar a informação perdida.

Surge então a necessidade de fundir a informação entre estes dois métodos, de forma a estimar com o menor erro a posição real do robô.

Para atingir este objectivo, recorreu-se à implementação de um EKF (*Extended Kalman Filter*) discreto. O Filtro de Kalman Estendido é um algoritmo recursivo utilizado na previsão dos estados de um sistema dinâmico não-linear. Para isso, o algoritmo baseia-se na medição da “qualidade” dos dados observados a partir de uma matriz de covariâncias, garantindo assim a convergência para um solução óptima. Partindo de um conjunto de equações matemáticas o estado do robô é obtido, de forma a minimizar a matriz covariância do erro estimado [40]. A opção pela escolha deste método deve-se à não linearidade do modelo cinemático do robô anteriormente apresentado e devido ao facto de as distribuições de probabilidade do erro dos sensores utilizados poderem ser aproximadas por distribuições gaussianas.

O algoritmo do EKF baseia-se em duas fases distintas: estimação e actualização. Durante a fase de estimação, é utilizado o modelo cinemático do sistema e o estado anterior para estimar o estado actual do robô. Já na fase de actualização é utilizada a informação das medidas existentes de forma a corrigir essa estimação, alcançando assim uma nova estimativa do estado mais precisa [40].

Assim, durante a fase de estimação, equações 5.25 e 5.26, será utilizado os sensores do sistema de odometria (*encoders*) para obter uma estimação do estado actual do robô. O erro da estimação do estado baseado neste método de localização irá aumentar gradualmente devido à integração do ruído dos sensores.

Durante a fase de actualização, equações 5.27, 5.32 e 5.33, será utilizado o método de localização absoluta, baseado nos sensores infravermelhos, permitindo assim diminuir o erro da estimação do estado actual.

Mais informação sobre o motivo da sua utilização bem como para uma explicação mais aprofundada dos seus conceitos, a leitura das referências bibliográficas [40, 46] é aconselhada.

Partindo então do modelo cinemático do robô, descritas no sub capítulo 5.3.1 e considerando que os sinais de controlo mudam somente nos instantes de amostragem, a estimação do estado no instante $t = t_K$, isto é, $X(k^-)$ é dada por:

$$X(k^-) = \begin{bmatrix} x(k) \\ y(k) \\ \theta(k) \end{bmatrix} = \begin{bmatrix} x(k) + d(k) * \cos(\theta(k) + \frac{\Delta\theta(k)}{2}) \\ y(k) + d(k) * \sin(\theta(k) + \frac{\Delta\theta(k)}{2}) \\ \theta(k) + \Delta\theta(k) \end{bmatrix} \quad (5.25)$$

De seguida é calculada a matriz covariância do estado $P(k^-)$. Esta matriz se nos lembrarmos foi definida a quando da realização do modelo estocástico do erro da odometria (equação 5.17).

$$P(k^-) = \phi^*(k) * P(k-1) * \phi^*(k)^T + Q(k) \quad (5.26)$$

Posteriormente, e já na fase de actualização, é calculado o ganho do filtro de Kalman:

$$K(k) = P(k^-) * H^*(k)^T * (H^*(k) * P(k^-) * H^*(k)^T + R(k))^{-1} \quad (5.27)$$

A matriz H^* , matriz das observações, e tendo em conta que o sistema de localização absoluta só permite actualizar a coordenada perpendicular à posição da parede é definida por:

- Para paredes verticais

$$H^*(k) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (5.28)$$

- Para paredes horizontais

$$H^*(k) = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (5.29)$$

$R(k)$ é matriz covariância do ruído das medidas obtidas através do sistema de localização absoluto. Assumindo que o ruído que afecta a medição de x, y e θ são independentes entre si, a matriz $R(k)$ é uma matriz diagonal. Para o cálculo dos valores da diagonal principal foi medida a variância das medidas, x e y e θ , obtidas tendo em conta posições diferentes do robô em relação à parede. Assim a matriz $R(k)$ é definida por:

- Para paredes verticais

$$R(k) = \begin{bmatrix} \text{Var}(x) & 0 \\ 0 & \text{Var}(\theta) \end{bmatrix} \quad (5.30)$$

- Para paredes horizontais

$$R(k) = \begin{bmatrix} \text{Var}(y) & 0 \\ 0 & \text{Var}(\theta) \end{bmatrix} \quad (5.31)$$

O estado do robô é então actualizado:

$$X(k) = X(k^-) + K(k) * (Z(k) - H^* * X(k^-)) \quad (5.32)$$

$Z(k)$ é o vector das medidas (posição(segundo x ou y) e orientação do robô), obtidas a partir dos sensores infravermelhos.

Finalmente é efectuada a actualização da covariância do estado:

$$P(k) = (I - K(k) * H^*(K)) * P(k^-) \quad (5.33)$$

5.3.6 Resultados obtidos - Métodos de Localização

Nesta secção serão apresentados os resultados, obtidos por simulação, referentes às trajectórias obtidas recorrendo apenas à odometria do veículo e à odometria mais sensores infravermelhos. De salientar que a informação dos sensores infravermelhos será fundida com a informação do sistema de odometria recorrendo ao filtro de Kalman estendido apresentado anteriormente. O veículo real não foi utilizado para a obtenção destes resultados uma vez, que ainda se encontra em fase de construção, não possuindo ainda sensores para que um método de localização absoluta possa ser implementado.

Para a obtenção destes resultados foi definida uma trajectória no chão de fabrica desenvolvido em SimTwo. Esta trajectória é apresentada na figura 5.9.

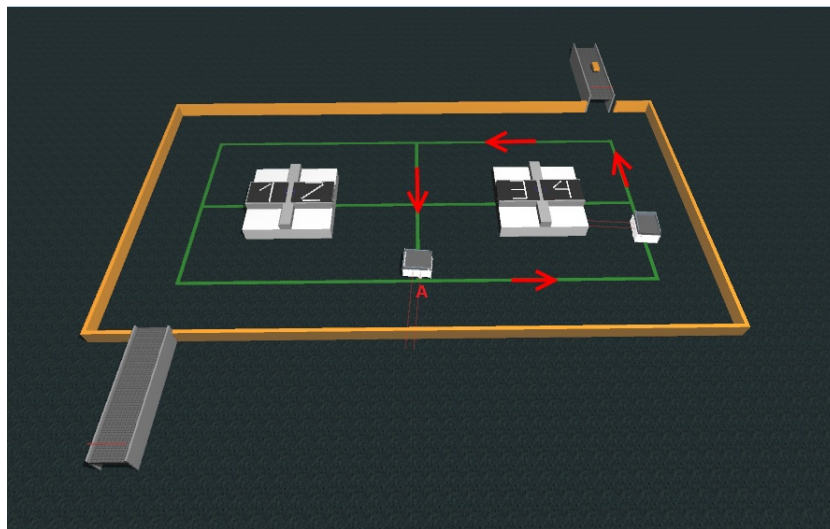


Figura 5.9: Percurso definido para o AGV

O AGV partirá então do ponto A(0,0.8) irá efectuar o percurso assinalado pelas setas a vermelho. O AGV percorrerá três vezes o circuito, parando no final no ponto A(0,0.8).

Tal como referido anteriormente, foi adicionado ruído aos sensores de odometria (*encoders*), sendo também neste caso adicionado aos sensores infravermelhos por forma a obter um comportamento em simulação mais próximo do real.

Na figura, 5.10 e 5.11 são então apresentados os resultados referentes à trajectória do AGV tendo em conta a utilização apenas do método de localização relativo (odometria).

Como podemos observar, e comparando as figuras 5.10 e 5.11, podemos ver que o método de odometria acumula um erro, de tal forma que a trajectória do veículo em cada volta se distância

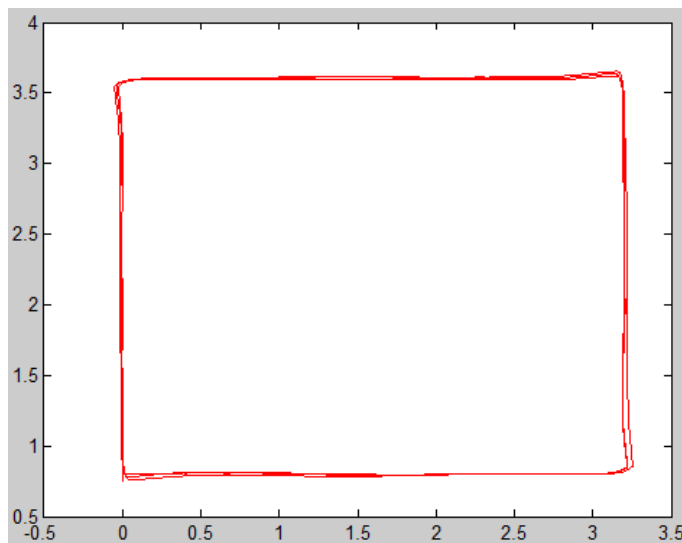


Figura 5.10: Trajetória do robô - localização recorrendo apenas ao sistema de odometria

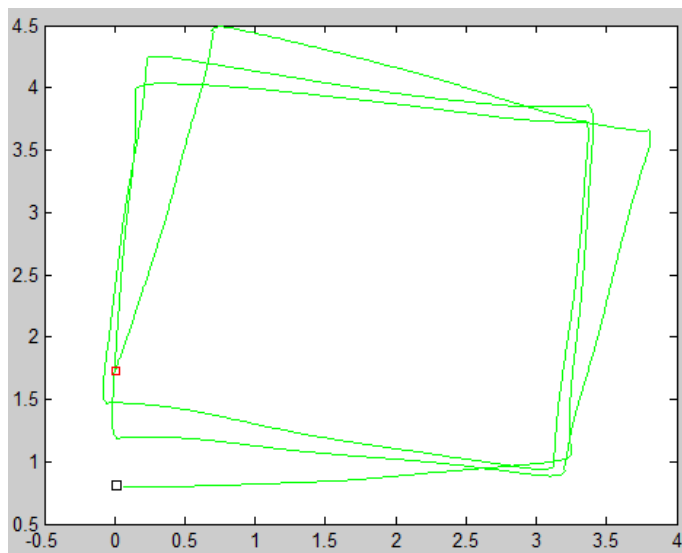


Figura 5.11: Trajetória do robô recorrendo apenas ao sistema de odometria Posição real

mais da trajetória pretendida. O quadrado preto e o vermelho, na figura 5.11, marca a posição inicial e final respectivamente.

Nas figuras 5.12 e 5.13 são apresentados os resultados obtidos para a fusão de informação do sistema de odometria e sensores infravermelhos.

O veículo irá corrigir a sua posição recorrendo à medição da distância às paredes laterais. Neste caso foram utilizadas as paredes superior, inferior e lateral direita às estações de trabalho 3 e 4. No entanto as paredes não poderão ser utilizadas como meio de localização durante as rotações bem, como em locais irregulares (como é o caso do tapete de transporte colocado na parede superior), devido aos sensores utilizados, estando então limitada a uma zona de cada parede a correcção da posição dos veículos. Neste contexto, pode-se verificar que durante o

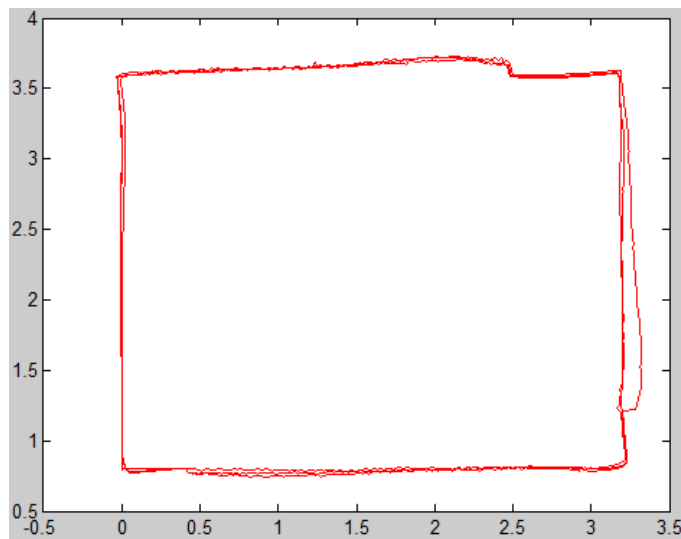


Figura 5.12: Trajectória do robô - Sistema de odometria mais sensores infravermelhos

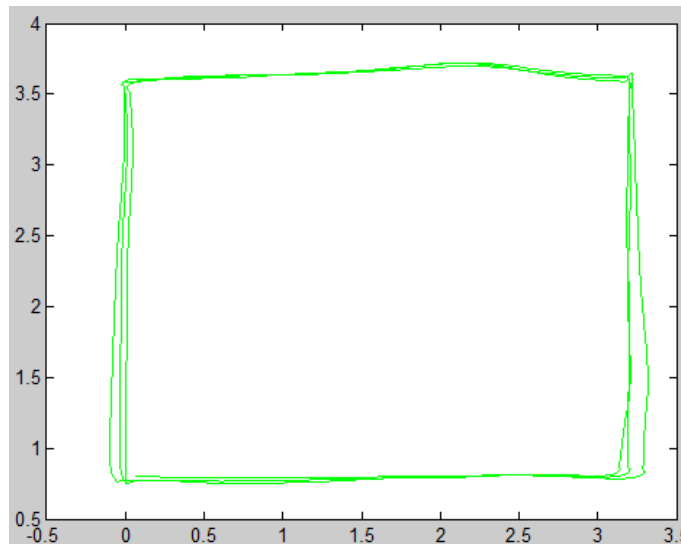


Figura 5.13: Trajectória do robô com a utilização do sistema odometria mais sensores infravermelhos- Posição real

lanço de trajectória restante (aresta esquerda do quadrado) o veículo se guiará apenas recorrendo à odometria. Será ainda importante lembrar mais uma vez que o veículo, devido ao método de localização utilizado, corrigirá a sua posição apenas no referencial perpendicular à parede.

Observando as figuras, 5.12 e 5.13 podemos verificar que o veículo já consegue seguir a trajectória pretendida, validando assim o sistema de localização implementado.

5.4 Aplicação externa de controlo

Para permitir a interação dos utilizadores com o ambiente industrial desenvolvido em SimTwo, e para o desenvolvimento dos algoritmos de decisão e controlo, foi desenvolvida uma

aplicação externa, em Lazarus (figura 5.14).

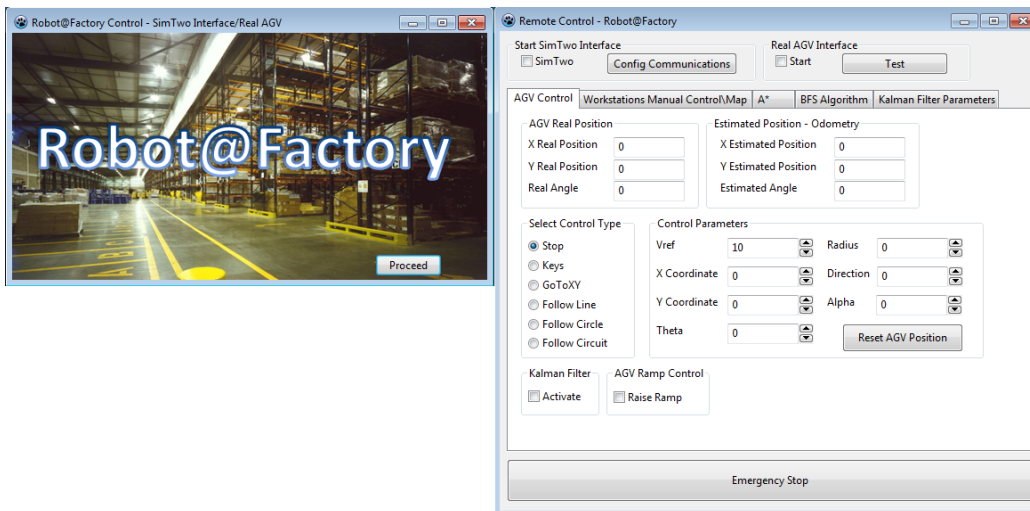


Figura 5.14: Aplicação de controlo remoto

O software Lazarus incorpora a linguagem *Free Pascal*, sendo facilmente integrado quer no sistema operativo Windows quer em Linux. Pelo facto da linguagem *Free Pascal* ser orientada a objectos, a construção de interfaces com o utilizador, o acesso a dispositivos e sistema periféricos torna-se bastante fácil de realizar. Desta forma, a aplicação de controlo e decisão foi desenvolvida tendo como base a programação orientada a eventos [40].

De seguida, na figura 5.15, a arquitectura do sistema desenvolvido é apresentada.

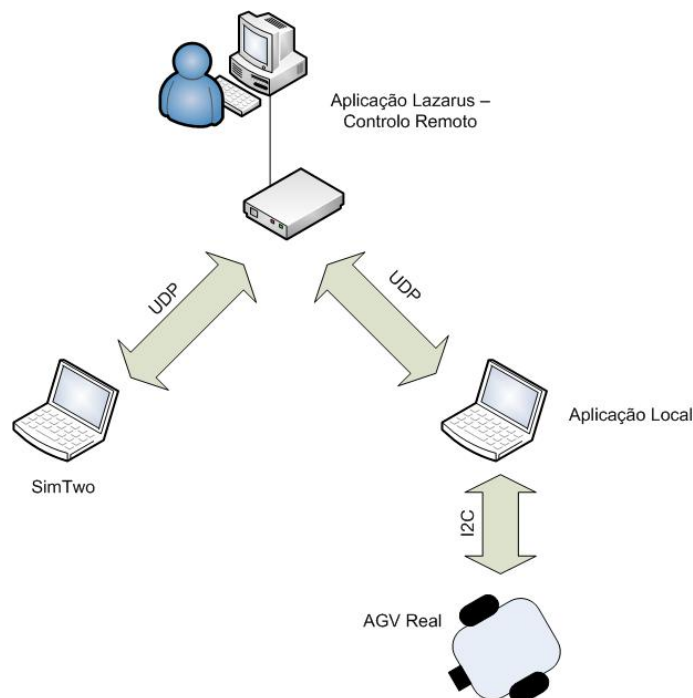


Figura 5.15: Arquitectura do sistema desenvolvido

Como se pode observar a interacção entre os diferentes componentes do sistema é realizado tendo em conta dois protocolos de comunicação distintos: UDP (*User Datagram Protocol*) e I2C.

Na próxima secção o protocolo de comunicação UDP será alvo de uma descrição mais detalhada. Em relação ao protocolo de comunicação I2C, e tendo em conta que o programa de controlo local responsável pela leitura e envio de comandos suportados pelo drive dos motores (a respectiva *datasheet* poderá ser encontrada em [47]) já se encontrar desenvolvido, figura 5.16, não foi alvo de estudo nesta dissertação. Foi apenas necessário integrar esta aplicação com a aplicação de controlo remoto, tendo sido utilizado para isso, e tal como apresentado na figura 5.15, o protocolo UDP.

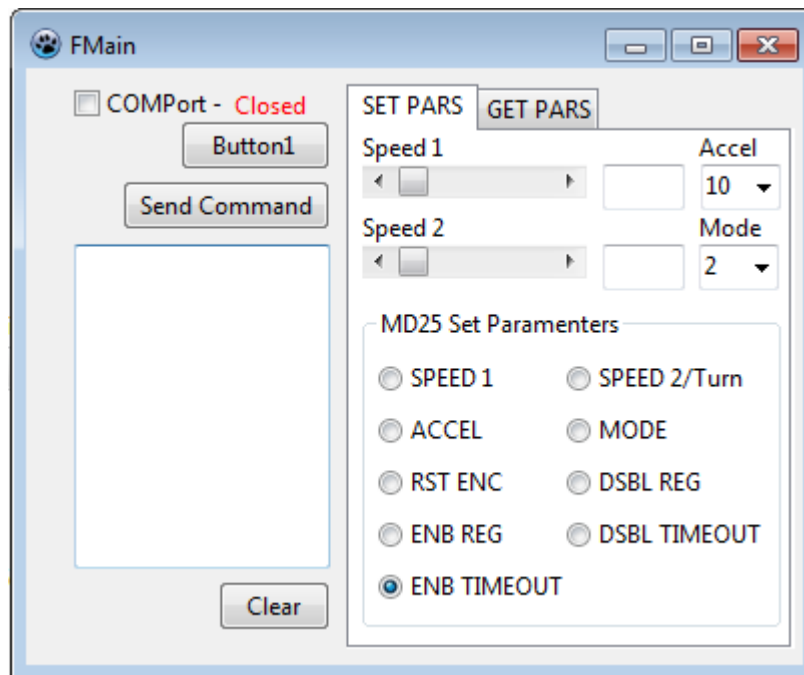


Figura 5.16: Aplicação local

5.4.1 Protocolo de comunicação UDP

Para a comunicação entre a plataforma de controlo e a plataforma de simulação SimTwo e entre a aplicação de controlo e o computador responsável pelo controlo local, localizado no robô real foi utilizado o Protocolo de comunicação UDP (*User Datagram Protocol*) [48]. Este protocolo é um protocolo não orientado à conexão não garantindo por isso que os pacotes cheguem ao seu destino e de forma ordenada, não se verificando o *handshaking* para os dados recebidos ou para troca de informações de controlo de fluxo. A integridade dos dados pode ser conferida pelo sistema de detecção de erros Checksum.

Contudo este protocolo é tipicamente o mais utilizado em aplicações de tempo real, sobretudo devido à sua velocidade de envio de pacotes de rede. Se tivermos em conta que num sistema de tempo real a 25hz (caso do SimTwo) são enviados 25 pacotes de dados pela rede por segundo verificamos que o facto de não haver garantia de entrega dos pacotes não representa uma situação

critica, uma vez que a transmissão/recepção de dados é realizada de 40 em 40 ms(25hz), na eventualidade de um pacote não ser entregue, o pacote seguinte chega num intervalo de tempo muito reduzido torna o sistema tolerante a falhas.

O protocolo de comunicação UDP permite a troca de dados entre sistemas *host*. O seu cabeçalho é extremamente simples contendo apenas os portos de origem(opcional) e destino, o comprimento da mensagem e o Checksum (opcional) (figura 5.17).

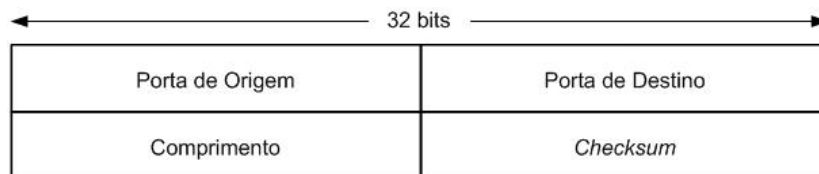


Figura 5.17: Cabeçalho da mensagem Protocolo UDP

5.4.2 Interação entre Aplicações

Tal como referido no início desta secção, foi desenvolvida uma aplicação de controlo remoto, responsável por controlar tanto o ambiente fabril desenvolvido em simTwo como também o AGV real construído.

As acções definidas nesta aplicação são condicionadas por eventos internos e externos. Os internos correspondem a um *timer* de 40 ms, que controla o tempo entre ciclos de controlo. Durante este ciclo algoritmos mais complexos, nomeadamente controlo de estações de trabalho, localização e navegação dos AGVs são executados. Esta aplicação, tendo então em conta os algoritmos definidos, envia para o SimTwo, caso esteja seleccionado este interface, a velocidade de referência de cada uma das rodas motrizes dos AGVs, bem como os sinais de controlo das 4 estações de trabalho definidas.

Os eventos externos a que esta aplicação está sujeita referem-se nomeadamente às mensagens enviados pela aplicação SimTwo e aplicação local de controlo do AGV real. Assim, e começando pela aplicação SimTwo cujo tempo de ciclo corresponde a 40 ms, tal como já foi referido, encarrega-se apenas de obter os dados correspondentes aos impulsos dos *encoders*, posição e orientação no referencial mundo, e medidas dos sensores infravermelhos definidos no ambiente virtual e enviar por UDP para a aplicação de controlo remoto (camada superior). Por sua vez recebe da aplicação Controlo remoto as referências de velocidade e actua nos drivers dos motores. Na figura 5.18 é apresentado um esquema resumo da troca de dados entre estas duas aplicações.

Em relação à aplicação local, mais uma vez foi definido um *timer* de 40 ms, responsável pelo controlo do tempo entre ciclos de controlo. Assim esta aplicação lê dos drivers dos motores reais o número de impulsos dos *encoders* e envia para a aplicação de controlo remoto. Por outro lado recebe da aplicação controlo remoto a velocidade de referência de cada uma das rodas motrizes e envia o comando para os *drivers* dos motores. Na figura 5.19 a troca de dados entre a aplicação controlo remoto e controlo local é apresentada

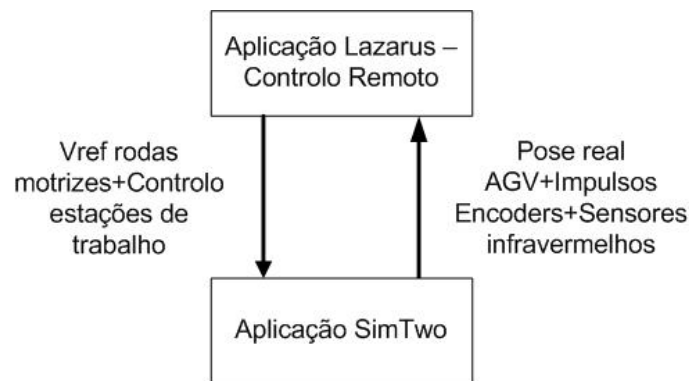


Figura 5.18: Troca de dados entre a aplicação de controlo remoto e SimTwo

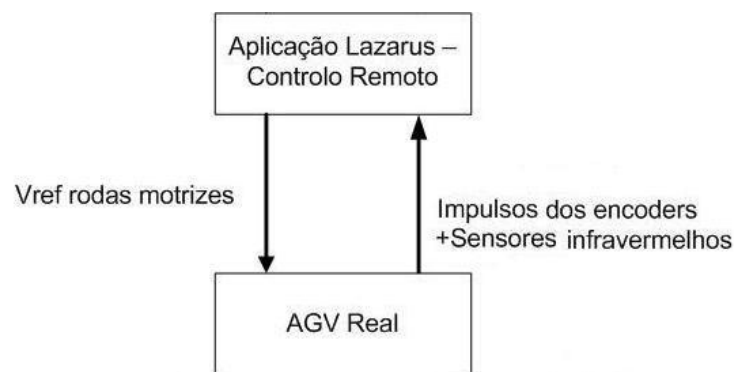


Figura 5.19: Troca de dados entre a aplicação Controlo Remoto e Controlo Local

Nas próximas secções serão apresentados alguns dos algoritmos desenvolvidos na aplicação de controlo remoto, úteis tanto para a construção da aplicação de demonstração, como para uma futura utilização por parte dos concorrentes.

5.5 Funções básicas de movimento implementadas

Para o controlo do movimento do veículo no ambiente industrial foram desenvolvidas as seguintes rotinas:

- GoToXY
- Follow Line
- Follow Circle
- Stop

Estas rotinas de baixo nível poderão ser utilizadas para a definição de trajetórias para o veículos, sendo úteis para o controlo dos AGVs no ambiente industrial.

5.5.1 GoToXY

A rotina GoToXY, (`GotoXYTheta(xpos,ypos,thetapos,Vd: double)`), recebe como parâmetros de entrada a velocidade desejada, o ponto para onde pretende-se que o veículo se dirija e a sua orientação final. Como saídas temos a velocidade de cada um das rodas motrizes. Esta rotina levará o veículo para o ponto pretendido segundo uma linha recta, sempre que possível.

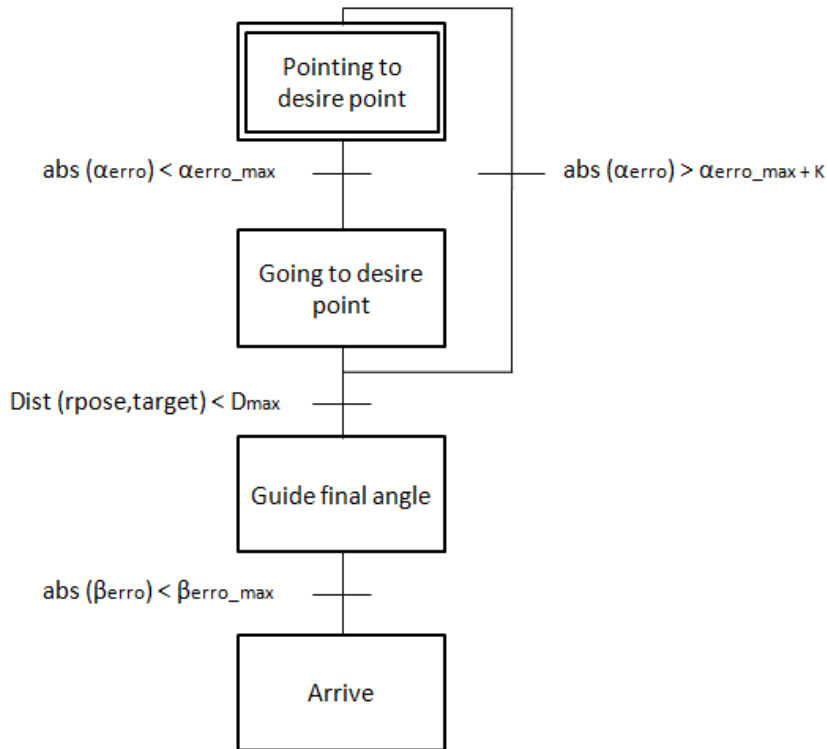


Figura 5.20: Máquina de estados da rotina GoToXY

$$\alpha_{erro} = \text{normalize_ang}(\text{orientationPoint} - \theta_r, 0) \quad (5.34)$$

$$\beta_{erro} = \text{normalize_ang}(\text{desireTheta} - \theta_r, 0) \quad (5.35)$$

A função $Dist(.)$ calcula a distância entre dois pontos no referencial global. Já a função $\text{normalize_ângulo}(.)$ normaliza o ângulo para valores de 0 a 180 e de -180 a 0, possuindo uma variante para ângulos em graus e outra para radianos.

5.5.2 Follow Line

Esta rotina controla o veículo de forma a que este siga uma recta definida no referencial global. Para tal o seus parâmetros de entrada são um ponto pertencente à recta, o ângulo da recta em relação ao referencial mundo, a velocidade desejada e a direcção (`FollowLine(xl,yl,alphal,Vd,Dir: double)`). Mais uma vez a saída do algoritmo é a velocidade de cada uma das rodas motrizes.

Sequência de passos do algoritmo:

- Orientação do robô para o ponto da recta
- Aproximação do robô na velocidade máxima (em direcção à recta).
- Quando a distância entre o robô e a recta é menor que um erro máximo o robô deve aproximar-se da recta com um raio de curvatura que lhe permita atingir a recta orientado paralelamente a esta.
- No estado final o robô percorre a recta (correção de ângulo e distância a recta).

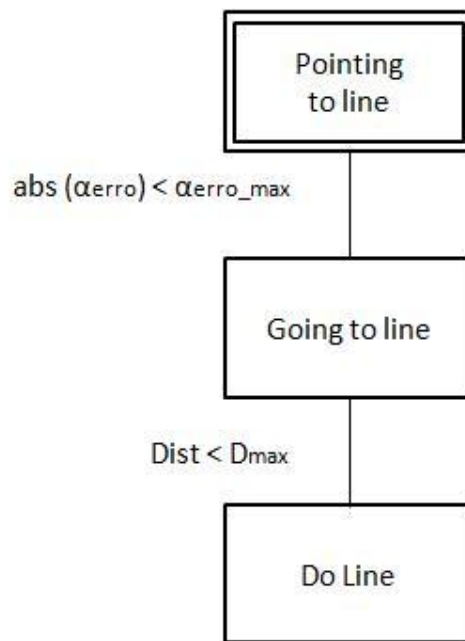


Figura 5.21: Máquina de estados da rotina Follow Line

5.5.3 Follow Circle

A rotina Follow Circle foi implementada com o objectivo criar uma função capaz de controlar o veículo durante uma curva (arco de uma circunferência). Esta rotina recebe como parâmetros de entrada o centro de uma circunferência, o raio de curvatura, a velocidade desejada e a direcção (FollowCircle(xc,yc,R,Vd,Dir: double)). De seguida é apresentada a máquina de estados da rotina.

$$dist = \text{sqrt2}(xc,x,yc,y) \quad (5.36)$$

$$\alpha_{erro} = \text{normalize_ang}(\text{orientationPoint} - \theta_r, 0) \quad (5.37)$$

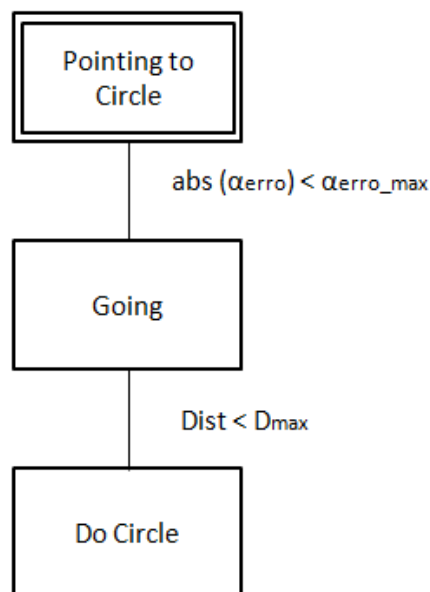


Figura 5.22: Máquina de estado Follow Circle

O veículo inicia assim a sua trajectória por se orientar para a periferia da circunferência. Quando a distância à periferia for inferior a uma constante (D_{max}) o veículo executa o arco de circunferência.

5.5.4 Stop

Esta rotina é executada quando o botão Stop é invocado. Esta encarrega-se apenas de colocar as velocidade de cada umas rodas motrizes igual a zero e inicializar todos os estados das três rotinas anteriores definidas. Esta função tem como saída a velocidade de cada uma das rodas motrizes.

5.6 Planeamento de trajectórias - Ferramentas de Apoio

Um dos maiores desafios no que diz respeito ao controlo de veículos autónomos, corresponde a selecção de trajectórias que o robô deverá seguir em cada instante. Este problema torna-se ainda mais crítico quando o número de veículos aumenta. De lembrar que pretendemos que o caminho seleccionado seja o caminho óptimo, uma vez que a trajectória de cada um dos veículos irá influenciar o tempo de transporte de cada peça, tendo implicação directa na eficiência do sistema.

Com o intuito de disponibilizar algumas ferramentas de apoio ao concorrentes, os quais poderão optar pelo seu uso, foram estudados alguns algoritmos usualmente utilizados em problemas de *pathfinding*. Foram então seleccionados os algoritmos *Breadth-First Search* (BFS) e *A-star* (A^*). De seguida cada um dos algoritmos será apresentado de forma mais detalhada, sendo no final comparados alguns resultados obtidos.

5.6.1 Algoritmo BFS

O algoritmo *Breadth-First Search* (BFS) é um algoritmo que amplia e examina todos os nós de um grafo, ou seja, partindo do nó inicial executa uma procura exhaustiva em largura, sem considerar o seu nó alvo, até o encontrar. No entanto este algoritmo apenas retorna a solução óptima no caso de estarmos perante um grafo não pesado.

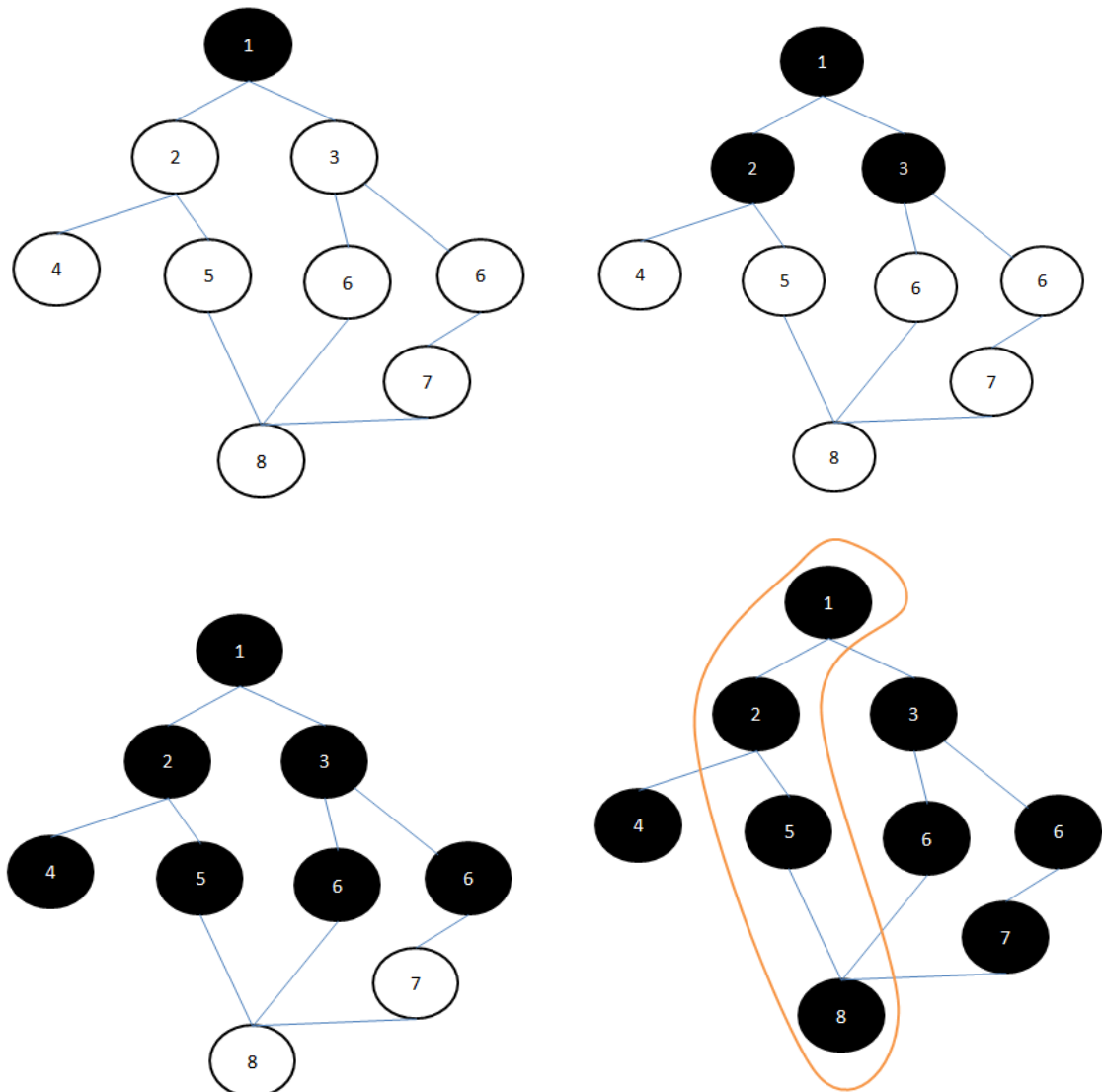


Figura 5.23: Algoritmo *Breadth-First Search* (BFS) - Exemplo

Do ponto de vista do algoritmo implementado, foram criados dois vectores que irão guardar para cada nó os vizinhos correspondentes, de forma a que no final o caminho possa ser reconstruído desde o nó final para o nó inicial.

De forma a que o algoritmo seja melhor compreendido, na figura 5.23 é apresentado um pequeno exemplo. O objectivo será partindo do nó 1 chegar ao nó 8 pelo caminho mais curto.

Como se pode verificar o caminho mais curto foi encontrado. No entanto o grafo seleccionado para o exemplo não foi escolhido ao acaso. Como se pode verificar existem dois caminhos mais curtos: 1-2-5-8 e 1-3-6-8. Assim o algoritmo selecciona para o caminho mais curto, entre dois caminhos com a mesma distância, aquele em que o primeiro pai não comum é o que apresenta menor índice (neste caso o 2). Este facto vai ter alguma influência no grafo construído para o chão de fábrica modelado em SimTwo.

Tendo então descrito um dos algoritmos implementados, de seguida é apresentado o grafo definido para o chão de fábrica desenvolvido em SimTwo (figura 5.24). Este grafo foi construído tendo em conta as trajectórias restritas definidas no chão de fábrica (a verde na figura 5.2).

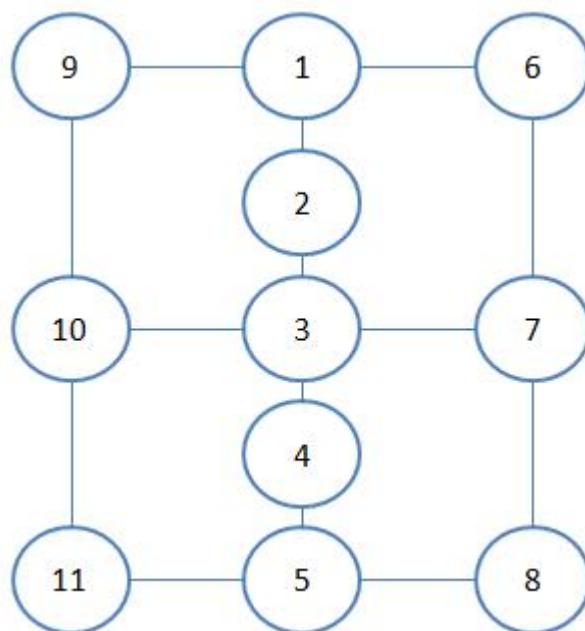


Figura 5.24: Grafo Layout SimTwo

Como podemos observar, no percurso entre estações de trabalho foi adicionado um maior número de nós. Esta facto deve-se à necessidade de que o percurso entre as estações de trabalho (no grafo os nós (1,2,3,4 e 5) deva ser evitado, com a excepção de quando o veículo pretende ir para o nó 3. Juntamente com o factor anterior houve um cuidado na numeração dos nós, uma vez que se pretende, que a trajectória do AGV desde o nó inicial para um nó final, seja realizada por forma a que este execute um menor número de rotações possíveis (o AGV executa as acções de rotação com velocidade mais baixa que em movimento linear).

Na figura 5.25, é apresentado um exemplo do algoritmo implementado.

5.6.2 Algoritmo de Pesquisa A*

O algoritmo de pesquisa A*(Astar) foi pela primeira vez dado a conhecer em 1968, por Peter Hart, Nils Nilsson e Bertram Raphael [49] sendo largamente utilizado em problemas de

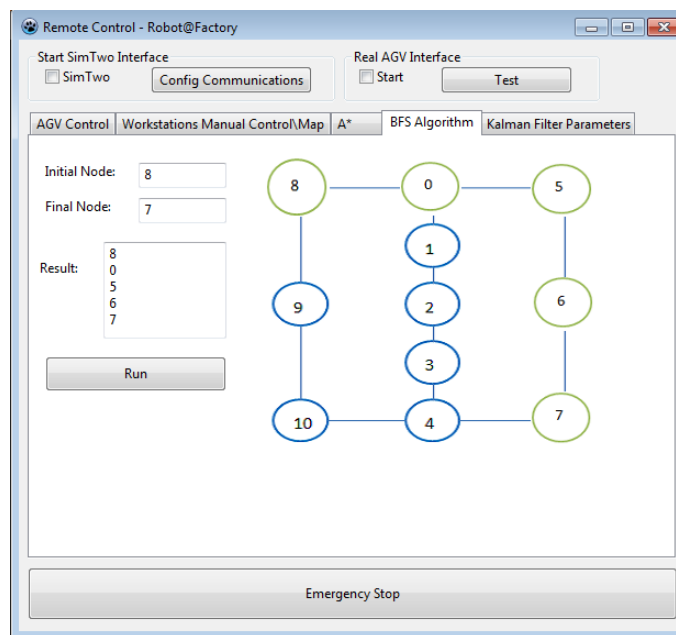


Figura 5.25: Exemplo - Algoritmo BFS implementado

pathfinding. Este algoritmo surge da combinação de dois algoritmos muito utilizados na área de investigação operacional. São eles o algoritmo de *Dijkstra* e *Best-first*.

Este algoritmo distingue-se de muitos outros, nesta área de aplicação, devido ao facto de recorrer a uma função heurística, usualmente denominada $F(n)$, de proximidade à solução final. Esta função é composta pela soma de duas outras funções $g(n)$ e $h(n)$ ($F(n) = g(n) + h(n)$). De seguida é apresentada uma descrição mais detalhada de cada uma das funções.

- $g(n)$ - Esta função é uma função determinística, indicando o custo desde o nó inicial até ao nó corrente (nó actual em pesquisa).
- $h(n)$ - Esta função retorna uma estimativa do custo do percurso desde o nó corrente até ao nó terminal (nó objectivo). Esta função permite embeber conhecimento sobre o domínio do problema na pesquisa de soluções. O valor estimado por esta função não deverá tomar valor superiores aos reais (função heurística admissível), sendo por isso normalmente utilizada em algoritmos de caminho mínimo, a distância em linha recta entre dois pontos (nó inicial e nó objectivo). A performance do algoritmo A* é muito dependente desta função heurística.
- $F(n)$ - Esta função, e tal como já referido, representa o somatório da função $g(n)$ e $h(n)$, indicando assim o custo estimado mínimo, desde o nó inicial até ao nó terminal.

O algoritmo A* inicia a sua pesquisa com o nó inicial, mantendo numa fila aberta com os nós vizinhos ao nó corrente. Quanto menor for $F(n)$ maior é a prioridade desse nó.

Em cada etapa do algoritmo, o nó com o menor valor de $F(n)$ é retirado da fila, os valores de f e h dos seus vizinhos são actualizados em conformidade, sendo adicionados à fila. O algoritmo continua até que um nó objectivo tenha um valor de $F(n)$ inferior a qualquer outro nó presente na

fila (ou até que a fila esteja vazia). O nó objectivo poderá ser preterido várias vezes, se existirem nós na fila que tomem valores de $F(n)$ inferiores, uma vez que podem levar a um caminho mais curto desde o nó inicial até ao final. O valor de $F(n)$ no final do algoritmo, é a distância mínima do nó inicial até ao nó final, uma vez que a função h , para o nó final, toma valor igual a zero. A medida que o algoritmo decorre poderá ser guardado para cada nó seleccionado o seu antecessor, permitindo no final reconstruir, do nó final para o nó inicial, o percurso do caminho mais curto.

Para a nossa aplicação foi utilizado o algoritmo A* já desenvolvido para a equipa de futebol robótico da FEUP, tendo-se procedido à construção do mapa do ambiente fabril, modelado em SimTwo, e sua integração na aplicação de controlo desenvolvida em Lazarus.

Na figura 5.26 é apresentada a janela na aplicação de controlo correspondente ao algoritmo A* e o respectivo mapa (grelha com resolução 144x128).

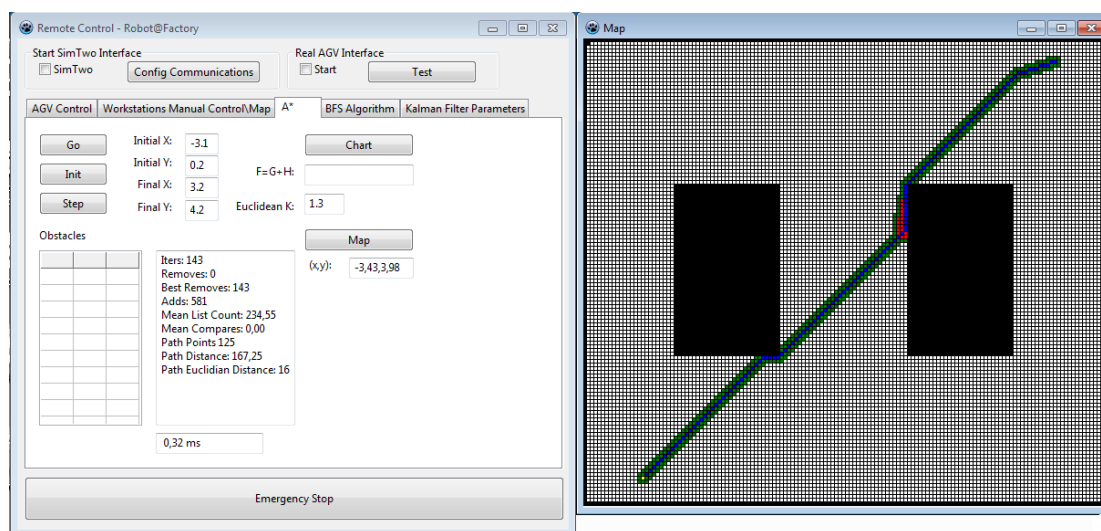


Figura 5.26: Exemplo - Algoritmo A*

De salientar que os obstáculos (estações de trabalho) foram construídos tendo em conta as suas dimensões definidas em SimTwo mais uma constante, que corresponde a um pouco mais de metade da largura do veículo. Esta constante foi adicionada por forma a que o durante o caminho calculado o veículo não colida com os obstáculos.

5.6.3 Comparação entre o algoritmo BFS e A*

Avaliando ambos os algoritmos, pelos exemplos aqui apresentados, podemos concluir que o algoritmo BFS é um algoritmo mais indicado caso os concorrentes optem por trajectórias restritas (a verde no chão de fabrica). Bastará então associar a cada nó um ponto no mapa e a medida que o veículo vai chegando a cada nó, o nó seguinte é atribuído à sua trajectória.

Quanto ao algoritmo A* pode-se verificar que será tirada uma maior vantagem caso seja utilizado para a definição de trajectórias “livres” e flexíveis. Recorrendo a este algoritmo os AGVs executarão o percurso mais directo entre o nó inicial e o nó final, minimizando assim o tempo de transporte. Finalmente este algoritmo também, poderá ser uma ferramenta poderosa para as

tarefas de evitar colisões uma vez que os *AGVs* no chão de fábrica poderão ser considerados como obstáculos móveis podendo ser considerados no mapa deste algoritmo.

5.7 Robô em desenvolvimento

Nesta secção o robô real, que se encontra em construção, é apresentado. Tal como referido anteriormente, este robô serviu de base para a construção do veículo em SimTwo. Trata-se então de um robô de tracção diferencial de dimensões reduzidas, com 36 cm de comprimento e 30 cm de largura (ver figura 5.27).



Figura 5.27: Robô desenvolvido



Figura 5.28: Sistema: Motor, Driver e rodas

Tal como se pode observar apenas a estrutura base foi construída. Na figura 5.28 é apresentado o sistema de driver, motor e rodas utilizado. Cada um destes componentes será de seguida apresentado de forma mais detalhada.

5.7.1 Motores e *Encoder*

Na figura 5.29 é apresentada a solução utilizado no robô. Trata-se de um motor DC, o qual já inclui caixa redutora e *encoder*. O motor é caracterizado por uma potência nominal de 4.22 W, tensão nominal 12V, uma relação da caixa redutora de 30:1. O *encoder* utilizado é um *encoder* de efeito de Hall incremental, com uma resolução de 360 impulsos (à saída da caixa redutora) por

cada volta completa. O sistema completo aqui apresentado permite uma velocidade mínima (não nula) de 1.5 rpm e máxima de 200 rpm (sem carga).



Figura 5.29: Sistema EMG30 - Motor, caixa redutor e *encoder*

5.7.2 Driver MD25



Figura 5.30: Driver Motor DC - Devantech MD25 Dual 12V 2.8A H-Bridge

O driver aqui apresentado é um driver desenhado para funcionar em conjunto com os motores EMG30, anteriormente apresentados. Este driver permite o controlo de dois motores em simultâneo. Para a comunicação com o drive, a comunicação série ou I2C poderá ser utilizada.

Características Principais:

- Permite ler os *encoders* dos motores, e retorna o número de impulsos por forma a obter a distância percorrida e direcção.
- Permite dois modos de operação: um de controlo individual de cada motor e outro combinado em que é enviado o sinal de velocidade e de viragem.

- Permite a leitura das corrente nos motores.
- Permite regular a aceleração de forma independente de cada motor

5.7.3 Rodas



Figura 5.31: Roda do robô

Na figura 5.31 é apresentada um roda do veículo. A roda tem um diâmetro de 100mm, tendo uma área de contacto com o solo reduzida, diminuindo a sua tracção em pisos irregulares. No entanto, tendo em conta as dimensões do veículo e a aplicação no qual será utilizada, este factor não será muito relevante.

5.7.4 Alimentação

Finalmente, a bateria utilizada para a alimentação do veículo, é uma bateria de 12 V de chumbo, preparada para ciclos de descarga completo.



Figura 5.32: Alimentação robô

Capítulo 6

Conclusões e Trabalho Futuro

6.1 Conclusões

Esta dissertação pode ser devida em duas fases. Uma primeira fase consistiu na concepção e desenvolvimento de um modelo de simulação de um sistema de logística interna para uma empresa de revestimento de peças por pó. O sistema de logística interna é composto por um sistema de transporte de peças entre postos de trabalho, tendo cada peça uma sequência de operações e tempos de processamento específicos. Este modelo foi construído sobre a ferramenta de simulação ARENA.

Após uma análise inicial verificou-se que a flexibilidade do processo de fabrico se encontrava abaixo dos requisitos, tendo influência directa no seu rendimento. Foram propostas diferentes soluções para o problema apresentado, destacando-se a solução que consiste na substituição de parte do sistema actual de transporte por um baseado em *AGVs*. Esta mudança permitiu que a flexibilidade do sistema de produção fosse aumentada, poupando ao mesmo tempo, espaço e recursos quando comparado com outras soluções propostas. A modularidade do sistema foi ainda aumentada permitindo que adaptações futuras possam ser realizadas de forma mais simples sem obrigar a parar o processo de produção.

Assim podemos afirmar que o processo de transporte de peças no chão de fábrica de uma empresa, assume um papel crítico na produtividade do sistema de produção, na medida em que este depende da eficiência e eficácia do sistema de transporte utilizado.

Uma segunda fase, correspondeu à construção de um sistema de produção, utilizando para isso a ferramenta de simulação SimTwo. Foram desenvolvidas quatro estações de trabalho e o veículo responsável pelo transporte das peças. O sistema de produção, construído em simulação, foi pensado tendo em vista a sua construção real, num futuro próximo.

Algoritmos de controlo e decisão que afectam o comportamento do veículo foram desenvolvidos. Aqui foi possível perceber a complexidade que os sistema de logística interna baseado em *AGVs* apresentam. O seu sistema de localização, a definição das suas trajectórias, as estratégias para evitar a colisão entre veículos, a afectação dos *AGVs* por forma a que a eficiência do sistema

de transporte seja levada ao extremo, representam problemas de elevada complexidade, com influência directa na eficiência e eficácia com que operam.

6.2 Trabalho Futuro

Como trabalho futuro proponho, e começando pelo sistema de produção modelado em ARENA, que a avaliação de uma logística interna baseada em *AGVs* para outros sistema de produção, por forma a comparar diferentes cenários de produção e *layout* fabris, seja realizado. Os resultados daí provenientes poderão permitir concluir em que situações a utilização de um sistema de transporte baseada em *AGVs* é uma mais valia em relação aos restantes meios de transporte mais utilizados. Este trabalho surge do facto de se ter verificado alguma escassez de informação, em relação a sistemas baseados em *AGVs* actualmente implementados e seu impacto na produção, a quando da realização do estado da arte.

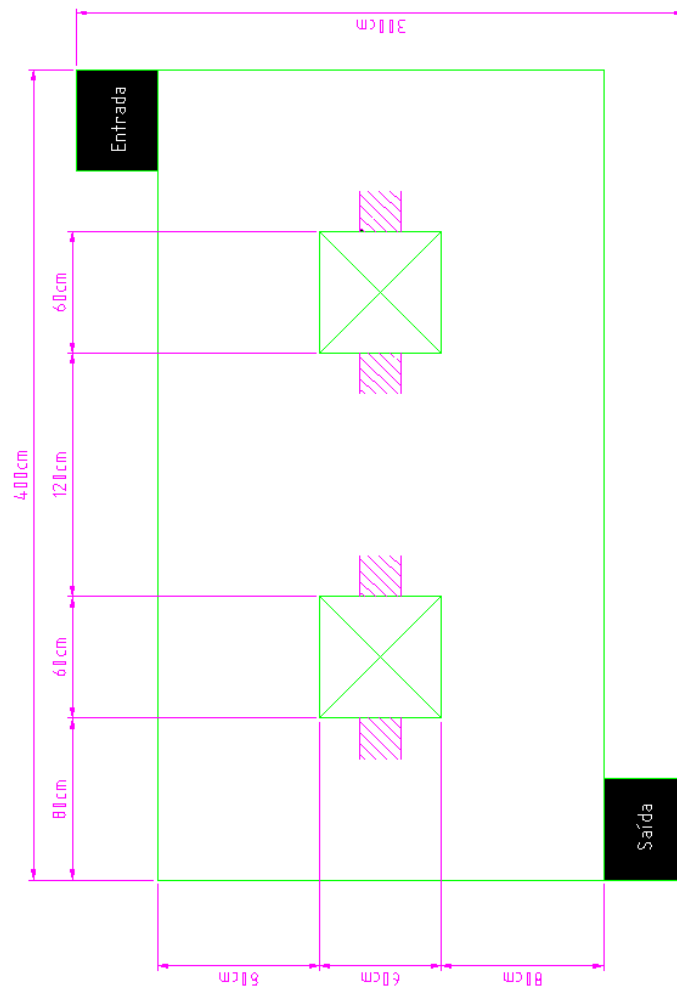
Para o sistema construído em SimTwo, desde já existe a necessidade de realizar a sua construção real, bem como a finalização da construção do veículo. Estas tarefas levam desde já a necessidade de escolha de motores, sensores e componentes que daí advêm que permitam o controlo das estações de trabalho, bem como a implementação de um sistema de localização absoluta e detecção de obstáculos para o veículo.

Em relação ao veículo desenvolvido, e após a realização de alguns testes pode-se verificar que o comportamento dinâmico dos motores em SimTwo não estavam a reproduzir fielmente o comportamento dos motores reais, sendo por isso proposto a realização de alguns testes adicionais com os motores reais de forma a que o seu comportamento dinâmico seja avaliado. Os resultados obtidos poderão ser comparados com os obtidos em SimTwo permitindo assim ajustar os respectivos parâmetros.

Por último, e a quando da finalização da dissertação, verificou-se que o SimTwo já possuía sensores de linhas brancas no pavimento. Daí resulta a necessidade de mudar a cor das trajetórias definidas no chão de fábrica da cor verde para branco e implementar o respectivo algoritmo de controlo.

Anexo A

Esquemático Layout Fabril Robot Factory



Referências

- [1] Porto marítimo de roterdão. Disponível em <http://nl.wikipedia.org/wiki/Portaal:Maritiem/Afbeeldingen>.
- [2] H. Martínez-Barberá D. Herrero-Pérez. Modeling Distributes Transportation Systems Composed of Flexible Automated Guided Vehicles in Flexible Manufacturing Systems. *Transaction on Industrial Informatics*, páginas 1 – 15, 2010.
- [3] Jun Ota Ryosuke Chiba e Tamio Arai. Design of Robust Flow-path Network for AGV Systems using Competitive Co-evolution with Packaging. Em *IEEE/RSJ International Conference on Intelligent Robots and Systems(IROS 2005)*, páginas 3137–3142, Agosto 2005.
- [4] Romme G. Berends P. Simulation as a research tool in management studies. *European Management Journal*, 17(6):576–583, 1999.
- [5] Automated Guided Vehicles / Self Guided Vehicles. Disponível em <http://wings.buffalo.edu/eng/mae/courses/460-564/AGV.pdf>.
- [6] Automated Guided Vehicles Project. Disponível em <http://www.steinbockus.com/>.
- [7] Automatic guided vehicle. Disponível em http://en.wikipedia.org/wiki/Automated_Guided_Vehicle.
- [8] History of automatic guided vehicle systems. Disponível em <http://www.agvsystems.com/res/history.htm>.
- [9] Wade W. Hamilton. A Methodology That Integrates The Schedulling Of Job Sequencing And Agv Dispatching. Tese de mestrado, Virginia Polytechnic Institute and Sate University, 1992.
- [10] Akiko Shinozaki Satoshi Hoshino, Jun Ota e Hideki Hashimoto. Higly Efficient Agv Transportation System Management Using Agent Cooperation and Container Storage Planning. Em *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2005)*, páginas 1588–1593, Outubro 2002.
- [11] Satoshi Hoshino, Jun Ota, Akiko Shinozaki, e Hideki Hashimoto. Optimal design, evaluation, and analysis of agv transportation systems based on various transportation demands. Em *IEEE- Innternational Conference on Robotics and Automation*, páginas 1400–1406, 2005.
- [12] Arun S. Development of Determinist Collision Avoidance. Tese de mestrado, Rochester Institute of Tecnology, 2008.

- [13] Kousuke Inque Ryousuke Chiba Tomokazu Hirano Jun Ota, Tamio Arai. Flexible transportation system by cooperation of conveyor-loaded agv's. Em *IEEE International Conference on Robotics and Automation*, 2000.
- [14] Tipologias AGV's disponíveis no mercado. Disponível em <http://www.agvsystems.com/faqs/q19.htm>.
- [15] Soluções de carregamento de baterias. Disponível em http://www.egeminusa.com/pages/agvs/agvs_battery_charging.html.
- [16] Soluções de carregamento de baterias. Disponível em <http://www.jbtc-agv.com/en/knowledge/how%20the%20agv%20system%20works.aspx>.
- [17] Soluções de carregamento de baterias. Disponível em <http://www.logisticshandling.com/absolutem/templates/article-agv.aspx?articleid=94&zoneid=1>.
- [18] Soluções de carregamento de baterias. Disponível em <http://www.bastiansolutions.com/equipment/automated-guided-vehicles/default.asp>.
- [19] Soluções de carregamento de baterias. Disponível em http://www.jervisbwebb.com/brochures/Bul-3083_Solutia_Application_Profile.pdf.
- [20] Soluções de carregamento de baterias. Disponível em http://www.vahleinc.com/battery_charging_contacts.html.
- [21] Soluções de carregamento de baterias. Disponível em <http://www.agvsystems.com/faqs/q19.htm>.
- [22] Segurança em AGV's. Disponível em http://www.agve.se/page/by_desc/safety.
- [23] Henri R. A. Landeweerd Ronald J. Mantel. Design and Operational Control of an AGV system. *Int J. Production Economics* 41, páginas 257–266, 1995.
- [24] J. M. A. Tanchoco Pius J. Egbelu. Potentials for bi-directional guide-path for automated guided vehicle based system. *Int. J. Prod. Res.*, 24(5):1075–1097, 1986.
- [25] Jun Ota Ryosuke Chiba e Tamio Arai. Integrated Design with Classification of Transporter Routing. Em *IEEE - Conference of Intelligent Robots and Systems*, páginas 1820–1825, Outubro 2002.
- [26] S. S. Gurav B. R. Sarker. Route planning for automated guided vehicles in a manufacturing facility. *International journal of production research*, 43(21):4659–4683, November 2005.
- [27] Glossary of Terms. Disponível em <http://www.physicalgeography.net/physgeoglos/s.html#system>.
- [28] Flexim. Disponível em <http://www.flexsimsoft.com.br/gerenciador/uploads/NewsletterQ4WhySimulate.pdf>.
- [29] Joana G. Rodrigues. An Analysis and Evaluation of Discrete Production Systems: A Simulation based approach. Tese de mestrado, Virginia Polytechnic Institute and State University, 1992.

- [30] António S. Paiva. Geração automática de modelos de simulação de uma linha de produção na indústria têxtil. Tese de mestrado, Universidade do Minho Escola de Engenharia, 2005.
- [31] Jorge L. Silva. Modelagem e Simulação Discreta de Sistemas. Relatório té, Universidade Estadual do Ceará.
- [32] Paulo J. Marques. Simulação de um Sistema Automático de Logística Interna para a Indústria de Calçado. Tese de mestrado, Faculdade de Engenharia Universidade do Porto, 2007.
- [33] Simulador Rockwell ARENA. Disponível em <http://www.arenasimulation.com/>.
- [34] Benjamin Melamed Tayfur Altiok. *Simulation Modeling and Analysis with ARENA*. Open University Press, elsevier edição, 2007.
- [35] Masami Konishi Masakazu Ando, Tatsushi Nishi e Juan Imai. Autonomous decentralized route planning method and its application to a dynamic route planning. *International Symposium on Computational Intelligence in Robotics and Automation*, 2:718–723, Agosto 2003.
- [36] Lima D. Localização absoluta de robôs móveis em ambientes industriais. Tese de mestrado, Faculdade de Engenharia Universidade do Porto, 2010.
- [37] Maykol A. Localização e mapeamento na robótica móvel em ambientes não-industriais. Tese de mestrado, Faculdade de Engenharia Universidade do Porto, 2010.
- [38] SimTwo. Disponível em <http://paginas.fe.up.pt/~paco/wiki/index.php?n=Main.SimTwo>.
- [39] Ode(open dynamics engine. Disponível em <http://www.ode.org/>.
- [40] Alves H. Navegação e controlo de um veículo móvel. Tese de mestrado, Faculdade de Engenharia Universidade do Porto, 2010.
- [41] Pascal script. Disponível em http://wiki.freepascal.org/Pascal_Script.
- [42] Everett H. R. Feng L. Borenstein, J. Where am i? sensors and methods for mobile robot positioning. University of Michigan, 1996.
- [43] Sobreira H. Clever robot. Tese de mestrado, Faculdade de Engenharia Universidade do Porto, 2010.
- [44] T. Tsubouchi S. Arimoto Tonouchi, Y. Fusion of dead-reckoning positions with a workspace model for a mobile robot by bayesian inference. Em *International Conference on Intelligent Robots and Systems*, páginas 1347–1354, 1994.
- [45] J.L. Chenavier, F; Crowley. Position estimation for mobile robot using vision and odometry. *Proceedings of the IEEE International Conference on Emerging Technologies and Factory Automation*, 3(0-8186-2720-4):2588–2593, 1992.
- [46] Negenborn R. Robot localization and kalman filters on finding your position in a noisy world. Tese de mestrado, Utrecht University, 2003.
- [47] Driver motores (md25). Disponível em <http://www.robotshop.ir/files/products/datasheets/md25.pdf>.

- [48] User datagram protocol (udp). Disponível em http://pt.wikipedia.org/wiki/User_Datagram_Protocol.
- [49] B.Raphael P. Hart, N. Nilsson. A formal basis for the heuristic determination of minimum cost paths. *Transactions on Systems Science and Cybernetics SSC4*, 2, 1968.