# UM SIMULADOR PARA VEÍCULOS AUTÔNOMOS

Luis Felipe Saldanha de Menezes

Dissertação submetida ao corpo docente do Instituto de Matemática e Núcleo de Computação Eletrônica da Universidade Federal do Rio de Janeiro como parte dos requisitos necessários para obtenção do grau de Mestre em Informática.

Orientadora: Prof<sup>a</sup>. Eliana Prado Lopes Aude, PhD.

Orientador: Profº. Ernesto Prado Lopes, PhD.

Rio de Janeiro, RJ - BRASIL

JUNHO - 2004

Luis Felipe Saldanha de Menezes

# Um simulador para veículos autônomos

Dissertação submetida ao corpo docente do Instituto de Matemática e Núcleo de Computação Eletrônica da Universidade Federal do Rio de Janeiro como parte dos requisitos necessários para obtenção do grau de Mestre em Informática.

Rio de Janeiro, 30 de Junho de 2004.

Aprovada por:

Prof<sup>a</sup>. Eliana Prado Lopes Aude, Ph.D. (NCE/UFRJ)

Prof<sup>o</sup>. Ernesto Prado Lopes, Ph.D. (UFRJ)

Profº Fabrício Alves Barbosa da Silva, Ph.D

Profº. João Carlos Machado, Ph.D

Menezes, Luis Felipe Saldanha de

Um simulador para veículos autônomos / Luis Felipe Saldanha de Menezes - Rio de Janeiro, 2004.

xii, 88 p.

Dissertação (Mestrado em Informática) – Universidade Federal do Rio de Janeiro – Instituto de Matemática / Núcleo de Computação Eletrônica, 2004.

Orientadora: Eliana Prado Lopes Aude

Orientador: Ernesto Prado Lopes

1. Veículos Autônomos 2. Trajetórias Contínuas 3. Automação e Robótica.

CDD

MENEZES, Luis Felipe Saldanha de, **Um simulador para veículos autônomos**; aplicação de uma trajetória mais suave para o problema de navegação de veículos autônomos. Dissertação (Mestrado em Informática) – Núcleo de Computação Eletrônica / Instituto de Matemática, Universidade Federal do Rio de Janeiro, Rio de Janeiro, 2004.

Este trabalho consiste na construção de um simulador da arquitetura proposta em (LOPES et al, 2001) e (AUDE et al, 2003) para o veículo autônomo CONTROLAB AGV, baseado em campos potenciais. Foi proposto em (LOPES et al, 2001), um algoritmo de desvio de obstáculos para um veículo autônomo (CONTROLAB AGV), no qual adota-se uma estratégia similar a de pessoas cegas que desviam de obstáculos quando estão caminhando. O AGV (Autonomous Guided Vehicle) movimenta-se em um ambiente com uma planta baixa já conhecida e usa uma bengala eletrônica consistindo de sensores infravermelhos para detectar obstáculos desconhecidos. O campo é adaptado cada vez que a bengala encontra um obstáculo. Dependendo da arquitetura do sistema veículo e bengala pode ser necessária suavizar a trajetória entre dois pontos consecutivos do campo discretizado, tornando-a compatível com a cinemática do veículo autônomo. O planejamento de trajetória do AGV, baseado em campos potenciais, consiste na construção de uma função potencial adaptativa que determina um campo vetorial. Este campo indica a direção do robô, a cada instante, a partir de sua localização. No trabalho proposto em (AUDE et al, 2003) uma função polinomial de quinta ordem é definida entre dois pontos do grid de valores potenciais calculados para suavizar a trajetória. As condições de interpolação da função polinomial de quinta ordem harmonizam a dinâmica do AGV e permite que ele desempenhe movimentos complexos. Através do simulador CONTROLAB AGV é possível estudar essas trajetórias contínuas e então dinamizar e minimizar os testes realizados para o estudo do comportamento do AGV.

v

MENEZES, Luis Felipe Saldanha de, **A simulator for autonomous guided vehicles**; Rio de Janeiro, 2004. Dissertação (Mestrado em Informática) – Núcleo de Computação Eletrônica / Instituto de Matemática, Universidade Federal do Rio de Janeiro, Rio de Janeiro, 2004.

This work consist in development of simulator of architecture proposal in (LOPES et al, 2001) e (AUDE et al, 2003) for an obstacle avoidance algorithm, which used potential fields and similar strategy to that adopted by a blind person to avoid obstacles whilst walking. The problem analyzed in (LOPES et al, 2001) consists of an AGV (Autonomous Guided Vehicle), which moves within an office environment with a known floor plan and uses an "electronic stick" made up of infrared sensors to detect unknown obstacles in its path. Initially, a global potential navigation function, defined for each room in the floor plan, incorporates information about the dimensions of the room and the position of the door, which the AGV must use to leave the room. Whilst the AGV moves, this global potential navigation function is properly modified to incorporate information about any newly detected obstacle. The field is adapted each time that the "electronic stick" finds an obstacle. Depending on the architecture of the system vehicle and "electronic stick" it can be necessary to alleviate the trajectory between consecutive points of the field, becoming it compatible with the kinematics of the independent vehicle. The planning of trajectory of the AGV, based on potential fields, consists of the construction of adaptive potential function that determines a vector field. This field indicates the direction of the robot, at every moment, from its localization. In the previous work in (AUDE et al, 2003) a polynomial function of fifth order it is defined between points grid of potential values calculated to alleviate the trajectory.

Figura 1.1 - O ambiente CONTROLAB - AGV
Figura 1.2 - A arquitetura do CONTROLAB AGV5
Figura 1.3 - (a) Planta baixa, com origem e destino do caminho - (b) grafo de conectividade, incluindo S e G
Figura 1.4 - Espaço de configuração no R <sup>2</sup> 9
Figura 1.5 - Campo potencial original de um recinto12
Figura 1.6 - Linhas Equipotenciais do campo original12
Figura 1.7 - Locomoção do AGV em tempo real13
Figura 2.1 – Um exemplo de metodologia de simulação aplicável em qualquer modelo de simulação
Figura 2.2 – Execução gradativa de Sistemas de tempos discretos
Figura 2.3 Tabela com um exemplo de sistemas de tempos discretos
Figura 3.1 - Grid representando os pontos do ambiente por onde o AGV gera sua trajetória34
Figura 3.2 - <i>Grid</i> original da sala sendo atualizado, a partir do <i>grid</i> atualizado pontual após a detecção de um obstáculo
<ul> <li>Figura 3.2 - <i>Grid</i> original da sala sendo atualizado, a partir do <i>grid</i> atualizado pontual após a detecção de um obstáculo</li></ul>
<ul> <li>Figura 3.2 - <i>Grid</i> original da sala sendo atualizado, a partir do <i>grid</i> atualizado pontual após a detecção de um obstáculo</li></ul>
<ul> <li>Figura 3.2 - <i>Grid</i> original da sala sendo atualizado, a partir do <i>grid</i> atualizado pontual após a detecção de um obstáculo</li></ul>
<ul> <li>Figura 3.2 - <i>Grid</i> original da sala sendo atualizado, a partir do <i>grid</i> atualizado pontual após a detecção de um obstáculo</li></ul>
<ul> <li>Figura 3.2 - <i>Grid</i> original da sala sendo atualizado, a partir do <i>grid</i> atualizado pontual após a detecção de um obstáculo</li></ul>
<ul> <li>Figura 3.2 - <i>Grid</i> original da sala sendo atualizado, a partir do <i>grid</i> atualizado pontual após a detecção de um obstáculo</li></ul>
<ul> <li>Figura 3.2 - <i>Grid</i> original da sala sendo atualizado, a partir do <i>grid</i> atualizado pontual após a detecção de um obstáculo</li></ul>
<ul> <li>Figura 3.2 - <i>Grid</i> original da sala sendo atualizado, a partir do <i>grid</i> atualizado pontual após a detecção de um obstáculo</li></ul>
<ul> <li>Figura 3.2 - <i>Grid</i> original da sala sendo atualizado, a partir do <i>grid</i> atualizado pontual após a detecção de um obstáculo</li></ul>

Figura 4.4 - Geração do vetor aceleração a partir do vetor velocidade	63
Figura 5.1 – Trajetória normalizada com <b>tempo = 1seg</b> ), violando as restrições de cinemática	a do
	/3
Figura 5.2 – Trajetória Real com tempo <b>t = tempo</b> <sub>SPLINE</sub>	76
Figura 5.3 – Continuidade entre as trajetórias spline	77
Figura 5.4 Trajetória spline definida no simulador	78
Figura 5.5 – Trajetória definida para condições iniciais baseadas em relação a velocidade aceleração no campo	e a 79
Figura 5.6 - Trajetória definida para condições iniciais baseadas em relação a velocidade aceleração no campo	e a 80
Figura 5.7 Trajetória spline com condição inicial de aceleração tangencial e normal = 0	81
Figura 5.8 Trajetória spline com condição inicial de aceleração tangencial e normal baseado campo potencial	o no 81
Figura 5.9 - Comportamento do campo potencial do ambiente simulado através das isolir com a detecção do primeiro obstáculo	nhas 82
Figura 5.10 - Comportamento do campo potencial do ambiente simulado através das isolir com a detecção do segundo obstáculo	nhas 83
Figura 5.11 Simulação de um ambiente real	84
Figura 5.12 Simulação em um segundo ambiente definido para os experimentos	84

# SUMÁRIO

1.	INTRODUÇÃO1				
	1.1 Motivação				
		1.1.1 Projeto CONTROLAB AGV	2		
	1.2	SIMULAÇÃO	14		
		1.2.1 Aplicações e Benefícios da simulação	15		
	1.3	Objetivos da tese	16		
	1.4	Conteúdo da tese	18		
2.	TÉCN	NICAS DE SIMULAÇÃO	19		
	2.1	PROCESSO DE SIMULAÇÃO	20		
	2.2	METODOLOGIA DE SIMULAÇÃO	21		
	2.3	CONCEITOS DE MODELAGEM DE SISTEMAS DE SIMULAÇÃO	22		
	2.4	Formalismo de especificação de sistemas de tempos Discretos (E	OTSS)24		
		2.4.1 Especificação do modelo DTSS para o simulador CONTROLAB-	AGV 26		
3.	SIMU	ULADOR CONTROLAB AGV			
	3.1	ARQUIVOS DE PARÂMETROS			
	3.2	REPRESENTAÇÃO DOS AMBIENTES DE SIMULAÇÃO (GERAÇÃO E ATUALIZ	AÇÃO DOS		
	<i>GRIDS</i> D	DO AMBIENTE)	34		
	3.3	INTERFACE GRÁFICA			
	3.4	ISOLINHAS	41		
	3.5	PLANEJADOR DE TRAJETÓRIA GLOBAL	46		
	3.6	PLANEJADOR DE TRAJETÓRIA LOCAL	47		
		3.6.1 Geração do próximo passo	47		
		3.6.2 Detecção de obstáculos	49		
4.	O PR	ROBLEMA SIMULADO	55		
	4.1	TRAJETÓRIAS CONTÍNUAS	56		
	4.2	Trajetória <i>Quintic Spline</i>	57		
	4.3	TRAJETÓRIA QUINTIC SPLINE APLICADA AO PROBLEMA SIMULADO	60		
		4.3.1 Condições iniciais de definição da trajetória	60		

		4.3.2	Cálculo do novo tempo para gerar a nova trajetória quintic spline	64		
		4.3.3	Valores de status do AGV	69		
5.	EXPE	ERIÊNC	IAS SIMULADAS	72		
	5.1	INTRO	DDUÇÃO	72		
	5.2	DESC	RIÇÃO DOS EXPERIMENTOS	72		
	5.3	RESU	LTADOS	73		
		5.3.1	Condições Iniciais do experimento	73		
		5.3.2	Trajetórias em diferentes planos de ambiente	83		
6. CONCLUSÕES						
RF	REFERÊNCIAS BIBLIOGRÁFICAS8					

## 1.1 Motivação

Os veículos autônomos (RMA – Robôs Móveis Autônomos) têm atraído a atenção de um grande número de pesquisadores da área de Inteligência Artificial e Robótica, devido ao desafio que esta área de pesquisa nos propõe: construir sistemas de veículos autônomos, que permitam os veículos sentir o ambiente a fim de realizar tarefas complexas.

Os robôs têm auxiliado o homem em diversas tarefas, tais como a exploração espacial, mineração e exploração de petróleo em águas profundas, fábricas onde há perigo de contaminação e no auxílio a pessoas com deficiências físicas. Devido ao grande desenvolvimento da tecnologia da informática, especialmente a área de hardware, potentes computadores podem ser usados para desenvolver novas técnicas que permitem um robô executar tarefas cada vez mais complexas.

Atualmente existem robôs móveis atuando em diferentes áreas, como por exemplo: robôs desarmadores de bombas, robôs usados para a exploração de ambientes hostis e a condução de veículos (carros) robotizados.

Essas técnicas estão permitindo que esses robôs executem tarefas complexas, que até então só eram executadas por seres-humanos, tais como: aprender com a experiência, navegar por ambientes desconhecidos, desviar de obstáculos imprevistos e reagir de forma inteligente a estímulos do ambiente e trabalhar em conjunto com outros robôs (time) para otimizar a realização das tarefas.

Os veículos autônomos possuem em comum a capacidade de interagir com seu ambiente de trabalho, recebendo informações através de seus sensores e gerando os comandos necessários para que suas tarefas sejam executadas com sucesso.

Num ambiente onde existem obstáculos, o problema de navegação de um robô móvel é tradicional em robótica. Algumas variações do tema, como obstáculos dinâmicos versus estáticos ou robôs holonômicos versus não-holonômicos ainda tornam esse problema mais complexo e mais difícil de resolver (LATOMBE, 1991).

Existem vários trabalhos na literatura envolvendo veículos autônomos. É apresentado em (OSÓRIO et al., 2001) um sistema de controle inteligente de veículos autônomos. Esse sistema é responsável pela automatização da tarefa de condução de um veículo, onde se busca obter um sistema de controle robusto capaz de estacionar um carro em uma vaga paralela. O sistema, chamado SEVA (Simulador de Estacionamento de Veículos Autônomos) permite controlar o carro através da leitura de um conjunto de sensores, gerando os comandos de aceleração e de giro de direção, de modo a localizar e estacionar o carro em uma vaga.

O sistema SEVA conta com um controlador baseado em um sistema especialista (conjunto de regras heurísticas) e com um controlador gerenciado por uma Rede Neural Artificial com aprendizado supervisionado.

#### 1.1.1 Projeto CONTROLAB AGV

O projeto, que vem sendo desenvolvido no laboratório CONTROLAB, no Núcleo de Computação Eletrônica (NCE) da Universidade Federal do Rio de Janeiro UFRJ, consiste no desenvolvimento de um veículo, que opera sobre um ambiente definido por uma planta baixa conhecida. Os veículos autônomos, conhecidos como AGV, recebe comandos da Internet através de rede sem fio e é capaz de se deslocar neste ambiente executando a tarefa desejada. O AGV deve desviar de qualquer obstáculo que venha interceptar sua trajetória.

A arquitetura inicialmente proposta para o veículo autônomo CONTROLAB AGV, visto em (AUDE et al, 1999a), (AUDE et al, 1999b), utilizava um sistema de desvio de obstáculos baseado na fusão das informações de um sistema de visão com um sistema de sonares. Este sistema foi implementado e testado.

O desempenho do sistema se mostrou sensível à forma dos obstáculos, levando-se a uma nova proposta para a arquitetura do AGV. Na nova arquitetura, o planejador de trajetórias (LOPES et al, 2001) utiliza uma "bengala" eletrônica adotando uma estratégia similar à utilizada por uma pessoa cega enquanto se locomove por um ambiente para evitar os obstáculos desconhecidos. Esta "bengala" eletrônica é implementada somente com sensores de infravermelho. Uma das grandes vantagens deste subsistema é a utilização de sensores de baixo custo.

Outra modificação fundamental, na nova proposta da arquitetura, se deu no subsistema de planejamento local. Este subsistema é responsável pelo cálculo da trajetória entre a posição atual do AGV e a próxima porta a ser atingida por ele, dentro de uma sala. Na arquitetura antiga, esta trajetória era definida utilizando um algoritmo baseado em campos potenciais aproximados. Os erros decorrentes desta aproximação eram corrigidos por um sistema inteligente de regras. Na nova arquitetura, o cálculo do campo é feito sem aproximação, substituindo o sistema de equações lineares por um sistema de equações diferenciais parciais.

#### 1.1.1.1 O Ambiente CONTROLAB-AGV

Vários subsistemas inteligentes foram projetados e aprimorados ao longo do tempo, integrando-se ao ambiente CONTROLAB-AGV (CARNEIRO et al, 1999), (AUDE et al, 1999a), (AUDE et al, 1999b), (LOPES et al, 2001). Como mostra a Figura 1.1, seus principais componentes são:

- AGV Autonomous Guided Vehicle, veículo autônomo equipado com:
  - · Sensores de infravermelho;
  - Transmissores/receptores de ondas de radiofreqüência, para comunicação com o servidor de ordens;
  - Motores DC, que comandam três rodas de forma independente;
  - Dispositivos de hardware e módulos de software destinados a armazenar toda inteligência e informações necessárias a sua operação.
- Subsistema de Controle: responsável pelo acionamento e controle dos motores, para que o AGV descreva o caminho determinado pelo Planejador de Trajetórias.
- Subsistema Cliente/servidor: integra várias estações clientes, que enviam pedidos para a locomoção do AGV ao servidor de ordens, via Internet. Toda comunicação com o AGV fica a cargo do servidor. O servidor também é equipado com transceptores de ondas de radiofreqüência.
- Arquiteto CONTROLAB: um tipo especial de cliente; uma ferramenta destinada à edição gráfica da planta baixa do ambiente, a ser transmitida pelo servidor ao AGV.
- Planejador de Trajetória: Sistema residente no AGV, composto por dois subsistemas: o
  planejador global (responsável por definir a melhor seqüência de portas que serão visitadas
  pelo AGV baseado na descrição do ambiente e nas ordens de locomoção recebidas do
  servidor de ordens) e o planejador local (responsável pelo cálculo dinâmico da trajetória
  entre dois pontos consecutivos da trajetória global, inclusive com o ajuste de rota em tempo
  real, caso uma colisão com algum obstáculo seja detectada).

 Subsistema de Detecção dos Estados das Portas: este sistema está em desenvolvimento e será responsável pela identificação das portas e de seus estados (aberta, fechada ou entreaberta).



Figura 1.1 - O ambiente CONTROLAB - AGV

A Figura 1.1 mostra um esquema lógico da comunicação entre os diversos subsistemas na operação do CONTROLAB-AGV. Inicialmente, uma planta baixa do ambiente é gerada pelo arquiteto, e enviada ao servidor via Internet. Esta descrição do ambiente é então transmitida ao AGV, e também enviada via Internet para todos os clientes. Desta forma, cada cliente pode visualizar o ambiente e a localização do AGV.

Os clientes podem então enviar pedidos de locomoção do AGV ao servidor, especificados por uma localização de destino dentro do ambiente. Todas ordens são armazenadas em uma fila de pedidos no servidor, e transmitidas seqüencialmente ao AGV. Cada ordem será cumprida pelo AGV, que se locomoverá ao seu destino, onde aguardará a próxima ordem.

Durante todo o trajeto percorrido, o AGV transmitirá suas coordenadas ao servidor de ordens, que as enviará a todos os clientes. Desta forma, todo o deslocamento do AGV pelo ambiente poderá ser monitorado por cada estação conectada ao servidor (Figura 1.2).



Figura 1.2 - A arquitetura do CONTROLAB AGV

Quando uma ordem é recebida, o planejador global gera um grafo a partir das informações do ambiente, da posição inicial, e da localização de destino do AGV. Uma **trajetória global** é então determinada como o caminho de menor distância, no grafo, entre os nós de origem e de destino; onde os nós intermediários representam a seqüência de portas a serem atravessadas pelo AGV.

O caminho a ser seguido a cada trecho da trajetória global é determinado pelo planejador local. Esse planejador local utiliza uma função de navegação global para calcular, a cada passo, uma nova direção a ser seguida. As sucessivas coordenadas de cada trecho serão transmitidas para o subsistema de CONTROLE, responsável pela correta movimentação do AGV, passo a passo, até a localização final do trecho em questão.

Esta função de navegação global utiliza campos potenciais, calculando a direção a ser tomada pelo AGV, como o vetor simétrico ao gradiente de uma função potencial, definida na região de cada recinto. Para otimizar o cálculo da função de navegação, todos os potenciais do ambiente são gerados *a priori* e armazenados em vários *grid*'s de valores potenciais, um para cada recinto. Com os valores previamente calculados, a computação da função envolverá apenas operações mais elementares, como será mostrado adiante.

A função de navegação também incorpora o desvio de obstáculos baseando-se em campos potenciais: para cada obstáculo detectado, uma única carga repulsiva pontual será adicionada ao campo de potenciais do recinto. Para que esta operação seja o mais simples e eficiente possível, um único *grid* adicional de potenciais gerados por esta carga repulsiva será gerado antecipadamente.

Com todos os potenciais previamente calculados, os potenciais do recinto poderão ser atualizados rapidamente, apenas adicionando-se os respectivos valores dos dois campos previamente calculados. Desta forma, a nova direção do movimento é calculada rapidamente, permitindo a navegação do AGV com o desvio de obstáculos em tempo real.

#### 1.1.1.2 Planejador Global de Trajetória

Entre dois pontos sucessivos da trajetória, dada pelo campo potencial, o AGV segue uma curva polinomial de 5<sup>a</sup> ordem. Esta curva deve ser suave, garantindo que as restrições de cinemático do veículo sejam satisfeitas. As coordenadas desta curva devem ser enviadas para o subsistema de controle, responsável pela correta movimentação do AGV, passo a passo até a localização final do trecho em questão.

O planejador global é responsável por gerar e analisar todos os caminhos que atinjam a localização de destino especificada na ordem recebida, considerando todas as possibilidades de trajeto do AGV pelas portas e recintos do ambiente. A partir desta análise, o planejador global irá gerar uma trajetória global para o veículo, que será transmitida ao planejador local.

O planejamento global é baseado em um **grafo de conectividade** – um grafo ponderado, gerado a partir de informações:

- Da posição inicial do AGV;
- Da planta baixa do ambiente;
- Da posição final desejada.

A trajetória global será então escolhida como o caminho de menor custo entre as posições inicial e final do AGV neste grafo, cuja construção é descrita a seguir.

Sejam  $S(x, y) \in G(x, y) - Start position \in Goal position – as localizações das posições inicial e final do AGV, respectivamente. Sejam <math>D_i(x, y) - Doors – a localização do ponto médio de cada porta do ambiente. O conjunto de vértices do grafo conterá nós associados aos pontos$ *S*,*G* $, e a cada porta <math>D_i$  do ambiente. Cada aresta entre dois destes nós será ponderada com um custo proporcional à distância entre as localizações associadas aos respectivos nós.

As arestas do grafo serão então geradas da seguinte forma. Para cada par de portas que se comunicam com um recinto em comum, a respectiva aresta será incluída no grafo, "conectando-se" as portas correspondentes. Isto indica que o recinto pode ser "percorrido de uma porta à outra" pelo AGV. Arestas adicionais serão incluídas entre o nó S e cada "porta" presente no recinto inicialmente ocupado pelo AGV, indicando todas as possibilidades de saída do recinto. Processo análogo irá gerar arestas correspondentes ao recinto de destino, conectando-se suas portas ao nó G. A partir de todas as arestas geradas, com seus custos estabelecidos, calcula-se então o caminho mais curto entre os nós S e G, determinando-se a trajetória global.

A Figura 1.3 ilustra um exemplo em que o veículo terá que se locomover por um ambiente contendo diferentes recintos. A planta baixa do ambiente, previamente armazenada no AGV, é mostrada na Figura 1.3(a), onde são também indicados todos os possíveis caminhos entre *S* e *G*. O grafo de conectividade correspondente é ilustrado na Figura 1.3(b), onde é mostrado, em destaque, o caminho mais curto,  $S \rightarrow D_{15} \rightarrow D_{2a5} \rightarrow D_{2a2b} \rightarrow G$ , com um custo total de **294.3**.



Figura 1.3 - (a) Planta baixa, com origem e destino do caminho - (b) grafo de conectividade, incluindo S e G

No exemplo anterior, pode-se observar que a trajetória global é composta por quatro segmentos:  $(S, D_{15})$ ,  $(D_{15}, D_{2a5})$ ,  $(D_{2a5}, D_{2a2b})$ , e  $(D_{2a2b}, G)$ . Cada segmento é caracterizado por um par de vértices, e corresponde a uma aresta do menor caminho entre S e G, no grafo de conectividade.

Observa-se que cada aresta do caminho determina um trecho da trajetória totalmente contido em um único recinto do ambiente (ainda no exemplo anterior, o segmento " $(D_{15}, D_{2a5})$ " indica um trecho em que o veículo percorrerá o recinto "**5**"). Pode-se então representar a trajetória global como um conjunto de trechos consecutivos, caracterizados pelos seus dois pontos extremos e por algum recinto do ambiente.

A próxima etapa do planejamento de trajetória consistirá, então, na divisão da trajetória global em trechos consecutivos, a serem percorridos pelo AGV. Cada trecho será então transmitido ao planejador local, que será responsável pelo efetivo deslocamento do AGV pelo recinto, até a posição desejada. Este processo será repetido para cada trecho do caminho, ate que a posição final *G* seja alcançada pelo veículo.

Observa-se que nenhuma informação sobre possíveis obstáculos estará disponível até que o AGV percorra o ambiente (e o subsistema de detecção de obstáculos indique tal informação). Será então de responsabilidade do planejador local o cálculo do desvio de obstáculos detectados, de forma que cada trecho seja corretamente percorrido.

#### 1.1.1.3 Planejador Local de Trajetória

A grande inovação deste trabalho (LOPES et al, 2001), consiste na utilização de campos potenciais previamente gerados e armazenados, tendo como conseqüência, que a função de navegação poderá ser rapidamente adaptada cada vez que um obstáculo desconhecido obstrua a passagem do veículo. Este cálculo envolve operações de pequena carga computacional, possibilitando sua utilização para o planejamento em tempo real.

O planejador local é responsável pela trajetória do AGV dentro de cada recinto contido na planta baixa. Logo, dada a posição atual do AGV, o planejador calcula, a cada passo, as novas coordenadas que definirão o trajeto do AGV bem como sua direção. Este cálculo é determinado por uma **função de navegação global**, utilizando campos vetoriais potenciais. A função de navegação ainda incorpora, também baseado em campos vetoriais, o desvio de rota caso algum obstáculo seja detectado.

#### 1.1.1.4 A Teoria de Campos Potenciais no Planejamento de Trajetória

A aplicação da teoria de campos potenciais (EPSTEIN, 1962) para o problema de navegação tem sido muito estudada em robótica (LATOMBE, 1991); especialmente a importância do uso de funções harmônicas na geração de campos potenciais artificiais (KIM, 1991) (KIM, 1992), como mostrado a seguir.

Uma função f, definida em um domínio aberto e conexo  $\Gamma \subset \mathbb{R}^n$ , e pertencente à classe  $\mathbb{C}^2$ (·), são dita harmônica em  $\Gamma$ , se  $\nabla^2 f = 0$ , onde  $\nabla^2 = \partial^2 / \partial x_1^2 + \ldots + \partial^2 / \partial x_n^2$ .

A importância da utilização de funções harmônicas deve-se às suas importantes propriedades:

- · Qualquer combinação linear de funções harmônicas é harmônica;
- Valores de mínimo ou máximo de uma função harmônica ocorrem nas fronteiras do seu domínio;
- Aplicando-se transformações de rotação, escala, espelhamento ou translação, a uma função harmônica, esta continua a ser harmônica.

A utilização de funções harmônicas tem sido aplicada a diversos trabalhos em projetos do ambiente CONTROLAB (SILVEIRA, 1997), (AUDE, 1999), (LOPES, 2001a), (LOPES, 2001b). No CONTROLAB-AGV, devido ao formato cilíndrico do veículo, o espaço de configuração representa o AGV como um ponto no R<sup>2</sup>, onde os obstáculos serão restrições aos valores de cada coordenada.

O planejamento de trajetória, baseado em campos potenciais, pode então ser resumido na construção de uma função potencial que determina um campo vetorial. Este campo indica a direção do robô, a cada instante, a partir de sua localização. Esta **função de navegação<sup>1</sup>** é uma função harmônica  $\phi$ , tal que o vetor  $-\nabla \phi$  indicará a direção do movimento do robô (KHATIB, 1986).

#### 1.1.1.5 A Geração de uma Função de Navegação Global para o AGV

Como mencionado, o espaço de configuração da função de navegação será um subconjunto do  $R^2$ , mais especificamente o retângulo determinado pelo recinto por onde o veículo se locomove. A posição do AGV será considerada como o centro do eixo de tração do veículo. Seja então um recinto do ambiente, tal como ilustrado pela Figura 1.4, cujo comprimento e largura sejam respectivamente iguais a **d** e **b**, e com uma porta de saída localizada em (**d**, **y**<sub>0</sub>), de largura  $\Delta$ **y**.



Figura 1.4 - Espaço de configuração no R<sup>2</sup>

A construção de uma função de navegação global para este recinto será dada inicialmente pelo cálculo de uma função potencial de navegação inicial  $F_0$  (x, y), que é a solução do seguinte problema de Dirichlet para a equação de Laplace:

• 
$$\partial^2 \mathbf{u} / \partial \mathbf{x}^2 + \partial^2 \mathbf{u} / \partial \mathbf{y}^2 = 0$$

<sup>&</sup>lt;sup>1</sup> Na função de navegação as únicas informações relevantes são as dimensões do recinto e a localização da porta de saída. Como estas informações estão disponíveis antes da movimentação inicial do veículo, os valores de  $F_0(x,y)$  podem ser gerados e armazenados antecipadamente

• 
$$\mathbf{u}(\mathbf{x}, \mathbf{y}) = 0$$
, para  $\mathbf{x} = 0 \in 0 = \mathbf{y} = \mathbf{b}$  ou  
 $\mathbf{y} = 0 \in 0 = \mathbf{x} = \mathbf{d}$  ou Equação (1)  
 $\mathbf{y} = \mathbf{b} \in 0 = \mathbf{x} = \mathbf{d}$   
•  $\mathbf{u}(\mathbf{d}, \mathbf{y}) = \mathbf{u}_0$ , se  $0 < \mathbf{y}_0 = \mathbf{y} = \mathbf{y}_0 + \Delta \mathbf{y}$   
 $= 0$  caso contrário  
onde  $\mathbf{u}_0 < 0$ 

A solução da Equação (1), onde o potencial na região da porta de saída tem um valor negativo  $\mathbf{u}_0$ , e para os limites do retângulo (paredes do recinto) é fixado em 0, é dada por:

$$\mathbf{u}(\mathbf{x},\mathbf{y}) = \sum E_n \sinh(n\pi \mathbf{x}/\mathbf{b}) \sinh(n\pi \mathbf{y}/\mathbf{b}),$$
 onde

$$\mathbf{E}_{n} = 2\mathbf{u}_{0} \left[ \cos(n\pi \mathbf{y}_{0}/\mathbf{b}) - \cos(n\pi(\mathbf{y}_{0}+\Delta \mathbf{y})/\mathbf{b}) \right] / \pi n \sinh(n\pi \mathbf{d}/\mathbf{b})$$

A solução da Equação (1) é uma função de navegação, onde as únicas informações relevantes são as dimensões do recinto e a localização da porta de saída. Como estas informações estão disponíveis antes da movimentação inicial do veículo, esses valores (F0 (x, y)) podem ser gerados e armazenados antecipadamente. Para cada recinto do ambiente, então, será gerado um *grid* retangular com os valores dos potenciais em toda a sua região. Desta forma, a direção a ser tomada pelo AGV poderá ser computada rapidamente.

#### 1.1.1.6 Adaptação da Função de Navegação para o Desvio de Obstáculos

O desvio de obstáculos é implementado utilizando-se uma função potencial harmônica radial  $\phi(x, y)$ , cujo centro de simetria cilíndrica é gerado por uma carga repulsiva pontual e cuja equação é dada por:

 $\phi(x, y) = (\lambda/2\pi) \log r$ , onde *r* é a distância Euclidiana do ponto (x, y) ao centro de simetria, e

 $\lambda$  uma constante negativa.

Para cada situação de colisão eminente, esta única função potencial  $\phi$  é associada ao obstáculo detectado, que é modelado como um ponto no *grid*, como se segue.

Quando o AGV detecta uma situação de colisão, o ponto do obstáculo mais próximo ao AGV é identificado. O centro de simetria de  $\phi$  é então "deslocado" para o ponto do *grid* mais próximo do ponto do obstáculo, denominado **ponto de colisão**. Esta nova função  $\phi_c$  que será gerada pelo reposicionamento da carga pontual, é então adicionada à função de navegação do recinto. Desta forma, existe uma "adaptação" da função de navegação global, adicionando-se a este a nova função potencial  $\phi_c$ .

Para que a adaptação da função de navegação seja efetuada rapidamente, os potenciais originais de  $\phi(x, y)$  também são previamente gerados e armazenados em um único *grid* adicional. Quando um obstáculo for detectado, os valores do *grid* da função original  $\phi$  serão adicionados, em posições relativas a partir do ponto de colisão, aos potenciais armazenados no *grid* do recinto em questão.

Após **n** obstáculos detectados, a função de navegação  $F_n$ , é adaptada de acordo com os n pontos de colisão detectados:

$$F_n = A_n F_0 + \sum A_c \phi_c, \qquad \text{Equação (2)}$$

onde  $A_n$  e  $A_c$  são fatores de multiplicação que variam conforme a movimentação do AGV.

 $A_n$  cresce com o número obstáculos detectados;  $A_c$  é definido como uma função do gradiente atualmente verificado no ponto de colisão considerado. Como visto, a adaptação da função de navegação será então realizada pela simples adição de dois campos potenciais, cujos valores são previamente conhecidos, operação que demanda uma carga computacional relativamente pequena.

#### 1.1.1.7 Resultados Experimentais

Para uma avaliação de como o planejador local controlaria o efetivo deslocamento do AGV por um determinado ambiente, experimentos foram realizados, considerando-se um recinto retangular com uma única porta de saída. Esses experimentos foram realizados sem considerar os limites supostos pela dinâmica do veículo, levando-se em conta apenas a teoria aplicada ao AGV para definir a trajetória descrita pelo centro de eixo traseiro.

A Figura 1.5 mostra as linhas de fluxo do campo potencial armazenado no *grid* original do recinto considerado, sem que nenhuma informação sobre obstáculos seja considerada. De acordo com o desenvolvimento matemático discutido na seção anterior, foi definido um baixo valor para o potencial sobre a porta de saída ( $\mathbf{u}_0 = -2$ ), sendo fixado em 0 nas paredes que delimitam o recinto.



Figura 1.5 - Campo potencial original de um recinto

A Figura 1.6 mostra as linhas equipotenciais do campo potencial mostrado acima. São mostrados em destaque alguns pontos do *grid*, com seus respectivos potenciais logo abaixo.



Figura 1.6 - Linhas Equipotenciais do campo original

O movimento real do AGV é ilustrado na Figura 1.7, onde o veículo é representado como um círculo e a "bengala eletrônica" através de um segmento circular distante 40cm do seu corpo. Nesta situação, o deslocamento inicial do AGV é determinado pelas linhas de fluxo do campo original. Durante o seu percurso, os obstáculos serão detectados pela bengala eletrônica. Novos campos potenciais serão então gerados pelas cargas pontuais, posicionadas nos pontos de colisão (mostrados em destaque), e serão adicionados ao campo original. A função de navegação adaptada determinará, então, uma nova rota para o AGV, livre de colisões com os obstáculos.



Figura 1.7 - Locomoção do AGV em tempo real

O planejador de trajetória descrito visa implementar o desvio de obstáculos em tempo real, utilizando para isso uma estratégia similar à utilizada por pessoas cegas enquanto se locomovem. O subsistema de detecção de obstáculos implementa, utilizando sensores de infravermelho, um mecanismo similar a uma "bengala eletrônica".

O planejamento de trajetória é realizado em duas etapas. Na primeira, uma trajetória global é definida a partir da descrição do ambiente e da posição final desejada. Este planejamento é realizado *a priori*, determinando uma trajetória composta por diversos trechos que conduzirão o veículo pelas portas e recintos do ambiente.

Na segunda etapa, um planejamento local determina o efetivo deslocamento do AGV para cada trecho da trajetória global. Este planejamento utiliza campos potenciais que determinam, a cada passo do AGV, a sua trajetória naquele instante.

Para o planejamento local, foi apresentado o modelo de uma função de navegação global que utiliza campos potenciais previamente gerados e armazenados em *grid*'s associados aos recintos do ambiente. Esta função de navegação é adaptada para implementar o desvio de obstáculos em tempo real. Para isto, um outro campo potencial será gerado, baseado em uma carga pontual repulsiva, e também previamente armazenado em um *grid* adicional.

Cada obstáculo detectado durante o percurso será então modelado como um ponto do *grid*, onde será posicionada uma carga pontual. Esta operação envolve a adição de dois campos potenciais, cujos valores já estão armazenados, consumindo baixo custo computacional, o que permite a utilização e adaptação da função de navegação para tempo real.

Os experimentos para o planejamento local de trajetória se mostraram promissores para sua utilização em tempo real. Porém não foram levadas em consideração as limitações de cinemática do veículo autônomo. A relevância da estratégia aqui apresentada é baseada nos seguintes fatores:

- · Demanda uma pequena carga computacional;
- · Pode ser utilizada para o desvio de obstáculos de diferentes formas;
- Requer sensores de baixo custo;
- · Utiliza uma função de navegação livre de mínimos locais.

## 1.2 Simulação

A simulação trata-se de um ferramental disponibilizado pela área de pesquisa operacional que permite a geração de cenários, a partir dos quais pode-se: orientar o processo de tomada de decisão, obter análises e avaliações de sistemas e propor soluções para a melhoria de performance. Sendo que, todos estes procedimentos podem ter por conotação parâmetros técnicos e, ou, econômicos.

Com os avanços na área de informática, modernos equipamentos e novas linguagens de programação e de simulação tem permitido empregar a técnica de simulação nas diversas áreas do conhecimento humano.

Esses fatos têm propiciado: projetar e analisar sistemas industriais, avaliar performance de hardware e software em sistemas de computação, analisar desempenho de armas e estratégias militares, determinar freqüência de pedidos de compra para recomposição de estoques, projetar e administrar sistemas de transportes como: portos e aeroportos, e configurar sistemas de atendimento em hospitais, supermercados e bancos.

No caso específico das engenharias, a adoção da técnica de simulação tem trazido benefícios como:

- · A previsão de resultados na execução de uma determinada ação,
- · A redução de riscos na tomada decisão,
- · A identificação de problemas antes mesmo de suas ocorrências,
- A eliminação de procedimentos em arranjos industriais que não agregam valor à produção,
- A realização de análises de sensibilidade,
- A redução de custos com o emprego de recursos (mão-de-obra, energia, água e estrutura física) e;
- A revelação da integridade e viabilidade de um determinado projeto em termos técnicos e econômicos.

#### 1.2.1 Aplicações e Benefícios da simulação

Diversos benefícios podem ser alcançados durante uma simulação em diversas fases de concepção e projeto de qualquer sistema que se queira implementar.

#### Concepção do sistema

Diferentes concepções podem ser avaliadas com o auxílio da simulação. Diferentes equipamentos e/ou conceitos de funcionamento podem ser testados, de maneira relativamente rápida e a baixo custo. A simulação permite um refinamento do sistema, desde sua fase de concepção.

#### Escolha da Ferramenta de Simulação

A escolha da ferramenta considera fatores tais como:

- Características da simulação a ser realizada;
- · Características das ferramentas disponíveis;
- · Ambiente disponível para simulação;
- · Capacitação da equipe de projetistas que realizarão a implementação do simulador;
- · Orçamento disponível para compra da ferramenta caso não exista disponível;
- Tempo disponível para realizar a simulação.

#### Implementação do sistema

Durante a fase de projeto e implementação, a simulação é uma importante ferramenta para o detalhamento do sistema. Algoritmos de controle, detalhes dos equipamentos, detalhes da interface com outros sistemas e procedimentos próprios do cliente podem ser modelados e analisados, gerando decisões ou alterações no sistema que garantam seu desempenho.

#### Modernização e atualização do sistema

A introdução de novos equipamentos, novas interfaces ou formas de controle de um sistema já implantado podem ser avaliadas com o uso da simulação. Diferentes cenários de utilização do sistema podem ser analisados, permitindo avaliação precisa dos pontos que precisam modernização ou atualização.

#### Coleta de informações do problema

De posse dos objetivos gerais da simulação, um levantamento completo das informações disponíveis sobre o problema é realizado: revisão de desenhos, *layout* e especificações de equipamentos e sistemas de controle, entendimento das interfaces com outros sistemas, operadores e dos processos envolvidos. Toda informação é registrada e documentada, sendo posteriormente revisada em conjunto com o cliente a fim de garantir sua integridade e a integridade do modelo.

#### Desenvolvimento do modelo

Nesta etapa, o modelo é efetivamente construído utilizando uma linguagem de simulação apropriada para o caso específico. O código do programa é amplamente comentado e desenvolvido de forma modular. Isto garante a documentação do projeto e facilita eventuais revisões ou modificações do modelo.

#### Validação do modelo

Com o código e o programa prontos, é necessário certificar-se de que o modelo construído representa o sistema que se deseja estudar e que as respostas ou informações obtidas com ele são representativos. A validação é feita testando-se o comportamento do modelo sob situações conhecidas e de domínio claro.

#### Análise de sensibilidade

Durante esta etapa são realizadas experiências com o modelo. Através da variação de parâmetros internos do modelo é possível alterar as condições de funcionamento ou reações do sistema e avaliar o impacto sobre toda instalação. Ex: Alterando alguns parâmetros de configuração de um ambiente definido por um sistema, podem ocorrer grandes impactos nos resultados apresentados.

#### Documentação

A última etapa consiste na geração de relatórios e conclusões a respeito dos resultados da simulação, incluindo as recomendações. Os resultados são documentados, incluindo desde a fase de levantamento de dados até os relatórios finais, de modo a se gerar um material único a respeito do modelo e sua análise.

## 1.3 Objetivos da tese

Como já foi visto, foi proposto em (LOPES et al, 2001) um projeto que consiste no desenvolvimento do CONTROLAB-AGV, um veículo autônomo que opera sobre um ambiente controlado definido por uma planta baixa conhecida. Este veículo recebe comandos da Internet através de rede sem fio e é capaz de se deslocar neste ambiente, executando a tarefa desejada. Nesse projeto, o veículo autônomo (AGV), desvia de qualquer obstáculo que venha interceptar sua trajetória.

Foram realizados experimentos e foram encontrados resultados promissores para o procedimento proposto.Os resultados promissores serviram para avaliar essas funções e mostrar o que realmente acontece com os campos potenciais de um determinado ambiente pelo qual o AGV atuará.

No entanto, para visualizar os resultados dos experimentos de maneira mais correta e realísticamente possível, aplicando a definição de uma trajetória contínua, necessita-se a construção de um simulador de veículos autônomos.

O simulador em questão, é desenvolvido devido a grande necessidade de dinamizar o estudo do comportamento do AGV. Através do mesmo, possibilita-se a análise de pontos importantes da trajetória definida, monitorar e/ou alterar todas as variáveis relacionadas à cinemática do movimento do veículo e conseqüentemente minimizar os testes exaustivos nas experiências práticas com o sistema real do AGV.

Além dessas características, o simulador CONTROLAB AGV deve possibilitar testes de diversas aplicações envolvendo o AGV, em diferentes ambientes de simulação.

O problema em questão consiste, com o auxílio do simulador, definir um conjunto de parâmetros para a construção dos campos potenciais que gerem trajetórias suaves e que atendam a cinemática do veículo autônomo (AGV).

O problema de planejamento de trajetórias surge em qualquer aplicação, que dispondo de um veículo, físico ou simulado, pretenda fazer uso de mobilidade num espaço com obstáculos. Existem vários métodos para a planificação de caminhos, sendo que a escolha depende do grau de complexidade e da qualidade do meio aonde o veículo irá se movimentar, bem como de características do próprio veículo.

Como forma de avaliação do planejamento usam-se critérios como o comprimento, o tempo, a distância aos obstáculos ou a suavidade dos caminhos. Em (LATOMBE, 1991) encontram-se diversos métodos clássicos para o planejamento de trajetórias.

Foi proposto em (AUDE et al, 2003), a aplicação de uma trajetória contínua para um veículo autônomo com uma bengala eletrônica para detectar obstáculos desconhecidos. Essa trajetória contínua é dada por uma função polinomial de quinta ordem e definida entre dois pontos com valores potenciais calculados. As condições de interpolação da função polinomial de quinta ordem harmonizam a trajetória definida pelo AGV, atendendo às suas restrições dinâmicas e permitindo que ele desempenhe movimentos complexos. Com isso é definida uma seqüência de pontos, que devem ser interpolados de tal maneira que os vetores velocidade e aceleração sejam contínuos e que atendam às restrições de cinemática do AGV.

Scheuer and Fraichard desenvolveram em (SCHEUER, FRAICHARD, 1996) um plano de trajetória contínua, livre de colisão, os quais são baseados em curvas das quais as curvaturas são funções lineares do tamanho de seus arcos.

Através do simulador será possível estudar o comportamento de veículos autônomos para diferentes granularidades do grid de potenciais do campo em cada ambiente. Assim otimizam-se os testes realizados no veículo autônomo e a análise do plano de navegação do veículo autônomo, obtendo-se trajetórias contínuas mais suaves para atender as restrições de cinemática do veículo autônomo.

## 1.4 Conteúdo da tese

No capítulo 2, os modelos de simulação existentes e o formalismo de especificação do modelo dinâmico utilizado pelo simulador CONTROLAB AGV são apresentados. O capítulo 3 descrevve o sistema simulado e a operação do simulador CONTROLAB AGV. Esse simulador será usado como ferramenta para a análise da definição do novo plano de trajetória contínua para o veículo autônomo (AGV). A descrição do modelo de movimento do AGV usando trajetória contínua e da aplicação da técnica que será utilizada para resolver o problema das restrições cinemáticas no veículo autônomo é discutida no capítulo 4. O capítulo 5 descreve os testes realizados no simulador CONTROLAB AGV levando em consideração a redefinição dos cálculos para a geração da nova trajetória contínua no veículo, e a análise dos resultados apresentados. Finalmente, no capítulo 6, a conclusão desse trabalho é discutida e futuros desenvolvimentos que podem ser realizados no simulador CONTROLAB AGV são apresentados.

A simulação trata-se de uma ferramenta de apoio à tomada de decisão, a qual possibilita a redução de riscos e custos envolvidos em um processo de mudança, uma vez que os mesmos podem ser pré-determinados antes da efetiva mudança ocorrer. Segundo Hollocks,

"a simulação computacional é uma técnica de pesquisa operacional, que envolve a criação de um programa computacional, que representa alguma parte do mundo real de forma que experimentos no modelo original predizem o que acontecerá na realidade." (HOLLOCKS, 1992, p.12).

Assim, a idéia básica é tornar o modelo, um veículo para inferir questões do tipo "o que aconteceria se...?".

Simulação pode ser definida como o processo de projetar um modelo de um sistema real e conduzir experimentos com esse modelo, com o propósito de compreender o comportamento do sistema ou avaliar várias estratégias para a operação do sistema.

O computador permite o uso de sofisticados recursos gráficos, matemáticos e lógicos para a construção do modelo do sistema, proporcionando ao mesmo tempo flexibilidade, agilidade e confiabilidade no projeto. O uso de simulação reduz assim riscos de novas implementações e dá suporte a decisões complexas, por vezes envolvendo elevados investimentos.

Modelos de simulação são empregados, por exemplo, na programação de produção de uma empresa, na análise de problemas de uma célula de manufatura ou de um sistema de movimentação de materiais.

A simulação permite a análise e o teste de diferentes alternativas de funcionamento de um sistema, com variações de parâmetros diversos, formas de controle, priorização de eventos, diferentes equipamentos, velocidades e ciclos de trabalho.

Variações aleatórias em processos e ciclos de trabalho podem ser incorporadas ao modelo, aproximando-o do funcionamento real do sistema. Com a simulação, gargalos e limitações de capacidade podem ser revelados ainda na fase de concepção e projeto. Soluções alternativas podem ser geradas, testadas e certificadas, garantindo o desempenho da instalação muito antes de sua implementação física.

Portanto, uma simulação abrange dois grandes aspectos: modelagem e experimentação e tem dois grandes objetivos: compreender e prescrever.

## 2.1 Processo de simulação

Após a criação do modelo já se pode codificá-lo para poder simulá-lo, porém essa codificação vai representar apenas de 30% a 40% do esforço total em um estudo de simulação.

Existem alguns elementos importantes de um estudo de simulação bem sucedido:

- Conhecimento de metodologias de simulação, modelos estocásticos, probabilidade e estatística;
- · Formulação correta do problema;
- · Boas informações sobre o funcionamento do sistema e sua lógica;
- · Escolha do software de simulação apropriado;
- Utilização de procedimentos estatísticos apropriados para interpretação dos resultados.

Existem também algumas falhas bem comuns:

- Na obtenção de um conjunto bem definido de objetivos no início do estudo da simulação;
- Nível inadequado de detalhes (pouco ou muito detalhamento);
- De comunicação com a gerencia do sistema a ser simulado durante o estudo da simulação;
- Interpretações equivocadas por parte da equipe da simulação da operação do sistema a ser simulado;
- · De compreensão da simulação por parte da gerencia;
- · Tratar a simulação de forma amadora, como um exercício de curso;
- Na formação de equipe com conhecimentos de metodologias e técnicas de simulação;
- Na obtenção de dados representativos do comportamento do sistema;
- · No software de simulação (muito complexo e com documentação inadequada);
- Crença de que software de simulação sofisticado e com recursos amigáveis, prescindem de conhecimentos técnicos da teoria de simulação;
- No uso de animação.

De acordo com (PEGDEN, 1995), a simulação computacional de sistemas também pode ser definida como um processo de projetar um modelo de um sistema real e de procedimentos de experimentos com este modelo. Isso é feito com o propósito de conhecer o comportamento do sistema e/ou avaliar estratégias para a sua operação. Pode-se imitar o comportamento de, praticamente, qualquer tipo de operação ou processo do mundo real.

## 2.2 Metodologia de simulação

Diante dos aspectos positivos e negativos do processo de simulação é possível definir uma metodologia que permite avaliar se o uso de um projeto de simulação, em determinados casos ou sistemas, pode ser totalmente viável. Se for viável até que ponto os resultados são importantes para um determinado sistema real. A Figura 2.1 mostra um exemplo de metodologia de simulação que pode ser aplicada em qualquer modelo de simulação.



Figura 2.1 - Um exemplo de metodologia de simulação aplicável em qualquer modelo de simulação

Após a formulação do problema, definem-se as características fundamentais para a definição do sistema. Com essas características definidas, avalia-se o uso da simulação. No caso afirmativo formula-se o modelo, através de algum formalismo, e em seguida preparam-se os dados que serão usados na implementação do modelo de simulador. Na implementação verifica-se a validação do modelo através dos dados obtidos pelo simulador e seu comportamento.

Confirmada a validação do modelo, aplicam-se planejamentos estratégicos e táticos através dos resultados apresentados, para analisar, após a experimentação, se os dados obtidos na simulação são aderentes aos dados obtidos de sistemas reais utilizados como referência. Os resultados utilizáveis são documentados e em seguida implementados no sistema real.

## 2.3 Conceitos de modelagem de sistemas de simulação

Um modelo é uma especificação do sistema em qualquer nível de discussão. O conceito mais comum de um modelo de simulação é definido como um conjunto de instruções, regras, equações para gerar um comportamento de entrada e saída. Os modelos formam os componentes básicos em modelos mais complexos.Podem-se distinguir algumas categorias de modelos de simulação:

- Modelo de Simulação Estático representa o sistema em um instante determinado, isto é, o modelo não varia ao longo do tempo.
- · Modelo de Simulação Dinâmico representa a evolução do sistema ao longo do tempo.
- Modelo de Simulação Determinístico o sistema não depende de nenhuma variável probabilística (aleatória).
- Modelo de Simulação Estocástico o sistema depende de variáveis probabilísticas (aleatórias). Sistemas de computação, de redes de comunicação e de serviços a clientes, entre outros, estão nesta categoria. Em geral utilizam filas de chegada de tarefas em que as chegadas ocorrem de acordo com alguma distribuição de probabilidade.
- Modelo de Simulação Discreto o sistema depende de variáveis que assumem valores discretos, isto é, em um domínio de valores finitos ou enumeráveis tais como o conjunto de números inteiros.
- **Modelo de Simulação Contínuo** depende das variáveis que assumem valores contínuos, isto é, em um domínio de valores contínuos tais como o conjunto de números reais.

Como um conjunto de instruções, um modelo precisa de muitos agentes capazes de seguir instruções e gerar um comportamento. Esses agentes são chamados de simuladores. Um simulador é qualquer sistema de computação capaz de executar um modelo para gerar seu comportamento.

Separando o conceito de modelo e simulador, provemos um grande número de benefícios para o sistema:

- O mesmo modelo, expressado em um formalismo, deve ser executado por diferentes simuladores, abrindo assim o caminho para portabilidade e interoperabilidade num alto nível de abstração.
- Algoritmos simuladores para vários formalismos devem ser formulados e sua exatidão rigorosamente estabelecida.
- Os recursos requeridos para simular corretamente um modelo sustentam a medida de sua complexidade.

Os simuladores podem operar em duas modalidades de tempo:

- Tempo real A escala de tempo é a real, isto é os eventos ocorrem e são tratados na mesma escala de tempo correspondente ao sistema real. Simuladores de jogos ou para treinamento se enquadram nesta categoria. Nestes sistemas um operador humano interage com o simulador em tempo real.
- Tempo simulado Não acompanha a escala de evolução do tempo real. Um ano do tempo de simulação pode decorrer em poucos segundos de processamento. São utilizados para análises de desempenho em que o interesse é pelas medidas de desempenho.

A exatidão básica de um simulador nada mais é que a relação dos resultados apresentados na simulação com os resultados do modelo dentro de um conjunto satisfatório de experimentos. Simulação correta de um modelo é garantida se forem gerados, fielmente, a trajetória de saída, o estado e a trajetória de entrada do modelo dado.

Na prática os simuladores são construídos para executar não somente um modelo, mas uma família de possíveis modelos. A flexibilidade é necessária ainda mais se for aplicável para um conjunto de aplicações.

Existem dois aspectos principais para a teoria da modelagem:

- Níveis de especificação de sistema são os níveis nos quais podem descrever como os sistemas comportam-se e os mecanismos que fazem seu trabalho e a maneira que eles fazem.
- Formalismo de especificação de sistemas são os tipos de estilos de modelagem, como contínuos ou discretos, que modeladores podem usar para construir modelos de sistemas.

No formalismo de especificação de sistemas, a teoria de sistema diferencia-se entre estrutura do sistema (a constituição interna de um sistema) e o seu comportamento (sua manifestação externa).

Visto como uma caixa preta, o comportamento externo de um sistema é a relação que este impõe entre o histórico do tempo de entrada e o histórico do tempo de saída. O comportamento de entrada e saída do sistema consiste de pares de registro de dados (segmento de tempo de entrada e segmento de tempo de saída) colhidos de um modelo ou sistema real.

A estrutura interna de um sistema inclui seu estado, o mecanismo de transição de estado (ditando como entradas transformam estados correntes dentro de estados sucessores) e também o estado de saída. Conhecendo a estrutura do sistema real é possível deduzir (analisar ou simular) seu comportamento.

Um conceito de estrutura importante é o de decomposição, ou seja, como um sistema deve ser quebrado em componentes de sistemas. Um segundo conceito é o de composição, isto é, como componente de sistemas devem ser acoplados juntos para formar um sistema grande.

## 2.4 Formalismo de especificação de sistemas de tempos Discretos (DTSS)

De acordo com (ZEIGLER, 2000), o formalismo DTSS (*discrete time system specification*) é mais intuitivo para a compreensão de todas as formas dos modelos dinâmicos. Esse formalismo assume um modo de execução gradativo, como mostra a figura 2.2.

Num instante de tempo particular o modelo está num estado particular e isto define como esses estados mudam e o que pode ocorrer no próximo instante. O próximo estado depende do estado corrente e também de qual é a influência do ambiente atual.



Figura 2.2 - Execução gradativa de Sistemas de tempos discretos

O sistema de tempos discretos possui numerosas aplicações. As aplicações mais conhecidas são em sistemas digitais, onde o relógio define os passos do tempo discreto. Mas sistemas de tempos discretos são também freqüentemente usados como aproximações de sistemas contínuos. Por exemplo: nessas aplicações é escolhida uma unidade de tempo (1 segundo, 1 minuto ou 1 ano) para definir um relógio artificial e o sistema é representado quando o estado muda de um instante de observação para o outro.

Na construção de um modelo de tempos discretos, precisa-se definir como o estado corrente e a entrada de dados do ambiente determinam o próximo estado do modelo.

Em modelos de tempo discreto o tempo avança em passos discretos, nos quais assume-se que são múltiplos inteiros de períodos básicos tais como 1 segundo, 1 dia ou 1 ano. A transição ou saída de dados da tabela discutida seria interpretada como a especificação do tempo excedente das mudanças de estado, da seguinte forma:

- Se o estado no tempo t é q e se a entrada no tempo t é x então o estado no tempo t+1 será d(q,x) e a saída y no tempo t será λ(q,x).
- d é chamado função de transição de estado e é o conceito mais abstrato para as primeiras três colunas da tabela mostrada na figura 2.3. λ é chamada de função de saída e corresponde as primeiras duas e última colunas.

• As formas abstratas, **d** e  $\lambda$ , constitui a maneira mais geral de mostrar a informação de transição e saída.

Através do exemplo de sistema de tempos discretos abaixo temos:

 $\begin{pmatrix}
\mathbf{d} (0,0) = 0 \\
\mathbf{d} (0,1) = 1 \\
\mathbf{d} (1,0) = 0 \\
\mathbf{d} (1,1) = 1
\end{pmatrix}$ 

Na tabela abaixo é possível mostrar um modelo resumido do exemplo descrito acima.

Estado	Dado de	Próximo	Dado de	
Corrente	Entrada	Estado	Saída	
		(δ)	(λ)	
0	0	0	0	
0	1	1	1	
1	0	0	0	
1	1	1	1	

Figura 2.3 Tabela com um exemplo de sistemas de tempos discretos

Essa tabela pode ser sumarizada mais compactamente como:  $d(q,x) \in \lambda(q,x)$ . A seqüência de estados, q(0), q(1), q(2) é a trajetória de estados. Tendo um estado arbitrário inicial q(0), estados subseqüentes são determinados por q(t+1) = d(q(t),x(t)). Similarmente a trajetória de saída é dada por  $y(t) = \lambda(q(t),x(t))$ .

A especificação de sistemas de tempos discretos é então definida em uma estrutura  $M = \{X, Y, Q, \delta, \lambda, c\}$ .

Aonde

X é o conjunto de valores de entrada

Y é o conjunto de valores de saída,

Q é o conjunto de estados,

 $\delta$ : Q x X  $\blacktriangleright$  Q é a função de transição de estados,

 $\lambda$ : é a função de saída onde  $\lambda$ : Q  $\searrow$  Y (sistemas em função apenas do tempo e do estado) ou  $\lambda$ : Q x X  $\searrow$  Y (sistemas em função do tempo, do estado e dos dados de entrada)

c é uma constante empregada para especificar o tempo base.

## 2.4.1 Especificação do modelo DTSS para o simulador CONTROLAB-AGV

Como já foi visto, o simulador CONTROLAB-AGV consiste de um veiculo autônomo que possui um deslocamento efetivo a partir de uma trajetória global previamente definida. O deslocamento é determinado, em cada trecho, pelo planejador local. O planejador global utiliza uma função de navegação global para calcular, a cada passo, uma nova direção a ser tomada.

Esta função de navegação global utiliza campos potenciais para calcular a direção a ser tomada, através do vetor simétrico ao gradiente de uma função potencial. A função potencial é definida na região de cada recinto. Para otimizar o cálculo da função de navegação, todas as cargas potenciais do ambiente são geradas, a priori, e armazenadas em *grid*'s de valores potenciais (um *grid* para cada recinto).

A função de navegação também incorpora o desvio de obstáculos baseando-se em campos potenciais, ou seja, para cada obstáculo detectado, uma única carga repulsiva pontual será adicionada ao campo de potenciais do recinto através de uma função de adaptação.

Com todos os valores potenciais previamente calculados, os valores potenciais do recinto poderão ser atualizados rapidamente, apenas adicionando-se os respectivos valores dos dois campos previamente calculados. Desta forma, gera-se uma mudança no fluxo do campo potencial do determinado recinto e conseqüentemente uma nova direção do veículo autônomo, desviando do obstáculo detectado.

Com os experimentos que já foram realizados, viu-se a necessidade de se criar um simulador para o CONTROLAB AGV para dinamizar os teste realizados, implantando um plano de definição de uma trajetória suave e contínua.

Através de uma trajetória parametrizada *quintic spline*, ou seja, trajetória que permite interpolação de uma seqüência de pontos arbitrários, pode-se obter uma continuidade na construção dessa trajetória.

Como será visto no capítulo 4, para que essa trajetória seja criada é necessário resolver o problema da interpolação em G<sup>2</sup>. Para isso, é preciso determinar a curva polinomial de ordem mínima que é interpolada entre os pontos inicial  $\mathbf{p}_A = \{\mathbf{x}_A, \mathbf{y}_A\}$  e final  $\mathbf{p}_B = \{\mathbf{x}_B, \mathbf{y}_B\}$  da sub trajetória ou sub-rota.

Em cada um desses pontos estão associados o vetor tangente inicial (associado ao ponto  $p_A$ ) e o vetor tangente final (associado ao ponto  $p_B$ ). Esses vetores são definidos pelos ângulos  $\theta_A$  e  $\theta_B$ (definidos pela direção do AGV) e as curvaturas escalar  $K_A$  e  $K_B$ .

Os dados de interpolação:  $\mathbf{p}_A$ ,  $\mathbf{p}_B$ ,  $\theta_A$ ,  $\theta_B \in [0,2\pi)$ ,  $\mathbf{k}_A \mathbf{e} \mathbf{k}_B \in \Re$ , podem ser determinados arbitrariamente. Para resolver o problema da interpolação, considera-se uma curva polinomial *quintic*  $\mathbf{p}(\mathbf{u}) = [\mathbf{x}(\mathbf{u}) \mathbf{y}(\mathbf{u})]^T$ ,  $\mathbf{u} \in [0,1]$  onde

 $\begin{aligned} \mathbf{x}(\mathbf{u}) &:= \mathbf{x}_0 + \mathbf{x}_1 \mathbf{u} + \mathbf{x}_2 \mathbf{u}^2 + \mathbf{x}_3 \mathbf{u}^3 + \mathbf{x}_4 \mathbf{u}^4 + \mathbf{x}_5 \mathbf{u}^5 \\ \mathbf{y}(\mathbf{u}) &:= \mathbf{y}_0 + \mathbf{y}_1 \mathbf{u} + \mathbf{y}_2 \mathbf{u}^2 + \mathbf{y}_3 \mathbf{u}^3 + \mathbf{y}_4 \mathbf{u}^4 + \mathbf{y}_5 \mathbf{u}^5 \end{aligned}$ 

Os coeficientes  $x_0$ ,  $y_0$ ,  $x_1$ ,  $y_1$ ,  $x_2$ ,  $y_2$ ,  $x_3$ ,  $y_3$ ,  $x_4$ ,  $y_4$ ,  $x_5$ ,  $y_5$  são determinados a partir das condições iniciais para o cálculo da interpolação.

A variável **u** representa o tempo de amostragem ou observação (**ta**) para o cálculo da interpolação, ou seja, para cada **u** (**u** = **ta**) é definida uma nova posição para o veículo autônomo. De acordo com a definição teórica esse tempo varia no intervalo de [0,1], ou seja, para **u** = **0** o veículo autônomo estaria no ponto inicial (**P**<sub>A</sub>) da sub trajetória e para **u** = **1** o veículo autônomo estaria no ponto final da sub trajetória (**P**<sub>B</sub>).

Dentro da especificação desse plano de definição de trajetória (*quintic spline*), que será aplicado no simulador CONTROLAB AGV, consideram-se os seguintes aspectos:

- A planta baixa com as informações do ambiente por onde o veículo autônomo irá gerar sua trajetória é considerado o principal dado de entrada do simulador, já que as funções de navegação e para o cálculo da trajetória dependem dessas informações;
- Os parâmetros do campo potencial e os parâmetros do *Grid* definidos no ambiente são informações importantes para o simulador e são considerados dados de entrada do simulador;
- Ao aplicar a primitiva *quintic spline* é necessário atribuir condições iniciais para o cálculo dos coeficientes que definirão a curva polinomial *quintic*, resolvendo assim o problema da interpolação. Pode-se assumir que as condições iniciais são dados de entrada para a definição da curva polinomial na trajetória do veículo autônomo;
- A carga repulsiva que será adicionada no ponto de colisão onde o obstáculo foi detectado também pode ser considerada como um dado de entrada;
- Ao definir um tempo u de intervalo de domínio teórico de [0,1], temos tempos de amostragem definidos para o cálculo da curva polinomial. A cada u, sendo u = ta (tempo de amostragem) existe uma mudança de posição na trajetória para o veículo autônomo, logo u = ta também é um dado de entrada para o simulador. Como o tempo de detecção de obstáculos está relacionado com o tempo de amostragem, este também é considerado um dado de entrada;

Como saídas do simulador CONTROLAB AGV consideram-se as informações que mostram o comportamento do veículo autônomo durante sua trajetória. Tais informações são:
- A posição atual do veículo autônomo;
- A velocidade atual nas rodas direita e esquerda do veículo autônomo;
- A aceleração atual nas rodas direita e esquerda do veículo autônomo;
- A interface do simulador;
- A trajetória contínua que o veículo autônomo define;
- Os obstáculos detectados;
- O grid de potenciais atualizado após a detecção de um obstáculo e
- As isolinhas que mostram pontos equipotenciais do ambiente.

Na figura 2.4 abaixo é mostrado um gráfico de entradas e saídas para o modelo do simulador CONTROLAB AGV.

## ENTRADAS

```
SAÍDAS
```



Figure 2.4 - Entradas e Saídas do simulador CONTROLAB AGV

O sistema CONTROLAB-AGV contextualiza-se como um sistema de tempos discretos e assim é possível especificá-lo em um modelo do formalismo DTSS clássico da mesma forma.

O modelo do simulador do sistema CONTROLAB-AGV, de acordo com a especificação do formalismo DTSS aplicado por (ZEIGLER, 2000), pode ser definido da seguinte forma:

$$DTSS/_{AGV} = (X, Y, Q, \delta, \lambda, c)$$

Aonde

X é um conjunto de valores de entrada arbitrários, como foi observado anteriormente nele incluem-se os seguintes dados:

As **condições iniciais** da definição da trajetória que será simulada de acordo com a primitiva *quintic spline*;

A planta baixa do ambiente onde o veículo autônomo está definindo sua trajetória;

Os parâmetros do grid e do campo definido;

O passo do AGV;

**O tempo de resposta da bengala e o tempo de amostragem**. Como as condições iniciais são valores arbitrários o seu domínio é dado pelo conjunto dos números reais (**X**=**R**).

Como exemplo, existem os parâmetros definidos como **condições iniciais** para o cálculo dos coeficientes que definem a curva polinomial, resolvendo o problema da interpolação. Esses parâmetros são:

- $\mathbf{p}(\mathbf{0}) \in \mathbf{p}(\mathbf{1})$  representam a curva polinomial nos pontos inicial ( $p_A$ ) e final ( $p_B$ ) normalizados em um tempo que varia de 0 a 1, ou seja, o módulo do vetor velocidade definido nos pontos inicial e final da trajetória;
- **K**<sub>A</sub> e **K**<sub>B</sub> representam a curvatura nos pontos inicial (p<sub>A</sub>) e final (p<sub>B</sub>);
- e<sub>1</sub>(0) e e<sub>1</sub>(1) representam os vetores tangentes no ponto inicial e final, sendo que e<sub>1</sub>(u) é dado por p'(u)/||p'(u)||, onde p'(u) representa a derivada da curva polinomial p(u), que é o módulo do vetor velocidade.

Y é um conjunto de valores de saída arbitrários, nele incluem-se os valores de saída mostrados anteriormente. Esses valores irão mostrar o comportamento e as características do AGV durante a definição da trajetória no simulador.

Então **Y=P(u) (um conjunto arbitrário)**. Como são valores arbitrários o seu domínio é dado pelo conjunto dos números reais (Y=R). Como exemplo, na aplicação do simulador CONTROLAB AGV existem os seguintes valores de saída:

Nova posição calculada do AGV (x(u),y(u));

O módulo do vetor velocidade (|**P'(u**)|);

e os módulos dos vetores aceleração tangencial (|P''(u)|) e aceleração normal.

**Q** é o conjunto de estados do simulador CONTROLAB-AGV. Esse conjunto pode ser definido pelos estados que o ambiente (recinto que o AGV está inserido) apresenta.

No exemplo do CONTROLAB AGV pode-se considerar o conjunto de valores de saída como sendo o conjunto dos estados do sistema. O domínio pode ser definido pelo conjunto {"estados do ambiente"} e o conjunto de valores reais **R**, já que engloba o domínio do tempo decorrido e dos valores arbitrários de entrada (Q={"estados do ambiente"} X R X R).

O campo potencial apresentado em um determinado momento, no ambiente de simulação, também pode ser considerado como um estado do ambiente em que o AGV está inserido.

Através do estado **Q** calculam-se os coeficientes da curva polinomial na definição da trajetória, Logo, de acordo com o fluxo do campo potencial apresentado em determinado momento no ambiente, calculam-se algumas condições iniciais como:  $|\mathbf{p'(0)}| \in |\mathbf{p'(1)}|$  (módulo do vetor velocidade nos pontos inicial e final da trajetória),  $\mathbf{K}_A \in \mathbf{K}_B$  (curvatura nos pontos inicial e final da trajetória) e  $\mathbf{e}_1$  (**0**) e  $\mathbf{e}_1$  (**1**) (representam os vetores tangentes no ponto inicial e final).

 $\delta$ : Q x X  $\blacktriangleright$  Q é a função de transição de estados. Essa função pode ser traduzida por  $\delta(q(\mathbf{u}), \mathbf{x}(\mathbf{u}))$ .

 $\lambda$ : Q x X  $\blacktriangleright$ Y é a função de saída. No caso do simulador CONTROLAB AGV, o sistema está em função do tempo de amostragem  $\mathbf{t}_A$ , do estado Q (estado do ambiente) e dos dados de entrada X. A função de saída é dada por  $\lambda$  (q(u),x(u)).

c é uma constante empregada para especificar o tempo base. No simulador CONTROLAB AGV essa constante pode ser traduzida pelo **tempo da** *spline*. O passo da *spline* é o tamanho do passo, gerado pelo campo potencial, que o veículo autônomo vai tomar na sua locomoção definindo o ponto final da trajetória *spline*.

O passo da *spline* é um parâmetro do simulador. A cada passo do AGV é feito o cálculo da curva polinomial para definir a próxima posição do AGV dentro da trajetória. Logo, o **passo da** *spline* =  $n^{\circ}$  de unidades do passo do AGV \* (passo do AGV). O número de unidades do passo do AGV também é um parâmetro do simulador.

O tempo de amostragem normalizado **u da trajetória**, é calculado em função do **tempo da spline** (c). Como será visto no capítulo 4, sendo  $\mathbf{t}_A$  tempo de amostragem real do sistema, o tempo de amostragem normalizado **u** será:  $\mathbf{u} = \mathbf{t}_A / \mathbf{c}$ .

O gráfico da figura 2.7 mostra como seria o comportamento do modelo do simulador CONTROLAB-AGV durante a definição de uma trajetória. O gráfico mostra a relação dos conjuntos **X** (conjunto de valores de entrada), conjunto **S** (conjunto dos estados do ambiente em que o AGV está inserido) e o conjunto **Y**(conjunto dos valores de saída) em relação ao tempo de amostragem **u**. No resultado apresentado em Y<sub>4</sub> a trajetória *spline é* definida através dos pontos obtidos pelos os outros resultados, por onde o AGV passou.



Figura 2.7. Gráfico do modelo de DTSS clássico do sistema de simulação do CONTROLAB AGV.

Como dito anteriormente, a simulação trata-se de uma ferramenta de apoio à tomada de decisão. Com a simulação é possível reduzir riscos e custos envolvidos em um processo de mudança no sistema real, uma vez que os mesmos podem ser pré-determinados antes da efetiva mudança ocorrer.

Com o uso da simulação é possível resolver problemas de limitações de capacidade de um sistema real, sem colocá-lo em prática. Estes problemas podem ser resolvidos ainda na fase de concepção e projeto, pois variações aleatórias em processos e ciclos de trabalho podem ser incorporadas ao modelo, aproximando-o do funcionamento real do sistema. Soluções alternativas podem ser geradas, testadas e certificadas, garantindo o desempenho da instalação muito antes de sua implementação física.

O simulador para o projeto CONTROLAB AGV está sendo desenvolvido devido a grande necessidade de dinamizar o estudo do comportamento do AGV (veículo autônomo). Com o simulador é possível analisar pontos importantes da trajetória definida e monitorar e/ou alterar todos os parâmetros relacionados à cinemática do movimento do veículo. Conseqüentemente, é possível eliminar riscos e custos, minimizando os testes exaustivos nas experiências práticas com o sistema real do projeto CONTROLAB AGV.

O simulador CONTROLAB AGV é implementado no compilador GCC<sup>2</sup>. É usada também uma biblioteca para programação de jogos chamada Allegro para obter uma melhor resolução gráfica na interface do simulador.

# 3.1 Arquivos de parâmetros

O simulador CONTROLAB possui vários parâmetros aplicáveis no sistema. Esses parâmetros estão armazenados em dois arquivos de entrada: Labor\_tr.pb e Paramtr.dat. O arquivo Labor\_tr.pb contém parâmetros que define a planta baixa do ambiente por onde o AGV executará sua trajetória. Esses parâmetros são:

- · As coordenadas limites de toda área de atuação do AGV;
- · As coordenadas das paredes do ambiente por onde o AGV atuará;
- A coordenada de cada porta dos recintos que são definidos no ambiente;

<sup>&</sup>lt;sup>2</sup> Em Inglês, GNU Compiler Collection

- · As coordenadas da posição inicial do AGV;
- · As coordenadas do espaço livre para movimentação do AGV.

Como unidade de medida, esses parâmetros estão definidos em metros no arquivo Labor\_tr.pb, porém no simulador é pré-definida uma escala para esses valores. Esses valores são usados para construir a interface do ambiente por onde o AGV atuará. É possível criar diferentes ambientes por onde o AGV pode ser simulado. A interface gráfica do simulador CONTROLAB AGV utiliza a biblioteca Allegro, e permite a definição de opções de visualização como: cor de fundo, cor de desenho, escala de visualização e cor do texto.

O arquivo paramtr.dat define parâmetros funcionais do sistema. Esses parâmetros são utilizados na geração e adaptação da função de navegação global e dos *grids* de valores potenciais que representam o ambiente de simulação. Tais parâmetros são:

- Valores potenciais existentes nas paredes das salas (recintos) do ambiente de simulação;
- Valor potencial na posição alvo do AGV (goal position);
- Número de iterações para o cálculo do grid de valores potenciais que representam as salas (recintos) no ambiente;
- Valor da força de repulsão da carga pontual posicionada em cada ponto do obstáculo detectado;
- Tamanho do passo do grid das salas (recintos) no ambiente, em metros;
- Tamanho do passo do campo, em metros;
- Fator de atualização dos valores potenciais do grid das salas no ambiente, de acordo com o número de pontos de colisão identificados ao longo do trajeto;
- Número de unidades do passo do AGV, que multiplicado pelo passo do AGV formam o passo da *spline*.

O sistema do simulador CONTROLAB AGV é dividido em quinze módulos, onde cada módulo define um conjunto de funcionalidades específicas do sistema. O módulo principal é o geren\_tr.c, onde estão representadas as principais rotinas do simulador. Estas rotinas chamam rotinas de outros módulos dependendo da funcionalidade a ser utilizada. A maioria das rotinas é parametrizada de forma que qualquer eventual alteração dentro do sistema não precise modificar totalmente o sistema.

O sistema do simulador CONTROLAB AGV inicia lendo a descrição da planta baixa, a partir dos parâmetros atribuídos no arquivo Labor\_tr.pb. Em seguida o sistema inicia as variáveis de definição da trajetória local, a partir dos parâmetros contidos no arquivo Paramtr.dat.Em seguida, esses valores são armazenados em uma representação do ambiente de simulação, de acordo com os cálculos da função de navegação global.

# 3.2 Representação dos ambientes de simulação (geração e atualização dos *grids* do ambiente)

Existem várias formas de representar um ambiente dentro de um sistema. A forma usada no simulador CONTROLAB AGV foi a representação de mapas baseado em *grids*. A representação de ambientes em *grids* é mais fácil para construir e de se manter. Além disso, essa representação é discreta e está pronta para ser usada em conjunção com métodos de função harmônica (função de navegação global) para um plano de trajetória.

Antes do AGV (veículo autônomo) iniciar a sua trajetória local (a trajetória que o veículo percorrerá no ambiente para atingir a posição final - *goal position*), o sistema gera *grids* retangulares que representam um modelo físico do recinto (sala) configurado.

O conjunto de pontos definidos nos *grids* representa um conjunto de determinados pontos de uma sala (recinto) definida no ambiente. Cada ponto está afastado do outro por uma mesma distância. Essa distância é determinada pelo **passo do** *grid*, um parâmetro determinado pelo arquivo Paramtr.dat.

A Figura 3.1 mostra o *grid* original de uma sala (recinto), em um ambiente configurado por onde o AGV percorrerá sua trajetória. Na posição inferior direita, está representada no *grid*, a porta (*goal position*) que liga uma sala à outra no ambiente. Os pontos do *grid* estão separados por uma distância igual, definido pelo passo do *grid*.



Figura 3.1 - Grid representando os pontos do ambiente por onde o AGV gera sua trajetória

Para cada ponto de um *grid*, modelando um determinado ponto da sala (recinto) de um ambiente, associa-se um valor potencial calculado a partir de uma função harmônica (função de navegação global, mostrada no Capítulo 1). O conjunto desses pontos com valores associados geram o *grid* original da sala (recinto).

Os valores potenciais associados a cada ponto da parede são fixados em zero. Com o cálculo dos valores potenciais associados a cada ponto do *grid* é gerado um fluxo de campo potencial para o recinto especificado. De acordo com o fluxo desse campo o AGV irá se basear para definir sua trajetória.

A construção da função harmônica de navegação global foi mostrada no capítulo 1.2. Através da função potencial é determinado um campo vetorial. Este campo indica a direção do robô, a cada instante, a partir de sua localização. Essa **função de navegação<sup>3</sup>** é uma função harmônica  $\phi$ , tal que o vetor  $-\nabla \phi$  indicará o deslocamento do robô dentro do planejador local de trajetória.

Antes do cálculo dos valores potenciais associados a cada ponto da sala (recinto), gerando o *grid* original, o sistema aloca espaço de memória suficiente para garantir o cálculo de todos os pontos do *grid*.

Além do *grid* original da sala, também é alocado espaço de memória para um outro *grid* definido: *grid* atualizado da sala. O conjunto de pontos que formam o *grid* atualizado da sala é iniciado com os valores potenciais do *grid* original. O *grid* atualizado da sala será o novo *grid* original da sala, no momento em que o AGV detectar algum obstáculo através da "bengala" eletrônica, durante sua trajetória.

O grid atualizado é o grid original modificado, após a detecção de algum obstáculo. Esse grid é gerado por uma adaptação da função de navegação (**função de adaptação**). Esta função define uma atualização do grid original e com isso há uma mudança no fluxo do campo potencial na sala, que conseqüentemente gera um desvio na trajetória do AGV.

O ponto na sala, onde foi detectado um obstáculo, é definido como um **ponto detectado**. Muitas vezes, esse ponto, não representa exatamente um ponto definido do *grid* original da sala. Por isso o sistema define um ponto, chamado **ponto de colisão**, que é exatamente o ponto do *grid* original da sala mais próximo do ponto do obstáculo detectado pela bengala eletrônica (ponto detectado).

<sup>&</sup>lt;sup>3</sup> Na função de navegação as únicas informações relevantes são as dimensões do recinto e a localização da porta de saída. Como estas informações estão disponíveis antes da movimentação inicial do veículo, os valores de  $F_0(x,y)$  podem ser gerados e armazenados antecipadamente

Como foi visto no capítulo 1, no ponto de colisão é adicionado um valor potencial, calculado a partir de uma função harmônica de adaptação, para que com aumento do valor potencial nesse ponto, ocorra uma mudança do campo vetorial resultante e então gere um desvio da trajetória do AGV diante desse ponto.

Nos outros pontos do *grid* original da sala, próximos do ponto de colisão, o potencial associado é gerado a partir do fator de atualização da sala (parâmetro do arquivo Paramtr.dat) e da função harmônica de adaptação que define o grid atualizado pontual. Então, o novo *grid* atualizado da sala é gerado da seguinte forma:

```
gridAtualizadoSala [i,j] = gridOriginalSala [i,j] * fator sala +
gridAtualizadoPontual [i,j];
```

onde:

- gridAtualizadoSala[i,j] é o valor potencial associado ao ponto do grid atualizado da sala, localizado através dos índices i e j que representam coordenadas definindo uma posição na sala;
- gridOriginalSala[i,j] é o valor potencial associado ao ponto do *grid* original da sala, localizado através dos índices i e j que representam coordenadas definindo uma posição na sala;
- fator sala é o fator de atualização da sala, ou seja, um parâmetro no arquivo Paramtr.dat que é definido de acordo com o número de pontos de colisão identificados ao longo da trajetória do AGV;
- gridAtualizadoPontual [i,j] é o valor potencial associado ao ponto do *grid* atualizado pontual, localizado através dos índices i e j que representam coordenadas definindo uma posição na sala.

O grid atualizado pontual é definido a partir do grid original pontual. Os valores potenciais associados aos pontos do grid original pontual são dados por uma função potencial harmônica radial, mostrada no capítulo 1. Essa função é centralizada em qualquer posição definida no grid. No caso, a função harmônica radial será centralizada na posição do ponto de colisão. Como visto, esse ponto é determinado por uma eventual detecção de algum obstáculo durante a trajetória desenvolvida pelo AGV.

A Figura 3.2 mostra como o *grid* original da sala do ambiente é atualizado. O *grid* em verde representa o *grid* original da sala. No momento em que a "bengala" eletrônica (feixe em formato de cone em vermelho) detecta um obstáculo (retângulo em azul), um **ponto de detecção** é definido (em cinza). Logo, o **ponto de colisão** representado no *grid* em amarelo é gerado como o ponto do *grid* atualizado da sala mais próximo do **ponto detectado**.

Os valores potenciais dos pontos do *grid* atualizado pontual (na figura em azul) são adicionados aos valores potenciais associados aos pontos do *grid* original da sala (na figura em verde). A adição desses valores é feita a partir do centro de simetria da função potencial radial do *grid* atualizado pontual. Assim, o sistema faz o deslocamento do centro de simetria da função potencial radial, definida no *grid* atualizado pontual, para o ponto de colisão definido no *grid* original da sala.

Com isso, os pontos do *grid* original da sala, próximos do ponto de colisão, também atualizam os seus valores potenciais. A variação dos potenciais nos pontos próximos do ponto de colisão depende da função potencial radial, que define esses potenciais em função da distância desses pontos ao centro de simetria (ou seja, o ponto de colisão).



Figura 3.2 - *Grid* original da sala sendo atualizado, a partir do *grid* atualizado pontual após a detecção de um obstáculo

O grid atualizado pontual é gerado da seguinte forma:

onde:

- gridAtualizadoPontual [i,j] é o valor potencial associado ao ponto no grid atualizado pontual localizado através dos índices i e j, que representam coordenadas definindo uma posição na sala;
- gridOriginalPontual [i,j] é o valor potencial associado ao ponto no *grid* original pontual localizado através dos índices i e j, que representam coordenadas definindo uma posição na sala;

fator de carga é o fator de adição de carga, definido no sistema. Esse fator de carga possui um peso (em percentagem) dado pelo usuário. Esse peso é multiplicado pelo módulo do vetor velocidade, definido no ponto de colisão para definir o potencial adicionado.

Devido ao campo vetorial potencial, gerado pelos valores potenciais associados em cada ponto do *grid*, através da função de navegação, o sistema calcula um módulo para o vetor de campo gerado em cada ponto do *grid*. O módulo do vetor de campo representa o módulo do vetor velocidade naquele determinado ponto.

Através do tamanho do passo do AGV e do número de unidades do passo do AGV, parâmetros definidos no arquivo de parâmetros param\_tr.dat, calcula-se o passo da *spline* no simulador. Através desse passo que o AGV define sua trajetória spline. O passo do AGV definirá a posição que o AGV terá que tomar, durante sua trajetória. Assim:

passoSpline = passoAGV \* numUnidadesPassoAGV.

A cada passo do AGV é possível tomar importantes decisões de controle para gerar sua trajetória. Tais decisões são: atualização do *grid* devido à detecção de algum obstáculo, definição da velocidade (m/s), aceleração tangencial do AGV (m/s<sup>2</sup>) e a aceleração normal (m/s<sup>2</sup>) no ponto inícial e final da trajetória. Essas decisões são fundamentais para gerar o plano de trajetória a ser definida pelo AGV.

# 3.3 Interface Gráfica

Com os *grids* já calculados é feita a iniciação gráfica usando a biblioteca Allegro. A iniciação gráfica especifica as propriedades dos elementos da planta baixa, como por exemplo: cor de fundo, cor de frente, escala do gráfico, cores do modo texto. Todas as características gráficas são definidas em um "*bitmap*" criado em tempo de execução. Em seguida, os desenhos gráficos definidos no *bitmap*, são transferidos para a tela de interface do simulador (*screen*).

Como mostra a Figura 3.3, a tela do simulador CONTROLAB AGV é definida através da planta baixa (Labor\_tr.pb), que inclui parâmetros definidos no sistema como a posição dos obstáculos no ambiente, a posição da porta, o tamanho das paredes, o tamanho da área livre e a posição inicial e final da trajetória do AGV.



Figura 3.3 - Tela de simulação do AGV em um ambiente

Para representar o AGV no simulador é mostrado um círculo de raio de 25cm e a bengala eletrônica é representada por um conjunto de sete feixes (retas) de alcance de 85cm a partir do centro de massa do AGV.

As rodas (duas traseiras e uma dianteira) são representadas por traços no interior do círculo que representa o AGV. A distância entre as rodas traseiras (eixo) é de 33,4cm e a distância do eixo até o centro da roda da frente é de 28cm. O tamanho das rodas traseiras é de 14,5cm e da roda da frente é de 7,5cm. Todos esses valores são definidos de acordo com uma escala em relação aos valores reais tomados como exemplo de uma sala do laboratório CONTROLAB AGV

O arquivo de planta baixa Labor\_tr.pb pode definir um ambiente com mais de uma sala (recinto). Nesse tipo de área o AGV fará a sua trajetória passando pela(s) porta(s) (com posição também definida no arquivo Labor\_tr.pb) até atingir a posição final de sua trajetória, ou seja, o objetivo do AGV. Na Figura 3.3, mostrada acima, observa-se um ambiente de simulação com dois recintos distintos e uma porta por onde o AGV deve passar para atingir a posição final no ponto G.

Na interface do simulador CONTROLAB AGV, além da tela de simulação no ambiente, também são plotados parâmetros e valores de status que representam resultados durante a geração da trajetória do AGV. Esses resultados, além de mostrar um perfil da cinemática do AGV durante a simulação, mostra até que ponto o AGV atende suas restrições durante a definição de sua trajetória.

Como mostra a Figura 3.4, na tela são mostrados parâmetros de restrições da cinemática do AGV, como velocidade máxima em metros por segundo e aceleração tangencial máxima em m/s<sup>2</sup>, sendo que esses valores podem ser alterados pelo usuário.

Também são mostrados na tela do simulador, os parâmetros de simulação como o tamanho do passo do AGV em metros, o tamanho do passo do *grid* (definido no arquivo de parâmetros param\_tr.dat) e o tempo de amostragem em segundos para cada sub passo na geração da trajetória do AGV.

Para cada ponto de controle (ponto inicial e final de uma sub trajetória) a partir dos quais vai ser gerada a trajetória do AGV, foram plotados a sua posição no ambiente em metros, a velocidade em m/s, o ângulo de direção do AGV em radianos, a aceleração tangencial em m/s<sup>2</sup> e a aceleração normal em m/s<sup>2</sup>.

Em cada tempo de amostragem são mostrados os valores de status do AGV durante sua trajetória. Esses valores são a posição corrente em metros, a velocidade corrente em m/s, a aceleração tangencial corrente em m/s<sup>2</sup>, a aceleração normal corrente em m/s<sup>2</sup>. Todos esses valores são determinados para o eixo do AGV e para suas rodas direita e esquerda.

Além desses valores, são mostrados durante a trajetória do AGV, o raio de curvatura que o AGV define na trajetória em metros, a derivada do raio de curvatura em metros, o ângulo de direção da roda da frente do AGV em radianos e o ângulo de orientação do AGV na trajetória em radianos.

Os cálculos de todos esses valores são mostrados no capítulo 4, onde será mostrada a definição da trajetória do AGV no simulador CONTROLAB AGV.



Figura 3.4 - Interface de simulador CONTROLAB AGV

## 3.4 Isolinhas

No simulador CONTROLAB AGV é possível traçar linhas que ligam pontos do *grid* (representação do ambiente), que possuem valores potenciais iguais em um determinado momento no simulador. Essas linhas são chamadas de isolinhas e permitem um melhor estudo do comportamento do campo potencial gerado no grid. De acordo com o campo potencial definido no ambiente de simulação, a trajetória definida pelo veículo autônomo é perpendicular às isolinhas.

Para mostrar as isolinhas na tela do simulador CONTROLAB AGV são verificados todos os valores potenciais associados em cada ponto do ambiente especificado. Em seguida são verificados os pontos que possuem um valor potencial definido pelo usuário. Com os pontos identificados e armazenados, é feita a interpolação desses pontos através de curvas cúbicas polinomiais paramétricas, mais especificamente curvas *Hermites*.

Inicialmente, para gerar as isolinhas necessita-se descobrir os pontos com os mesmos valores potenciais solicitado pelo usuário, no *grid* de valores potenciais. O *grid* em questão é percorrido de quadrante em quadrante (conjunto de quatro pontos do *grid*), da menor coordenada x para a maior coordenada x e da menor coordenada y para a maior coordenada y.

Para cada quadrante do grid de valores potenciais, verifica-se em cada par de pontos do quadrante se o valor potencial solicitado pelo usuário está entre os valores potenciais desses dois pontos. Se esse valor estiver entre os valores dos dois pontos, então, um novo ponto será interpolado entre esses dois pontos. Logo, o novo ponto será um ponto onde a isolinha irá passar. Para todo ponto interpolado em um quadrante, deve existir um outro ponto no mesmo quadrante por onde a isolinha irá passar. Após estarem definidos todos os pontos com valor potencial igual ao solicitado pelo usuário, gera-se a curva que definirá a isolinha em cada quadrante do *grid* 

A figura 3.5, mostra um exemplo de *grid* com valores potenciais. Em cada quadrante do *grid* observam-se em vermelho, os pontos com o valor potencial de **0,04** (solicitado pelo usuário), por onde a curva que definirá a isolinha será traçada. Observa-se que para cada quadrante existem dois pontos por onde será traçada a curva que representa a isolinha.



Figure 3.5 - Verificação dos pontos no grid com valor potencial 0,04

De acordo com (FOLEY, 1990), a forma *Hermite* do segmento de curva cúbica polinomial é determinada pelos pontos  $P_1$  e  $P_4$  (ponto inicial e final da sub trajetória a ser definida) que serão interpolados e os vetores tangentes (vetor velocidade)  $R_1$  e  $R_4$  associados respectivamente nos pontos  $P_1$  e  $P_4$ .

Esses pontos podem ser expressos em forma de uma matriz denominada de matriz base hermite  $M_h$ . Para encontrar a matriz base hermite  $M_h$ , que relaciona o vetor geométrico hermite  $G_H$  aos coeficientes polinomiais, definem-se quatro equações, uma para cada ponto de limitação.

Os polinômios cúbicos que definem um segmento curvilíneo [Q(t) = x(t) y(t) z(t)] são determinados da seguinte forma:

x(t)	=	$a_{\mathrm{x}}$	t <sup>3</sup>	+ b	<sub>x</sub> t <sup>2</sup>	2 +	$\mathtt{C}_{\mathtt{x}}$	t +	d <sub>x</sub> ,		
y(t)	=	$a_{\rm y}$	t <sup>3</sup> +	b' p'	, t²	+	$\mathbf{C}_{\mathbf{y}}$	t +	d <sub>y</sub> ,		
z(t)	=	$a_z$	t <sup>3</sup>	+ b	z t <sup>2</sup>	2 +	$C_z$	t +	$d_{\mathrm{z}}$ ,	$0 \leq t \leq 1.$	(1)

O segmento de curva Q(t) é definido pelas limitações nos pontos inicial e final da curva, pelos vetores tangentes e a continuidade entre esses segmentos de curva. Cada polinômio cúbico mostrado acima tem quatro coeficientes, então quatro limitações serão necessárias, o que permite formular quatro equações com quatro incógnitas que serão resolvidas.

Para mostrar como os coeficientes são dependentes dos quatro pontos de limitação, mostrase que a curva cúbica paramétrica é definida por  $Q(t) = T \ge C$ .

A matriz de coeficientes é escrita como  $C = M \times G$ , aonde M é uma matriz de base 4x4 representando os coeficientes e G é um vetor geométrico de quatro elementos, representando as condições iniciais para a definição da curva..

Considerando o R<sup>2</sup>, Gx, Gy referem-se ao vetor dos componentes x, y respectivamente do vetor geométrico. M e G diferem para cada tipo de curva. Os elementos de M e G são constantes, então o produto T x M x G é justamente três polinômios cúbicos em t dado por.

$$Q(t) = [x(t) \ y(t)] = [t^{3} \ t^{2} \ t \ 1] \qquad \begin{pmatrix} m_{11} \ m_{12} \ m_{13} \ m_{14} \\ m_{21} \ m_{22} \ m_{23} \ m_{24} \\ m_{31} \ m_{32} \ m_{33} \ m_{34} \\ m_{41} \ m_{42} \ m_{43} \ m_{44} \end{pmatrix} \begin{pmatrix} G_{1} \\ G_{2} \\ G_{3} \\ G_{4} \end{pmatrix}$$
(2)

Reescrevendo x(t) a partir das equações (1) e (2), que definem os polinômios em x e y temos as equações (3):

x(0), x(1), y(0) e y(1) são encontradas pela substituição direta na equação definida acima:

Como no caso geral a equação (3) diferenciada fica x'(t) =  $[3t^2 2t 1 0] M_H * G_{Hx}$ .e y'(t) =

 $[3t^2 2t 1 0] M_H * G_{Hy}$ . A equação do vetor tangente pode ser escrita como:

As equações podem ser reescritas na forma de matriz da seguinte maneira:

$$\begin{pmatrix} P_1 \\ P_4 \\ R_1 \\ R_4 \end{pmatrix}_x = G_{Hx} = \begin{pmatrix} 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 \\ 3 & 2 & 1 & 0 \end{pmatrix} M_H * G_{Hx}$$

Para essa equação e a correspondente expressão para y seja satisfeita,  $M_{\rm H}$  deve ser o inverso da matriz 4 X 4.

$$M_{\rm H} = \begin{pmatrix} 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 \\ 3 & 2 & 1 & 0 \end{pmatrix}^{-1} \begin{pmatrix} 2 & -2 & 1 & 1 \\ -3 & 3 & -2 & -1 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix}$$

 $M_H$  pode agora ser usado em  $x(t) = T * M_H * G_H$ , para encontrar x(t) baseado no vetor geométrico  $G_{Hx}$ . Similarmente em  $y(t) = T * M_H * G_{Hy}$  Então:

$$Q(t) = [x(t) y(t)] = T * M_H * G_H$$

onde 
$$G_H$$
 é o vetor das condições iniciais  $\left( egin{array}{c} P_1 \\ P_4 \\ R_1 \\ R_4 \end{array} \right)$ 

Na Figura 3. é possível mostrar uma família de curvas cúbicas paramétricas Hermite no plano (x,y), com a direção do vetor tangente R1 no ponto P1 e o vetor tangente R4 no ponto P4. Somente o vetor tangente R1 varia para cada curva Hermite. O vetor tangente R4 é fixo para todas as curvas.



Figura 3.6 - Família de curvas cúbicas paramétricas Hermites

Para gerar as isolinhas, formadas pelas curvas cúbicas paramétricas *hermites*, o usuário define o número de isolinhas que deverão ser plotadas, o potencial da primeira isolinha a ser plotada e a variação de potencial de uma isolinha para outra, no caso de serem plotadas mais de uma isolinha.

A figura 3.6 mostra um conjunto de 10 isolinhas plotadas, sendo que a primeira isolinha define pontos com um potencial inicial de 0.02 e a variação de potencial de uma isolinha para outra é de 0,0025. Nesse exemplo, as isolinhas foram plotadas mostrando o campo original, campo este definido pela repulsão das paredes e pela atração da porta. Pode-se observar que a trajetória definida é perpendicular as isolinhas.



Figura 3.7 - 10 isolinhas plotadas mostrando o campo original do ambiente

A rotina para definir as isolinhas é descrita da seguinte forma:

```
type CoefficientArray = array[1..4] of real
```

begin

```
MoveAbs3 (cx[4], cy[4], cz[4]); {t=0: start x(0), y(0), z(0)}

\delta := 1/n; {Dividindo o tempo para o número de passos definidos}

t:=0;

for i := 1 to n do

begin

    t := t + \delta; t2 := t * t; t3 := t2 * t;

    x := cx[1] * t3 + cx[2] * t2 + cx[3] * t + cx[4];

    y := cy[1] * t3 + cy[2] * t2 + cy[3] * t + cy[4];
```

```
z := cz[1] * t3 + cz[2] * t2 + cxz3] * t + cz[4];
```

DrawAbs3 (x,y,z);

End

# 3.5 Planejador de trajetória global

A partir das posições inicial e final do AGV, o simulador utiliza o planejador de trajetória global para gerar um caminho ótimo que contenha a seqüência de portas a serem atravessadas pelo veículo durante sua trajetória. Esse caminho não considera possíveis obstáculos no ambiente, que somente serão detectados durante o trajeto do AGV dentro de cada sala (recinto).

Como mostrado no capítulo 1, para criar a trajetória global é gerado um grafo, aonde as portas por onde o AGV fará seu trajeto, representam os vértices desse grafo. A partir desse grafo é calculado o menor caminho pelo qual o AGV irá percorrer para atingir a posição alvo, o ponto G na figura 4.7 (*goal position*).

O menor caminho é definido através do peso que cada aresta do grafo possui. Esse peso é definido pela distância que o AGV terá que percorrer para chegar até a porta, ou seja, o peso é menor quanto menor for o caminho pelo qual será feito seu trajeto, até chegar a posição alvo (*goal position*). A Figura 3.8 mostra a simulação de um caminho global com as posições inicial (ponto S) e final (Ponto G), em um ambiente simples com dois recintos e apenas uma porta.



Figura 3.8 - Trajetória Global do AGV

End

## 3.6 Planejador de trajetória local

O trajeto em cada recinto é então simulado utilizando o subsistema de planejamento local. A função de navegação global, descrita anteriormente, determina para as coordenadas do ambiente um vetor indicando a nova direção do campo a ser seguida pelo AGV. Essa função é calculada a partir do campo potencial inicialmente gerado e armazenado em *grids* de valores potenciais representando cada sala, como visto anteriormente nesse capítulo. O passo do *grid*, que é a distancia entre os pontos vizinhos do *grid*, é também outro parâmetro de ajuste do simulador.

Logo, para cada ponto do *grid* original da sala, existe um valor potencial associado e através desses valores potenciais associados em cada ponto, define-se um campo vetorial potencial. Com o campo vetorial potencial é possível definir o módulo do vetor de campo no ponto, que na prática é o módulo do vetor velocidade nesse ponto. A direção desse vetor de campo (vetor velocidade) indicará o próximo passo do AGV na sala (recinto). Esta nova posição do próximo passo é calculada, através do parâmetro do tamanho do passo do AGV(passo da *spline*) pelo vetor unitário desta direção.

#### 3.6.1 Geração do próximo passo

Para calcular a direção do vetor de campo que definirá a direção do próximo passo, é feita a interpolação dos pontos do *grid* original da sala, que são adjacentes ao ponto onde o AGV se encontra. Isto é feito, pois o ponto onde o AGV está, muitas vezes não se encontra na exata posição de ponto do *grid* (representação do ambiente). Com isso, para gerar o potencial ou o vetor de campo nesse ponto, é necessário interpolar os valores associados aos pontos adjacentes do *grid*.

Como mostra a Figura 3.9, a interpolação dos pontos é feita da seguinte forma:

Para um ponto no *grid* (na figura 3.10 em vermelho) a ser interpolado existem quatro pontos adjacentes do *grid* (na figura 3.10 em verde), que possuem vetores de campo já definidos ( $v_1$ ,  $v_2$ ,  $v_3$ ,  $v_4$ ). Como foi visto, a distância entre os pontos adjacentes no *grid* é sempre igual e definida pelo passo do *grid* (d).

Com as distâncias do ponto a ser interpolado aos eixos horizontais e verticais dos pontos do *grid* ( $d_1$ ,  $d_2$ ,  $d_3$ ,  $d_4$ ), define-se o vetor de campo unitário nos eixos x e y do *grid*, na posição onde o ponto a ser interpolado se encontra ( $v_{1,2}$ ,  $v_{2,3}$ ,  $v_{3,4}$ ,  $v_{4,1}$ ).



Figura 3.9 - Interpolação para definir o vetor campo no ponto onde o AGV está e a direção do próximo passo do AGV

Logo esses vetores são definidos por:

 $\begin{array}{l} v_{1,2} = (d_2 \ / \ d_1 + \ d_2) & * \ v_1 + (d_1 \ / \ d_1 + \ d_2) & * \ v_2 \ (\texttt{eixo horizontal}) \\ v_{2,3} = (d_3 \ / \ d_3 + \ d_4) & * \ v_2 + (d_4 \ / \ d_3 + \ d_4) & * \ v_3 \ (\texttt{eixo vertical}) \\ v_{3,4} = (d_2 \ / \ d_1 + \ d_2) & * \ v_4 + (d_1 \ / \ d_1 + \ d_2) & * \ v_3 \ (\texttt{eixo horizontal}) \\ v_{4,1} = (d_3 \ / \ d_3 + \ d_4) & * \ v_1 + (d_4 \ / \ d_3 + \ d_4) & * \ v_4 \ (\texttt{eixo vertical}) \end{array}$ 

Interpolando com apenas um dos eixos (exemplo: eixo horizontal), é possível usar dois vetores para definir o vetor de campo resultante ( $v_r$ ) no ponto interpolado.

 $v_r = (d_4 / d_3 + d_4) * v_{3,4} + (d_3 / d_3 + d_4) * v_{1,2}$ 

assim,

 $\begin{array}{l} v_{r} = (d_{4} \ / \ d_{3} + \ d_{4}) & \ast (d_{2} \ / \ d_{1} + \ d_{2}) \ \ast \ v_{4} + (d_{1} \ / \ d_{1} + \ d_{2}) \ \ast \ v_{3} \ + \\ & (d_{3} \ / \ d_{3} + \ d_{4}) \ \ast \ (d_{2} \ / \ d_{1} + \ d_{2}) \ \ast \ v_{1} \ + \ (d_{1} \ / \ d_{1} + \ d_{2}) \ \ast \ v_{2}; \\ v_{r} = (d_{4} \ \ast \ d_{2} \ / \ (d_{3} + \ d_{4}) \ \ast \ (d_{1} + \ d_{2})) \ \ast \ v_{4} \ + \ (d_{1} \ \ast \ d_{4} \ / \ (d_{1} + \ d_{2}) \ \ast \ v_{2}; \\ & (d_{3} + \ d_{4})) \ \ast \ v_{3} \ + \ (d_{3} \ \ast \ d_{2} \ / \ (d_{3} + \ d_{4}) \ \ast \ (d_{1} \ + \ d_{2})) \ \ast \ v_{1} \ + \\ & (d_{3} \ \ast \ d_{1} \ / \ (d_{3} \ + \ d_{4}) \ \ast \ (d_{1} \ + \ d_{2})) \ \ast \ v_{1} \ + \\ & (d_{3} \ \ast \ d_{1} \ / \ (d_{3} \ + \ d_{4}) \ \ast \ (d_{1} \ + \ d_{2})) \ \ast \ v_{2}; \end{array}$ 

Observando a Figura 3.9 pode-se afirmar que  $(d_3 + d_4) * (d_1 + d_2) = d^2$ 

Então:

$$v_r = (d_4 * d_2 / (d^2)) * v_4 + (d_1 * d_4 / (d^2)) * v_3 + (d_3 * d_2 / (d^2)) * v_1 + (d_3 * d_1 / (d^2)) * v_2;$$

Observa-se também, que  $(d_4 * d_2) = A_{2,4}$ ,  $(d_1 * d_4) = A_{1,4}$ ,  $(d_3 * d_2) = A_{3,2}$ ,  $(d_3 * d_1) = A_{3,1}$ ;

Logo:

$$v_r = (A_{2,4}/(d^2)) * v_4 + (A_{1,4}/(d^2)) * v_3 + (A_{3,2}/(d^2)) * v_1 + (A_{3,1}/(d^2)) * v_2;$$

Como é possível definir as áreas  $A_{2,4}$ ,  $A_{1,4}$ ,  $A_{3,2}$ ,  $A_{3,1}$ , através das distâncias  $d_1$ ,  $d_2$ ,  $d_3$ ,  $d_4$  que são calculadas a partir do *grid*, então é gerado o valor unitário do vetor de campo V<sub>r</sub>.

Com o vetor de campo unitário resultante  $V_r$  definido, é possível definir a direção do ponto do próximo passo do AGV na trajetória a ser definida. Como mostra a Figura 10, com o passo do AGV e o vetor de campo unitário  $V_r$  resultante é possível definir o ponto do próximo passo do AGV.



Figura 3.10 - Gerando o próximo passo do AGV

Após gerar o próximo passo, o AGV gera uma sub trajetória até chegar na posição do próximo passo. Essa sub-trajetória é definida, respeitando a continuidade necessária para que o AGV faça o trajeto respeitando suas restrições de cinemática e gere uma trajetória mais suave, como será visto no próximo capítulo.

Na nova posição (próximo passo definido) será calculado outro vetor de campo, definindo a nova direção do AGV. Esse processo se repete até a porta de saída do recinto ou o ponto destino (*goal position*).

#### 3.6.2 Detecção de obstáculos

Se algum obstáculo for detectado pela bengala eletrônica, a função de navegação global é adaptada, atualizando-se os *grids* de potenciais, que representam a sala corrente. Conforme descrito anteriormente, um ponto do obstáculo é detectado e então é gerado um ponto de colisão (ponto do grid mais próximo ao ponto detectado). Uma partícula repulsiva é então posicionada neste ponto, gerando um novo *grid* atualizado(*grid* original pontual) de valores potenciais onde os potenciais serão adicionados ao *grid* atual da sala.

A bengala eletrônica possui sete feixes de sensores infravermelhos, onde cada feixe detecta um ponto de um possível obstáculo à frente da trajetória a ser definida pelo AGV. Para detectar um obstáculo no sistema verifica-se a interceptação do arco, que define a limitação de alcance dos sensores da bengala eletrônica, em alguma reta do polígono que define um obstáculo.

Se for detectado apenas um ponto de colisão, ou seja, apenas um feixe do sensor que simula a bengala eletrônica detectar o obstáculo, então o peso aplicado à carga a ser adicionada é definido por:

 $\lambda_1 = \lambda * | v_1 |$ 

onde

- $\lambda$  é o fator de carga;
- $|v_1|$  é o módulo do vetor de campo (ou vetor velocidade) no ponto de colisão.

No caso de ser detectado mais de um ponto de colisão, o fator de adição de carga é determinado pelo módulo do vetor velocidade nesse ponto de colisão específico e pelos módulos dos vetores velocidades dos outros pontos detectados da seguinte forma:

 $\lambda_1 = \lambda + \Sigma_{j=1}^{N} | v_j | / n$ 

onde:

- n é o número de pontos de colisão detectados;
- $\lambda$  é o fator de carga;
- ·  $|v_n|$  é o módulo do vetor de campo (ou vetor velocidade) nos n pontos de colisão gerados.

Devido a problemas de melhor adaptação da configuração física do ambiente (posição dos obstáculos) e para otimizar a trajetória definida pelo AGV, existem duas estratégias de detecção de obstáculo definidas no sistema. A primeira estratégia baseia-se em gerar pontos de colisão para todos os pontos detectados pela bengala eletrônica.

Como mostra a Figura 3.5, o AGV detecta três pontos e depois gera três pontos de colisão ocasionando o desvio do obstáculo devido a mudança do campo potencial no ambiente.



Figura 3.5 - AGV gerando pontos de colisão a partir de todos os pontos detectados

A segunda estratégia baseia-se em gerar dois pontos de colisão em um determinado extremo da bengala eletrônica, somente se o outro extremo da bengala não possuir pontos detectados. O extremo da bengala eletrônica significa os dois feixes extremos do sensor infravermelho, que simula a bengala (primeiro ou último feixe).

Como mostra a Figura 3.6, o AGV detecta três pontos e depois gera dois pontos de colisão em um extremo da bengala eletrônica, já que no outro extremo da bengala eletrônica não há pontos detectados. Isso ocasiona um desvio de obstáculo diferente ao desvio gerado na primeira estratégia de detecção.

Essa diferença de desvio acontece devido a um número menor de potenciais adicionados no ponto de colisão, gerando uma mudança do campo potencial diferente da mudança do campo ocorrida com a aplicação da primeira estratégia. Isto pode ser observado, comparando a Figura 3.5 com a Figura 3.6, onde a trajetória definida na primeira estratégia é diferente da trajetória definida na segunda estratégia.



Figura 3.6 - Geração pontos de colisão a partir de dois pontos detectados de um extremo

Como parâmetro do sistema de simulação, existe um peso (percentagem) que é associado à carga que será adicionada nos pontos detectados. Esse parâmetro serve para otimizar a trajetória que está sendo definida pelo AGV, já que existem pontos no ambiente que geram um campo potencial maior do que outros pontos.

Toda vez que um obstáculo é detectado, o usuário além de definir a estratégia a ser utilizada para geração do ponto de colisão, é preciso também definir um peso (em percentagem) que será aplicado à carga a ser adicionada. Quanto maior esse peso, maior será o desvio definido pelo AGV durante a sua trajetória.

Se um mesmo peso for aplicado no potencial a ser adicionado em diferentes pontos de colisão no ambiente em questão, irão existir pontos em que a carga adicionada não vai ser suficiente para gerar um desvio pelo veículo autônomo através da atualização do campo potencial. Por um outro lado, vão existir pontos no ambiente em que a carga a ser adicionada torna o desvio do veículo autônomo muito grande.

Como os pontos próximos da porta do ambiente possuem uma carga potencial maior que em pontos mais afastados da porta, precisa-se aplicar pesos diferentes no potencial a ser adicionado para pontos diferentes do ambiente dependendo da distância desses pontos para a porta do ambiente.

Na Figura 3.7, é possível observar que com um peso de 60% no valor potencial a ser adicionada no ponto de colisão, o desvio feito pelo AGV em um ponto afastado da porta, é mais satisfatório em relação ao desvio feito pelo AGV em um ponto mais próximo da porta, com o mesmo peso de 60% na carga.



Figura 3.7 - Desvio do AGV em pontos diferentes no ambiente com o mesmo peso aplicado na carga potencial a ser adicionada

O AGV detecta algum obstáculo sempre durante sua trajetória, mais especificamente cinco vezes o tempo de amostragem definido(5 \* 0.02 segundos) antes de chegar ao ponto final da sub trajetória. Como foi visto, a sub-trajetória é a trajetória definida pelo passo do AGV e pela direção do vetor de campo no ponto em que o AGV se encontra.

Na Figura 3.8, é possível verificar o momento em que a bengala eletrônica detecta um obstáculo durante sua trajetória. No exemplo, pode-se verificar a detecção de três pontos do obstáculo por três feixes da bengala eletrônica.



Figura 3.8 - Detecção do obstáculo

Na Figura 3.9, são mostrados em vermelho os pontos de colisão identificados a partir dos pontos detectados em verde, após serem definidos a estratégia e o peso a ser adicionado na carga potencial.

No exemplo da Figura 3.9, é possível verificar dois pontos de colisão para três pontos detectados, apesar de a estratégia utilizada no exemplo ser a de gerar pontos de colisão para todos os pontos detectados. Isso acontece, devido aos pontos detectados estarem bem próximos uns dos outros. Como o ponto de colisão é definido pelo ponto do *grid* mais próximo do ponto detectado, existem pontos detectados que estão bem próximos, que geram o mesmo ponto de colisão.



Figura 3.9 - Próxima posição da trajetória após detecção do obstáculo

Verifica-se que após serem definidos os pontos de colisão, o *grid* da sala é atualizado e o próximo passo do AGV para definir sua trajetória é diferente do passo que seria definido em sua trajetória original. Com a detecção do obstáculo, é gerado um desvio da trajetória do AGV definido pela mudança do campo potencial na sala, por onde o AGV define sua trajetória.

Como foi visto, a trajetória gerada pelo AGV é definida de acordo com campo vetorial potencial apresentado no recinto por onde o mesmo é simulado. Através dos vetores de campo, no ponto onde o AGV se encontra, e do passo do *grid*, são definidos sub trajetos até a posição onde o AGV deve chegar. Com as sub-trajetórias é definida a trajetória local do AGV no recinto especificado.

No entanto, para se gerar uma trajetória local mais suave, as sub-trajetórias definidas devem apresentar uma certa continuidade entre elas, para que as restrições cinemáticas do AGV não sejam extrapoladas e reflitam a continuidade da trajetória.

No próximo capítulo, é mostrado como são definidas as sub trajetórias do AGV e que fatores e condições do sistema devem ser considerados para gerar uma trajetória local mais suave e contínua, em todos os pontos do trajeto definido pelo veículo autônomo (CONTROLAB AGV).

A navegação é a atividade básica de um robô móvel, habilitando-o ao livre deslocamento pelo seu espaço de trabalho, alcançando metas pré-definidas, realizando tarefas e desviando de obstáculos (LATOMBE, 1991). Isso é obtido através de sistemas de navegação presentes em todos os robôs deste tipo. Os robôs são diferentes nas formas de executarem os seus trabalhos, porém chegam objetivamente na geração ou rastreamento de uma trajetória, evitando o choque com obstáculos e preservando a integridade do robô.

Como já foi visto, foi proposto em (LOPES et al, 2001) um projeto que consiste no desenvolvimento do CONTROLAB-AGV, um veículo autônomo que opera sobre um ambiente controlado definido por uma planta baixa conhecida. Neste projeto, o veículo autônomo define sua trajetória através do campo potencial calculado no ambiente por uma função de navegação.

Os resultados foram satisfatórios, porém era necessário criar um simulador para dinamizar os testes realizados e aplicar um plano de definição de uma trajetória mais suave e contínua e que atenda as restrições mecânicas e cinemática do veículo.

Na Figura 4.1 é possível observar um dos experimentos realizado para projeto CONTROLAB-AGV.



Figura 4.1 - Experimento realizado no projeto CONTROLAB AGV

Em (AUDE et al, 2003), foi proposta a interpolação da trajetória definida por dois pontos consecutivos do grid do campo por uma função polinomial de quinta ordem. As condições de interpolação da função polinomial harmonizam a trajetória atendendo às restrições impostas pela cinemática do AGV, permitindo que ele desempenhe movimentos mais complexos.

## 4.1 Trajetórias contínuas

A necessidade da utilização de curvaturas contínuas, para definir uma trajetória, começou a ser desenvolvida por (NELSON, 1989), que imaginou dois tipos distintos de caminhos: *quintic* **Cartesiano** para caminhos que possuem um número grande de mudanças de manobras (direção) durante a navegação e *spline* **polar** para navegação com curvas simétricas. Com isso, é possível suavizar segmentos de trajetórias que serão interpolados por pontos, onde a curvatura é fixada em zero.

Usando um caminho *quintic spline*, ou seja, curvas simétricas de quinta or*dem*, foi possível definir um plano de trajetória, interpolando uma seqüência de pontos arbitrários e garantindo a continuidade geométrica de segunda ordem.

O modelo de movimento de um veículo pode ser definido através do seguinte sistema:

 $\begin{cases} x'(t) = v . \cos \theta(t) \\ y'(t) = v . \sin \theta(t) \\ \theta'(t) = (v / 1) . \tan d(t) \end{cases}$ 

Onde:

- x e y são as coordenadas cartesianas do eixo central do veículo, relacionadas ao ambiente onde o veículo está inserido;
- v é a velocidade do veículo no seu ponto simetria;
- $\theta$  é o ângulo definido pelo veículo em relação aos eixos x e y;
- L é a distância entre os eixos (traseiro e dianteiro) do veículo;
- d é o ângulo definido pela roda da frente do veículo, ou seja, é a variável de controle para guiar o veículo.

Na Figura 4.2 observa-se o modelo de movimento do veículo através do AGV.



Figura 4.2 - Modelo de movimento do AGV

## 4.2 Trajetória Quintic Spline

Como foi visto em (AUDE et al, 2003), a trajetória *quintic spline* permite interpolação de uma seqüência de pontos arbitrários, com uma continuidade geométrica de segunda ordem ( $G^2$ ) e obtendo uma curvatura contínua. Quando uma curva **p**(**u**) tem continuidade geométrica de segunda ordem, dizemos que **p**(**u**) é uma curva  $G^2$  se o vetor de curvatura é contínuo ao longo da curva.

De acordo com (PIAZZI, 2002), a curva no plano  $\{x,y\}$  pode ser descrita pela parametrização  $\mathbf{p}(\mathbf{u}) = [\mathbf{x}(\mathbf{u}) \ \mathbf{y}(\mathbf{u})]^T$  com o parâmetro real  $\mathbf{u} \in [\mathbf{u}_0, \mathbf{u}_1]$ . Considerando  $\mathbf{p}(\mathbf{u})$  como uma curva orientada que começa no ponto  $\mathbf{p}(\mathbf{u}_0) = [\mathbf{xa} \ \mathbf{ya}]^T$  e termina em  $\mathbf{p}(\mathbf{u}_1) = [\mathbf{xb} \ \mathbf{yb}]^T$ , dizemos que a curva  $\mathbf{p}(\mathbf{u})$  é regular se existe  $\mathbf{p}'(\mathbf{u})$ , 1<sup>a</sup> derivativa de  $\mathbf{p}(\mathbf{u})$ , também definido em  $[\mathbf{u}_0, \mathbf{u}_1] \in \mathbf{p}'(\mathbf{u}) \neq \mathbf{0} \ \forall \mathbf{u} \in [\mathbf{u}_0, \mathbf{u}_1]$ .

Para que essa trajetória seja criada é necessário resolver o problema da interpolação em  $G^2$ . Para isso, é preciso determinar a curva polinomial de ordem mínima que é interpolada entre os pontos inicial  $\mathbf{p}_A = \{\mathbf{x}_A, \mathbf{y}_A\}$  e final  $\mathbf{p}_B = \{\mathbf{x}_B, \mathbf{y}_B\}$  da sub trajetória ou sub-rota.

Em cada um desses pontos estão associados o vetor tangente inicial (associado ao ponto  $p_A$ ) e o vetor tangente final (associado ao ponto  $p_B$ ). Esses vetores são definidos pelos ângulos  $\theta_A$  e  $\theta_B$ e as curvaturas escalar **K**<sub>A</sub> e **K**<sub>B</sub>, como mostra a Figura 4.3.



Figura 4.3 - Problema de interpolação no plano {x,y}

Os dados de interpolação:  $p_A$ ,  $p_B$ ,  $\theta_A$ ,  $\theta_B \in [0,2\pi)$ ,  $k_A e k_B \in \Re$  podem ser determinados arbitrariamente. Para resolver o problema da interpolação, considera-se uma curva polinomial *quintic*  $p(u) = [x(u) \ y(u)]^T$ ,  $u \in [0,1]$  onde

 $\begin{aligned} \mathbf{x}(\mathbf{u}) &:= \mathbf{x}_0 + \mathbf{x}_1 \mathbf{u} + \mathbf{x}_2 \mathbf{u}^2 + \mathbf{x}_3 \mathbf{u}^3 + \mathbf{x}_4 \mathbf{u}^4 + \mathbf{x}_5 \mathbf{u}^5 \\ \mathbf{y}(\mathbf{u}) &:= \mathbf{y}_0 + \mathbf{y}_1 \mathbf{u} + \mathbf{y}_2 \mathbf{u}^2 + \mathbf{y}_3 \mathbf{u}^3 + \mathbf{y}_4 \mathbf{u}^4 + \mathbf{y}_5 \mathbf{u}^5 \end{aligned}$ 

As condições iniciais de interpolação são:

$$P(0) = p_{A}, p(1) = p_{B}$$

$$e_{1}(0) = \begin{bmatrix} \cos \theta_{A} \\ \sin \theta_{A} \end{bmatrix}, e_{1}(1) = \begin{bmatrix} \cos \theta_{B} \\ \sin \theta_{B} \end{bmatrix}$$

$$k(0) = k_{A}, k(1) = k_{B}$$

Onde:

- p(0) e p (1) representam a curva polinomial nos pontos inicial  $(p_A) e$  final  $(p_B)$  normalizados em um tempo que varia de 0 a 1;
- K<sub>A</sub> e K<sub>B</sub> representam a curvatura nos pontos inicial (p<sub>A</sub>) e final (p<sub>B</sub>);
- e1(0) e e1(1) representam os vetores tangentes no ponto inicial e final, sendo que e1(u) é dado por p'(u)/||p'(u)||, onde p'(u) representa a derivada da curva polinomial p(u), que é o módulo do vetor velocidade.

Satisfazendo essas condições, a curva polinomial *quintic* tem seus coeficientes expressos da seguinte forma:

 $xo = x_A$ 

 $x_1$  =  $\eta_1$  . cos  $\theta_A$ 

 $x_2$  =  $\frac{1}{2}$  ( $\eta_3$  . cos  $\theta_A$  -  $A_{NA}$  . sen  $\theta_A$ );  $x_3 = 10(x_B - x_A) - (6\eta_1 + 3/2.\eta_3) \cos \theta_A - (4\eta_2 - 1/2\eta_4) \cos \theta_B +$  $3/2(\,A_{NA}$  . sen  $\theta_{A})$  –  $\frac{1}{2}(\,A_{NB}$  . sen  $\theta_{B})\,;$  $x_4 = -15(x_B - x_A) - (8\eta_1 + 3/2.\eta_3) \cos \theta_A - (7\eta_2 - \eta_4) \cos \theta_B 3/2(A_{NA} \cdot sen \theta_A) + (A_{NB} \cdot sen \theta_B);$  $x_5 = 6(x_B - x_A) - (3\eta_1 + 1/2, \eta_3) \cos \theta_A - (3\eta_2 - 1/2\eta_4) \cos \theta_B +$  $1/2\,({\tt A}_{\rm NA}$  . sen  $\theta_{\rm A})$  –  $1/2\,({\tt A}_{\rm NB}$  . sen  $\theta_{\rm B});$  $yo = y_A$  $y_1 = \eta_1$ . sen  $\theta_A$  $y_2 = \frac{1}{2} (\eta_3 \cdot \sin \theta_A + A_{NA} \cdot \cos \theta_A);$  $y_3 = 10(y_B - y_A) - (6\eta_1 + 3/2.\eta_3) \operatorname{sen} \theta_A - (4\eta_2 - 1/2\eta_4) \operatorname{sen} \theta_B 3/2(\,A_{NA}$  . sen  $\theta_A)$  +  $\frac{1}{2}(\,A_{NB}$  . sen  $\theta_B\,)\,;$  $y_4 = -15(y_B - y_A) + (8\eta_1 + 3/2, \eta_3) \operatorname{sen} \theta_A + (7\eta_2 - \eta_4) \operatorname{sen} \theta_B +$  $3/2\,(\,A_{NA}$  . cos  $\theta_{A}\,)$  –  $(\,A_{NB}$  . cos  $\theta_{B}\,)\,;$  $y_5 = 6(y_B - y_A) - (3\eta_1 + 1/2, \eta_3) \operatorname{sen} \theta_A - (3\eta_2 - 1/2\eta_4) \operatorname{sen} \theta_B 1/2({\rm A_{NA}}$  . cos  $\theta_{\rm A})$  +  $1/2({\rm A_{NB}}$  . cos  $\theta_{\rm B});$ 

onde:

- x<sub>A</sub> e x<sub>B</sub> são as componentes em x da posição, definida pela curva polinomial, nos pontos inicial (p<sub>A</sub>) e final (p<sub>B</sub>) da sub trajetória respectivamente;
- y<sub>A</sub> e y<sub>B</sub> são as componentes em y da posição, definida pela curva polinomial, nos pontos inicial (p<sub>A</sub>) e final (p<sub>B</sub>) da sub trajetória respectivamente;
- sen θ<sub>A</sub> e cos θ<sub>A</sub> é o seno e co-seno do ângulo formado pelo vetor de campo no ponto inicial p<sub>A</sub>, definido pelo campo vetorial potencial da sala;
- sen  $\theta_B$  e cos  $\theta_B$  é o seno e co-seno do ângulo formado pelo vetor de campo no ponto final  $p_B$ , definido pelo campo vetorial potencial da sala;
- $\eta_1$  é o módulo do vetor velocidade (vetor de campo) no ponto inicial p<sub>A</sub> da sub trajetória;
- $\eta_2$  é o módulo do vetor velocidade (vetor de campo) no ponto final p<sub>B</sub> da sub trajetória;
- $\eta_3$  é o módulo do vetor aceleração tangencial no ponto final p<sub>A</sub>, definido a partir do campo vetorial potencial;
- $\eta_4$  é o módulo do vetor aceleração tangencial no ponto final  $p_B$ , definido a partir do campo vetorial potencial;
- A<sub>NA</sub> e A<sub>NB</sub> são os módulos dos vetores aceleração normal nos pontos inicial (p<sub>A</sub>) e final (p<sub>B</sub>) respectivamente, definido a partir do campo vetorial potencial;

Os parâmetros reais  $\eta_i$ , onde i = 1,...,4, que aparecem nos coeficientes, podem ser colocados juntos na forma de um vetor de quarta dimensão  $\eta_i := [\eta_1, \eta_2, \eta_3, \eta_4]T$ , resultando em uma curva paramétrica denotada como **p**(**u**;  $\eta$ ) ou, informalmente,  $\eta$ -spline.

# 4.3 Trajetória Quintic spline aplicada ao problema simulado

Na definição da trajetória *quintic spline*, os veículos autônomos são configurados como um ponto no R<sup>2</sup>. Para tratar o problema do movimento do CONTROLAB AGV, foi necessário o cálculo do tempo de percurso de forma a garantir que as velocidades e as acelerações nas rodas do veículo autônomo não ultrapassem os valores máximos permitidos.

Através dessa estimativa de tempo é possível calcular uma sub trajetória mais otimizada e suave. Assim, através dessa sub trajetória, o veículo autônomo se desloca sem violar as suas restrições de cinemática.

#### 4.3.1 Condições iniciais de definição da trajetória

No caso do problema simulado, o projeto CONTROLAB AGV, a melhor estimativa inicial para o tempo de trajeto é o tempo final do domínio teórico para a trajetória *spline* normalizada, ou seja, tempo  $\mathbf{t} = \mathbf{1seg}$ .

Com o tempo  $\mathbf{t} = \mathbf{1seg}$  e através da trajetória *spline* normalizada a ser definida, verifica-se o quanto o AGV violará suas restrições de cinemática, percorrendo essa trajetória *spline*. No sistema é definido um tempo de amostragem  $\mathbf{t}_a$ , do qual todos os tempos calculados no sistema devem ser múltiplos.

Para calcular a trajetória *quintic spline* é necessário definir suas condições iniciais. Tais condições são definidas pelos parâmetros  $\eta_1 \in \eta_2$ , que determinam o módulo da derivada da curva polinomial nos pontos inicial  $p_A$  ( $\eta_1 := ||\mathbf{p'}(\mathbf{0})||$ ) e final pB ( $\eta_2 := ||\mathbf{p'}(\mathbf{1})||$ ) da sub trajetória (subrota), com **u** (tempo)  $\in [\mathbf{0},\mathbf{1}]$ . Logo, no simulador CONTROLAB AGV, os parâmetros  $\eta_1 \in \eta_2$  representam o módulo do vetor velocidade nos pontos inicial  $\mathbf{p}_A$  e final  $\mathbf{p}_B$  da sub trajetória (subrota) do trajeto local do AGV.

Também são definidos como condições iniciais os parâmetros  $\eta_3 \in \eta_4$ . Esses parâmetros determinam, respectivamente, o módulo normalizado da derivada segunda da curva polinomial nos pontos inicial  $p_A (\eta_3 := 2x_2 \cos\theta_A + 2y_2 \sin\theta_A)$  e final pB ( $\eta_4 := [2x_2 + 6x_3 + 12x_4 + 20x_5]$ .  $\cos\theta_A + [2y_2 + 6y_3 + 12y_4 + 20y_5]$ .  $\sin\theta_A$ ) da sub trajetória (sub-rota), com u (tempo)  $\in [0,1]$ . No simulador CONTROLAB AGV, os parâmetros  $\eta_3 \in \eta_4$  representam, o módulo da aceleração tangencial nos pontos inicial  $p_A$  e final  $p_B$  da sub trajetória (sub-rota) do trajeto local do AGV.

As condições iniciais são determinadas pelo campo potencial gerado no ambiente de simulação e pelos valores de status do AGV (velocidades e acelerações do centro do eixo e das rodas traseiras e dianteira), durante sua trajetória.

Logo, nos pontos inicial **pA** e final **pB** da sub trajetória são definidos, como condições iniciais, os seguintes parâmetros: a velocidade em m/s, a aceleração tangencial em  $m/s^2$  e a aceleração normal em  $m/s^2$ .

De acordo com a definição da trajetória *spline*, em qualquer plano de curva  $\mathbf{p}(\mathbf{u}) = [\mathbf{x}(\mathbf{u}) \mathbf{y}(\mathbf{u})]^{\mathrm{T}}$ , a curvatura escalar é dada por:

k(u) = x'(u). y''(u) - x''(u).  $y'(u) / (x'(u)^{2} + y'(u)^{2})^{3/2}$ 

onde:

- x'(u) é a 1<sup>a</sup> derivada do componente em x (x(u)) da curva polinomial *quintic spline;*
- x''(u) é a 2<sup>a</sup> derivada do componente em x (x(u)) da curva polinomial *quintic spline*;
- y'(u) é a 1<sup>a</sup> derivada do componente em y (y(u)) da curva polinomial *quintic spline*;
- y''(u) é a 2<sup>a</sup> derivada do componente em y (y(u)) da curva polinomial *quintic spline*;
- u é o tempo decorrido, com  $u \in [0,1]$ .

Através da estimativa de tempo final ( $\mathbf{t} = \mathbf{1seg}$ ) para a trajetória *spline* normalizada, das condições iniciais do simulador (velocidades e as acelerações tangenciais e normais no ponto inicial e final do trajeto) e dos parâmetros do veículo autônomo defini-se a trajetória *quintic spline*, porém com resultados ruins que violam as restrições de cinemática do veículo autônomo.

Esses resultados certamente ultrapassam os seus limites de cinemática, pois o tempo t=1segé insuficiente para que o veículo autônomo percorra essa trajetória atendendo as suas restrições de cinemática. Para isso, necessita-se dar uma estimativa de tempo maior para a definição da trajetória. O novo tempo estimado de trajetória pode, então, ser calculado em relação às violações das restrições que ocorreu para trajetória normalizada, definida com uma estimativa de tempo (t = 1seg).

Como foi dito anteriormente, as curvas que definem a trajetória *quintic spline* devem apresentar uma continuidade entre elas. Por isso, a condição inicial, definida no ponto inicial  $\mathbf{p}_{\mathbf{A}}$  para a sub trajetória, deve ter o mesmo valor apresentado no último tempo de amostragem (**ta**) da sub trajetória *spline* anterior. Logo, a velocidade, a aceleração tangencial e a aceleração normal definidos no último tempo de amostragem **ta** da sub trajetória anterior serão as condições iniciais do ponto  $\mathbf{p}_{\mathbf{A}}$  dessa nova próxima sub trajetória.

Na primeira sub trajetória *spline* definida pelo AGV, necessita-se que o mesmo, no início da trajetória, ande em linha reta por um instante tempo. Isso deve acontecer sem haver uma quebra grande de inércia. Por isso no início da primeira sub trajetória definida, o AGV anda em linha reta em uma velocidade de cruzeiro. A velocidade de cruzeiro é definida como sendo um quinto da velocidade máxima de engenharia (**velocidade**  $_{cruzeiro}$  = **velocidade**  $_{máxima de engenharia}$  / **5**). O AGV andará em movimento uniformemente acelerado com o tempo definido por:

Tempo <sub>trajeto em linha reta</sub> = velocidade <sub>cruzeiro</sub> / aceleração tangencial <sub>máxima de</sub> engenharia

Onde:

- velocidade cruzeiro é a velocidade de cruzeiro definida durante a trajetória em linha reta;
- aceleração tangencial máxima engenharia é a aceleração tangencial máxima permitida pelo o AGV.

O status do AGV, no fim da sub trajetória anterior, mantém-se como condições iniciais no primeiro tempo de amostragem (**ta**) da próxima trajetória *spline* a ser definida. Com isso, o AGV inicia seu trajeto com os valores gerados na trajetória anterior, mantendo a continuidade.

Como será visto a seguir, as variáveis de status do AGV são calculadas durante a simulação de acordo com os cálculos da trajetória *spline* a ser definida.

A previsão para a velocidade no ponto final  $\mathbf{p}_{\mathbf{B}}$ , ou seja, a velocidade de chegada do AGV na sub trajetória é definida por uma velocidade menor que a velocidade máxima do AGV (0.211736 m/s<sup>2</sup>).

No caso do simulador observou-se que o melhor comportamento do AGV seria com uma previsão de velocidade no ponto final  $\mathbf{p}_{\mathbf{B}}$  sendo a velocidade de cruzeiro ( $|\mathbf{V}_{\mathbf{B}}|_{\text{ponto final } \mathbf{p}\mathbf{B}}$  = **vel.cruzeiro**), ou seja, a velocidade de chegada é a mesma que a velocidade de saída. Logo a condição inicial para a velocidade no ponto final  $\mathbf{p}_{\mathbf{B}}$  é determinada pela velocidade de cruzeiro ( $\eta_2$  =  $|\mathbf{V}_{\mathbf{B}}|_{\text{ponto final } \mathbf{p}\mathbf{B}}$ ).

As condições iniciais de aceleração tangencial e aceleração normal no ponto final  $\mathbf{p}_{B}$  são determinadas em relação ao campo gerado no ambiente. O cálculo desses valores é feito através dos vetores de campo (vetores velocidade)  $\mathbf{v}_{B} \in \mathbf{v}_{B'}$ , definidos no ponto final ( $\mathbf{p}_{B}$ ) e em um ponto posterior ( $\mathbf{p}_{B'}$ ) bem próximo do ponto final  $\mathbf{p}_{B}$ . Com a diferença vetorial entre esses vetores, é gerado um vetor aceleração. Com a projeção do vetor aceleração  $\mathbf{a}$ , nos eixos x do vetor de direção do campo (ou vetor velocidade)  $\mathbf{v}_{B} \in \mathbf{y}$  de direção ortogonal a  $\mathbf{v}_{B}$ , o vetor aceleração tangencial e o vetor aceleração normal no ponto são representados.

A Figura 4.4 mostra o vetor aceleração **a** sendo gerado a partir da diferença vetorial dos vetores **vB e vB'** durante um período de tempo, de dois pontos próximos **pB e pB'**. Transladando esse vetor aceleração **a** no ponto **pB**, defini-se o vetor aceleração tangencial  $\mathbf{a}_{T}$  e o vetor aceleração normal  $\mathbf{a}_{N}$ , através do produto vetorial e escalar desses vetores.



Figura 4.4 - Geração do vetor aceleração a partir do vetor velocidade

Com o objetivo de seguir a curvatura gerada pelo campo durante a definição da trajetória, o cálculo do módulo do vetor aceleração tangencial no ponto final  $\mathbf{p}_{\mathbf{B}}$  da sub trajetória é feito utilizando o mesmo fator que determina a relação entre as velocidades da trajetória e a do campo, já que se deseja que a curvatura da trajetória siga a do campo.

Logo, as acelerações são definidas como:

```
A_N = vel.cruzeiro<sup>2</sup> / r.curvatura
A_N _{CAMPO} = vel.campo<sup>2</sup> / r.curvatura
```

Então, o fator de continuidade do raio de curvatura é definido por: fator continuidade =  $A_N / A_N _{CAMPO}$  = vel.cruzeiro<sup>2</sup> / vel.campo<sup>2</sup>

Logo, o valor da aceleração tangencial, dado como condição inicial no ponto final  $\mathbf{p}_{B}$ , é o módulo do vetor aceleração tangencial |  $\mathbf{a}_{T}$  | <sub>campo</sub>, calculado a partir do campo gerado, multiplicado pelo fator de continuidade do raio de curvatura.

```
|a_{T}|_{ponto final pB} = (vel.cruzeiro<sup>2</sup> / vel.campo<sup>2</sup>). | <math>a_{T} |_{campo}
```

Esse valor de aceleração tangencial no ponto final  $\mathbf{p}_{B}$  é definido como condição inicial no ponto  $p_{B}$ . Logo,

 $\eta 4 = |\mathbf{a}_{\mathrm{T}}|_{\mathrm{ponto final pB}}$ 

O mesmo acontece para o módulo do vetor aceleração normal como condição inicial no ponto final  $\mathbf{p}_{B}$ , Esse valor é dado pelo módulo do vetor aceleração normal no campo $|\mathbf{a}_{N}|_{campo}$  multiplicado pelo fator de continuidade do raio de curvatura.

```
|a_N|_{ponto final pB} = (vel.cruzeiro^2 / vel.campo^2). | a_N |_{campo}
```
#### 4.3.2 Cálculo do novo tempo para gerar a nova trajetória quintic spline

Como foi visto anteriormente, após a geração de uma trajetória *spline* normalizada, definida por um tempo estimado de **t=1seg**, percebe-se que as restrições de cinemáticas do AGV são violadas. No entanto, o objetivo da definição dessa trajetória normalizada *spline* (com tempo estimado **t=1seg**) é mostrar o quanto as restrições de cinemática do AGV são violadas na trajetória.

Para o cálculo da trajetória *spline* normalizada de domínio teórico ( $t \in [0,1]$ ), as condições iniciais para o problema simulado são definidos da seguinte forma.

```
\begin{split} \eta 1 &= \text{vel.}_{\text{traj. anterior}} \\ \eta 2 &= \text{acel.}_{\text{TANG. no ta anterior}} \\ \eta 3 &= \text{vel.}_{\text{Cruzeiro}} \\ \eta 4 &= \text{acel.}_{\text{TANG. CAMPO}} * (\text{Vel.}_{\text{Cruzeiro}}^2 / |\text{Vel.}_{\text{Campo}}|^2) \\ \text{Acel}_{\text{NORM pl}} &= \text{acel.}_{\text{NORM no ta anterior}} \\ \text{Acel}_{\text{NORM p2}} &= \text{acel.}_{\text{NORM CAMPO}} * (\text{Vel.}_{\text{Cruzeiro}}^2 / |\text{Vel.}_{\text{Campo}}|^2) \end{split}
```

Onde, pela restrição de continuidade do movimento temos:

- $\eta_1$  (módulo do vetor velocidade no ponto inicial  $p_A$  da trajetória) = **vel.** Traj.Anterior (módulo da velocidade no ponto final  $p_B$  da trajetória anterior);
- η<sub>2</sub> (módulo do vetor aceleração tangencial no ponto final p<sub>A</sub> da trajetória) = acel. <sub>TANG. no</sub> ta anterior (módulo da aceleração tangencial no ponto final p<sub>B</sub> da trajetória anterior, para manter a continuidade);
- $\eta_3$  (módulo do vetor velocidade no ponto final  $p_B$  da trajetória) = **vel.** <sub>Cruzeiro</sub> é o módulo da velocidade de cruzeiro;
- $\eta_4$  (módulo do vetor aceleração tangencial no ponto final p<sub>B</sub>, definido a partir do campo vetorial potencial) = **acel.** <sub>TANG. CAMPO</sub> (módulo da aceleração tangencial no campo) \*  $(Vel._{Cruzeiro}^2 / |Vel._{Campo}|^2)$
- Acel <sub>NORM p1</sub> (módulo da aceleração normal no ponto inicial da trajetória p1) = acel. <sub>NORM</sub> no ta anterior é o módulo da aceleração normal no ponto final p<sub>B</sub> da trajetória anterior, para manter a continuidade;
- Acel <sub>NORM p2</sub> (módulo da aceleração normal no ponto final da trajetória p2) = acel. <sub>NORM</sub> <sub>CAMPO</sub> é o módulo da aceleração normal no campo definido no ambiente \* (Vel.<sub>Cruzeiro</sub><sup>2</sup>/ |Vel.<sub>Campo</sub>|<sup>2</sup>);
- tempo spline é o novo tempo da trajetória spline calculado.

Baseado nos resultados apresentados nessa trajetória *spline* normalizada calcula-se um novo tempo para a definição da trajetória. Isso é definido com o objetivo principal de percorrer a trajetória com uma estimativa de tempo maior, não violando as restrições de engenharia do AGV (velocidade máxima e aceleração tangencial máxima).

O novo tempo estimado da trajetória é calculado a partir dos seguintes fatores:

- tempo final definido no domínio teórico da trajetória quintic spline normalizada (t=1seg);
- relação entre a maior velocidade (em módulo) nas rodas traseiras do veículo autônomo AGV e a velocidade máxima de engenharia (em módulo) do veículo autônomo, obtida na trajetória *spline* com tempo de domínio teórico (t=1seg),;
- relação entre a maior aceleração tangencial (em módulo) nas rodas traseiras do AGV, obtida na trajetória *spline*;

Assim, obtém-se uma relação entre o tempo (**t=1seg**) de definição teórica da trajetória *quintic spline* e o tempo em uma trajetória *spline* que possivelmente não violará as restrições do veículo autônomo. Através dessa relação, as violações apresentadas na primeira *spline* são compensadas na segunda trajetória *spline*.

Para esse cálculo defini-se a maior velocidade apresentada durante a primeira trajetória *spline*. Como os parâmetros de entrada são relativos ao ponto no centro do eixo do AGV, deve-se calcular a maior velocidade (em módulo) para os pontos nas rodas traseiras do AGV, durante a trajetória *spline*, pois os limites de engenharia são aplicados às mesmas. O cálculo é feito da seguinte forma:

```
velocidade máxima nas rodas = |velocidade eixo . (r.curvatura +
  (eixo/2)/r.curvatura)| máximo
```

Onde:

- velocidade <sub>eixo</sub> é a velocidade no eixo do AGV, em um determinado ponto da trajetória spline (valor normalizado);
- r.curvatura é o raio de curvatura em um determinado ponto da trajetória *spline*, calculado através da definição dada (1/K(u)), (valor normalizado);
- eixo é o tamanho do eixo (distância entre as duas rodas traseiras) do AGV.

Para obter a maior aceleração tangencial (em módulo), aplica-se o mesmo procedimento feito para o cálculo da velocidade, ou seja, deve-se calcular o valor máximo da aceleração tangencial verificado nas rodas traseiras do AGV, durante a trajetória *spline* percorrida em 1 segundo. O cálculo é feito da seguinte forma:

```
ACELERAÇÃO TANGENCIAL MAIOR NA RODA ESQUERDA
aceleração tangencial máxima nas rodas = | aceleração tang. eixo . (1 - eixo
  / 2 . r.curvatura) + velocidade eixo . ((eixo . r.curvatura') / 2 .
  r.curvatura<sup>2</sup>)| maximo
ACELERAÇÃO TANGENCIAL MAIOR NA RODA DIREITA
aceleração tangencial máxima nas rodas = | aceleração tang. eixo . (1 + eixo
  / 2 . r.curvatura) - velocidade eixo . ((eixo . r.curvatura') / 2 .
  r.curvatura<sup>2</sup>)| maximo
```

Onde:

- aceleração tang.eixo é a aceleração tangencial no eixo do AGV, em um determinado ponto da trajetória *spline* (valor normalizado);
- velocidade <sub>eixo</sub> é a velocidade no eixo do AGV, em um determinado ponto da trajetória spline (valor normalizado);
- **r.curvatura** é o raio de curvatura em um determinado ponto da trajetória *spline*, calculado através da definição dada (1/K(u)), (valor normalizado);
- r.curvatura' é a derivada do raio de curvatura, em um dado tempo u, já calculado anteriormente;
- eixo é o tamanho do eixo (distância entre as duas rodas traseiras) do AGV.

Então, para calcular um novo tempo estimado de trajetória *spline*, em função do tempo de domínio teórico (tempo teórico=1), temos:

```
Se a relação das velocidades (velocidade máxima nas rodas / velocidade máxima
de engenheria) é MAIOR que a relação das acelerações (aceleração
tangencial máxima nas rodas / aceleração tangencial máxima de engenharia)
```

Então:

```
tempo spline = tempo teórico . (velocidade máxima nas rodas / velocidade máxima de
engenheria)
```

Senão:

```
tempo <sub>spline</sub> = tempo <sub>teórico</sub>. (aceleração tangencial <sub>máxima nas rodas</sub> /
aceleração tangencial <sub>máxima de engenharia</sub>)
```

Onde:

- tempo teórico é o tempo (t=1) de domínio teórico definido na trajetória quintic spline para o simulador;
- velocidade máxima nas rodas é calculado anteriormente, através da trajetória *spline* com o tempo de domínio teórico t = 1;
- velocidade máxima de engenharia é o parâmetro de entrada referente a velocidade máxima que o AGV pode atingir (limitação de engenharia). (velocidade máxima de engenheria = 0,33 m/s);
- aceleração tangencial máxima nas rodas, calculado anteriormente, através da trajetória spline com o tempo de domínio teórico;
- aceleração tangencial máxima de engenharia é o parâmetro de entrada referente a aceleração tangencial máxima que o AGV pode atingir (limitação de engenharia). (aceleração tangencial máxima de engenharia  $AGV = 0.88 \text{ m/s}^2$ ).
- tempo spline é o novo tempo da trajetória spline, calculado a partir da violação das restrições do veículo autônomo, durante a definição da trajetória spline normalizada;

Com isso, o novo tempo para gerar a nova trajetória *spline* (**tempo** <sub>spline</sub>) é aumentado em relação ao tempo de domínio teórico (**t=1seg**). O aumento do tempo estimado para definir uma trajetória determinará uma diminuição nos valores de status (velocidade e aceleração tangencial) do AGV, já que agora o AGV terá que fazer a trajetória *spline* em um tempo maior.

Como os cálculos no simulador são realizados com valores de status (velocidade e aceleração tangencial) normalizados ( $t \in [0,1]$ ), esses valores devem ser convertidos para valores reais para serem apresentados na tela do simulador.

Esses cálculos são feitos em relação ao novo tempo (**tempo** <sub>spline</sub>) a partir dos valores normalizados (velocidade e aceleração tangencial) da seguinte forma:

```
velocidade REAL = velocidade NORMALIZADA / tempo spline
aceleração tangencial REAL = aceleração tangencial NORMALIZADA / (tempo
spline * tempo spline)
aceleração normal REAL = aceleração normal NORMALIZADA / (tempo spline * tempo
spline)
```

Onde:

- velocidade REAL é a velocidade real em um determinado ponto da trajetória;
- aceleração tangencial <sub>REAL</sub> é a aceleração tangencial real em um determinado ponto da trajetória;
- aceleração normal <sub>REAL</sub> é a aceleração normal real em um determinado ponto da trajetória;
- velocidade NORMALIZADA é a velocidade normalizada calculada em um determinado ponto da trajetória;
- aceleração tangencial NORMALIZADA é a aceleração tangencial normalizada calculada em um determinado ponto da trajetória;
- aceleração normal <sub>NORMALIZADA</sub> é a aceleração normal normalizada calculada em um determinado ponto da trajetória;
- tempo spline é o novo tempo da trajetória spline, calculado a partir da violação das restrições do veículo autônomo, durante a definição da trajetória spline normalizada;

As condições iniciais da trajetória *spline* são definidas com o objetivo de manter a continuidade. Portanto, para definir a próxima trajetória *spline* necessita-se que suas condições iniciais sejam contínuas com relação a trajetória anterior.

A dilatação do tempo definida anteriormente, gera tempos diferentes para percorrer duas trajetórias reais adjacentes. Esta diferença acarreta descontinuidade entre as condições finais e iniciais de duas trajetórias consecutivas. Considerando uma trajetória *spline* real **Traj A** e uma outra trajetória subseqüente **Traj B**, os valores de status para o ponto final  $P_B$  da **Traj A** devem ser iguais as condições iniciais para o ponto inicial  $P_A$  da **Traj B**.

No entanto, os valores de status para o ponto final  $P_B$  da trajetória **Traj A** são calculados em função do novo tempo de *spline* para a trajetória **Traj A** (**Tempo** <sub>spline</sub> **T**raj **A**), já as condições iniciais para o ponto inicial  $P_A$  da Trajetória **Traj B** são calculados em função do novo tempo de *spline* para a trajetória **Traj B** (**Tempo** <sub>spline</sub> **Traj B**)

Para resolver esse problema é necessário definir uma outra trajetória spline normalizada, determinando novas condições iniciais para que os valores de status, quando transformados para a trajetória real gere continuidade em seus pontos extremos.

Logo, as novas condições iniciais para a definição da 2ª trajetória spline normalizada devem ser:

```
 \begin{aligned} \eta 1 &= \text{vel.}_{traj. anterior} * (\text{tempo}_{spline} / \text{tempo}_{spline anterior}) \\ \eta 2 &= \text{acel.}_{TANG. no ta anterior} * (\text{tempo}_{spline} / \text{tempo}_{spline anterior}) \\ \eta 3 &= \text{vel.}_{Cruzeiro} \\ \eta 4 &= \text{acel.}_{TANG. CAMPO} * (\text{Vel.}_{Cruzeiro}^2 / |\text{Vel.}_{Campo}|^2) \\ \text{Acel}_{NORM p1} &= \text{acel.}_{NORM no ta anterior} * (\text{tempo}_{spline} / \text{tempo}_{spline anterior}) \\ \text{Acel}_{NORM p2} &= \text{acel.}_{NORM CAMPO} * (\text{Vel.}_{Cruzeiro}^2 / |\text{Vel.}_{Campo}|^2) \end{aligned}
```

Onde:

- $\eta_1$  é o módulo do vetor velocidade (vetor de campo) no ponto inicial  $p_A$  da trajetória;
- $\eta_2$  é o módulo do vetor aceleração tangencial no ponto final  $p_A$  da trajetória;
- $\eta_3$  é o módulo do vetor velocidade (vetor de campo) no ponto final p<sub>B</sub> da trajetória;
- $\eta_4$  é o módulo do vetor aceleração tangencial no ponto final p<sub>B</sub>, definido a partir do campo vetorial potencial;
- Acel NORM p1 é o módulo da aceleração normal no ponto inicial da trajetória p1;
- Acel <sub>NORM p2</sub> é o módulo da aceleração normal no ponto final da trajetória p2;
- · vel. Cruzeiro é o módulo da velocidade de cruzeiro;
- vel. Traj.Anterior é o módulo da velocidade no ponto final pB da trajetória anterior;
- vel. CAMPO é o módulo da velocidade no campo definido no ambiente;
- acel. TANG. no ta anterior é o módulo da aceleração tangencial no ponto final p<sub>B</sub> da trajetória anterior, para manter a continuidade;
- acel. NORM no ta anterior é o módulo da aceleração normal no ponto final p<sub>B</sub> da trajetória anterior, para manter a continuidade;
- · acel. NORM CAMPO é o módulo da aceleração normal no campo definido no ambiente;
- tempo spline é o novo tempo calculado da trajetória spline.atual;
- tempo spline anterior é o novo tempo calculado da trajetória spline.anterior;

Através das novas condições iniciais e, conseqüentemente, da geração da nova trajetória *spline* real são feitos experimentos para observar se a trajetória definida pelo AGV apresenta a continuidade esperada e se os valores de status não violam as restrições do AGV.

#### 4.3.3 Valores de status do AGV

Todos os cálculos realizados para a trajetória *spline* são feitos com os valores normalizados, ou seja, valores relacionados ao domínio teórico ( $\mathbf{u} \in [0,1]$ ). Os valores reais são transformados em normalizados através do tempo inicial e final estipulado para o AGV percorrer a trajetória (**Parâmetro** normalizado = **Parâmetro** Real .(tempo Final – tempo inicial)). Como o tempo inicial nas sub trajetórias locais (subrota) é fixado em zero, o cálculo se baseia apenas no tempo Final (**Parâmetro** normalizado = **Parâmetro** Real . tempo Final).

Os valores de status do AGV são definidos de acordo com a curva polinomial que define a trajetória do AGV. Como todos os cálculos são normalizados, os valores de status do AGV são transformados para valores reais, através do tempo final que define a trajetória *spline*. O tempo final estipulado para definir uma trajetória *spline* é um tempo múltiplo do tempo de amostragem  $\mathbf{t}_{a}$ , portanto os valores de status são mostrados na tela do simulador a cada tempo de amostragem ( $\mathbf{t}_{a}$ ), durante a definição da trajetória.

Para cada tempo de amostragem **t**<sub>a</sub> da trajetória definida pelo AGV temos como valores de status do AGV: a velocidade no eixo do AGV, velocidade na roda esquerda e direita do AGV, a aceleração tangencial no eixo do AGV, na roda esquerda e direita do AGV, a aceleração normal no eixo do AGV, na roda esquerda e direita do AGV. Além do raio de curvatura, são também valores de status, a derivada do raio de curvatura e o ângulo definido pela roda da frente.

O raio de curvatura (r.curvatura) do AGV é definido como o inverso da curvatura  $\mathbf{k}(\mathbf{u})$ , para  $\mathbf{u} \in [0,1]$ . Com o cálculo de  $\mathbf{k}(\mathbf{u})$  definido pelo plano de definição da trajetória e já mostrado acima, o raio de curvatura no simulador CONTROLAB AGV é dado por:

r.curvatura = 1 / k(u)

onde

• **k**(**u**) é a curvatura em dado tempo u, dada por k(u) = x'(u). y''(u) – x''(u). y'(u) / (x'(u)<sup>2</sup> + y'(u)<sup>2</sup>)<sup>3/2</sup>.

Já a derivada do raio de curvatura (r.curvatura'), em um dado tempo u  $(\mathbf{u} \in [0,1])$ , é definida como r.curvatura' =  $(1 / k(\mathbf{u}))'$ .

De acordo com a trajetória *quintic spline*, em um dado tempo u ( $\mathbf{u} \in [0,1]$ ), a velocidade no eixo do AGV (vel <sub>eixo AGV</sub>) é definida pela 1<sup>a</sup> derivada do componente em  $\mathbf{x}(\mathbf{u})$  e  $\mathbf{y}(\mathbf{u})$  da curva polinomial *quintic spline* da seguinte forma:

vel  $_{eixo AGV} = V((x'(u) \cdot x'(u)) + (y'(u) \cdot y'(u))) / tempo _{final}$ 

onde:

•  $\mathbf{x}'(\mathbf{u})$  é a 1<sup>a</sup> derivada do componente em x (x(u)) da curva polinomial *quintic spline;* 

• y'(u) é a 1<sup>a</sup> derivada do componente em y (y(u)) da curva polinomial *quintic spline;* 

- **u** é o tempo de amostragem (ta) naquele instante, com  $u \in [0,1]$ .
- tempo <sub>final</sub> é o tempo final estimado para a definição da trajetória *spline*, usado para transformar o valor parametrizado em valor real.

Já as velocidades na roda direita (vel RD AGV) e esquerda (vel RE AGV) do AGV, são definidas da seguinte forma:

```
vel <sub>RE AGV</sub> = vel <sub>eixo AGV</sub> . (r.curvatura - (eixo / 2) / r.curvatura)
vel <sub>RD AGV</sub> = vel <sub>eixo AGV</sub> . (r.curvatura + (eixo / 2) / r.curvatura)
```

onde:

- vel eixo AGV é a velocidade no ponto do eixo do AGV, já calculada anteriormente;
- **r.curvatura** é o raio de curvatura no ponto onde o AGV, já calculado anteriormente;
- eixo é distância entre as rodas traseiras do AGV que definem o tamanho do eixo do AGV.

De acordo com a primitiva *quintic spline*, em um dado tempo u ( $\mathbf{u} \in [0,1]$ ), a aceleração tangencial no ponto do eixo do AGV (acel tang <sub>eixo AGV</sub>) é definida pela 1<sup>a</sup> e 2<sup>a</sup> derivada do componente em x(u) e y(u) da curva polinomial *quintic spline* da seguinte forma:

acel tang  $_{eixo AGV} = x'(u) \cdot x''(u) + y'(u) \cdot y''(u) / tempo _{final}^{2} \cdot vel$  $_{eixo AGV}$ 

onde:

- $\mathbf{x}'(\mathbf{u})$  é a 1<sup>a</sup> derivada do componente em x (x(u)) da curva polinomial *quintic spline*;
- y'(u) é a 1<sup>a</sup> derivada do componente em y (y(u)) da curva polinomial *quintic spline*;
- **x''(u)** é a 2<sup>a</sup> derivada do componente em x (x(u)) da curva polinomial *quintic spline;*
- y''(u) é a  $2^a$  derivada do componente em y (y(u)) da curva polinomial *quintic spline*;
- vel eixo AGV é a velocidade no ponto do eixo do AGV, já calculada anteriormente;
- tempo <sub>final</sub> é o tempo final estimado para a definição da trajetória *spline*, usado para transformar o valor parametrizado em valor real.

Já as acelerações tangenciais na roda direita (acel tang.  $_{RD AGV}$ ) e esquerda (acel tang.  $_{RE AGV}$ ) do AGV, são definidas da seguinte forma:

onde:

- acel tang <sub>eixo AGV</sub> é a aceleração tangencial no ponto do eixo do AGV, já calculada anteriormente;
- vel eixo AGV é a velocidade no ponto do eixo do AGV, já calculada anteriormente;
- r.curvatura é o raio de curvatura no ponto onde o AGV, em dado tempo u, já calculado anteriormente;
- r.curvatura' é a derivada do raio de curvatura, em um dado tempo u, já calculado anteriormente;

• eixo é distância entre as rodas traseiras do AGV que definem o tamanho do eixo do AGV.

Também, de acordo com a primitiva *quintic spline*, em um dado tempo u ( $\mathbf{u} \in [0,1]$ ), a aceleração normal no ponto do eixo do AGV (acel normal <sub>eixo AGV</sub>) é definida pela velocidade no ponto do eixo do AGV e pelo raio de curvatura definidos nesse tempo u, da seguinte forma:

acel normal eixo AGV = (vel eixo AGV)<sup>2</sup> / r.curvatura

O ângulo da roda da frente do AGV (ang <sub>RODA FRENTE</sub>) é definido pela distância do ponto do eixo do AGV com o centro da roda da frente do AGV (dist <sub>eixo RODA FRENTE</sub>) e pelo raio de curvatura (r.curvatura), através do cálculo do arco tangente, da seguinte forma:

ang RODA FRENTE = arctang (dist eixo RODA FRENTE / r.curvatura)

A derivada do ângulo da roda da frente (ang <sub>RODA FRENTE</sub>') define a velocidade angular gerada na roda da frente.

No próximo capítulo são mostrados os testes realizados para observar a definição da trajetória *spline* pelo AGV, o problema simulado. Durante os testes, observa-se que algumas restrições de cinemática específicas do AGV influenciaram nos resultados obtidos durante a definição da trajetória definida pelo AGV.

## 5.1 Introdução

Foram realizadas experiências com o problema simulado (projeto CONTROLAB AGV). Essas experiências têm por objetivo observar as características do simulador e se através delas, pode-se obter uma análise do comportamento do veículo autônomo durante a definição de sua trajetória. Assim, pode-se avaliar o sistema simulado e propor soluções para a melhoria de performance. As características do simulador CONTROLAB AGV são retratadas durante a simulação, não somente pela interface gráfica do AGV, como também através dos valores de status para o AGV, ou seja, valores que definem a cinemática do AGV.

Como já foi visto, os valores de status e a trajetória definida pelo o AGV são mostrados no simulador. Dentre os valores mostrados temos:

- posição do AGV (tanto da roda traseira esquerda, como da roda traseira direita) em metros;
- velocidade na roda direita e esquerda do veículo em m/s;
- aceleração tangencial e normal nas rodas direita e esquerda do veículo em m/s<sup>2</sup>;
- raio de curvatura e sua derivada, em cada ponto específico da trajetória do AGV, em metros;
- · ângulo de orientação da roda dianteira do AGV (roda de controle) em radianos.

Os valores de status do AGV são atualizados de ta (tempo de amostragem) em **ta**, ou seja, para cada novo ponto da sub-trajetória definida pelo AGV os valores mudam. No final da definição da sub trajetória é possível avaliar o comportamento que o AGV apresentará durante a definição da sua trajetória no contexto de testes reais, sem precisar executar testes exaustivos com o sistema real do AGV.

# 5.2 Descrição dos Experimentos

Os testes foram realizados no simulador, definindo os seguintes parâmetros de entrada para o simulador:

- limites de engenharia máximo de velocidade (em metros por segundo) e aceleração (m/s<sup>2</sup>) do AGV;
- passo do grid em metros, passo do campo em metros e o tempo de amostragem do AGV em segundos;

- posição, velocidade em m/s, direção em radianos para a posição inicial e final definida para cada sub trajetória (trajetória local) do trajeto global.;
- aceleração tangencial e normal em m/s<sup>2</sup> para a posição inicial e final definida para cada sub trajetória (trajetória local) do trajeto global.

Para o problema simulado, os valores dessas variáveis são definidos da seguinte forma:

- limites de engenharia máximo de velocidade: 0,33m/s;
- limites de engenharia máximo de aceleração: 0,88m/s;
- passo do grid: 0,1m e depois analisado com 0,2m e 0,4m;
- passo do campo: 0,4m;
- tempo de amostragem: 0,02m.

Esse contexto foi definido durante os testes no simulador. Foi preciso configurar os parâmetros da melhor forma possível para manter a regularidade do sistema e definir uma trajetória contínua e que ao mesmo tempo atenda às restrições mecânicas do AGV. Para melhor simular o ambiente real, um dos ambientes simulados foi definido em relação a um ambiente real existente que representa o laboratório CONTROLAB AGV. Essa representação foi feita respeitando uma escala.

## 5.3 Resultados

Foram realizados experimentos para duas condições iniciais diferentes, ou seja, duas trajetórias diferentes, com a observação do comportamento do campo potencial no ambiente simulado através da geração das isolinhas. Também foram simulados experimentos para dois tipos diferentes de ambientes, permitindo assim a análise e o teste de diferentes alternativas de funcionamento de um sistema.

### 5.3.1 Condições Iniciais do experimento

Para a construção dos testes a serem realizados é preciso definir as condições iniciais do CONTROLAB-AGV para a definição de sua trajetória contínua.

Como já foi visto, as condições iniciais da trajetória contínua do AGV são dadas através do valor do módulo de sua velocidade, do módulo da aceleração tangencial e do módulo da aceleração normal.

Para a 1<sup>a</sup> trajetória normalizada, defini-se as seguintes condições iniciais, nos pontos inicial  $P_A$  e final  $P_B$  da trajetória:

CONDIÇÕES INICIAIS NO PONTO  ${\tt P}_{\tt A}$  INICIAL DA TRAJETÓRIA

```
V<sub>0</sub> (velocidade inicial prevista) = Velocidade de Cruzeiro (Vel.
Cruzeiro = Vel. Max. Engenharia / 4);
AT<sub>0</sub> TANG.(aceleração tangencial inicial prevista)= 0;
AN<sub>0</sub> NORMAL.(aceleração normal inicial prevista)= 0;
CONDIÇÕES INICIAIS NO PONTO FINAL P<sub>B</sub> DA TRAJETÓRIA
V<sub>F</sub> (velocidade final prevista) = Velocidade de Cruzeiro (Vel. Cruzeiro
= Vel. Max. Engenharia / 4);
AT<sub>F</sub> TANG. (aceleração tangencial final prevista)= 0
AN<sub>F</sub> NORMAL. (aceleração normal final prevista)= 0
```

Onde:

- $V_0$  é o módulo da velocidade prevista no ponto inicial da sub-trajetória;
- · Velocidade de cruzeiro é a velocidade máxima de engenharia / 4;
- $A_{T0}$  é o módulo da aceleração tangencial prevista no ponto inicial da sub-trajetória;
- $A_{N0}$ é o módulo da aceleração normal prevista no ponto inicial da sub-trajetória;
- V<sub>F</sub> é o módulo da velocidade prevista no ponto final da sub-trajetória;
- $A_{F0}$  é o módulo da aceleração tangencial prevista no ponto final da sub-trajetória;
- $A_{N0}$ é o módulo da aceleração normal prevista no ponto final da sub-trajetória;

Como já foi visto, para o AGV gerar uma trajetória que atenda às suas restrições, define-se um tempo de trajetória, calculado através da definição da trajetória com previsão de tempo normalizado (**t=1seg**). Observando a figura 5.1, a trajetória *spline*, com previsão de tempo normalizado (**t=1seg**), viola as principais restrições de cinemática do AGV como velocidade e aceleração tangencial. Além de violar as restrições, essa trajetória apresenta-se não comportada.



Figura 5.1 – Trajetória normalizada com **tempo = 1seg**), violando as restrições de cinemática do veículo autônomo

Porém, o objetivo dessa trajetória normalizada é proporcionar um ponto de partida para o cálculo da melhor estimativa de tempo para definição da trajetória que será gerada pelo AGV. Como foi visto, esse cálculo é definido através da proporção que as restrições de cinemática do AGV são violadas.

Neste caso, como os valores de status do AGV, tais como aceleração tangencial e velocidade, violam os valores máximos permitidos para o AGV é preciso aumentar o tempo da trajetória na mesma proporção, para que a velocidade e/ou aceleração tangencial diminuam seus valores até atingir um valor igual ou próximo ao máximo permitido. Isso acontece porque a distância pela qual será definida a trajetória é a mesma para a trajetória normalizada com tempo t=1s.

Como foi visto, após o cálculo do novo tempo (**tempo** <sub>spline</sub>) de definição de trajetória, baseado nas violações das restrições do veículo autônomo e do tempo de definição da trajetória anterior (**tempo** <sub>spline anterior</sub>), define-se as novas condições iniciais para a trajetória *spline* respeitando a continuidade entre os pontos da trajetória.

Logo as novas condições iniciais para a geração da trajetória *spline* real são:

```
CONDIÇÕES INICIAIS NO PONTO INICIAL P_A DA TRAJETÓRIA

V_F (velocidade final prevista) = Velocidade de Cruzeiro * (tempo <sub>SPLINE</sub>

/ tempo <sub>SPLINE ANTERIOR</sub>);

AT<sub>F</sub> TANG. (aceleração tangencial final prevista)= 0
```

```
AN<sub>F</sub> NORMAL. (aceleração normal final prevista)= 0
CONDIÇÕES INICIAIS NO PONTO FINAL P<sub>B</sub> DA TRAJETÓRIA
V<sub>F</sub> (velocidade final prevista) = Velocidade de Cruzeiro;
AT<sub>F</sub> TANG. (aceleração tangencial final prevista)= 0
AN<sub>F</sub> NORMAL. (aceleração normal final prevista)= 0
```

Na figura 5.2 abaixo, observa-se que as restrições do veículo autônomo na foram violadas no mesmo ponto apresentado. Logo, pode-se avaliar o comportamento do AGV através do simulador CONTROLAB AGV com as condições iniciais determinadas acima, para a definição de sua trajetória.



Figura 5.2 – Trajetória Real com tempo t = tempo <sub>SPLINE</sub>

A trajetória definida pelo AGV deve apresentar características suaves e contínuas. Como já foi visto, para garantir a continuidade na trajetória definida pelo AGV, as condições iniciais no ponto inicial de uma sub-trajetória definida devem possuir valores de status, gerados pelo o AGV, no último tempo de amostragem da sub-trajetória anterior.

Na figura 5.3, pode-se observar a manutenção dos valores de status do AGV de um ponto final de uma sub trajetória para o ponto inicial de próxima sub trajetória, mostrando que esses pontos são exatamente os mesmos. Esses valores são aplicados como condições iniciais no ponto inicial da definição da próxima sub trajetória, garantindo a continuidade entre as trajetórias.



Figura 5.3 - Continuidade entre as trajetórias spline

Além do comportamento do veículo autônomo através dos valores de status, com o simulador é possível observar como ficou definida a trajetória do veículo autônomo. Na figura 5.4, mostrada abaixo, observa-se como a trajetória para essas condições iniciais ficou definida.



Figura 5.4 Trajetória spline definida no simulador

Como já foi visto, as novas condições iniciais para a aceleração tangencial e aceleração normalizada são calculadas em função da velocidade e as acelerações tangencial e normal definidas no campo potencial do ambiente de simulação.

Com o objetivo de seguir a curvatura gerada pelo campo durante a definição da trajetória e respeitando a continuidade, o cálculo do módulo do vetor aceleração tangencial no ponto final  $\mathbf{p}_{\mathbf{B}}$  da trajetória é feito também com relação a um fator que determina a continuidade do raio de curvatura.

Como os módulos das acelerações tangencial e normal são definidos pelo vetor aceleração tangencial e normal, gerado no campo potencial do ambiente e como a previsão do valor do módulo da velocidade é definida pela velocidade de cruzeiro. Para gerar um fator que defina a continuidade do raio de curvatura é necessário fazer uma relação entre as acelerações definidas pela velocidade no campo. Logo:

```
A_N = vel.cruzeiro<sup>2</sup> / r.curvatura
A_N <sup>CAMPO</sup> = vel.campo<sup>2</sup> / r.curvatura
```

Então, o fator de continuidade do raio de curvatura é definido por:

fator continuidade =  $A_N / A_N^{CAMPO}$  = vel.cruzeiro<sup>2</sup> / vel.campo<sup>2</sup>

Com isso, os valores dados como condições iniciais, no ponto inicial  $\mathbf{p}_{A}$  e no ponto final  $\mathbf{p}_{B}$ para a aceleração tangencial e a aceleração normal na 2<sup>a</sup> trajetória *spline* definida são calculados da seguinte forma:

NO PONTO INICIAL PA

```
|a<sub>T</sub>|ponto inicial pA = ((vel.cruzeiro<sup>2</sup> / vel.campo<sup>2</sup>). | a<sub>T</sub> |<sub>campo</sub>) * (tempo
spline<sup>2</sup> / tempo spline ANTERIOR<sup>2</sup>)
|a<sub>N</sub>|ponto inicial pA = ((vel.cruzeiro<sup>2</sup> / vel.campo<sup>2</sup>). | a<sub>N</sub> |<sub>campo *</sub>) * (tempo
spline<sup>2</sup> / tempo spline ANTERIOR<sup>2</sup>)
NO PONTO FINAL P<sub>B</sub>
|a<sub>T</sub>|ponto final pA = (vel.cruzeiro<sup>2</sup> / vel.campo<sup>2</sup>). | a<sub>T</sub> |<sub>campo</sub>
|a<sub>N</sub>|ponto final pB = (vel.cruzeiro<sup>2</sup> / vel.campo<sup>2</sup>). | a<sub>N</sub> |<sub>campo</sub>
```

Observa-se que para as acelerações tangencial e normal também deve ser considerada a continuidade da trajetória, por isso, seus valores também são baseados nos novos tempos calculados nesta trajetória e na trajetória anterior (**tempo** <sub>spline</sub> e **tempo** <sub>spline ANTERIOR</sub>).

Na figura 5.5 e 5.6, pode-se observar o comportamento do veículo autônomo e da definição da sua trajetória, através do simulador.



Figura 5.5 – Trajetória definida para condições iniciais baseadas em relação a velocidade e a aceleração no

campo



Figura 5.6 - Trajetória definida para condições iniciais baseadas em relação a velocidade e a aceleração no campo

Nas figuras 5.7 e 5.8 abaixo, é feita uma comparação dos dois experimentos realizados com as duas trajetórias *splines* diferentes (condições iniciais diferentes). Para a trajetória definida com as condições iniciais para as acelerações igual a zero ( $A_{TANG} = 0$  e  $A_{NORMAL}=0$ ), existem pontos que não violam as restrições de cinemática do veículo autônomo. Porém, os valores de status da aceleração tangencial, chegam mais próximo das restrições para a aceleração do que na trajetória definida com condições iniciais calculadas em relação ao campo potencial.



Figura 5.7 Trajetória spline com condição inicial de aceleração tangencial e normal = 0



Figura 5.8 Trajetória spline com condição inicial de aceleração tangencial e normal baseado no campo potencial

A definição do plano de trajetória contínua baseou-se em uma representação pontual para os veículos autônomos, ou seja, desconsiderando restrições importantes de um veículo autônomo durante a definição de sua trajetória.

Um dos fatores importantes observados no simulador CONTROLAB-AGV é a continuidade da derivada do raio de curvatura. Através da derivada do raio de curvatura é definida a velocidade angular da roda da frente do AGV, roda esta que determina os movimentos de transição da trajetória do AGV.

Através das isolinhas plotadas no simulador pode-se observar o comportamento do campo potencial do ambiente simulado. Nas figuras 5.9, 5.10 observa-se que a trajetória definida pelo veículo autônomo é perpendicular as isolinhas. Após duas detecções de obstáculos observa-se a alteração no campo potencial definido no ambiente simulado através das isolinhas.

Na figura 5.9 são plotadas 10 isolinhas com um potencial inicial de -0,02 com uma variação de uma isolinha para outra de 0,0025. É possível observar a alteração das isolinhas nos pontos de detecção do obstáculo, onde são adicionadas cargas potenciais.



Figura 5.9 - Comportamento do campo potencial do ambiente simulado através das isolinhas com a detecção do primeiro obstáculo.

Na figura 5.10 são plotadas 16 isolinhas com um potencial inicial de -0,05 e com uma variação de uma isolinha para outra de 0,0025. O potencial inicial é alterado devido ao campo potencial em pontos mais próximos da porta (goal) ser maior e por já haver uma atualização do campo devido à 1ª detecção do obstáculo.



Figura 5.10 - Comportamento do campo potencial do ambiente simulado através das isolinhas com a detecção do segundo obstáculo

### 5.3.2 Trajetórias em diferentes planos de ambiente

O simulador CONTROLAB AGV permite criar diversos cenários para a simulação do veículo autônomo. Com esse recurso, o simulador permite obter análises e avaliações de sistemas e propor soluções para a melhoria de performance.

Para observar o comportamento do AGV em diferentes ambientes de simulação, foram realizados testes de simulação do CONTROLAB AGV em dois tipos de planos de ambientes específicos.

Existe um ambiente que simula exatamente os experimentos realizados no sistema real. Esse experimento mostra um ambiente que representa uma sala do laboratório do CONTROLAB AGV de acordo com uma escala definida. Na figura 5.11 observa-se a simulação desse ambiente.



Figura 5.11 Simulação de um ambiente real

Na figura 5.12 observa-se o segundo ambiente. Esse ambiente é parecido com o anterior porém possui algumas diferenças. Essas diferenças são quanto à posição de saída do AGV, os tamanhos dos obstáculos existentes, além das granularidades dos grid's que representam cada ambiente. Isto acaba tornando este tipo de ambiente menos comportado que o ambiente anterior.



Figura 5.12 Simulação em um segundo ambiente definido para os experimentos

Nesse trabalho foi desenvolvido um simulador (CONTROLAB AGV) para veículos autônomos, que possibilita o estudo de trajetórias contínuas definidas por campos potenciais variantes no tempo. Uma função polinomial de quinta ordem é utilizada para interpolar dois pontos consecutivos de referido campo.

Observando os experimentos simulados conclui-se que com as facilidades apresentadas no simulador CONTROLAB AGV é possível gerar cenários, a partir dos quais pode-se: orientar o processo de tomada de decisão, obter análises e avaliações dos sistemas simulados e assim propor soluções para a melhoria de performance..

Com os resultados obtidos através das ferramentas implementadas (tais como: o cálculo dos valores de status, a definição da trajetória *spline* e a criação das isolinhas), foi possível otimizar os estudos obtendo os seguintes benefícios:

- a previsão de resultados na execução de uma determinada ação, como por exemplo na detecção de um obstáculo,
- a identificação de problemas antes mesmo de encontrá-los na execução do sistema real;
- a realização de análises e conclusões para as trajetórias definidas, como por exemplo a verificação se as restrições de cinemática do AGV são violadas. Podendo dinamizar os testes através das alterações dos parâmetros no próprio simulador,
- a redução de custos com o emprego de recursos que seriam usados no desgaste dos testes exaustivos do sistema real;
- a criação de vários cenários para simulação, ou seja, diferentes tipos e configurações do ambiente e dos obstáculos.

Com o simulador é possível também estudar o comportamento de veículos autônomos para diferentes granularidades do grid de potenciais do campo em cada ambiente.

Experiências com o simulador permitiram a definição dos valores dos módulos das velocidades e acelerações nas extremidades de cada *spline*, de forma que o AGV execute uma trajetória compatível com sua dinâmica.

Com o contexto e os parâmetros observados nos testes realizados e mostrados no capítulo anterior, viu-se que é possível definir uma trajetória contínua e suave para veículos autônomos.

Alguns fatores importantes na cinemática do veículo autônomo ainda não foram considerados no simulador. Fatores como um eventual atrito existente nas rodas traseiras do veículo, deve ser analisado, pois interferem diretamente na definição da trajetória.

Como trabalhos futuros, é possível aplicar novas opções de trajetórias curvilíneas contínuas, avaliando se o comportamento apresentado pelo AGV em diferentes trajetórias contínuas definidas e assim estabelecer aquela que apresentar um comportamento melhor.

Está sendo estudada também, uma heurística para ser aplicada na associação de pesos, para a carga repulsiva nos pontos de colisão detectados pela bengala eletrônica. Ao invés do usuário colocar o percentual de carga adicionada, o simulador em tempo de execução define esse percentual. Assim observando fatores importantes como: distância do obstáculo, distância da porta ou do ponto final da trajetória, o simulador avaliaria o percentual de carga a ser adicionado no ponto onde foi detectado o obstáculo.

Está sendo elaborado um aperfeiçoamento da interface do simulador. As análises e conclusões podem ser otimizadas através de recursos avançados de visualização (Ex::"zoom"), para melhorar a visualização do campo potencial dos ambientes de simulação e da visualização das isolinhas plotadas em determinado momentos no simulador.

- AUDE E.P.L.; LOPES, E.P., SILVEIRA, J.T.C.; SILVA, F.A.B.; MENEZES, L.F.S. et al. Real-Time Obstacle Avoidance performed by an Autonomous Vehicle throughout a Smooth Trajectory using an 'Electronic Stick'. In: International Conference on Intelligent Robots and Systems (IROS 2003), October 2003, Las Vegas, USA.
- AUDE, E.P.L. et al. Controlab MUFA: A Multi Level Fusion Architecture for Intelligent Navigation of a Telerobot. In: IEEE Conference On Robotics and Automation, 1999, Detroit. Procedures... Detroit: 1999a, p. 465-472.
- \_\_\_\_\_\_\_. Integration of Intelligent Systems and Sensor Fusion within the CONTROLAB AGV. In: SPIE'S International Symposium on Intelligent Systems and Advanced Manufacturing Mobile Robots XIV, 1999, Boston. **Procedures...** Boston: 1999b.
- CARNEIRO, G.H.M.B. et al. An Integrated Request Server Architecture for Telecommanding the CONTROLAB AGV through Real Time Data and Image. In: 42<sup>nd</sup> Midwest Symposium on Circuits and Systems, 1999, Las Cruces, New Mexico, USA **Procedures...** Las Cruces, New Mexico, USA: 1999.
- FOLEY, J. D., 1942; DAM, A. V.; FEINER, S. K.; HUGHES, J. F.; Computer Graphics: principles and practice, Addison-Wesley publishing company, USA, 1990, p. 483-488.
- PEDGEN, PEDGEN, C. D.; SHANON, R. E.; SADOWSKI, R. P. Introduction to simulation using SIMAN, McGraw-Hill Book Company 2 nd ed., New York,1995.
- EPSTEIN, B. Partial Differential Equations: An Introduction, McGraw-Hill Book Company, New York: 1962.
- HOLLOCKS, B. A well-kept secret? Simulation in manufacturing industry reviewed. *Conference* Announcement EUROSIM'92. Capri: 1992, p. 12-17.
- KROGH, B. H. A Generalized Potential Field Approach to Obstacle Avoidance Control. In International Robotics Research Conference, 1984, Pennsylvania. Procedures... Pennsylvania: 1984, p. 1150-1156.
- LATOMBE, J. C. Robot Motion Planning, Kluever Academic Publishers, Boston: 1991, p. 1-47.

- LOPES, E.P. et al. Application of a Blind Strategy for Obstacle Avoidance with the use of Potential Fields. In: IEEE Conference On Robotics and Automation, 2001, Seoul. Procedures... Seoul: 2001, p. 2911-2916.
- NELSON, W., Continuous-curvature paths for autonomous vehicles. In: IEEE Conference On Robotics and Automation, 1989, Arizona. Procedures... Arizona: 1989, v. 3, p. 1260–1264.
- OSÓRIO, F.; HEINEN, F.; FORTES L. Controle Inteligente de Veículos Autônomos: Automatização do Processo de Estacionamento de Carros, In: Mestrado em Computação Aplicada – PIPCA, UNISINOS – Universidade do Vale do Rio dos Sinos Centro de Ciências Exatas e Tecnológicas – C6, São Leopoldo, RS, 2001.
- PEGDEN, C. D.; SADOWSKI R. P.; SHANNON R. E.; Introduction to Simulation using SIMAN, 2<sup>nd</sup> Edition, Mc Graw-Hill, New York, NY, USA, 1995
- PIAZZI, A. et al. Quintic G2 Splines for the Iterative Steering of Vision-Based Autonomous Vehicles. In: IEEE Transactions on Intelligent Transportation Systems, 2002, Bologna. Procedures... Bologna: 2002, v. 3, n. 2, p. 27-36.
- SCHEUER, A.; FRAICHARD T. Planning Continuous-Curvature..., In: IEEE-RSJ Int. Conf. on Intell. Robots and Syst., Vol.3, Osaka, JP, Nov., 1996.
- SILVEIRA, J.T.C.; Aude E.P.L. et al. Trajectory Planning with Obstacle Avoidance within CONTROLAB, Anais do Workshop em Robótica Inteligente (WRI 1997), Brasília, Brazil, August, 1997.
- ZEIGLER, B. P.; PRAEHOFER H.; KIM T. G. Theory of Modeling and Simulation, Academic Press, Second Edition, San Diego, USA, 2000, p. 38-44 p. 144-145.

This document was created with Win2PDF available at <a href="http://www.win2pdf.com">http://www.win2pdf.com</a>. The unregistered version of Win2PDF is for evaluation or non-commercial use only. This page will not be added after purchasing Win2PDF.