

Algoritmos de Navegação Fuzzy

Uma Análise Qualitativa

Proposta para Trabalho de Conclusão do
Curso de Engenharia de Computação,
apresentado à UTFPR como requisito
parcial para obtenção do título de
Engenheiro de Computação.

Curitiba
2010

**Alexandre Jacques Marin
Júlio Cesar Nardelli Borges
Yuri Antin Wergrzn**

**Algoritmos de Navegação Fuzzy
Uma Análise Qualitativa**

Proposta para Trabalho de Conclusão do
Curso de Engenharia de Computação,
apresentado à UTFPR como requisito
parcial para obtenção do título de
Engenheiro de Computação.

Orientador:

Nome do orientador: João Alberto Fabro

E-mail do orientador: fabro@dainf.ct.utfpr.edu.br

Assinatura do orientador:

Curitiba

2010

Sumário

<u>Lista de Figuras</u>	4
<u>1 Introdução</u>	5
<u>1.1 Objetivo Geral</u>	6
<u>1.2 Objetivos Específicos</u>	6
<u>2 Metodologia</u>	7
<u>2.1 Fundamentos</u>	7
<u>2.2 Tecnologias</u>	7
<u>3 RECURSOS DE HARDWARE E SOFTWARE</u>	9
<u>3.1 Recursos de Hardware</u>	9
<u>3.2 Recursos de Software</u>	9
<u>3.3 Viabilidade</u>	10
<u>4 Contexto</u>	11
<u>5 Projeto a ser desenvolvido e resultados iniciais</u>	12
<u>5.1 Projeto de Software</u>	12
<u>5.2 Projeto de Hardware</u>	19
<u>6 Procedimentos de teste e validação</u>	22
<u>7 Análise de Riscos</u>	24
<u>8 Cronograma do Projeto</u>	29
<u>9 Conclusões</u>	30

10 Referências Bibliográficas 31

Lista de Figuras

<u>Figura 1 - Diagrama de casos de uso</u>	13
<u>Figura 2 - Diagrama em blocos do projeto</u>	19
<u>Figura 3 - Diagrama de Gantt</u>	29

1 Introdução

O problema do controle de navegação de robôs móveis autônomos é um campo da Engenharia da Computação que representa um grande desafio, devido ao fato de o ambiente ser dinâmico, haver sensoriamento sujeito a ruídos e exigências de controle e tomada de decisão em tempo real. Um sistema de navegação deve garantir que o robô móvel atinja satisfatoriamente o destino de sua trajetória, enviando ao robô os comandos necessários para a sua locomoção, de maneira precisa e suave, ao mesmo tempo em que permite reações rápidas às mudanças de ambiente para evitar colisões.

Na robótica móvel, existem dois principais paradigmas que guiam os projetos de diversas arquiteturas de sistemas de navegação: o reativo e o deliberativo. O paradigma reativo procura reproduzir a reação imediata dos animais aos estímulos do ambiente. Geralmente, arquiteturas reativas são empregadas como uma camada de nível inferior na navegação de robôs móveis, pois apresentam a vantagem de resposta em tempo real uma vez que mapeiam a leitura dos sensores, diretamente, em ações. Arquiteturas deliberativas, por outro lado, intercalam o processo da tomada de decisão, desde a percepção até a ação, com uma etapa de planejamento a qual demanda grande tempo computacional, impedindo a atuação do robô em tempo real. Atualmente, são definidas arquiteturas híbridas, conjugando ambos os paradigmas[1].

Ao realizar uma breve pesquisa para análise do estado da arte, foi possível perceber que existem vários trabalhos que apresentam novos métodos para navegação autônoma através do uso de lógica Fuzzy. Entretanto, não foi encontrado um trabalho propondo a comparação entre métodos já existentes. Com o intuito de preencher esta lacuna, a equipe optou desenvolver este projeto.

Um dos modelos que será avaliado neste trabalho é o modelo ED-FCM proposto por Mendonça no ano de 2010[2]. A sigla ED-FCM significa *Event-Driven Fuzzy Cognitive Maps* e quer dizer Mapas Cognitivos Difusos Dirigidos A Eventos.

Para realizar a comparação destes algoritmos, será necessária uma plataforma robótica, que será obtida pela reconstrução do robô Bellator [citar de que laboratório é esse robô]. O Bellator é um robô móvel que estava em construção em outro projeto mas foi abandonado. Este robô terá de ser reconstruído e adaptado às necessidades da equipe, levando em conta também a possibilidade de utilização do mesmo para projetos futuros.

As principais motivações deste trabalho são o desenvolvimento de uma plataforma robótica adequada para pesquisas acadêmicas, tendo em vista a falta de plataformas disponíveis com este propósito, e o estudo de

sistemas de controle de navegação robótica, que é um ramo de pesquisa de elevado valor acadêmico. A possibilidade de disponibilizar uma nova plataforma robótica para futuros estudos acadêmicos e estudar um algoritmo de navegação pouco explorado são os maiores incentivos da equipe para a realização deste trabalho.

1. Objetivo Geral

Construir uma plataforma robótica estável cujo hardware e software deverão ser documentados, testar dois algoritmos de navegação fuzzy nessa plataforma e compará-los segundo critérios a ser definidos. [Aqui se definem os critérios, exemplo, tamanho e complexidade do código, custo computacional, tempo de resposta, eficácia].

De maneira geral, a plataforma estável deve ser capaz de detectar obstáculos próximos a mesma através de sensores de distância. Aquela também deverá se deslocar por meio de um par de motores de corrente contínua acoplados a rodas revestidas de borracha e em cujos respectivos eixos estarão acopladas uma chave óptica e um encoder regular. A plataforma deve executar comandos de movimentação, os quais são: translação para frente e para trás, e rotação, no sentido horário e anti horário. Um sistema microcontrolado deve processar os sinais dos sensores de distância e dos encoders das rodas, e implementar rotinas de baixo nível para acessar os dados dessa leitura. Esse sistema também deve implementar rotinas de baixo nível para controlar a potência dos motores e calcular deslocamento, velocidade e aceleração linear e angular da plataforma.

A plataforma deve ser extensível a qualquer outro sistema de processamento. O microcontrolador daquela deve implementar rotinas de baixo nível para comunicação de dados. A plataforma deve apresentar interface de alto nível para leitura dos dados dos sensores e recebimento de comandos de movimentação.

A documentação de hardware e software da plataforma deve atender aos requisitos de um projeto de engenharia. [Aqui se definem os diagramas, documentos, etc, exemplo, diagrama de blocos, diagrama esquemático, diagramas UML (diagrama de casos de uso, diagrama de classes, diagrama de sequência, diagrama de estados, etc.)].

Os dois algoritmos de navegação serão implementados em um host interligado à plataforma robótica. Esse host será capaz de ler os dados dos sensores do robô e enviar comandos de movimentação ao mesmo.

Os testes desses algoritmos serão realizados inicialmente em um ambiente aberto e livre de obstáculos, onde o robô deverá deslocar-se de um ponto inicial a um ponto final com velocidade constante. Posteriormente,

o ambiente será limitado por paredes e serão acrescentados obstáculos entre os dois pontos dessa trajetória. Em cada caso, os dois algoritmos serão experimentados sob as mesmas condições.

Através dos dados obtidos com os experimentos, será escrito um relatório de resultados onde é analisado o comportamento de cada um desses dois algoritmos e exposto um comparativo envolvendo os seguintes critérios: tamanho e complexidade do código, custo computacional, tempo de resposta e eficácia.

O trabalho deverá ser apresentado na forma de monografia onde serão descritas todas as atividades desenvolvidas nesse projeto e relatados os resultados obtidos.

1.2 Objetivos Específicos

A plataforma robótica desse projeto será construída aproveitando-se os componentes de hardware e software do robô Bellator, o qual se encontra desmontado e fora de condições de uso.

Será proposto um projeto de engenharia o qual especifica a reconstrução do robô. Inicialmente, este será analisado para identificar componentes de hardware defeituosos ou obsoletos e para realizar uma avaliação da situação do software instalado. O projeto de engenharia deverá contemplar a descrição dessa análise.

Os componentes de hardware defeituosos deverão ser trocados por novos e os obsoletos serão atualizados, sob a condição de que essa atualização seja vantajosa e não implique severas alterações no hardware original. O software avaliado será aproveitado e uma análise será proposta para levantar novos requisitos de implementação. O projeto de engenharia deverá contemplar a descrição das alterações em hardware e a análise de requisitos do software.

O projeto de hardware e software deve especificar uma plataforma capaz de atender aos requisitos apresentados, em linhas gerais, no objetivo geral desse trabalho.

O projeto será submetido aos orientadores para avaliação e aprovação. Uma vez aprovado, dar-se-á início a implementação. Concluída essa etapa, a plataforma será submetida a testes de unidade para validar as rotinas de leitura de sensores e envio de comandos de movimentação.

Estando a plataforma testada, será proposto um projeto no qual serão especificados os algoritmos de navegação a serem implementados. Uma revisão bibliográfica dos algoritmos de navegação fuzzy deverá ser

apresentada.

O projeto de implementação dos algoritmos e a revisão bibliográfica serão apresentados aos orientadores para aprovação. Quando aprovados, dar-se-á início a implementação dos mesmos. Concluída essa etapa, os algoritmos serão testados em ambientes sob as condições descritas no objetivo geral.

A monografia deverá apresentar a descrição dos resultados obtidos no projeto, implementação e testes da remontagem do robô e do projeto, implementação e testes dos algoritmos de navegação. Os resultados relativos a estes deverão apresentar um comparativo cujos critérios de comparação ainda estão para ser definidos.

Vale salientar que um dos objetivos específicos deste trabalho é realizar um projeto de engenharia sob a orientação dos professores. Deverá ser apresentado um cronograma de atividades, levantamento de requisitos dos projetos citados, implementação e testes de unidade. Simula-se com isso o desenvolvimento de um produto em que os professores são os clientes. Será entregue além da plataforma em funcionamento, uma documentação técnica do hardware e do software da mesma. [Aqui verificar com os professores o tipo de documentação técnica a ser entregue].

2 Metodologia

Atualmente, o robô Bellator está desmontado e a equipe não sabe o que está funcionando ou não. A primeira etapa do projeto consiste em obter acesso ao robô e analisar o que pode ser aproveitado e o que ainda falta ser feito.

Após esta análise inicial, será necessário testar todas as funcionalidades do robô (acionamento dos motores, comunicação entre as camadas, funcionamento dos sensores, encoders, entre outros). Então, serão feitos os ajustes necessários para adequar a plataforma robótica às necessidades do projeto.

Ao final da reconstrução do robô, dois algoritmos fuzzy serão testados nele e a eficiência dos algoritmos de navegação será comparada.

[Aqui especificar melhor o passo a passo]

2.1 Fundamentos

Por enquanto, o único paradigma definido é o *ED-FCM (Event-Driven Fuzzy Cognitive Maps)* para a navegação do robô.

2.2 Tecnologias

A seguir serão descritas as principais tecnologias que serão utilizadas durante este projeto, tanto de hardware como de software.

- A linguagem de programação utilizada será C (para a implementação dos algoritmos de navegação e lógica de controle do robô).
- O software MATLAB também será utilizado para a realização de alguns testes (principalmente o paradigma ED-FCM). [6]

3 RECURSOS DE HARDWARE E SOFTWARE

Nesta seção serão descritos os principais recursos de software e hardware a serem utilizados neste projeto, com mais detalhes em relação a seção anterior. Todos estes recursos já estão disponíveis mas ainda está pendente a aprovação para a utilização por parte do Professor Heitor Silvério Lopes.

3.1 Recursos de Hardware

O Robô Bellator possui uma placa C8051F340, na qual será implementada a lógica de controle, e uma placa mãe VIA EPIA M-Series Mini-ITX Mainboard, onde podem ser executados os algoritmos de navegação.

Conjunto de materiais do robô

- 1 Unidade Micro-Controlada C8051F340
- 1 VIA EPIA M-Series Mini-ITX Mainboard
- 6 Sensores de distância IR 2Y0A02F98
- 1 Placa de Roteamento e Alimentação dos Sensores
- 2 Motores Bosch FPG 12V
- 2 Pontes H L 298
- 1 Bateria Unybatt 12V-7.2 Ampére-hora

3.2 Recursos de Software

Como já foi mencionado anteriormente, a linguagem de programação utilizada será C (para a implementação dos algoritmos de navegação e lógica de controle do robô). Além disso, o software MATLAB também será utilizado para a realização de alguns testes dos algoritmos de navegação.

3.3 Viabilidade

A viabilidade (tanto técnica quanto financeira) do projeto depende do acesso ao robô para testes e verificação de seu estado atual. A equipe só pôde até então visualizar o robô desmontado, com suas peças empilhadas em uma caixa.

4 Contexto

O projeto em questão é uma proposta apresentada pelo professor João Alberto Fabro (orientador) do Departamento de Informática (DAINF), pelo professor Heitor Silverio Lopes (atual responsável pelo robô) do Departamento de Eletrônica (DAELN) , e pelo pesquisador Márcio Mendonça, atualmente realizando seu doutorado pelo Programa de Pós-Graduação em Engenharia Industrial (CPGEI).

Mendonça está desenvolvendo uma pesquisa em navegação robótica envolvendo abordagem FCM. Em seu trabalho, ele propõe um aprimoramento dessa metodologia, a ED-FCM. Contudo, atualmente o modelo proposto é simulado computacionalmente via MATLAB. A proposta é embarcar o algoritmo de navegação ED-FCM em hardware, possibilitando avaliar o modelo em uma plataforma robótica real e sob condições reais de operação. Após isso, esse trabalho será estendido, onde será implementado outro modelo de navegação e, ao final, será apresentada uma comparação entre eles.

5 Projeto a ser desenvolvido e resultados iniciais

Esta seção visa uma descrição mais detalhada de como será realizado o projeto e qual a função de cada parte do mesmo, contendo os projetos de software e hardware.

5.1 Projeto de Software

Para o projeto de software, inicialmente foi feito um levantamento de requisitos, onde os seguintes requisitos foram definidos:

- **Requisitos Funcionais:**

O Software Embarcado está dividido em 2 partes: O software de controle da placa C8051F340 e o software da VIA EPIA M-Series Mini-ITX Mainboard (abreviada para Mainboard). Toda a comunicação mencionada a seguir será feita via porta serial.

C8051F340:

1. Receber leitura dos sensores
2. Enviar sinais de controle ao robô
3. Receber comandos da Mainboard
4. Enviar dados de Navegação para a Mainboard

Mainboard:

1. Receber dados de Navegação da C8051F340
2. Executar o algoritmo de Navegação
3. Enviar comandos para a C8051F340

Requisitos Não-Funcionais:

1. Compatível com o protocolo de mensagens do robô Bellator

Em seguida, foi feito o diagrama de casos de uso, que segue abaixo:

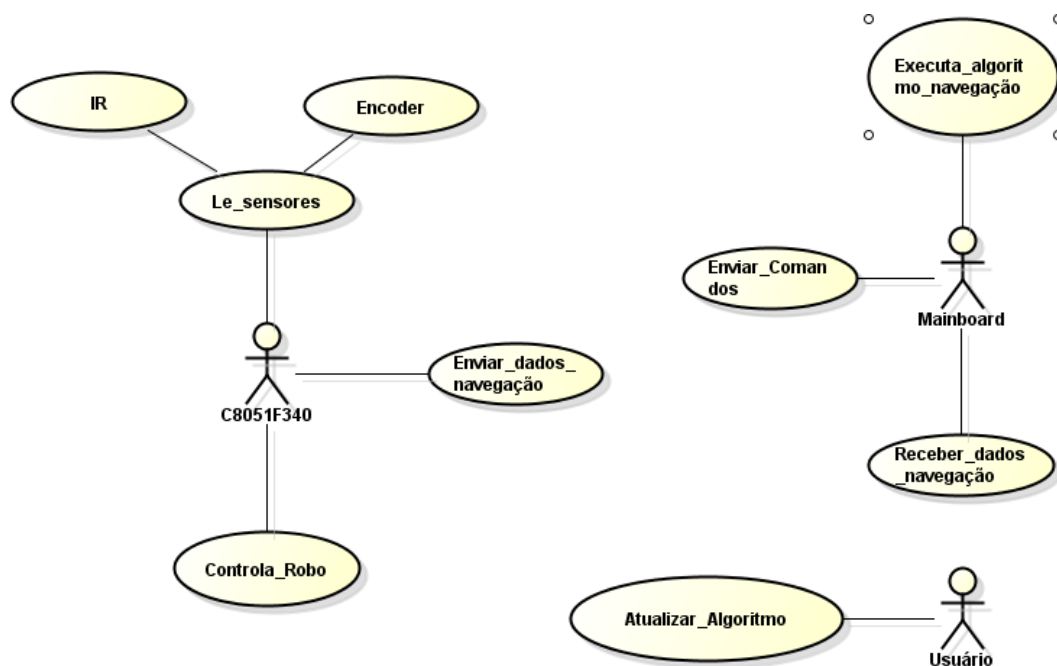


Figura 1 - Diagrama de Casos de Uso
Especificação de Casos de Uso

A seguir, a especificação dos casos de uso presentes no diagrama de Casos de Uso.

Caso de Uso: Le_sensores

Descrição: Descreve a leitura de todos os sensores do robô

Ator Principal: C8051F340

Ator de Suporte:

Ator de Bastidor:

Pré-condições:

1. O robô está ligado e pronto para executar comandos
2. Os sensores estão conectados e alimentados apropriadamente

Pós-condições:

1. O robô possui dados atualizados sobre o valor de leitura atual de cada sensor.

Fluxo Básico:

1. Para cada sensor, o robô lê o valor atual disponível na porta correspondente ao robô.
2. O robô converte este valor para a leitura correspondente do sensor e armazena estes dados.

Fluxo Alternativo:

Regras de Negócio:

Caso de Uso: Enviar_dados_navegação

Descrição: Descreve o envio de dados dos sensores para a Mainboard

Ator Principal: C8051F340

Ator de Suporte: Mainboard

Ator de Bastidor:

Pré-condições:

1. O robô está ligado e a C8051F340 está conectada à Mainboard
2. A C8051F340 possui uma nova leitura de todos os sensores

Pós-condições:

1. Os dados foram enviados por via serial para a Mainboard

Fluxo Básico:

1. A C8051F340 envia os dados de acordo com o protocolo de comunicação do robô Bellator

Fluxo Alternativo:

Regras de Negócio:

Caso de Uso: Executa_algoritmo_navegacao

Descrição: Caso de uso para o robô executar um algoritmo de navegação selecionado

Ator Principal: Mainboard

Ator de Suporte: Usuário

Ator de Bastidor:

Pré-condições:

1. Existe pelo menos um algoritmo de navegação carregado no robô.
2. O robô está ligado e pronto para executar comandos.

Pós-condições:

1. O robô está executando o algoritmo de navegação carregado e navegando no ambiente onde se encontra.

Fluxo Básico:

1. O usuário liga o robô
2. A Mainboard inicializa e executa o algoritmo de navegação

Fluxo Alternativo: 1a. Se nenhum algoritmo de navegação está carregado, o robô não faz nada.

Regras de Negócio:

Caso de Uso: Enviar_comandos

Descrição: Caso de uso que descreve como a Mainboard envia comandos de controle à C8051F340.

Ator Principal: Mainboard

Ator de Suporte:

Ator de Bastidor:

Pré-condições:

1. Robô ligado com C8051F340 conectada à Mainboard

Pós-condições:

1. A C8051F340 recebeu os comandos da Mainboard e os executa

Fluxo Básico:

1. A Mainboard monta a mensagem contendo o comando de controle e de acordo com o protocolo do Bellator.
2. A Mainboard envia mensagem à C8051F340.
3. A C8051F340 recebe os comandos e os executa

Fluxo Alternativo:

Regras de Negócio:

Caso de Uso: Controla_Robo

Descrição: Caso de uso que descreve como a C8051F340 controla os motores do robô

Ator Principal: C8051F340

Ator de Suporte:

Ator de Bastidor:

Pré-condições:

1. Robô ligado com a C8051F340 conectada às pontes H que controlam o robô
2. A C8051F340 possui um comando para executar

Pós-condições:

1. Controle do robô efetuado

Fluxo Básico:

1. A C8051F340 envia sinais de acionamento para às pontes H de acordo com o comando da Mainboard

Fluxo Alternativo:

Regras de Negócio:

Caso de Uso: Receber_dados_navegação

Descrição: Caso de uso que descreve como a Mainboard recebe e interpreta dados recebidos da C8051F340

Ator Principal: Mainboard

Ator de Suporte: C8051F340

Ator de Bastidor:

Pré-condições:

1. Robô ligado com C8051F340 conectada à Mainboard
2. A C8051F340 enviou dados novos dos sensores para a Mainboard

Pós-condições:

1. A Mainboard possui a posição atualizada aproximada do robô.

Fluxo Básico:

1. Os novos dados recebidos da C8051F340 são armazenados
2. Os novos dados dos sensores são interpretados para deduzir a posição aproximada do robô
3. A nova posição é armazenada e utilizada com o algoritmo de navegação

Fluxo Alternativo:

Regras de Negócio:

Caso de Uso: Atualizar_Algoritmo

Descrição: Caso de uso que descreve a atualização do algoritmo de navegação

Ator Principal: Usuário.

Ator de Suporte: Software Embarcado.

Ator de Bastidor:

Pré-condições:

1. O Robô está ligado e existe uma conexão de rede entre o robô e um computador no qual encontra-se o código do algoritmo atualizado

Pós-condições:

1. O Algoritmo de navegação está programado para execução no robô

Fluxo Básico:

1. É feita uma cópia do novo algoritmo do computador para a Mainboard através da conexão de rede
2. O robô é programado para executar o algoritmo na próxima inicialização
3. A conexão é desativada e o robô é desligado.

Fluxo Alternativo:

Regras de Negócio:

5.2 Projeto de Hardware

Nesta seção o projeto será descrito em módulos, e cada módulo terá uma breve descrição de suas funções e requisitos. Para isto, segue um diagrama em blocos (visão alto nível) do projeto.

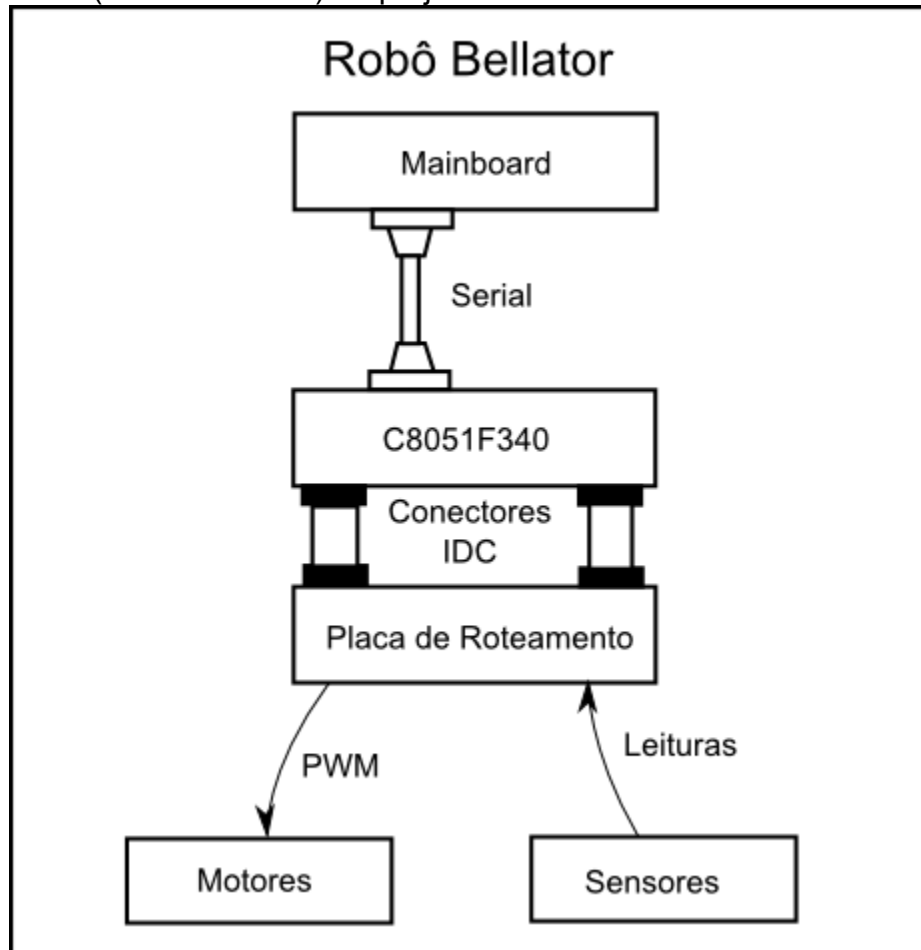


Figura 2 - Diagrama em blocos do projeto

Descrição dos Módulos Funcionais

Mainboard:

É o módulo responsável por executar os algoritmos de navegação desenvolvidos e realizar a comunicação com a C8051F340. Ele deverá:

- Comunicar-se com a C8051F340 via interface serial;
- Executar os algoritmos de navegação desenvolvidos;

C8051F340:

O controle do robô é realizado pela placa C8051F340, responsável por enviar sinais de controle para realizar a movimentação do robô e realizar a leitura dos sensores do mesmo. Este módulo faz parte da plataforma Bellator e é composto pelos blocos principais:

- Conexão com os sensores de distância e encoders através da placa de roteamento;
- Conversores A/D para leitura dos sensores;
- Módulos de interface serial via SCI;

Motores e Sensores:

São a interface da plataforma robótica com o ambiente. São responsáveis por:

- Movimentação do robô
- Prover leituras da distância do robô a obstáculos ao seu redor
- Prover informações sobre a rotação das rodas

6 Procedimentos de teste e validação

O procedimento de teste e validação dos módulos seguirá a metodologia de teste de software "Caixa-Cinza", que consiste de uma mescla dos métodos "Caixa-Branca" e "Caixa-Preta". Isto significa que, para a realização dos testes, serão consideradas as estruturas interna e externa do software, ou seja, será analisado o comportamento dos módulos a partir da análise do código e a partir da análise da saída dada pelo código para certa entrada.

Serão realizados três tipos de testes:

Testes de unidade

Compõem a primeira fase de testes, onde cada módulo do sistema será avaliado isoladamente.

Nesta fase serão testados os seguintes módulos:

- Comunicação serial entre a Mainboard e a placa C8051F340;
- Funcionamento e acionamento correto dos motores;
- Funcionamento dos sensores
- Interpretação das mensagens recebidas na C8051F340 da Mainboard;

Testes de integração

Sucedem a fase de testes de unidade. Nessa etapa, os módulos serão combinados e testados em grupo.

Pode ser visualizada como a fase onde os seguintes casos de uso serão testados:

- Leitura dos sensores (Caso de uso: Le_sensores);
- Envio de comandos para o robô (Caso de uso: Enviar_comandos);
- Execução de comandos pela C8051F340 (Caso de uso:Controla_Robo);

Teste de sistema

Sucedem a fase de testes de integração. Nessa etapa, o sistema será testado como um todo e compreenderá o seguinte teste:

- Executar um algoritmo de navegação no robô.

Critérios de Aceitação para os Testes e Validações

Os critérios para a aceitação dos testes descritos anteriormente serão:

- Para os testes de unidade, é necessário que o tempo de resposta na comunicação serial seja adequado. Além disso, o resultado da

interpretação das mensagens deve ser um comando válido para o robô.

- Para os testes de integração, é necessário que o fluxo de cada caso de uso seja obedecido;
- Para o teste de sistema, é necessário que o robô responda corretamente durante a execução do algoritmo de navegação embarcado.

7 Análise de Riscos

A seguir será feita a análise de riscos relativa ao projeto. Cada risco é identificado, classificado com um impacto e probabilidade de 1 a 5, com as devidas justificativas.

- Risco: O Robô Bellator é danificado.

1ª Etapa: Identificação do Risco

Denominação do risco: O robô Bellator é danificado.

Descrição do risco: O robô Bellator é danificado durante seu manuseio.

2ª Etapa: Avaliação do Risco

Impacto: (x) 5 Alta () 4 Média/Alta () 3 Média () 2 Média/Baixa () 1 Baixo

Explique:

- Se o robô for ligado de forma inadequada ou alguma implementação em hardware estiver errada, o robô poderá ser danificado.
- Se a roda do robô ficar presa, o motor poderá queimar.
- Algum curto-circuito pode danificar o hardware do robô.
- Batidas decorrentes da queda do robô ou colisões com obstáculos também podem provocar danos.

Probabilidade: () 5 Alta () 4 Média/Alta () 3 Média (x) 2 Média/Baixa () 1 Baixo

Explique:

Considera-se probabilidade média/baixa como a resultante de todos os riscos que podem haver durante o manuseio do robô.

3ª Etapa: Desenvolvimento da Resposta do Risco

Prevenir:

- Sempre designar o responsável que estiver manuseando o robô.
- Não manuseá-lo enquanto o mesmo estiver ligado.
- Isolar as placas do casco do robô com folha de papel ou fixá-las com parafusos.
- Sempre que houver nova ligação em hardware certificar-se de que as ligações estão corretas.
- Evitar colisões do robô com obstáculos.

- Risco: A placa C8051F340 é danificada.

1ª Etapa: Identificação do Risco

Denominação do risco: A placa C8051F340 é danificada.

Descrição do risco: A placa C8051F340 é danificada durante o uso.

2ª Etapa: Avaliação do Risco

Impacto: (x) 5 Alta () 4 Média/Alta () 3 Média () 2 Média/Baixa () 1 Baixo

Explique:

Se a placa C8051F340 for ligada de forma inadequada ou alguma ligação em hardware estiver incorreta, a mesma poderá ser danificada.

Probabilidade: () 5 Alta () 4 Média/Alta () 3 Média (x) 2 Média/Baixa () 1 Baixo

Explique:

Considera-se probabilidade de nível 2 pois esse risco apesar de pouco provável é plausível e pode acontecer durante a utilização da placa.

3ª Etapa: Desenvolvimento da Resposta do Risco

Prevenir:

- Sempre designar o responsável que estiver manuseando a placa.
- Não manuseá-la enquanto a mesma estiver ligada.
- Sempre que houver nova ligação em hardware certificar-se de que as ligações estão corretas.

- Risco: Perda dos códigos fonte.

1ª Etapa: Identificação do Risco

Denominação do risco: Perda dos códigos fonte.

Descrição do risco: O dispositivo em que estão armazenados os arquivos de código fonte estraga.

2ª Etapa: Avaliação do Risco

Impacto: () 5 Alta () 4 Média/Alta (x) 3 Média () 2 Média/Baixa () 1 Baixo

Explique: Considera-se impacto médio porque há possibilidade de recuperar os arquivos em um backup.

Probabilidade: () 5 Alta () 4 Média/Alta () 3 Média () 2 Média/Baixa (x) 1 Baixo

Explique: Considera-se probabilidade nível 1 porque pode-se armazenar os códigos em um servidor qualquer, diminuindo muito as chances de perdê-los (seria necessário perder tanto a cópia no computador pessoal quanto a cópia do servidor).

3ª Etapa: Desenvolvimento da Resposta do Risco

Prevenir:

Evitar armazenar os códigos fontes somente em pendrives ou outros dispositivos de armazenamento móveis de pouca confiança.

Manter um repositório de arquivos em um equipamento central, por exemplo, um computador pessoal.

Efetuar backup dos arquivos de código fonte.

Fazer upload do código para um servidor quando houverem mudanças

significativas

- Risco: Atrasos no Desenvolvimento do Projeto.

1ª Etapa: Identificação do Risco

Denominação do risco: Atraso no Desenvolvimento do Projeto.

Descrição do risco: O desenvolvimento do projeto é atrasado devido a dificuldades na implementação dos diversos módulos do sistema a ser desenvolvido no projeto.

2ª Etapa: Avaliação do Risco

Impacto: () 5 Alta () 4 Média/Alta () 3 Média (x) 2 Média/Baixa () 1 Baixo

Explique: O Impacto deste risco não é muito elevado visto que não há um deadline bem definido para o fim do projeto, a data de término é flexível.

Probabilidade: () 5 Alta (x) 4 Média/Alta () 3 Média () 2 Média/Baixa () 1 Baixo

Explique: A probabilidade de ocorrência de atrasos durante o desenvolvimento do projeto é média/alta, pois existem vários fatores que contribuem para isso, tais como:

- Problemas ao remontar o robô.
- Necessidade de pesquisa adicional para resolver problemas que inicialmente não existiam.
- Problemas ao implementar novas partes do robô e/ou surgimento de novas tarefas após a análise da situação do robô.

Estes fatores podem causar atrasos em diversos estágios do desenvolvimento do projeto e, por consequência, um possível atraso no cronograma do projeto.

3ª Etapa: Desenvolvimento da Resposta do Risco

Prevenir:

Realizar testes de unidade e integração com frequência, sempre que forem realizadas alterações em módulos e códigos que são críticos na integração do sistema, visando detectar e resolver problemas de integração rapidamente, antes que estes causem atrasos maiores.

- Risco: Indisponibilidade temporária do robô Bellator

1ª Etapa: Identificação do Risco

Denominação do risco: Indisponibilidade temporária do robô Bellator.

Descrição do risco: O robô poderá ser usado por outras pessoas e pode se tornar indisponível por algum tempo.

2ª Etapa: Avaliação do Risco

Impacto: () 5 Alta (x) 4 Média/Alta () 3 Média () 2 Média/Baixa () 1 Baixo

Explique: Considera-se impacto de nível 4 já que a reconstrução do robô

é o foco do projeto, assim como a adição de algumas funcionalidades ao mesmo.

Probabilidade: () 5 Alta () 4 Média/Alta () 3 Média (x) 2 Média/Baixa () 1 Baixo

Explique:

A probabilidade é média/baixa pois nas conversas com o professor responsável pelo robô foi mencionado que ninguém está trabalhando com o robô, e não existe outra pessoa e/ou equipe tentando conseguir acesso ao mesmo.

3ª Etapa: Desenvolvimento da Resposta do Risco

Prevenir:

Manter a comunicação com o responsável pelo robô para que possíveis indisponibilidades não surpreendam a equipe.

- Risco: Sistema apresenta latência elevada.

1ª Etapa: Identificação do Risco

Denominação do risco: Sistema apresenta latência elevada.

Descrição do risco: Após o teste do sistema como um todo, é constatado que a latência do sistema é muito elevada.

2ª Etapa: Avaliação do Risco

Impacto: () 5 Alta (x) 4 Média/Alta () 3 Média () 2 Média/Baixa () 1 Baixo

Explique: Embora a latência elevada não seja muito prejudicial para a maioria das funções do robô, ela pode ser prejudicial para a plotagem dos pontos de trajeto do robô real no simulador, visto que isto deve ser feito em tempo real. O impacto disto é elevado pois, a determinação de que a latência do sistema é elevada só pode ser constatada quando todo o sistema já estiver, mesmo que de forma limitada, funcional e, além disso, pode ser difícil detectar a causa da latência neste estágio do projeto.

Probabilidade: () 5 Alta () 4 Média/Alta () 3 Média (x) 2 Média/Baixa () 1 Baixo

Explique: A probabilidade de ocorrência deste risco não é muito elevada.

Código de comunicação lento ou ineficiente e problemas de integração não detectados podem causar latência elevada no sistema.

3ª Etapa: Desenvolvimento da Resposta do Risco

Prevenir:

Realizar testes de integração focados na medição da latência gerada por cada módulo.

Desenvolver código de comunicação eficiente, com protocolo que evite grandes atrasos na comunicação.

8 Cronograma do Projeto

Segue o cronograma considerando que a equipe pode começar seus trabalhos com o robô no início de abril. Sujeito à grandes mudanças dependendo tanto do acesso ao robô quanto da situação do mesmo.

[illegible]

9 Conclusões

A atividade realizada inicialmente constituiu-se principalmente na escolha do tema do projeto e de um professor que se responsabilizasse pela orientação do mesmo. As tarefas seguintes envolveram a listagem das atividades do projeto, confecção do diagrama PERT/CPM, levantamento de requisitos, formulação do diagrama de casos de uso, descrição dos Use Case Points, descrição do projeto de hardware, análise de riscos e determinação do cronograma.

Resta uma pendência em relação ao acesso ao robô para averiguação do seu estado atual, só então será possível determinar a viabilidade do projeto.

Na opinião de Mendonça, esse projeto aborda um tema inovador e de forte interesse acadêmico, situação a qual poderá render publicações de artigos ao nível nacional e internacional, futuramente.

10 Referências Bibliográficas

- [1] FRACASSO, PAULO T.; COSTA, ANNA H.R. Navegação Reativa de Robôs Móveis Autônomos Utilizando Lógica Nebulosa com Regras Ponderadas, 2005
- [2] MENDONÇA, M.; ARRUDA, L.V.R.; NEVES, F. Qualitative Autonomous Navigation System Employing Event Drive-Fuzzy Cognitive Maps and Fuzzy Logic, 2010
- [3] MATLAB - The Language Of Technical Computing. Disponível em <<http://www.mathworks.com/products/matlab/>>. Acesso em: 07 abr. 2010.