

VICTOR ANTONIO CACCIACARRO LINCOLN

**INTERFACE DE
RECONHECIMENTO DE VOZ E
CONTROLE DA PLACA *PRODIGY*
MOTION APLICADA A ROBÓTICA
CIRÚRGICA**

Trabalho de Conclusão de Curso apresentado
à Escola de Engenharia de São Carlos, da
Universidade de São Paulo

Curso de Engenharia Elétrica com ênfase em
Eletrônica

ORIENTADORA: Profa. Dra. Liliane Ventura
CO-ORIENTADORA: Dra. Idágene Aparecida Cestari

São Carlos
2007

Dedicatória

Aos meus pais, pelo apoio sempre incondicional e pela educação primordial.

A minha namorada Leila, pela força nas horas mais importantes, sempre com sua dedicação e ajuda inestimável.

Agradecimentos

Aos meus pais, familiares e amigos pelo apoio e por acreditarem em mim em todas as fases da minha vida.

A Dra. Idágene Aparecida Cestari pelo dedicado e eficiente trabalho de orientação e pela oportunidade. Espero sinceramente ter correspondido.

A Dra. Liliane Ventura Schiabel pela confiança no meu trabalho e seu esforço na minha orientação e formação acadêmica.

A equipe do Laboratório de Eletrônica da Divisão de Bioengenharia, pelo apoio técnico e humano.

A todas as pessoas que direta ou indiretamente contribuíram para a execução deste trabalho.

Índice

Introdução.....	9
1. Cirurgia Minimamente Invasiva.....	10
2. Cirurgia Auxiliada por Robôs	12
3. Metodologia.....	13
3.1. Interface de Comando por Voz	13
3.1.1. Algoritmo de Reconhecimento de Comandos de Voz	13
3.1.2. Módulo de Gerenciamento de Tarefas	14
3.2. Parâmetros de Análise.....	17
3.2.1. Sistemas de Controle e Acionamento	17
3.2.2. Motor de Passo – Características	17
3.3. <i>Prodigy Motion Card</i>	18
3.3.1. <i>Placa Prodigy Motion Card</i>	18
3.3.2. <i>Software Pro-Motion</i>	19
4. Resultados	20
4.1. Testes no <i>Software Pro-Motion</i>	20
4.2. Algoritmo “Interface PMC”	25
4.3. Algoritmo “PMCVox”	29
5. Discussão e Conclusão	34
6. Referências Bibliográficas	35

Lista de Figuras

Figura 1: Esquema de um endoscópio rígido, com os pontos de acoplamento da câmera CCD e da fibra óptica para iluminação do campo. (SANTOS, E; et al, 2004).	10
Figura 2: Esquema dos instrumentos utilizados na cirurgia minimamente invasiva. (SANTOS, E; et al, 2004).	11
Figura 3: Diagrama de blocos do algoritmo de reconhecimento de voz utilizando a biblioteca de programação IBM SAPI. (SANTOS, E; et al, 2004).	14
Figura 4: Diagrama de blocos do módulo de gerenciamento de tarefas. (SANTOS, E; et al, 2004).	15
Figura 5: Diagrama de estados do módulo de gerenciamento de tarefas. NR = Não-reconhecido; T1 = Temporizador 1; T2 = Temporizador 2. (SANTOS, E; et al, 2004).	16
Figura 6: Diagrama de Configuração com 4 Eixos de Ação. (PMD, 2007).	18
Figura 7: Diagrama de Blocos Interno <i>Prodigy-PCI</i> (PMD, 2007).	19
Figura 8: Gráfico Trapezoidal – Movimento Único.	21
Figura 9: Gráfico Trapezoidal – Movimento Manual.	21
Figura 10: Gráfico Trapezoidal – Movimento Automático.	21
Figura 11: Gráfico de Controle por Velocidade – Movimento Manual.	22
Figura 12: Gráfico Curva S – Movimento Automático.	22
Figura 13: Gráfico Trapezoidal – 4 Eixos Simultâneos – Mov. Automático.	23
Figura 14: Gráfico Trapezoidal – 2 Eixos Simultâneos – Mov. Automático.	24
Figura 15: Gráfico Trapezoidal – 2 Eixos Simultâneos – Mov. Automático Distintos.	24
Figura 16: Diagrama de Blocos Algoritmo “Interface PMC”	26
Figura 17: <i>Software</i> “Interface PMC”	27
Figura 18: Menu do <i>software</i> “Interface PMC”.	27
Figura 19: Arquivo texto de saída – “Interface PMC”. Esquerda: Eixos isolados. Direita: Eixos simultâneos, entrada Analógica.	28
Figura 20: Diagrama de Blocos Algoritmo “Interface PMC”	30
Figura 21: <i>Software</i> “PMCVox”. Esquerda: Mensagem de Estado. Direita: Palavra reconhecida	31
Figura 22: Menu do <i>software</i> “PMCVox”.	31
Figura 23: Arquivo texto de saída - “PMCVox”. Esquerda: Comandos de movimento. Direita: Comandos de parada e espera do sistema.	32
Figura 24: Interface de configuração do <i>Software</i> “Interface PMC”	33

Lista de Tabelas

Tabela 1: Vocabulários Hierárquicos.	15
Tabela 2: Motor de Passo – Pulso e Direção – Eixo 1.	20
Tabela 3: Motor de Passo – Pulso e Direção – Integração dos Eixos.	23
Tabela 4: Características funcionais do <i>software</i> “Interface PMC”.	25
Tabela 5: Protocolo de Correlação (Comando de voz – Movimento – Eixo).....	29

Resumo

É proposta a interface entre o sistema de reconhecimento de voz desenvolvido com a biblioteca de programação SMAPI (*Speech Manager Application Programming Interface*) da IBM e o módulo robótico composto pela placa de controle (*Prodigy-PCI Motion Cards*). A interface consiste em: um algoritmo com as rotinas padrões de configuração da placa de controle, apresentando uma interface simples de ajustes e execução de comandos; um sistema de reconhecimento de voz otimizado para a estrutura da interface robótica - essa interface é realizada por uma interação direta entre os sistemas e interpreta os comandos de voz em ações definidas; e uma rotina básica em linguagem de programação "C++", que foi implementada para a interação entre as palavras de comando, configuração da placa e acionamento do motor.

Palavras-Chave: Robótica Cirúrgica; *Prodigy Motion Card*; Reconhecimento de Voz; SMAPI; Cirurgia Minimamente Invasiva.

Abstract

An interface between a voice recognition system developed using the IBM SAPI (Speech Manager Application Programming Interface) programming library, and the robotic system composed of a motion control board (*Prodigy-PCI Motion Cards*) for robotic applications is proposed. The interface is composed by: an algorithm with the standard configuration routines of board control - developed for presenting a simple interface for adjustments and execution of commands; the optimization of voice recognition system for the structure of the robotic interface - performed by a direct interaction between systems that interpret voice commands into defined actions; and the algorithm for interface controlling which combines the designed algorithms using simple routines in the programming language "C++" for the interaction of the words of command, configuration of the board and motion control.

Keywords: Robotic surgery; Prodigy Motion Card; Speech Recognition; SAPI; minimally invasive surgery.

Introdução

O trabalho visa utilizar técnicas de reconhecimentos de comandos de voz isolados no controle de movimentos de um posicionador robótico de endoscópio utilizado em procedimentos cirúrgicos. O sistema utiliza um vocabulário hierárquico pré-configurado com palavras de comando baseadas na biblioteca de programação SMAPI desenvolvida anteriormente [SANTOS, E; et al, 2004]; a placa de controle robótico Prodigy Motion Card; e a biblioteca de programação em C/C++, *C-Motion*.

O foco do projeto é situado no instrumento fundamental que constitui no posicionador da câmera óptica endoscópica. A movimentação da câmera endoscópica obedece comandos de voz do cirurgião, através de um microfone posicionado próximo à boca. A utilização de comandos de voz para o acionamento de manipuladores é uma estratégia muito utilizada, pois os comandos de voz são simples, intuitivos e permitem que o cirurgião permaneça com as mãos livres.

A vantagem do sistema robótico sobre o tradicional relaciona-se ao chamado "grau de liberdade de movimentação", direção na qual o instrumento pode se mover. Os sistemas de braço robótico simulam os movimentos humanos de articulação de braço, cotovelo e punho.

O projeto desenvolvido descreve o algoritmo de acesso da placa de controle, oferecendo uma interface simples que possibilita a configuração, simulação e execução de movimentos e parâmetros de controle.

O trabalho tem por objetivo a integração do sistema de reconhecimento de voz com a placa de controle. O algoritmo desenvolvido possibilita uma interface simples e responde aos comandos de voz com ações de controle.

1. Cirurgia Minimamente Invasiva

As intervenções cirúrgicas realizadas em órgãos ou tecidos internos geralmente exigem grandes incisões para que o cirurgião possa ter acesso aos tecidos e órgãos sob cirurgia. Já a cirurgia minimamente invasiva é aquela feita por meio de pequenas incisões, com a menor agressão tecidual possível. Os cirurgiões atualmente podem visualizar com segurança quase todas as partes do corpo humano através de um dispositivo óptico chamado de “endoscópio” [Cohn, M; et al, 1995], o qual é conectado por meio de um cabo de fibra óptica, a uma fonte de luz que ilumina a região operada e é acoplado a uma câmera de vídeo CCD cujas imagens são transmitidas para um monitor de vídeo. A Figura 1 esboça esquematicamente as principais partes de um endoscópio na utilização em cirurgias minimamente invasivas.

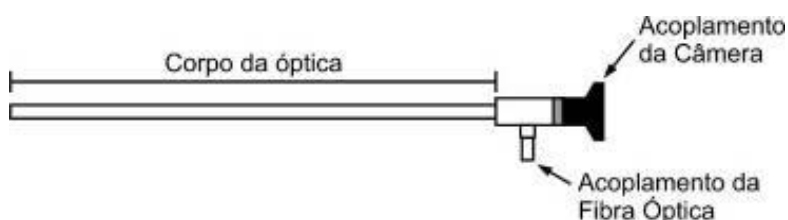


Figura 1: Esquema de um endoscópio rígido, com os pontos de acoplamento da câmera CCD e da fibra óptica para iluminação do campo. (SANTOS, E; et al, 2004).

Assim a cirurgia minimamente invasiva proporciona que o trauma à pele e músculos seja reduzido, beneficiando o paciente com uma menor dor no período pós-operatório. Outra vantagem muito importante é o índice de infecção menor, devido à mínima exposição dos tecidos, as manobras mais sutis e um menor tempo de permanência hospitalar [Cohn, M; et al, 1995; Ruurda, J; et al, 2004; Stassen, H; et al, 2001; D'atellis, N; et al, 2002; Morgan, J; et al, 2004; Georgesson, K; Owings, E; 2000; Rosen, J; et al, 1999]. Entretanto, também apresenta certas limitações, como aumento do tempo de operação [Kolvenbach, R; et al, 2004], difícil manuseio, movimentos restritos do instrumental cirúrgico e redução da sensação tátil e da percepção visual de profundidade pelo cirurgião [Cohn, M; et al, 1995; Troccaz, J; Peshkin, M; Davies, B; 1998; Falk, V; et al, 1999],.

A Figura 2 ilustra a disposição típica dos instrumentos cirúrgicos e da ótica em um procedimento minimamente invasivo.

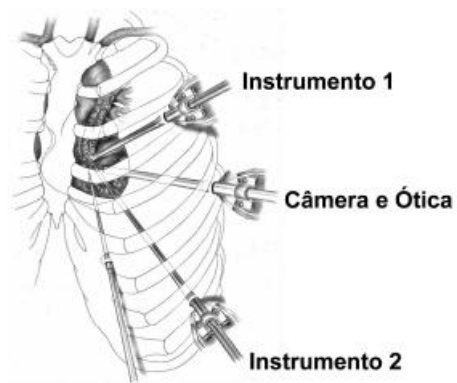


Figura 2: Esquema dos instrumentos utilizados na cirurgia minimamente invasiva. (SANTOS, E; et al, 2004).

2. Cirurgia Auxiliada por Robôs

A Robótica Cirúrgica tem como principal objetivo o aproveitamento da robótica na assistência dos procedimentos cirúrgicos e na tentativa de evitar as dificuldades das cirurgias minimamente invasivas [SABBATINI, R; 1999]. Essa assistência pode resultar em grande precisão e destreza dos movimentos do cirurgião [KAPPERT, U; et al, 2001], possibilitando novos tipos de cirurgias que anteriormente não eram realizadas de forma minimamente invasiva atualmente possam optar por esse método. A assistência robótica nos procedimentos cirúrgicos é geralmente dada por um braço posicionador, sendo o endoscópio muitas vezes controlado por voz [Slade, A; et al, 1997], o que permite ao cirurgião o manuseio do instrumental cirúrgico.

As plataformas que utilizam a robótica cirúrgica são compostas por diversos elementos que atuam em conjunto:

- Interface homem/máquina;
- Sistema gerenciado por computador, integrando as diversas partes componentes e possibilitando a interação entre o sistema robótico e o cirurgião;
- Sistemas de comando, controle e ativação dos robôs;

Outros aspectos que devem ser observados correspondem ao sistema de segurança, pois os sistemas robóticos devem seguir rigorosas exigências de segurança. Destacando-se:

- Tolerância a falhas: Os sistemas robóticos devem ser projetados para evitar danos ao paciente em casos de falha do sistema;
- Procedimentos de emergência: Os sistemas devem permitir a retirada rápida do instrumental cirúrgico para a execução de uma cirurgia convencional de emergência;
- Preservação dos tecidos adjacentes ao campo: Os instrumentos cirúrgicos e o endoscópio devem possuir movimentos limitados de modo a preservar os órgãos e tecidos nos arredores da região operada;
- Eliminação de comandos falso-positivos: Ao receber um comando não identificado, o sistema deve ignorá-lo por completo e não executar uma ação incorreta;
- Movimentos precisos: Os manipuladores devem possuir uma estrutura mecânica livre de vibrações e os sistemas de controle não devem permitir oscilações.

3. Metodologia

3.1. Interface de Comando por Voz

O algoritmo de reconhecimento de comandos de voz forma uma interface possibilitando o movimento controlado do braço robótico através de palavras de comando [SANTOS, E; 2003; SANTOS, E; et al, 2004]. A interface apresentada neste trabalho é composta por dois componentes principais:

- a) Um algoritmo de reconhecimento de comandos isolados de voz baseado na biblioteca de programação IBM SAPI anteriormente desenvolvido.
- b) Um módulo de gerenciamento de tarefas;

3.1.1. Algoritmo de Reconhecimento de Comandos de Voz

A biblioteca de programação SAPI (*Speech Manager Application Programming Interface*) desenvolvida pela IBM, possibilita o reconhecimento de comandos de voz em tempo real. Essa biblioteca é distribuída na forma dinâmica (*DLL*) que pode ser acessada por um aplicativo que utilize as suas funções.

A SAPI tem como entrada um sinal de áudio, o algoritmo reconhece a palavra que foi pronunciada relacionada com um vocabulário pré-configurado e retorna ao aplicativo uma palavra de reconhecimento (*string*). Assim fica a cargo do aplicativo a tomada de decisões através do estudo do texto recebido, relacionando cada palavra reconhecida com uma tarefa específica. Quando a palavra não é reconhecida o aplicativo comunica o usuário e não executa nenhuma ação.

O programa de reconhecimento de voz toma como base a criação de diferentes vocabulários hierárquicos em tempo de execução que podem ser ativados ou desativados de acordo com a classe das palavras e comandos que se quer identificar. No processo de reconhecimento o sinal de áudio é comparado com uma série de características espectrais, temporais, fonéticas e lingüísticas que são relacionadas com cada palavra do vocabulário, escolhendo assim a palavra com probabilidade maior com o áudio captado. Contudo, se a probabilidade de correlação for abaixo de um limiar de rejeição que pode ser configurado, o comando pronunciado é ignorado e um erro falso-negativo é registrado.

A Figura 3 um diagrama de blocos do algoritmo utilizando a biblioteca SAPI de reconhecimento de comandos de voz. Um módulo de aquisição de áudio realiza a aquisição do sinal de áudio do microfone a uma frequência de 22 kHz com resolução de 16 bits e esse sinal é pré-processado e enviado à SAPI que tenta reconhecer a palavra pronunciada. A SAPI corresponde a um módulo externo ao algoritmo, tendo

como entrada os parâmetros enviada pelo aplicativo e como saída mensagens que devem ser tratadas por um módulo particular, essas mensagens contêm a palavra reconhecida e a probabilidade do acerto estimado, já se a palavra não for identificada uma mensagem em branco é enviada o módulo de tratamento de exceções é acionado.

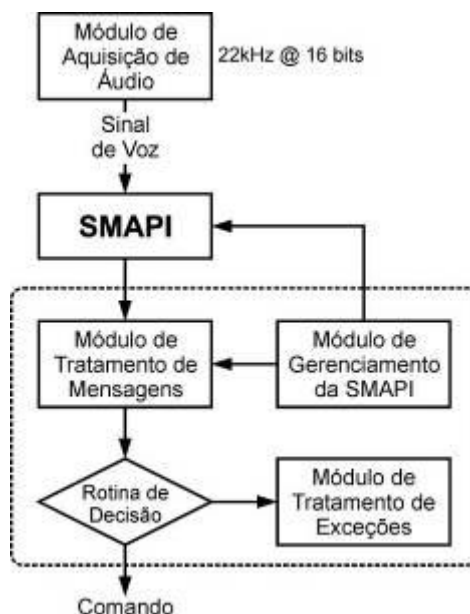


Figura 3: Diagrama de blocos do algoritmo de reconhecimento de voz utilizando a biblioteca de programação IBM SAPI. (SANTOS, E; at el, 2004).

3.1.2. Módulo de Gerenciamento de Tarefas

O módulo gerenciador de tarefas é um sistema de segurança do equipamento. O algoritmo é uma máquina de estados que recebe a palavra reconhecida pela SAPI e ativa ou desativa os vocabulários hierárquicos de maneira adequada. Esta manipulação de vocabulários tem a função de reduzir o número de palavras válidas em cada estado, minimizando o número de erros do tipo falso-positivo e evitando que o sistema robótico execute movimentos errôneos.

O conjunto de instruções possui nove palavras distribuídas em três vocabulários hierárquicos, sendo cada vocabulário associado a um estado do gerenciador de tarefas. A Tabela 1 apresenta os três vocabulários utilizados, o Vocabulário I corresponde aos movimentos que podem ser realizados pelo robô. O comando “Dorme” coloca o sistema em estado de inatividade que não é possível iniciar nenhum movimento, aumento a segurança do sistema. O Vocabulário II corresponde ao comando válido durante os movimentos. O Vocabulário III corresponde ao comando para ativar o sistema que está em estado inativo.

Tabela 1: Vocabulários Hierárquicos.

Vocabulário		
I	II	III
“Sobe”		
“Desce”		
“Esquerda”		
“Direita”	“Parado”	“Acorda”
“Avança”		
“Recua”		
“Dorme”		

A Figura 4 apresenta um diagrama de blocos do módulo de gerenciamento de tarefas. Os comandos enviados pela SMAPI são analisados por uma rotina de decisão que determina se a palavra recebida é ou não um comando válido, assim se o comando for identificado o programa gera as instruções que são enviadas ao sistema de controle pelo módulo de comunicação.

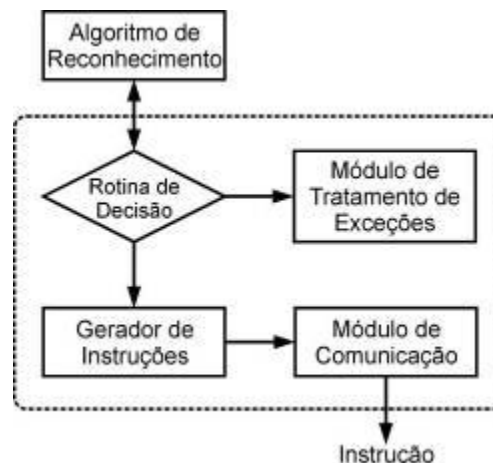


Figura 4: Diagrama de blocos do módulo de gerenciamento de tarefas. (SANTOS, E; at el, 2004).

A rotina de decisão contém a máquina de estados do algoritmo e além de gerenciar as transições de estado, ela informa ao algoritmo de reconhecimento qual vocabulário deve ser ativado ou desativado em cada estado. A Figura 5 apresenta o diagrama de estados do módulo, indicando quais palavras reconhecidas acionam as transições de estado correspondentes, uma característica importante é o fato de apenas um vocabulário hierárquico estar ativo em cada estado e que não ocorre transição de estado quando a palavra não for identificada pela SMAPI (falso-negativos, indicados na figura por “NR”), a menos que o manipulador esteja em movimento.

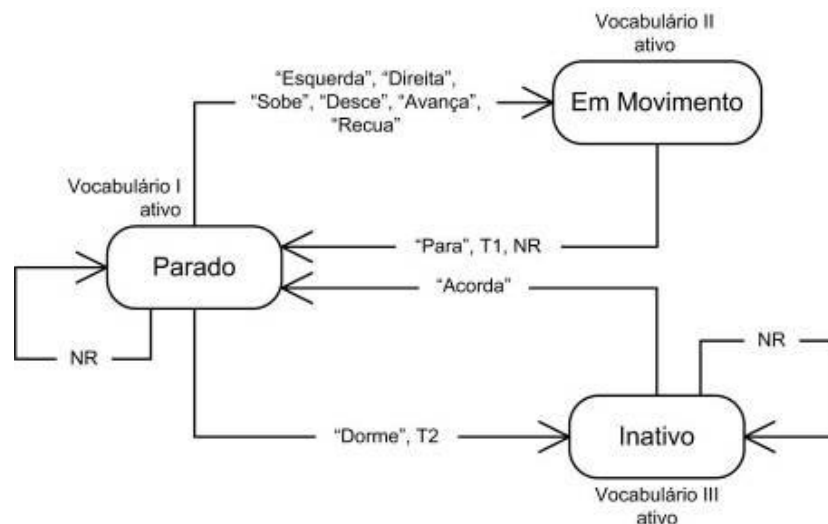


Figura 5: Diagrama de estados do módulo de gerenciamento de tarefas. NR = Não-reconhecido; T1 = Temporizador 1; T2 = Temporizador 2. (SANTOS, E; et al, 2004).

As transições de estados podem ainda ser ativadas pelos temporizadores T1 e T2, sendo o primeiro para garantir que o movimento não permanecerá por mais de 3 s mesmo que o comando de parada não seja pronunciado, já o segundo temporizador T2 garante que o sistema entre no estado inativo após 30 s sem o recebimento de palavras de comando.

3.2. Parâmetros de Análise

O sistema de controle foi escolhido seguindo a características estabelecidas abaixo, obedecendo aos critérios de segurança e o tipo de motor utilizado.

3.2.1. Sistemas de Controle e Acionamento

A escolha do tipo de motor está relacionada às especificações do robô e com as características solicitadas pelo sistema de acionamento que deve apresentar as seguintes características:

- a) Baixa velocidade (possivelmente abaixo de 60 rpm);
- b) Baixo torque e possibilidade de controle de torque;
- c) Possibilidade de implementar aspectos de segurança no *driver*;
- d) Possibilidade de controlar vários motores a partir de uma única porta;
- e) Possibilidade de interpolar trajetórias (no caso dos manipuladores articulados).

3.2.2. Motor de Passo – Características

- Possibilita a implementação de *drivers* microcontrolados o que oferece uma ótima flexibilidade de programação, já que é possível implementar uma ampla gama de algoritmos.
- Desenvolvimento de um *driver* genérico, com apenas a troca do estado de potência de acordo com o motor utilizado.
- Controle de vários motores a partir de uma porta de comunicação e um PC.
- Baixo custo

3.3. Prodigy Motion Card

3.3.1. Placa Prodigy Motion Card

A placa *Prodigy Motion Card* [PMD, 2007] oferece alto desempenho e robustez para controle de movimento aplicado à robótica, disponível na configuração *PCI* e oferecendo suporte para múltiplos tipos de motor (*DC brush*, *Brushless DC*, Motor de passo e Micropasso) e até quatro eixos de controle como ilustrado na Figura 6. Os eixos de comando representam as possibilidades de controle isolado ou simultâneo do sistema robótico, assim cada eixo controla um motor que corresponde a um grau de liberdade do braço robótico.

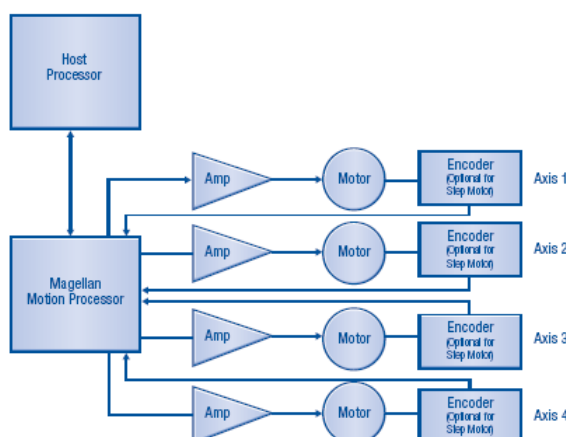


Figura 6: Diagrama de Configuração com 4 Eixos de Ação. (PMD, 2007).

Baseada na tecnologia *PMD's Magellan Motion Processor*, a placa oferece modos de ação pré-programados selecionados pelo usuário, incluindo Curva-S, Trapezoidal, Velocidade Controlada. A placa suporta parâmetros de configuração como posição, velocidade, aceleração e *Jerk*, gerando a correspondente trajetória.

O diagrama de blocos internos da placa é ilustrada na Figura 7, podendo ser dividida em três áreas de atuação:

- **Magellan Motion Processor:** São funções programáveis ligadas diretamente ao processador permitindo um controle sofisticado do comportamento global da placa.
- **Funções específicas da placa:** São funções programáveis controladas pelo processador *Magellan*, mas que acessam varias áreas da placa, incluindo as entradas e saídas digitais (I/O), comunicação e memória.
- **Processamento de Sinais e Funções de Hardware:** Uma parte substancial da placa fornece o condicionamento de sinais e outras funções associadas e não programáveis.

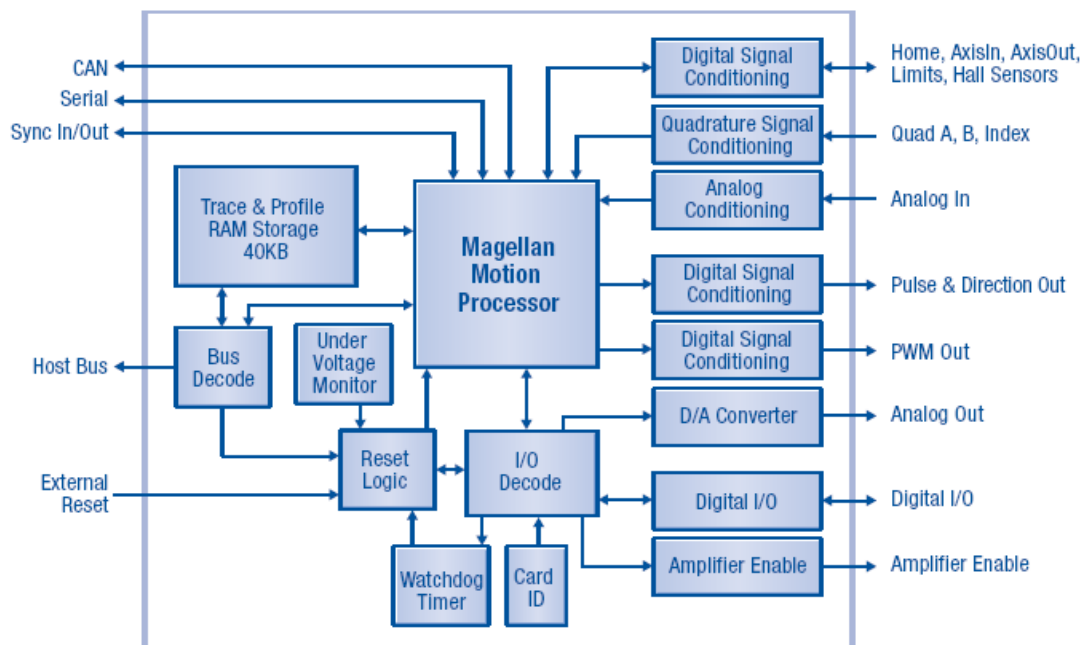


Figura 7: Diagrama de Blocos Interno *Prodigy-PCI* (PMD, 2007).

A placa disponibiliza ainda a biblioteca *C-Motion* em C/C++ que contem todas as funções necessárias para a comunicação com o processador *Magellan Motion*, permitindo a criação e integração de aplicativos desenvolvidos pelo próprio usuário, oferecendo uma ferramenta técnica no desenvolvimento de sistemas robóticos.

3.3.2. Software *Pro-Motion*

O *Pro-Motion* [PMD, 2007] é uma ferramenta oferecida para facilitar o desenvolvimento do sistema baseado no *Magellan Motion*. É um programa sofisticado, fácil de usar e que utiliza uma interface padrão. O sistema *Pro-Motion* otimiza o desenvolvimento de sistemas robóticos por facilitar o ajuste e o monitoramento de todos os parâmetros e execuções possíveis do projeto. Com ferramentas de ajuste de parâmetros e análise de dados e através de uma interface simples, facilidades como a configuração passo a passo do motor e dos eixos de trabalho, osciloscópio de monitoramento do sistema em tempo real, auto-ajustes, suporte as comunicações paralela, serial, *CAN* e *PCI*.

4. Resultados

4.1. Testes no *Software Pro-Motion*

Utilizando as ferramentas oferecidas pelo *Pro-Motion* para identificação das funções essenciais da placa de controle, parâmetros de desempenho e uma base estrutural para o desenvolvimento de um *software* novo, os seguintes testes foram realizados:

a) Análise de cada perfil de movimento (trapezoidal, velocidade controlada, Curva-S) e cada tipo de ação (automático manual e simples).

Todos os experimentos foram configurados para o motor de passo, determinado pelos parâmetros de análise, e trabalho no eixo 1, representando o movimento em um motor (um grau de liberdade).

A Tabela 2 apresenta as configurações para cada teste realizado, com o perfil utilizado, seus parâmetros de ajuste e o tipo da ação.

Tabela 2: Motor de Passo – Pulso e Direção – Eixo 1.

Perfil	Trapezoidal			Velocidade Controlada	Curva – S
Teste	A	B	C	D	E
Movimentação	Única	Manual	Automática	Manual	Automática
Aceleração (<i>step/ms²</i>)	0.01	0.01	0.01	0.01	0.01
Velocidade (<i>step/ms</i>)	0.01	0.01	0.01	±0.01 ±0.05	0.01
Posição1 (<i>step</i>)	100	100	10	100	5
Posição 2 (<i>step</i>)	-	-100	-10	-	-5
Delay (ms)	-	-	500	-	0
Jerk (<i>step/ms³</i>)	-	-	-	-	0.1

Os testes forneceram uma base para a criação de um novo *software* mais simples e proporcionou uma visão completa das funcionalidades que o *software* e a placa de controle oferecem.

- *Perfil Trapezoidal*

Movimento único: o movimento é realizado apenas uma vez até a posição determinada, na Figura 8 a execução do movimento pode ser observada.

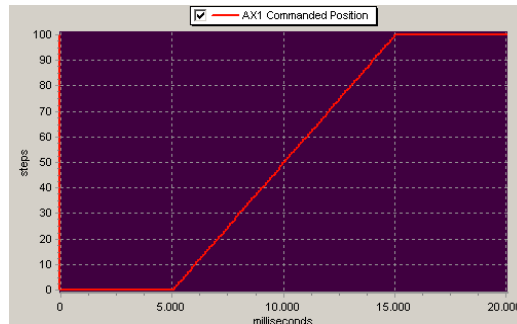


Figura 8: Gráfico Trapezoidal – Movimento Único.

Movimento Manual: o movimento é inicializado manualmente e o sistema responde quantas vezes for acionado, realizando o movimento entre as posições estabelecidas. A Figura 9 apresenta a ação para um acionamento.

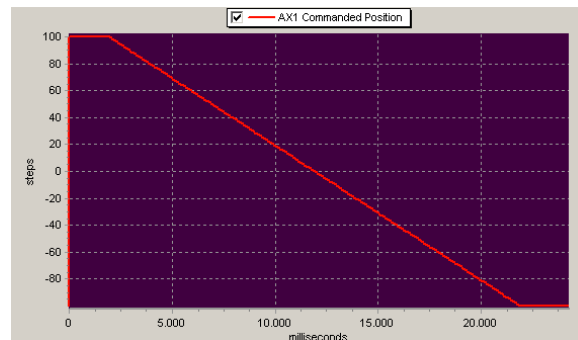


Figura 9: Gráfico Trapezoidal – Movimento Manual.

Movimento Automático: o movimento é realizado automaticamente entre as posições determinadas obedecendo ao intervalo especificado pelo *delay*. A Figura 10 apresenta a simulação do movimento realizado para quatro ações.

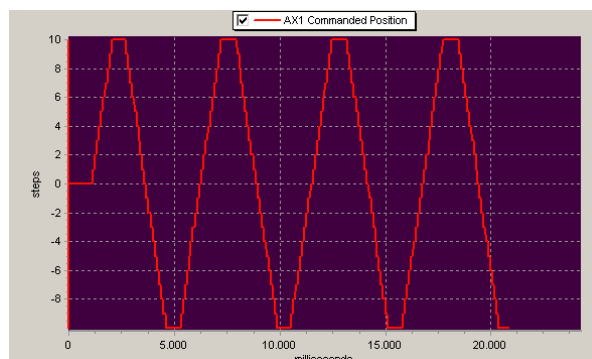


Figura 10: Gráfico Trapezoidal – Movimento Automático.

- *Velocidade Controlada*

Movimento Controlado pela Velocidade: o movimento é especificado apenas pela velocidade, positiva ou negativa, determinando o sentido do movimento. As mudanças do movimento se dão pelas trocas de velocidade, sendo necessário um *delay* para as alterações dos valores, na Figura 11 podemos observar as alterações da velocidade (sentido e valor) e o espaçamento entre as alterações (*delay*).

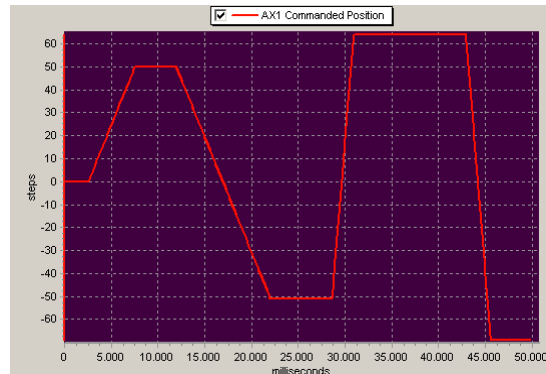


Figura 11: Gráfico de Controle por Velocidade – Movimento Manual.

- *Curva-S*

Movimento Automático: o movimento é realizado automaticamente entre as posições determinadas com o *delay* especificado e com a taxa máxima de variação de aceleração estipulada (*Jerk*) que pode ser observada na Figura 12 pelas variações criadas (degraus).

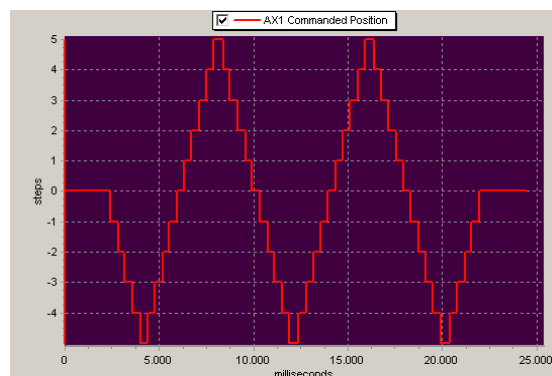


Figura 12: Gráfico Curva S – Movimento Automático.

b) Integração de cada eixo de controle separadamente e em conjunto, principalmente aos movimentos simultâneos entre dois e ou até quatro eixos.

Todos os experimentos foram configurados para o motor de passo, perfil Trapezoidal escolhido por se apresentar um perfil de fácil configuração e que permite uma boa manipulação dos eixos de controle integrados, e movimentação automática facilitando uma futura implementação do sistema de controle automático.

Os testes para integração dos eixos são representados na Tabela 3, e correspondem aos movimentos com cada eixo utilizado simultaneamente, respeitando sua respectiva relação determinada arbitrariamente.

Tabela 3: Motor de Passo – Pulso e Direção – Integração dos Eixos.

Eixo de Ação	1-2-3-4	1 – 2	3 – 4	
Teste	F	G	H	
Eixo	Todos	1 e 2	3	4
Aceleração (step/ms ²)	0.01	0.01	0.01	0.01
Velocidade (step/ms)	0.01	0.01	0.01	0.03
Posição1 (step)	100	50	15	15
Posição 2 (step)	0	-50	-15	-15
Delay (ms)	-	500	1000	500

- Eixos: 1, 2, 3 e 4: a execução simultânea de movimentos nos quatro eixos representa a situação extrema de interação e pode ser observada na Figura 13, sendo todos os eixos configurados com os mesmos parâmetros e o pequeno deslocamento observado no início de cada gráfico corresponde a inicialização individual de cada eixo.

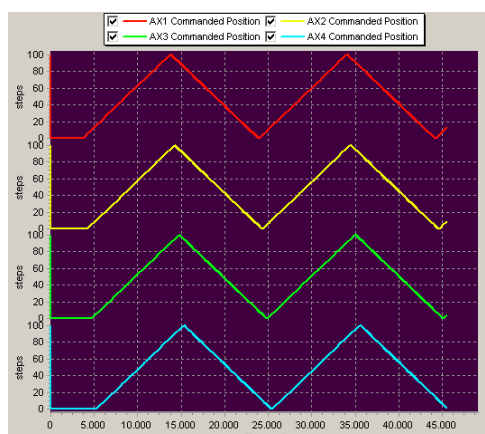


Figura 13: Gráfico Trapezoidal – 4 Eixos Simultâneos – Mov. Automático.

- Eixos: 1 e 2: uma interação mais simples é realizada na simulação de dois eixos de comando com parâmetros idênticos, representada na Figura 14.

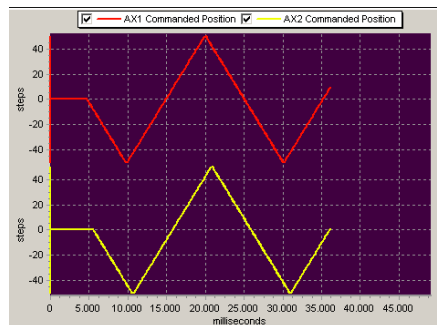


Figura 14: Gráfico Trapezoidal – 2 Eixos Simultâneos – Mov. Automático.

- Eixos: 3 e 4: a execução de movimentos simultâneos porém distintos em dois eixos de controle é observada na Figura 15.

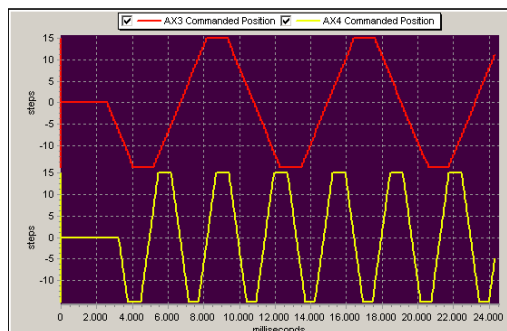


Figura 15: Gráfico Trapezoidal – 2 Eixos Simultâneos – Mov. Automático Distintos.

4.2. Algoritmo “Interface PMC”

Tendo como base o *software Pro-Motion* e os testes de parâmetro e funcionalidade realizados anteriormente, um algoritmo de configurações e movimentos básicos foi implementado (utilizando-se o *Microsoft Visual Studio.NET 2003* e a biblioteca de programação *C-Motion*) com interface simples e portando contendo somente as configurações essenciais para acionamento da placa, pré-ajuste de movimentos, leitura de sinais analógicos, armazenamento e monitoramento das mensagens e do estado do sistema.

A estrutura básica do algoritmo “Interface PMC” é composta pelo ajuste automático das configurações essenciais da placa (estabelecer a comunicação *PCI*; selecionar o tipo do motor conectado; ajustar a frequência e o tipo da saída desejada; controle do sistema de monitoramento e armazenamento de mensagens). E pelos ajustes manuais definidos pelo usuário (sentido do movimento; controle da posição; velocidade e aceleração do movimento; definição dos eixos em qual a ação deve ser executada; entrada analógica habilitada; armazenamento das mensagens). A Tabela 4 apresenta a estrutura do *software* “Interface PMC”.

Tabela 4: Características funcionais do *software* “Interface PMC”.

Ajustes Automáticos
Configuração da Interface PCI
Controle das Mensagens
Tipo do Motor
Frequência e Tipo da Saída
Ajustes Manuais
Sentido: Direita, Esquerda
Posição, Velocidade e Aceleração
Eixo de Trabalho: 1,2,3,4
Entrada Analógica Habilitada
Armazenamento das Mensagens
Inicialização do Sistema

O algoritmo é estruturado basicamente em três sistemas que interagem.

- **Interface Usuário:**

Possibilita o acesso ao programa, sendo o ponto de partida e meio de entrada de informações, este sistema oferece e monitora os ajustes realizados pelo usuário, apresenta as mensagens de erro e o estado do sistema em um monitor.

- **Bloco do Programa Central:**

Constitui o principal sistema do algoritmo, pois é responsável pelas configurações automáticas, o tratamento das mensagens, a leitura das entradas analógicas e a execução do movimento.

- **Biblioteca C-Motion:**

É o bloco externo que é utilizado por todos os sistemas do bloco central. A biblioteca *C-Motion*, representa aqui o conjunto de funções, variáveis e dependências que são utilizados pelo algoritmo do programa, todos os comandos acessíveis da placa de controle tem a sua função e variável de acesso, relacionados em um comando específico.

O algoritmo completo é ilustrado na Figura 16.

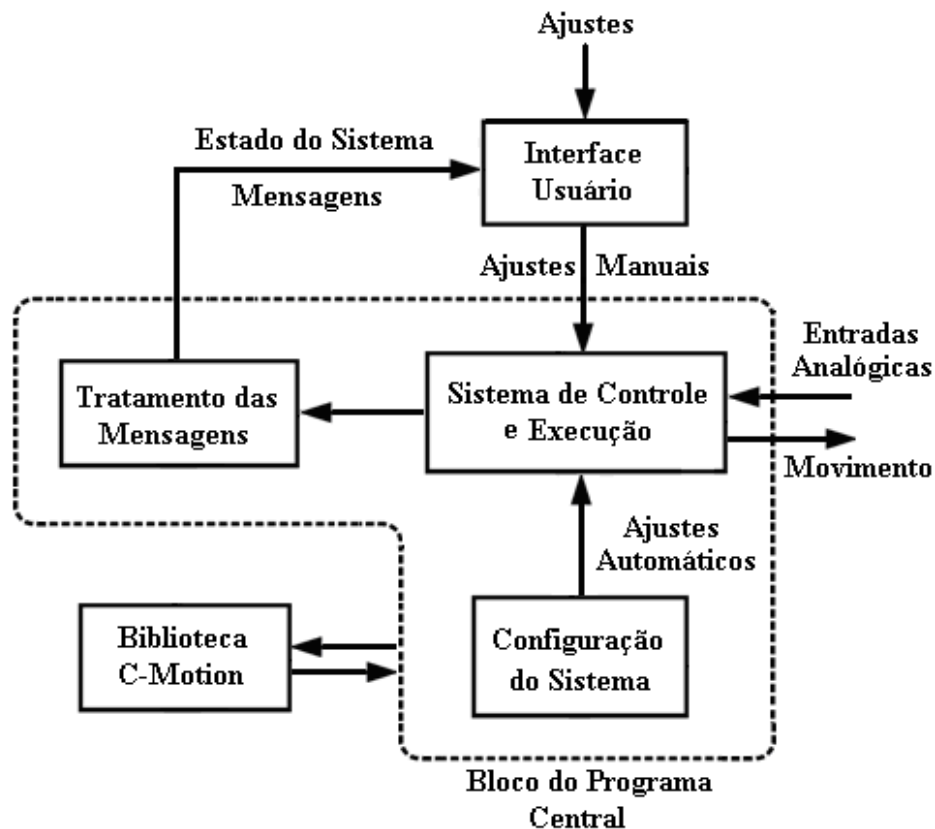


Figura 16: Diagrama de Blocos Algoritmo “Interface PMC”

O *Software* “Interface PMC” apresenta, como ilustrado na Figura 17, um visual simples e interativo. O *software* é inicializado com configurações pré-definidas (Posição=10; Velocidade=500; Aceleração=10; Eixo de comando = Eixo 1; Direção = Direita; Entrada Analógica = In a), estes ajustes têm o efeito de facilitar e simplificar a

utilização do programa em um primeiro uso. Enquanto o movimento está sendo executado uma mensagem dando a opção ao usuário de abortar o movimento é mostrada no monitor, que apresenta as mensagens e o estado do programa na parte esquerda da janela, oferecendo também a opção de limpar a tela.

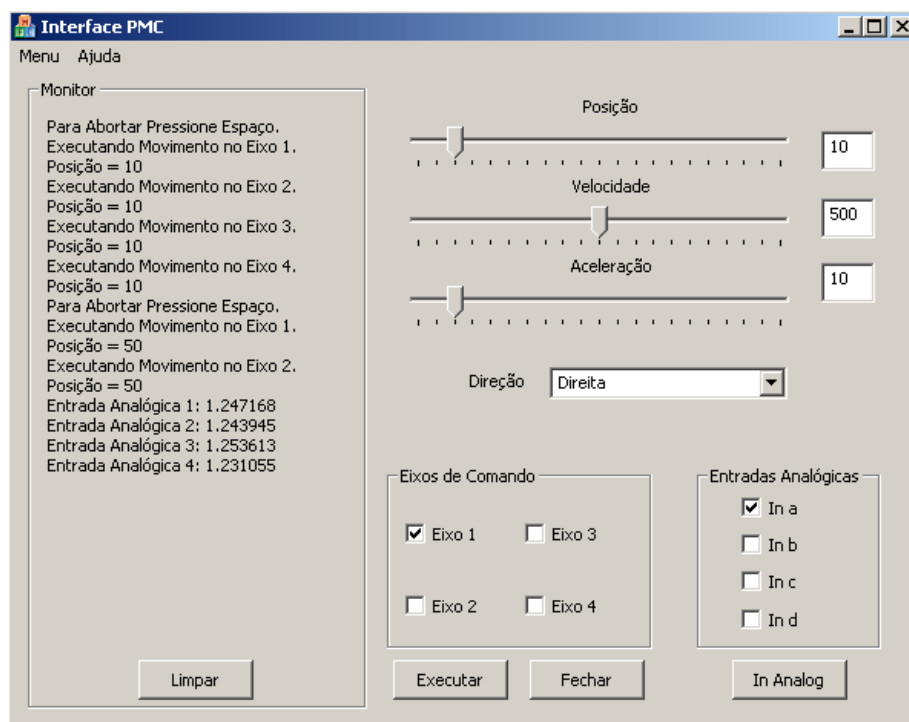


Figura 17: Software “Interface PMC”.

Os eixos de comando podem ser selecionados de forma isolada ou simultânea, com a seleção mínima de 1 e máxima de até 4 eixos; os ajustes paramétricos de posição, velocidade e aceleração são alterados pela barra de ajustes; a direção do movimento é alterada pela caixa de seleção; e as entradas analógicas são habilitadas de “a” até “d” e registradas no monitor pela opção “In Analog”.

O Menu ilustrado na Figura 18 apresenta as opções de salvar as mensagens do sistema, informações sobre o software, versão da placa de controle, e encerramento do programa.



Figura 18: Menu do software “Interface PMC”.

Quando requisitado, o programa salva as informações apresentadas no monitor em um arquivo texto, determinado pelo próprio usuário, um exemplo deste arquivo é ilustrado na Figura 19. O arquivo é estruturado contendo a hora da gravação e as mensagens apresentadas no monitor.

<pre>14:41:51 Executando Movimento no Eixo 1. Posição = 10 14:41:53 Executando Movimento no Eixo 2. Posição = 10 14:41:55 Executando Movimento no Eixo 3. Posição = 10 14:41:57 Executando Movimento no Eixo 4. Posição = 10 14:42:00 Executando Movimento no Eixo 1. Posição = -10 14:42:02 Executando Movimento no Eixo 2. Posição = -10 14:42:07 Executando Movimento no Eixo 3. Posição = -10 14:42:10 Executando Movimento no Eixo 4. Posição = -10 14:42:48 Executando Movimento no Eixo 1. Abortado! Posição = 5</pre>	<pre>14:49:54 Executando Movimento no Eixo 1. Posição = 50 Executando Movimento no Eixo 2. Posição = 50 Executando Movimento no Eixo 3. Posição = 50 Executando Movimento no Eixo 4. Posição = 50 14:49:59 Executando Movimento no Eixo 1. Posição = -50 Executando Movimento no Eixo 2. Posição = -50 Executando Movimento no Eixo 3. Posição = -50 Executando Movimento no Eixo 4. Posição = -50 14:50:02 Entrada Analógica 1: 1.250391 14:50:03 Entrada Analógica 2: 1.253613 14:50:05 Entrada Analógica 3: 1.237500 14:50:06 Entrada Analógica 4: 1.234277</pre>
---	---

Figura 19: Arquivo texto de saída – “Interface PMC”. **Esquerda:** Eixos isolados. **Direita:** Eixos simultâneos, entrada Analógica.

Na Figura 19 (Esquerda), temos os movimentos isolados dos eixos 1,2,3 e 4 para direita e esquerda, posição igual a 10, e movimento isolado do eixo 1 para direita, mas abortado pelo usuário na posição 5. Na Figura 19 (Direita), os movimentos simultâneos dos eixos 1,2,3 e 4 para direita e esquerda, posição igual a 50; e a leitura das Entrada Analógicas 1,2,3 e 4.

4.3. Algoritmo “PMCVox”

Este algoritmo é composto da integração do algoritmo de reconhecimento de voz e do algoritmo de configuração e comando de movimentos “Interface PMC”. Esta integração foi realizada com a união dos dois programas em uma única estrutura com uma interface simples e que proporcione uma integração total do sistema. Cada movimento é relacionado com um comando de voz e com seus eixos de ação respeitando o protocolo listado na Tabela 5.

Cada eixo de comando responde por um grau de liberdade e seu respectivo motor, os conjuntos escolhidos na Tabela 5 foram determinados arbitrariamente para executar interações variadas entre os eixos.

Tabela 5: Protocolo de Correlação (Comando de voz – Movimento – Eixo)

Comando de Voz	Protocolo	
	Movimento	Eixos
“Sobe”	Esquerda	1 e 2
“Desce”	Direita	1 e 2
“Esquerda”	Esquerda	3 e 4
“Direita”	Direita	3 e 4
“Avança”	Esquerda	1 e 3
“Recua”	Direita	2 e 4
“Para”	Para	Todos

O algoritmo de reconhecimento de voz foi alterado e ajustado para a adaptação com o novo programa. O bloco de gerenciamento de tarefas foi integrado para ativar os movimentos determinados no protocolo e o bloco de comunicação apresenta uma interação entre o sistema de reconhecimento de voz e a placa de controle centralizada no mesmo contexto.

O *software* “Interface PMC” foi adaptado para interagir aos comandos enviados pelo sistema de reconhecimento de voz, realizando os movimentos pré-ajustados em seus respectivos eixos, respeitando o protocolo estabelecido, nesta adaptação apenas o ajuste da velocidade de movimentação é possível, os demais ajustes ficam agora a cargo do sistema de configuração automático.

O algoritmo é estruturado basicamente em seis sistemas que interagem.

- **Interface Usuário – “PMCVox”:**

Possibilita o acesso ao programa pelo ajuste de parâmetros da placa e do sistema de reconhecimento de voz, e principalmente responde aos comandos de voz oferecendo uma interatividade através de mensagens e da apresentação do estado do sistema no monitor.

- **“InVox” Reconhecimento de Voz:**

Este sistema realiza o mesmo processo de reconhecimento de voz através da máquina de estados, porém adaptado no bloco de comunicação para interagir diretamente com o sistema de comando da placa.

- **“Interface PMC” Bloco Central:**

Responsável pelas configurações automáticas, leitura das entradas analógicas e a execução do movimento, também adaptado para executar os movimentos através da palavra de comando identificada pelo sistema de reconhecimento de voz.

- **Biblioteca C-Motion:**

Bloco externo, de acesso às funções e variáveis do sistema de controle, que é utilizada pelos sistemas do bloco central.

- **SMAPI:**

É o bloco externo, responsável pelo processo de reconhecimento de voz.

O algoritmo completo é ilustrado na Figura 20.

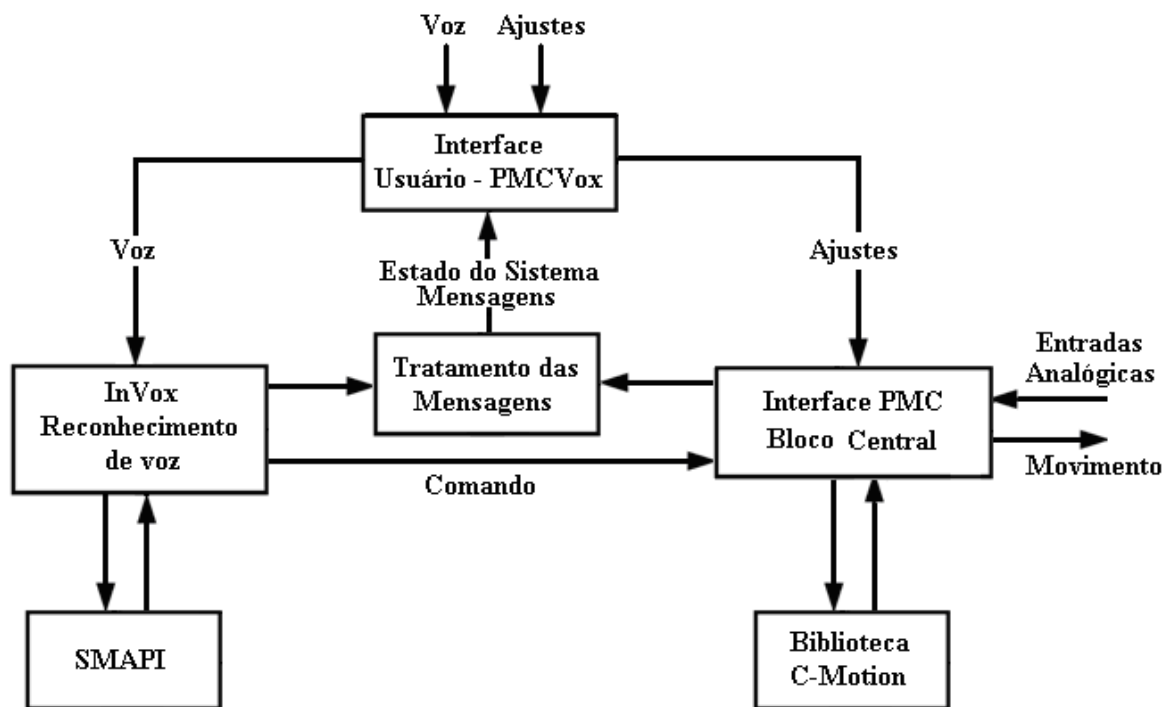


Figura 20: Diagrama de Blocos Algoritmo “Interface PMC”

O Software “PMCVox” apresenta, como ilustrado na Figura 21, um visual simples. O software é inicializado com configurações pré-definidas (Posição; Velocidade; Aceleração), estes ajustes são definidos pela leitura de um arquivo

“Setup.pmd” que contem as ultimas configurações utilizadas pelo programa. O display superior apresenta ao usuário o estado do programa ou a palavra reconhecida.

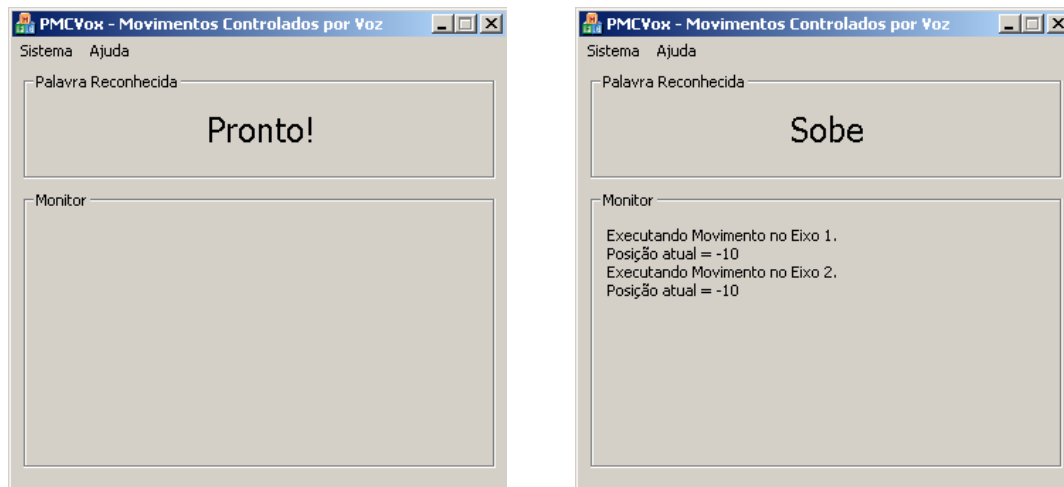


Figura 21: Software “PMCVox”. **Esquerda:** Mensagem de Estado. **Direita:** Palavra reconhecida

Na Figura 21 (Esquerda) é ilustrado o início do programa e a mensagem de que o programa está apto a ser utilizado, na Figura 21 (Direita) temos a palavra reconhecida e na caixa de mensagens do monitor o movimento realizado.

O Menu ilustrado na Figura 22 apresenta as opções de ajustes do sistema de reconhecimento de voz e do sistema de controle, registro das mensagens do sistema e das palavras reconhecidas, encerramento do programa, informações sobre o software e versão da placa de controle.

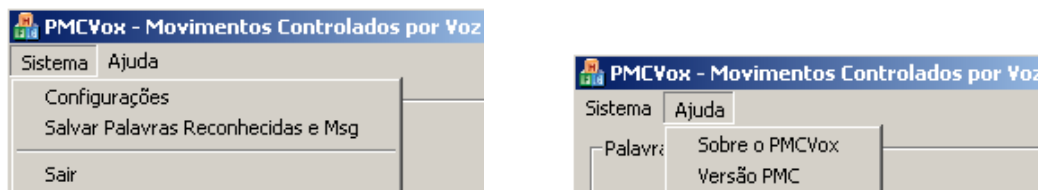


Figura 22: Menu do software “PMCVox”.

Quando requisitado, o programa salva as informações apresentadas no monitor em um arquivo texto, determinado pelo próprio usuário, um exemplo deste arquivo é ilustrado na Figura 23. O arquivo é estruturado contendo a hora da gravação e as mensagens apresentadas no monitor, com a palavra reconhecida e os movimentos executados.

13:17:14	Sobe	13:29:12	Dorme
Executando Movimento no Eixo 1.		13:29:20	Acorda
Posição atual = -10		13:29:25	Sobe
Executando Movimento no Eixo 2.		Executando Movimento no Eixo 1.	
Posição atual = -10		Posição atual = -10	
13:17:17	Para(T)	Executando Movimento no Eixo 2.	
13:17:19	Desce	Posição atual = -10	
Executando Movimento no Eixo 1.		13:29:26	Para
Posição atual = 10		13:29:34	Desce
Executando Movimento no Eixo 2.		Executando Movimento no Eixo 1.	
Posição atual = 10		Posição atual = 10	
13:17:22	Para(T)	Executando Movimento no Eixo 2.	
13:17:25	Esquerda	Posição atual = 10	
Executando Movimento no Eixo 3.		13:29:34	Para
Posição atual = -10		13:29:38	Avança
Executando Movimento no Eixo 4.		Executando Movimento no Eixo 1.	
Posição atual = -10		Posição atual = -10	
13:17:28	Para(T)	Executando Movimento no Eixo 3.	
13:17:30	Direita	Posição atual = -10	
Executando Movimento no Eixo 3.		13:29:40	Para
Posição atual = 10		13:29:49	Direita
Executando Movimento no Eixo 4.		Executando Movimento no Eixo 3.	
Posição atual = 10		Posição atual = 10	
13:17:33	Para(T)	13:29:49	Para
13:17:36	Avança	13:30:19	Dorme(T)
Executando Movimento no Eixo 1.		13:30:30	Acorda
Posição atual = -10			
Executando Movimento no Eixo 3.			
Posição atual = -10			
13:17:39	Para(T)		
13:17:42	Recua		
Executando Movimento no Eixo 2.			
Posição atual = 10			
Executando Movimento no Eixo 4.			
Posição atual = 10			
13:17:45	Para(T)		

Figura 23: Arquivo texto de saída - “PMCVox”. **Esquerda:** Comandos de movimento. **Direita:** Comandos de parada e espera do sistema.

Na Figura 23 (Esquerda), temos os comandos registrados para todas as palavras de comando que implicam em um movimento definido, sendo que ao final de cada movimento o sistema para automaticamente. Na Figura 23 (Direita), os comandos para o sistema dormir e acordar, alguns comandos de movimento e por fim o sistema entrando em estado de espera automaticamente após trinta segundos sem palavras pronunciadas.

Os ajustes do sistema de reconhecimento de voz e do sistema de controle são realizados por uma nova janela, que oferece ao usuário as opções de alterar o limiar de rejeição em uma escala de 0 a 10; a velocidade de reconhecimento com quatro

níveis de magnitude, sendo um automático; e a velocidade de movimento do robô com três valores possíveis, como ilustrado na Figura 24.

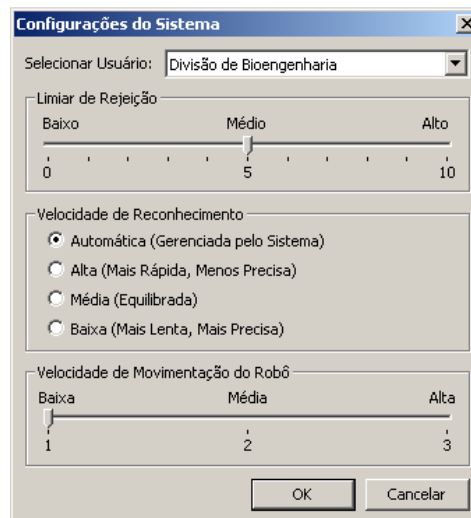


Figura 24: Interface de configuração do *Software* “Interface PMC”.

5. Discussão e Conclusão

A utilização de comandos de voz para o acionamento de manipuladores para o posicionamento de endoscópio é uma opção muito utilizada. Os comandos de voz são simples, consideravelmente intuitivos, possuem tempos de aprendizado e adaptação curtos e permitem ao cirurgião que permaneça constantemente com as mãos livres.

O trabalho utilizou o reconhecimento de comandos de voz no controle de movimentos básicos através da placa de controle Prodigy Motion Card, aplicada à robótica cirúrgica.

O primeiro algoritmo implementado, “Interface PMC”, oferece o acesso direto as funções essenciais da placa de controle *PMC*, permitindo a configuração, execução e monitoração de ações, realizando movimentos simples em até quatro eixos de comando.

O algoritmo “PMCVox”, foco central do trabalho, permitiu um sistema válido na integração das principais funções do algoritmo de reconhecimento de voz e de controle da placa de comando, realizando as ações nos eixos configurados através de comandos de voz estabelecidos nos vocabulários hierárquicos, além de apresentar ao usuário um monitor para acompanhamento do sistema.

As simulações realizadas comprovam o funcionamento computacional dos *softwares* e servem como uma base estrutural para a implementação do sistema. Novas simulações seguindo normas rigorosas de segurança na área computacional, robótica e médica serão realizadas em testes experimentais com o módulo de potência simulando o controle dos movimentos dos motores.

Na próxima fase do projeto um algoritmo complementar de controle robótico será avaliado e selecionado para o desenvolvimento de um protótipo robótico.

6. Referências Bibliográficas

Cohn, M.B., et al, "Surgical application of mili-robots", Journal of Robotic Systems, 12(6):401-406, 1995.

D'atellis, N., et al, "Robotic-assisted cardiac surgery: Anesthetic and postoperative considerations", Journal of Cardiothoracic and Vascular Anesthesia, 16(4):397-400, 2002.

Falk, V., et al, "Quality of computer enhanced totally endoscopic coronary bypass graft anastomosis – comparison to conventional technique", European Journal of Cardio-thoracic Surgery, 15:260-265, 1999.

Georgesson, K.E., Owings, E., "Advances in minimally invasive surgery in children", American Journal of Surgery, 180:362-364, 2000.

KAPPERT, U. et al. "Technique of closed chest coronary artery surgery on the beating heart". European Journal of Cardio-thoracic Surgery, v. 20, p. 765–769, 2001.

Kolvenbach, R., et al, "Total laparoscopically and robotically assisted aortic aneurysm surgery: A critical evaluation", Journal of Vascular Surgery, 39(4):771-776, 2004.

Morgan, J.A., et al, "Robotic techniques improve quality of life in patients undergoing atrial septal defect repair", Annals of Thoracic Surgery, 77(4):1328-1333, 2004.

PMD - Performance Motion Devices, endereço eletrônico: <http://www.pmdcorp.com> acessado em 23/10/2007.

Rosen, J., et al, "Force controlled and teleoperated endoscopic grasper for minimally invasive surgery: experimental performance evaluation", IEEE Transactions on Biomedical Engineering, 46(10):1212-1221, 1999.

Ruurda, J.P., et al, "Robot-assisted versus Standard Videoscopic Aortic Replacement. A Comparative Study in Pigs", European Journal of Vascular and Endovascular Surgery, 27(5):501-506, 2004.

SABBATINI, R. M. E. Os robôs chegam à sala cirúrgica. Revista Check-up, v. 2, n. 12, Setembro 1999.

SANTOS, E. T. P. Sistema de Reconhecimento de Comandos Isolados de Voz para Aplicação em Robótica Cirúrgica. Dissertação (Mestrado) — Escola Politécnica da Universidade de São Paulo, 2003

SANTOS E. T. P., CESTARI I. A., Interface de comando e Estratégia de Acionamento de um Braço Manipulador para Posicionamento de Endoscópio. In: XIX Congresso Brasileiro de Engenharia Biomédica, 2004, João Pessoa/Paraíba/Brasil. XIX Congresso Brasileiro de Engenharia Biomédica, 2004. v. 5. p. 781-784.

Slade, A.P., et al, “An integrated control structure for surgical assist robotics for laparoscopy”, Anais do XIV Congresso Brasileiro de Engenharia Mecânica – COBEM’97, São Paulo, Brasil, [s.n.], 1997.

Stassen, H.G., et al, “Man-machine aspects of minimally invasive surgery”, Annual Reviews in Control, 25:111-122, 2001.

Troccaz, J., Peshkin, M., Davies, B., “Guiding systems for computer-assisted surgery: introducing synergistic devices and discussing the different approaches”, Medical Image Analysis, 2(2):101-119, 1998.