



Universidade do Minho
Escola de Engenharia

Diogo Alexandre Pires Lourenço

Controlo Automático dos Parâmetros
Cinemáticos de uma guia/mesa linear
atuada por servomotor



Universidade do Minho
Escola de Engenharia

Diogo Alexandre Pires Lourenço

Controlo Automático dos Parâmetros
Cinemáticos de uma guia/mesa linear
atuada por servomotor

Tese de Mestrado
Ciclo de Estudos Integrados Conducentes ao
Grau de Mestre em Engenharia Mecânica

Trabalho efetuado sob a orientação do
Professor Doutor Eurico Seabra

e co-orientação do:
Doutor Hélder Puga

DECLARAÇÃO

Diogo Alexandre Pires Lourenço

Endereço eletrónico: diogoplourenco@gmail.com Telefone: 926 436 219

Número do Bilhete de Identidade: 11940621

Título dissertação /tese

Controlo automático dos parâmetros cinemáticos de uma mesa/guia linear actuada por servomotor.

Orientador:

Professor Doutor Eurico Seabra

Universidade do Minho

Co-Orientador:

Professor Doutor Hélder Fernandes Puga

Universidade do Minho

Ano de conclusão:

Designação do Mestrado:

Ciclo de Estudos Integrados Conducentes ao Grau de Mestre em Engenharia Mecânica

DE ACORDO COM A LEGISLAÇÃO EM VIGOR, NÃO É PERMITIDA A REPRODUÇÃO DE QUALQUER PARTE DESTA TESE/TRABALHO

Universidade do Minho, ___/___/_____

Assinatura: _____

Agradecimentos

Quero manifestar o meu agradecimento ao Prof. Joaquim Barbosa, pelo incentivo que me deu quando ingressei na Universidade do Minho. Gostaria de manifestar o meu sincero agradecimento aos colegas Carolina, João Paulo e Domingos, pelo seu companheirismo e amizade. O meu agradecimento ao Prof. Eurico Seabra pelo seu entusiasmo e disponibilidade, e também ao Prof. Hélder Puga por todo o seu apoio. Agradeço muito especialmente à minha família por tudo.

Resumo

O presente trabalho reporta o estudo, implementação e teste de um sistema mecatrónico desenvolvido para o controlo de uma mesa/guia linear de um único eixo XX atuada por servomotor. Nele são continuamente abordadas as componentes de *software*, de electrónica e de hardware. Utilizando-se o *software* de programação *Labview* desenvolveram-se rotinas para o controlo semiautomático e automático da mesa, que através do seu painel frontal podem ser ativadas pelo operador. A interatividade do *software* e do seu processador é conseguida através das entradas e saídas digitais da placa de aquisição de dados utilizada, a *NI DAQ PCI-1200*.

O acionamento da mesa / guia linear é conseguido através do conjunto servomotor *R88M-UE40030V-BS1* e do seu controlador específico, do fabricante *OMRON*, estes recebem as instruções conectando-se directamente à *DAQ* através do seu cabo de programação e por intermédio do conector.

A sua programação baseia-se assim num *software* muito configurável, possuindo o próprio um Painel Frontal de interação intuitiva e de fácil utilização.

Com este sistema de programação e controlo automático de uma mesa/guia linear, é possível a automatização do processo de uma máquina ferramenta válida, a definir em futuros trabalhos, procurando dar resposta às necessidades de uma atividade industrial.

Abstract

This paper reports the study, implementation and testing of a mechatronic system developed for the control of a single axis XX linear guide actuated by servomotor. In this document are continually attended to the software components, electronics and hardware components. Using the *Labview* programming software we could develop routines for semi-automatic and automatic control of the linear guide (screw ball motion). Through the *Labview* Front Panel, the system can be controlled by the worker. The interactivity of the *Labview* software and others mechatronic components has been achieved through the Inputs and Outputs of the data acquisition board used the NI PCI 1200, DAQ.

Motion the linear guide is possible by the set servomotor R88M - UE40030V - BS1 and their specific servo driver, OMRON; they are instructed by connecting directly to the DAQ through its control cable and through the connector. The system is thus based on very configurable software and their Front Panel with intuitive interaction, and easy to be used.

With this control system, and their automatic control of a linear guide, it'll be possible to automate the process of a machine tool, to be defined in a future study, seeking to respond to the needs of an industrial activity.

Índice

Agradecimentos	i
Resumo	iii
Abstract	v
Índice	vii
Lista de figuras	ix
Lista de tabelas	xi
Capítulo 1 – Introdução.....	13
1.1 – Motivação do trabalho.....	16
1.2 – Objetivo do trabalho.....	16
1.3 – Estrutura do Relatório	17
Capítulo 2 – Motores Eléctricos.....	19
2.1 – Generalidades	19
2.2 – Servomotores.....	21
2.3 – Parâmetros de controlo.....	27
Capítulo 3 – Sistema Mecatrónico	29
3.1 – Definição do projecto de mecatrónica.....	29
3.2 – Descrição do sistema mecatrónico adotado	30
Capítulo 4 – Sistema Eletrónico e Mecânico	33
4.1 – Servomotor utilizado	33
4.2 – Placa de Aquisição utilizada.....	34
4.3 – Circuitos eletrónicos utilizados	36
4.4 – Seleção, Montagem e Condicionamento de Sinal dos Transdutores.....	41
4.5 – Sistema Mecânico	42
Capítulo 5 – Programação no Labview	47
5.1 – Programa desenvolvido	48
5.2 – Controlo Semiautomático e Automático do Sistema	53

5.3. Teste das Entradas e Saídas do Sistema	61
Capítulo 6 – Implementação Física e Análise dos Resultados	63
6.1 – Descrição do protótipo.....	63
6.2 – Ensaio de validação.....	65
6.3 – Análise dos resultados	70
Capítulo 7 – Conclusões e trabalhos futuros	77
Referências bibliográficas	81
Anexo I. Extrato do Instruction Manual, Computer Monitor Software (Version 1.1), OMNUC U, Chapter 5, Operation.	83
Anexo II. Ficha técnica da Placa de Aquisição de dados utilizada. PCI 1200, <i>National Instruments</i>	85
Anexo III. Folha de cálculo do controlo dos parâmetros cinemáticos da Mesa / Guia. Electronic Gear = 4.	89
Anexo IV. Folha de cálculo do controlo dos parâmetros cinemáticos da Mesa / Guia. Electronic Gear = 40.	91

Lista de figuras

Figura 1.1: Fresadora CNC com mesa, “pronum cnc machine”	15
Figura 2.1: Síntese dos vários tipos de motores elétricos utilizados na indústria. Adaptado de [2].	20
Figura 2.2: Motor AC <i>Brushless</i> , ABB.	21
Figura 2.3: Figura explicativa do funcionamento de um Encoder ótico digital.....	22
Figura 2.4: Sistema de controlo do servomotor (PWM sinusoidal driving function). Adaptado de [7].	23
Figura 2.5: Curva característica do Torque vs. Rotação do Servomotor utilizado. Adaptado de [9].	24
Figura 2.6: Servo motor utilizado R88M-UE40030V-BS1, OMRON. Transmissão mecânica.....	25
Figura 2.7: Representação do ganho electrónico do controlador utilizado. Adaptado de [9].....	26
Figura 3.1: Organização dos constituintes de um projecto de mecatrónica. Proposto por [5].	29
Figura 3.2: Sistema mecatrónico adotado no controlo automático da mesa / guia linear.	30
Figura 3.3: Aspeto do sistema mecatrónico desenvolvido, no Laboratório de Fundição do DEM.	31
Figura 3.4: Sistema Mecatrónico: Formalismo de Programação. Etapas e Tarefas.....	31
Figura 4.1: Dimensões exteriores do Servomotor. Adaptado de [9].	33
Figura 4.2: Dimensões exteriores do Controlador. Adaptado de [9].	34
Figura 4.3: Aspeto geral da DAQ instalada na torre da CPU	35
Figura 4.4: Placa electrónica utilizada, “ <i>bread board</i> ”, existente no DEM.	36
Figura 4.5: Representação do circuito electrónico elementar utilizado.....	37
Figura 4.6: Comprimento mínimo dos pulsos de comando do controlador. Adaptado de [9]	38
Figura 4.7: Cabo de potência do servomotor	38
Figura 4.8: Alimentação eléctrica exterior aos pinos 14 e 18. Adaptado de [9].....	39
Figura 4.9: Representação das Entrada/Saídas do terminal CN1 do Controlador. Adaptado de [9].	40
Figura 4.10: Representação dos fins de curso utilizados.	41
Figura 4.11: Esquema Eletrónico do fim de curso utilizado.....	42
Figura 4.12: Aspeto do perfil longitudinal e transversal da mesa/guia linear. Adaptado de [21]	43
Figura 4.13: Representação tipo do fuso de esferas e porca. Adaptado de [20].....	43
Figura 4.14: Representação da mesa / guia linear utilizada. Visualização do Solidworks.	44
Figura 4.15: Vista de frente, flange de acoplamento do servomotor à mesa/guia. Visualização do <i>AutoCad</i>	44
Figura 4.16: Fotografia da união elástica com disco utilizada.	45
Figura 5.1: Aspeto do Painel Frontal, na CPU utilizado.....	48
Figura 5.2: Visualização da Seleção do modo de operação (“Ring”).	49
Figura 5.3: Sub-rotina elementar de saída de pulsos (“ <i>While Loop</i> ”), no modo manual. Diagrama de Blocos do Labview.	51
Figura 5.4: Sistema de segurança implementado em todas as sub-rotinas e rotinas.	52
Figura 5.5: Visualização da seleção da velocidade e aceleração, comum a todos os modos de operação.	53
Figura 5.6: Sub-rotina no Labview para o modo de operação Manual. Diagrama de Blocos do Labview. 54	

Figura 5.7: Visualização dos comandos no modo de operação Automático. Painel Frontal do Labview. .	54
Figura 5.8: Sub-rotina “ <i>tempo de espera</i> ”. Diagrama de blocos do Labview.....	55
Figura 5.9: Sub-rotina “ <i>estágio no patamar recuado</i> ”. Diagrama de blocos do Labview.	56
Figura 5.10: Sub-rotina “ <i>Foward Pulse</i> ”. Diagrama de blocos do Labview.	57
Figura 5.11: Sub-rotina “ <i>Operação concluída</i> ”. Diagrama de blocos do Labview.....	58
Figura 5.12: Visualização das “ <i>Sequence Structures</i> ” nos modos Automático e Semiautomático.	58
Figura 5.13: Controlo do posicionamento e da velocidade instantânea do sistema.	60
Figura 5.14: Cálculo do <i>Electronic Gear</i> e do Número de rotinas necessárias por rotação.	60
Figura 5.15: Conector utilizado. Ligação das entradas e saídas da DAQ ao cabo de comando CN1.....	61
Figura 5.16: Teste da entrada 2 do porto B, com a injeção de 5 V no pino correspondente.	62
Figura 6.1: Aspeto do sistema mecatrónico desenvolvido. Indicação dos seus principais componentes.	64
Figura 6.2: Interface para o controlo dos User Parameters <i>Cn-24</i> e <i>Cn-25</i> . Visualização do <i>UMNUC</i>	65
Figura 6.3: CN3, ligação do cabo de série RS232 para comando dos “ <i>User Parameters</i> ” da Servo Drive.	65
Figura 6.4: Dinâmica de melhoria do projecto de acordo com os princípios da mecatrónica.....	69
Figura 6.5: Electronic Gear = 40. Gráfico da relação das velocidades com a duração das rotinas.	70
Figura 6.6: Electronic Gear = 4. Gráfico da relação das velocidades com a duração das rotinas.	72
Figura 6.7: Electronic Gear = 4 e 40. Gráfico da relação da distância percorrida com o número de rotinas.	73
Figura 6.8: Electronic Gear = 4 e 40, Aceleração = $1 \text{ mm}^2/\text{s}$. Gráfico que relaciona a velocidade média e instantânea com a distância percorridas.....	74
Figura 6.1: Electronic Gear = 40. Relação da velocidade média com a aceleração (distância = 200 mm , velocidade inicial = 1 mm/s).	75

Lista de tabelas

Tabela – 4.1 – Resumo das características das interfaces I/O da DAQ.....	35
--	----

Capítulo 1 – Introdução

A automação industrial veio facilitar a programação e implementação do conteúdo programado nas tarefas dos processos de fabrico, diminuir os erros do operador em cada acto e na prossecução da repetição das operações, simultaneamente permite também reduzir custos de produção, aumentando a flexibilidade da linha de fabrico e aumentando a rapidez de execução [1, 2].

Atualmente existe uma grande exigência de qualidade ao mais baixo custo. Simultaneamente os produtos são cada vez mais diferenciados, produtos muito diversos e tecnicamente evoluídos, com elevada densidade de componentes, o que exige um grande esforço na organização e planeamento dos sistemas de produção e, na modernização e otimização das técnicas e processos de fabrico [1, 2].

Para além disso os tempos de vida dos produtos são curtos, passado pouco tempo os produtos comercializados são descontinuados, o que exige uma contínua evolução e adequação dos processos de fabrico. Estas exigências não são compatíveis com linhas de produção rígidas, direccionadas para a produção em grande massa e com grande inércia para adaptação a novos produtos. Ao invés, exigem-se sim sistemas produtivos flexíveis, capazes de responder rapidamente a alterações ou introdução de novos produtos, de uma nova encomenda, de pequena ou média escala, permitindo assim cumprir os prazos e compromissos assumidos com o cliente. Nestas circunstâncias é imprescindível que mantenham igualmente a elevada e constante qualidade [1, 2].

Para tal, nas modernas instalações produtivas os sistemas de produção são organizados em células individualizadas e flexíveis de produção, de um ou mais processos de fabrico, que incorporam equipamentos tecnologicamente robustos para a execução das tarefas, como por exemplo, robôs manipuladores, robôs transportadores, máquinas ferramenta de controlo numérico computadorizado (*CNC*), sensores e autómatos (*PLC*) programáveis, assim como todas as redes de dados e os circuitos de comando e potência necessários. A flexibilidade destes equipamentos advém do facto de serem altamente programáveis, em que a sua função é em grande parte definida pelo *software* e conteúdo de programação memorizado, minimizando-se a interferência do operador [1, 2].

A uma escala superior, aliando uma correcta e prévia identificação, sequencialização e programação dos processos de fabrico de cada produto e, utilizando um *software* de programação normalizado e transversal ao máximo de suportes de hardware em uso, é possível tornar os sistemas de produção instalados mais versáteis e inteligentes [1, 2].

Genericamente, os equipamentos necessários à automatização de um processo de fabrico, são os processadores e unidades de memória, tais como os computadores e periféricos (computadores pessoais, controladores lógicos programáveis), os equipamentos e ferramentas e produção (robôs, máquinas ferramenta, máquinas *CNC*, transportadores automatizados, sensores e diversos atuadores.), as redes de dados e os *softwares CAD, CAE e CAM* adequado a cada fase do projecto [1, 2]. Atualmente, a metodologia de uso integrado do conjunto destes *softwares* designa-se por *CIM* (“*computer integrated manufacturing*”).

O *CAD* (“*computer aided design*”) é uma ferramenta informática que permite o desenho de projectos em computador, permitindo a rotação e obtenção de várias vistas das peças, e a visualização tridimensional dos mesmos, já desde há algum tempo, existem *softwares* deste tipo no mercado, tais como, o *AutoCAD*, o *Solidworks* ou o *Catia* [1].

O *CAE* (“*computer aided engineering*”) permite, calcular, simular e verificar as características mecânicas dos produtos tridimensionais *3D*, inicialmente concebidos através de *CAD* sem ter que os construir fisicamente [1].

Os *softwares* de *CAM* (“*computer aided manufacturing*”) são ferramentas que fazem a leitura automática da representação geométrica dos produtos desenvolvidos no *CAD* e *CAE*, calculam o caminho da ferramenta e geram uma linguagem que os equipamentos de comando numérico computadorizado (*CNC*) entendem para o fabrico das peças, trata-se da interface *CAD/CAM* [1].

Este tipo de programação *CNC*, auxiliada e automática, é depois transmitida às máquinas *CNC* através dos *CPU*, redes de dados ou mais localmente por intermédio dos *PLC* (controlos lógicos programáveis).

No caso da programação manual directa na máquina ou manual auxiliada por computador, isto é, no caso da leitura geométrica das peças não ser automática, como descrito anteriormente, o programador escreve ele próprio o programa fonte onde define a geometria da peça e/ou os movimentos da ferramenta [1]. A programação directa na máquina e a programação manual têm, como tal, o inconveniente de não

serem tão produtivas, pois o programador despende muito tempo no cálculo das trajetórias da ferramenta e na digitação dos dados geométricos [1]. No caso da programação directa em máquina, este ponto é ainda mais gravoso, pelo facto da máquina ter que permanecer parada durante a programação [1]. Como se constará no decorrer do relatório, no âmbito do presente trabalho pretendeu-se implementar um sistema de controlo de um equipamento, cujos movimentos são programados manualmente com o auxílio de um *CPU*, recorrendo à interface do *software* de programação *Labview*.

Os comandos de um programa *CNC* são os responsáveis pelo acionamento de uma máquina *CNC*. Dependendo dos dados da peça a ser maquinada, as linhas de comando de um programa *CNC* devem conter todas as indicações sobre o percurso, avanço e tipo de movimento da ferramenta, indicações sobre a velocidade e outras indicações que fazem acionar funções auxiliares, como por exemplo ligar eixo dum torno [1].

Outra função e vantagem dos *softwares CAM* é a simulação do processo de fabrico das peças, onde se pode visualizar todas as etapas de fabrico e maquinação de uma peça. Nomeadamente a definição da geometria e dimensões do bloco a maquinar e a escolha das melhores ferramentas e sequências de maquinação (desbaste, semi-acabamento, acabamento e redução de raios) [1].

Geralmente as máquinas *CNC* possuem um sistema de posicionamento de dois eixos, X e Y, e uma coluna que se movimenta ao longo do eixo Z. As máquinas *CNC* têm frequentemente acopladas ferramentas robustas permitindo-lhes executar tarefas com muita precisão e repetibilidade [1].



Figura 1.1: Fresadora CNC com mesa, "prorum cnc machine"

São exemplos de máquinas CNC os tornos, equipados com comando numérico, que podem realizar todas as operações de torneamento (tornear, facear, fazer canais, rosca, contornos, desbaste, furação) com recurso a programação. Existem também as máquinas ferramenta do tipo fresadora equipadas com comando numérico, que executam todas as operações possíveis de fresagem [1].

Adicionalmente, os programas de *CNC* criados manualmente ou automaticamente num programa *CAM*, têm ainda que passar pelo próprio controlador da máquina *CNC* (o controlador do atuador). É este controlador que faz a transformação do comando de programação em comandos de potência, a linguagem máquina [1].

Existem várias razões que impulsionaram o aparecimento das máquinas de comando numérico computadorizado, nomeadamente a criação de produtos de geometria complexa e alta com precisão e conseguir um menor tempo entre o projecto do produto e a fabricação do mesmo [1].

1.1 – Motivação do trabalho

Com a realização deste trabalho pretendeu-se criar um sistema de controlo automático, utilizado na automação da mesa/guia linear XX existente no DEM, atuada por um servomotor, tendo por base o *software* de programação *Labview* e uma placa de aquisição de dados, visando a automatização de um processo de fabrico industrial, por exemplo de uma máquina ferramenta, relativamente pouco complexa, e económica.

1.2 – Objetivo do trabalho

O objetivo deste trabalho é o desenvolvimento do controlo automático dos parâmetros cinemáticos de uma guia/mesa linear atuada por servomotor visando a sua implementação num sistema de automação de uma máquina ferramenta.

Desta forma pretende-se projetar, desenvolver e implementar um sistema de controlo de posição e velocidade de uma guia/mesa linear atuada por servomotor que permita o controlo automático dos parâmetros cinemáticos do seu patim, nomeadamente:

- O percurso de trabalho percorrido;

- O posicionamento para inicialização;
- A velocidade;
- A aceleração;
- Os tempos de espera no patamar de trabalho;
- Tempo de espera no patamar de início.

Pretendeu-se com este trabalho fazer a programação da operação da mesa/guia linear de forma manual auxiliada por computador, utilizando o *software Labview* e implementar o correspondente sistema mecatrónico. Absorvendo as correspondentes particularidades e dificuldades, em contraponto com o que seria a programação CNC.

1.3 – Estrutura do Relatório

Este trabalho está organizado numa primeira parte, capítulo 1, no qual se aborda a importância da automação industrial e das metodologias adoptadas nos processos produtivos modernos. No capítulo 2 é feita uma abordagem teórica aos motores eléctricos e em particular dos servomotores, utilizando como base o servomotor utilizado neste trabalho e, indicados os parâmetros do seu movimento que se pretende controlar. No capítulo 3 far-se-á uma abordagem teórica do projecto de mecatrónica e a apresentação do projecto mecatrónico implementado. A explanação do sistema mecânico e electrónico utilizados, incluindo os seus componentes, equipamentos e circuitos, será feita no capítulo 4. A temática da programação no Labview, incluindo as rotinas de programação para controlo semiautomático e automático, será abordada no capítulo 5 e, finalmente, no capítulo 6 é feita uma apresentação global do protótipo materializado, dos ensaios de validação realizados e é feita a análise dos resultados obtidos. No capítulo 7 são apresentadas as conclusões que se podem tirar do trabalho realizado assim como, são apresentadas sugestões para futuros trabalhos sobre este tema.

Capítulo 2 – Motores Elétricos

2.1 – Generalidades

A grande maioria dos movimentos que são realizados pelos equipamentos na indústria é acionada por meio de motores elétricos. Um motor elétrico é um atuador alimentado por corrente elétrica, que produz um movimento giratório no seu eixo que, através de acopladores mecânicos adequados se consegue aproveitar para efectuar diversas tarefas na indústria [8].

Os motores elétricos têm dois constituintes elementares, o rotor e o estator. O rotor é a parte giratória da máquina elétrica, está alojado no interior do estator e da carcaça do motor e consegue-se apenas identificar o seu eixo no exterior. É a partir da caixa de bornes localizada no exterior da carcaça que é feita a alimentação elétrica ao rotor (através das escovas) ou ao estator, dependendo do tipo de motor [8]. Poderá ser feita a alimentação e o comando direto desta forma, ou então, para aplicações mais complexas, é introduzido um circuito de comando e potência, mediado ou não por um controlador, comandado por uma unidade de programação e processamento, um *PLC* ou um *CPU*, com o *software* adequado.

Relativamente ao tipo de corrente elétrica utilizada, os motores elétricos podem ser classificados de corrente contínua (CC, ou DC, corrente directa) ou de corrente alternada (AC). Devido à facilidade de conexão, à baixa manutenção e ao baixo custo de fabrico, os mais utilizados são os motores de corrente alternada, os motores de corrente contínua são utilizados em aplicações mais específicas [8].

Os motores de corrente alterna, em função do número de fases da alimentação, podem ser monofásicos ou trifásicos. Os monofásicos alimentam-se entre o neutro e a fase, os trifásicos alimentam-se entre as três fases [8].

A síntese dos vários tipos de motores elétricos utilizados na indústria, é feita na figura seguinte:

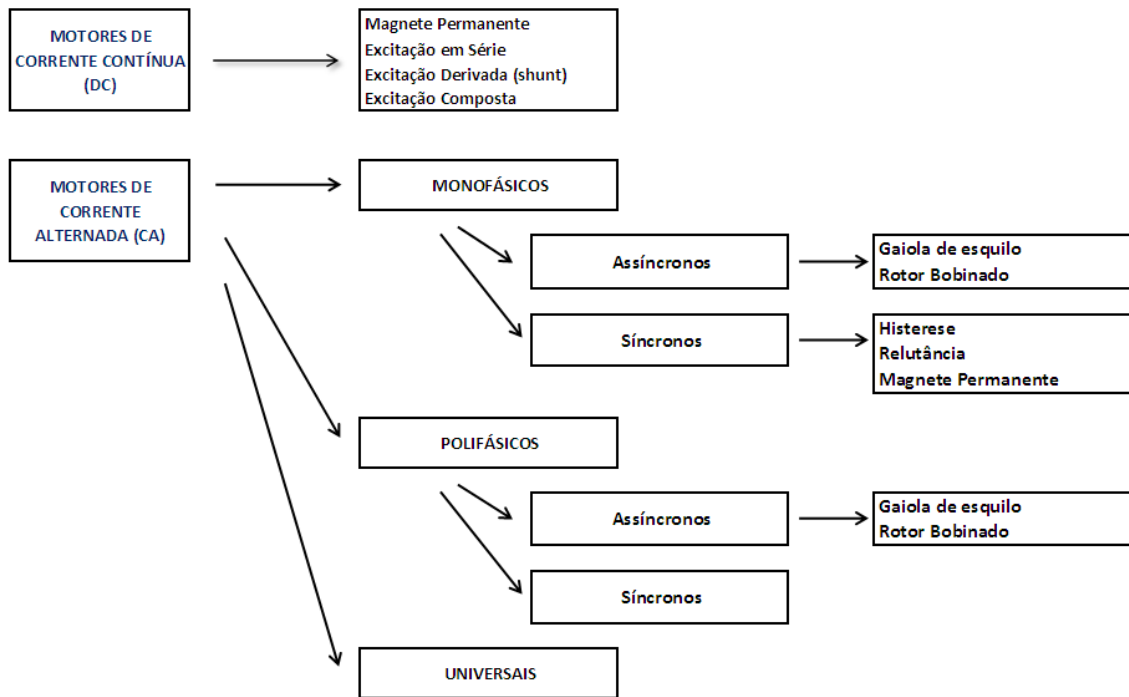


Figura 2.1: Síntese dos vários tipos de motores elétricos utilizados na indústria. Adaptado de [2].

Dependendo do tipo de motor, o rotor pode estar ou não bobinado. O rotor é bobinado no caso dos motores com comutação, nos quais a corrente elétrica é introduzida no rotor sendo que o estator (a parte fixa da máquina) permanece sempre magnetizado (neste caso poderá ser constituído por chapas magnéticas ranhuradas). A rotação do motor é assim promovida pela comutação da corrente para os filamentos do rotor. No caso dos motores sem comutação, o rotor está permanentemente magnetizado e a corrente elétrica é introduzida e percorre o estator (neste caso poderá ser constituído por peças polares bobinadas), a rotação do motor é assim promovida pela movimentação da corrente no estator [6, 3].

Como referido, nos motores com comutação, a introdução da corrente elétrica no rotor, é feita por meio de contactos deslizantes, os comutador e escovas, no entanto, no sentido de diminuir os custos de manutenção associados a estas superfícies deslizantes, desenvolveram-se os motores sem escovas, os *brushless* [3]. Nestes motores é controlado o fluxo eletromagnético aos pólos sequenciais do estator, por sua vez o rotor permanentemente magnetizado é impelido a seguir esse fluxo, dando-se a sua rotação desejada. Este tipo de motores, pelo seu modo de funcionamento discreto, entre polos discretos do estator, facilita a noção do posicionamento veio do rotor e o seu controlo [3].



Figura 2.2: Motor AC *Brushless*, ABB.

A seleção do motor elétrico depende das especificações da alimentação, depende das condições ambientais e depende das exigências de carga e das condições de serviço (potência, rotação, binário, esforços mecânicos, ciclo da operação, etc.). [1, 2]

2.2 – Servomotores

A maioria dos motores elétricos convencionais não foi inicialmente pensada para se sujeitar a um controlo muito restrito do seu funcionamento assim, por exemplo, o seu torque e comportamento dinâmico são afectados quando operam a velocidades reduzidas e a velocidades elevadas, podendo, em resultado, queimar-se o seu sistema de controlo electrónico [3]. A procura da optimização das características dos motores, resultou então no desenvolvimento dos servomotores. Estes tipos de motores, têm como características gerais a sua baixa inércia, a constância de elevados torques a velocidades baixas e elevadas, e a capacidade de poderem situar o seu eixo numa posição angular determinada, e manter essa posição até que seja dada nova ordem [3]. Estes motores tornaram-se importantes para aplicações em que seja necessário ter o controlo total da posição e da velocidade angular do eixo do motor, como por exemplo braços robotizados, máquinas ferramenta, máquinas *CNC*. [8].

Genericamente, o servomotor é um equipamento que faz parte de um servo mecanismo. Esse servo mecanismo é constituído por um sistema integrado com um servomotor (atuador), um sensor, um controlador e uma unidade de programação.

Um atuador servomotor pode incorporar como motor primário um motor elétrico de diferentes tipos, nomeadamente, um motor de passo, um motor *DC* (corrente contínua, com escovas) ou um motor *Brushless DC* (corrente contínua, sem escovas), um motor *AC* Síncrono (com um resolver acoplado) ou um motor *AC*

Brushless (assíncrono, semelhante a um motor trifásico gaiola de esquilo, com um encoder acoplado) [10, 14].

Os motores *Brushless*, pelo seu controlo fácil e pelo seu comportamento mais suave são servomotores muito versáteis [3]. O facto de estes motores não terem comutação e escovas garante-lhes também baixa manutenção [11]. Mais à frente neste capítulo, serão mais aprofundadas as características destes servomotores.

Como já referido, os servomotores, para a monitorização do seu movimento, têm acoplado um sensor, normalmente um *encoder* digital ou um *resolver* digital (também poderia ser um potenciómetro, ou seja um resistor variável, ligado ao eixo), que permite devolver informações de seguimento (*feedback*) dos parâmetros do movimento rotacional do veio, isto é, em todos os instantes, fornece, ao controlador, informações sobre a velocidade, torque, direção e posição do eixo do servomotor.

No caso do sensor acoplado ao servomotor ser um *encoder* óptico, como é o nosso caso, este utiliza a rotação de um disco com uma “linha” opaca, que interrompe num determinado instante um feixe de luz que o atravessa continuamente e é lido pelo foto-detector, resultando numa falha de luz que é lida. Os *encoders* são muito precisos e fáceis de instalar, devendo ser a primeira opção para qualquer sistema, têm no entanto o problema da sua longevidade que poderá não ser suficiente [11].

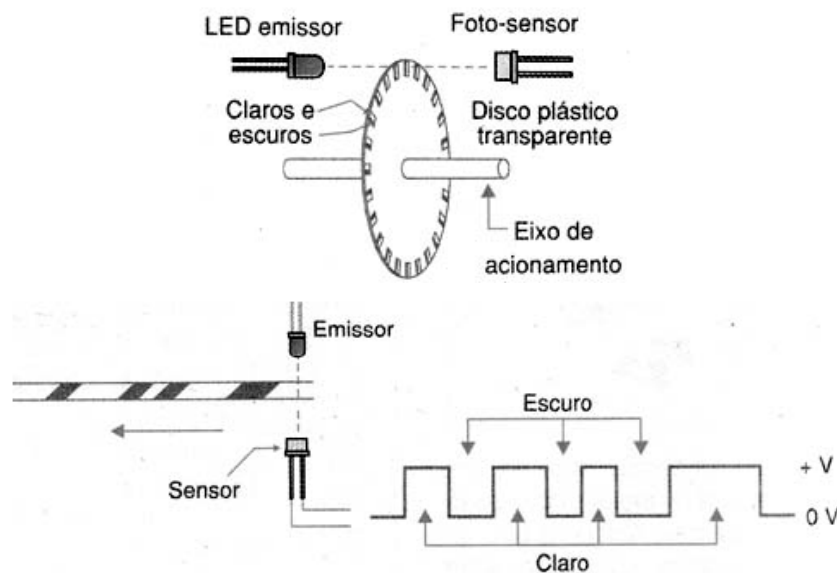


Figura 2.3: Figura explicativa do funcionamento de um Encoder óptico digital.

Adaptado de [14].

Essas informações de *feedback* são processadas no circuito de interno do controlador, sendo feito automaticamente o ajuste do movimento do servomotor, utilizando uma rotina de correcção do erro do seu movimento (posição, velocidade e torque) para os parâmetros indicados pela unidade de programação [10].

Não havendo alteração da ordem dada unidade de programação (mantendo-se o pulso codificado), e se o eixo já atingiu a posição correcta, o servomotor manterá a sua posição até que nova ordem de posicionamento seja recebida, senão, o servomotor rodará na direcção correcta até se atingir a posição pretendida [11]. Existe assim um sistema de controlo de malha fechada, que verifica a cada instante as características do movimento do eixo do servomotor, corrigindo-o e fazendo-o adquirir os parâmetros de movimento desejados.

O circuito de controlo, instalado no interior do controlador, é formado por circuitos e componentes electrónicos, estes subdividem-se na parte de controlo (que integra um microprocessador) e na parte de potência (que integra os transístores na saída alimentação eléctrica para o servomotor). O controlador é geralmente composto por um oscilador e um controlador *PID* (controlo proporcional integrativo e derivativo) que recebe o sinal do sensor acoplado (movimento do eixo) e o sinal de programação de comando (pulsos), e calcula e fornece a alimentação eléctrica ao servomotor através do cabo de potência [11].

O controlador converte desta maneira a energia da alimentação eléctrica recebida da rede (no caso do controlador utilizado, 200/230 VAC (170 to 253 V) 50/60 Hz), sintetizando uma onda sinusoidal (onda sinusoidal da pontos discretos) com as características (amplitude de corrente e tensão) e frequência determinadas para que o servomotor corresponda com o movimento pretendido [7, 11].

Apresenta-se na figura seguinte um diagrama de blocos típico deste tipo de controlo, em malha fechada.

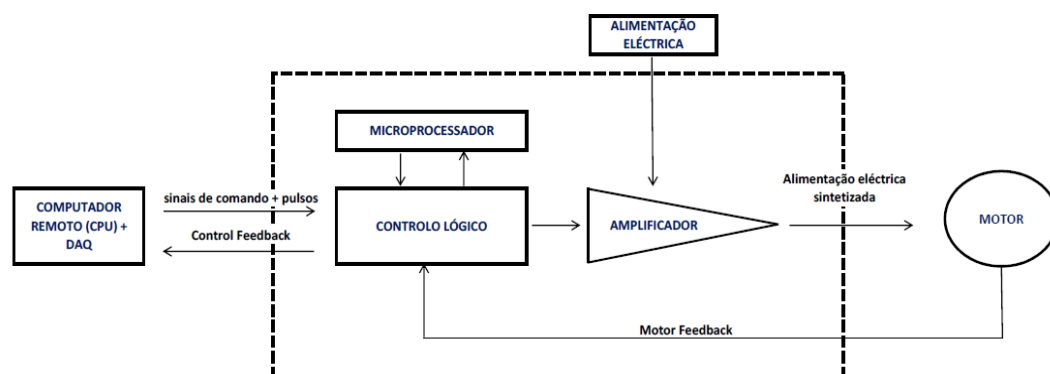


Figura 2.4: Sistema de controlo do servomotor (PWM sinusoidal driving function). Adaptado de [7].

O movimento de rotação de um servomotor é comandado pelos pulsos enviados pela unidade de programação para o seu controlador, e pela sua permanência. Mediante a leitura desses pulsos, através dos métodos *PWM* (Modulação por largura de Pulsos, “*pulse width modulation*”) ou *PFM* (“*pulse frequency modulation*”), o controlador sintetiza a alimentação elétrica correspondente [7, 11].

No nosso caso, o controlador utilizado da *Omron*, R88D-UEP12V, lê a modulação de pulsos *PWM* (“*pulse width modulation*”). O controlador atua através de um *converter* (conversor AC/DC, que permite trabalhar com corrente fixa) e de um inverter (DC/AC) com funcionamento comandado pelos comprimentos dos pulsos (*PWM*). Isto é, controlador transforma a modulação de pulsos recebida numa alimentação elétrica trifásica sinusoidal de amplitude de corrente / voltagem e frequência calculadas, que é enviada para os filamentos do motor, para cada uma das três fases, através da variação do *on time swictes* dos transístores da saída [4].

Como já referido, essa alimentação elétrica sintetizada será adequada para que o servomotor adquira as características do movimento (posição/velocidade) indicadas pela unidade de programação e, debite o torque exigido pela carga a que está correspondentemente sujeito. Resultante da manutenção (ou variação) do torque, o fator de carga do servomotor é assim variável, de acordo com a alimentação elétrica sintetizada, em cada instante, pelo seu controlador [7].

O controlador ao regular o fluxo elétrico de alimentação do servomotor, controla o torque nominal e máximo no eixo, é o chamado controlo proporcional, visto que a energia consumida pelos servomotores (a sua potência libertada) é proporcional à carga (torque) a que estão submetidos (através do fator multiplicativo do torque instantaneamente requerido para obtenção da corrente absoluta que é necessária enviar para o servomotor). Desta maneira é possível controlar o torque do veio do motor, podendo-o manter de forma constante, numa larga faixa de rotação [4, 12].

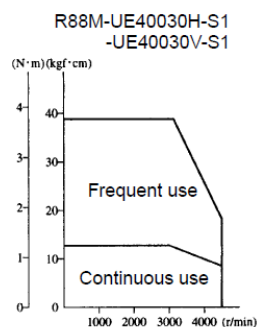


Figura 2.5: Curva característica do Torque vs. Rotação do Servomotor utilizado. Adaptado de [9].

Como se pode analisar na figura anterior, de facto, para além de uma larga faixa de controlo da rotação (até 1:4500 rpm), outra das principais características dos servomotores, como o utilizado, é a possibilidade de manutenção do seu torque praticamente constante numa larga faixa de rotação (na figura até às 3000 rpm), assim como a sua alta capacidade de sobrecarga (cerca de 3 x M_o , torque nominal do motor) sem que ocorra o risco de desmagnetização [9, 12].

Isto explica-se na prática, pelo facto de os motores brushless, como é o caso do servomotor utilizado, poderem ter mais do que um pólo do estator energizado ao mesmo tempo, permitindo o desenvolvimento e manutenção de torques elevados e a diminuição do efeito da variação sinusoidal do torque [7].

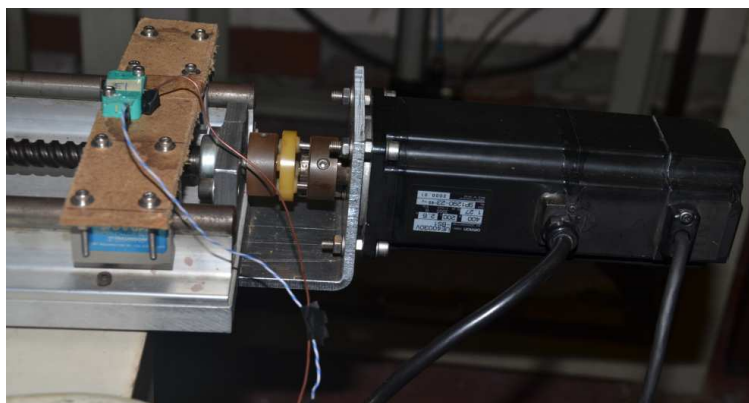


Figura 2.6: Servo motor utilizado R88M-UE40030V-BS1, OMRON. Transmissão mecânica por união elástica e sensor fim de curso.

Simultaneamente, devido às grandes solicitações de dinâmica (acelerações, desacelerações e inversão de sentido de rotação) a que normalmente estão sujeitos os servomotores, estes devem possuir um elevado rácio torque – inércia do motor, sendo que para tal um elevado fluxo elétrico deverá atravessar a baixa massa dos ímãs do rotor. Por conseguinte, a massa destes deverá ser menor do que a dos ímãs dos restantes motores, excepto os motores de baixa inércia (“*low inertia motors*”), [7, 12]. Um das técnicas mais utilizadas são os cortes transversais no rotor, através destes cortes reduz-se a massa indutora de inércia do rotor [12]. Outro método para a redução da massa do rotor passa por usar veios ocos (com o buraco no seu interior) [3].

Finalmente, um servomotor com travão não pode rodar livremente, o eixo desse servomotor só é libertado se o travão eletromagnético que está acoplado for

desbloqueado, ou seja for magnetizado, ou seja, se houver corrente elétrica de alguma forma controlada pelo operador [11].

Genericamente, para a escolha do tipo de motor elétrico do servomecanismo, deverá primeiramente ponderar-se o servomotor de passo, pois é o mais económico, mas para situações em que haja requisitos de operação mais severos, em termos de torque, velocidade e dinâmica (inversão de sentido de rotação, de cargas, de acelerações), ou quando a ausência de manutenção for um fator importante, deverá optar-se pelos servomotores brushless [4].

O servomotor utilizado neste trabalho, é um motor de corrente alternada, trifásico e *brushless*, é portanto motor *AC servo brushless*. Por conseguinte, o controlador terá que gerar uma corrente trifásica de forma sinusoidal. A frequência dos pulsos introduzida no controlador, multiplicada pela resolução implícita do próprio controlador, o ganho electrónico, resulta numa determinada frequência de pulsos (originando uma série de pontos discretos que no seu conjunto formam a curva sinusoidal de alimentação elétrica) que origina a velocidade de rotação do motor, medida em rotações por segundo [4, 9].

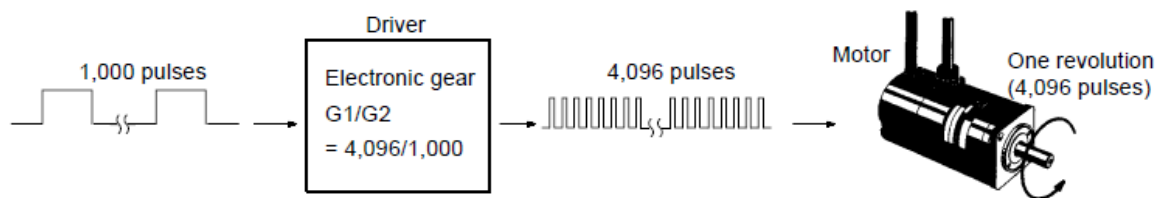


Figura 2.7: Representação do ganho electrónico do controlador utilizado. Adaptado de [9].

O termo *brushless motors* é utilizado para identificar um conjunto próprio de servomotores, apesar dos motores de passo e os motores *AC* de indução na sua constituição serem também *brushless*, uma vez que não têm escovas. De facto o motor de passo, com os filamentos retirados do rotor e passados para os pólos do estator, ficando o rotor permanentemente magnetizado (por ímanes permanentes), pode ser considerado a base do desenvolvimento do *servomotor brushless*, sem contudo, as limitações impostas pela comutação mecânica (a comutação no motor de passo é mecânica), que no caso deste é electrónica [4].

Existem dois tipos de motores *brushless*, os motores de corrente *DC* trapezoidal (os *DC brushless servomotors*), e os motores de corrente alternada sinusoidal (os *AC brushless servomotors*) [4].

Os *AC brushless servomotors*, tal como o utilizado neste trabalho, são operados através da condução da corrente eléctrica no seu circuito externo, no estator, que deverá derivar até à posição pretendida do rotor (sendo necessários meios para conduzir essa corrente automaticamente). Estes motores operam assim de forma similar aos motores *AC* assíncronos, através da aplicação de correntes sinusoidais nos filamentos da bobinagem do estator, o torque dos servomotores virá a ter características sinusoidais, ainda que atenuadas, como anteriormente referido [4].

Usualmente, um motor *brushless* é alimentado por duas ou três fases e o rotor tem quatro ou seis pólos e, por sua vez, cada fase tem os seus filamentos distribuídos por todos os pólos do estator [4]. A corrente deve percorrer o correcto afastamento entre fases ao longo do perímetro do estator, que no caso dum *AC brushless* trifásico esse afastamento deve ser de 120°. Devido ao afastamento entre polos do rotor e do estator, e para termos uma operação suave a baixas velocidades, necessitamos de ter um dispositivo de controlo com alta resolução para controlar a comutação. Necessitamos portanto duma comutação electrónica associada a um resolver ou a um encoder de alta resolução.

Neste caso, de um servomotor *brushless* trifásico, o circuito electrónico do controlador é complexo pois tem que debitar as três correntes eléctricas (três fases) na correcta correlação entre elas, para cada posição do rotor.

Utilizando as três fases, consegue-se ter mais do que um pólo do estator energizado ao mesmo tempo, e deste modo é possível manter por mais tempo a constância dos 90° ideais de ângulo de torque (ângulo ideal entre o campo eletromagnético e a corrente eléctrica dos filamentos da bobinagem, no qual se gera o maior binário) mesmo numa velocidade baixa (as velocidades mais elevadas tendem a amenizar as flutuações do torque), minimizando-se as flutuações no torque.

2.3 – Parâmetros de controlo

Como já referido anteriormente, com a realização deste trabalho pretendeu-se implementar um sistema de controlo automático do movimento dum patim instalado numa mesa/guia linear, que poderá ser utilizado num processo de fabrico industrial. Para tal implementou-se o controlo ativo dos seguintes parâmetros do movimento do servomotor que o aciona:

- Sentido do movimento: associado sentido de rotação do servomotor, ou *Forward Pulse* (sentido dos ponteiros do relógio) ou *Reverse Pulse* (sentido contrário aos ponteiros do relógio);
- Limites do movimento (através de fins de curso mecânicos);
- Velocidade do movimento;
- Aceleração;
- Distância de posicionamento na inicialização (para posicionamento inicial);
- Distância de trabalho a percorrer (na execução da tarefa);
- Os tempos de espera nos patamares de início e de trabalho;
- Mobilização do travão do servomotor.

Para além destes parâmetros do movimento, activamente comandados pelo operador, o controlador do servomotor utilizado devolve ainda a seguinte informação:

- *Alarm output*;
- *INP*.

No nosso caso o *INP* indica quando a posição final é atingida, com um número de pulsos igual ou inferior ao previsto. Por sua vez a informação lógica do *Alarm output* indica que o servomotor entrou em erro quando a posição pré-determinada não é atingida por qualquer motivo, ou quando o torque é ultrapassado.

Inerentes à operação do servomotor, existem ainda mais dois parâmetros que é necessário controlar, são estes, o *Run* (que coloca em modo de *standby* o servomotor, à espera das ordens dos pulsos) e o *Desviation counter reset* (que permite fazer *reset* do alarme instalado no servomotor, quando este por algum motivo ocorre).

Capítulo 3 – Sistema Mecatrónico

3.1 – Definição do projecto de mecatrónica

Este trabalho é na sua natureza um projecto de mecatrónica, uma vez que mobiliza e envolve num sistema de integrado o conjunto dos componentes de mecânica, de electrónica (circuitos e hardware), de controlo e de *software* (Labview) [5]. Podem-se verificar na figura seguinte os elementos base constituintes de um projecto de mecatrónica.

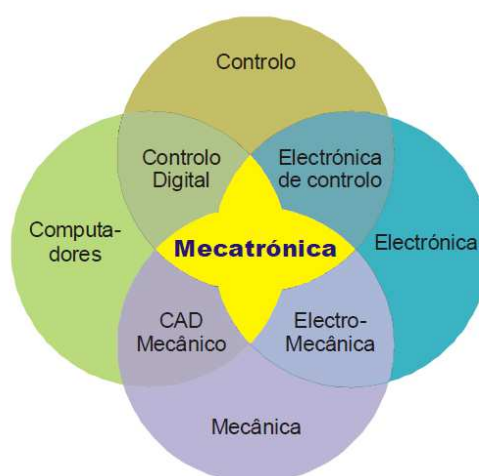


Figura 3.1: Organização dos constituintes de um projecto de mecatrónica. Proposto por [5].

Seguindo o esquema organizacional proposto na figura anterior, e fazendo a interação com o projecto de mecatrónica que se desenvolveu, encontramos o conjunto do *Labview* e da Placa de Aquisição (DAQ) como exemplo de sistema de “*Controlo Digital*”, temos o circuito electrónico do controlador como um exemplo de aplicação da “*Electrónica de controlo*”, encontramos o próprio motor eléctrico primário como um componente de “*Eletromecânica*” e temos ainda a parte da estrutura física da mesa / guia linear, da sua engrenagem e união ao servomotor que constituem um sistema de “*Mecânica*”.

A interação dos componentes utilizados, principalmente os das áreas de electrónica, controlo e computadores (*software*), ainda que já pré-definidos os equipamentos que se iriam utilizar, obrigou ao estudo da interface dos diferentes sinais de comunicação (nomeadamente, a compreensão e controlo de sinais digitais recebidos e enviados pela *DAQ*) e, ao estudo do processamento dos mesmos, tudo numa lógica de controlo de processos, ou seja, foram criadas rotinas fechadas de leitura, cálculo (processamento) e escrita de instruções.

3.2 – Descrição do sistema mecatrónico adotado

Utilizando a abordagem teórica apresentada no subcapítulo anterior, identificam-se de seguida os elementos que constituem o sistema mecatrónico implementado:

- Mesa / guia linear com o patim acoplado, cujo movimento é transmitido por uma engrenagem do tipo sem-fim;
- Servomotor, dispositivo para o acionamento eletromecânico, equipado com o encoder para o feedback do seu movimento;
- Controlador dedicado do servomotor;
- Placa de aquisição de dados *DAQ*, para a interface de dados (leitura e escrita);
- Computador tipo *CPU* (unidade de programação e processamento dos sinais de programação);
- Os fins-de-curso mecânicos;
- O circuito electrónico de condicionamento dos sinais (sinais da programação de comando e sinais provenientes dos fins-de-curso mecânicos);
- Alimentação eléctrica
- *Software* de programação de comando, o *Labview*.

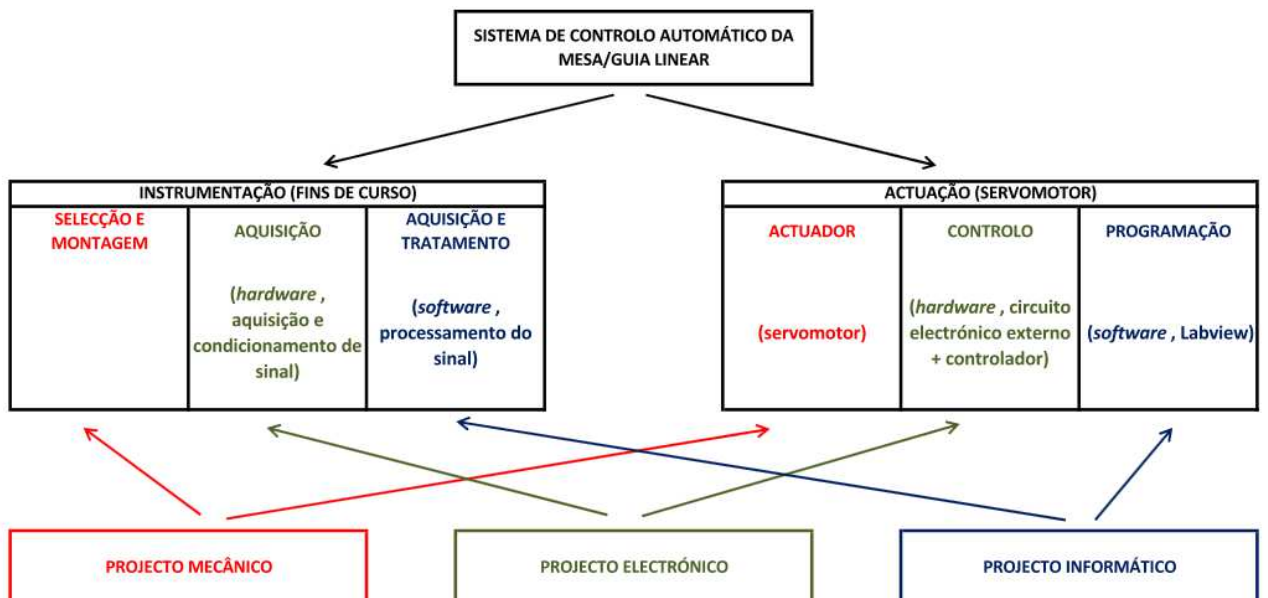


Figura 3.2: Sistema mecatrónico adotado no controlo automático da mesa / guia linear.

A estrutura organizacional, de ligação, dependência e enquadramento, desenvolvida entre os diferentes elementos do sistema mecatrónico adotado, pode ser projetada na figura seguinte.

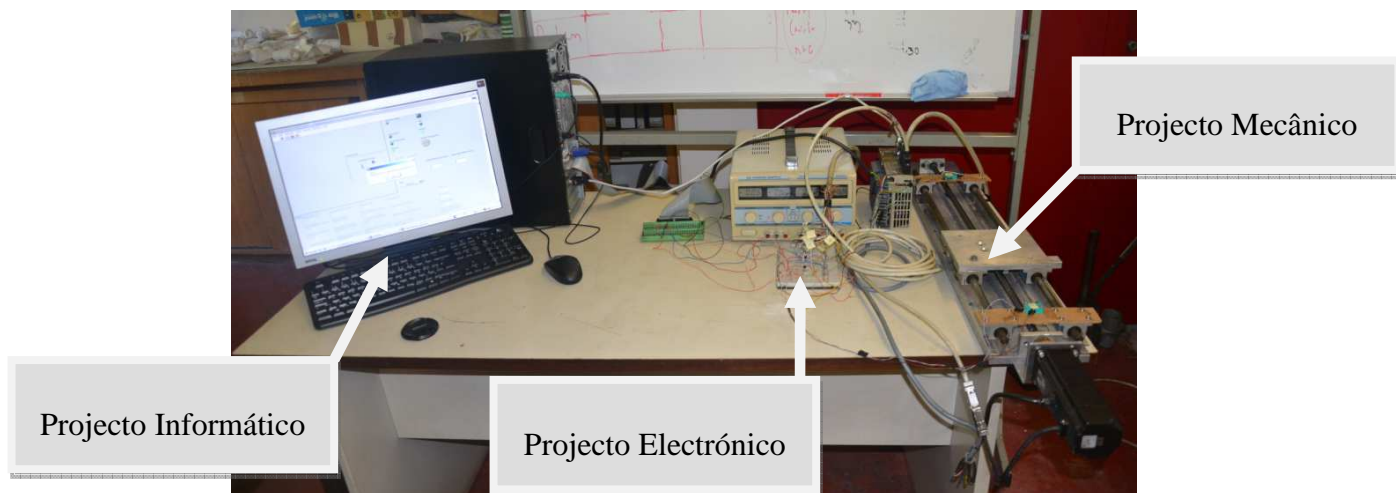


Figura 3.3: Aspeto do sistema mecatrónico desenvolvido, no Laboratório de Fundição do DEM.

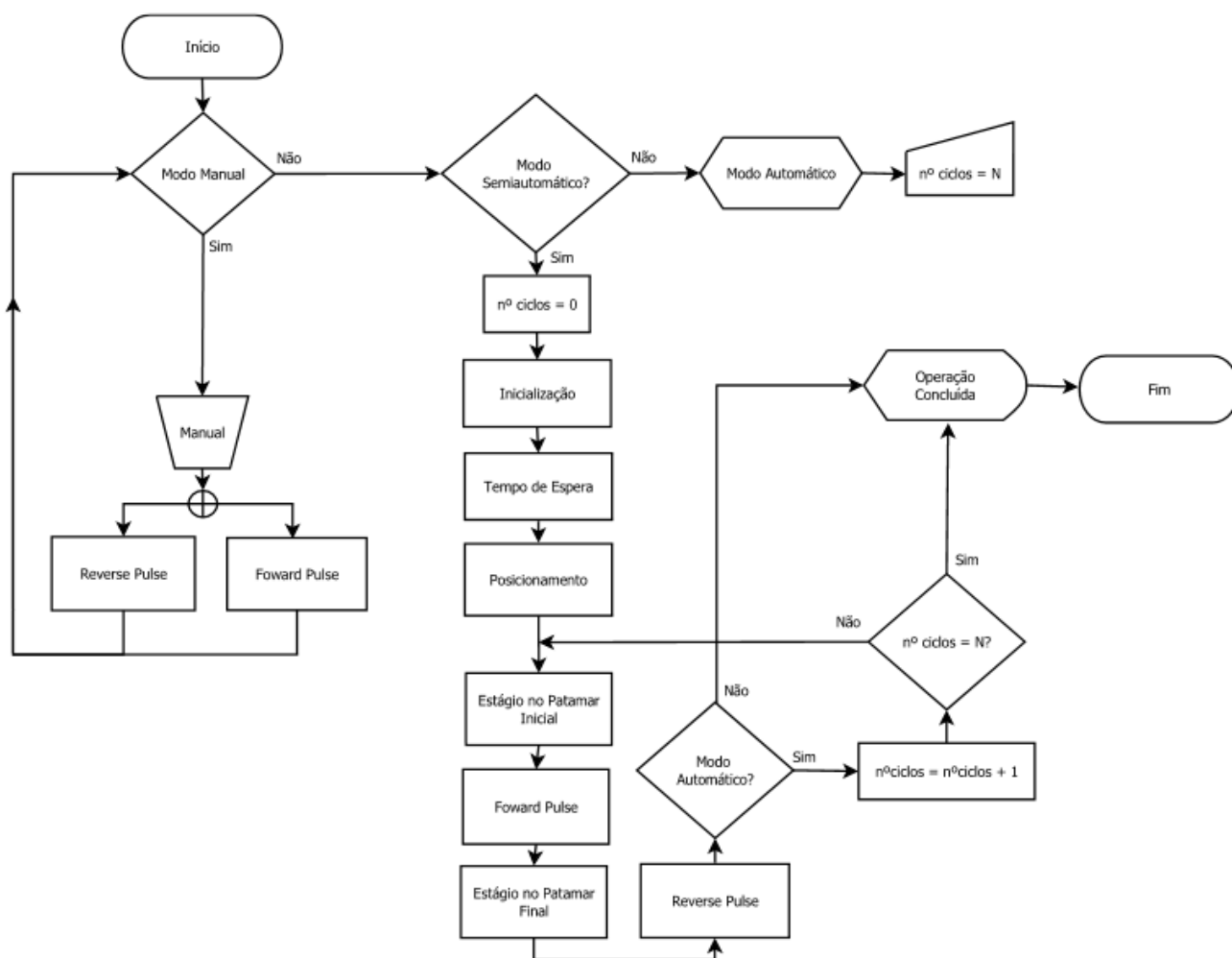


Figura 3.4: Sistema Mecatrónico: Formalismo de Programação. Etapas e Tarefas.

A figura anterior apresenta o resumo (sequência lógica) das principais etapas e tarefas do programa desenvolvido em software Labview, para a operação da mesa/guia linear nos modos semiautomático e automático. O desenvolvimento deste programa será mais à frente apresentado e explicado, no Capítulo 5 deste relatório.

Capítulo 4 – Sistema Eletrónico e Mecânico

4.1 – Servomotor utilizado

O servomotor utilizado neste trabalho foi um equipamento da *OMRON*, modelo *R88M-UE40030V-BS1* que opera através do seu controlador dedicado da *OMRON* modelo *R88D-UEP12V*. O controlador é alimentado por corrente elétrica alterna monofásica da rede, de *200/230 VAC (170 to 253 V) 50/60 Hz* [9]. Os fios de alimentação elétrica devem ser ligados ao controlador no bloco da alimentação aos terminais *R* e *T*. O fio de protecção deve ser convenientemente ligado à massa.

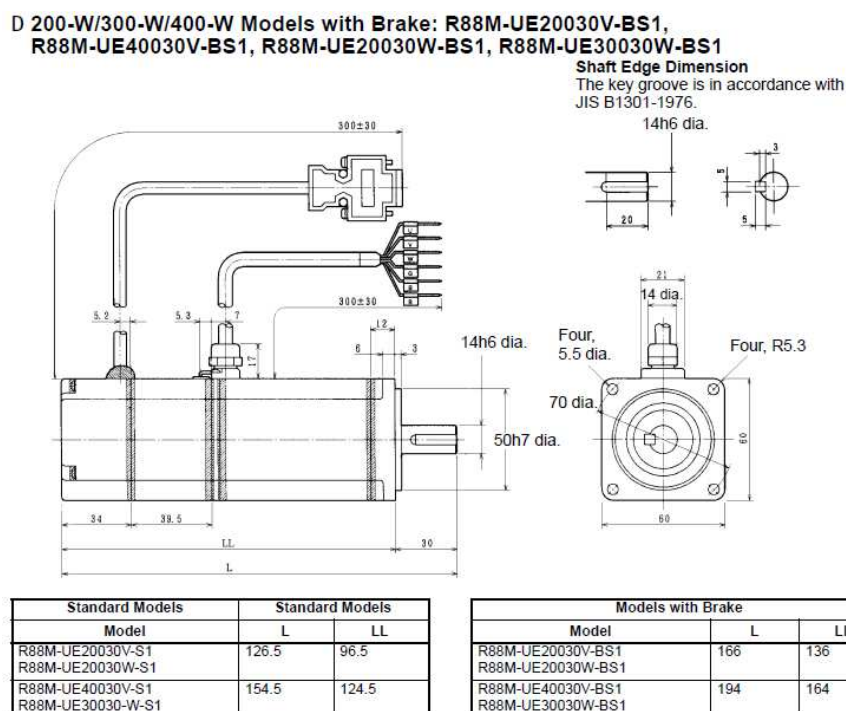


Figura 4.1: Dimensões exteriores do Servomotor. Adaptado de [9].

É um servomotor com potência de 400W, de 3000 RPM e com um torque de 1,27 N*m. A utilização típica deste servomotor será as máquinas de baixa inércia, robôs, tapetes transportadores, etc.) assim como em sistemas mecânicos de rigidez média, como são as mesas XY [9].

O controlador, modelo *R88D-UEP12V*, pesa aproximadamente 1,2 kg, tem uma amplitude de corrente de saída contínua de 3,7 A, no cabo de potência que segue para o servomotor, e tem uma frequência máxima de pulsos de 200 *kpps*. A corrente de saída para o servomotor é alterna trifásica sintetizada pelo controlador.

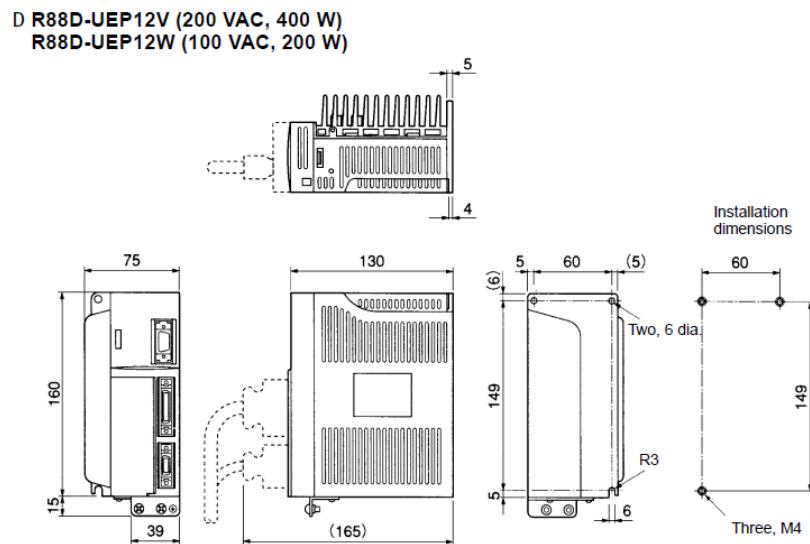


Figura 4.2: Dimensões exteriores do Controlador. Adaptado de [9].

4.2 – Placa de Aquisição utilizada

Para a leitura e escrita dos sinais de comando digitais, instalou-se no computador fixo uma placa de aquisição e controlo de dados, usou-se para tal a placa existente no Laboratório de Automação do DEM, uma a *PCI 1200* da *National Instruments*.

O software da placa de aquisição de dados, *Nidaq 6.93_PCI 1200*, instalou-se de seguida uma vez que o *Labview* tem que importar as características da mesma para a partir daí se conseguir ter acesso às funcionalidades do menu *Data Aquisition* do *Labview*. Sem o software da placa de aquisição de dados instalado, não estão disponíveis na “*Functions Palette*” as funções de aquisição de dados necessárias para a programação da aplicação no *Labview*, tais como “*Port Config*”, “*Port Read*” e “*Port Write*”.

O comando do servomotor foi concebido usando-se sinais de comando digitais, através da interface das 24 entradas e saídas digitais I (*input*) / O (*output*) que a placa possui, subdivididas em 3 portos (portos *A*, *B* e *C*) com 8 bits cada (numerados de 0 a 7). Os portos

podem-se configurar como portos de escrita (output) ou de leitura se sinais (input), um porto não pode ser de entrada e saída simultaneamente. Estes bits operam com níveis máximos de tensão $\pm 5V$ e assimilam os níveis lógicos de tensão *TTL*.

Tabela – 4.1 – Resumo das características das interfaces I/O da DAQ

RESUMO DAS CARACTERÍSTICAS INTERFACES I/O DIGITAIS:	
Número de entradas:	24 I/O (três portos com 8-bits cada)
Tensão compatível:	5 V/TTL
Níveis Lógicos digitais nas entradas (input):	
Nível lógico 0:	-0.3 V a 0.8 V
Nível lógico 1:	2.2 V a 5.3 V
Níveis Lógicos digitais nas saídas (output):	
Nível lógico 0:	(Iout = 2.5 mA) < 0.4 V
Nível lógico 1:	(Iout = 2.5 mA) > 3.7 V
Protecção das portas:	
Tensão admissível na entrada (input):	-0.5 to 5.5 V powered on, ± 0.5 V powered off

A placa possui também 8 entradas analógicas (*Analog Inputs*) com uma capacidade de leitura de amostras de 100 kS/s e 2 saídas analógicas, com uma resolução na conversão analógica-digital de 12-bit. A gama de tensão de entrada na placa pode ser programada para ser unipolar 0 – 10V ou bipolar $\pm 5V$. A placa possui ainda 3 contadores / *timers*. Estes elementos não foram utilizados neste trabalho, as interfaces digitais foram suficientes.

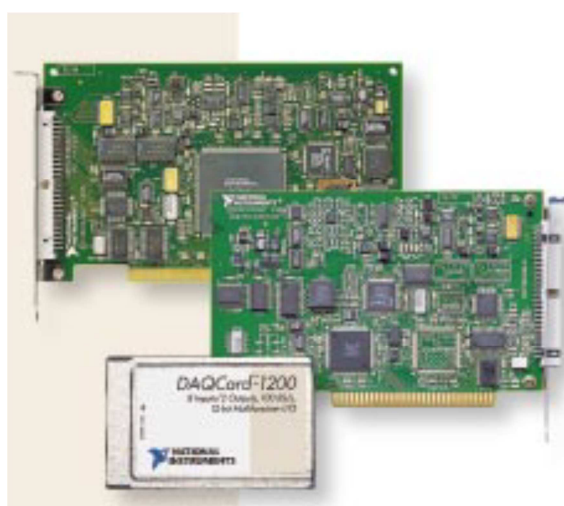


Figura 4.3: Aspeto geral da DAQ instalada na torre da CPU

A placa *DAQ* da *NI* utilizada trabalha com níveis de corrente de saída baixos, 2,5 mA, assim, e como este valor de corrente não é suficiente para fornecer os sinais de entrada do nosso controlador, houve necessidade de realizar um circuito electrónico exterior com uma

alimentação de corrente contínua de 5V e transístores que permite amplificar o sinal que segue posteriormente para o controlador.

4.3 – Circuitos electrónicos utilizados

Como anteriormente abordado, os sinais da programação de comando do movimento do servomotor, referentes às ordens *Foward Pulse*, *Reverse Pulse* e *Deviation counter reset*, enviados para o controlador são sinais lógicos que provêm do *Labview*, através das saídas digitais da *DAQ*, de 5V e 2,5 mA, com níveis de tensão *TTL*. Por sua vez, a corrente requerida pelo controlador nas suas correspondentes entradas é de 6mA a 3V [9]. Houve assim necessidade de utilizar um circuito electrónico externo adicional a intermediar este processo, para amplificar os sinais provenientes DAQ.

Para a materialização desse circuito electrónico utilizou-se uma placa electrónica do DEM na qual se conectou uma fonte de alimentação contínua de 5V. Arquetetou-se de seguida um circuito electrónico que mediante transístores amplifica a corrente que seguirá para o controlador.

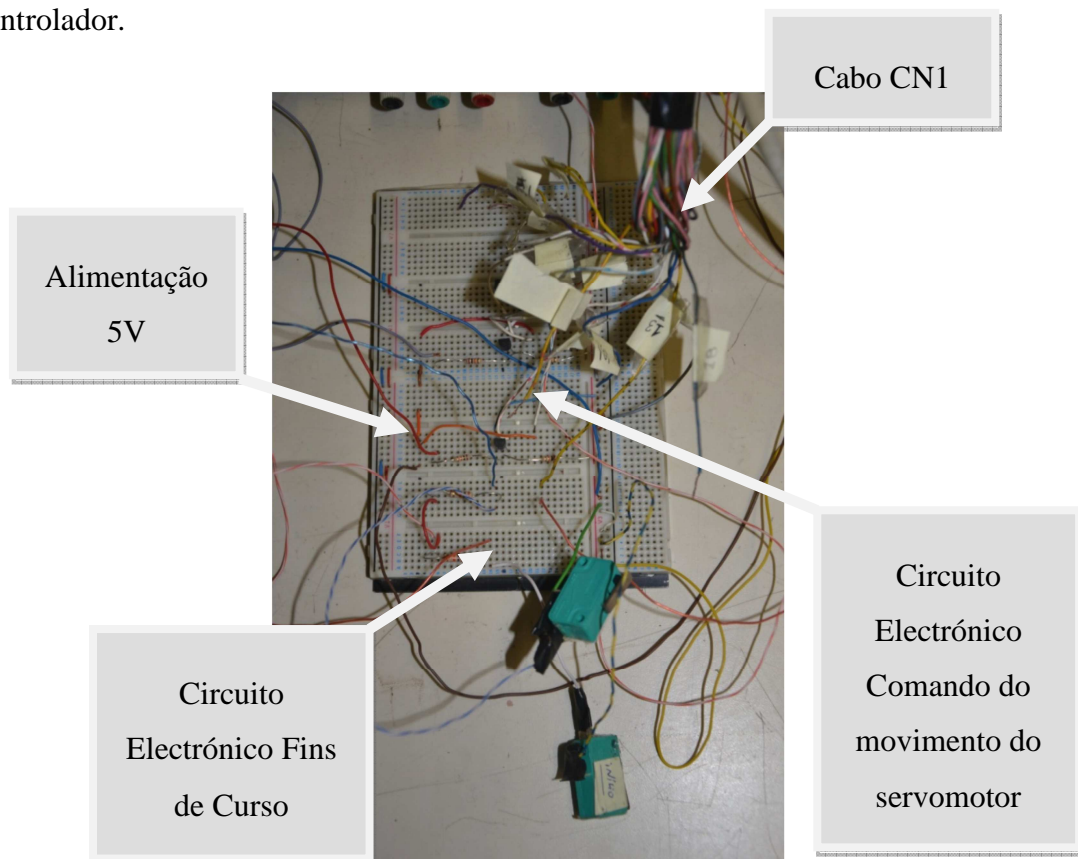


Figura 4.4: Placa electrónica utilizada, “bread board”, existente no DEM.

O sinal digital saído da *DAQ* é ligado à base de cada transístor *NPN* utilizado (*BC 548*), abrindo a passagem e amplificando a corrente proveniente da fonte de alimentação exterior. A entrepor-se entre a corrente eléctrica saída da *DAQ* e a base do transístor foi criada uma ligação à massa e foram colocadas resistências, para reduzir a tensão de entrada na base (que basta ser superior a 0,70V) e para proteger a *DAQ* de tensões demasiado elevadas ou eventuais curto-circuitos. A geometria de cada circuito dedicado, repete-se para as instruções *Forward pulse*, *Reverse Pulse* e *Deviation counter reset*.

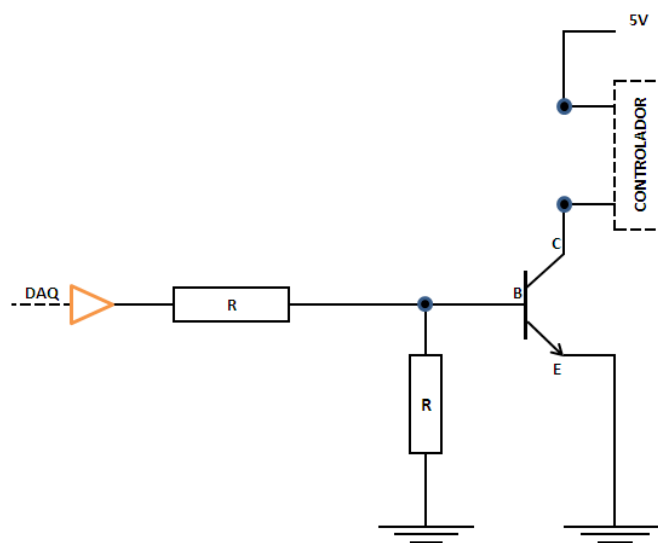


Figura 4.5: Representação do circuito electrónico elementar utilizado.

Para além dos três sinais de comando indicados anteriormente, é necessário ainda amplificar as instruções de seguimento do movimento do servomotor utilizados, que são o Run e o Alarm Reset. Para o condicionamento destes dois sinais, realizou-se um circuito electrónico externo distinto, utilizando ligações disponíveis na mesma placa de electrónica, e alimentado por uma corrente eléctrica de 12 a 24 V e 30 mA.

Os sinais resultantes do circuito electrónico externo, que seguem para o controlador, e que usam sinais digitais com níveis de tensão *TTL*, devem ter comprimentos mínimos de 2,5 μ s (figura seguinte), para poderem ser lidos pelo controlador.

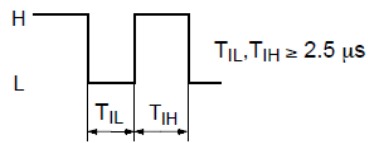


Figura 4.6: Comprimento mínimo dos pulsos de comando do controlador. Adaptado de [9]

A interligar o controlador e o servomotor existe um cabo de seguimento, *feedback*, (cabo proveniente do encoder) e um cabo de potência (que inclui o cabo de controlo do travão).

No cabo de potência seguem os fios que se conectarão aos três terminais de outputs, que o controlador possui, destinados ao servomotor, são estes o terminal *U* (vermelho), o terminal *V* (branco) e o terminal *W* (azul), correspondentes às três fases de saída para o servomotor, por sua vez o fio verde será o condutor de ligação à massa. O cabo de potência possui também dois fios para o controlo do travão do motor, um é a massa e o outro fica conectado ao correspondente circuito electrónico externo de condicionamento de sinal e comando pelos sinais provenientes do *Labview* através da *DAQ*.

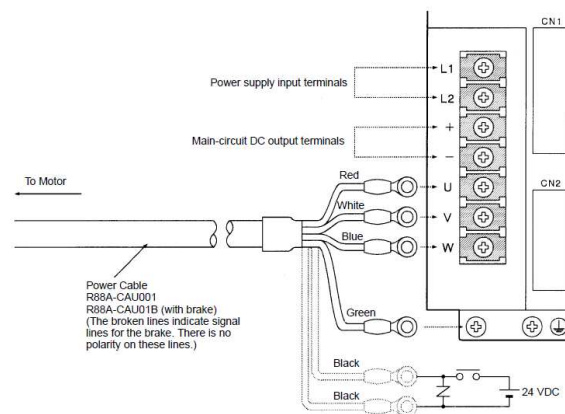


Figura 4.7: Cabo de potência do servomotor

O cabo proveniente do encoder conecta-se à porta *CN2*, sendo a corrente de saída do servomotor para o controlador de 24 V, 20mA, com uma resolução de 1,024 pulsos por revolução (rotação), com uma frequência máxima de 200 *kpps*.

As instruções do *Labview* por intermédio da *DAQ* são dadas ao controlador através dum cabo de comando, que está ligado aos pinos do conector da placa de aquisição de dados e cuja outra extremidade se liga na porta *CNI* do controlador. Esse cabo de comando materializa-se num cabo de 36 pinos, do qual não se utilizou todos os pinos, apenas os seguintes: pinos 1 e 2 (*reverse pulse*), pinos 3 e 5 (*forward pulse*), pinos 4 e 5 (*deviation counter reset*), pino 7 (*travão*), pino 8 (*INP, posição cumprida*), pino 13 (entrada da alimentação de 12 V a 24 V que alimentará os circuitos das instruções *Run* e *Alarm reset*), o pino 14 (*Run*) e o pino 18 (*Alarm reset*) [9].

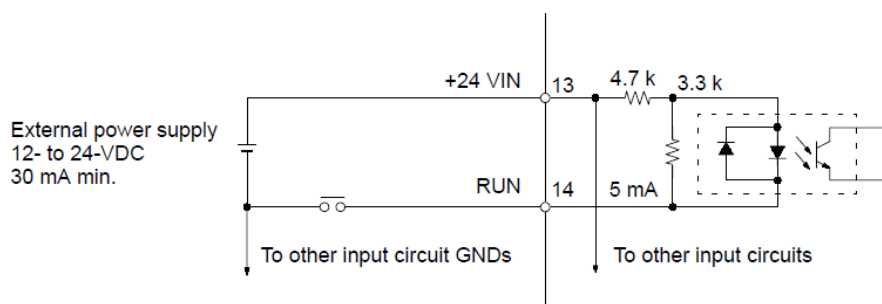


Figura 4.8: Alimentação elétrica exterior aos pinos 14 e 18. Adaptado de [9].

Uma vez identificados os pinos na extremidade, cujos fios foram previamente expostos, estes foram conectados directamente à placa de electrónica usada neste trabalho para condicionar os sinais, possibilitando ao controlador de receber as instruções dadas pelo *Labview* através da placa de aquisição dados *DAQ PCI-1200* com o respectivo sinal condicionado.

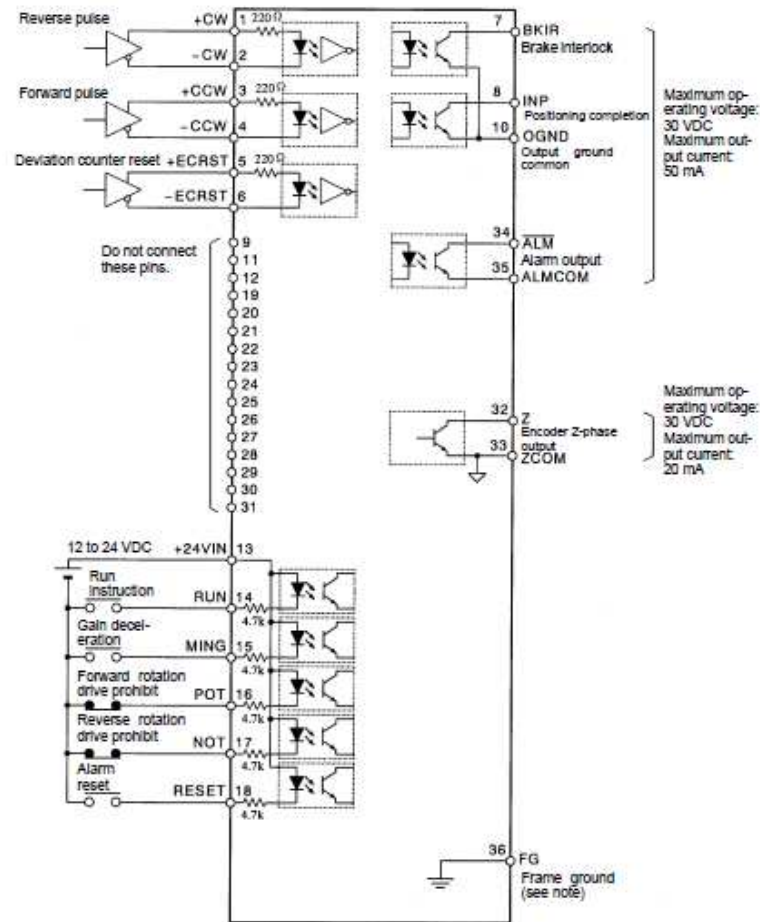


Figura 4.9: Representação das Entradas/Saídas do terminal CN1 do Controlador. Adaptado de [9].

Como anteriormente referido, as instruções controladas pelo nosso programa no *Labview* são o *Run*, *Reverse Pulse*, *Forward Pulse*, *Deviation Counter reset* e *Alarm reset*. Por sua vez, as saídas do controlador, resultantes de sinais provenientes do encoder do servomotor, são a indicação do estado do travão *BKRI*, a indicação de posição completa (*INP*) e a indicação de erro na operação do servomotor (*Alarm output*).

É a instrução *Run* que permite ativar a alimentação do circuito principal do controlador e colocar o servomotor em *standby*, a aguardar instruções.

Relativamente à instrução do travão eletromagnético do servomotor, *BKRI*, foi criado um circuito de condicionamento de sinal independente ligado a uma fonte de alimentação autónoma de 24V. Esse circuito paralelo ficará dependente da instrução de *Run*, estando o *Run* ativo é dada ordem de comando ao controlador para desativar o travão (magnetizando-o), que é reencaminhada para o servomotor, através do cabo de potência.

O travão magnético do servomotor fica atuado quando o *Run* está *OFF*, ou seja quando o motor não estiver magnetizado. O travão do motor não deve ser atuado quando a instrução *RUN* está ativa, pois o travão sofrerá um desgaste anormal [9]. Por esta razão o controlador prevê uma decalagem de tempo entre a instrução *Run* e a operação do servomotor, só uma fração de segundo depois do *Run* estar *OFF* é atuado o travão, ou seja é desligada a alimentação ao travão. O travão do servomotor desacelera o servomotor para pará-lo quando a instrução *Run* está *OFF*, o Alarm está *ON* ou quando a alimentação elétrica é desligada [9].

Relativamente à velocidade dos pulsos que seguem do controlador para o servomotor, esta resulta da multiplicação da velocidade dos pulsos provenientes da unidade de programação, através da *DAQ*, pelo ganho electrónico do próprio controlador proveniente da fórmula $G1 / G2$. $G1$ e $G2$ são dois fatores que podem ser impostos pelo operador. Se $G1/G2 = 1$, ou seja, se não houver ganho de velocidade electrónico, o servomotor fará uma rotação completa a cada 4.096 pulsos [9].

4.4 – Seleção, Montagem e Condicionamento de Sinal dos Transdutores

Para a deteção da presença do patim nos fins de curso da mesa / guia linear foram seleccionados dois sensores eletromecânicos, que foram montados imediatamente antes do barramento físico existente no início e no fim do movimento do patim. Esses fins de curso são sensores de presença com contacto do tipo *Miniature Basic Switch D3V* da *Omron*.



Figura 4.10: Representação dos fins de curso utilizados.

Os dois fins de curso utilizados são interruptores com contactores do tipo normalmente aberto, ou seja, quando na posição de repouso não deixam passar corrente elétrica. Nestas circunstâncias o respectivo pino do porto de entradas digitais está a receber uma tensão 5V, valor lógico 1. Quando se dá o contacto com o patim da mesa / guia linear, através de uma patilha existente numa das faces do sensor, o contactor é fechado deixando passar corrente através dele que descarrega à massa. Nesse instante o pino de entrada deixa de receber

tensão, lendo portanto o valor lógico 0. Assim permanece até que seja reposicionado o patim numa posição intermédia aos fins de curso, deixando de interferir com os sensores. O circuito utilizado repete-se para os dois fins de curso.

Fazendo a correspondente programação no *software Labview*, isto é, admitindo a presença contínua de 5V nos pinos de entrada conectados aos circuitos dos fins de curso como posição do patim válido, e o contrário, como posição inválida, colocou-se a funcionar o sistema mecatrónico concebido integrado com os sensores de fins de curso, balizando o movimento do patim.

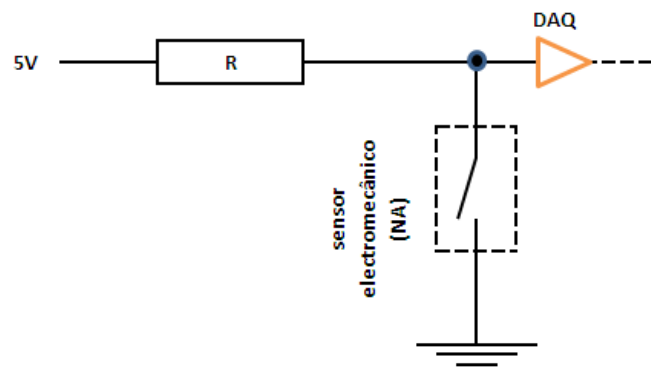


Figura 4.11: Esquema Eletrónico do fim de curso utilizado.

4.5 – Sistema Mecânico

A componente mecânica do sistema mecatrónico implementado é constituída por uma mesa/guia linear, cujo patim desliza sobre dois carris cilíndricos contínuos, através de quatro “almofadas” de apoio, da marca *Thomson*, modelo *Super Pillow Block SPB-8-OPN*. A engrenagem de transmissão de movimento é materializada por um veio do tipo fuso de esferas, *ball screw*, que transforma movimento rotacional em movimento rectilíneo, segundo o eixo longitudinal da guia. Trata-se de um fuso rotativo, cuja porca está acoplada à inferior do patim, e que tem dois pontos de apoio nos extremos, dois rolamentos. Numa das extremidades, o movimento de rotação proveniente do servomotor é transmitido ao parafuso, fuso de esferas, através de uma união elástica.

Os apoios do patim, identificados em cima, dispõem de rolamentos esféricos na superfície de contacto, para se deslocarem sobre os carris com o mínimo de atrito. Estes blocos de apoio são do tipo aberto, para permitir o encaixe nos dos carris contínuos da guia linear, conforme se visualiza na figura seguinte.

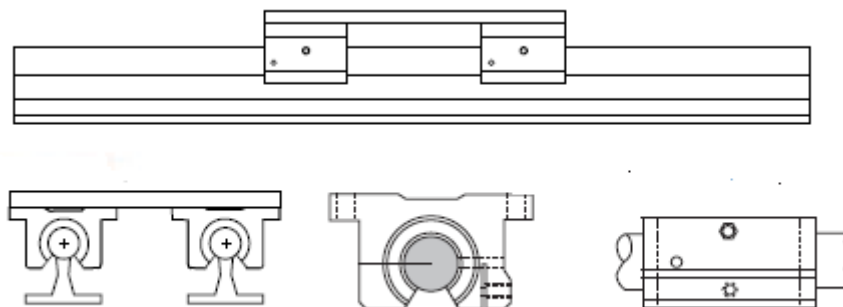


Figura 4.12: Aspeto do perfil longitudinal e transversal da mesa/guia linear. Adaptado de [21]

O tipo de mesa linear utilizada, com apoios contínuos, é recomendada para suportar, transportar e elevar de cargas, e quando é necessária rigidez no sistema (proporcionada pelo seu apoio contínuo), sendo, segundo o catálogo do fabricante, a solicitação dinâmica resistente é de 2581 Newton, ou seja 645 Newton por cada apoio [21].

A grande vantagem na utilização do fuso de esferas é a obtenção de movimentos suaves e precisos, a necessidade de baixo binário para o início do movimento, elevadas velocidades, o movimento silencioso e a vida útil prolongada [19]. Nos fusos de esferas o contacto entre o parafuso e a porca é feito por intermédio de um elevado número de esferas de rolamento, que rolam nas depressões do dentado (do parafuso e da porca), reduzindo o atrito no movimento. Esta redução do atrito conduz a economia no consumo de energia do sistema. A força de contrato é decomposta pelo elevado número de esferas de rolamento.

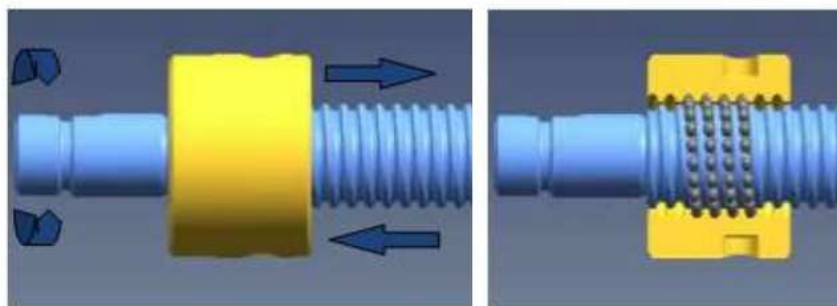


Figura 4.13: Representação tipo do fuso de esferas e porca. Adaptado de [20]

O aspeto geral da mesa/guia linear retirado do *Solidworks* é apresentado na figura seguinte.

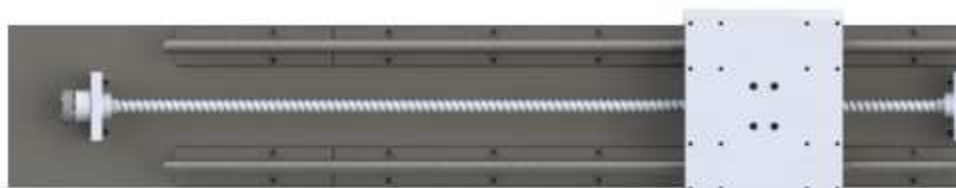


Figura 4.14: Representação da mesa / guia linear utilizada. Visualização do Solidworks.

Para a fixação do servomotor à estrutura da mesa, desenvolveu-se uma estrutura em forma de “L” fixa à estrutura da mesa por meio de dois parafusos M8, com cabeça sextavada interior, que foi executada nas Oficinas do DEM. Para a fixação do servomotor, na posição que permitisse a união de veios concêntrica, elaborou-se uma flange devidamente posicionada e fixa com quatro parafusos, porca e contraporca.

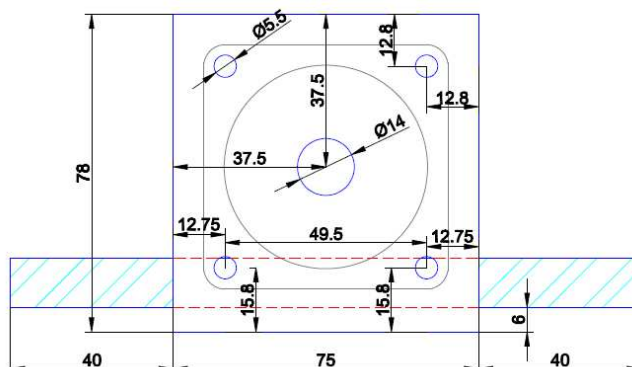


Figura 4.15: Vista de frente, flange de acoplamento do servomotor à mesa/guia. Visualização do AutoCad.

A união do servomotor à mesa/guia linear trata-se assim uma união rígida, contudo, como anteriormente referido, a união dos veios motor e movido, concêntricos, foi efetivada por meio de uma união elástica.

Esta trata-se de uma união elástica com disco intermédio (com um elemento central, um disco em borracha endurecida que absorve todos os tipos de desalinhamento), tipo disco *Brampton*, que é solicitada para a transmissão do binário por meio de três pinos metálicos ligados aos pratos metálicos extremos, que por sua vez estão fixos ao veio motor e movido, por meio de um parafuso de fixação.

A opção pela utilização de uma união flexível com alguma flexibilidade torsional, para ligar o veio do motor elétrico ao veio movido (fuso de esferas), revelou-se vantajosa por ter a capacidade de colmatar erros de fabrico e instalação, dado que consegue absorver e compensar alguns desalinhamentos entre os veios (paralelo, axial, angular e torsional) e minimizar vibrações e choques, poderia ainda ser utilizada para limitar potência máxima transmitida. As uniões flexíveis são relativamente baratas e a sua capacidade de transmissão pode ser superior a 115cv e 1750 rpm [16, 17, 18].

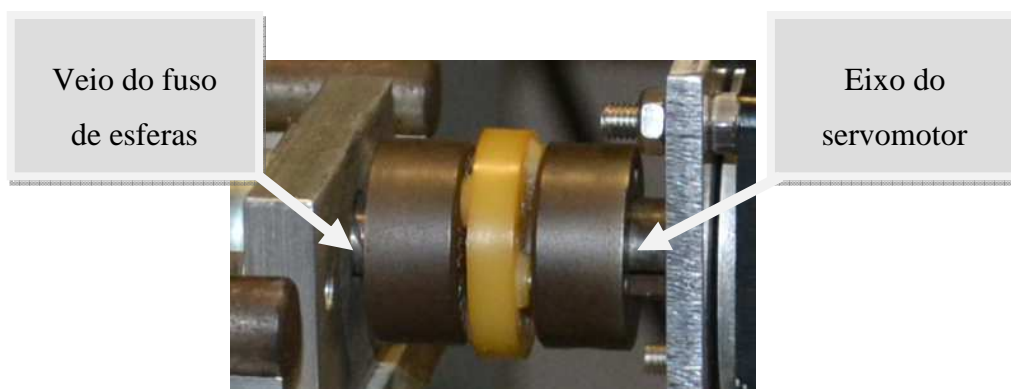


Figura 4.16: Fotografia da união elástica com disco utilizada.

Os dados de partida para a seleção de uma união de veios deste tipo são, sucintamente, o binário/potência a transmitir [N.m], a velocidade de rotação [rpm], o diâmetro do veio e o tipo de máquina motora e máquina movida.

Outro dado que deve ser tido em conta, são as condições de funcionamento do sistema mecânico, ou seja, o tipo de solicitações a que está sujeito e o modo como atuam, as condições de carga (constante ou com variações bruscas, o funcionamento sem choques, ou outro tipo de funcionamento), deste modo determina-se o fator de serviço, que também interfere na seleção da união. Recorre-se depois aos gráficos e tabelas de seleção das uniões de veios dos fabricantes, *Brampton-Renold* ou tipo *Rotex*, por exemplo, para o pré-dimensionamento da união elástica. Nesses próprios catálogos estão indicadas as dimensões e os desalinhamentos admissíveis das uniões.

Para evitar o movimento angular e axial entre os veios (motor e movido) e os pratos metálicos extremos da união elástica, utilizaram-se parafusos de fixação. Os parafusos têm cabeça sextavada interior, para ficar completamente embebidos nas peças, e estão inseridos nas ranhuras longitudinais existentes nas pontas cilíndricas dos veios, os parafusos são assim

usados para transmitir binário (neste caso o veio é fixo em rotação e livre em translação, por ação de uma ranhura longitudinal junto da extremidade do veio).

Capítulo 5 – Programação no Labview

Este capítulo abordará o tema da programação de comando do servomotor através do *software Labview*. O Labview é uma linguagem de programação desenvolvida e comercializada pela *National Instruments*.

Na interface com o programador, o *Labview* não é uma linguagem de programação em código de texto, é sim uma linguagem de programação gráfica e organizada em diagramas de rotinas e sub-rotinas, através dos quais o operador tem a possibilidade de visualizar os dados e ordens a fluírem, ou seja, todo o caminho que eles percorrem, como são coordenados e direcionados.

O painel de trabalho designa-se por Diagrama de Blocos (“*Block Diagram*”), e nele o programador consegue colocar, retirar e mobilizar os Instrumentos Virtuais (*VI*s) que necessita, consegue alocá-los dispostos e interligados num diagrama, de modo a constituírem o programa lógico pretendido.

Os Instrumentos Virtuais (*VI*s) são subprogramas de aquisição e processamento de dados, que já se encontram criados no Labview, fazendo parte da sua base de dados (“*VI Library*”), ou que podem ser criados pelo próprio programador. Estes subprogramas podem depois ser assemblados noutros programas que se criem posteriormente, e que necessitem das suas funcionalidades que estes garantem.

O Labview possui também outro painel de interface com o programador, este designa-se por Painel Frontal (“*Front Panel*”). Este painel e os instrumentos aí inseridos estão intrinsicamente ligados aos *VI*s do Diagrama de Blocos. No Painel Frontal podem-se assemblar um conjunto de controlos e indicadores pré-definidos no *Labview*, os que o programador desejar (por exemplo comutadores, botões rotativos, visores alfanuméricos, gráficos, etc.), tendo em conta que será também neste painel que o operador virá a definir as variáveis da operação do equipamento, no caso deste estudo os parâmetros do movimento da mesa/guia linear.

É no Painel Frontal que são introduzidas as ordens e os parâmetros de comando do sistema, permitindo ao programador (na fase de desenvolvimento) e operador (na fase de

operação), a interação com o programa e a visualização dos resultados de saída das aplicações que criou, facultando assim o controlo interactivo do sistema.

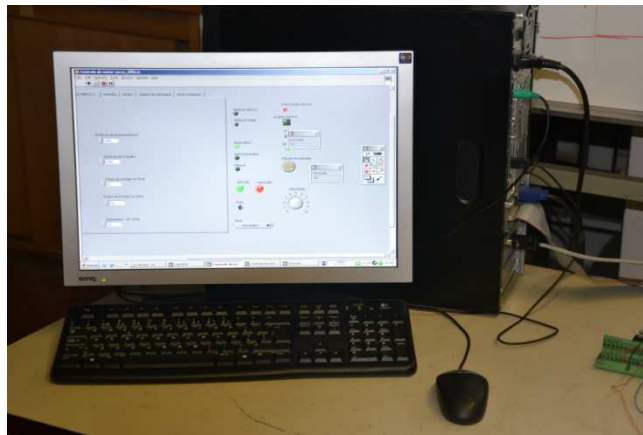


Figura 5.1: Aspeto do Painel Frontal, na CPU utilizado.

O *software Labview* permite uma velocidade de processamento elevada, dado que tem um compilador automático que permite gerar simultaneamente e de forma oculta um código de programação otimizado, permitindo, na execução da aplicação, obter velocidades de execução idênticas às dos programas compilados em linguagem C [10].

5.1 – Programa desenvolvido

A versão do *Labview* utilizada para o desenvolvimento do projecto mecatrónico foi a versão [6.1], da *National Instruments*, cuja licença está disponível no DEM.

Como se depreende da exposição feita nos capítulos anteriores acerca do controlo do servomotor, o programa desenvolvido no *Labview* terá que assimilar e comandar as seguintes variáveis implícitas do funcionamento do servomotor:

- *Run*;
- *Forward Pulse* e *Reverse Pulse*;
- *Desviation counter reset*;
- O travão;
- Velocidade (mm/s);
- Aceleração (mm^2/s);
- Alarme por excesso de torque (*Alarm output*);
- Fins de curso eletromecânicos.

Para além destas, foram estipulados os possíveis modos de funcionamento semiautomático e automático, que é possível selecionar através dum módulo de seleção *Ring* disposto no Painel Frontal. Nos modos de funcionamento automático e semiautomático, outras variáveis são também comandadas, designadamente:

- Distâncias de posicionamento (inicialização) e de trabalho (mm);
- Tempos de paragem (s).



Figura 5.2: Visualização da Seleção do modo de operação (“Ring”).

Painel Frontal do Labview.

Para o desenvolvimento do programa de comando do servomotor, recorreu-se às funcionalidades dos *VI*s já disponíveis na base de dados da versão [6.1] do *Labview*. Recorrendo à *Functions Palette*, na *Palette View*, e selecionando o botão da *Data Acquisition* encontraram-se alguns tipos de *VI*s, tais como *Analog inputs*, *Analog outputs* e os *Digital I/O*. Utilizou-se este último tipo de *VI*s, *Digital I/O*, uma vez que este incorpora as funções digitais que permitem escrever, ler e apagar informações dos pinos de portos de input ou output da Placa de Aquisição de Dados.

Como anteriormente referido, o *software Labview* permite a implementação simultânea de vários *VI*s, sendo que cada um deles representa um subprograma, dispondo assim do próprio painel de controlo e visualização (“*Front Panel*”) e diagrama de blocos de programação (“*Block Diagram*”). Estes permanecem ocultos e só se tornam ativos após mobilização do programador nesse sentido.

Para facilitar a implementação do *software Labview*, os seus *VI*s estão já preparados para assimilar e configurar as características e capacidades contidas nos componentes e

circuitos electrónicos de vários *hardwares* comerciais, designadamente a Placa de Aquisição de Dados utilizada neste trabalho, *DAQ* da *National Instruments*.

Durante o desenvolvimento do programa no *Labview*, foi necessário criar uma forma de comunicar com os canais físicos da *DAQ*, os portos lógicos. Para tal utilizaram-se os *VIs Port config*, *Port write* e *Port read* para que os dados gerados pelo programa pudessem ser comunicados à placa de aquisição de dados e, através desta, por meio dos pinos de um conector, ao controlador do movimento do servomotor.

No Diagrama de Blocos, a comunicação entre os *VIs*, fórmulas e sub-rotinas é efectuada por intermédio das ligações representadas a verde (*booleana*) ou a azul (a *alfanumérica*) ou a cor rosa (texto ou associação de dados provenientes de *Clusters*). Estas ligações são canais virtuais sincronizados de forma que os blocos possam trocar dados entre si. Por outro lado, quando se dá um erro num dos blocos, este é comunicado através da linha roxa.

Os sistemas de manufatura instalados em linhas de produção actuais podem ser operados em modo manual, semiautomático e automático. Os seus atuadores, os motores eléctricos são normalmente utilizados em modo manual quanto está a ser feita a manutenção na linha de produção ou quando existe um erro na operação automática ou semiautomática (para diagnóstico, correcção e inicialização do sistema), procurou-se implementar a mesma lógica no sistema desenvolvido.

O programa no *Labview* fornece os sinais digitais no porto de *output*, para o modo de funcionamento seleccionado (automático, semiautomático ou manual), para tal precisa de ter a definição dos vários parâmetros que decidem o funcionamento do motor para quando este opere em modo manual, semiautomático ou automático.

As variáveis velocidade (mm/s) e aceleração (mm^2/s) são definidas pelo operador no painel frontal e são comuns a todos os modos de operação (manual, automático e semiautomático). Estas variáveis podem contudo ser alteradas a qualquer momento, mesmo durante a operação semiautomática e automática.

Através do programa desenvolvido, estas variáveis controladas pelo operador determinam os correspondentes parâmetros do movimento da mesa/guia linear. Como se pode verificar na figura seguinte, através das operações de cálculo representadas (no canto

inferior direito), estas variáveis são directamente incorporadas no cálculo do tempo de duração das sub-rotinas geradoras de pulsos, impondo o seu valor, encaminhado depois para o VI designado *Wait* (relógio) colocado no interior do mesmo ciclo *While Loop*.

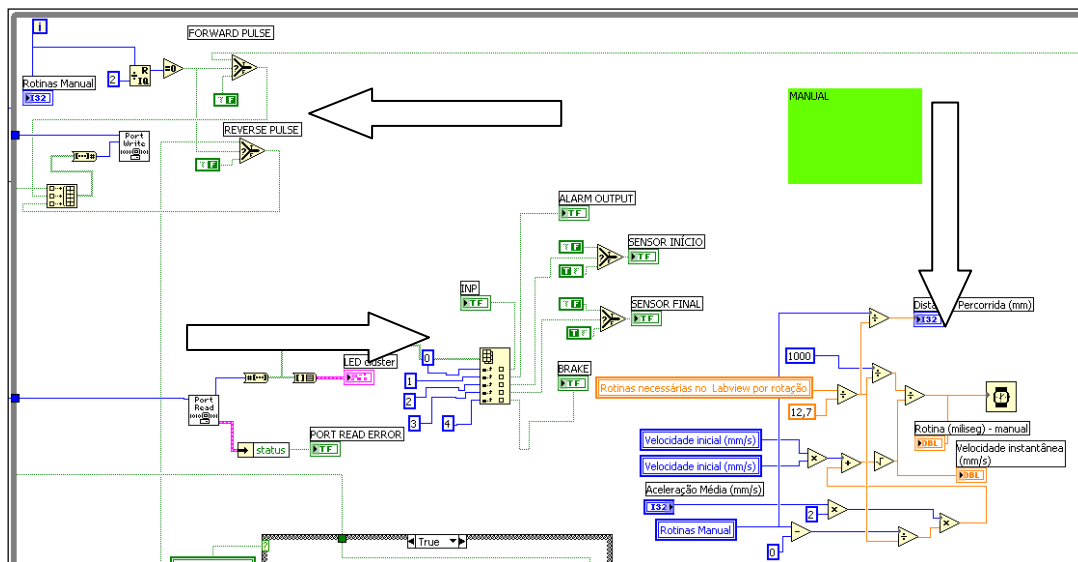


Figura 5.3: Sub-rotina elementar de saída de pulsos (“While Loop”), no modo manual. Diagrama de Blocos do Labview.

Na figura anterior estão também representadas (no canto superior direito) as operações de cálculo conducentes à saída de pulsos lógicos, endereçados para a *DAQ* através do *VI Port Read*. Enquanto os sinais de comando são escritos no porto de *output*, o programa lê simultaneamente os sinais no porto de *input*, no nosso caso ele lê os sinais dos fins de curso mecânicos, o *Alarm output*, a posição cumprida (*INP*) e o sinal do travão (*BKIR*). A programação da leitura dos sinais do porto de entrada está também explanada na figura anterior (ao centro).

Na execução deste programa foram implementados vários *VI*s pré-definidos, dos quais já destacámos os *VI*s que representam as funcionalidades da *DAQ* (leitura e escrita de dados), mas também foram utilizadas funções matemáticas, funções de controlo booleano (*Início* e *Paragem*) e de seleção (*Ring*), como por exemplo o instrumento de seleção do modo de funcionamento do servomotor. Foram também utilizadas as funções de indicadores, e dispositivos de visualização do estado de execução do programa, tipo *Leds*.

No *Labview* a sequência de execução das tarefas do programa é determinada pelo fluxo de dados representado no Diagrama de Blocos assim, a ativação da função de cada um dos

instrumentos virtuais *VI*s utilizados é determinada pelas ligações que se estabeleceram entre eles.

Foram criadas rotinas compostas por diversas sub-rotinas que se processam em simultâneo e paralelo dentro dum mesmo bloco. Um mesmo *VI* (ou bloco), pode ser executado de forma independente ou incorporado noutra *VI* de nível superior (mais abrangente).

Simultaneamente com a programação das operações de cálculo dos pulsos, e da sua cadência, outra parte muito importante da programação no *Labview* está direcionada para o sistema de segurança do programa em caso de ocorrência de qualquer anomalia. Em todas as rotinas e sub-rotinas são lidos os sinais do porto de entrada da *DAQ*, designadamente o *Alarm Output*, *Paragem*, *Sensor de Início* e *Sensor Final*, e articulado o sistema de paragem imediata do programa no caso dum destes ter valor lógico 1. O *Alarm Output* interrompe o valor do *Run* através da quebra existente numa sub-rotina, esta só é retomada após a confirmação pelo operador de todos componentes mecânicos (na figura seguinte, no canto superior direito). A ativação de qualquer um dos outros sinais de entrada origina a saída imediata da sub-rotina e das rotinas subsequentes, até parar o programa. Como esta saídas das rotinas se dá em milissegundos, a operação do sistema mecatrónico é assim parada de imediato, garantindo condições de segurança.

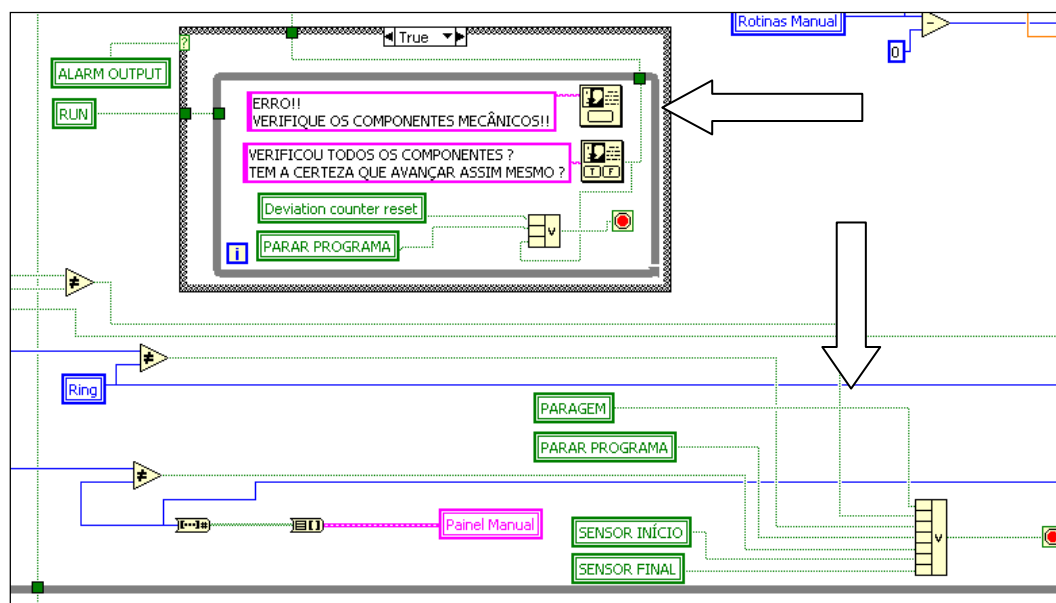


Figura 5.4: Sistema de segurança implementado em todas as sub-rotinas e rotinas.

Diagrama de Blocos do Labview.

5.2 – Controlo Semiautomático e Automático do Sistema

Tendo por objetivo a automatização, pretendeu-se que o sistema fosse capaz de variar os percursos percorridos (posicionamento e de trabalho), o tempo no estágio nos patamares (inicial e final), a velocidade e a aceleração (constante) sempre que o utilizador assim o indicasse no painel frontal (interface) do sistema. Como anteriormente vimos no capítulo do projecto mecânico e electrónico, estes três parâmetros do movimento do servomotor variam de acordo com as instruções (sinais) gerados por intermédio da Placa de Aquisição de dados *DAQ*, comandada pela programação do *Labview*.

Foram definidos no painel frontal do Labview três modos de operação, o modo manual, o modo semiautomático e o modo automático.

Previu-se que o modo manual fosse apenas utilizado em situação de recurso, mais concretamente, em situações de ocorrência de erros do servomotor (*Alarm Output*) para reposicionar a mesa, ou, em situações de teste do sistema, por exemplo durante a sua manutenção. No modo manual, o operador comanda o *Forward Pulse*, o *Reverse Pulse* e o *Run*. Como anteriormente referido, o operador define também a velocidade e aceleração do servomotor.

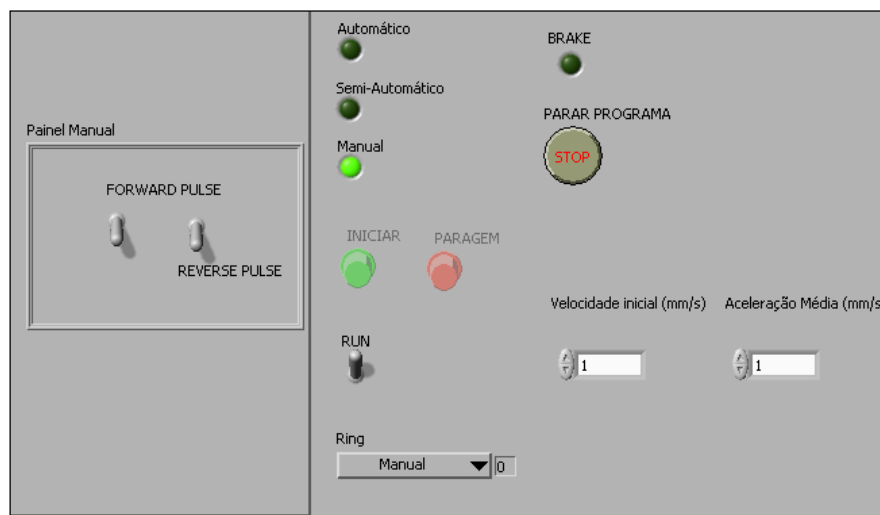


Figura 5.5: Visualização da seleção da velocidade e aceleração, comum a todos os modos de operação.

Painel Frontal do Labview.

No modo de operação manual, os sinais lidos pelos fins de curso eletromecânicos são também tidos em consideração, por questões de segurança. Existem portanto e apenas os limites posicionais extremos ao comando da operação (mantém-se como é óbvio o alarme por

excesso de torque e o INP). Na figura-se seguinte representa-se o aspeto geral da sub-rotina do Labview para o modo de operação manual. Os principais componentes já foram referidos anteriormente neste capítulo.

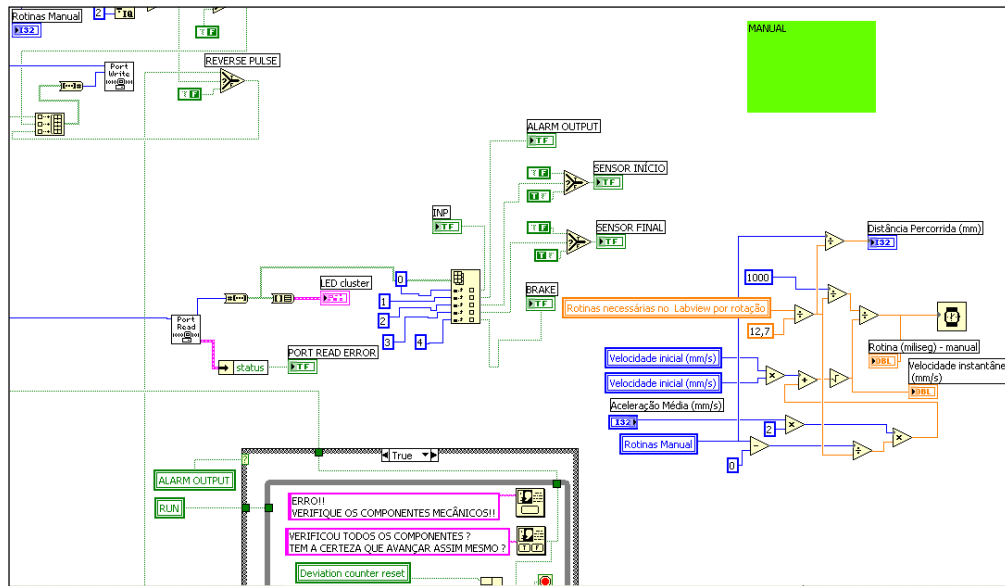


Figura 5.6: Sub-rotina no Labview para o modo de operação Manual. Diagrama de Blocos do Labview.

No modo semiautomático, o operador define no início de cada operação os diversos parâmetros da mesma, tais como, o percurso de posicionamento, o tempo de estágio inicial, o percurso da operação e de retorno e o tempo de estágio no patamar. A velocidade e a aceleração são também definidas.

No modo de operação automático, para além dos parâmetros definidos no modo semiautomático estabelece-se também o número de ciclos, ou seja o número de vezes que a operação é executada sem parar o processo.

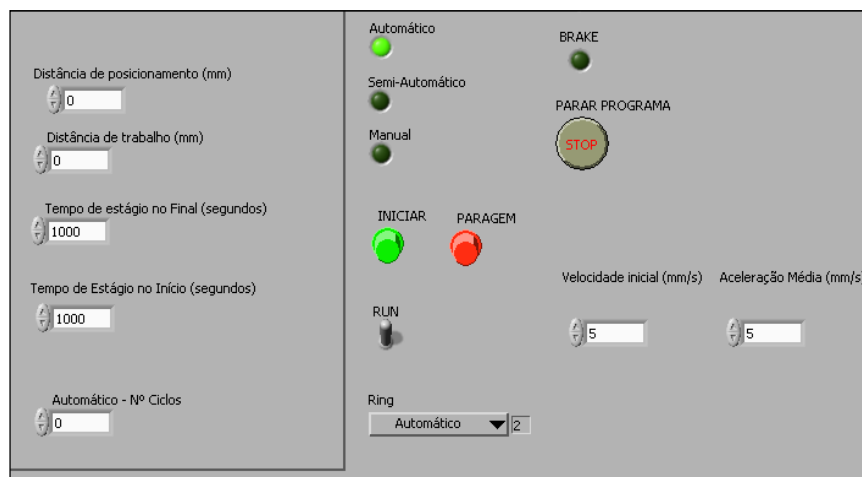


Figura 5.7: Visualização dos comandos no modo de operação Automático. Painel Frontal do Labview.

Para a seleção e comando dos modos de operação automático e semiautomático, foram criadas rotinas e sub-rotinas na programação do *Labview*, assentes numa estrutura de controlo *Boleano e alfanumérico* de transferência de dados, usando estruturas de *case* (“*Case Structure*”) e estruturas de sequencialização (“*Sequence Structure*”) do tipo de filme (com quadros sequenciais) e finalmente a unidade básica de programação no *Labview* “*While Loop*”, que efetuam o procedimento de acordo com os parâmetros pré-estabelecidos, designadamente a velocidade, percursos, ciclos de repetição e tempos de paragem.

A arquitetura de programação estabelecida nos modos semiautomático e automático foi a que se explica nos parágrafos seguintes.

Começo com o quadro de “*inicialização*”, em que quando a correspondente rotina é executada, a mesa/guia linear recua (reverse pulse) até intercetar o fim de curso inicial.

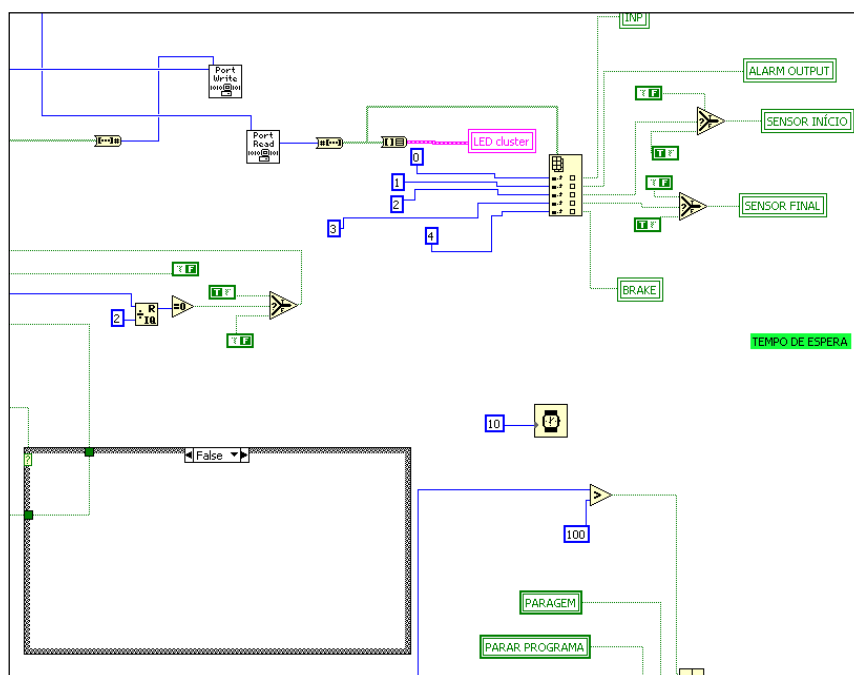


Figura 5.8: Sub-rotina “tempo de espera”. Diagrama de blocos do Labview.

Uma vez intercetado o fim de curso inicial, o sistema entra numa rotina designada “*tempo de espera*”, que promove o afastamento suficiente (através de pulsos *Foward Pulse*) para a desativação do fim de curso inicial, este quadro tem um tempo de duração de 1 segundo (100 rotinas multiplicados por 10 milissegundos de duração cada uma). Houve necessidade de introdução deste quadro porque, devido à elevada velocidade de processamento das rotinas no *Labview*, a duração da desativação do fim de curso inicial, e o

correspondente atraso na passagem ao valor lógico 0, provocava a saída de segurança das rotinas dos quadros sequenciais subsequentes.

Posteriormente entra-se na rotina de “*posicionamento*”, em que o sistema avança o curso correspondente à distância indicada pelo operador para o posicionamento (*mm*). Nesta rotina calculam-se e transmitem-se os pulsos correspondentes ao *Foward Pulse*.

Concluído o posicionamento, o sistema entra numa rotina que promove o tempo de espera no designado *patamar recuado*. A duração desse tempo de espera é controlada pelo operador (*s*) no Painel Frontal. Nesta rotina, lêem-se os sinais de entrada, mas não são transmitidos sinais de saída *Foward Pulse* ou *Reverse Pulse*.

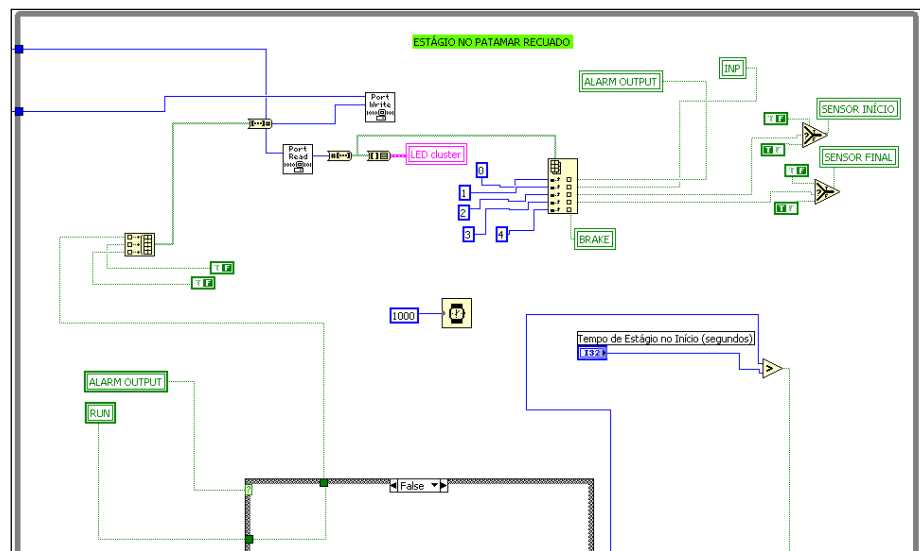


Figura 5.9: Sub-rotina “*estágio no patamar recuado*”. Diagrama de blocos do Labview.

Terminado o tempo de espera no patamar inicial, o sistema mecatrónico inicia a seu avanço de trabalho, correspondente à distância de trabalho (*mm*) indicada pelo operador no Painel Frontal. Neste quadro sequencial do Labview promovem-se os pulsos correspondentes ao *Foward Pulse* e lêem-se todos os sinais de entrada. O aspeto desta rotina do Labview é apresentado de seguida.

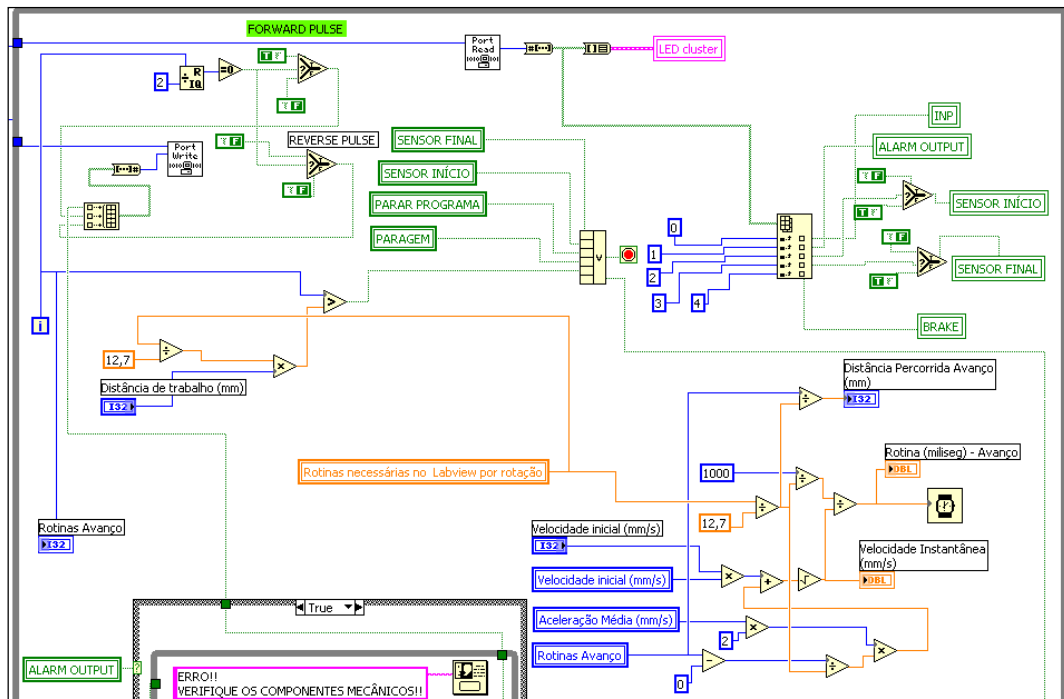


Figura 5.10: Sub-rotina "Forward Pulse". Diagrama de blocos do Labview.

Terminado o avanço, o sistema mecatrónico entra novamente num quadro de tempo de espera no patamar avançado. A definição deste tempo de espera é feita pelo operador e é independente do tempo de espera no patamar recuado.

Esgotado o tempo de espera no patamar avançado, o sistema inicia novamente o movimento desta feita, de recuo, através duma rotina que promove pulsos no sentido contrário, *Reverse Pulse*. O quadro desenvolvido no Labview é idêntico ao já apresentado anteriormente para o movimento de avanço. A distância de recuo é a mesma indicada para a distância de avanço, ou seja, a distância de trabalho (*mm*).

Concluído o recuo, a mesa/guia linear chega novamente ao patamar de trabalho recuado, reiniciando-se novamente o ciclo de trabalho no caso da operação no modo automático, este processo repete-se o número de ciclos definido pelo operador no Painel Frontal; ou indicando que a operação está concluída no caso do modo semiautomático, tal como ilustrado na figura seguinte.

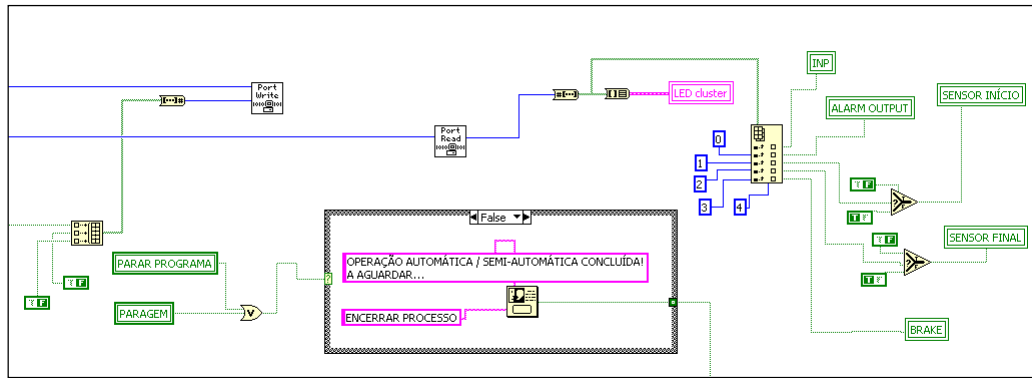


Figura 5.11: Sub-rotina “Operação concluída”. Diagrama de blocos do Labview.

Tal como já anteriormente ressaltado, todas as sub-rotinas têm um instrumento “Port Read” que lê constantemente os sinais lógicos nas entradas correspondentes à ativação dos fins-de-curso “Sensor de início” e “Sensor final”, e os controles “Paragem” (paragem de emergência na operação) e “Parar Programa” (parar o programa na programação) instalados no *Painel Frontal* do *Labview*. Se algum destes sinais tiver valor lógico verdadeiro, as sub-rotinas são imediatamente galgadas e a emissão de pulsos para o servomotor cessa. Neste caso a operação de reposicionamento do servomotor apenas pode ser executada no modo manual.

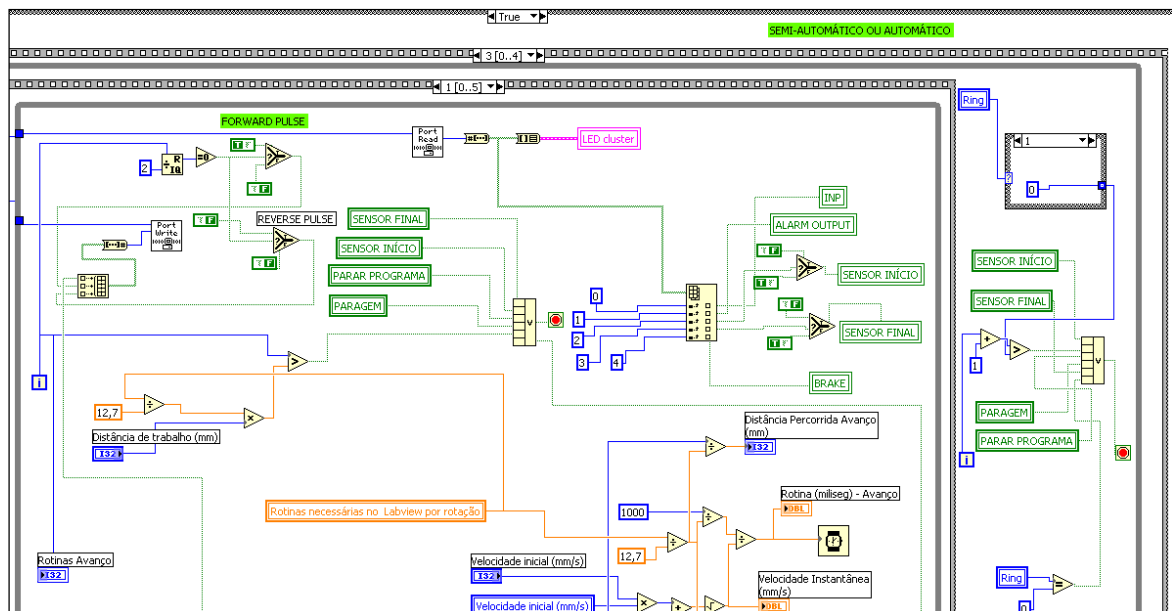


Figura 5.12: Visualização das “Sequence Structures” nos modos Automático e Semiautomático.

Diagrama de Blocos do Labview.

A sub-rotina de introdução dos sinais de comando dos pulsos, os sinais que seguem do CPU para o Controlador, é idêntica em todos os modos de operação, esta utiliza uma estrutura *while loop* e numa fórmula de cálculo e comparação, ligada aos incrementos do ciclo de *while loop*. Resultam desta forma sinais intermitentes para o exterior enviados para a *DAQ*, através do *Port Write*, com uma frequência dependente do *timer* também instalado no interior da sub-rotina *while loop*.

Toda a operação do modo semiautomático e automático é balizada por sensores de fins de curso mecânicos. Uma vez intercetados estes fins de curso o programa é parado e a continuidade da operação ou o reposicionamento têm que ser feitos no modo manual. A paragem do programa Labview funciona como um sistema de segurança do equipamento, uma vez que desativa os sinais de output.

Todo o programa é enquadrado num *While Loop* externo, que mantém o código executado continuamente.

Para além da automatização da mesa/guia linear, importa também agora debruçarmos sobre as operações de cálculo necessárias introduzir no *Labview* para possibilitar o controlo efetivo dos parâmetros cinemáticos do movimento. Essas operações estão representadas na figura seguinte.

No canto inferior esquerdo verifica-se a implementação da distância de posicionamento, relacionando-a com número de rotinas necessárias para induzir uma rotação no servomotor, e também com o passo do parafuso sem fim que transmite o movimento ao patim. O passo do parafuso sem fim é de $\frac{1}{2}$ polegada, ou seja, 12,7 mm.

No canto inferior direito está explícito o procedimento adotado para o controlo da velocidade instantânea do sistema, com base nos pressupostos do parágrafo anterior e, fazendo-a depender dos valores da velocidade inicial e aceleração introduzidos pelo operador no Painel Frontal. Este cálculo está fundamentado nas equações da cinemática das partículas, com aceleração constante, e em cada rotina executada, em função da evolução da distância de posicionamento, conduz ao valor do tempo de duração dessa rotina a finalizar.

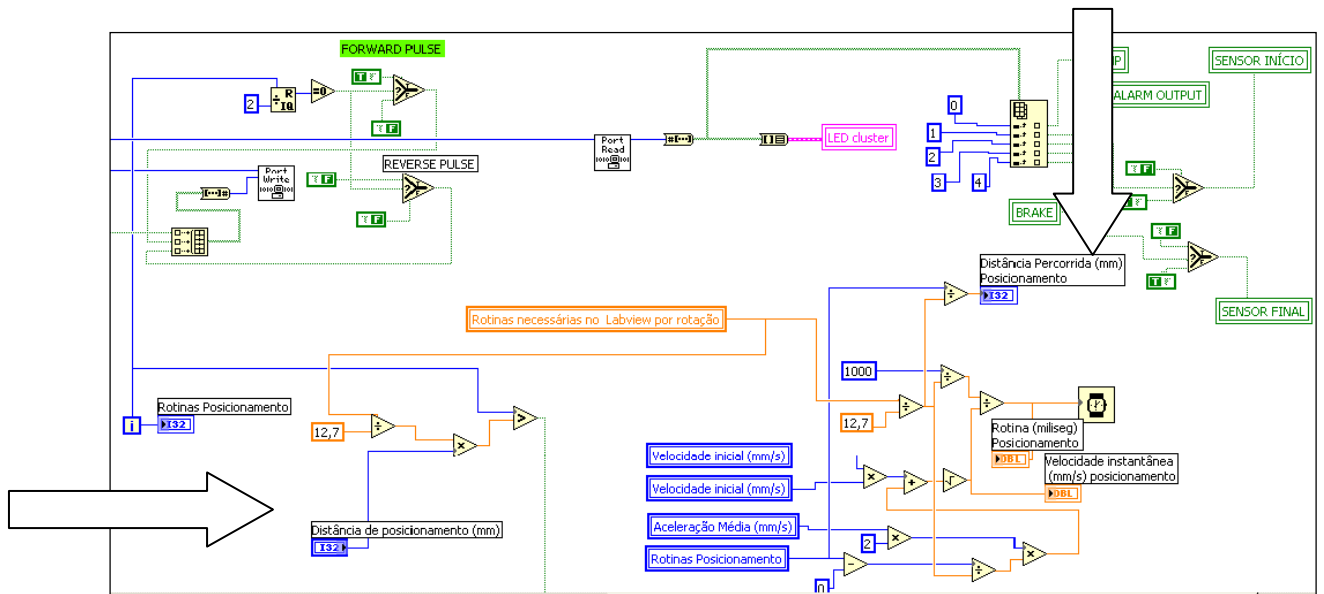


Figura 5.13: Controlo do posicionamento e da velocidade instantânea do sistema.

Diagrama de Blocos do Labview.

Relativamente à determinação do número de rotinas necessárias executar no Labview por rotação, cuja grandeza influi na implementação da distância de posicionamento e na velocidade instantânea do sistema (dependente da duração de cada rotina), o seu cálculo no desenvolvido no Labview apresenta-se na figura seguinte. Este está fundamentado num requisito do servo motor, isto é, necessita receber 4096 pulsos para fazer uma rotação completa; e no conceito de Electronic Gear (= $G1/G2$) da *Servo Drive*. Esta engrenagem electrónica faz a desmultiplicação dos pulsos produzidos no Labview pelo valor de Cn-24 ($G1$) / Cn-25 ($G2$), obrigando a mais ou menos rotinas executadas no Labview para uma rotação do servomotor. O cálculo da *Electronic Gear* e a sua importância serão abordados no Capítulo 6 deste relatório.

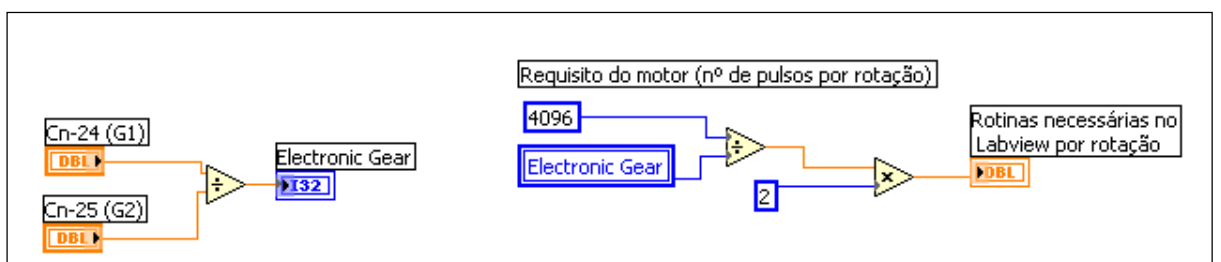


Figura 5.14: Cálculo do *Electronic Gear* e do Número de rotinas necessárias por rotação.

Diagrama de Blocos do Labview

5.3. Teste das Entradas e Saídas do Sistema

Recorrendo à interface com a placa de aquisição de dados utilizada, a *PCI DAQ 1200*, após terem sido programadas as entradas e saídas digitais no *software Labview*, pôde-se rapidamente testar as soluções de programação que foram sendo desenvolvidas. Através da leitura de um multímetro ligado às saídas do conector da DAQ testaram-se as transmissões bit-a-bit e monitorizou-se o padrão digital da modulação por largura de pulso programada (PWM).

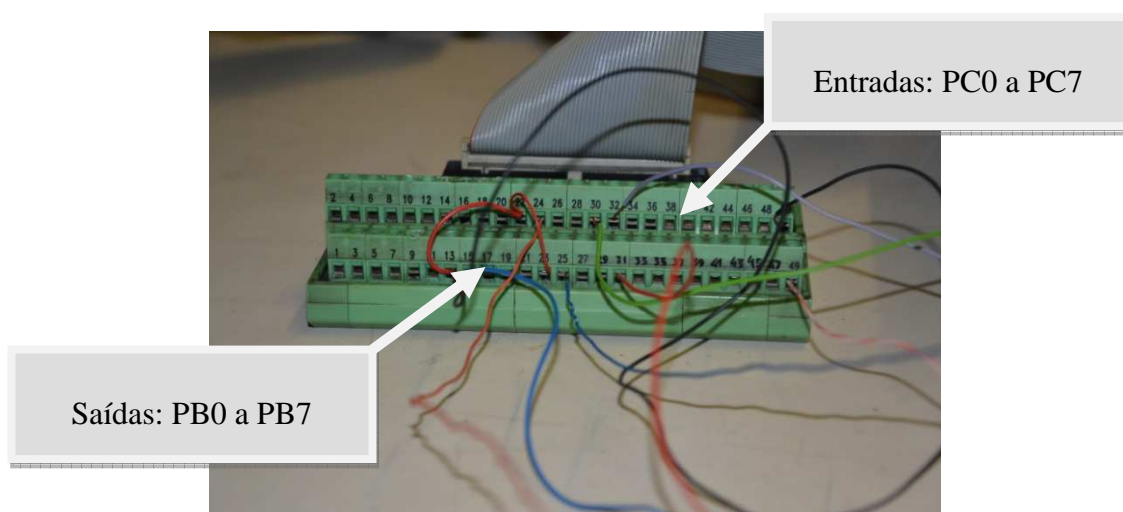


Figura 5.15: Conector utilizado. Ligação das entradas e saídas da DAQ ao cabo de comando CN1.

Através da funcionalidade *Probe* do Labview, que permite saber a cada instante quais os dados que percorrem uma determinada linha de fluxo na direção da *Port write* binária, no nosso caso em lógica binária, confrontaram-se esses resultados com os níveis lógicos de tensão efetiva de saída nos bits do conector, verificando-se que a DAQ estava a funcionar corretamente.

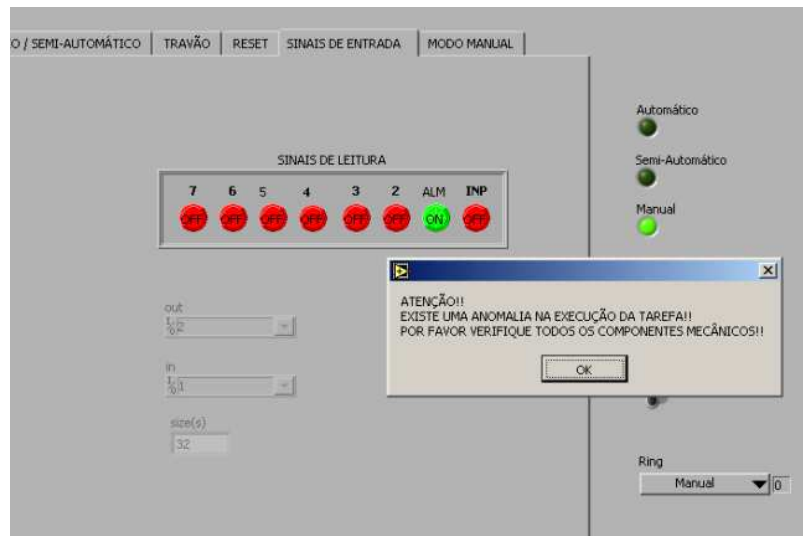


Figura 5.16: Teste da entrada 2 do porto B, com a injeção de 5 V no pino correspondente.

Diagrama de Blocos do Labview.

Este método expedito, utilizado também nas restantes entradas do porto *B*, permitiu fazer o teste inicial da componente de *software* e *hardware* informático do sistema, ainda sem a componente do condicionamento do circuito electrónico utilizado e sem o risco de se provocar qualquer tipo de entropia ao sistema mecânico.

Com o teste das entradas e saídas do sistema, pretendeu-se também verificar a manutenção do estado dos sinais digitais (oscilações), verificaram-se as ligações eletromecânicas (por aperto), assegurando que não se verificam falsas mudanças do estado programado e que os seus valores se mantêm por uma certa duração, conforme a largura de pulso programada.

Capítulo 6 – Implementação Física e Análise dos Resultados

6.1 – Descrição do protótipo

Uma vez desenvolvida a programação do Labview, e testadas as entradas e saídas digitais que se pretendeu utilizar, procedeu-se à implementação física de todos os componentes do sistema mecatrónico desenhado para o controlo da mesa / guia linear.

O sistema mecânico tal como descrito nos capítulos anteriores, é, sucintamente, constituído por uma mesa / guia linear com um patim acoplado, existente no DEM. O patim desliza sobre dois carris longitudinais paralelos, e o movimento linear é permitido apenas numa direcção, sendo imposto por uma engrenagem do tipo de parafuso sem-fim, cujo veio motor (o parafuso sem fim) é apoiado nas suas duas extremidades. Numa das extremidades este veio é solidarizado com o movimento do eixo do motor por meio de uma união elástica concêntrica.

Para a construção física do sistema de *hardware* e de electrónica, também já descrito nos capítulos anteriores, utilizaram-se os equipamentos existentes no Laboratório de Automação do DEM, ou seja, o servomotor da marca OMRON, referência R88M-EU 400 30V – BS1 e o respectivo controlador (servo drive) com a referência R88D – UEP 12V, assim como uma placa electrónica experimental (“bread board”) e uma fonte alimentação eléctrica CC com saídas de 24V e 5V. Os restantes pequenos componentes electrónicos, tais como transístores, resistências, leds, díodos, fins de curso, foram adquiridos numa loja da especialidade, tendo sido o seu custo pouco representativo.

Os sensores eletromecânicos fins de curso, o de início e o de fim, foram acoplados ao chassi da mesa/guia linear, por intermédio dos quatro patins de um único carril, individuais, existentes, dois mais dois, que se solidarizaram por intermédio de uma placa fabricada no laboratório, constituindo-se dois blocos limites, que não podem ser ultrapassados pela mesa/guia. A vantagem deste sistema é a possibilidade de posicionamento dos sensores fins de curso em diferentes posições, podendo deste modo estudar o comportamento do sistema com diferentes cursos limites.

Na figura seguinte pode-se visualizar o aspeto final do protótipo desenvolvido montado para este trabalho, preparado para a execução dos ensaios de validação.

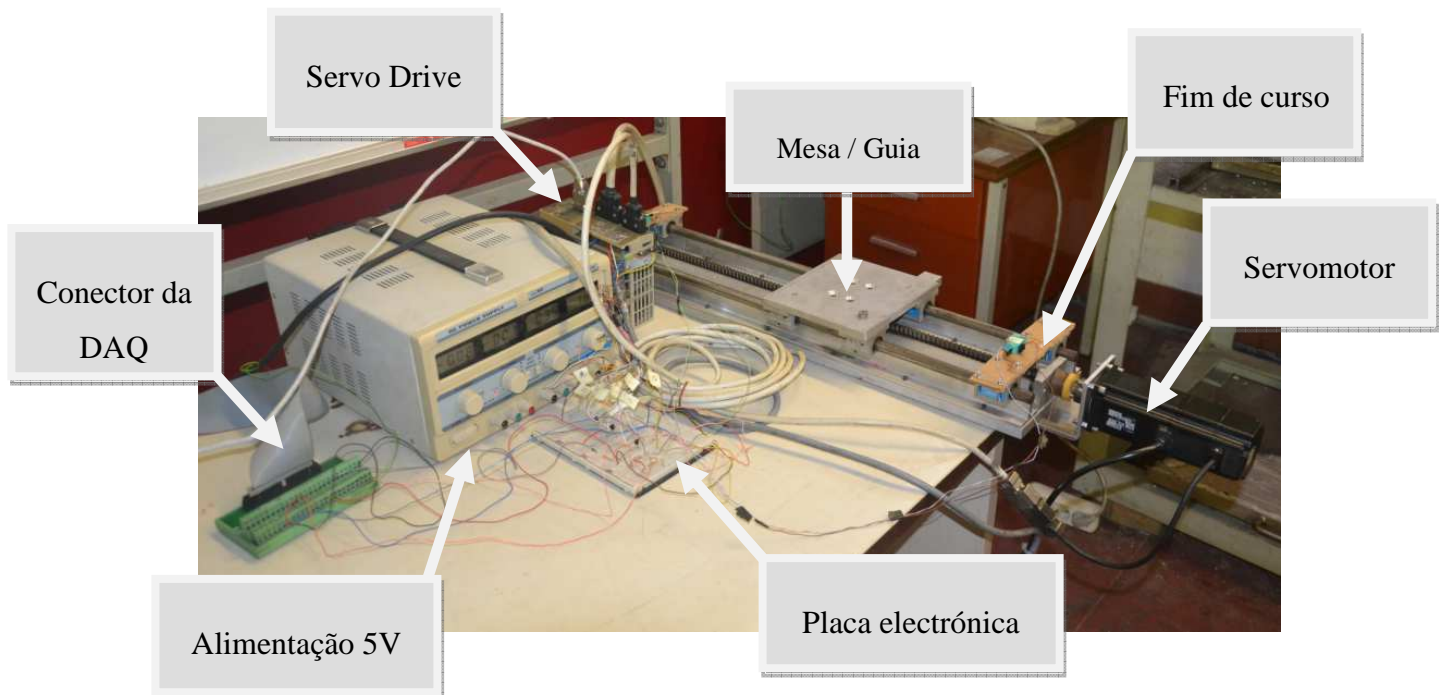


Figura 6.1: Aspeto do sistema mecatrónico desenvolvido. Indicação dos seus principais componentes.

A velocidade do servomotor é comandada pelo operador no Painel de Controlo, sendo que depois é implementada no sistema pelo programa Labview através de um temporizador, que não deixa terminar uma rotina (ciclo de tarefas) sem que um intervalo de tempo se esgote.

Para a alteração da *Electronic Gear* gravada no *Servo Drive* recorreu-se à utilização da mesma fonte externa (CPU) e à ligação através de cabo de série RS232, para alterar os parâmetros de comando do controlador (“*User Parameters*”) usados para o cálculo da engrenagem electrónica (“*Electronic Gear*”), tema já abordado no subcapítulo 2.2. Relembremos que a velocidade de rotação do servo motor é comandada por uma cadência de pulsos, que resulta da multiplicação dos pulsos provenientes dos ciclos do *Labview* (através do *hardware*) pela *Electronic Gear* calculada, no seu controlador (*servo drive*).

Por defeito a *Electronic Gear* é igual a 4, $G1/G2 = 4$, ou seja, por cada pulso fornecido ao controlador através da DAQ, o controlador multiplica essa frequência por 4. Por sua vez o servomotor para fazer uma rotação completa, tem que receber 4,096 pulsos. O controlador (“*servo drive*”) tem implícito um conjunto de *User Parameters* (dados de base), com base nos quais executa as suas funções.

Através do software *OMNUC (U-Series Computer Monitor Software)*, desenvolvido pela *OMRON* para a comunicação com o controlado (“*servo drive*”), para o comando dos *Setup Parameters* e dos *User Parameters* dos controladores da Série *UEP*, como o utilizado neste trabalho, alteraram-se os parâmetros correspondentes ao *G1* (parâmetro *Cn-24*) e ao *G2* (parâmetro *Cn-25*) resultando que a relação destes passou de 4 (gravada de fábrica, à partida) para 40, $Cn-24 = 40$, $Cn-25 = 1$.



Figura 6.2: Interface para o controlo dos User Parameters *Cn-24* e *Cn-25*. Visualização do *UMNUC*.

Através desta alteração conseguiu-se elevar a velocidade do servomotor para a gama de valores exigidos para operação da mesa / guia linear.

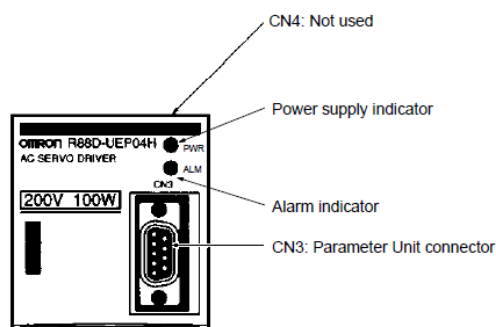


Figura 6.3: CN3, ligação do cabo de série RS232 para comando dos “*User Parameters*” da Servo Drive.

6.2 – Ensaio de validação

Foram realizados testes com o sistema mecatrónico montado da forma descrita, pretendeu-se que a única diferença para a operação real fosse o facto do patim da mesa/guia linear não estar efectivamente carregado ou acoplado a uma ferramenta.

Com a realização dos ensaios de validação, procurou-se verificar o comportamento dinâmico do sistema, e se o mesmo está de acordo com o calculado,

assim como identificar todos os eventuais problemas relacionados com a variabilidade dos parâmetros de comando introduzidos pelo operador. Pretendeu-se também verificar a eficiência das diferentes sub-rotinas do programa *Labview* desenvolvido e o comportamento do próprio sistema mecânico.

Procurou-se desta forma avaliar o desempenho do sistema mecatrónico completo, mesmo no caso de um eventual erro de operação de uma das partes que o constituem [5].

Os parâmetros cinemáticos do movimento da mesa / guia que controlados foram os seguintes:

- Deslocamento da mesa / guia linear, medido em *mm*.
- Velocidade, medida em *mm/s*;
- Aceleração, medida em *mm²/s*.

O comportamento dinâmico da mesa/guia foi previsto partindo dos dados de base do sistema mecatrónico, que são os requisitos e parâmetros do *software (Labview)*, do *hardware* (servo e drive) e as características dos componentes mecânicos.

O cálculo do comportamento dinâmico, foi então previsto numa folha de cálculo *Excel*, sistematizada da seguinte forma:

- **Requisitos:**

Requisito do Motor (fixo) =	4096	pulsos /rotação	* nº de pulsos já desmultiplicados / por rotação
-----------------------------	------	-----------------	--

Passo da rosca do parafuso sem-fim: 1/2 polegada	* igual a 12,700 mm
--	---------------------

- **Parâmetros da Servo Drive definidos pelo utilizador:**

Cn-24, G1 =	4	(User	(A)
Cn-25, G2 =	1	Parameters)	(B)

- Electronic Gear calculada: (implementado na servo drive, em função de Cn-24 e Cn-25):

G1/G2 =	4 (Electronic Gear)	(C) = (A) / (B)
---------	---------------------	-----------------

- Cálculos implementados no Software Labview:

Pulsos Efectivos nas saídas da Placa (DAQ)	1024 por rotação	(D) = 4096 / (C)
Rotinas no Labview	2048 por rotação	(E) = (D) x 2

- Cálculos auxiliares:

Avanço (mm / rotina do Labview):	0,006	mm/rotina	(F) = 12,700 / (E)
Nº de Rotinas no Labview para avançar 1mm:	161,260	rotinas	(G) = (E) / 12,700
Duração da Rotina (milissegundos):	6,201	milissegundos	(I) = 1000 / (G)

- Parâmetros dinâmicos controlados pelo utilizador:

Aceleração média =	mm ² /s
Velocidade inicial =	mm/s
Distância =	Mm

Uma vez definidos os parâmetros cinemáticos pelo operador, as distâncias, a velocidade inicial e a aceleração média, deduziu-se o comportamento dinâmico da mesa/guia linear por analogia com a análise cinemática do diagrama de corpo livre, com um dos movimentos possíveis, ou seja, o movimento de translação do conjunto segundo o eixo XX, sobre os carris.

Pretendeu-se com este estudo a verificação e confirmação experimental dos deslocamentos, velocidades e acelerações do mecanismo calculadas, assim como, determinar os patamares de velocidades e acelerações máximas admissíveis que o sistema pode ter, para que este adopte um comportamento equilibrado e seguro. As velocidades e acelerações máximas limites estão directamente relacionadas com o tempo de duração da rotina arredonda, qualquer que seja a relação entre a velocidade e a aceleração, a duração da rotina calculada, não pode ser inferior a 0,5 milissegundos, sob pena do Labview a entender como infinitesimalmente pequena, resultando um

comportamento descontrolado do servomotor. A velocidade mínima pode ser bastante reduzida, dependendo da resolução do servomotor (4096 pulsos por rotação completa) e do tempo de duração máximo das rotinas.

Como condição primeira condicionou-se a translação máxima da mesa/guia sobre os patins ao afastamento físico existente entre os dois fins de curso, descontado do comprimento do patim, fixou-se esta distância em 400mm

Simultaneamente impuseram-se também as condições de velocidade inicial e aceleração média introduzidas no sistema.

Um movimento deste tipo, com aceleração constante, rege-se pelas seguintes equações da cinemática de partículas:

$$v = v_i + a_{constante} \times t$$

$$s = s_i + v_i \times t + \frac{1}{2} a_{constante} \times t^2$$

E para determinar a velocidade em função da posição:

$$v^2 = v_i^2 + 2 \times a_{constante} \times (s - s_1)$$

Esta última fórmula foi adaptada no *Labview*, partindo dos cálculos auxiliares indicados anteriormente, isto é, transformou-se a posição S_i em número de rotinas, a partir desta adaptação, conseguiu-se determinar a velocidade instantânea em cada ponto (em cada posição) em função do número de rotinas executadas no *Labview*, da velocidade inicial e da aceleração média definidas.

Por sua vez, fazendo o cálculo inverso, conseguiu-se calcular a duração de cada rotina que o *Labview* precisa de realizar, para se conseguir uma determinada velocidade instantânea. Por influência da aceleração, a velocidade instantânea é variável e por conseguinte a duração de cada rotina do *Labview* varia também necessariamente.

Em termos de testes laboratoriais procurou-se determinar os valores de velocidade e aceleração que podemos introduzir no Painel de Controlo do *Labview* e por conseguinte induzir no acionador do sistema mecatrónico, para que esta ferramenta seja eficaz, isto é, para que a mesa/guia linear se desloque dentro de dois cenários:

1º - A mesa se desloque com a maior rapidez possível, com equilíbrio e segurança.

2º - Para que a mesa se desloque com a maior precisão possível.

Como caso limite simulou-se também a introdução no Painel de Controlo de um percurso de operação demasiado longo, desta forma o patim intercetará necessariamente um dos fins-de-curso eletromecânicos. O fim de curso intercetado remeterá um sinal que uma vez entrado no *hardware* e assimilado pelo *software* obrigará à paragem do sistema. Podendo a partir daí ser operado unicamente no modo manual, até que o sensor fim de curso deixe de ser intercetado.

Durante a realização dos testes, a leitura do comportamento do sistema em cada simulação e ensaio realizado permitiu perceber erros nas diferentes componentes do sistema desenvolvido (software, electrónico e mecânico), levando a que fossem continuamente introduzidas melhorias no projecto mecatrónico. Este processo de melhoria contínua originou que melhorias fossem progressivamente introduzidas até que a avaliação do desempenho do sistema fosse satisfatória [5].

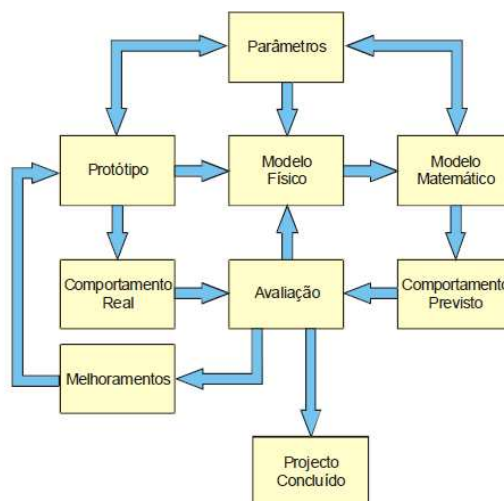


Figura 6.4: Dinâmica de melhoria do projecto de acordo com os princípios da mecatrónica.

Adaptado de [5]

6.3 – Análise dos resultados

Tal como referido no subcapítulo anterior, durante a realização dos testes procurou-se validar o comportamento dinâmico da mesa / guia linear, avaliando os valores dos três parâmetros cinemáticos já indicados, a distância percorrida (mm), a velocidade (mm/s) e a aceleração (mm^2/s), simultaneamente com o número de rotinas executadas no *Labview* e a sua duração (s). Com o recurso à funcionalidade do *Servo Drive*, *Electronic Gear* ($= G1/G2$), ensaiou-se a operação do sistema mecatrónico em dois modos, para perceber a variância no controlo dos parâmetros cinemáticos induzida pela desmultiplicação dos pulsos.

Primeiramente ensaiou-se o sistema com a *Electronic Gear* ($G1/G2$) = 40, testou-se depois a operação com diferentes valores de velocidade inicial, e com aceleração nula e diferentes valores de aceleração constante. No gráfico seguinte representam-se os resultados para o espectro de velocidades de 10 a 20 mm/s . Foram interpretados três tipos de valores para a velocidade média, a velocidade “calculada” (determinada a partir das equações da cinemática das partículas), a velocidade real (determinada no ensaio, recorrendo a um cronómetro e a uma distância estabelecida, através dos fins de curso) e a velocidade “arredondada” (calculada a partir das equações da cinemática das partículas, entrando no seu cálculo a duração da rotina arredondada para a unidade *milissegundos*).

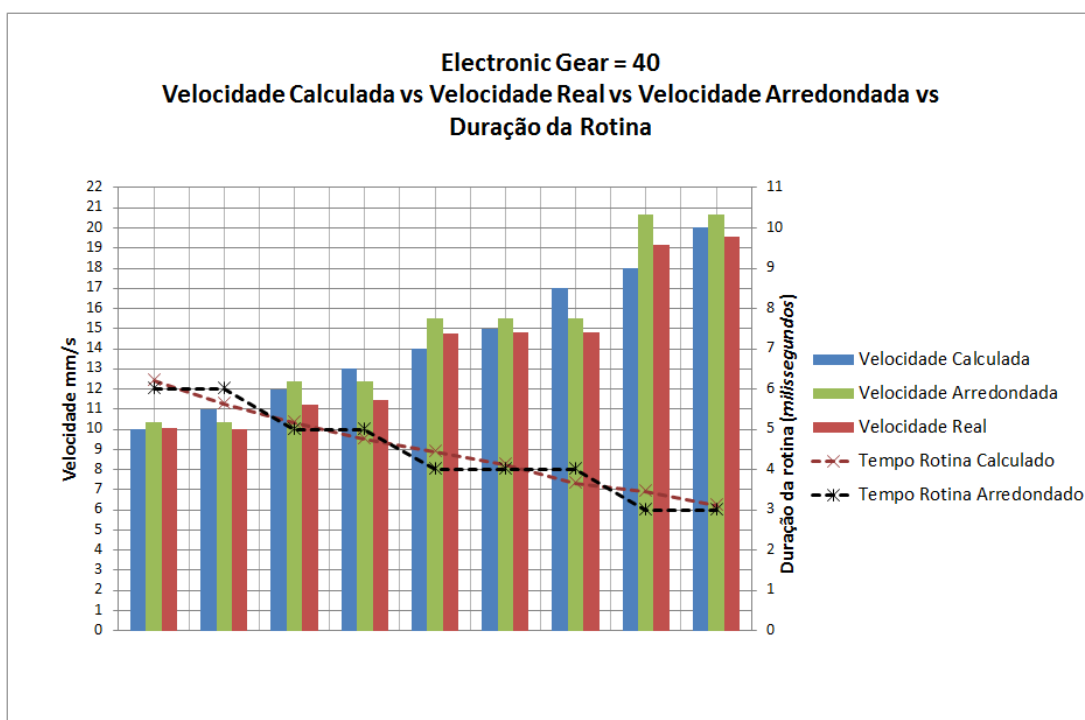


Figura 6.5: Electronic Gear = 40. Gráfico da relação das velocidades com a duração das rotinas.

Com a *Electronic Gear* = 40 e para o espectro de velocidades 10 a 20 mm/s pode-se verificar que a variação da média velocidade real, medida, acompanha de próximo a velocidade “arredondada” e a própria velocidade “calculada”. O tempo de duração da rotina no *Labview*, também representado no gráfico anterior, é inversamente proporcional às velocidades. Facilmente se compreende que para se conseguir maior velocidade de pulsos, menor terá que ser a duração de cada rotina.

Na análise do gráfico anterior constata-se também que a velocidade “arredondada” acompanha muito de próximo a velocidade real. Isto confirma o modelo de controlo e de cálculo implementado. Sendo que neste modelo, através do *Labview*, a menor divisão da unidade de tempo é o milissegundo, pelo que o tempo de duração da rotina é sempre arredondado para esta divisão. Adoptando este critério e calculando a velocidade arredondada determina-se o comportamento real do sistema.

Ensaiaando-se o mesmo sistema agora com a *Electronic Gear* = 4, ou seja, o factor de desmultiplicação de pulsos na Servo Drive passa de 40 para 4, constaram-se os resultados representados no gráfico em baixo. O valor das velocidades verificado nesta situação é menor, variando de 1 a 12 mm/s. Limitou-se à velocidade máxima de 12 mm/s porque para esta velocidade o tempo de duração calculado da rotina no *Labview* é de apenas 0,517 milissegundos, se a velocidade fosse de 13 mm/s o tempo de duração calculado da rotina passaria para 0,477 milissegundos. O tempo de duração calculado é contudo diferente do tempo de assimilado pelo *Labview* devido à sua precisão, assim, para o tempo de duração de 0,477 milissegundos o *Labview* assume um valor infinitesimalmente pequeno o que origina uma cadência de pulsos e correspondentemente uma velocidade muito elevada, desequilibrada, fazendo disparar o alarme do próprio servomotor, conforme se constatou laboratorialmente.

Por outro lado a valor da velocidade de 1 mm/s teve que ver apenas com a precisão da caixa de introdução das velocidades no Painel de Controlo do *Labview*, com alguma adaptação sistema admitiria velocidades ainda menores, aumentando aos tempos de duração das rotinas do programa.

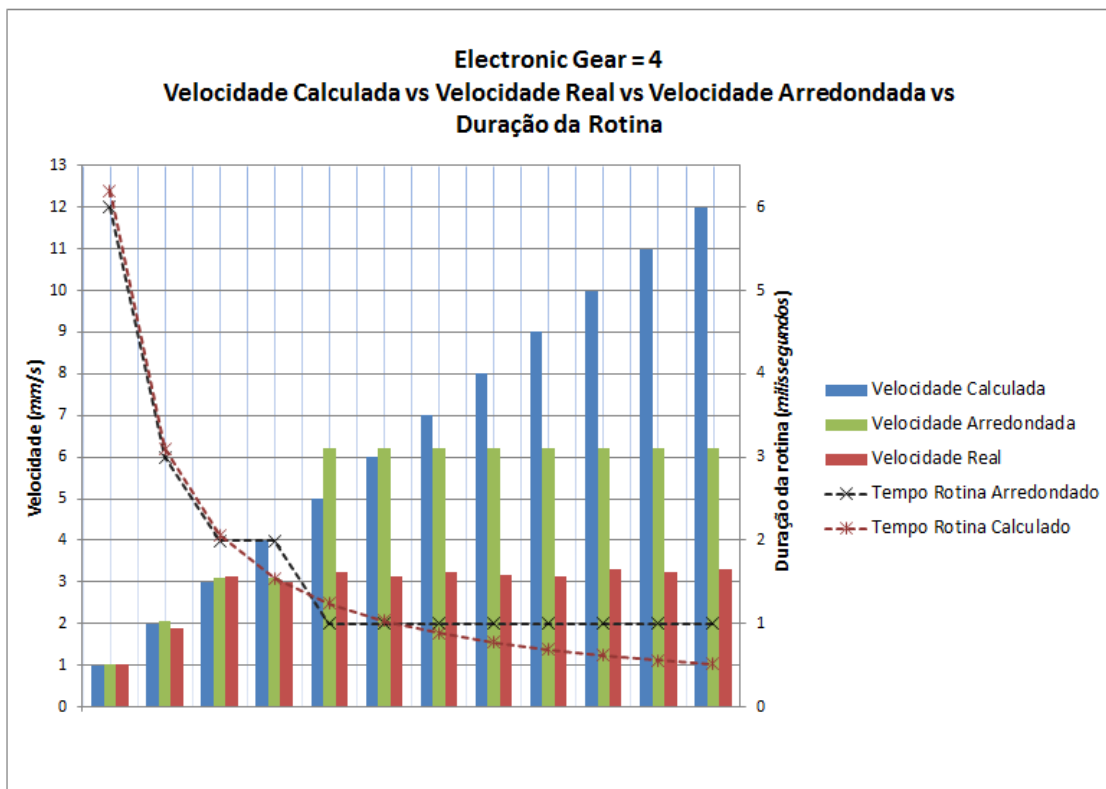


Figura 6.6: Electronic Gear = 4. Gráfico da relação das velocidades com a duração das rotinas.

Da análise do gráfico anterior verifica-se que o sistema mecatrónico, para a *Electro Gear* = 4 não tem um comportamento linear como se demonstrou ter para a *Electro Gear* = 40. Isto deve-se ao facto do tempo de duração das rotinas para valores de velocidade maiores, acima de 4 *mm/s*, e como já referido até 12 *mm/s*, se aproximar de um milissegundo, sendo que a partir desse valor a variação da duração das rotinas no Labview é da ordem das décimas do milissegundo, entrando nesta fase o tempo de duração da rotina num patamar de valor arredondado de um milissegundo. Significa portanto que a velocidade arredondada passaria a apresentar um valor constante de 6,201 *mm/s*. No entanto a velocidade real medida é cerca de metade desse valor, aproximadamente 3,2 *mm/s*. Da análise destes valores pode-se eventualmente formular que embora o tempo de duração arredondado lido pelo Labview seja de 1 milissegundo, a limitação da capacidade de processamento do próprio CPU garante apenas o processamento das rotinas com tempos de duração iguais ou superiores a 2 milissegundos, originado assim uma velocidade real condizente com a cadência de pulsos que daí resulta.

A diferença mais elementar entre o comportamento deste sistema mecatrónico adotando uma *Electronic Gear* = 4 e uma *Electronic Gear* = 40, verifica-se no número

de rotinas necessárias executar no Labview para que a mesa/guia linear avance o curso de 400 mm estabelecido. Como se pode verificar no gráfico seguinte para que o sistema avance os mesmos 400mm, no caso da Electronic Gear = 4 o número de rotinas necessárias é dez vezes mais do que caso da Electronic Gear = 40. Significa portanto que para a mesma velocidade média indicada, a duração de cada rotina no primeiro caso terá que 10 vezes menor.

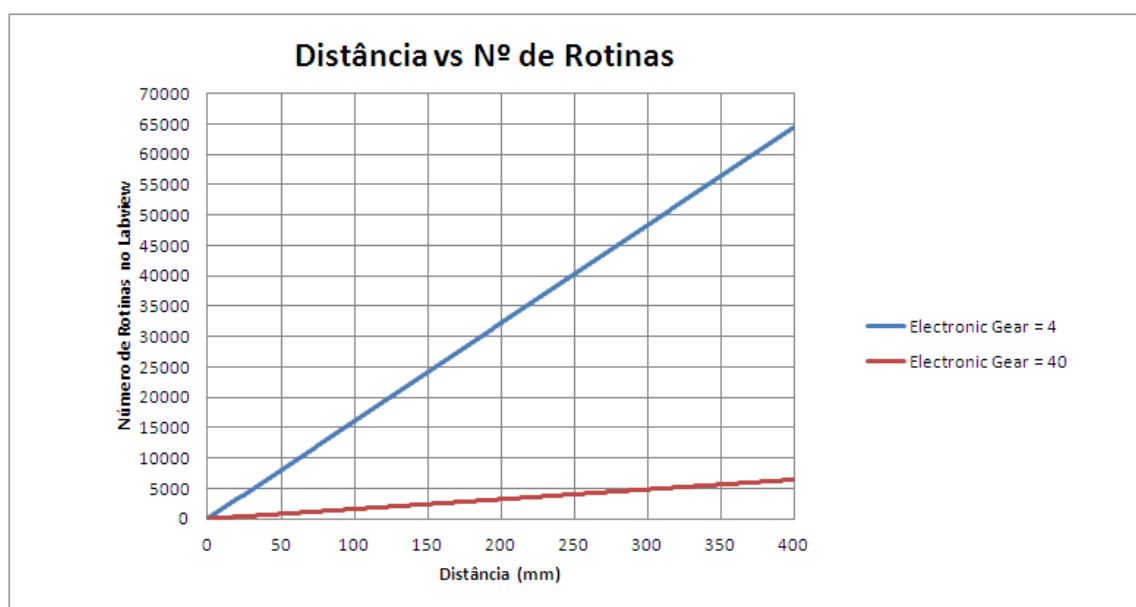


Figura 6.7: Electronic Gear = 4 e 40. Gráfico da relação da distância percorrida com o número de rotinas.

Isto torna o sistema mecatrónico mais dependente da capacidade do processador do CPU onde está instalado o Labview, pelo facto de se lerem mais amiúde os sinais de entrada, o sistema torna-se também mais sensível às suas pequenas variações.

No gráfico seguinte representam-se agora os resultados calculados das velocidades instantâneas correspondentes à rampa de aceleração de $1 \text{ mm}^2/\text{s}$ e velocidade inicial de 1 mm/s , para as Electronic Gear de 40 e de 4, e para um curso estabelecido de 200mm. Sendo mais fácil medir no ensaio a velocidade média conseguida nesse curso (recorrendo a cronómetro), representam-se também as curvas correspondentes à velocidade média acumulada calculada em cada ponto do curso, podendo-se confrontar no final com a velocidade média real medida, para a validação deste modelo.

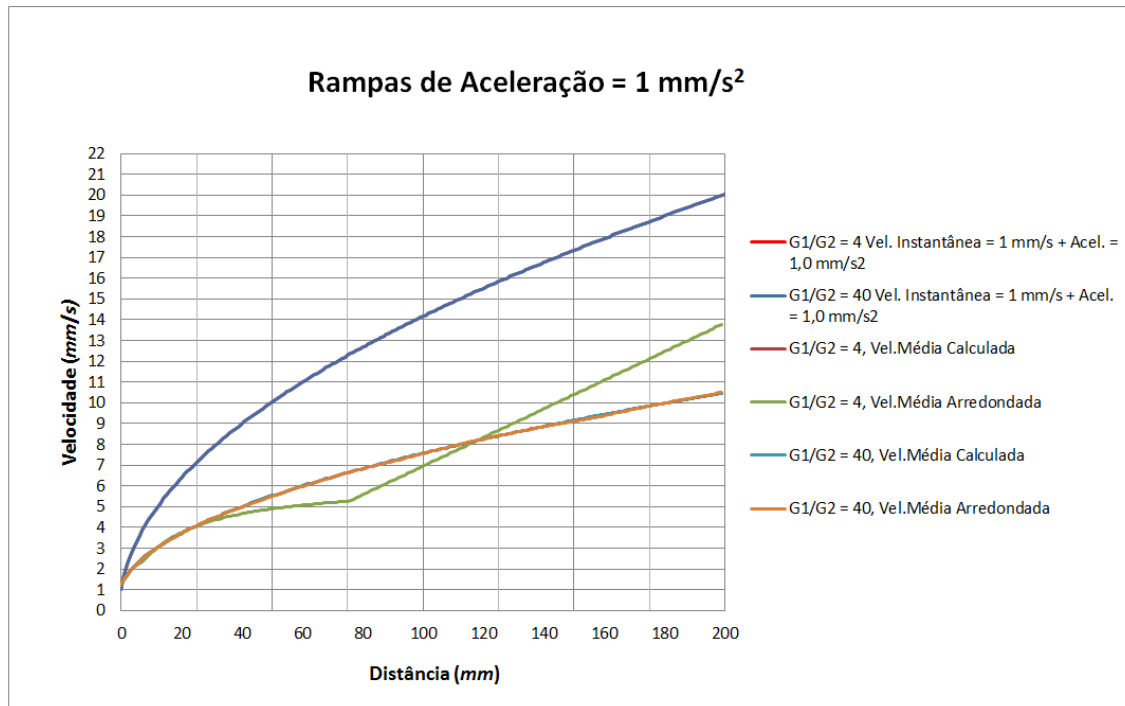


Figura 6.8: Electronic Gear = 4 e 40, Aceleração = 1 mm²/s. Gráfico que relaciona a velocidade média e instantânea com a distância percorridas.

Tal como já explicado anteriormente neste subcapítulo, quando $G1/G2 = 4$, as curvas de velocidades para valores de velocidade instantânea de 12 mm/s deixam de ser válidas uma vez que o sistema adota um comportamento brusco e indesejado, esse ponto é facilmente detetável no gráfico anterior quando a velocidade média arredondada passa a ter um comportamento linear, aumentando a taxa de crescimento dos seus valores. O restante comportamento das curvas de velocidades é o esperado, resultante da equação de segundo grau da cinemática de partículas.

De registar ainda que neste caso, devido ao efeito da aceleração, ou seja o gradiente de aumento de velocidade induzido (e portanto gradiente de diminuição do tempo de duração das rotinas do *Labview*), não se verificam, nomeadamente no caso da *Electronic Gear = 4*, as grandes diferenças entre a velocidade média “calculada” e a velocidade média “arredondada”, que se verificam no caso de a velocidade ser constante. A introdução da aceleração pode assim melhorar a controlabilidade da velocidade da mesa/guia linear para a *Electronic Gear = 4*, desde que para pequenas distâncias e acelerações (pelas razões já referidas anteriormente).

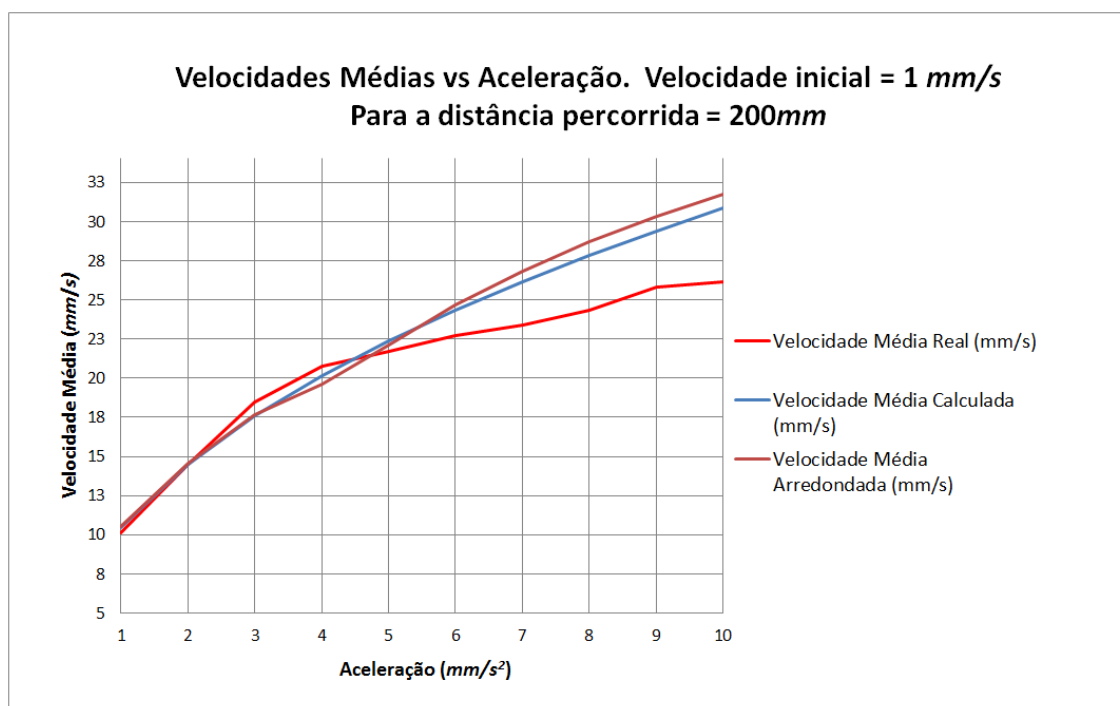


Figura 6.1: Electronic Gear = 40. Relação da velocidade média com a aceleração (distância = 200 mm, velocidade inicial = 1mm/s).

No gráfico seguinte registam-se os valores da velocidade média, conseguida no final de um curso de 200 mm, e partindo de uma velocidade inicial de 1 mm/s, correspondentes a diferentes valores de aceleração, de 1 a 10 mm²/s. Os valores reais da velocidade média foram medidos laboratorialmente, este gráfico compara-os com os valores da velocidade média “calculada” e velocidade média “arredondada”, para uma *Electronic Gear* = 40.

O gráfico demonstra que para valores de aceleração de 1 a 5 mm²/s, o modelo de controlo implementado funciona, uma vez que as “velocidades” previstas são muito próximas da velocidade média real. Acima de 5 mm²/s a velocidade real medida, apesar de manter a tendência de crescimento, assume valores inferiores às velocidades do modelo, sendo que essa diferença aumenta gradualmente. Esta diferença pode eventualmente explicar-se por duas razões, ou por erros de leitura (desfasamento de paragem do cronómetro), que em virtude do aumento da velocidade para a mesma distância de 200 mm, se tornam cada vez mais preponderantes, ou motivada pela insuficiente capacidade de processamento da CPU, já referido anteriormente, que limita a duração mínima das rotinas a 2 milissegundos.

Capítulo 7 – Conclusões e trabalhos futuros

A importância dos servomotores manifesta-se na sua crescente utilização para o acionamento de equipamentos, na execução de diversas tarefas semiautomáticas e automáticas, em diferentes tipos de indústria. Com a implementação de um sistema mecatrónico, automatiza-se o processo possibilitando uma libertação de meios humanos, uma vez que deixa de ser necessária a presença contínua do operador para controlar o equipamento durante toda a duração de uma tarefa.

Contudo, para os servomotores poderem ser correctamente utilizados, tem que ser devidamente estudada a sua aplicação. Para tal é necessário saber os princípios eletromecânicos de geração de movimento rotacional, estudar a electrónica, e o controlo automático de malha fechada destes equipamentos feito pelos controladores dedicados (servo drives).

Os servomotores são particularmente importantes nas tarefas em que seja necessário controlar completamente o posicionamento e velocidade da ferramenta que acionam, assim como o binário do seu eixo de saída, em cada instante.

A utilização do servomotor em conjunto com uma placa de aquisição de dados, DAQ, explorando a capacidade de comunicação bidirecional desta, que permite receber e enviar sinais digitais aos restantes componentes do sistema mecatrónico (no nosso caso, os fins de curso e o controlador), assimilar a informação dos sensores e comandar os parâmetros de saída do controlador (do gerador da corrente e pulsos do driver), em conjunto com o *software* Labview e com o suporte de processamento de um computador comercial comum (a CPU), revelou-se eficiente e versátil para fins de controlo automático da mesa/guia linear.

Por sua vez, o *Labview* fornece uma interface com o operador, o “*Front Panel*”, através do qual o operador pode gerir os parâmetros de comando do servomotor da melhor forma, assim como monitorizar o seu comportamento. Para facilitar a instalação integrada de todos os componentes de *hardware* e *software* o *Labview* possui módulos de comunicação já pré-definidos, na sua base de dados, desenvolvidos pela *National Instruments*, e que permitem reconhecer e trabalhar em conjunto com a DAQ.

Para o sistema de controlo implementado, a DAQ revelou ter um número de Entradas/Saídas de tensão e de níveis lógicos suficientes, e ter um bom desempenho na aquisição e transmissão de dados, para as diferentes frequências de pulsos seleccionadas para comando da velocidade. Verificou-se também para as *Electronic Gear* mais baixas, no caso estudado igual a 4, a necessidade de execução mais rotinas (para avançar a mesma distância, com a mesma velocidade), determina que o sistema fique mais suscetível a pequenas variações nos sinais de entrada, isto compreende-se porque em cada rotina há leitura dos sinais de entrada, ou seja, os sinais de entrada são lidos mais espessamente.

Uma limitação deste sistema implementado, poderá eventualmente ser, para sistemas mecatrónicos mais complexos, o número de entradas / saídas digitais (*digital I/O*), uma vez que a DAQ possui três portos digitais, e que cada porto só poderá ser configurado como de entradas ou de saídas (um porto não pode ter entradas e saídas em simultâneo). Assim sendo um dos portos terá que ser configurado como um porto de saídas (oito saídas, bits, que são suficientes para comandar o servomotor), outro dos portos terá que ser configurado como de entradas, para aquisição dos dados dos sensores instalados. O terceiro porto só pode ser configurado como de entradas ou de saídas, pelo que um dos portos anteriores terá que ser suficiente para um tipo transmissão de dados de todo o sistema mecatrónico, que se lhe destina. Este problema poderá ser ultrapassado ou minorado, utilizando também as entradas e saídas analógicas da DAQ e correspondentes funcionalidades do *software* Labview, que podem ser exploradas em trabalhos futuros.

A análise dos resultados obtidos, abordada no capítulo anterior, na comparação dos mesmos para os casos de *Electronic Gear* = 40 e 4, permite também confirmar que a adoção de *Electronic Gear* mais elevadas (40) torna o controlo da velocidade e da aceleração mais fidedigno (os resultados medidos reais são muito próximos dos resultados modelados), principalmente para velocidades mais elevadas. Por outro lado, a adoção de *Electronic Gear* mais baixos (4) permite obter mais precisão no movimento da guia / mesa linear. Este comportamento era expectável, uma vez que para relações de engrenagens mais baixas (*electronic gear* pequenos) são necessárias muito mais rotinas discretas para a rotação do servomotor, controlando o número de rotinas, conseguem-se distâncias de posicionamento muito precisas (da ordem dos 0,01mm).

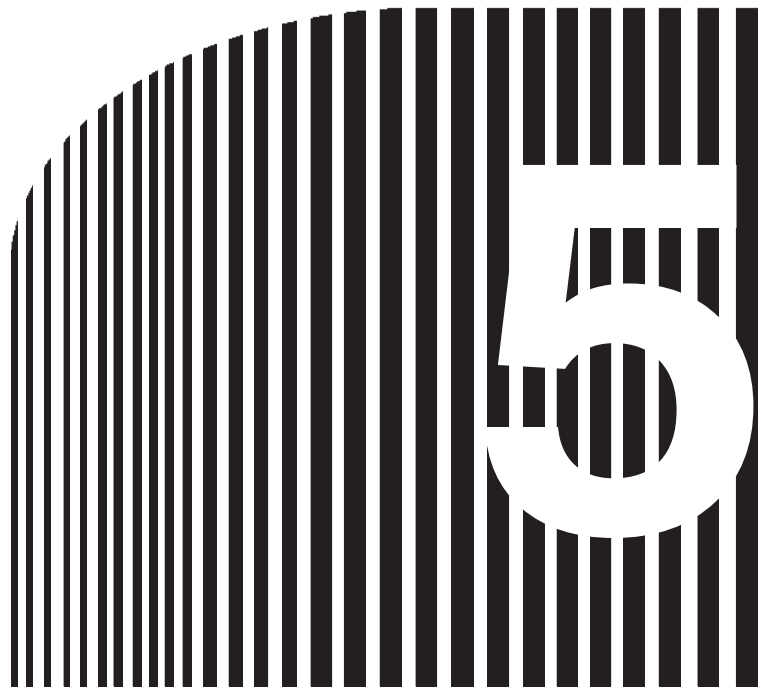
Como sugestões para trabalhos futuros, dando continuidade ao trabalho realizado, indica-se o controlo do binário do servomotor em cada instante, colocando a mesa/guia linear em carga (através do estudo do movimento da guia/mesa linear traduzindo-o sob a forma de diagrama de corpo livre, explicitando todas as forças), nestas condições seria também importante o estudo da redução do consumo energético do sistema mecatrónico implementado, abordando o seu circuito de realimentação do servomotor (e comparar com o consumo medido sem a realimentação ativa).

Referências bibliográficas

- [1] Sítio da www.dei.isep.ipp.pt/~paf/proj/Julho2003/CAD_CAE_CAM_CIM.pdf [online]. [Acedido a 1 de Junho 2012].
- [2] J. Norberto Pires. “Automação Industrial”, 4ª Edição, ETEP, 2007.
- [3] Gottlieb, Irving M. “Electric Motors and Control Techniques”, 2ª Edição, McGRAW Hill, 1994.
- [4] Considine, M. Douglas. “Process / Industrial Instruments and Controls Handbook”. 4ª Edição, McGRAW Hill, 1993.
- [5] Sítio da http://paginas.fe.up.pt/~robosoc/sci/PhD_Armando_Sousa_lo_qual.pdf [online]. [Acedido a 13 de Novembro 2012].
- [6] Paulo Oliveira. “Curso de Automação Industrial”, 3ª Edição, ETEP.
- [7] Considine. “Standard Handbook of Industrial Automation”, Chapman na Hall, 1986.
- [8] Juan Carlos Martín, María Pilar García. "Automatismos Industriales", Electricidad y Electrónica, Editorial Editex, SA, 2011.
- [9] OMNUC - Series EU Models. AC Servomotors/Drivers (100 to 750 W Pulse-train Inputs), User´s Manual, Omron.
- [10] Sítio da <http://www.anaheimautomation.com/manuals/forms/servo-motor-guide.php> [online]. [Acedido a 13 de Novembro de 2013]
- [11] Sítio da: http://industrial.omron.pt/pt/products/catalogue_pdf/default.html [online]. [Acedido a 2 de Junho de 2013]
- [12] Sítio da http://www.ebah.com.br/content/ABAAAA_4IAG/servomotores?part=2 [online]. [Acedido a 2 de Junho de 2013]
- [13] Sítio da http://www.pronum.pt/index.php?option=com_content&view=article&id=47&Itemid=54&lang=pt [online]. [Acedido a 4 de Abril de 2013]
- [14] Sítio da <http://www.newtoncbraga.com.br/index.php/como-funciona/5454-mec128> [online]. [Acedido a 29 de Abril de 2013]
- [15] OMNUC U, Series Servo Driver. Computer Monitor Software (Verson 1.1), Instruction Manual, Omron.

- [16] David W. South, Jon R. Mancuso. "Mechanical Power Transmission Components", Marcel Dekker, Inc.1994
- [17] Niemann. "Elementos de Máquinas", Volume III, Editora Edgard Blucher Ltda, 1991.
- [18] A. Sousa Miranda. Apontamentos Órgão de Máquinas, Universidade do Minho.
- [19] Sítio da <http://www.thomsonlinear.com>. Precision_Ball_Screws_ctuk [online]. [Acedido a 08 de Outubro de 2013].
- [20] Sítio da <http://www.thomsonlinear.com>. Selecting_Sizing_Ball_Screw_Drives_taen [online]. [Acedido a 07 de Outubro de 2013].
- [21] Sítio da <http://www.thomsonlinear.com>. Roundrail_1CC_Continuously_Supported_en [online]. [Acedido a 07 de Outubro de 2013].

**Anexo I. Extrato do Instruction Manual, Computer Monitor
Software (Verson 1.1), OMNUC U, Chapter 5, Operation.**



Chapter 5

• Operation •

- 5-1 Basic Operating Procedure
- 5-2 Operation Tree
- 5-3 System Monitor
- 5-4 Monitor Function
- 5-5 Setting Parameters
- 5-6 Alarm Display
- 5-7 Jog Operation
- 5-8 Auto Tuning
- 5-9 Adjustment
- 5-10 Graph
- 5-11 Servo Driver Selection
- 5-12 Off Line Edit

This chapter describes the basic operations of the Computer Monitor Software in on line mode. The following explains each of these operations.

The chapter concludes with a description of the off line edit mode.

The screens used in this chapter are from the R88D-UT□□H or R88D-UA□□HA(LA) type. While the Servo Driver display contents may differ the operating procedures remain the same. The following abbreviations are used to distinguish the functions relating to different Servo Driver models.

H/L: R88D-UA□□H/-UA□□L, R88D-UP□□H/-UP□□L

HA/LA: R88D-UA□□HA/-UA□□LA, R88D-UP□□HA/-UP□□LA

UT: R88D-UT□□H, R88D-UT□□V/-UT□□W

UE: R88D-UEP□□H/-UEP□□L

5-1 Basic Operating Procedure

■ Operation Flow

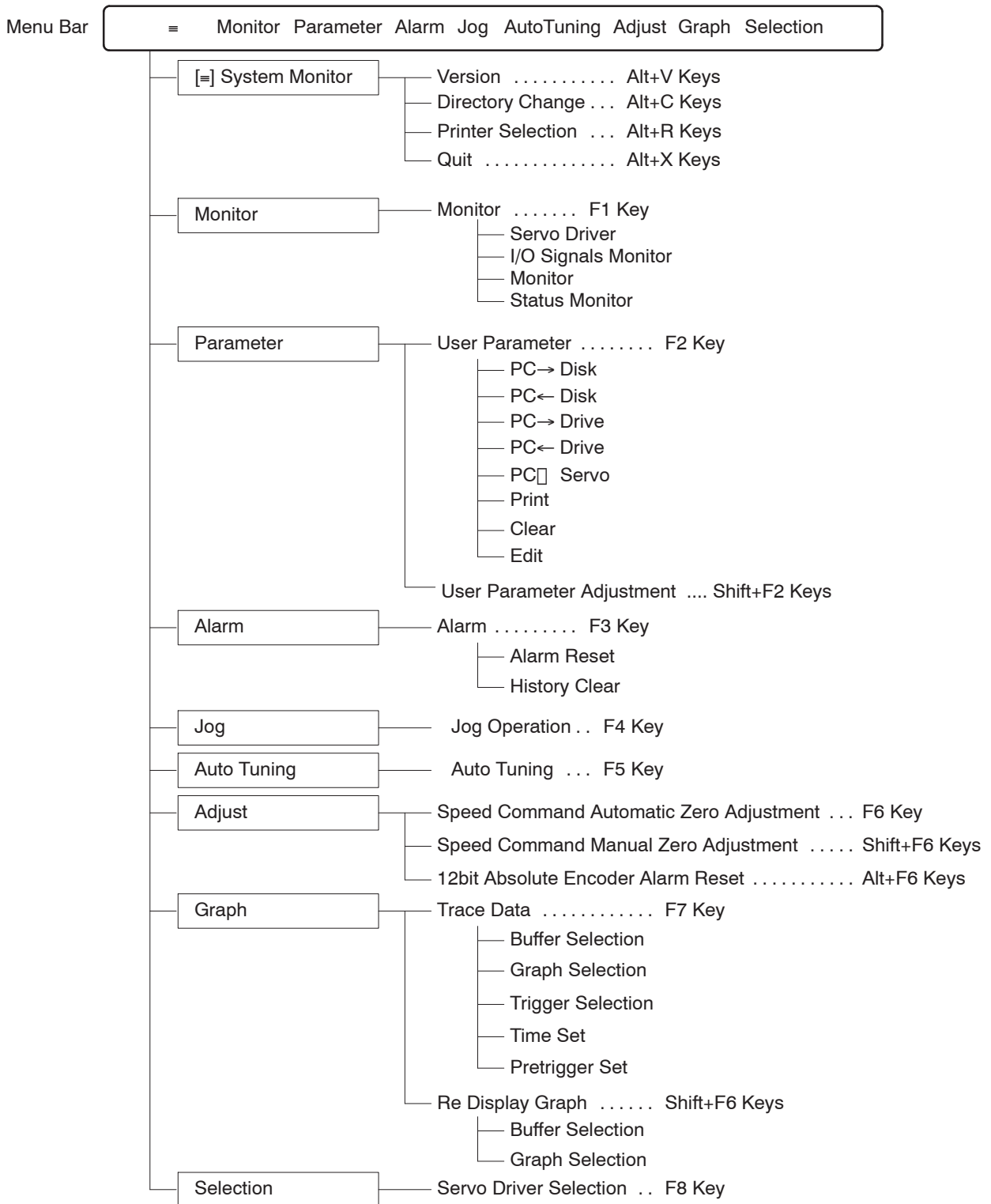
START

1. Pre-start check	H/L	HA/LA	UT	UE	Chapter 4
<ul style="list-style-type: none"> • Servo Driver Unit No. setting (for multiple axis communication). • Cable connection. 					
2. Starting the Computer Monitor Software	H/L	HA/LA	UT	UE	Chapter 4
<ul style="list-style-type: none"> • Servo Driver model setting. • Selecting the RS-232C port. • Unit No. setting (for multiple axis communication). 					
3. Editing Parameters	H/L	HA/LA	UT	UE	Section 5-5
<ul style="list-style-type: none"> • Reading parameters from the disk. • Changing parameters. • Transferring parameters to the Servo Driver. • Switching the Servo Driver ON and OFF. 					
4. Control offset automatic adjustment		HA/LA	UT		Section 5-9
<ul style="list-style-type: none"> • Adjusting the analog voltage offset for speed command and torque command (for analog input). 					
5. Jog operation		HA/LA	UT	UE	Section 5-7
<ul style="list-style-type: none"> • Setting the motor revolution speed and jog operation. 					
6. Mechanical load and connectivity (refer to the appropriate product operation manual)					
7. Auto tuning		HA/LA	UT	UE	Section 5-8
<ul style="list-style-type: none"> • Carrying out auto tuning according to mechanical rigidity. 					
8. Test run (refer to appropriate product operation manual)					
9. Graph display		HA/LA	UT	UE	Section 5-10
<ul style="list-style-type: none"> • Checking running condition velocity and torque graph. 					
10. Run (refer to appropriate product operation manual)					

END

5-2 Function Tree

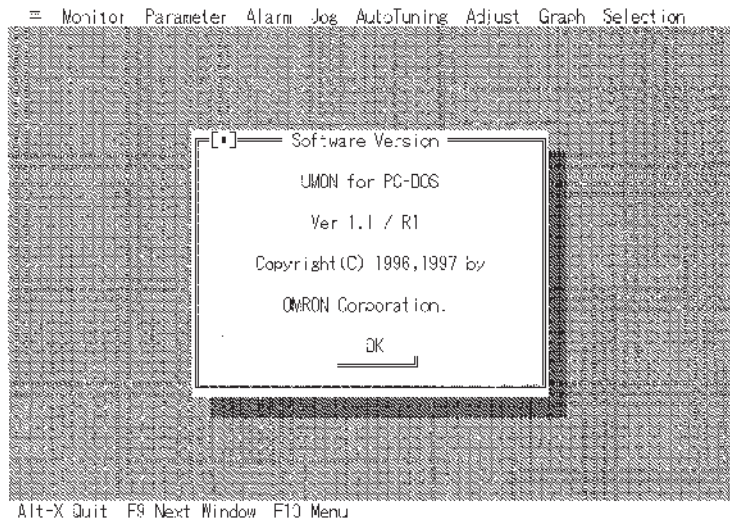
The following tree diagram provides a summary of Computer Monitor Software functions.



5-3 System Monitor

H/L, HA/LA, UT, UE

■ Version



1. Software Version Window

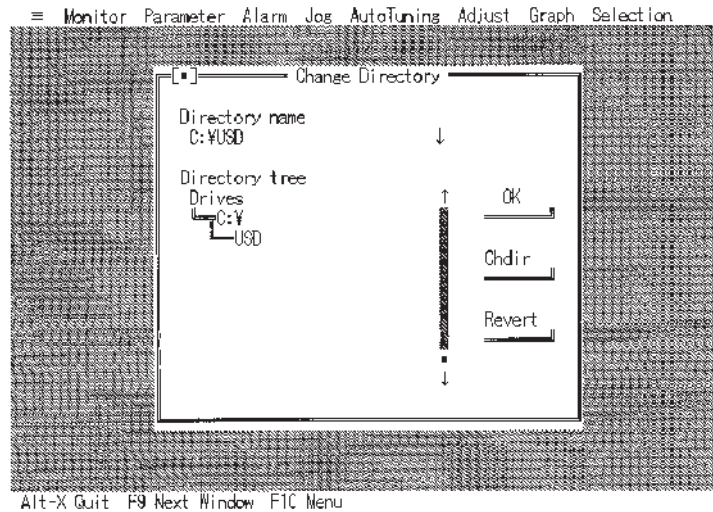
Click on or select [=] - [Version].

2. Closing Software Version Window

Click on [OK] or the [•] in the upper-left part of the window.

Press Space Key, Return Key, or Escape Key.

■ Changing the Directory



[Directory Change] is used when changing the drive or directory for reading or saving parameter files and graph data files (trace buffers).

1. Change Directory Window

Click on or select [=] - [Change Directory].

2. Change Directory

Use one of the following two methods.

Entering the Directory Name Directly

- ☞ Type the directory name in the [Directory Name] field and press the Return Key. This operation will change the directory and close the Change Directory window.

Selecting the Directory from the Screen

- Change the Drive

- ☞ Double-click on [Drives] in the [Directory tree] field. The drive name is displayed. Double-click on the desired drive name.
- ☞ Select the [Directory Tree] field using the Tab Key. When [Drives] is selected, press the Space Key or Return Key. The drive's directories will be displayed.

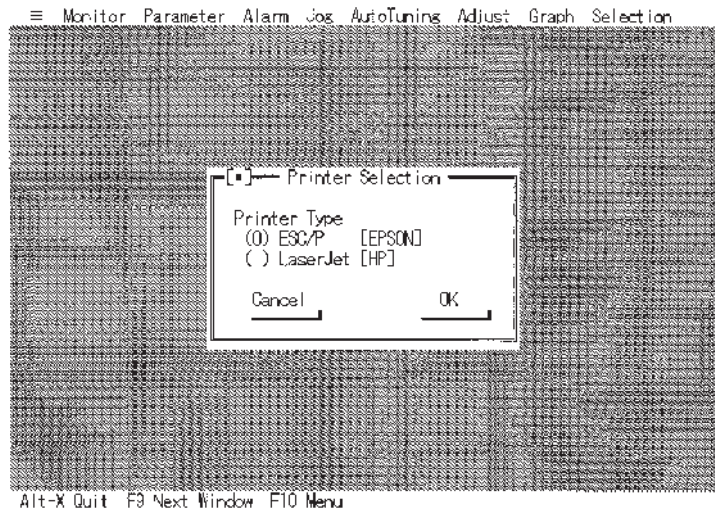
- Select the Directory from the Screen

- ☞ Double-click on the desired directory name. Click on [OK].
- ☞ Select the desired directory name using the Tab Key. Press the Space Key or Return Key. Select [OK] using the Tab Key, then press the Space Key or Return Key. This will change the directory and close the Change Directory window.

3. Other Operations

Selecting [Revert] will return the directory to its previous setting.

■ **Selecting the Printer**



[Printer Selection] is used when selecting the printer model connected to the IBM PC/AT or compatible.

1. Printer Selection Window

- ☞ Click on or select [=] - [Printer Selection].

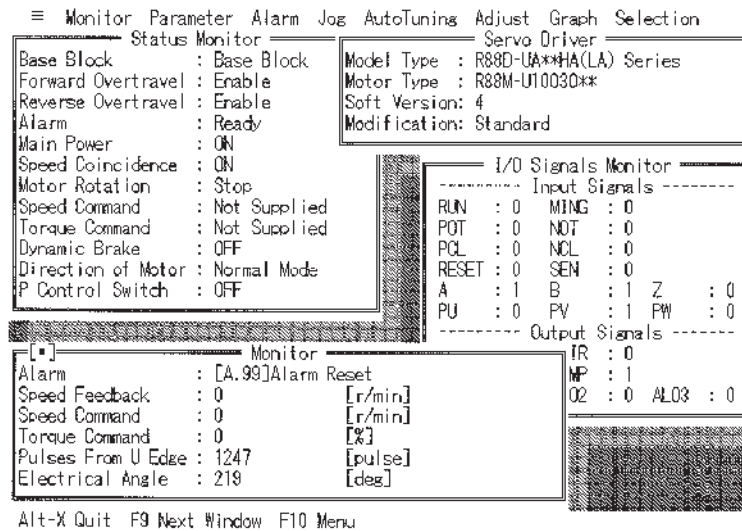
2. Printer Selection

- ☞ Click on the selected printer, then click on [OK].
- ☞ Select [Printer Type] using the Tab Key. Select the printer using the Up and Down Cursor Keys. Select [OK] using the Tab Key, then press the Space Key or Return Key.

5-4 Monitor Function

HA/LA, UT, UE

Selecting the Servo Driver from the screen automatically displays the Monitor Display screen. The following explains how to use the four screens (on line mode). For information on each monitor, refer to the appropriate product operation manual.



■ Changing the Active Window

- An active window is any window on the screen that can be closed or moved to another location on the screen.
- The active window is at the front of the screen.
- The active window has a bold frame with a [*] mark on the upper-left part of the window.

Changing the Active Window

Click on the desired window.

Press the F9 Key. Repeatedly pressing the F9 Key will make each window active in turn.

■ Moving the Active Window

- The active window can be re-located by dragging with the mouse.

Moving the Active Window

Position the cursor on the upper window frame border and drag to the desired location.

■ Closing the Active Window

- For easier viewing close any unnecessary windows.

Closing Active Windows



Click on the [*] mark on the upper-left part of the window.

Press the Escape Key. This operation will close the window and make the underlying window active.

■ **Displaying the Window**

- The following operation will re-display a window that has been removed from the screen.

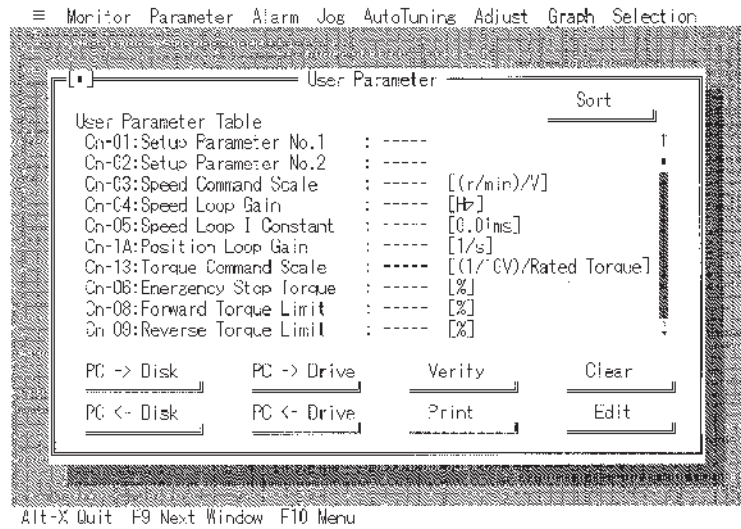
Displaying a Monitor Window

  Click on or select [Monitor] - [Monitor]. All windows will be displayed.

5-5 Setting Parameters

H/L, HA/LA, UT, UE

In parameter settings, each user parameter can be changed in real-time through such functions as parameter edit, parameter transfer to disk or Servo Driver, and parameter comparison check. When connected to an OMRON Position Control Unit, use the existing Position Control Unit-compatible parameter files to speed-up the setting operation.



5-5-1 Editing Parameters

■ Editing Setup Parameters

1. Opening the User Parameter Window

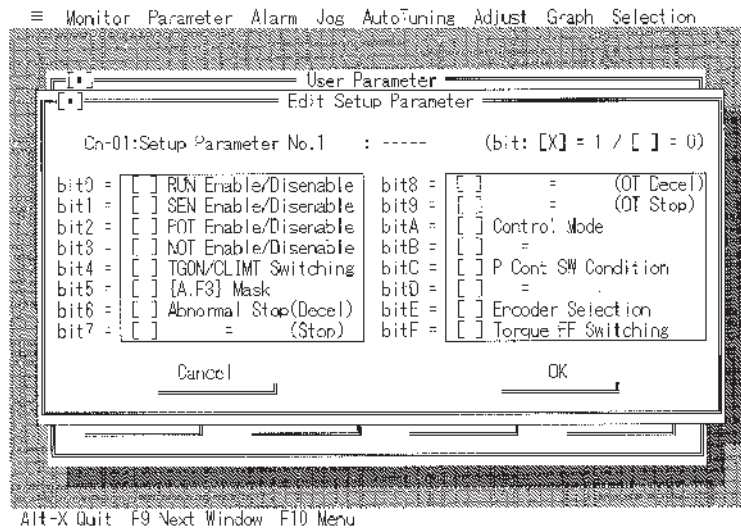
Click on or select [Parameter] - [User Parameter].

2. Selecting [Setup Parameter No.1] or [No.2]

Double-click on [Setup Parameter No.1] or [No.2].


Select [User Parameter Table] using the Tab Key. Select [Setup Parameter No.1] or [No.2] using the Down Cursor Key. Press the Return Key.


The Edit Setup Parameter window opens.



3. Setting Each Bit to 0 or 1

- The setup parameter sets each bit to 0 or 1.
- [] and [X] indicate 0 and 1 settings respectively.
- For more information on the definition of setup parameter bit, refer to the appropriate product operation manual.

 Click on [] to the right of the desired bit No. Change the bit setting accordingly by selecting [] or [X] for the respective 0 and 1 settings.

 Select bit No. 0 using the Tab Key. Press the Tab Key again to select descending items. Select the desired bit No. setting change and press the Space Key. [] and [X] indicate 0 and 1 settings respectively.

4. Saving the Setting Information to the IBM PC/AT or Compatible Memory

- Select [OK] to save setting information to the IBM PC/AT or compatible memory. Select [Cancel] to cancel the setting information.

 Click on [OK].

 Select [OK] using the Tab Key, then press the Space Key or Return Key.

Edit Setup Parameter window closes and program returns to User Parameter window.

5. Editing Other User Parameter Numbers

- Refer to the following section for editing user parameter numbers from Cn-03 on.

■ Editing User Parameter Numbers from Cn-03

1. Opening the User Parameter Window


  Click on or select [Parameter] - [User Parameter].

Note This operation is not necessary if the window is already open after setting the setup parameter.

2. Selecting the Desired User Parameter Setting

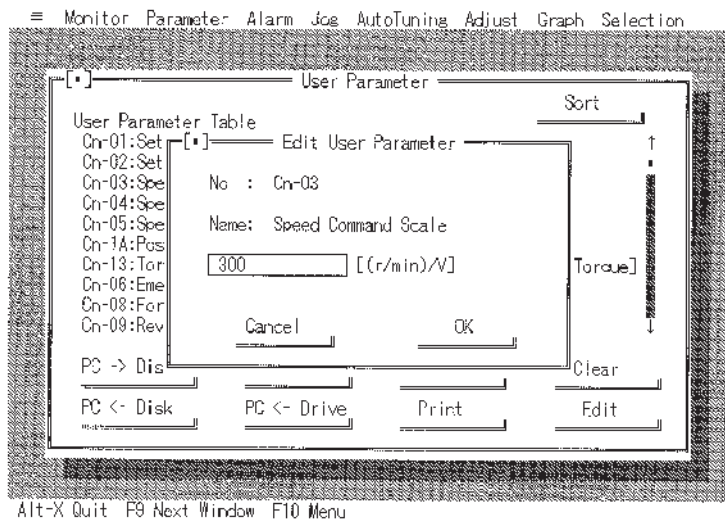
- For information on user parameter settings refer to the appropriate product operation manual.

 Double-click on the desired user parameter item setting.

 Select the [User Parameter Table] using the Tab Key. Use the Up and Down Cursor Keys to select the desired user parameter item. Press the Space Key or Return Key.

Note All user parameters are displayed in numerical order or in the order of contents by clicking on [Sort]. The UE model does not have this function.

Edit User Parameter window opens.



3. Entering the Setting and Saving to the IBM PC/AT or Compatible Memory

- Type in the setting.
- Select [Cancel] without pressing the Return Key to cancel the entered setting.
- Select [OK] or [Cancel] as follows:

 Click on [OK] or [Cancel].

 Select [OK] or [Cancel] using the Tab Key. Press the Space Key or Return Key.

Edit User Parameter window closes and the program returns to User Parameter window.

4. Editing Other User Parameter Numbers

5. Go Back to No. 2.

5-5-2 Transferring Parameters between IBM PC/AT or Compatible and Disk

This section explains how to save the parameters recorded in IBM PC/AT or compatible memory to disk. This section also explains how to read the disk's parameter files to the IBM PC/AT or compatible memory.

The target disk for saving and reading is set according to *Changing the Directory* in 5-3 *System Monitor*. When using a parameter file-dedicated disk or directory, select [=] - [Change Directory] before setting.


■ Saving from IBM PC/AT or Compatible to Disk

1. Opening the User Parameter Window

  Click on or select [Parameter] - [User Parameter].

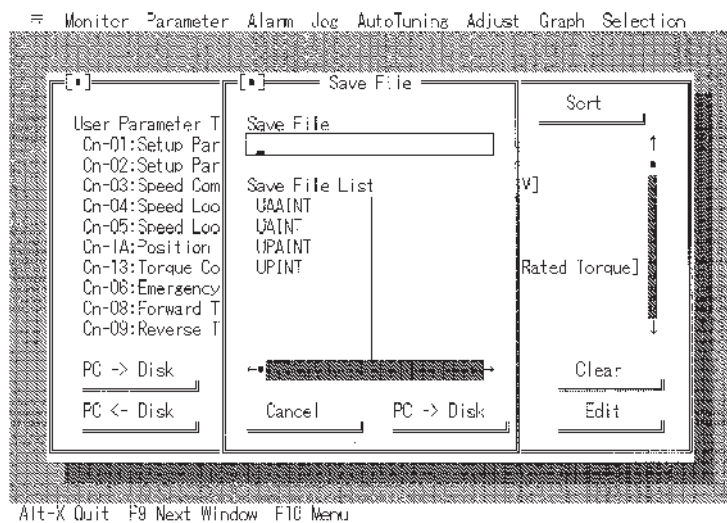
Note This operation is not necessary if the window is still open after setting the parameters.

2. Selecting [PC→Disk]

 Click on [PC→Disk].

 Select [PC→Disk] using the Tab Key. Press the Space Key or Return Key.

Save File window opens.



3. Entering (Selecting) the File Name

Creating a New File

- Type the File Name into the [Save File] field. Press the Return Key or select [PC→Disk].
- Select [PC→Disk] as follows:

 Click on [PC→Disk].

 Select [PC→Disk] using the Tab Key. Press the Space Key or Return Key.

Note When entering the File Name do not add an extension. One will be allocated automatically.

Typing Over an Existing File

- ☞ Double-click on a file name in the [Save File List].
- ☞ Select [Save File List] using the Tab Key. Select the file name using the Up and Down Cursor Key and press the Return Key.

Parameters are saved to disk, Save File window closes, and program returns to the User Parameter window.

■ **Reading the Disk to the IBM PC/AT or Compatible**

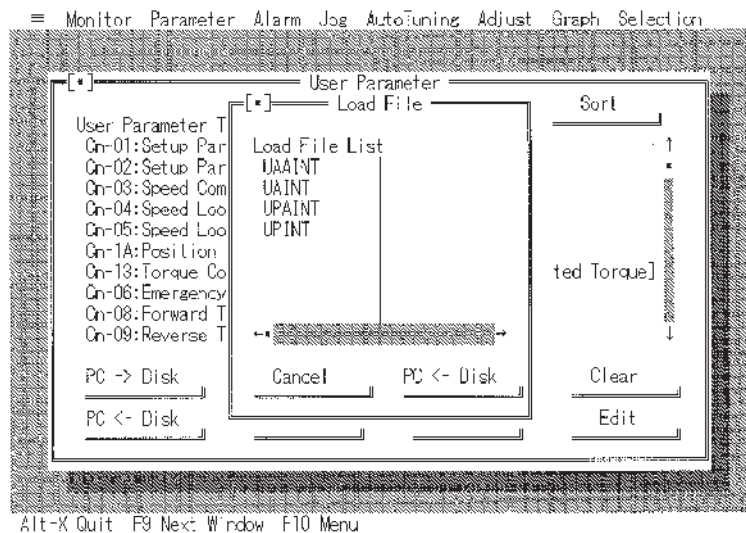
1. Opening the User Parameter Window

- ☞ Click on or select [Parameter] - [User Parameter].

Note This operation is not necessary if the window is already open.

2. Selecting [PC←Disk]

- ☞ Click on [PC←Disk].
- ☞ Select [PC←Disk] using the Tab Key then press the Space Key or Return Key.
Load File window opens.



3. Selecting a File Name

- ☞ Double-click on a file name in the [Load File List].
- ☞ Select [Load File List] using the Tab Key. Select the file name using the Up and Down Cursor Keys. Press the Return Key.

Parameters are read to the IBM PC/AT or compatible, Load File window closes, and program returns to the User Parameter window.

● Existing Parameter Files

To speed-up the setting operation use the existing OMRON Position Control Unit-compatible parameter files in the Computer Monitor Software.

File name	Servo Driver model	Applicable Position Control Unit
UAINIT	R88D-UA□□H/L	C500-NC222-E, CV500-MC221/MC421
UPINT	R88D-UP□□H/L	C200H-NC112/NC211 3G2A5-NC111-EV1
UAAINT	R88D-UA□□HA/LA	C500-NC222-E, OMNUC N115/N116, OMNUC U43/U45 CV500-MC221/MC421
UPAINT	R88D-UP□□HA/LA	C200H-NC112/NC211, 3G2A5-NC111-EV1
UTHINT	R88D-UT□□H	C200H-NC112/NC211, 3G2A5-NC111-EV1
UTHANL	R88D-UT□□H	C500-NC222-E, CV500-MC221/MC421
UTWINT	R88D-UT□□W/V	C200H-NC112/NC211, 3G2A5-NC111-EV1
UTWANL	R88D-UT□□W/V	C500-NC222-E, CV500-MC221/MC421
UEPINT	R88D-UEP□□H/L	C200H-NC112/NC211, 3G2A5-NC111-EV1

5-5-3 Transferring Parameters between the IBM PC/AT or Compatible and Servo Driver

This section explains how to transfer parameters stored in IBM PC/AT or compatible memory to the Servo Driver. This section also explains how to read Servo Driver parameters to the IBM PC/AT or compatible memory.

Caution When connecting to a **HA/LA** Servo Driver, make sure a Unit No. has been set and the Unit No. is correct before transferring parameters to the Servo Driver. Transferring an incorrect Unit No. to the Servo Driver may damage the Servo Driver and IBM PC/AT or compatible.

Note 1. When using multiple axis communication, a Servo Driver Unit No. of 0 or a duplicate Servo Driver Unit No. may damage the Servo Driver and IBM PC/AT or compatible.

Note 2. In the User Parameter window, a “----” condition in the [Unit Number]field indicates no Unit No. setting. In this condition, no parameters will be transferred to the Servo Driver.

■ Transferring from the IBM PC/AT or Compatible to Servo Driver

1. Opening the User Parameter Window

  Click on or select [Parameter] - [User Parameter].

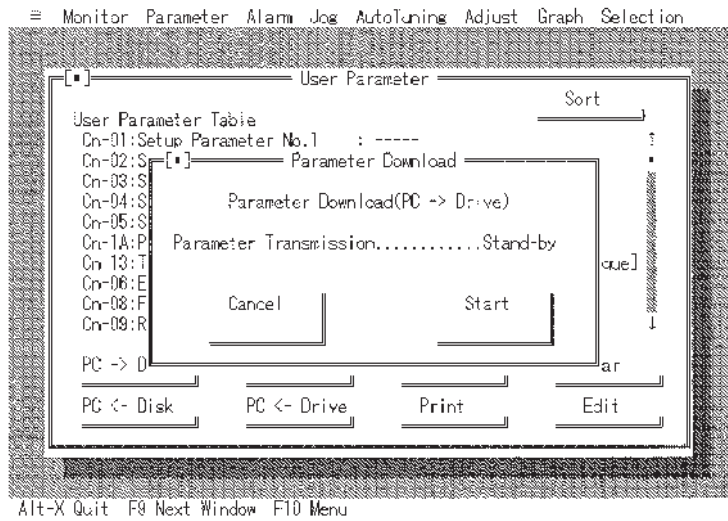
Note This operation is not necessary if the window is already open.

2. Selecting [PC→Drive]

 Click on [PC→Drive].

 Select [PC→Drive] using the Tab Key. Press the Space Key or Return Key.

Parameter Download window opens.



3. Selecting [Start]

Click on [Start].

Select [Start] using the Tab Key. Press the Space Key or Return Key.

Parameters are transferred to Servo Driver, Parameter Download window closes, and program returns to the User Parameter window.

4. Switching the Servo Driver OFF then ON

Refer to the following.

Caution when Transferring Parameters to Servo Driver

- Certain parameters will be enabled when the Servo Driver is switched OFF then ON again. Once these parameters have been changed, switch OFF the parameter-loaded Servo Driver. Make sure the screen's indicator is not lit before switching ON again. This applies to the Setup Parameters as well as some User Parameter Numbers. For further information refer to the appropriate product operation manual.
- When connecting a R88D-UA□□HA/LA model and the Setup Parameter Cn-01 bit No. E (Encoder Selection) has been modified, switch the Servo Driver ON and restart the Computer Monitor Software or re-execute the [Servo Driver Selection]. Normally, the [12 bit Absolute Encoder Alarm Reset] is not activated.

■ **Transferring Parameters from Servo Driver to the IBM PC/AT or Compatible**

1. Opening the User Parameter Window

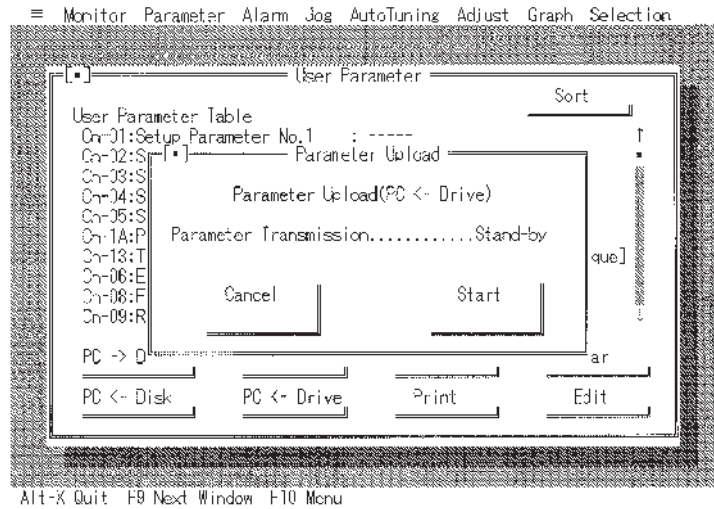
Click on or select [Parameter] - [User Parameter].

Note This operation is not necessary if the window is already open.

2. Selecting [PC←Drive]

Click on [PC←Drive].

- ☞ Select [PC←Drive] using the Tab Key. Press the Space Key or Return Key.
Parameter Upload window opens.



3. Selecting [Start]

- ☞ Click on [Start].
- ☞ Select [Start] using the Tab Key. Press the Space Key or Return Key.
Parameters are transferred to IBM PC/AT or compatible, Parameter Upload window closes, and program returns to the User Parameter window.

5-5-4 Comparing Parameters

The user can compare the settings of user parameters in the personal computer's memory with those in the Servo Driver.

■ Comparing Parameters

1. Opening the User Parameter Window

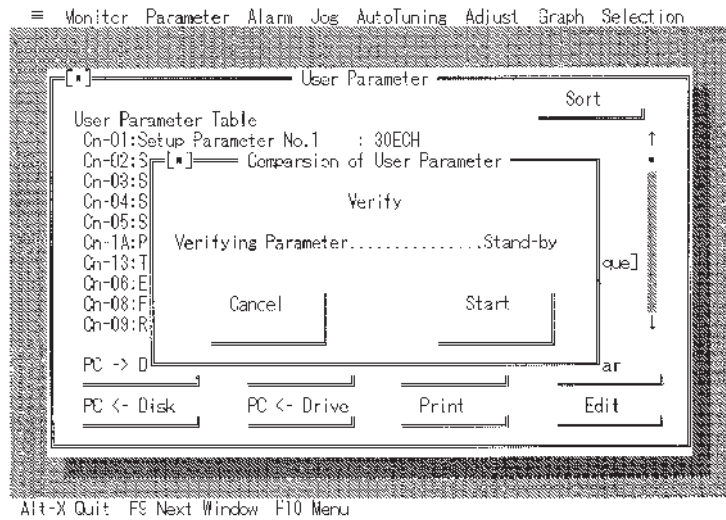
- ☞ ☞ Click on or select [Parameter] - [User Parameter].

Note This operation is not necessary if the window is already open.

2. Selecting [PC □ Servo]

- ☞ Click on [PC □ Servo].
- ☞ Select [PC □ Disk] using the Tab Key. Press the Space Key or Return Key.

Comparison of User Parameter window opens



3. Selecting [Start]

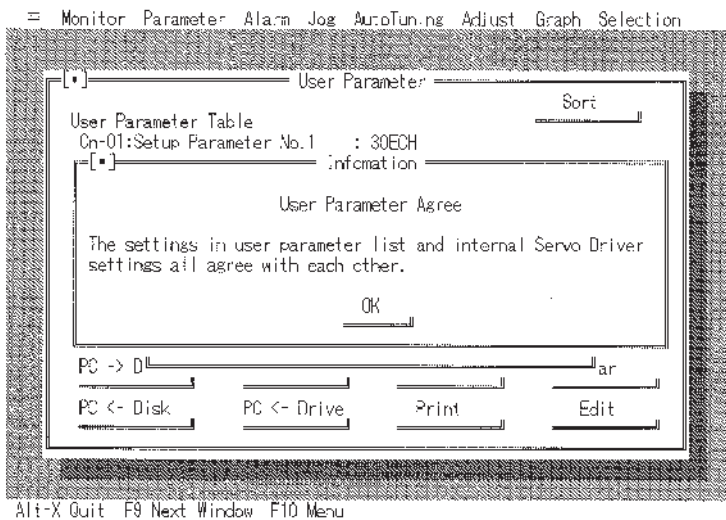
Click on [Start].

Select [Start] using the Tab Key. Press the Space Key or Return Key.

The message "Verifying Parameter...Cn-□□" appears and the set parameters are compared.

Result of comparison: All in agreement

Information window opens.



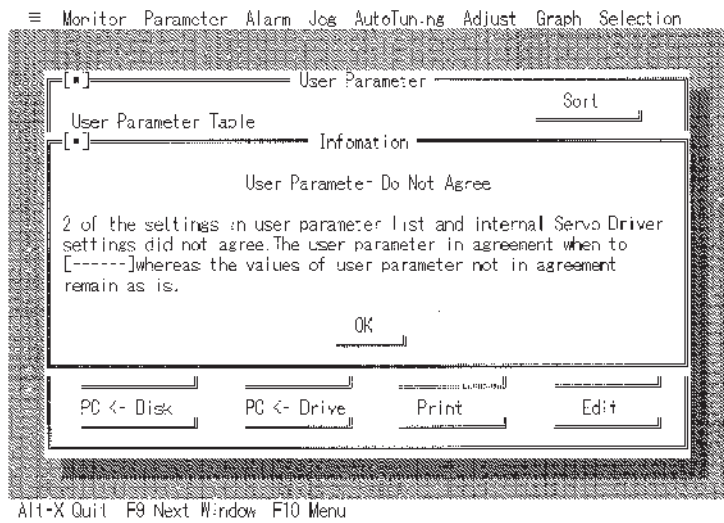
Click on [OK].

Press the Space Key or Return Key.

The Information window closes and the User Parameter window appears again.

Result of comparison: Not in agreement

Information Window opens.



- The number of user parameters not in conformity is indicated.

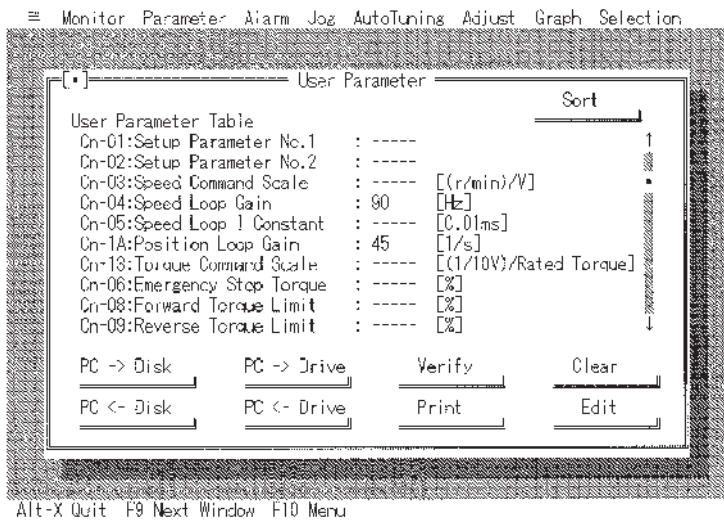


Click on [OK].



Press the Space Key or Return Key.

The Information window closes, the User Parameter Window appears, and the number of user parameters not in agreement appears.



- The settings in user parameters not in agreement remain as they are. Check these settings.

Note Before comparing parameters, be sure that all the parameters in the user parameter list have been set (i.e., none of the parameters indicates “-----” instead of a set value).

5-5-5 Clearing Parameters


This section describes how to clear all parameters stored in the IBM PC/AT or compatible memory (or revert to a “no setting entered” condition).

1. Opening the User Parameter Window

  Click on or select [Parameter] - [User Parameter].

Note This operation is not necessary if the window is already open.

2. Selecting [Clear]

 Click on [Clear].

 Select [Clear] using the Tab Key. Press the Space Key or Return Key.

All parameters are cleared.

- Once the parameters are cleared “-----” will appear in the setting field.

5-5-6 Printing Parameters


This section describes how to obtain parameter printouts.

1. Opening the User Parameter Window

  Click on or select [Parameter] - [User Parameter].

Note This operation is not necessary if the window is already open.

2. Selecting [Print]

 Click on [Print].

 Select [Print] using the Tab Key. Press the Space Key or Return Key.

Sample Parameter Printout

```

R88D-UT**H Series User Parameter          1996-02-14  [19:33:59]

Cn-01:Setup Parameter No.1      : 30ECH
Cn-02:Setup Parameter No.2      :
Cn-11:Number of Encoder Pulses: 8192  [P/R]
Cn-2B:Control Mode Selection    : 1
Cn-2A:Motor Selection           : 145
Cn-03:Speed Command Scale       : 300  [(r/min)/V]
Cn-04:Speed Loop Gain           : 80   [Hz]
Cn-05:Speed Loop I Constant     : 2000 [0.01ms]
Cn-1A:Position Loop Gain        : 40   [1/s]
Cn-1C:Bias Rotational Speed     : 0    [r/min]
Cn-1D:Feed-Forward Amount       : 0    [%]
Cn-17:Torque Filter Time        : 4    [100us]
Cn-28:Compensating Gain         : 0
Cn-0C:P Control SW(Torque)      : 200  [%]
Cn-0D:P Control SW(Speed)       : 0    [r/min]
Cn-0E:P Control SW(Accel)       : 0    [10(r/min)/s]
Cn-0F:P Control SW(Counter)     : 10   [Command Unit]
Cn-0A:Encoder Divider Rate      : 1000 [P/R]
Cn-24:Electronic Gear Ratio G1: 4
Cn-25:Electronic Gear Ratio G2: 1
Cn-07:Soft Start Time(Accel)    : 0    [ms]
Cn-23:Soft Start Time(Decel)    : 0    [ms]
Cn-26:Pulse Accel/Decel Time    : 0    [0.1ms]
Cn-27:Feed-Forward Gain Filter: 0    [0.1ms]
Cn-08:Forward Torque Limit      : 350  [%]
Cn-09:Reverse Torque Limit      : 350  [%]
Cn-18:Forward Current Limit     : 100  [%]
Cn-19:Reverse Current Limit     : 100  [%]
Cn-06:Emergency Stop Torque     : 350  [%]
Cn-13:Torque Command Scale      : 30   [(1/10V)/Rated Torque]
Cn-14:Speed Limit(T Cont I)     : 3000 [r/min]
Cn-2D:Output Signal Selection   : 210
Cn-0B:TGON Speed                : 20   [r/min]
Cn-29:Position Lock Speed       : 10   [r/min]
Cn-22:VCMP Output Width        : 10   [r/min]
Cn-1B:INP Output Width         : 3    [Command Unit]
Cn-1E:Counter Overflow Level    : 1024 [x256 Command Unit]
Cn-12:Brake Timing 1           : 0    [10ms]
Cn-15:Brake Command Speed      : 100  [r/min]
Cn-16:Brake Timing 2           : 50   [10ms]
Cn-10:Jog Speed                 : 500  [r/min]
Cn-1F:No.1 Internal Speed       : 100  [r/min]
Cn-20:No.2 Internal Speed       : 200  [r/min]
Cn-21:No.3 Internal Speed       : 300  [r/min]
Cn-2C:Encoder Power Voltage     : 52500 [0.1mV]

```

5-5-7 Adjusting Parameters

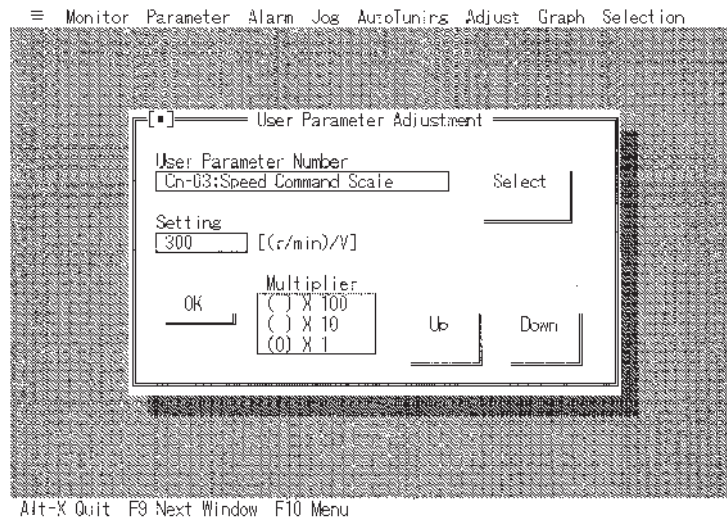
The user can make real-time changes in each user parameter. Therefore, the user parameter can be edited as if the R88A-PR02U or R88A-PR03U Parameter Unit was being used.

■ Adjusting Parameters

1. Opening the User Parameter Adjustment Window

  Select [Parameter]-[User Parameter Adjustment].

User Parameter Adjustment window opens.



2. Selecting the User Parameter

Click on [Select].

The User Parameter Select window opens.

Double-click on the user parameter item to be set.

The User Parameter Adjustment window appears again.

Select [Select] using the Tab Key.

Press the Space Key or Return Key.

The User Parameter Select window opens.

Select the list of user parameters using the Tab Key.

Select the user parameter item using the Up Key or Down Key and press the Return Key.

The User Parameter Adjustment window appears again.

Note All user parameters are displayed in numerical order or in the order of contents by clicking on [Sort]. The UE model does not have this function.

3. Changing the Set Value

Select one the of multiplier values and click on it.

Click on [Up] or [Down]. By clicking on the button each time, the set value changes with the multiplier value selected.

Select [Multiplier] using the Tab Key and select one of the multiplier values using the Up Key or Down Key.

Select [Up] or [Down] using the Tab Key.

Press the Space Key or Return Key. By pressing the Space Key or Return Key each time, the set value changes with the multiplier value selected.

4. Closing the User Parameter Adjustment Window.

Click on the [•] mark on the upper-left part of the window.

Press the Escape Key.

Precautions for Parameter Adjustment

- Some user parameters are not enabled unless the Servo Driver is turned off and on after they are set. If the settings in such parameters are changed, turn off the Servo Driver, check that all indicators on the front panel are off, and turn on the Servo Driver.
- If user parameters are changed using the parameter adjustment function, no user parameters set in the personal computer's memory will be changed.

5-6 Alarm Display

HA/LA, UT, UE

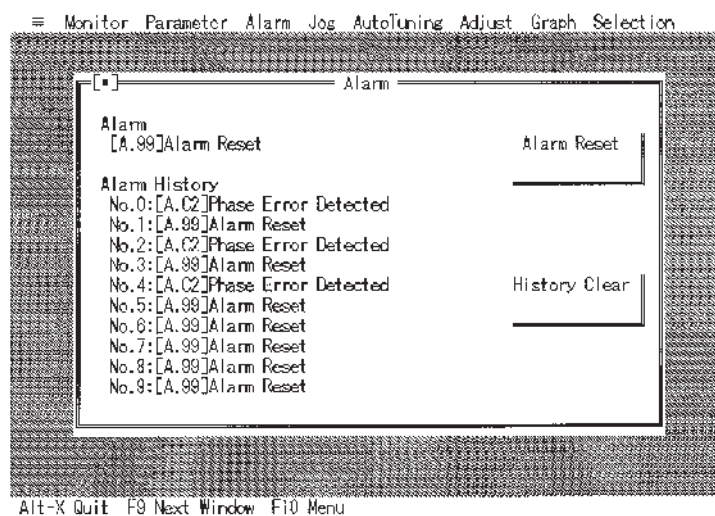
By selecting [Alarm] - [Alarm], the user can check present and past alarm conditions. It also enables the user to reset a present alarm as well as clear past alarm records. Should an A.81 or A.82 alarm condition occur when using the Absolute Encoder, select [Adjust] - [12 Bit Absolute Encoder Alarm Reset].

■ Checking the Alarm

1. Opening the Alarm Window

  Click on or select [Alarm] - [Alarm].

Alarm window opens.



2. Closing the Alarm Window

 Click on the [•] in upper-left part of the window.

 Press the Escape Key.

■ Resetting the Alarm

Should the Servo Driver activate an alarm, reset the alarm from the IBM PC/AT or compatible.

1. Selecting [Alarm Reset]

 Click on [Alarm Reset].


 Select [Alarm Reset] using the Tab Key. Press the Space Key or Return Key.

Alarm is reset.

■ Clearing Alarm Records

Clears alarm condition records.

1. Selecting [History Clear].

 Click on [History Clear].

 Select [History Clear] using the Tab Key. Press the Space Key or Return Key.

Alarm history is cleared and all historical data reverts to Alarm Reset.

5-7 Jog Operation

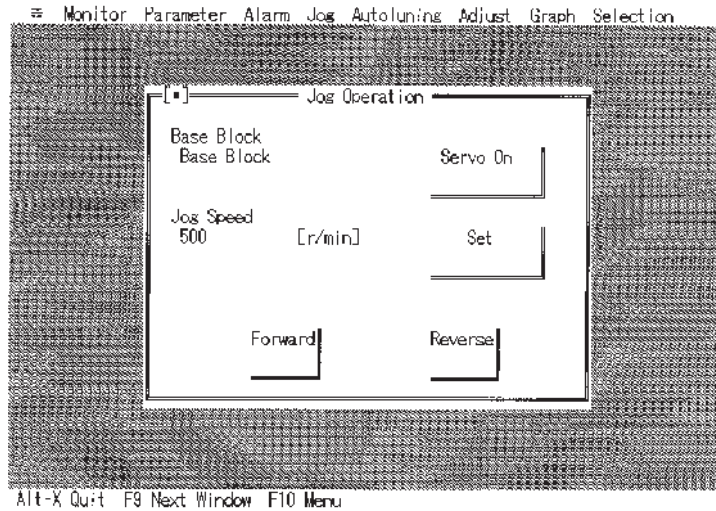
HA/LA, UT, UE

Jog speed enables normal or reverse motor rotation.

■ Operating Jog

1. Opening the Jog Operation Window

- ☞ Click on or select [Jog] - [Jog Operation].
Jog Operation window opens.



2. Selecting [Servo On] (See note 1.)

- ☞ Click on [Servo On].
- ☞ Select [Servo On] using the Tab Key. Press the Space Key or Return Key.
[Run] is displayed.

3. Selecting Rotation Direction (Refer to the next section to change jog speed)

- ☞ Click on [Forward] or [Reverse].
- ☞ Select [Forward] or [Reverse] using the Tab Key.

4. Rotating the Motor

- ☞ Click on [Forward] or [Reverse]. The mouse toggle function allows switching between Forward and Reverse.
- ☞ Select [Forward] or [Reverse], and press the Space Key. The Space Key toggle function allows switching between Forward and Reverse. When repeating the toggle function, the motor may stop momentarily. This is caused by a momentary interruption in the command input.

5. Selecting [Servo Off] (See note 2.)

- ☞ Click on [Servo Off].
- ☞ Select [Servo Off] using the Tab Key. Press the Space Key or Return Key.

6. Closing the Jog Operation window

- ☞ Click on [•] in the upper-left part of the window.
- ☞ Press the Escape Key.

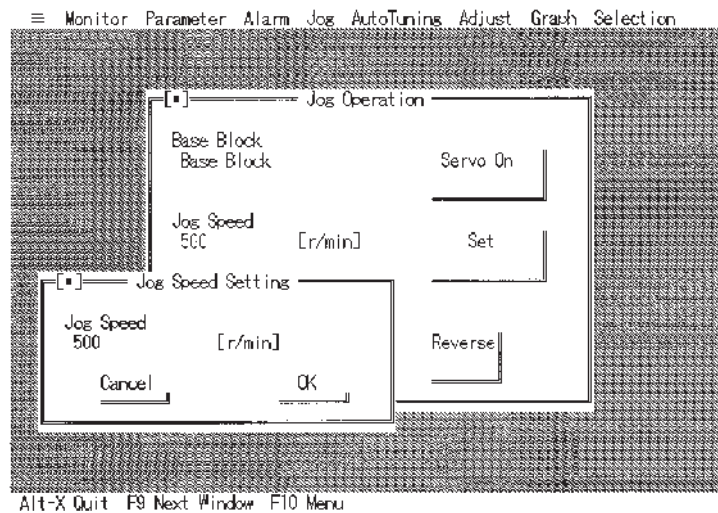
Note 1. When bit No. 0 of the set up parameter No. 1 is set to “1,” there is no need to select [Servo On] because the Servo Driver is normally in servo on status. The situation is the same when the RUN signal is input (ON) from outside.

- Note 2.** Even when bit No. 0 of the set up parameter No. 1 is set to “1,” the [Servo Off] switch is valid. However, the Servo Driver will turn back ON if the Jog Operation window is closed. When the RUN signal is input (ON) from outside, the [Servo Off] switch is invalid regardless of parameter setting.
- Note 3.** When the Servo Driver is turned ON/OFF with the external signal, the [Base Block] and [Run] displayed inside the Jog Operation window may not be consistent with its real situation.

■ **Changing Jog Speeds**

1. Opening the Job Speed Setting Window

- ☞ Click on [Set].
- ☞ Select [Set] using the Tab Key. Press either the Space Key or Return Key.
Jog Speed Setting window opens.



2. Setting the Jog Speeds (See Note 1)

- ☞ Click on [Jog Speed]. Type the desired jog speed setting, then press the Return Key.
- ☞ Select [Jog Speed] using the Tab Key. Type the desired jog speed setting, then press the Return Key.

Note [OK] can be selected instead of pressing the Return Key.

Jog Speed Setting window is closed and Jog Operation window is restored.

- Confirm that the [Jog Speed] display has been changed.
- Changing the jog speed also changes user parameter of the Servo Driver. However, parameters registered on the IBM PC/AT or compatible memory will remain unchanged.

Note 1. The UE model cannot make jog speed changes, and the jog speed is fixed to 500 rpm.

5-8 Auto Tuning

HA/LA, UT, UE

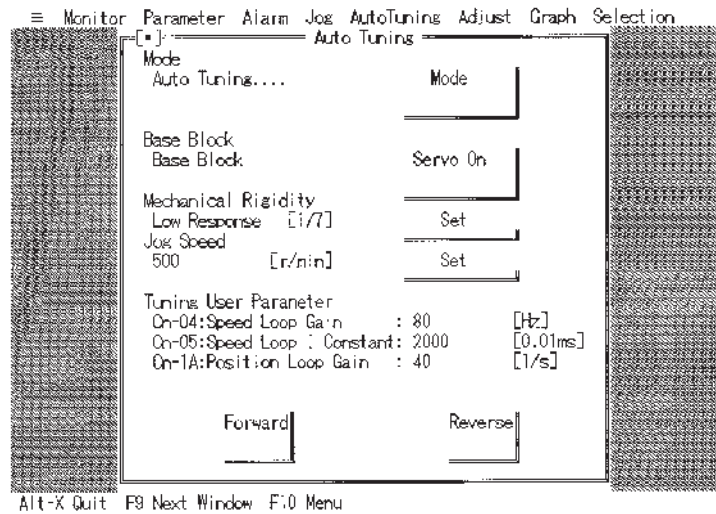
Auto tuning automatically adjusts the Servo System Gain according to set values of mechanical rigidity.

■ Auto Tuning

1. Opening the Auto Tuning Window


  Click on or select [Auto Tuning] - [Auto Tuning].

Auto Tuning window opens.

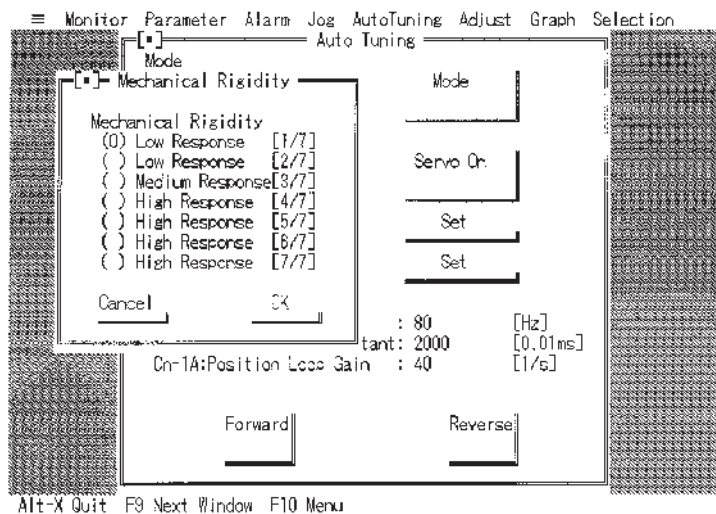


2. Opening the Mechanical Rigidity Setting Window

 Click on [Set] to the right of [Mechanical Rigidity].

 Select [Set] to the right of [Mechanical Rigidity] using the Tab Key. Press the Space Key or Return Key.

Mechanical Rigidity Setting window opens.



3. Setting the Mechanical Rigidity

- ☞ Click on the desired mechanical rigidity setting. Click on [OK].
- ☞ Select the [Mechanical Rigidity] list using the Tab Key. Select the mechanical rigidity using the Up and Down Cursor Keys, then press the Return Key.

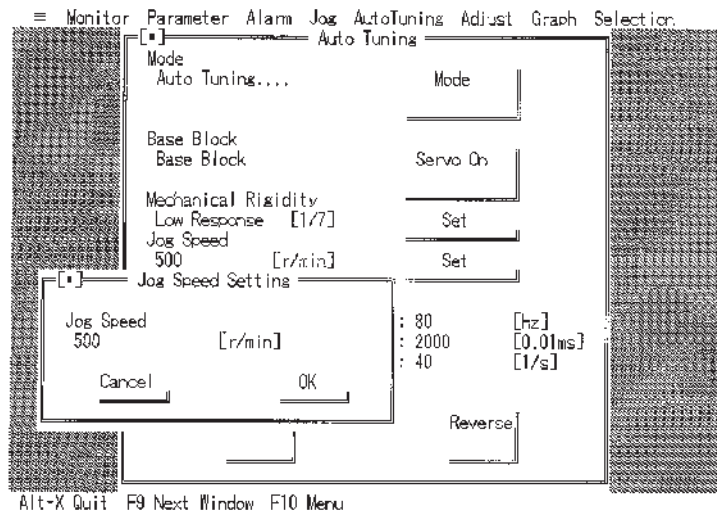
Mechanical Rigidity window closes and Auto Tuning window is restored.

4. Opening the Jog Speed Setting Window

- ☞ Click on [Set] to the right of [Jog Speed].
- ☞ Select [Set] to the right of [Jog Speed] using the Tab Key. Press the Space Key or Return Key.

Note The UE model cannot make jog speed changes, and the jog speed is fixed to 500 rpm.

Jog Speed Setting window opens.



5. Setting the Jog Speed

- ☞ Click on [Jog Speed]. Type the desired jog speed setting, then press the Return Key.
- ☞ Select [Jog Speed] using the Tab Key. Type the desired jog speed settings, then press the Return Key.

Note 1. [OK] can be selected instead of pressing the Return Key.

Note 2. In general, the jog speed should be at least 500 (r/min).

Note 3. The jog speed set here is not written to the Servo Driver’s User Parameter.

6. Selecting the [Servo On]

- ☞ Click on [Servo On].
- ☞ Select [Servo On] using the Tab Key. Press the Space Key or Return Key.

Note When bit No. 0 of the setup parameter No. 1 is set to “1,” there is no need to select [Servo On] because the Servo Driver is normally in servo on status. The situation is the same as when the RUN signal is input (ON) from outside.


Run is displayed.


7. Selecting Direction of Rotation

- ☞ Move the mouse cursor to [Forward] or [Reverse].

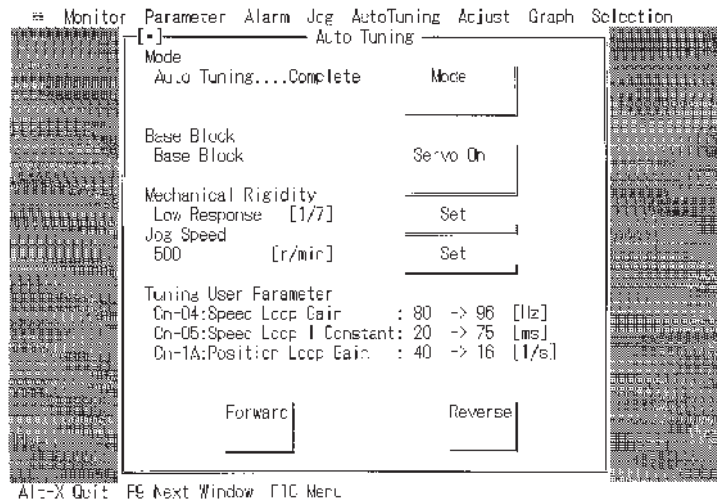
 Select [Forward] or [Reverse] position using the Tab Key.

8. Carrying out Auto Tuning

 Keep pressing the mouse button over [Forward] or [Reverse]. When auto tuning is complete, the [Mode] column will display [Auto Tuning Complete]. When [Complete] is displayed, release the button.

 Keep pressing the Space Key over [Forward] or [Reverse]. When the motor stops, release the Space Key. When the Space Key is released, the [Mode] column will display [Auto Tuning Complete].

Gain that has been changed is displayed in the [Tuning User Parameter] selection.




9. Do You Want to Perform Auto Tuning Again?

a) Closing the Auto Tuning Window

 Click on [•] in the upper-left part of the window.

 Press the Escape Key.

b) Selecting the Mode

 Click in [Mode].

 Select [Mode] using the Tab Key. Press the Space Key or Return Key. [Auto Tuning] appears in the [Mode] column.

10. Return to 2. Opening the Mechanical Rigidity Setting Window.

Note 1. Auto tuning can be performed only when the [Auto Tuning] message appears in the [Mode] column.

Note 2. Refer to operation manuals of relevant products for the mechanical rigidity setting criteria.

Note 3. Even when bit No. 0 of the setup parameter No. 1 is set to “1,” if the [Mode] column displays [Auto Tuning], the [Servo Off] switch is valid. However, when the Auto Tuning window is closed, the Servo Driver is turned back ON.

When the RUN signal is input (ON) from outside, the [Servo Off] switch will become invalid regardless of parameter setting.

Note 4. When Servo Driver is turned ON/OFF by the external signal, the [Base Block] and [Run] displayed in the [Base Block] column may not be consistent with its real situation.

5-9 Adjustment

HA/LA, UT

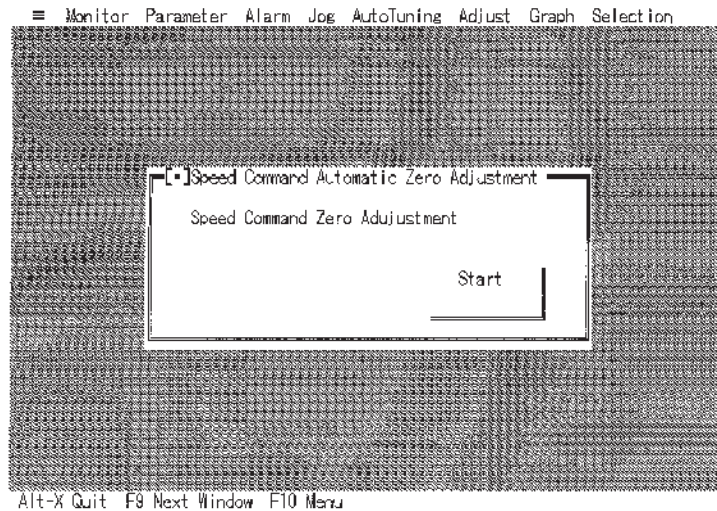
Adjustment is used for analog input. This function is not used for pulse-train inputs. Adjustment can be used for the following models.

Speed Command Automatic Zero Adjustment	HA/LA UT
Speed Command Manual Zero Adjustment	HA/LA UT
12 bit Absolute Encoder Set Up	HA/LA

■ Speed Command Automatic Zero Adjustment Method HA/LA, UT

1. Opening the Speed Command Automatic Zero Adjustment Window

- ☞ Click on or select [Adjust] - [Speed Command Automatic Zero Adjustment].
Speed Command Automatic Zero Adjustment window opens.



2. Setting the Voltage of Speed Command Input to 0 V

- Type the voltage of speed command input (REF) with which the number of rotations becomes zero (normally 0 V).

3. Selecting [Start]

- ☞ Click on [Start].
- ☞ Press the Space Key or Enter Key.

Makes an automatic zero adjustment and displays [Complete] upon completion.

- If [Speed Command Zero AdjustComplete] appears, the adjustment has been completed.

4. Closing the Speed Command Zero Adjustment Window

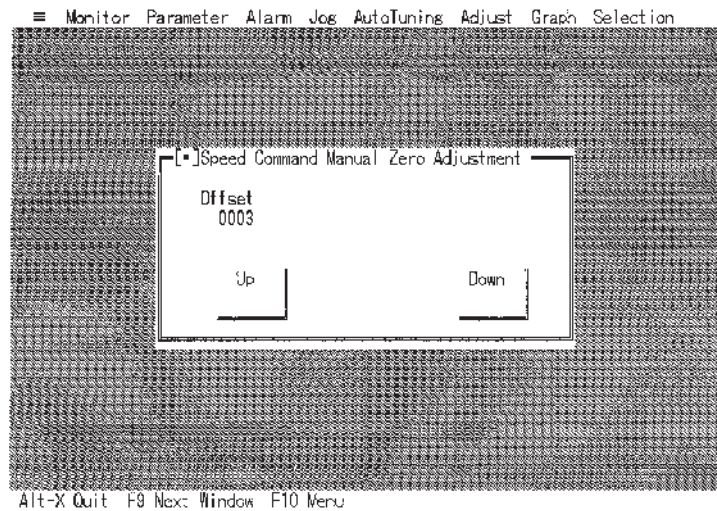
- ☞ Click on [•] in the upper-left part of the window.
- ☞ Press the Escape Key.

■ Speed Command Manual Zero Adjustment Method HA/LA, UT

1. Opening the Speed Command Manual Zero Adjustment Window

- ☞ Click on or select [Adjust] - [Speed Command Manual Zero Adjustment].

Speed Command Manual Zero Adjustment window opens.



2. Turning the Servo ON

- Input the Run signal (ON). When bit No. 0 of setup parameter No. 1 is set to “1,” this operation is not required.

3. Setting the Voltage of Speed Command Input to 0 V

- Type voltage of speed command input (REF) with which the number of rotations becomes zero (normally 0 V).

4. Setting the Direction that Provides Offset


- Select [Up] (providing offset on positive side) or [Down] (providing offset on negative side).


 Move the mouse cursor over [Up] or [Down].

 Select [Up] or [Down] using the Tab Key.

5. Adjusting Offset

- Adjust the offset so that motor or machine stops completely.

 Click on [Up] or [Down]. The offset value increases or decreases by 1 for every click.

 Press the Space Key. The offset value increases or decreases by 1 each time the Space Key is pressed.

6. Closing the Speed Command Manual Zero Adjustment Window

 Click on [•] in the upper-left part of the window.

 Press the Escape Key.

■ 12 bit Absolute Encoder Setup Method HA/LA

This function is used to reset alarms (A.81, A.82) relating to the Absolute Encoder. In addition, it is valid when the following conditions are met.

- Servo Driver model is R88-UA□□HA/LA
- Starting of the Computer Monitor Software or Servo Driver selection.
- Selection is performed under condition that the Absolute Encoder has been set for the setup parameters.

Caution When 12 bit Absolute Encoder Alarm Reset is performed, never fail to reset to the home position. Otherwise, injury or machine damage may occur.



Note 1. When 12 bit Absolute Encoder Alarm Reset is performed, the number of rotations of the Absolute Encoder is reset to 0. If the machine system is operated without resetting its home position, it will be positioned at a location different than originally intended. This situation, therefore, is very dangerous.

Note 2. 12 bit Absolute Encoder Alarm Reset is used for resetting the following events from the IBM PC/AT or compatible:

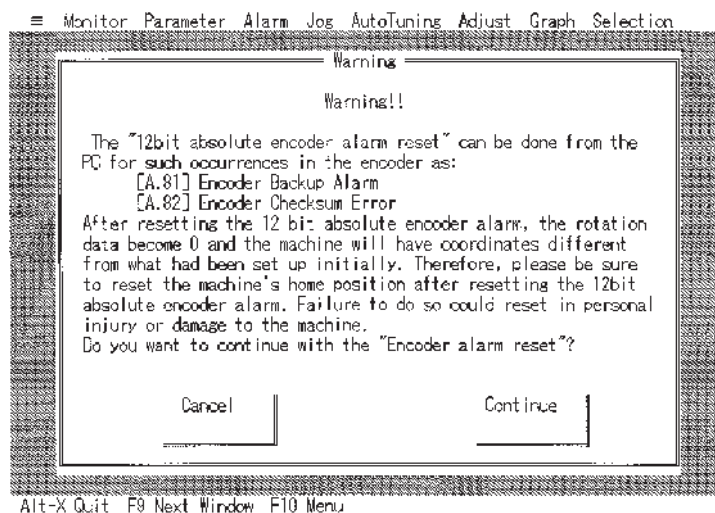
- A.81 Encoder Backup Alarm
- A.82 Encoder Checksum Error

These events originate from the Encoder.

1. Selecting [Adjust] - [12 bit Absolute Encoder Alarm Reset]


  Click on or select [Adjust] - [12 bit Absolute Encoder Alarm Reset].

Warning window opens.



2. Do You Want to Perform Setup?

a) Select [Cancel]

 Click on [Cancel].

 Press the Escape Key.

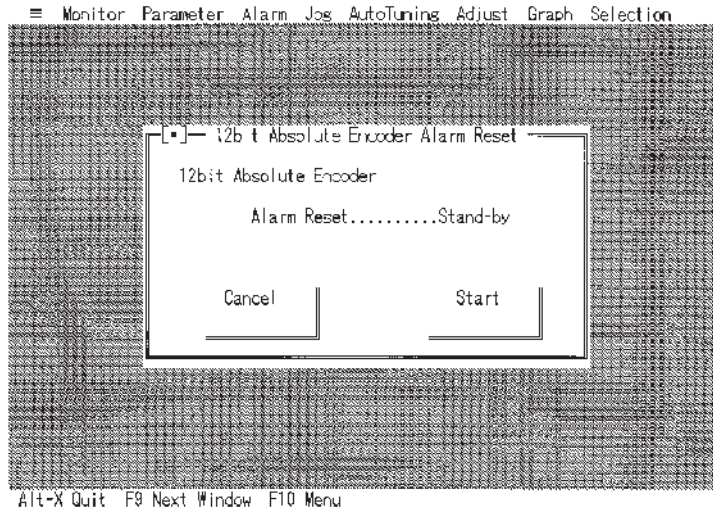
Warning window closes.

b) Select [Continue]


 Click on [Continue].

 Press the Space Key or Return Key.

[Alarm Reset Stand-by] is displayed.



3. Select [Start]

 Click on [Start].

 Press the Space Key or Return Key.

[Alarm Reset Resetting] is displayed.

[Alarm Reset Complete] is displayed.

12 bit Absolute Encoder Alarm Reset window closes.

5-10 Graph

HA/LA, UT, UE

This function retrieves and displays time series data such as torque command, speed command, speed feedback, and position deviation.

This function retrieves and displays the ON/OFF status of I/O signals as time series data.

It enables the storage of retrieved data on a disk or reading of data from a disk.

It can be used for sending a graph to the printer.

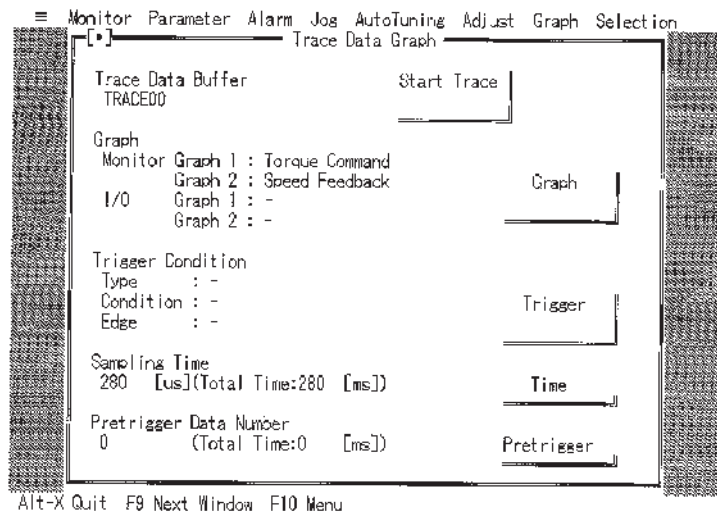
The disk and directory for Save/Read is set in *Changing the Directory in 5-3 System Monitor*. When a disk or directory is used for a data graph, select [≡] - [Directory Change] and set it up in advance.

5-10-1 Trace Data Graph

1. Opening the Trace Data Graph Window

Click on or select [Graph] - [Trace Data].

Trace Data Graph window opens.



2. Setting Up Trace Data Buffer

- Trace data buffer refers to the filename to which the data graph is saved to.

Click on [Trace Data Buffer]. Type the file name from the keyboard.


Select [Trace Data Buffer] using the Tab Key, then type the file name from the keyboard.

3. Selecting the Graph (If there is no need to select Graph, go to the next step, *Selecting the Trigger*)



- Data which can be displayed consists of the data graph (2 types) and ON/OFF data of the I/O signal (2 types).
- Either Torque Command or Speed Command can be selected for Graph 1 of the data graph. Either Speed Feedback or Position Deviation (in case of pulse-train inputs) can be selected for Graph 2.
- (0) refers to data selected for Graphs 1 and 2, and [X] refers to data selected for the I/O.

Click on [Graph] to open the Selection of Graph window. Click on the data to be displayed. When the setting is finished, click on [OK].

Selection of Graph window closes and Trace Data Graph window is restored.

-  Select [Graph] using the Tab Key. Press the Space Key or Return Key. The Selection of Graph window opens. Select [Graph 1] using the Tab Key, then select the data using Up and Down Cursor Keys. Select [Graph 2] using the Tab Key, then select the data using Up and Down Cursor Keys (for analog inputs, the cursor moves to the next I/O table). Move the cursor to the I/O table using the Tab Key and select the I/O to be displayed. Move the cursor using Up and Down Cursor Keys and select the I/O by pressing the Space Key. [X] is a selected I/O. If the Space Key is pressed again while the cursor is at [X], this [X] changes to [] and the selection is cancelled. Select [OK] when setting is complete. Selection of Graph window closes and Trace Data Graph window is restored.

4. Selecting the Trigger (If there is no need to select Trigger, go to the next step, *Setting the Time.*)

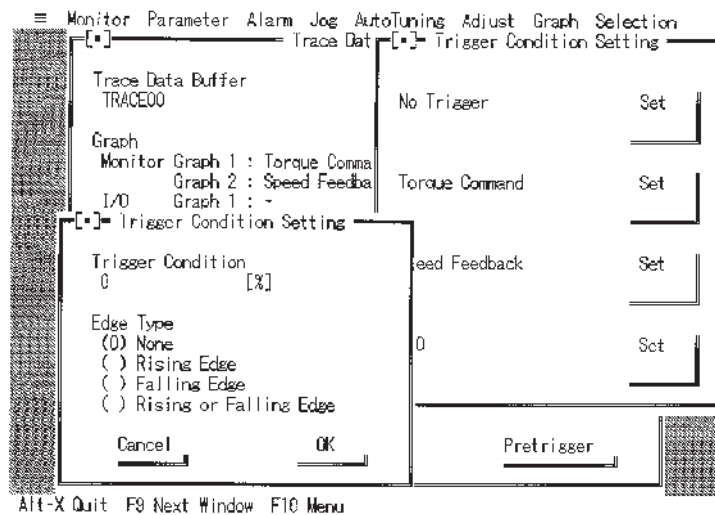
-  Click on [Trigger]. The Trigger Conditions Setting window opens. Click on [Set] to the right of the item to be set.
-  Select [Trigger] using the Tab Key. Press the Space Key or Return Key. The Trigger Condition Setting window opens. Select [Set] displayed to the right of the item to be set using the Tab Key.


In Case of No Trigger

- Trigger Condition Setting window is closes and Trace Data Graph window is restored.
- All [Trigger Condition] columns become “-”.

When Torque Command is Used as Trigger

- Trigger Condition Setting window for torque opens.

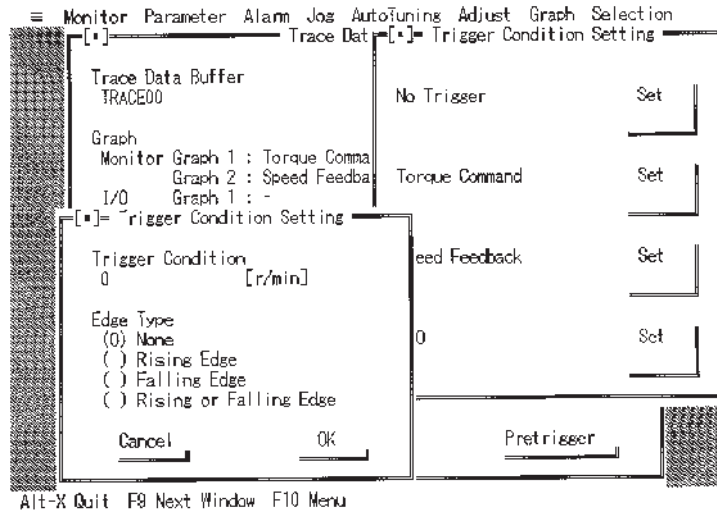


-  Click on [Trigger Condition] and type its numerical value. Click on one from among the [Edge Type]. Click on [OK]. Trigger Condition Setting window for torque and Trigger Condition Setting window close. Trace Data Graph window is restored.

- ☞ Select [Trigger Condition] using the Tab Key and type its numerical value.
- Select the [Edge Type] list using the Tab Key and select an edge type using the Up and Down Cursor Keys.
- Select [OK] using the Tab Key, then press the Space Key or Return Key.
- Trigger Condition Setting window for torque and Trigger Condition Setting window close. Trace Data Graph window is restored.

When Speed Feedback is Used as Trigger

- The Trigger Condition Setting window for Speed Feedback opens.

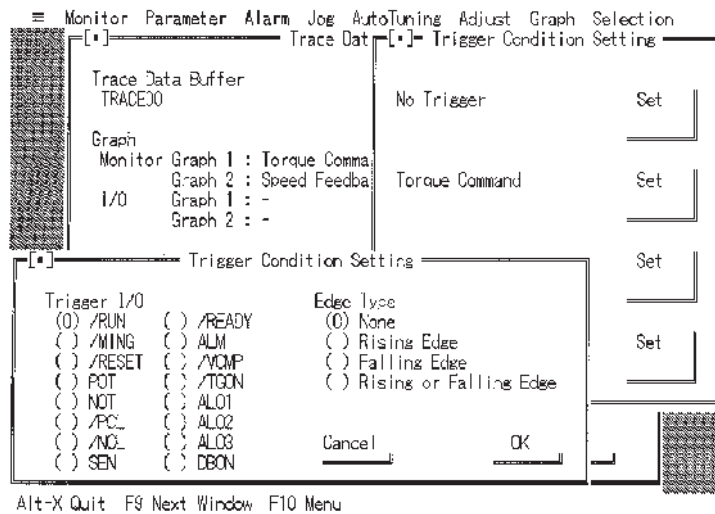


- ☞ Click on [Trigger Condition], then type its numeral value.
- Click on one from among the [Edge Type].
- Click on [OK].
- Trigger Condition Setting window for speed feedback and Trigger Condition Setting Window close. Trace Data Graph window is restored.

- ☞ Select [Trigger Condition] using the Tab Key and type its numerical value.
- Select the [Edge Type] list using the Tab Key and select an edge type using the Up and Down Cursor Keys.
- Select [OK] using the Tab Key, then press the Space Key or Return Key.
- Trigger Condition Setting window for speed feedback and Trigger Condition Setting window close. Trace Data Graph window is restored.

When I/O is Used as Trigger

- The Trigger Condition Setting window for I/O opens.



Click on the I/O to be used as trigger.
 Click on one from among the [Edge Type].
 Click on [OK].
 Trigger Condition Setting window for I/O and the Trigger Condition Setting window close. Trace Data Graph window is restored.

Select [Trigger I/O] using the Tab Key, then select the I/O to be used as a trigger by using the Cursor Keys.
 Select the [Edge Type] list using the Tab Key, then select an edge type using the Up and Down Cursor Keys.
 Select [OK] using the Tab Key, then press the Space Key or Return Key.
 Trigger Condition Setting window for I/O and Trigger Condition Setting window close. Trace Data Graph window is restored.

5. Setting the Time (If there is no need to set Time, go to the next step, *Setting the Pretrigger.*)

- Setting the sampling time
- Sampling time can be set arbitrarily in units of 250 μs.

Click on [Time].
 The Sampling Time Setting window opens.
 Click on [Up] or [Down]. The sampling time increases or decreases by 250 μs with each click.
 Click on [OK].
 Sampling Time Setting window closes and Trace Data Graph window is restored.

Select [Time] using the Tab Key, then press the Space Key or Return Key.
 The Sampling Time Setting window opens.
 Select [Up] or [Down] using the Tab Key.
 Press the Space Key.
 The sampling time increases or decreases by 250 μs with every click.
 Select [OK] using the Tab Key, then press the Space Key or Return Key.
 Sampling Time Setting window closes and Trace Data Graph window is restored.

6. Setting the Pretrigger (If there is no need to set Pretrigger, go to the next step, *Starting the Trace*.)

- Pretrigger setting sets how much data is to be stored prior to the trigger.



Click on [Pretrigger].

The Pretrigger Setting window opens.

Click on one from among the multipliers.

Click on [Up] or [Down]. The number of data items increases or decreases by the number set by [Multiplier] with each click.

Click on [OK].

Pretrigger Setting window closes and Trace Data Graph window is restored.



Select [Pretrigger] using the Tab Key, then press the Space Key or Return Key.

The Pretrigger Setting window opens.

Select [Multiplier] using the Tab Key, then select the multiplication rate using the Up and Down Cursor Keys.

Select [Up] or [Down] using the Tab Key.

Press the Space Key.

The number of data items increases or decreases by the number set by [Multiplier] each time the Space Key is pressed.

Select [OK] using the Tab Key, then press the Space Key or Return Key.

Pretrigger Setting window closes and Trace Data Graph window is restored.

7. Starting the Trace



Click on [Start Trace]. The Data Trace window opens.

Click on [Start]. The program is in [Trigger Wait] status.



Select [Start Trace] using the Tab Key, then press the Space Key or Return Key.

The Data Trace window opens.

Press the Space Key or Return Key.

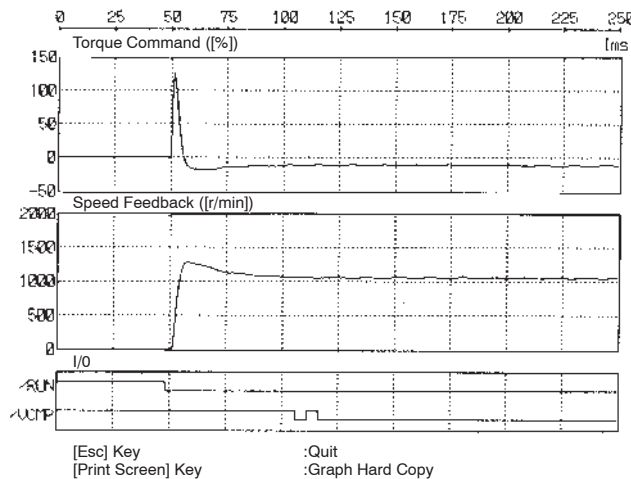
The machine is in [Trigger Wait] status.

Note 1. If a trigger has not been found, the program remains in the trigger wait status. In this case, press the Escape Key.

Data Trace window closes and Trace Data Graph window is restored.

Note 2. If a trigger has been found, [Data Trace Data Transmitting] is displayed and data is transmitted from the Servo Driver to the IBM PC/AT or compatible.

Graph is displayed.



- To print this graph, press the Print Screen Key of the IBM PC/AT or compatible.

8. Exiting the Graph Display

- Press the Escape Key.

9. Do You Want to Save the Data Graph?

- Select [Save]
- Select [Cancel]

10. Closing the Trace Data Graph Window

Click on [•] in the upper-left part of the window.

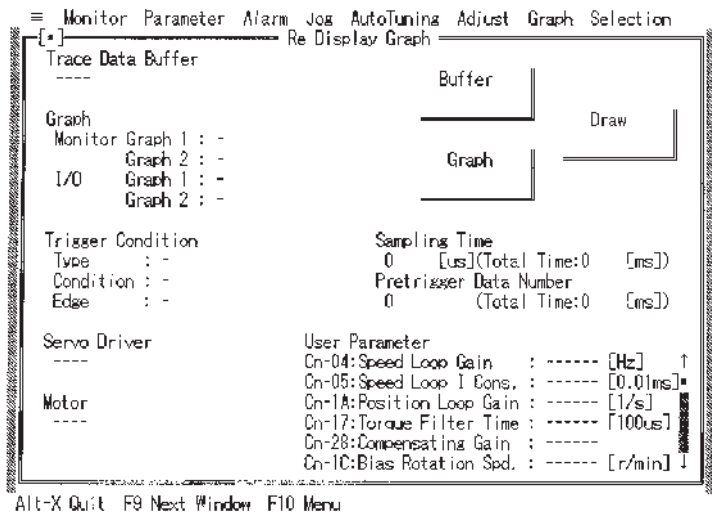
Press the Escape Key.

5-10-2 Re Display Graph

1. Opening the Re Display Graph Window

Click on or select [Graph] - [Re Display Graph].

The Re Display Graph window opens.



2. Selecting the Trace Buffer



Click on [Buffer].

The Graph Buffer Load window opens.

Double-click on the file name of the data to be displayed (or after clicking on the file name, click on [PC←Disk]).

Data is read from the disk and Re Display Graph window is restored.



Select [Buffer] using the Tab Key, then press the Space Key or Return Key.

The Graph Buffer Load window opens.

Select [Trace Data Buffer] using the Tab Key, then select the file name using the Up and Down Cursor Keys, and press the Return Key (or select the file name, then select [PC←Disk] and press the Space Key or Return Key).

Data is read from the disk and the Re Display Graph window is restored.

3. Selecting the Graph (If there is no need to select Graph, go to the next step, *Displaying the Graph.*)

- The I/O data to be displayed can be set again.



Click on [Graph].

The Selection of Graph window opens.

Click on the I/O to be displayed.

[X] is a selected I/O. Click on [X] deselects it and changes [X] to [].

Click on [OK].

Selection of Graph window closes and Re Display Graph window is restored.



Select [Graph] using the Tab Key, then press the Space Key or Return Key.

The Selection of Graph window opens.

Select the [I/O (Up to 2)] list using the Tab Key, then select the I/O using the Cursor Keys and press the Space Key.

[X] is a selected I/O.

[X] is a selected I/O. Pressing the Space Key changes [X] to [] and deselects the corresponding I/O.

Select [OK], then press the Space Key or Return Key.

Selection of Graph window closes and Re Display Graph window is restored.

4. Displaying the Graph

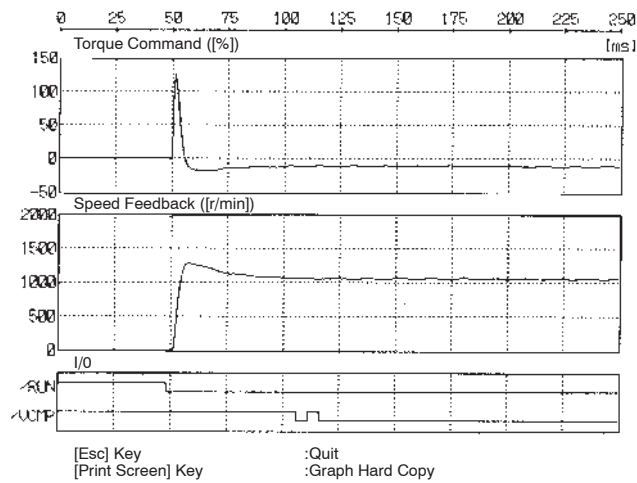


Click on [Draw].



Select [Draw] using the Tab Key, then press the Space Key or Return Key.

Graph is displayed.



- To print this graph, press the Print Screen Key of the IBM PC/AT or compatible.

5. Exiting the Graph Display

- Press the Escape Key.

6. Closing the Re Display Graph Window

- ☞ Click on [•] in the upper-left part of the window.
- ☞ Press the Escape Key.

5-11 Servo Driver Selection

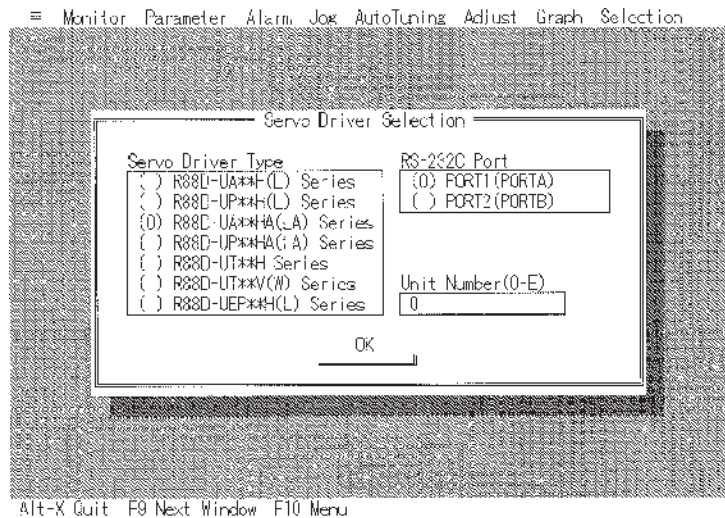
H/L HA/LA UT

The Servo Driver, set when the Computer Monitor Software was started, can be changed.

1. Opening the Servo Driver Selection window

Click on or select [Selection] - [Servo Driver Selection].

The Servo Driver Selection window opens.



2. Selecting the Servo Drive Type

Click on the Servo Drive type desired.

Select [Servo Driver Type] using the Tab Key, then select the Servo Driver using the Up and Down Cursor Keys.

3. Selecting the RS-232C Port

Click on the port desired.

Select [RS232C Port] using the Tab Key, then select the port using the Down Cursor Key.

4. Setting the Unit No. for **HA/LA** and **UT**

Click on [Unit Number].

Type the Unit No. from the keyboard, then press the Return Key (or click on [OK] instead of pressing the Return Key).

Select [Unit Number] using the Tab Key.

Type the Unit No. from the keyboard, then press the Return Key (or instead of pressing the Return Key, select [OK] using the Tab Key and then press the Space Key or Return Key).

Note No unit number setting is available for **H/L** or **UE**. (Multiple axis communication is not possible.)

Monitor display screen appears.

- The monitor display screen appears if the power supply of the Servo Driver has been turned ON, and the selected Servo Driver type matches the connected Servo Driver type.

- When the power supply of the Servo Driver has been turned ON, but the Servo Driver type does not match, an error message is displayed.
- The Servo Driver type is not discriminated between the R88D-UA□□H(L) and R88D-UP□□H(L), and the Servo Driver Select window is ignored.
- When the power supply of the Servo Driver has been turned OFF, the off line edit described in the following section can be performed.

5-12 Off Line Edit


H/L, HA/LA, UT, UE


Off line edit is used for editing U-series parameters or creating a Re Display Graph under condition that the power supply of Servo Driver is OFF or that the Servo Driver is not connected to an IBM PC/AT or compatible.

1. Start the Computer Monitor Software after switching the Servo Driver's power supply OFF or disconnecting the Servo Driver from an IBM PC/AT or compatible.

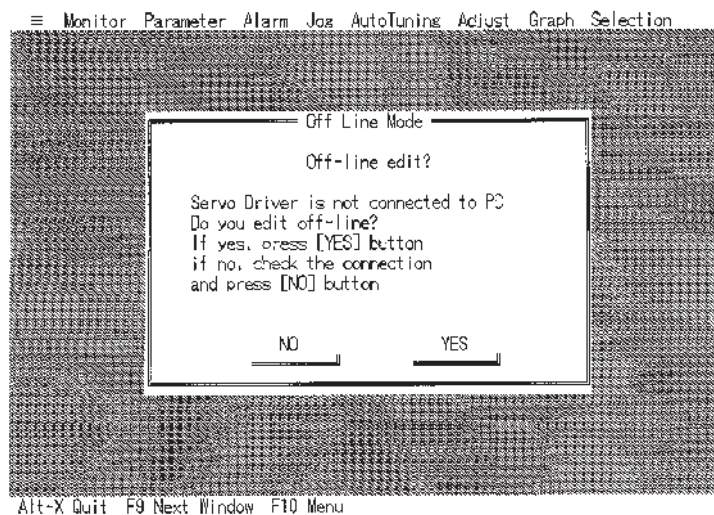
The Servo Driver Selection window opens.

2. Selecting the Servo Driver Type


 Click on the Servo Driver desired.
Click on [OK].

-  Select [Servo Driver Type] using the Tab Key, then select the Servo Driver desired using the Up and Down Cursor Keys.
Select [OK] using the Tab Key.
Press the Space Key or Return Key.

The Off Line Mode window opens.



3. Selecting Off Line Edit


 Click on [YES].


-  Press the Space Key or Enter Key. (With [YES] selected).

- When the R88D-UT□□W/V (conforming to EN standards) is selected, the Control Type Selection window is displayed. Select one of the two shown below:

[Speed/Torque Control]

[Position Control]

 Click on the [Control Type] desired.
Click on [OK].

-  Select [Control Type] using the Tab Key, then select the control type using the Up and Down Cursor Keys.
Select [OK] using the Tab Key.
Press the Space Key or Return Key.

4. Performing the Off Line Edit

Functions which can be used for the off line edit are shown below:

- Parameter Setting
 - Editing Parameters See section 5-5-1
 - Transferring Parameters
(Between IBM PC/AT or Compatible and Disk) See section 5-5-2
 - Clearing Parameters See section 5-5-5
 - Printing Parameters See section 5-5-6
- Re Display Graph See section 5-10-2
- Servo Driver Selection See section 5-11

**Anexo II. Ficha técnica da Placa de Aquisição de dados
utilizada. PCI 1200, *National Instruments***

Low-Cost Multifunction I/O – 100 kS/s, 12-Bit, 8 Analog Inputs

1200 Family

1200 Family

PCI-1200
DAQCard-1200
Lab-PC-1200
Lab-PC-1200AI
DAQPad-1200

Analog Inputs

8 single-ended, 4 differential channels
100 kS/s sampling rate
12-bit resolution

Analog Output (not for Lab-PC-1200AI)

2 channels, 12-bit resolution

Digital I/O

24 (5 V/TTL) lines in 8-bit ports

Counter/Timers

Three, 16-bit resolution

Triggering

Digital

Driver Software

NI-DAQ
Windows 2000/NT/9x
Mac OS*
*not for all hardware,
refer to page 192

Application Software

LabVIEW
LabWindows/CVI
ComponentWorks
VirtualBench
Measure
BridgeVIEW
Lookout

Calibration Certificate Included!

(refer to page 216)



Make sure you consider our new low-cost 12-bit E Series products – refer to page 239.

Bus	Analog Inputs	Resolution	Sampling Rate	Input Range	Analog Outputs	Resolution	Output Rate	Output Range	Digital I/O	Counter/Timers	Triggers
PCI, PCMCIA, ISA, Parallel Port	8 SE/4 DI	12 bits	100 kS/s	up to ± 5 V	2 ¹	12 bits ¹	1 kS/s ¹	± 5 V	24	3, 16-bit	Digital

¹No analog outputs on Lab-PC-1200AI

Table 1. 1200 Family Channel, Speed, and Resolution Specifications (refer to page 314 for more detailed specifications)

Overview

The 1200 Family devices are low-cost, multifunction I/O devices with up to 100 kS/s, 12-bit performance on 8 single-ended or 4 differential analog inputs. The 1200 Family features digital triggering capability; three 16-bit, 8 MHz counter/timers; two 12-bit analog outputs; and 24 digital I/O lines. The 1200AI does not include the two analog outputs.

Hardware Analog Input

The 1200 Family has eight single-ended or 4 differential 12-bit analog input channels. The voltage input range is software programmable for 0-10 V (unipolar) or ± 5 V (bipolar). A software-programmable gain amplifier has gain selections of 1, 2, 5, 10, 20, 50, or 100.

The single-channel sampling rate of the ADC is 100 kS/s. The PCI, DAQCard, and Lab-PC versions can sustain this rate, but the DAQPad cannot. The sustainable data transfer rate to a standard PC parallel port is 25 kS/s (depending on your computer). If you need an external solution with faster sampling, consider the DAQPad-6020E for USB, (100 kS/s, page 241) and the DAQPad-6070E for IEEE 1394 (1.25 MS/s, page 226). When scanning multiple channels, the maximum sampling rate for all 1200 family products is 83.3 kS/s at a gain of 1.

Data acquisition with the 1200 Family devices is available in three modes: 1) continuous acquisition of a single channel, 2) multichannel acquisition with continuous scanning, or 3) multichannel acquisition with interval scanning with a resolution of 1 μ s.

The 1200 Family devices have two hardware triggering modes – pretrigger mode and posttrigger mode. In pretrigger mode, the device collects samples until a trigger is received at the external trigger input, and then continues to collect a specified number of samples. In posttrigger mode, the device collects a specified number of samples after it receives a trigger.

Analog Output

The 1200 Family has two double-buffered 12-bit DACs that are connected to two analog output channels. You can independently configure each channel through software for unipolar (0-10 V) or bipolar (± 5 V) operation. The resolution of the 12-bit DAC is 2.44 mV in both polarities.

ACH0	1	2	ACH1
ACH2	3	4	ACH3
ACH4	5	6	ACH5
ACH6	7	8	ACH7
AISENSE/AINGD	9	10	DAC0OUT ¹
AGND	11	12	DAC1OUT ¹
DGND	13	14	PA0
PA1	15	16	PA2
PA3	17	18	PA4
PA5	19	20	PA6
PA7	21	22	PB0
PB1	23	24	PB2
PB3	25	26	PB4
PB5	27	28	PB6
PB7	29	30	PC0
PC1	31	32	PC2
PC3	33	34	PC4
PC5	35	36	PC6
PC7	37	38	EXTTRIG
EXTUPDATE	39	40	EXTCONV ⁺
OUTB0	41	42	GATB0
OUTB1	43	44	GATB1
CLKB1	45	46	OUTB2
GATB2	47	48	CLKB2
+5V	49	50	DGND

¹ Not available on Lab-PC-1200AI

Figure 1. 1200 Family I/O Connector

Data Acquisition

Low-Cost Multifunction I/O – 100 kS/s, 12-Bit, 8 Analog Inputs

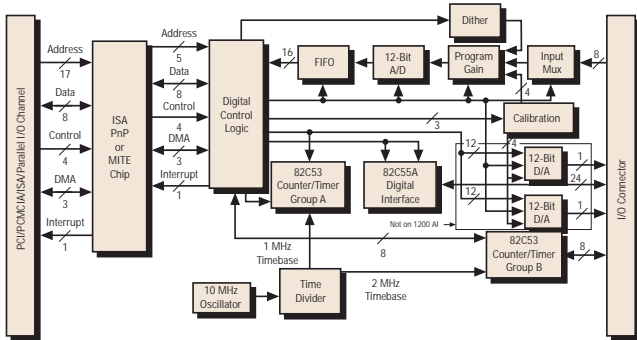


Figure 2. 1200 Family Hardware Block Diagram

DAQPad-1200 Power

The DAQPad-1200 is powered by any 9 to 42 VDC source. With the AC adapter unit included, you can power the DAQPad-1200 from any standard 120 VAC or 230 VAC source.

The optional BP-1 rechargeable battery pack is the same size as the DAQPad-1200 and powers it for 11 hours. The charger unit included with the BP-1 is not CE certified.

BNC and Breadboard Accessories

The BNC-2081 accessory has labeled BNC connections for the analog signals of the 1200 Family. The SC-2071 and SC-2072 are breadboard accessories with labeled and numbered screw terminals, respectively; the 1200 device must be in single-ended mode to use the SC-2071.

Self-Calibration

The analog inputs and outputs of the 1200 Family have self-calibration circuitry to correct for gain and offset errors. You can use the permanently-stored factory calibration constants or store additional sets of constants for different operating conditions in the user-modifiable section of an EEPROM on the device.



Make sure you consider our new low-cost E Series products – refer to page 239.

Digital I/O

The 1200 Family devices have 24 digital I/O lines that are 5 V/TTL compatible and configurable as three 8-bit ports for input, output, bidirectional, or handshaking modes. The digital output ports of the 1200 series can sink 2.5 mA on each line.

Counter/Timer

The 1200 Family uses two 82C53 counter/timer integrated circuits, each containing three independent 16-bit counter/timers. One 82C53 is dedicated for A/D and D/A timing. The three counters on the other 82C53 are available for general time-related functions, such as clock output, pulse output, and event counting. Of these three counters, one may be required for timing of analog I/O, and one has a built-in 20 MHz source clock.

I/O Connector

The I/O connector is a 50-pin male ribbon cable connector diagrammed in Figure 1. ACH<0..7> are eight analog input channels. DAC0OUT and DAC1OUT are the two analog output channels. EXUPDATE, EXTCONV, and EXTTRIG are TTL lines through which you can externally control analog I/O. CLKB<1..2>, GATB<0..2>, and OUTB<0..2> are the clock, gate, and output of the user-available counters. PA<0..7>, PB<0..7>, and PC<0..7> are the three 8-bit digital I/O ports.

DAQPad-1200 Parallel Port Interface

The DAQPad-1200 includes a 1 m parallel port cable with two 25-pin D-Sub connectors. The DAQPad-1200 rear panel includes two parallel port connectors, Port A for connection to the PC parallel port, and Port B, for transparent pass-through connection to any other standard parallel port device.

Ordering Information

1200 Family

PCI-1200 and NI-DAQ for

Windows 2000/NT/9x.....777386-01
Mac OS777097-01

DAQCard-1200 and NI-DAQ for

Windows 2000/NT/9x.....777087-01
Mac OS777087-02

Lab-PC-1200777227-01

Lab-PC-1200AI777292-01

DAQPad-1200 for Parallel Port with AC adapter

U.S. 120 VAC.....776895-01
Universal Euro 240 VAC.....776895-31
United Kingdom 240 VAC776895-06

Includes NI-DAQ for Windows 2000/NT/9x on CD unless otherwise noted. See pages 192 and 210 for more details.

BP-1 rechargeable battery pack with charger

120 VAC776896-01
240 VAC776896-31

BNC-2081776599-90

SC-2071776358-91

SC-2072776358-92

Example Configurations

Family	DAQ Board	Cable (page 297)	Accessory (page 292)
1200	PCI-1200	NB1 (180524-10)	CB-50LP (777101-01)
	DAQCard-1200	PR50-50F (182799-01)	CB-50LP (777101-01)
	Lab-PC-1200	NB1 (180524-10)	CB-50LP (777101-01)
	Lab-PC-1200AI	NB1 (180524-10)	CB-50LP (777101-01)
1200AI	DAQPad-1200	NB1 (180524-10)	CB-50LP (777101-01)

For more detailed cable and accessory options, refer to pages 286-299.

Specifications

1200 Family

These specifications are typical at 25 °C unless otherwise stated.

Analog Input Input Characteristics

Number of channels.....	8 single-ended, or 4 differential, software selectable
Type of ADC.....	Successive approximation
Resolution.....	12 bits, 1 in 4,096
Maximum sampling rate	
PCI, DAQCard, Lab-PC.....	100 kS/s
DAQPad.....	100 kS/s to FIFO; 25 kS/s to PC with standard Centronics port

Input signal ranges

Gain (Software Selectable)	Range (Software Selectable)	
	Bipolar	Unipolar
1	±5 V	0 to 10 V
2	±2.5 V	0 to 5 V
5	±1 V	0 to 2 V
10	±500 mV	0 to 1 V
20	±250 mV	0 to 500 mV
50	±100 mV	0 to 200 mV
100	±50 mV	0 to 100 mV

Input coupling.....	DC
Maximum working voltage (signal + common mode).....	In differential or NRSE mode, the negative input/AISENSE should remain within ±5 V (bipolar) or -5 to 2 V (unipolar) of AGND except for the DAQCard-1200, where the values are ±6 V (bipolar) and -6 to 2 V (unipolar). The positive input should remain within -5 V to +10 V of AGND. For the DAQCard-1200 in RSE mode, the input signal referenced to AGND should remain within ±5 V(bipolar) or 0 to 10 V (unipolar).

Overvoltage protection	
PCI, DAQCard, Lab-PC.....	±35 V powered on, ±25 V powered off
DAQPad.....	±42 V powered on, ±15 V powered off
Inputs protected.....	ACH<0..7>

FIFO buffer size	
PCI.....	4,096 samples
DAQCard.....	1,024 samples
Lab-PC.....	512 samples
DAQPad.....	2,048 samples

Data transfers	
PCI, Lab-PC.....	DMA, interrupts, programmed I/O
DAQCard, DAQPad.....	Interrupts, programmed I/O

DMA modes	
PCI.....	Scatter-gather
Lab-PC.....	Single transfer
Dither.....	Available

Transfer Characteristics

Relative accuracy.....	±0.5 LSB typical dithered, ±1.5 LSB max undithered
DNL.....	±1 LSB max
No missing codes.....	12 bits, guaranteed
Offset error	

Error		PCI, DAQCard, Lab-PC	DAQPad
Pregain	After calibration	±10 µV max	±5 µV max
	Before calibration	±20 mV max	±15 mV max
Postgain	After calibration	±1 mV max	±360 µV max
	Before calibration	±200 mV max	±150 mV max

Gain error (relative to calibration reference)	
After calibration.....	0.02% of reading max
Before calibration.....	±2% of reading max

Amplifier Characteristics

Input bias current	
PCI, DAQCard, Lab-PC.....	±100 pA
DAQPad.....	±200 pA
Input offset current.....	±100 pA

Input impedance	
Normal powered on.....	100 GΩ in parallel with 50 pF
Powered off.....	4.7 kΩ min
Overload.....	4.7 kΩ min
CMRR.....	70 dB typical, DC to 60 Hz

Dynamic Characteristics

Bandwidth (small signal -3 dB)

Gain	PCI, DAQCard, Lab-PC	DAQPad
1 to 10	250 kHz	400 kHz
20	150 kHz	200 kHz
50	60 kHz	80 kHz
100	30 kHz	40 kHz

Settling time for full-scale step

Gain	Accuracy ±0.024% (±1 LCB)		
	PCI, Lab-PC µs typical/µs max	DAQCard µs typical/µs max	DAQPad µs max
1	10/14	-/10	25
2 to 10	13/16	-/10	25
20	15/19	12/15	25
50	27/34	25/30	25
100	60/80	60/80	60

System noise (including quantization error) in LSB_{rms}

Gain	PCI, DAQCard, Lab-PC		DAQPad	
	Dither Off	Dither On	Dither Off	Dither On
1 to 50	0.3	0.5	0.3	0.6
100	0.5	0.7	0.6	0.8

Stability

Recommended warm-up time.....	15 minutes
Offset temperature coefficient	
Pregain.....	±15 µV/°C
Postgain.....	±100 µV/°C
Gain temperature coefficient	
PCI, DAQCard, Lab-PC.....	±40 ppm/°C
DAQPad.....	±50 ppm/°C

Analog Output (not for Lab-PC-1200A)

Output Characteristics

Number of channels.....	Two voltage
Resolution.....	12 bits, 1 in 4,096
Typical update rate.....	1 kS/s (system dependent)
Type of DAC.....	Double buffered, multiplying
Data transfers.....	Interrupts, programmed I/O

Transfer Characteristics

Relative accuracy.....	±0.25 LSB typical, ±0.50 LSB max ±0.50 LSB typical, ±1.0 LSB max (DAQCard)
DNL.....	±0.25 LSB typical, ±0.75 LSB max ±0.50 LSB typical, ±0.90 LSB max (DAQCard)
Monotonicity.....	12 bits, guaranteed
Offset error	
After calibration.....	±0.2 mV max; ±0.5 mV max for DAQCard
Before calibration.....	±50 mV max; ±75 mV max for DAQCard
Gain error (relative to internal reference)	
After calibration.....	0.01% of reading max 0.02% of reading max (DAQCard)
Before calibration.....	±1% of reading max ±2% of reading max (DAQCard)

Voltage Output

Ranges.....	0 to 10 V, ±5 V, software selectable
Output coupling.....	DC
Output impedance.....	0.2 Ω typical 0.5 Ω typical (DAQCard)
Current drive.....	±2 mA 1 mA max per channel (DAQCard)
Protection.....	Short circuit to ground 80 mA momentary short-circuit protection to ground (DAQCard)

1200 Family (continued)

Power-on state
 PCI, DAQCard, Lab-PC 0 V
 DAQPad 0 V bipolar mode, 5 V unipolar mode

Dynamic Characteristics

Settling time to full-scale range (FSR)
 PCI, Lab-PC 5 μ s to ± 1 LSB
 DAQCard 20 μ s to ± 1 LSB
 DAQPad 6 μ s to ± 1 LSB

Stability

Offset temperature coefficient
 PCI, DAQCard, Lab-PC ± 50 μ V/ $^{\circ}$ C
 DAQPad ± 60 μ V/ $^{\circ}$ C
 Gain temperature coefficient
 PCI, DAQCard, Lab-PC ± 30 ppm/ $^{\circ}$ C
 DAQPad ± 10 ppm/ $^{\circ}$ C

Digital I/O

Number of channels 24 I/O (three 8-bit ports; uses the 82C55A PPI)
 Compatibility 5 V/TTL

Digital logic levels

Level	Minimum	Maximum
Input low voltage	-0.3 V	0.8 V
Input high voltage	2.2 V	5.3 V
Output low voltage ($I_{out} = 2.5$ mA)	-	0.4 V
Output high voltage ($I_{out} = 2.5$ mA)	3.7 V	-

Handshaking 2-wire, 2 ports
 Power-on state Input
 Protection -0.5 to 5.5 V powered on, ± 0.5 V powered off
 Data transfers Interrupts, programmed I/O

Timing I/O

Number of channels 3 counter/timers (uses two 82C53 STCs)
 Protection -0.5 V to 5.5 V powered on, ± 0.5 V powered off
 Resolution 16 bits
 Compatibility 5 V/TTL, counter gate and clock inputs are pulled up with 100 k Ω onboard resistors
 Base clock available 2 MHz
 Base clock accuracy ± 50 ppm max, $\pm 0.01\%$
 Maximum source frequency 8 MHz
 Minimum source pulse duration 60 ns
 Minimum gate pulse duration 50 ns

Digital logic levels

Level	Minimum	Maximum
Input low voltage	-0.3 V	0.8 V
Input high voltage	2.2 V	5.3 V
Output low voltage ($I_{out} = 2.1$ mA for PCI, Lab-PC) ($I_{out} = 4$ mA for DAQCard, DAQPad)	-	0.45 V
Output high voltage ($I_{out} = 0.92$ mA for PCI, Lab-PC) ($I_{out} = 1$ mA for DAQCard, DAQPad)	3.7 V	-

Data transfer Interrupts, programmed I/O

Digital Trigger

Compatibility 5 V/TTL
 Response Rising edge
 Pulse width 50 ns min

Bus Interface

PCI Master, slave
 DAQCard, Lab-PC, DAQPad Slave

Power Requirements

+5 VDC ($\pm 5\%$)
 PCI 425 mA
 DAQCard 150 mA, 50 mA power-down mode
 Lab-PC 185 mA
 Lab-PC-1200AI 150 mA
 +12 VDC
 DAQPad 250 mA
 Power available at I/O connector +4.65 to +5.25 V, 400 mA fused
 +5 VDC, 500 mA (DAQCard)

Physical

Dimensions
 PCI, Lab-PC 17.5 x 10.6 cm (6.9 by 4.2 in.)
 DAQCard Type II PC Card
 DAQPad 14.6 by 21.3 by 3.8 cm
 (5.8 by 8.4 by 1.5 in.)
 Weight
 DAQPad 0.77 kg (1.7 lb)
 I/O connectors 50-pin male
 Paralle port connector
 Type 2 female 25-pin D-Sub, EPP and SPP (Centronics)
 Throughput 180 kbytes/s (EPP), 41 kbytes/s (Centronics)

Environment

Operating temperature 0 to 50 $^{\circ}$ C, DAQCard should not exceed 50 $^{\circ}$ C while in PCMCIA slot
 Storage temperature -20 to 70 $^{\circ}$ C
 Relative humidity 5% to 90% noncondensing

BP-1 Rechargeable Battery Pack

Output 12 VDC, 3.2 Ah
 Run time with DAQPad 5 h loaded at 350 mA from +5 V I/O;
 11 h unloaded
 Dimensions 14.6 by 21.3 by 3.8 cm
 (5.8 by 8.4 by 1.5 in.)
 Weight 1.92 kg (4.2 lb)

Certifications and Compliances¹

CE Mark Compliance **CE**

¹The charger included with the BP-1 is not CE compliant.

Para a configuração da *DAQ* utilizada, a *PCI 1200* da *NI*, no *Labview*, procurou-se no próprio programa essa opção, disponível após concluída a instalação do mesmo. Essa opção permite entrar na zona de configuração, na qual todos os dispositivos de *hardware* são identificados.

A procura do dispositivo da Placa de Aquisição de dados é automática, sendo apenas necessário instalar os seus *drivers*. De seguida a placa *DAQ PCI-1200* foi detectada como um dispositivo da *NI*.

Após a placa ter sido reconhecida e configurada, foi-lhe atribuído um *device number*, a partir do qual se pode começar a endereçar os *VIs* implementados no *software Labview*.

Anexo III. Folha de cálculo do controlo dos parâmetros cinemáticos da Mesa / Guia. Electronic Gear = 4.

Folha de cálculo

Electronic Gear = 4, até 200mm

Requisito do Motor **4.096** pulsos /rotação

* pulsos desmultiplicados / por rotação

Passo da rosca do parafuso sem-fim: 1/2 polegada

12.700 mm

* igual a

Ch-24, G1 =	4	User Parameters
Ch-25, G2 =	1	

Aceleração média = 1,000 mm/s²

Velocidade inicial = 1,000 mm/s

G1/G2 =	4 Electronic Gear
Pulsos Electivos	1.024.00 por rotação
Rotinas no Labview	2.048.00 por rotação

Avanço (mm/rotina do Labview): 0,006 mm/rotina

Rotinas no Labview para avançar 1mm: 161,260 rotinas

Duração da Rotina (milissegundos): 6,201 milissegundos

Distância a percorrer	Nº de Rotinas no Labview	Velocidade instantânea (função da distância)	Velocidade em cada ponto (função do nº de rotinas)	Duração da Rotina calculada	tempo acumulado entre pontos	Velocidade média (função duração da rotina calculada)	Duração da Rotina arredondada	tempo acumulado entre pontos	Velocidade média (função duração da rotina arredondada)	Diferença	Diferença
										valor absoluto	percentagem
mm		mm/s	mm/s	milissegundos	milissegundos	mm/s	milissegundos	milissegundos	mm/s	mm/s	%
0	0	1,000	1,000	6,201	0,000	0,000	6,000	0,000	0,000	0,000	
1	161	1,732	1,732	3,580	788,675	1,268	4,000	806,299	1,240	-0,028	-2,19%
2	323	2,236	2,236	2,773	512,282	1,537	3,000	564,409	1,459	-0,078	-5,09%
3	484	2,646	2,646	2,344	412,589	1,751	2,000	403,150	1,691	-0,060	-3,40%
4	645	3,000	3,000	2,067	355,649	1,933	2,000	322,520	1,908	-0,025	-1,30%
5	806	3,317	3,317	1,870	317,422	2,095	2,000	322,520	2,067	-0,028	-1,33%
6	968	3,606	3,606	1,720	289,431	2,242	2,000	322,520	2,189	-0,053	-2,38%
7	1.129	3,873	3,873	1,601	267,774	2,378	2,000	322,520	2,285	-0,093	-3,92%
8	1.290	4,123	4,123	1,504	250,367	2,505	2,000	322,520	2,362	-0,142	-5,68%
9	1.451	4,359	4,359	1,423	235,976	2,624	1,000	241,890	2,480	-0,143	-5,46%
10	1.613	4,583	4,583	1,353	223,817	2,737	1,000	161,260	2,639	-0,098	-3,58%
11	1.774	4,796	4,796	1,293	213,366	2,844	1,000	161,260	2,784	-0,060	-2,11%
12	1.935	5,000	5,000	1,240	204,257	2,947	1,000	161,260	2,918	-0,029	-0,99%
13	2.096	5,196	5,196	1,193	196,225	3,046	1,000	161,260	3,042	-0,004	-0,13%
14	2.258	5,385	5,385	1,152	189,073	3,141	1,000	161,260	3,157	0,016	0,50%
15	2.419	5,568	5,568	1,114	182,650	3,233	1,000	161,260	3,264	0,031	0,95%
16	2.580	5,745	5,745	1,079	176,841	3,322	1,000	161,260	3,363	0,041	1,25%
17	2.741	5,916	5,916	1,048	171,554	3,408	1,000	161,260	3,456	0,048	1,41%
18	2.903	6,083	6,083	1,019	166,715	3,492	1,000	161,260	3,544	0,052	1,48%
19	3.064	6,245	6,245	0,993	162,264	3,573	1,000	161,260	3,625	0,052	1,45%
20	3.225	6,403	6,403	0,968	158,151	3,653	1,000	161,260	3,702	0,049	1,35%
21	3.386	6,557	6,557	0,946	154,336	3,730	1,000	161,260	3,775	0,044	1,19%
22	3.548	6,708	6,708	0,924	150,785	3,806	1,000	161,260	3,843	0,037	0,97%
23	3.709	6,856	6,856	0,905	147,468	3,880	1,000	161,260	3,908	0,027	0,71%
24	3.870	7,000	7,000	0,886	144,361	3,953	1,000	161,260	3,969	0,016	0,41%
25	4.031	7,141	7,141	0,868	141,443	4,024	1,000	161,260	4,027	0,003	0,08%
26	4.193	7,280	7,280	0,852	138,694	4,093	1,000	161,260	4,082	-0,011	-0,28%
27	4.354	7,416	7,416	0,836	136,100	4,161	1,000	161,260	4,134	-0,027	-0,65%
28	4.515	7,550	7,550	0,821	133,647	4,228	1,000	161,260	4,184	-0,044	-1,05%
29	4.677	7,681	7,681	0,807	131,321	4,294	1,000	161,260	4,231	-0,063	-1,46%
30	4.838	7,810	7,810	0,794	129,113	4,359	1,000	161,260	4,277	-0,082	-1,89%
31	4.999	7,937	7,937	0,781	127,013	4,423	1,000	161,260	4,320	-0,103	-2,32%
32	5.160	8,062	8,062	0,769	125,011	4,485	1,000	161,260	4,361	-0,124	-2,77%
33	5.322	8,185	8,185	0,758	123,102	4,547	1,000	161,260	4,401	-0,146	-3,22%
34	5.483	8,307	8,307	0,747	121,278	4,608	1,000	161,260	4,439	-0,169	-3,67%
35	5.644	8,426	8,426	0,736	119,532	4,668	1,000	161,260	4,475	-0,193	-4,13%
36	5.805	8,544	8,544	0,726	117,860	4,727	1,000	161,260	4,510	-0,217	-4,59%
37	5.967	8,660	8,660	0,716	116,256	4,785	1,000	161,260	4,543	-0,242	-5,05%
38	6.128	8,775	8,775	0,707	114,715	4,843	1,000	161,260	4,576	-0,267	-5,51%
39	6.289	8,888	8,888	0,698	113,235	4,899	1,000	161,260	4,607	-0,293	-5,97%
40	6.450	9,000	9,000	0,689	111,810	4,955	1,000	161,260	4,636	-0,319	-6,44%

Folha de cálculo
Electronic Gear = 4, até 200mm

41	6.612	9.110	9.110	9.110	110.438	5.011	1.000	161.260	4.665	-0.346	-6.90%
42	6.773	9.220	9.220	109.115	5.065	5.065	1.000	161.260	4.693	-0.373	-7.35%
43	6.934	9.327	9.327	107.838	5.119	5.119	1.000	161.260	4.719	-0.400	-7.81%
44	7.095	9.434	9.434	106.606	5.173	5.173	1.000	161.260	4.745	-0.427	-8.26%
45	7.257	9.539	9.539	105.414	5.226	5.226	1.000	161.260	4.770	-0.455	-8.71%
46	7.418	9.644	9.644	104.262	5.278	5.278	1.000	161.260	4.794	-0.484	-9.16%
47	7.579	9.747	9.747	103.147	5.329	5.329	1.000	161.260	4.817	-0.512	-9.61%
48	7.740	9.849	9.849	102.066	5.381	5.381	1.000	161.260	4.840	-0.541	-10.05%
49	7.902	9.950	9.950	101.019	5.431	5.431	1.000	161.260	4.862	-0.569	-10.48%
50	8.063	10.050	10.050	100.004	5.481	5.481	1.000	161.260	4.883	-0.598	-10.92%
51	8.224	10.149	10.149	99.018	5.531	5.531	1.000	161.260	4.903	-0.628	-11.35%
52	8.386	10.247	10.247	98.061	5.580	5.580	1.000	161.260	4.923	-0.657	-11.77%
53	8.547	10.344	10.344	97.132	5.629	5.629	1.000	161.260	4.942	-0.686	-12.19%
54	8.708	10.440	10.440	96.228	5.677	5.677	1.000	161.260	4.961	-0.716	-12.61%
55	8.869	10.536	10.536	95.349	5.724	5.724	1.000	161.260	4.979	-0.745	-13.02%
56	9.031	10.630	10.630	94.494	5.772	5.772	1.000	161.260	4.997	-0.775	-13.43%
57	9.192	10.724	10.724	93.661	5.819	5.819	1.000	161.260	5.014	-0.805	-13.83%
58	9.353	10.817	10.817	92.850	5.865	5.865	1.000	161.260	5.030	-0.835	-14.23%
59	9.514	10.909	10.909	92.060	5.911	5.911	1.000	161.260	5.046	-0.865	-14.63%
60	9.676	11.000	11.000	91.289	5.957	5.957	1.000	161.260	5.062	-0.895	-15.02%
61	9.837	11.091	11.091	90.538	6.002	6.002	1.000	161.260	5.077	-0.925	-15.41%
62	9.998	11.180	11.180	89.805	6.047	6.047	1.000	161.260	5.092	-0.955	-15.79%
63	10.159	11.269	11.269	89.089	6.092	6.092	1.000	161.260	5.107	-0.985	-16.17%
64	10.321	11.358	11.358	88.390	6.136	6.136	1.000	161.260	5.121	-1.015	-16.54%
65	10.482	11.446	11.446	87.708	6.180	6.180	1.000	161.260	5.135	-1.045	-16.91%
66	10.643	11.533	11.533	87.041	6.224	6.224	1.000	161.260	5.148	-1.075	-17.28%
67	10.804	11.619	11.619	86.389	6.267	6.267	1.000	161.260	5.161	-1.106	-17.64%
68	10.966	11.705	11.705	85.751	6.310	6.310	1.000	161.260	5.174	-1.136	-18.00%
69	11.127	11.790	11.790	85.127	6.352	6.352	1.000	161.260	5.186	-1.166	-18.35%
70	11.288	11.874	11.874	84.517	6.395	6.395	1.000	161.260	5.199	-1.196	-18.70%
71	11.449	11.958	11.958	83.920	6.437	6.437	1.000	161.260	5.210	-1.226	-19.05%
72	11.611	12.042	12.042	83.335	6.478	6.478	1.000	161.260	5.222	-1.256	-19.39%
73	11.772	12.124	12.124	82.762	6.520	6.520	1.000	161.260	5.233	-1.286	-19.73%
74	11.933	12.207	12.207	82.201	6.561	6.561	1.000	161.260	5.244	-1.317	-20.07%
75	12.094	12.288	12.288	81.651	6.602	6.602	1.000	161.260	5.255	-1.347	-20.40%
76	12.256	12.369	12.369	81.112	6.642	6.642	1.000	161.260	5.266	-1.377	-20.73%
77	12.417	12.450	12.450	80.584	6.683	6.683	1.000	80.630	5.305	-1.377	-20.61%
78	12.578	12.530	12.530	80.065	6.723	6.723	1.000	0.000	5.374	-1.349	-20.06%
79	12.740	12.610	12.610	79.557	6.763	6.763	1.000	0.000	5.443	-1.319	-19.51%
80	12.901	12.689	12.689	79.058	6.802	6.802	1.000	0.000	5.512	-1.290	-18.97%
81	13.062	12.767	12.767	78.569	6.842	6.842	1.000	0.000	5.581	-1.261	-18.42%
82	13.223	12.845	12.845	78.088	6.881	6.881	1.000	0.000	5.650	-1.231	-17.89%
83	13.385	12.923	12.923	77.616	6.920	6.920	1.000	0.000	5.719	-1.201	-17.35%
84	13.546	13.000	13.000	77.153	6.958	6.958	1.000	0.000	5.788	-1.170	-16.82%
85	13.707	13.077	13.077	76.697	6.996	6.996	1.000	0.000	5.857	-1.140	-16.29%
86	13.868	13.153	13.153	76.250	7.035	7.035	1.000	0.000	5.926	-1.109	-15.77%
87	14.030	13.229	13.229	75.811	7.073	7.073	1.000	0.000	5.994	-1.078	-15.24%
88	14.191	13.304	13.304	75.379	7.110	7.110	1.000	0.000	6.063	-1.047	-14.72%
89	14.352	13.379	13.379	74.954	7.148	7.148	1.000	0.000	6.132	-1.016	-14.21%
90	14.513	13.454	13.454	74.536	7.185	7.185	1.000	0.000	6.201	-0.984	-13.69%
91	14.675	13.528	13.528	74.126	7.222	7.222	1.000	0.000	6.270	-0.952	-13.18%
92	14.836	13.601	13.601	73.722	7.259	7.259	1.000	0.000	6.339	-0.920	-12.68%
93	14.997	13.675	13.675	73.324	7.296	7.296	1.000	0.000	6.408	-0.888	-12.17%
94	15.158	13.748	13.748	72.933	7.332	7.332	1.000	0.000	6.477	-0.856	-11.67%
95	15.320	13.820	13.820	72.548	7.369	7.369	1.000	0.000	6.546	-0.823	-11.17%
96	15.481	13.892	13.892	72.170	7.405	7.405	1.000	0.000	6.615	-0.790	-10.67%
97	15.642	13.964	13.964	71.797	7.441	7.441	1.000	0.000	6.683	-0.757	-10.18%
98	15.803	14.036	14.036	71.429	7.476	7.476	1.000	0.000	6.752	-0.724	-9.68%
99	15.965	14.107	14.107	71.068	7.512	7.512	1.000	0.000	6.821	-0.691	-9.19%
100	16.126	14.177	14.177	70.711	7.547	7.547	1.000	0.000	6.890	-0.657	-8.71%
101	16.287	14.248	14.248	70.360	7.583	7.583	1.000	0.000	6.959	-0.623	-8.22%
102	16.449	14.318	14.318	70.015	7.618	7.618	1.000	0.000	7.028	-0.590	-7.74%
103	16.610	14.387	14.387	69.674	7.652	7.652	1.000	0.000	7.097	-0.556	-7.26%
104	16.771	14.457	14.457	69.338	7.687	7.687	1.000	0.000	7.166	-0.521	-6.78%
105	16.932	14.526	14.526	69.007	7.722	7.722	1.000	0.000	7.235	-0.487	-6.31%
106	17.094	14.595	14.595	68.681	7.756	7.756	1.000	0.000	7.304	-0.452	-5.83%
107	17.255	14.663	14.663	68.359	7.790	7.790	1.000	0.000	7.373	-0.418	-5.36%
108	17.416	14.731	14.731	68.042	7.824	7.824	1.000	0.000	7.441	-0.383	-4.89%
109	17.577	14.799	14.799	67.729	7.858	7.858	1.000	0.000	7.510	-0.348	-4.43%
110	17.739	14.866	14.866	67.421	7.892	7.892	1.000	0.000	7.579	-0.313	-3.96%
111	17.900	14.933	14.933	67.116	7.925	7.925	1.000	0.000	7.648	-0.277	-3.50%
112	18.061	15.000	15.000	66.816	7.959	7.959	1.000	0.000	7.717	-0.242	-3.04%
113	18.222	15.067	15.067	66.519	7.992	7.992	1.000	0.000	7.786	-0.206	-2.58%

Folha de cálculo
Electronic Gear = 4, até 200mm

114	18.384	15.133	15.133	66.227	8.025	0,000	0,000	7.855	-0,171	-2,12%
115	18.545	15.199	15.199	65.939	8.058	0,000	0,000	7.924	-0,135	-1,67%
116	18.706	15.264	15.264	65.654	8.091	0,000	0,000	7.993	-0,099	-1,22%
117	18.867	15.330	15.330	65.372	8.124	0,000	0,000	8.062	-0,062	-0,77%
118	19.029	15.395	15.395	65.095	8.156	0,000	0,000	8.130	-0,026	-0,32%
119	19.190	15.460	15.460	64.821	8.189	0,000	0,000	8.199	0,010	0,13%
120	19.351	15.524	15.524	64.550	8.221	0,000	0,000	8.268	0,047	0,57%
121	19.512	15.588	15.588	64.283	8.253	0,000	0,000	8.337	0,084	1,02%
122	19.674	15.652	15.652	64.019	8.285	0,000	0,000	8.406	0,121	1,46%
123	19.835	15.716	15.716	63.758	8.317	0,000	0,000	8.475	0,158	1,90%
124	19.996	15.780	15.780	63.500	8.349	0,000	0,000	8.544	0,195	2,33%
125	20.157	15.843	15.843	63.246	8.381	0,000	0,000	8.613	0,232	2,77%
126	20.319	15.906	15.906	62.994	8.412	0,000	0,000	8.682	0,269	3,20%
127	20.480	15.969	15.969	62.746	8.444	0,000	0,000	8.751	0,307	3,64%
128	20.641	16.031	16.031	62.500	8.475	0,000	0,000	8.819	0,345	4,07%
129	20.803	16.093	16.093	62.258	8.506	0,000	0,000	8.888	0,382	4,49%
130	20.964	16.155	16.155	62.018	8.537	0,000	0,000	8.957	0,420	4,92%
131	21.125	16.217	16.217	61.781	8.568	0,000	0,000	9.026	0,458	5,35%
132	21.286	16.279	16.279	61.546	8.599	0,000	0,000	9.095	0,496	5,77%
133	21.448	16.340	16.340	61.314	8.629	0,000	0,000	9.164	0,535	6,19%
134	21.609	16.401	16.401	61.085	8.660	0,000	0,000	9.233	0,573	6,62%
135	21.770	16.462	16.462	60.858	8.690	0,000	0,000	9.302	0,611	7,03%
136	21.931	16.523	16.523	60.634	8.721	0,000	0,000	9.371	0,650	7,45%
137	22.093	16.583	16.583	60.413	8.751	0,000	0,000	9.440	0,689	7,87%
138	22.254	16.643	16.643	60.193	8.781	0,000	0,000	9.508	0,727	8,28%
139	22.415	16.703	16.703	59.976	8.811	0,000	0,000	9.577	0,766	8,70%
140	22.576	16.763	16.763	59.762	8.841	0,000	0,000	9.646	0,805	9,11%
141	22.738	16.823	16.823	59.549	8.871	0,000	0,000	9.715	0,844	9,52%
142	22.899	16.882	16.882	59.339	8.900	0,000	0,000	9.784	0,884	9,93%
143	23.060	16.941	16.941	59.132	8.930	0,000	0,000	9.853	0,923	10,33%
144	23.221	17.000	17.000	58.926	8.960	0,000	0,000	9.922	0,962	10,74%
145	23.383	17.059	17.059	58.722	8.989	0,000	0,000	9.991	1,002	11,15%
146	23.544	17.117	17.117	58.521	9.018	0,000	0,000	10.060	1,041	11,55%
147	23.705	17.176	17.176	58.321	9.047	0,000	0,000	10.129	1,081	11,95%
148	23.866	17.234	17.234	58.124	9.076	0,000	0,000	10.197	1,121	12,35%
149	24.028	17.292	17.292	57.929	9.105	0,000	0,000	10.266	1,161	12,75%
150	24.189	17.349	17.349	57.735	9.134	0,000	0,000	10.335	1,201	13,15%
151	24.350	17.407	17.407	57.544	9.163	0,000	0,000	10.404	1,241	13,54%
152	24.511	17.464	17.464	57.354	9.192	0,000	0,000	10.473	1,281	13,94%
153	24.673	17.521	17.521	57.166	9.220	0,000	0,000	10.542	1,322	14,33%
154	24.834	17.578	17.578	56.981	9.249	0,000	0,000	10.611	1,362	14,73%
155	24.995	17.635	17.635	56.796	9.277	0,000	0,000	10.680	1,402	15,12%
156	25.157	17.692	17.692	56.614	9.306	0,000	0,000	10.749	1,443	15,51%
157	25.318	17.748	17.748	56.433	9.334	0,000	0,000	10.818	1,484	15,90%
158	25.479	17.804	17.804	56.255	9.362	0,000	0,000	10.887	1,525	16,28%
159	25.640	17.861	17.861	56.077	9.390	0,000	0,000	10.955	1,565	16,67%
160	25.802	17.916	17.916	55.902	9.418	0,000	0,000	11.024	1,606	17,06%
161	25.963	17.972	17.972	55.728	9.446	0,000	0,000	11.093	1,647	17,44%
162	26.124	18.028	18.028	55.556	9.474	0,000	0,000	11.162	1,688	17,82%
163	26.285	18.083	18.083	55.385	9.501	0,000	0,000	11.231	1,730	18,20%
164	26.447	18.138	18.138	55.216	9.529	0,000	0,000	11.300	1,771	18,58%
165	26.608	18.193	18.193	55.048	9.557	0,000	0,000	11.369	1,812	18,96%
166	26.769	18.248	18.248	54.882	9.584	0,000	0,000	11.438	1,854	19,34%
167	26.930	18.303	18.303	54.718	9.611	0,000	0,000	11.507	1,895	19,72%
168	27.092	18.358	18.358	54.555	9.639	0,000	0,000	11.576	1,937	20,09%
169	27.253	18.412	18.412	54.393	9.666	0,000	0,000	11.644	1,979	20,47%
170	27.414	18.466	18.466	54.233	9.693	0,000	0,000	11.713	2,020	20,84%
171	27.575	18.520	18.520	54.074	9.720	0,000	0,000	11.782	2,062	21,22%
172	27.737	18.574	18.574	53.917	9.747	0,000	0,000	11.851	2,104	21,59%
173	27.898	18.628	18.628	53.761	9.774	0,000	0,000	11.920	2,146	21,96%
174	28.059	18.682	18.682	53.606	9.801	0,000	0,000	11.989	2,188	22,33%
175	28.220	18.735	18.735	53.452	9.827	0,000	0,000	12.058	2,230	22,70%
176	28.382	18.788	18.788	53.300	9.854	0,000	0,000	12.127	2,273	23,06%
177	28.543	18.841	18.841	53.150	9.881	0,000	0,000	12.196	2,315	23,43%
178	28.704	18.894	18.894	53.000	9.907	0,000	0,000	12.265	2,357	23,79%
179	28.866	18.947	18.947	52.852	9.934	0,000	0,000	12.333	2,400	24,16%
180	29.027	19.000	19.000	52.705	9.960	0,000	0,000	12.402	2,442	24,52%
181	29.188	19.053	19.053	52.559	9.986	0,000	0,000	12.471	2,485	24,88%
182	29.349	19.105	19.105	52.414	10.013	0,000	0,000	12.540	2,528	25,24%
183	29.511	19.157	19.157	52.271	10.039	0,000	0,000	12.609	2,570	25,60%
184	29.672	19.209	19.209	52.129	10.065	0,000	0,000	12.678	2,613	25,96%
185	29.833	19.261	19.261	51.988	10.091	0,000	0,000	12.747	2,656	26,32%
186	29.994	19.313	19.313	51.848	10.117	0,000	0,000	12.816	2,699	26,68%

Folha de cálculo
Electronic Gear = 4, até 200mm

187	30.156	19.365				19.365				51.709	10.143	0,000	0,000	12,885	2,742	27,04%
188	30.317	19.416	19.416			19.416				51.571	10.168	0,000	0,000	12.954	2.785	27,39%
189	30.478	19.468	19.468			19.468				51.435	10.194	0,000	0,000	13.022	2.828	27,75%
190	30.639	19.519	19.519			19.519				51.299	10.220	0,000	0,000	13.091	2.872	28,10%
191	30.801	19.570	19.570			19.570				51.165	10.245	0,000	0,000	13.160	2.915	28,45%
192	30.962	19.621	19.621			19.621				51.031	10.271	0,000	0,000	13.229	2.958	28,80%
193	31.123	19.672	19.672			19.672				50.899	10.296	0,000	0,000	13.298	3.002	29,15%
194	31.284	19.723	19.723			19.723				50.767	10.322	0,000	0,000	13.367	3.045	29,50%
195	31.446	19.774	19.774			19.774				50.637	10.347	0,000	0,000	13.436	3.089	29,85%
196	31.607	19.824	19.824			19.824				50.508	10.372	0,000	0,000	13.505	3.132	30,20%
197	31.768	19.875	19.875			19.875				50.379	10.397	0,000	0,000	13.574	3.176	30,55%
198	31.929	19.925	19.925			19.925				50.252	10.423	0,000	0,000	13.643	3.220	30,89%
199	32.091	19.975	19.975			19.975				50.126	10.448	0,000	0,000	13.711	3.264	31,24%
200	32.252	20.025	20.025			20.025				50.000	10.473	0,000	0,000	13.780	3.308	31,58%

Anexo IV. Folha de cálculo do controlo dos parâmetros cinemáticos da Mesa / Guia. Electronic Gear = 40.

Folha de cálculo

Electronic Gear = 40, até 200mm

Requisito do Motor (fixo) **4.096** pulsos /rotação * pulsos desmultiplicados / por rotação

Passo da rosca do parafuso sem-fim: 1/2 polegada * igual a **12,700** mm

Cn-24, G1 =	40	User Parameters
Cn-25, G2 =	1	

G1/G2 =	40 Electronic Gear
Pulsos Electivos	102,40 por rotação
Rotinas no Labview	204,80 por rotação

Acceleção média = **1,000** mm/s²

Velocidade inicial = **1,000** mm/s

Avanço (mm/rotina do Labview): **0,062** mm/rotina

Rotinas no Labview para avançar 1mm: **16,126** rotinas

Duração da Rotina (milisegundos): **62,012** milisegundos

1000 controle

Distância a percorrer	Nº de Rotinas	Velocidade instantânea	Velocidade em cada ponto (função do nº de rotinas)	Duração da Rotina calculada	Tempo acumulado entre pontos	Velocidade média ("calculada")	Duração da Rotina arredondada	Tempo acumulado entre pontos	Velocidade média ("arredondada")	Diferença valor absoluto	Diferença percentagem
mm		mm/s	mm/s	milisegundos	milisegundos	mm/s	milisegundos	milisegundos	mm/s	valor absoluto	vel. "arredondada" - vel. "calculada"
0	0	1,000	1,000	62,012	0,000		62,000	0,000		0,000	%
1	16	1,732	1,732	35,802	788,675	1,268	36,000	790,173	1,266	-0,002	-0,19%
2	32	2,236	2,236	27,732	512,282	1,537	28,000	516,031	1,531	-0,006	-0,40%
3	48	2,646	2,646	23,438	412,589	1,751	23,000	411,213	1,747	-0,004	-0,23%
4	65	3,000	3,000	20,671	355,649	1,933	21,000	354,772	1,930	-0,003	-0,14%
5	81	3,317	3,317	18,697	317,422	2,095	19,000	322,520	2,088	-0,007	-0,34%
6	97	3,606	3,606	17,199	289,431	2,242	17,000	290,268	2,235	-0,007	-0,33%
7	113	3,873	3,873	16,011	267,774	2,378	16,000	266,079	2,372	-0,006	-0,25%
8	129	4,123	4,123	15,040	250,367	2,505	15,000	249,953	2,499	-0,005	-0,21%
9	145	4,359	4,359	14,226	235,976	2,624	14,000	233,827	2,620	-0,004	-0,14%
10	161	4,583	4,583	13,532	223,817	2,737	14,000	225,764	2,732	-0,005	-0,18%
11	177	4,796	4,796	12,930	213,366	2,844	13,000	217,701	2,836	-0,008	-0,28%
12	194	5,000	5,000	12,402	204,257	2,947	12,000	201,575	2,941	-0,006	-0,20%
13	210	5,196	5,196	11,934	196,225	3,046	12,000	193,512	3,042	-0,004	-0,13%
14	226	5,385	5,385	11,515	189,073	3,141	12,000	193,512	3,134	-0,007	-0,22%
15	242	5,568	5,568	11,138	182,650	3,233	11,000	185,449	3,224	-0,009	-0,27%
16	258	5,745	5,745	10,795	176,841	3,322	11,000	177,386	3,313	-0,009	-0,28%
17	274	5,916	5,916	10,482	171,554	3,408	10,000	169,323	3,401	-0,008	-0,22%
18	290	6,083	6,083	10,195	166,715	3,492	10,000	161,260	3,488	-0,004	-0,11%
19	306	6,245	6,245	9,930	162,264	3,573	10,000	161,260	3,570	-0,003	-0,09%
20	323	6,403	6,403	9,685	158,151	3,653	10,000	161,260	3,648	-0,005	-0,14%
21	339	6,557	6,557	9,457	154,336	3,730	9,000	153,197	3,726	-0,004	-0,12%
22	355	6,708	6,708	9,244	150,785	3,806	9,000	145,134	3,805	-0,001	-0,02%
23	371	6,856	6,856	9,045	147,468	3,880	9,000	145,134	3,881	0,001	0,02%
24	387	7,000	7,000	8,859	144,361	3,953	9,000	145,134	3,953	0,000	0,01%
25	403	7,141	7,141	8,683	141,443	4,024	9,000	145,134	4,022	-0,002	-0,05%
26	419	7,280	7,280	8,518	138,694	4,093	9,000	145,134	4,087	-0,006	-0,15%
27	435	7,416	7,416	8,362	136,100	4,161	8,000	137,071	4,155	-0,007	-0,16%
28	452	7,550	7,550	8,214	133,647	4,228	8,000	129,008	4,225	-0,004	-0,09%
29	468	7,681	7,681	8,073	131,321	4,294	8,000	129,008	4,292	-0,002	-0,05%

Folha de cálculo

Electronic Gear = 40, até 200mm

160	2.580	17.916	17.916			17.916		3.461	55.902	9.418	3.000	48.378	9.356	-0.062	-0.66%
161	2.596	17.972	17.972			17.972		3.450	55.728	9.446	3.000	48.378	9.388	-0.058	-0.62%
162	2.612	18.028	18.028			18.028		3.440	55.556	9.474	3.000	48.378	9.420	-0.054	-0.57%
163	2.629	18.083	18.083			18.083		3.429	55.385	9.501	3.000	48.378	9.451	-0.050	-0.53%
164	2.645	18.138	18.138			18.138		3.419	55.216	9.529	3.000	48.378	9.482	-0.047	-0.49%
165	2.661	18.193	18.193			18.193		3.408	55.048	9.557	3.000	48.378	9.514	-0.043	-0.45%
166	2.677	18.248	18.248			18.248		3.398	54.882	9.584	3.000	48.378	9.545	-0.039	-0.41%
167	2.693	18.303	18.303			18.303		3.388	54.718	9.611	3.000	48.378	9.576	-0.036	-0.37%
168	2.709	18.358	18.358			18.358		3.378	54.555	9.639	3.000	48.378	9.606	-0.032	-0.34%
169	2.725	18.412	18.412			18.412		3.368	54.393	9.666	3.000	48.378	9.637	-0.029	-0.30%
170	2.741	18.466	18.466			18.466		3.358	54.233	9.693	3.000	48.378	9.667	-0.026	-0.27%
171	2.758	18.520	18.520			18.520		3.348	54.074	9.720	3.000	48.378	9.697	-0.023	-0.23%
172	2.774	18.574	18.574			18.574		3.339	53.917	9.747	3.000	48.378	9.727	-0.020	-0.20%
173	2.790	18.628	18.628			18.628		3.329	53.761	9.774	3.000	48.378	9.757	-0.017	-0.17%
174	2.806	18.682	18.682			18.682		3.319	53.606	9.801	3.000	48.378	9.787	-0.014	-0.14%
175	2.822	18.735	18.735			18.735		3.310	53.452	9.827	3.000	48.378	9.816	-0.011	-0.11%
176	2.838	18.788	18.788			18.788		3.301	53.300	9.854	3.000	48.378	9.846	-0.008	-0.08%
177	2.854	18.841	18.841			18.841		3.291	53.150	9.881	3.000	48.378	9.875	-0.006	-0.06%
178	2.870	18.894	18.894			18.894		3.282	53.000	9.907	3.000	48.378	9.904	-0.003	-0.03%
179	2.887	18.947	18.947			18.947		3.273	52.852	9.934	3.000	48.378	9.933	-0.001	-0.01%
180	2.903	19.000	19.000			19.000		3.264	52.705	9.960	3.000	48.378	9.962	0.002	0.02%
181	2.919	19.053	19.053			19.053		3.255	52.559	9.986	3.000	48.378	9.990	0.004	0.04%
182	2.935	19.105	19.105			19.105		3.246	52.414	10.013	3.000	48.378	10.019	0.006	0.06%
183	2.951	19.157	19.157			19.157		3.237	52.271	10.039	3.000	48.378	10.047	0.008	0.08%
184	2.967	19.209	19.209			19.209		3.228	52.129	10.065	3.000	48.378	10.075	0.010	0.10%
185	2.983	19.261	19.261			19.261		3.219	51.988	10.091	3.000	48.378	10.103	0.012	0.12%
186	2.999	19.313	19.313			19.313		3.211	51.848	10.117	3.000	48.378	10.131	0.014	0.14%
187	3.016	19.365	19.365			19.365		3.202	51.709	10.143	3.000	48.378	10.159	0.016	0.16%
188	3.032	19.416	19.416			19.416		3.194	51.571	10.168	3.000	48.378	10.186	0.018	0.18%
189	3.048	19.468	19.468			19.468		3.185	51.435	10.194	3.000	48.378	10.214	0.020	0.19%
190	3.064	19.519	19.519			19.519		3.177	51.299	10.220	3.000	48.378	10.241	0.021	0.21%
191	3.080	19.570	19.570			19.570		3.169	51.165	10.245	3.000	48.378	10.268	0.023	0.22%
192	3.096	19.621	19.621			19.621		3.160	51.031	10.271	3.000	48.378	10.295	0.024	0.24%
193	3.112	19.672	19.672			19.672		3.152	50.899	10.296	3.000	48.378	10.322	0.026	0.25%
194	3.128	19.723	19.723			19.723		3.144	50.767	10.322	3.000	48.378	10.349	0.027	0.26%
195	3.145	19.774	19.774			19.774		3.136	50.637	10.347	3.000	48.378	10.375	0.028	0.27%
196	3.161	19.824	19.824			19.824		3.128	50.508	10.372	3.000	48.378	10.402	0.029	0.28%
197	3.177	19.875	19.875			19.875		3.120	50.379	10.397	3.000	48.378	10.428	0.030	0.29%
198	3.193	19.925	19.925			19.925		3.112	50.252	10.423	3.000	48.378	10.454	0.031	0.30%
199	3.209	19.975	19.975			19.975		3.104	50.126	10.448	3.000	48.378	10.480	0.032	0.31%
200	3.225	20.025	20.025			20.025		3.097	50.000	10.473	3.000	48.378	10.506	0.033	0.32%