RRR-robot : instruction manual

van Beek, A.M.

Published: 01/01/1998

Document Version
Publisher's PDF, also known as Version of Record (includes final page, issue and volume numbers)

Please check the document version of this publication:

• A submitted manuscript is the author's version of the article upon submission and before peer-review. There can be important differences between the submitted version and the official published version of record. People interested in the research are advised to contact the author for the final version of the publication, or visit the DOI to the publisher's website.
• The final author version and the galley proof are versions of the publication after peer review.
• The final published version features the final layout of the paper including the volume, issue and page numbers.

Link to publication

Citation for published version (APA):

General rights
Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

• Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
• You may not further distribute the material or use it for any profit-making activity or commercial gain
• You may freely distribute the URL identifying the publication in the public portal

Take down policy
If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

Download date: 29. Dec. 2018
Trainee project

Supervisor: ir. L. Kodde

Faculty of Mechanical Engineering
Eindhoven University of Technology (TUE)
March 1998
Operational warnings

1. Do not operate the robot unsupervised, or without the emergency stop within reach. The motors can rotate at high speed with high torque; beware of the rotating radius of the robot.

2. Do not touch any part of the robot when the power is switched on. Dangerously high voltage is present at the location of brush terminals (shielded with plexiglas covers), and inside the octagonal shaped, silver colored housing.

3. Keep clear of the rotating radius of the robot while the servos are powered and enabled.

4. Do not exceed the following velocities in order to stay below the allowed moment for alternating load, DM60: 0.5 [rps], DM30: 1 [rps], and DM15: 1.5 [rps].

5. Do not operate the DM15 servo while the DM60 and/or DM30 are at a continuous standstill in order to prevent wear or damage to the power sliprings due to welding effects.

6. Do not operate the DM30 servo while the DM60 is at a continuous standstill in order to prevent wear or damage to the power sliprings due to welding effects.

7. Ensure that the motor phases $V_A$, $V_B$, and $V_C$ are connected correctly. If not, the encoder feedback-loop results in unstable behavior, i.e., the motor will start rotating at maximum velocity and with maximum torque in a direction opposite to the desired trajectory.

8. After modifying the Simulink controller, always perform a “test run” to see if error messages occur.

9. When performing an experiment, do not start the controller before the servos are powered and enabled; the introduced tracking error may cause a large and unwanted response.

10. Do not use the enable circuit as an alternative emergency stop. There is a considerable delay (approximately one second) before opening the circuit has effect.

11. Use the emergency stop if the control software or the operating system crashes; or if error messages occur after the Simulink controller has been started.

12. Use the emergency stop and perform a (hard) reboot of the PC containing the MultiQ I/O board, if the encoder read-out becomes discontinuous or otherwise erratic.

13. Do not apply more than 5.3 [V] to the (analog) inputs of the MultiQ I/O board to avoid overload. As a consequence, do not use any of the analog driver outputs without additional overload protection since a surge of 8 [V] occurs when the power is switched on.
14. Ensure that the power is switched off before opening the Driver cabinet. Dangerously high voltage is present inside this unit.

15. Use the base connection to move or lift the robot (approximate weight 90 [kg]). Do not use the octagonal shaped, silver colored housing to lift the robot; unless with extreme care to prevent damage to the spherical joint at the base of the robot.
# Contents

Operational warnings ........................................ iii
Terminology .................................................. 1

1 Introduction ................................................ 3
   1.1 Objective .................................................. 3
   1.2 Getting started .......................................... 4

2 General description ......................................... 7

3 Manipulator frame ........................................... 11
   3.1 Composition ............................................. 11
   3.2 Assembly and installation .................................. 12
   3.3 Wiring ................................................... 18
      3.3.1 Power connections .................................. 18
      3.3.2 Signal connections .................................. 20
   3.4 Operation ................................................ 23
   3.5 Inertia parameters ...................................... 23

4 Joint actuation System ....................................... 25
   4.1 Dynaserv motors ......................................... 25
   4.2 Driver cabinet .......................................... 26
   4.3 Dynaserv drivers ........................................ 27
   4.4 Dynaserv brakes ........................................ 27

5 Control System .............................................. 29
   5.1 MultiQ plug-in and terminal board ................. 29
   5.2 External controls ..................................... 30
   5.3 Personal Computer .................................... 31
      5.3.1 PC Hardware ....................................... 31
      5.3.2 PC software ....................................... 31

6 Maintenance and inspection ................................ 33

A Addresses .................................................. 35
CONTENTS

B Components 37
B.1 Wincon software manual ........................................ 38
B.2 MultiQ board manual ............................................. 70
B.3 Manual: Dynaserv DD Servo-Actuator DM/SD series ............ 91
B.4 Manual: BE-A/B Type Dynamic Brake ......................... 113
B.5 Cabinet manual ................................................... 123
B.6 Power sliprings ................................................... 133
B.7 Signal sliprings ................................................... 135
B.8 Conceptual design ................................................ 137
Terminology

When applicable, parts are numbered from the base to the end-effector, i.e., the stationary base is link 0 (black parts), the rotating octagonal silver housing is the top of link 1, link 2 is green, and the end-effector is link 3.

<table>
<thead>
<tr>
<th>Short</th>
<th>Synonym</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>A/D</td>
<td>AD(C)</td>
<td>Analog-to-Digital (Converter)</td>
</tr>
<tr>
<td>AC</td>
<td></td>
<td>Alternating Current</td>
</tr>
<tr>
<td>BDC</td>
<td>Brushless DC</td>
<td>Brushless Direct Current (motor)</td>
</tr>
<tr>
<td>BE60</td>
<td>Brake 1</td>
<td>Electro-mechanical brake BE1060B</td>
</tr>
<tr>
<td>BE30</td>
<td>Brake 2</td>
<td>Electro-mechanical brake BE1030B</td>
</tr>
<tr>
<td>CTD</td>
<td>GTD</td>
<td>“Centrale/Gemeenschappelijke Technische Dienst”</td>
</tr>
<tr>
<td>D/A</td>
<td>DA(C)</td>
<td>Digital-to-Analog (Converter)</td>
</tr>
<tr>
<td>DC</td>
<td></td>
<td>Direct Current</td>
</tr>
<tr>
<td>DM60</td>
<td>Motor/servo 1</td>
<td>Dynaserv Motor DM1060B50*1, max. torque 60 [Nm]</td>
</tr>
<tr>
<td>DM30</td>
<td>Motor/servo 2</td>
<td>Dynaserv Motor DM1030B50*1, max. torque 30 [Nm]</td>
</tr>
<tr>
<td>DM15</td>
<td>Motor/servo 3</td>
<td>Dynaserv Motor DM1015B50*1, max. torque 15 [Nm]</td>
</tr>
<tr>
<td>Driver cabinet</td>
<td></td>
<td>Shielded cabinet for drivers, brakes and line filters</td>
</tr>
<tr>
<td>MultiQ</td>
<td>PC plug-in board</td>
<td>board with ADC, DAC, encoder inputs, and digital I/O</td>
</tr>
<tr>
<td>PSR-1</td>
<td>Power Slipring 1</td>
<td>SM140-6 and -2 combined</td>
</tr>
<tr>
<td>PSR-2</td>
<td>Power Slipring 2</td>
<td>M-SM204-12</td>
</tr>
<tr>
<td>RTW</td>
<td>Real Time Workshop</td>
<td>Matlab toolbox to generate C-code</td>
</tr>
<tr>
<td>SSR-1</td>
<td>Signal Slipring 1</td>
<td>Capsule AC-267 modified</td>
</tr>
<tr>
<td>SSR-2</td>
<td>Signal Slipring 2</td>
<td>Capsule AC-267 modified</td>
</tr>
<tr>
<td>SSR-3</td>
<td>Signal Slipring 3</td>
<td>Litton 12-ring capsule</td>
</tr>
<tr>
<td>SD60</td>
<td>Driver 1</td>
<td>Driver SD1060B52-2, driving DM60</td>
</tr>
<tr>
<td>SD30</td>
<td>Driver 2</td>
<td>Driver SD1030B52-2, driving DM30</td>
</tr>
<tr>
<td>SD15</td>
<td>Driver 3</td>
<td>Driver SD1015B52-2, driving DM15</td>
</tr>
<tr>
<td>Terminal board</td>
<td></td>
<td>Board with connectors to the internal MultiQ board</td>
</tr>
<tr>
<td>Watcom</td>
<td>Watcom C/C++</td>
<td>Watcom C/C++ compiler version 10.6</td>
</tr>
<tr>
<td>WinCon</td>
<td>WinCon V2.0</td>
<td>Software to interact with real-time running C-code</td>
</tr>
</tbody>
</table>
Chapter 1

Introduction

1.1 Objective

The RRR-robot is designed as a manipulator-like system (with three rotational degrees of freedom), to test a variety of advanced nonlinear control strategies. Since for high-speed tracking of complex trajectories, Coriolis and centrifugal torques form an essential part of the occurring nonlinear effects, the main requirement of this robot is to highlight these velocity dependent torques. This has led to two important features of the RRR-robot:

- the use of sliprings to facilitate unconstrained rotation of each link, and
- the use of direct-drive servos.

In addition, the system has a Simulink based control interface for fast implementation of various control strategies.

Figure 1.1: RRR-robot pictures, notice the brush block terminals to connect the two power sliprings.

The aim of this manual is to describe the composition and use of the RRR-robot as a whole. With regard to the individual components, only key parameters, and non-documented features
(from personal communications and experiences) are listed. For more detailed information, the reader is referred to the included component manuals in Appendix B. Before actually operating the robot, the user should also read the Wincon software manual in Appendix B.1.

1.2 Getting started

The procedure listed below can be used to get familiar with the basic operation of the RRR-robot and its control system. However, it is recommended to read the remaining of this manual and Appendix B.1 first in order to understand the entire background.

1. Switch on the PC.
2. Start Windows. Matlab should start automatically (in the directory c:\rrr\demo).
3. Inside the Matlab Command Window type: demo. The Simulink window with the controller should open.
4. Double click on the button init inside the Simulink window. The matrices $K_p$ and $K_d$ should be read into the Matlab workspace.
5. Select Generate and Built Realtime Code from the Code menu of the Simulink window. A DOS window should open, displaying the compilation procedure. This should be completed without errors (warnings may be disregarded). After successful compilation, close the DOS window.

Do not yet enable the servos.

6. Select Start from the Simulation menu of the Simulink window. Now Wincon should starts automatically and the controller should run “dry” without error messages.

After modifying the Simulink scheme, always perform a “test run” with the controller before enabling the servos. If error messages occur, recompile the controller.

7. Stop the controller by pressing the traffic light inside the Wincon window.
8. Use the File menu of the Wincon window to load demo. Now a number of plot windows should appear (if not, push the Scopes/Workspace button, and select the desired plot variables).
9. Make sure the controller is not running (the traffic light is red, the time display is frozen).
10. Switch on the central power of the Cabinet driver (if necessary pull out the emergency button on the cabinet).
11. Enable the servos by switching on the separate power supply.
12. Put one hand on the emergency stop.

Use the emergency stop if error messages occur after the Simulink controller has been started; if the control software or the operation system crashes; or if the encoder read-out becomes discontinuous or otherwise erratic.
13. Start the controller (push the traffic light or select start form the Simulink window). After 1 second delay, each joint should start moving at a constant velocity.
Chapter 2

General description

The RRR-robot is actuated by three brushless direct-drive servos with maximum torques of 60, 30 and 15 [Nm]. Each motor has its own driver to generate the driving stator phases (both are shown in Fig. 2.1 on the right). The so-called Dynaserv servo is of an outer-rotor type with internal encoder, and internal bearings, providing direct coupling between the outside housing and the attached link. Basically, the robot is constructed by simply combining these servos, i.e., joints and the links in a chain structure.

*Figure 2.1:* On the left, manipulator frame (servos, sliprings, and other frame parts); and on the right, the connections between one Dynaserv motor-driver pair.

To facilitate unconstrained rotation of all joints – the connection to and from the DM30 and DM15, shown in Fig. 2.1 by cable 2 and 5 – are implemented using (power and signal) sliprings. The DM60 is connected directly from the base.

Servos, sliprings, and other frame parts (e.g., links) together constitute the Manipulator frame (shown in Fig. 2.1 on the left). The assembly of the manipulator frame, and all required wiring is described in Chapter 3.
To bring the DM60 and DM30 (with considerable inertia) to a controlled stop, electro-mechanical brakes are placed between motor and driver to short the motor coil. Line filters are used to filter out external disturbances on the power supply (cable 1). Drivers, brakes and line filters are mounted in a shielded Driver cabinet. This cabinet has a central power supply which can be switched off by an external emergency stop. Servos, drivers, brakes, and line filters constitute the joint actuation system described in Chapter 4.

Figure 2.2: Schematic description and reference of the RRR-robot components.
In Chapter 5, the *Control system* is addressed. A PC plug-in control board, the MultiQ (Fig. 2.3), is used to communicate with the motor drivers, e.g., read-in the output of the encoders, and apply the torque command voltage. The terminal board of the MultiQ is placed inside a casing (the "controller" in Fig. 2.1), and connected by a 50-pole connection (cable 3) to each driver.

*Figure 2.3:* MultiQ plug-in board, placed inside the PC (bottom), and terminal board (top right)

The actual control algorithm is implemented in Simulink. If desired, the algorithm can be tested and analyzed by means of simulation. If the results are satisfactory, the simulink blocks can be automatically converted to C-code by the Real Time Workshop (RTW), and subsequently compiled by the Watcom C-compiler. The compiled C-code can be executed directly (using DOS), or in interaction with the Wincon software (using Windows). Using Wincon selected Simulink variables can be displayed and modified in real-time.

In Fig. 2.2 all main components of the RRR-robot are depicted schematically.
Chapter 3

Manipulator frame

3.1 Composition

The manipulator frame consists of servos, sliprings and non-standard frame parts (e.g. links) manufactured by the CTD. Furthermore, it includes all internal wiring. In Table 3.1 the main components are listed, together with their origin and mass. In Fig. 3.1, the conceptual design, all components are assembled. Based on this design the CTD constructed the non-standard components. Therefore, the design sketch can only serve as an indication for the composition of the final RRR-robot (several dimensions may be slightly altered).

Table 3.1: RRR-robot main components (see Fig. 3.1). With the exception of part 8, all parts manufactured by the CTD are made of Aluminum. The masses are either measured, estimated, calculated (indicated with a *), or supplied by the manufacturer.

<table>
<thead>
<tr>
<th>No.</th>
<th>Part</th>
<th>Source</th>
<th>Appendix</th>
<th>Mass [kg]</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>Base</td>
<td>CTD</td>
<td>B.8</td>
<td>15*</td>
</tr>
<tr>
<td>2.</td>
<td>Spherical joint</td>
<td>CTD</td>
<td>B.8</td>
<td>2*</td>
</tr>
<tr>
<td>3.</td>
<td>Housing DM60</td>
<td>CTD</td>
<td>B.8</td>
<td>6*</td>
</tr>
<tr>
<td>4.</td>
<td>DM60</td>
<td>Litton</td>
<td>B.3</td>
<td>12</td>
</tr>
<tr>
<td>5.</td>
<td>Signal slipring 1 (SSR-1)</td>
<td>Litton</td>
<td>B.7</td>
<td>&lt; 0.1</td>
</tr>
<tr>
<td>6.</td>
<td>Rotor extension DM60</td>
<td>CTD</td>
<td>B.8</td>
<td>3.504</td>
</tr>
<tr>
<td>7.</td>
<td>Power slipring 1 (PSR-1)</td>
<td>Litton</td>
<td>B.6</td>
<td></td>
</tr>
<tr>
<td>8.</td>
<td>Bearing 165-120-22</td>
<td>CTD</td>
<td>B.8</td>
<td>&lt; 0.5</td>
</tr>
<tr>
<td>9.</td>
<td>Bearing cover</td>
<td>CTD</td>
<td>B.8</td>
<td>&lt; 0.1</td>
</tr>
<tr>
<td>10.</td>
<td>Housing DM30</td>
<td>CTD</td>
<td>B.8</td>
<td>15.5</td>
</tr>
<tr>
<td>11.</td>
<td>Stator extension DM30</td>
<td>CTD</td>
<td>B.8</td>
<td>1.620</td>
</tr>
<tr>
<td>12.</td>
<td>DM30</td>
<td>Litton</td>
<td>B.3</td>
<td>7.5</td>
</tr>
<tr>
<td>13.</td>
<td>Signal slipring 2 (SSR-2)</td>
<td>Litton</td>
<td>B.7</td>
<td>&lt; 0.1</td>
</tr>
<tr>
<td>14.</td>
<td>Rotor extension DM30</td>
<td>CTD</td>
<td>B.8</td>
<td>1.482</td>
</tr>
<tr>
<td>15.</td>
<td>Power slipring 2 (PSR-2)</td>
<td>Litton</td>
<td>B.6</td>
<td>3.464</td>
</tr>
<tr>
<td>16.</td>
<td>Arm 2</td>
<td>CTD</td>
<td>B.8</td>
<td>3.908</td>
</tr>
<tr>
<td>17.</td>
<td>DM15</td>
<td>Litton</td>
<td>B.3</td>
<td>5.5</td>
</tr>
<tr>
<td>18.</td>
<td>Signal slipring 3 (SSR-3)</td>
<td>Litton</td>
<td>B.7</td>
<td>&lt; 0.1</td>
</tr>
</tbody>
</table>
Figure 3.1: RRR-robot conceptual design, used as a basis for the realization of the various non-standard components. Note that some dimensions were slightly altered during the manufacturing.

3.2 Assembly and installation

In the following, the assembly steps of the frame components are listed. Here, the emphasis is on the mechanical connections. In Section 3.3, the wiring of the sliprings will be treated in more detail.

1. The signal sliprings (5., 13., and 15.) are clamped and glued inside the three motors. The wires to and from the sliprings are bundled and shielded.
2. The Spherical joint (2.) and Housing DM60 (3.) are connected to the Base (1.) with two series of bolts at the bottom of the Base (1.).

3. The DM60 (4.) is bolted to the Spherical joint (2.), through the Base.

![Figure 3.2: Step 2 and 3: connections made from the bottom, three sets of bolts are tightened while the Base stands on its side.](image)

4. PSR-1 (7.) is placed around Rotor extension DM60 (6.) and secured with radial screws. Both are mounted on the rotor of the DM60 (4.). The 8 slipring cables are extended (see Table 3.3, third column). Now the brush block (part of 7.) can be placed.

![Figure 3.3: Step 4: PSR-1 (including brush block) and Rotor extension DM60. The three cable bundles inside the Rotor extension are, from left to right, encoder extension cables to DM15 and DM30; and the 8 extended PSR-1 leads.](image)

5. The Bearing (8.) and the Bearing cover (9.) are mounted.
6. Housing DM30 (10.) is placed on Rotor extension DM60 motor (6.). The cables for data and power originating from PSR-1 (7.) and SSR-1 (5.) enter Housing DM30 through 4 slots (which can be seen at the backside of Housing DM30 when the cover is removed). For details and the connection to PSR-2 (15.) see Table 3.3 and Table 3.5.

7. The bolts (screwed into plugs) connecting Housing DM30 (10.) to the Rotor extension DM60 (6.) are fastened from the inside of the housing. This is done before Stator extension DM30 (11.) is in place.

8. Stator extension DM30 (11.) is bolted to Housing DM30 (10.).

---

**Figure 3.4:** Step 5: additional bearing.

**Figure 3.5:** Step 6 and 7: mounting Housing DM30 (10.)

**Figure 3.6:** Step 8: mounting Stator extension DM30 (11.)
9. DM30 (12.) is mounted on Stator extension DM30 (11.) through the back of Housing DM30 (10.). The DM30 motor cable, the DM30 encoder cable and the PSR-2 bundles can be connected with the appropriate cables (entering through the 4 slots). For details see Table 3.3 and Table 3.5.

![Figure 3.7: Step 9: mounting DM30 (12.).](image)

10. PSR-2 (15.) is mounted on Rotor extension DM30 (14.). Both are placed over DM30 (12.) and bolted to arm2.

![Figure 3.8: Step 10: PSR-2 (15.) and Rotor extension DM30 (14.) are placed over DM30 (12.)](image)

11. Arm 2 (16.) is bolted to the DM30 motor, and Rotor extension DM30 (14.) is fixed to Arm 2. Now the brush block (part of PSR-2) can be placed, and subsequently connected to 8 of the 4 leads originating from PSR-1 (7.).
12. DM15 (17.) is mounted on Arm 2. (16.) The 12 leads from PSR-2 (15.) are grouped in sets of three and connected to the DM15 motor cable. The bundle from SSR-2 (13.) is split up and connected to the DM15 encoder cable and SSR-3 (18.).
Remark

- It is possible to remove DM30 and everything connected to it as a whole:
  - remove the PSR-2 brushes,
  - disconnect the power to DM30 and all signal wires in the DM30 housing, and
  - carefully support the parts to be disconnected, and unscrew the bolts (inside housing DM30) which connect the stator extension to the DM30.

Figure 3.11: Partial disassembly
3.3  Wiring

3.3.1  Power connections

The DM60 motor is directly connected to the SD60 driver with its (extended) motor cable (1=red, 2=white, 3=black, and 4=green), see Fig. 3.12.

To connect the DM30 and the DM15, power sliprings are used: PSR-1 (7.) and PSR-2 (15.). For the location of these sliprings see Fig. 3.1 and assembly steps 4 and 10. The main specifications of these rings are summarized in Table 3.2.

<table>
<thead>
<tr>
<th>property</th>
<th>PSR-1</th>
<th>PSR-2</th>
</tr>
</thead>
<tbody>
<tr>
<td>number of circuits</td>
<td>8</td>
<td>12</td>
</tr>
<tr>
<td>maximum current/circuit</td>
<td>15 [A]</td>
<td>6  [A]</td>
</tr>
<tr>
<td>maximum total load</td>
<td>15 [A] x 50 [V]/circuit</td>
<td>-</td>
</tr>
<tr>
<td>maximum velocity</td>
<td>250 [rpm]</td>
<td>150 [rpm]</td>
</tr>
<tr>
<td>circuit resistance</td>
<td>&lt; 0.1 [Ohm]</td>
<td>-</td>
</tr>
</tbody>
</table>

Table 3.2: Main specifications of power sliprings PSR-1 and PSR-2.

PSR-1 is connected from the base, through a brush-block terminal with 8 contact points, schematically depicted by the 8 “x”’s in Fig. 3.12 (right).

Figure 3.12: Connections to the DM60 (left) and PSR-1 with brush-block terminal (right). Each “x” denotes a clamp bolt (see also Fig. 1.1).

In Table 3.3, the same symbols are used to describe the motor cable connections. For example the DM15 motor cable from the SD15 driver with 4 leads is connected to the bottom 4 clamps of the PSR-1 terminal. Inside PSR-1, 4 leads are extended and (through slot 4 of Housing DM30, depicted by [o o o x]) led to the PSR-2 terminal shown in Fig. 3.13.

⚠️ Do not push any of the power leads in Housing DM30 (green, black, yellow and red) further back into slot 3 and 4. Then they might come into contact with power slipring 2.
The PSR-2 terminal has 24 brushes (12 circuits, 2 brushes per circuit) denoted by [::: ::: :::]. Groups of 6 brushes are interconnected to form 4 channels each consisting of 3 circuits. The same is done on the side of DM15 to power this last servo.

Table 3.3: Connections to power slipring 1 and 2 (PSR-1 and PSR-2). The “x” in [o o x o] denotes a cable entering motor housing 2 via slot 3. The “x” in [::: ::: xxx] denotes a connection to the fourth set of 6 brushes on the horizontal terminal block of slipring 2.

<table>
<thead>
<tr>
<th>driver</th>
<th>terminal</th>
<th>inside</th>
<th>housing</th>
<th>side PSR-2/DM30</th>
</tr>
</thead>
<tbody>
<tr>
<td>SD30</td>
<td>PSR-1</td>
<td>PSR-1</td>
<td>DM30</td>
<td></td>
</tr>
<tr>
<td>VA</td>
<td>red</td>
<td>x</td>
<td>red</td>
<td>via slot 3</td>
</tr>
<tr>
<td>VB</td>
<td>white</td>
<td>x</td>
<td>yellow</td>
<td>[o o x o]</td>
</tr>
<tr>
<td>VC</td>
<td>black</td>
<td>x</td>
<td>black</td>
<td>to DM30</td>
</tr>
<tr>
<td>GND</td>
<td>green</td>
<td>x</td>
<td>green</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>driver</th>
<th>terminal</th>
<th>side PSR-2/DM30</th>
</tr>
</thead>
<tbody>
<tr>
<td>SD15</td>
<td>PSR-2</td>
<td></td>
</tr>
<tr>
<td>GND</td>
<td>green</td>
<td>[::: ::: xxx]</td>
</tr>
<tr>
<td>VC</td>
<td>black</td>
<td>[::: ::: xxx]</td>
</tr>
<tr>
<td>VB</td>
<td>white</td>
<td>[::: xxx :::]</td>
</tr>
<tr>
<td>VA</td>
<td>red</td>
<td>[::: ::: :::]</td>
</tr>
</tbody>
</table>
### 3.3.2 Signal connections

The encoder cable of the DM60 is directly connected with the SD60 driver. To transfer the encoder signals of the DM30 and the DM15 and to transfer additional measurement signals, three signal sliprings are used: SSR-1 (5.), SSR-2 (13.), and SSR-3 (18.). The dimensions and main specifications of these rings are listed in Fig. 3.14 and Table 3.4.

![Figure 3.14: Signal sliprings: on the left, a photograph of the original AC-267 slipring; in the middle, the modified AC-267 (used for SSR-1 and SSR-2); and on the right, SSR-3.](image)

The stator of each slipring is clamped and glued on the inside of the servos (see position 5., 13., and 18. in Fig. 3.1), i.e., attached to the DM rotors. Since the slipring friction torques are small, the slipring leads are used to fix the slipring rotors to the DM stators. For the SSR-1 and SSR-2, tie-raps are used to tie 2 bundles of 18 wires to the motor cable and the encoder cable. Note that, every wire is attached separately to the slipring (see Fig. 3.14), i.e. there is no mechanical connection between the (white) bundle of 18 wires and the slipring itself. Below the base, a cable relieve clamp is placed to prevent any damage to SSR-1.

⚠️ Do not pull any of the bundled slipring cables (white) originating from SSR-1 (below the base), or SSR-2 (in the octagonal shaped, silver colored Housing DM30 (10.)).

| Table 3.4: Main specifications of the signal sliprings (see Fig. 3.14). |
|-----------------|---------|-----------------|
| property        | SSR-1 and 2 | SSR-3 |
| number of circuits | 36       | 12   |
| maximum current/circuit | 1.2 [A]   | 1 [A] |
| maximum total load | 25 [A]    | 50 [W] |
| maximum velocity  | 100 [rpm] | 120 [rpm] |
| total friction torque | 0.0054 [Nm] | 0.0023 [Nm] |

To reduce the sensitivity of the signal data to external disturbances (e.g. originating from the servos) two precautions have been taken:

- Slipring leads are bundled and provided with an external shield.
3.3 Wiring

- For the encoder feedback, redundant circuits are used to cancel out possible disturbances. In total 10 circuits are used: 2 for power, and 8 for data. The latter consist of 4 twisted pairs transmitting the pulse trains A, \(\bar{A}\) (A inverse), B, and \(\bar{B}\). Assuming a disturbance acts both A (or B) and the inverted, it can be canceled out as shown in Fig. 3.15.

\[\text{Figure 3.15: On the left, basic quadrature encoder output; and on the right complementary outputs to cancel out disturbances.}\]

The slipring wires are connected to the encoder leads of DM30 and DM15. On Arm 3, 12 wires are available for future sensors. All slipring wires are numbered on both the stator and rotor side. In Table 3.5 all connections are listed. For example, the blue wire with label 27 goes through the base to the rotor of SSR-1. On the stator side another blue wire (part of bundle 2) enters Housing DM30 through the second slot, and is connected to a blue-brown wire (again with label 27) leading to the rotor of SSR-2. On the stator side of SSR-2, the blue-brown wire is connected to a grey wire (label 3) belonging to SSR-3, and ending on link 3.
Table 3.5: Connections to the 36-circuit signal sliprings (SSR-1 and SSR-2) mounted inside the DM60 and the DM30 motor. Wires are numbered with the least significant number closest to the end of the wire, e.g. wire 31: —3 1- or -1 3—. SSR-1.1 [x o o o] denotes signal slipring 1, bundle 1, entering motor house 2 through the utmost left slot (Fig. 3.13).

<table>
<thead>
<tr>
<th>no.</th>
<th>SSR-1.1 [x o o o]</th>
<th>encoder DM30</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>red</td>
<td>1. red</td>
</tr>
<tr>
<td>2.</td>
<td>orange</td>
<td>1. shield</td>
</tr>
<tr>
<td>3.</td>
<td>black-orange</td>
<td>2. black</td>
</tr>
<tr>
<td>4.</td>
<td>black-red</td>
<td>2. black</td>
</tr>
<tr>
<td>5.</td>
<td>blue</td>
<td>3. blue</td>
</tr>
<tr>
<td>6.</td>
<td>blue-white</td>
<td>4. blue-white</td>
</tr>
<tr>
<td>7.</td>
<td>blue-orange</td>
<td>5. brown</td>
</tr>
<tr>
<td>8.</td>
<td>white-yellow</td>
<td>6. brown-white</td>
</tr>
<tr>
<td>9.</td>
<td>black-blue</td>
<td>7. green</td>
</tr>
<tr>
<td>10.</td>
<td>green-red</td>
<td>8. green-white</td>
</tr>
<tr>
<td>11.</td>
<td>blue-red</td>
<td>9. orange</td>
</tr>
<tr>
<td>12.</td>
<td>red-white</td>
<td>10. orange-white</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>SSR-1.2 [o x o o]</th>
<th>SSR-2.2</th>
<th>encoder DM15</th>
</tr>
</thead>
<tbody>
<tr>
<td>13. grey-red</td>
<td>grey-red</td>
<td>1. red</td>
</tr>
<tr>
<td>14. (dark) grey-red</td>
<td>red</td>
<td>1. shield</td>
</tr>
<tr>
<td>15. black-purple</td>
<td>black-purple</td>
<td>2. black</td>
</tr>
<tr>
<td>16. black-yellow</td>
<td>black</td>
<td>2. black</td>
</tr>
<tr>
<td>17. blue-brown</td>
<td>blue-red</td>
<td>3. blue</td>
</tr>
<tr>
<td>18. blue-yellow</td>
<td>blue-white</td>
<td>4. blue-white</td>
</tr>
<tr>
<td>19. brown</td>
<td>grey</td>
<td>5. brown</td>
</tr>
<tr>
<td>20. brown-white</td>
<td>red-white</td>
<td>6. brown-white</td>
</tr>
<tr>
<td>21. green</td>
<td>blue-yellow</td>
<td>7. green</td>
</tr>
<tr>
<td>22. green-white</td>
<td>white-yellow</td>
<td>8. green-white</td>
</tr>
<tr>
<td>23. orange</td>
<td>brown-red</td>
<td>9. orange</td>
</tr>
<tr>
<td>24. orange-white</td>
<td>red-yellow</td>
<td>10. orange-white</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>SSR-3</th>
<th>SSR-1.1 [x o o o]</th>
<th>SSR-2.1</th>
<th>SSR-3</th>
</tr>
</thead>
<tbody>
<tr>
<td>25.</td>
<td>purple</td>
<td>blue-(purple)</td>
<td>1. blue</td>
</tr>
<tr>
<td>26.</td>
<td>red-yellow</td>
<td>black-yellow</td>
<td>2. purple</td>
</tr>
<tr>
<td>27.</td>
<td>blue</td>
<td>blue-brown</td>
<td>3. grey</td>
</tr>
<tr>
<td>28.</td>
<td>blue-red</td>
<td>blue</td>
<td>4. brown</td>
</tr>
<tr>
<td>29.</td>
<td>purple-white</td>
<td>black-white</td>
<td>5. black</td>
</tr>
<tr>
<td>30.</td>
<td>white</td>
<td>yellow</td>
<td>6. yellow</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>SSR-3</th>
<th>SSR-1.1 [x o o o]</th>
<th>SSR-2.1</th>
<th>SSR-3</th>
</tr>
</thead>
<tbody>
<tr>
<td>31.</td>
<td>blue</td>
<td>blue-black</td>
<td>7. white</td>
</tr>
<tr>
<td>32.</td>
<td>grey</td>
<td>brown-white</td>
<td>8. brown-white</td>
</tr>
<tr>
<td>33.</td>
<td>black-green</td>
<td>black-red</td>
<td>9. red</td>
</tr>
<tr>
<td>34.</td>
<td>black</td>
<td>red</td>
<td>10. black-white</td>
</tr>
<tr>
<td>35.</td>
<td>black-white</td>
<td>green</td>
<td>11. orange</td>
</tr>
<tr>
<td>36.</td>
<td>yellow</td>
<td>purple-white</td>
<td>12. green</td>
</tr>
</tbody>
</table>
3.4 Remarks

- Two parallel circuits are used for one of two the encoder-power leads to both the DM30 and the DM15 encoder: wire 3 and 4 to SSR-1.1 and wire 15 and 16 to SSR-1.2 (and SSR-2.2). In a non-parallel configuration, two circuits could be saved, resulting in two additional data channels from the Base to Arm 2 (no further since SSR-3 has 12 circuits).
- Because SSR-3 has only 12 circuits, SSR-2 has 12 unused circuits (on bundle 1). Currently, these wires are connected to the shield of SSR-2.1.

3.4 Operation

Lifting
Preferably, use the Base (1. in Fig. 3.1) to move or lift the frame (approximate weight 90 [kg]). If this is not possible, use a strap around Housing DM60 (4., black). Do not use the octagonal shaped, silver colored Housing DM30 (10.) to lift the robot; unless with extreme care to prevent damage to the Spherical joint (2.) at the base of the robot.

Slipring standstill
Using the power slipring to transfer power while at a continuous standstill may cause wear and/or damage to the brushes and ring surfaces due to welding effects.

3.5 Inertia parameters

After completion of the robot, Microstation Modeler was used to generate a 3D model of the robot in order to calculate the inertia tensors of each joint-link pair expressed in the appropriate link coordinates (see Fig. 3.16).

Figure 3.16: Orientation of the coordinate frames: \( O_0 \) is placed at the center-base of link 1; \( O_1 \) at the intersection of the DM60 and DM30 rotation axes; \( O_2 \) at the intersection of the DM15 rotation axis and the line parallel to the length axis of Arm 2, through its center of gravity; and \( O_3 \) at the end of the line parallel to the length axis of Arm 3 through its center of gravity. Note that Arm3 is not yet present, so \( O_3 \) is located at the end of a virtual link with length 0.3 [m].
In Table 3.6, all moving components are grouped to form joints (the motors) and links.

<table>
<thead>
<tr>
<th>part</th>
<th>subcomponents</th>
<th>mass [kg]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Joint 1</td>
<td>DM60</td>
<td>12</td>
</tr>
<tr>
<td>Link 1 (vertical)</td>
<td>Rotor extension DM60, PSR-1, Housing DM30, and Stator extension DM30</td>
<td>20.6</td>
</tr>
<tr>
<td>Joint 2</td>
<td>DM30</td>
<td>7.5</td>
</tr>
<tr>
<td>Link 2 (green)</td>
<td>Rotor extension DM30, PSR-2 and Arm 2</td>
<td>8.9</td>
</tr>
<tr>
<td>Joint 3</td>
<td>DM15</td>
<td>5.5</td>
</tr>
<tr>
<td>Link 3</td>
<td>Arm 3 (not present)</td>
<td></td>
</tr>
</tbody>
</table>

The combined inertia tensor of Link $i$ and Joint $i$ with respect to frame $i$ is given by $J_i$:

$$J_1 = \begin{pmatrix} 2.2334 & -2.6560 \times 10^{-7} & 2.7755 \times 10^{-8} & 0 \\ -2.6560 \times 10^{-7} & 0.2213 & -0.01456 & -6.1425 \\ 2.7755 \times 10^{-8} & -0.01456 & 2.2623 & -0.2270 \\ 0 & -6.1425 & -0.2270 & 32.9305 \end{pmatrix}$$  \hspace{1cm} (3.1)

$$J_2 = \begin{pmatrix} 0.1921 & 1.9029 \times 10^{-4} & -0.2068 & -2.8439 \\ 1.9029 \times 10^{-4} & 0.7340 & 0 & 0 \\ -0.2068 & 0 & 0.6474 & -1.0222 \\ -2.8439 & 0 & -1.0222 & 16.3123 \end{pmatrix}$$  \hspace{1cm} (3.2)

$$J_3 = \begin{pmatrix} 0.01238 & 0 & 0 & 0 \\ 0 & 0.01190 & 0 & 0 \\ 0 & 0 & 0.012 & 0 \\ 0 & 0 & 0 & 5.4763 \end{pmatrix}$$  \hspace{1cm} (3.3)
Chapter 4

Joint actuation System

4.1 Dynaserv motors

The Dynaserv motors are self-contained units containing precision ball bearings, magnetic components, and integral feedback. The motor is outer-rotor, providing direct motion of the outside of the motor. On the inside, the motor has a hollow core (diameter 25 [mm]) which is part of the rotor. With regard to the stator, only the base part is visible.

In Table 4.1, the main specifications of the motors are summarized. The listed maximum torque and maximum velocity cannot be achieved simultaneously, as illustrated in Fig. 4.2.

The torque of each motor can be controlled by sending a command voltage to its respective driver (± 8 [V]). Although Litton stated the motor torques listed in Table 4.1, the relation between torque and voltage is not linear. Preliminary measurements confirm the trend shown in Fig. 4.2 (taken from the on-line manual of Compumotor): Moving Arm 2 to an horizontal position (14.62 [Nm], or 49% of 30 [Nm]) required input voltages between 2.6 and 2.9 [V] (31% to 34% of 8.5 [V]).
Table 4.1: Main specifications of the Dynaserv motors. Source: Appendix B.3 and personal communications with Litton (marked by *).

<table>
<thead>
<tr>
<th>property</th>
<th>unit</th>
<th>DM60 Motor 1</th>
<th>DM30 Motor 2</th>
<th>DM15 Motor 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>max. output torque</td>
<td>[Nm]</td>
<td>60</td>
<td>30</td>
<td>15</td>
</tr>
<tr>
<td>max. velocity</td>
<td>[rps]</td>
<td>2.4</td>
<td>2.4</td>
<td>2.4</td>
</tr>
<tr>
<td>mass</td>
<td>[kg]</td>
<td>12</td>
<td>7.5</td>
<td>5.5</td>
</tr>
<tr>
<td>inertia</td>
<td>[kg m²]</td>
<td>0.023</td>
<td>0.015</td>
<td>0.012</td>
</tr>
<tr>
<td>allowed axial load</td>
<td>[N]</td>
<td>3 \times 10^4 (compression), 3 \times 10^4 (tension)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>allowed moment load</td>
<td>[Nm]</td>
<td>200 (static load), 60 (alternating load*)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>allowed velocity</td>
<td>[rps]</td>
<td>0.5</td>
<td>1</td>
<td>1.5</td>
</tr>
<tr>
<td>torque constant*</td>
<td>[Nm/V]</td>
<td>15.6</td>
<td>7.8</td>
<td>3.9</td>
</tr>
<tr>
<td>max. power</td>
<td>[kW]</td>
<td>2.2</td>
<td>2.0</td>
<td>1.6</td>
</tr>
</tbody>
</table>

Do not exceed the following velocities in order to stay below the allowed moment for alternating load, DM60: 0.5 [rps], DM30: 1 [rps], and DM15: 1.5 [rps].

Ensure that the motor phases VA, VB, and VC are connected correctly. If not, the position feedback-loop results in unstable behavior, i.e., the motor will start rotating at maximum velocity and with maximum torque in a direction opposite to the desired trajectory.

Figure 4.2: On the left, the DM series motor performance (torque vs. velocity), on the right the saturation of the motor output torque as a function of the command voltage send to the driver.

4.2 Driver cabinet

The Driver cabinet contains three Dynaserv drivers (mounted on rails for access to the jumpers and the notch filter), line filters and two brakes (for DM60 and DM30). All drivers and brakes are powered by one central power supply which can be controlled by a switch, and two emergency stops (on the cabinet and remote).
4.3 Dynaserv drivers

Ensure that the power is switched off before opening the Driver cabinet (e.g. to change driver settings). Dangerously high voltage is present inside this unit.

On the back of the cabinet are connectors for (from left to right): the control of the brakes (two × 6 pol. DIN connectors), the encoder feedback from the motors (three × 15 pol. sub D), the power to the motors (three × 5 pol. Harting), and the remote emergency stop. Furthermore, here the extended CN-1 terminals leave the cabinet.

4.3 Dynaserv drivers

Since the Dynaserv motor is a brushless DC motor, the commutation of currents is accomplished by measuring the rotor position using an encoder. The driver is powered by 220 [V] AC (cable ① in Fig. 2.1). Based on the encoder feedback (cable ②) and the internal/external controller, the motor phases V_A, V_B, and V_C (cable ⑤) to actuate the motor are determined.

Each Driver has several control modes:

- Position Control Mode (I-PD, P-P and P-I type)
- Speed Control Mode (P and PI type)
- Torque Control Mode

When using the external controller (PC and MultiQ) the driver is set to the torque control mode (for more detailed information on the required jumper settings, see Appendix B.3)

Furthermore, each driver has a first-order lag filter and a notch filter to reduce vibrations. At present both are unused.

Although, the manual recommends a maximum filter offset voltage of 0.05 [V], this can, and should be set below 0.005 [V] to avoid a noticeable offset in the torque command voltage.

The control system interfaces with each driver through the CN-1 terminal (cable ② in Fig. 2.1). From the Driver cabinet all CN-1 connections are extended to the casing of the MultiQ plug-in board.

When switching on the power, a surge of approximately 8 [V] occurs at the driver outputs (CN-1, VEL, POSN and TORQ in Fig. 2.1)

4.4 Dynaserv brakes

The DM60 and DM30 are equipped with an electro-mechanical brake to bring the robot to a controlled stop. Each brake consists of a board with power electronics and relays to short the motor phases (V_A, V_B and V_C) either directly (at low velocities), or by using a dissipative RC-circuit (at high velocities). Simply stated, the brake is wired by cutting the motor cable (cable ⑤ in Fig. 2.1), and connecting both sides to the board. For more details see Appendix B.4.

Each brake has a separate power supply, and is mounted inside the Driver cabinet. On the backside of the Driver cabinet are connections to engage the breaks. If the power is switched off (power failure or emergency stop), the brakes are engaged automatically.
Chapter 5

Control System

5.1 MultiQ plug-in and terminal board

The MultiQ I/O board from Quanser Consulting has $8 \times 13$ bits ADC, $8 \times 12$ bits DAC, 8 digital I/O, 6 encoder inputs, and 3 hardware timers. Also included is a terminal board with several connectors.

Using separate test software the I/O times of the MultiQ board can be measured:

- Digital input (16 [bit]) 5 $\mu$s
- Digital output (16 [bit]) 2 $\mu$s
- Encoder read (24 [bit]) 5.5 $\mu$s
- ADC (16 [bit]) 20 $\mu$s
- DAC (16 [bit]) 5 $\mu$s

For a computed torque controller based on encoder data only (see Fig. 5.2), the total I/O time amounts to 31.5 $\mu$s. Using external timing it was verified that a sampling frequency of 1 kHz can be achieved without interruptions. Higher frequencies require the use of one of the on-board clock timers, and were not yet tested.

The terminal board (see Fig. 2.3) is located inside a casing with three 50-pole connectors. Here, the three driver-CN1 extensions (originating from the back of the Driver cabinet) are inserted. Inside the casing, the MultiQ inputs, and outputs are connected to the right CN1 pins. Furthermore, through here, the servo enable signal is fed to the drivers (originating from an independent power supply).

Connections to the three driver-CN1 extensions inside the casing:

- Encoder read (pin 13, 14, 29 and 30):
  The 4 encoder outputs are connected to a AM26LS32 line driver (white=[13. A+], brown=[14. A-], green=[28. B+], yellow=[30. B-], black=ground). The outputs are connected to the first three encoder DIN connectors (0, 1, and 2).

  Use the emergency stop and perform a (hard) reboot of the PC containing the MultiQ I/O board, if the encoder read-out becomes discontinuous or otherwise erratic.

- Torque command (pin 49 and 50):
  Pin [49. VIN] is connected to the core of a coax cable, the shield is connected to pin [50. AGND]. Using tulip connectors the three cables are plugged in analog MultiQ output 0, 1 and 2.
• Enable servo (pin 23 and 24):
  See Section 5.2.

Although, an analog velocity output is present (pin 17 and 18), currently, it is unused because:

• Differentiating the encoder yielded better (without a bias) results.
• A surge of 8 [V] occurs at the CN1 outputs when the power is switched on. Without additional overload protection, this may cause damage to the MultiQ board:

  Do not apply more than 5.3 [V] to the (analog) input of the MultiQ I/O board to avoid overload.
• A/D conversions are the most time consuming operations of the MultiQ board.

5.2 External controls

Enable Servos and Brakes  Both servos and brakes are enabled, i.e., made ready for operation, by closing a circuit (like the one shown in Fig. 5.1) between two pins of each driver-CN1 terminal (inside the terminal board casing), and each brake-CN2 terminal (extended to the back of the Driver cabinet).

Do not use the enable circuit as an alternative emergency stop. There is a considerable delay (approximately one second) before opening the circuit has effect.

Figure 5.1: Example (enable) circuit for both the three drivers and the two brakes. On the driver-CN1, the circuit is between pin [24. VCC], and [23. SRVON]. On the brake-CN2, it is between [3. VCC] and, [4. SRVON]. Normally the circuit is open, i.e. the SRVON-signal is set to HIGH, and driver (or brake) does not accept commands. After closing the circuit, i.e., setting SRVON to LOW, the driver (or brake) is ready for operation.

Emergency Stop

Always keep the (remote) emergency stop within reach while operating the robot. Use the emergency stop if error messages occur after the Simulink controller has been started; if the control software or the operation system crashes; or if the encoder read-out becomes discontinuous or otherwise erratic.
5.3 Personal Computer

5.3.1 PC Hardware

The PC configuration consists of a Pentium Pro 200 with 32 MB RAM, it has three free PCI slots and three ISA slots.

5.3.2 PC software

The PC software for controlling the robot consists of several programs (between brackets are the untested updated versions):

- The operating system Windows 3.11 (Windows 95).
- Matlab 4.2c (Matlab 5.1) and Simulink 1.3 (Simulink 2.1) to design, analyze, and implement the controller.
- The Real-Time Workshop 1.1C (RTW 2.1) to convert the controller – consisting of Simulink blocks – to C-code.
- The Watcom C-compiler 10.6 (Microsoft visual C compiler) to compile the generated C-code to an executable.
- Wincon V2.0 (Wincon V3.0) to run the executable under windows while displaying and modifying selected Simulink variables.

When performing an experiment, do not start the control algorithm before the servos are powered and enabled; the introduced tracking error may cause a large and unwanted response.

An example of a Simulink controller is shown in Fig. 5.2. This diagram also displays the trajectory generation, the blocks for the data acquisition board, and the feedback-loop signal flow.

![Simulink block scheme of a Passivity Based Computed Torque Controller](image)

Figure 5.2: Simulink block scheme of a Passivity Based Computed Torque Controller.

After modifying the Simulink controller, always perform a “test run” to see if error messages occur.
Chapter 6

Maintenance and inspection

Motors  Check daily if changes in noise level or excessive vibration occur. After 20,000 hours or 5 years, some parts may need to be replaced. When an overhaul or motor disassembly is required, contact Litton. (Appendix A).

Driver cabinet  To prevent problems due to poor insulation, periodically remove accumulated dust inside the Driver cabinet and the individual drivers.

Sliprings  To prevent problems due to poor insulation, periodically remove accumulated dust inside the Base (1.) and Pedestal; inside Housing DM30 (10.); and behind the (silver colored) cover on Arm 2 (16.).

Additional frame parts  The bearing 165-120-22 (8.) provided by the CTD has a lifespan of at least 15 year, and requires no maintenance.

Control system  Periodically, defragment the hard disk of the PC.
Appendix A

Addresses

Litton Precision Products International

product: Dynaserv servos, sliprings and brakes
address: Griendstraat 10
2921 LA Krimpen a/d Yssel
The Netherlands
tel. 0180-596888
fax. 0180-596899
contact person: Rolf van de Weijer
littonnl@worldaccess.nl
web site: http://www.litton.com
technical support servos: Reto Muff (Switzerland)
tel. 0041-1313 1450
fax. 0041-1313 1255
email: RMuff94444@aol.com

CTD

product: frame manufacturing
contact: Erwin Dekkers
W-hal 1.58
tel. 3356

GTD Groep Electronica/Energietechniek

product: Driver cabinet
Realization: B. Viveen
tel. 3494
Quanser Consulting Inc.

product: MultiQ I/O board and Wincon
address: 102 George Street
          Hamilton, Ontario
          Canada
tel. 001905 527-5208
fax. 001905 570-0177
web site: sales@quanser.com
          http://www.quanser.com

Scientific Software Benelux

product: Matlab
address: Bleulandweg 1B
          2803 HG Gouda
tel. 0182-537644
fax. 0182-570380

AC-2000 Software Ingenieurs

product: Watcom C/C++
address: Standerdmolen 8-302
          3990 DB Houten
          PO box 83
tel. 030-635 1840
fax. 030-635 1839
Appendix B

Components

In the following sections several component manuals and drawings are included:

B.1 Wincon software manual ................................................................. 38
B.2 MultiQ board manual ................................................................... 70
B.3 Manual: Dynaserv DD Servo-Actuator DM/SD series .................. 91
B.4 Manual: BE-A/B Type Dynamic Brake ........................................ 113
B.5 Cabinet manual ........................................................................... 123
B.6 Power sliprings ............................................................................ 133
B.7 Signal sliprings ............................................................................ 135
B.8 Conceptual design ....................................................................... 138
B.1 Wincon software manual

Instruction manual, 32 pages

WinCon V2.0
Quanser Consulting Inc.
WinCon™, Windows™ based
Realtime Controller for SIMULINK™
Quanser Consulting Inc

1.0 General Description

WinCon is a realtime program that will run SIMULINK controllers in Realtime under Windows. In order to use the controller you will need the following:

- MATLAB™
- SIMULINK™
- Real-Time Workshop™
- Watcom™ C++ compiler version 10.a or higher
- Windows™ 3.11 or Win95™
- DOS™ version 6.2
- 8 MB RAM
- IBM 486 compatible computer or higher, with math co-processor
- 1 Mbytes of free hard disk space

You will also need the SIMULINK device drivers for the data acquisition board you are using (A/D, D/A, Encoder input).

To date the following devices are supported:

- MultiQ : A/D, D/A, Quadrature Encoder interface board (Quanser Consulting Inc)
- DT2811: A/D, D/A interface board (Data Translation)
- CIO DAS16: A/D, D/A interface board (Computer boards)

The general functional description of WinCon is the following: Draw the SIMULINK controller and generate the realtime code (RTC) using the Real-Time Workshop. Run WinCon and load the RTC you created into WinCon and run it. You now have the controller you created running in realtime, at the sample rate you selected, under Windows. You can plot all the "Scope" variables you had selected in SIMULINK while the controller is running using the plotting facilities of WinCon. The data buffer allows you to store up to 16348 points per variable. Scope data can be saved to disk at will. You can change windows while the controller is running and start another Windows application without interfering with the performance of the running controller. Specifically you can start up SIMULINK, load the diagram of the controller you are running and change parameters in realtime while the controller is running in order to instantaneously observe the effects of the gains, filters or other parameters on system behaviour! Once you are satisfied with the performance, you can start WinCon independent of SIMULINK, load the RTC and run it in realtime. You can also run WinCon along with the RTC from SIMULINK on a computer that does not have SIMULINK on it.

WinCon is a trademark of Quanser Consulting Inc. All other trademarks are the properties of their respective owners.
2.0 Setting up

*It is assumed that you are well versed in Windows, MATLAB and SIMULINK*

2.1 Install WinCon

- Start Windows
- Insert the WinCon distribution diskette into your a: drive
- From Program Manager Select File
- Select Run from the Pull Down Menu
- Type a:setup Hit enter
- Select Full Installation.. Follow the instructions on the screen

2.2 Copying sample files

- make a directory c:\winsim on your hard drive
- copy the contents of a:\winsim to c:\winsim

2.3 Configuring the directories for the makefiles

2.3.1 Using any text editor (eg notepad) edit the file
c:\matlab\codegen\rt\tmf\wc_watt.tmf

and modify the lines

```
WATCOM_ROOT = C:\WATCOM
WINCON_ROOT = C:\WINCON
```

match the drive and directories for where you have installed WATCOM and WinCon

2.3.2 Re Compilers

- Watcom 10.0a users: change all references of "binw" to "binb" in the file
c:\matlab\codegen\rt\tmf\wc_watt.tmf (global substitute "binw" to "binb")

- Watcom should be able to generate Win3.11 code even if you are running on Win95 (ie 16 bit compile)

2.3.2 Edit autoexec.bat and ensure that c:\watcomc\binw is in the PATH

also ensure that the lines:

```
SET INCLUDE=C:\WATCOM\H;C:\WATCOM\H\WIN
SET WATCOM=C:\WATCOM
SET EDPATH=C:\WATCOM\EDDAT
```

Are appropriately directed in the autoexec.bat file
2.3.3 Edit `c:\matlab\matlabrc.m` and add the following directories to the MATLABPATH:

- `c:\winsim`
- `c:\matlab\codegen\rt\wincon\devices`
- `c:\matlab\codegen\rt\dos\devices`

For example, the following should be part of your `c:\matlab\matlabrc.m`:

```matlab
matlabpath([...
'C:\winsim',...
'C:\matlab\codegen\rt\wincon\devices',...
'C:\matlab\codegen\rt\dos\devices',...
'C:\MATLAB\toolbox\local',...
'C:\MATLAB\toolbox\fuzzy\fuzdemos',...
]);
```
3.0 Operation

The guidelines for operating WinCon are given below. In order to use the full potential of WinCon, remember that in Windows you switch active windows by pressing [Ctrl Tab]. You can also Switch to a desired Window by entering [Ctrl Esc] and selecting the Task from the Task List (or use [Alt Esc]).

Figure 3.1 winlib supplied with WinCon

1) Install a data acquisition board into your computer. To date, the following boards are supported by WinCon/SIMULINK:

   a) MultiQ - Quanser Consulting
   b) DT2811 - Data Translation
   c) CIO DAS16 - Computer Boards

If you have a different board, Quanser Consulting will write the appropriate drivers at a nominal fee.

2) Draw the controller in SIMULINK. Draw the SIMULINK controller as you usually would. The data acquisition blocks and other WinCon blocks are accessed by typing winlib in the MATLAB Command Window. Make sure the Data acquisition blocks are configured for the right base address of the device. If there is any data you would want to monitor while the controller is running, attach them to a SIMULINK Scope and select an appropriate name for the block.

Note the sample time of the data acquisition blocks. You must ensure that the sample time of these blocks is the same as the actual sample time of the controller. In case of MultiQ drivers, this is automatically performed by using the cg_get MATLAB function as shown in Figure 3.2. You may override this by typing in the sample time you want.
3) **Configure the Real Time Code Generator.** Select Code from the menu of the drawing of the controller. Select Real Time Options:

   a) **Step size:** set the controller sampling period (e.g., 0.005 seconds)
   b) **Template Makefile:** `wc_watc.tmf`
   c) **Build Command:** `make_wc`

   **Note about Template makefile:** You can also use `wc_watg.tmf`. You may make these choices the default by editing the file: `buildopt.m` (typically under `c:\matlab\toolbox\local`).

4) **Configure the simulation:** Select Simulation in the controller drawing. Select External (✓). Also from Simulation select External Options and enter `wc_comm` under File for External Simulation. Note that from this point on, the word "simulation" is a misnomer. You will be running a realtime controller but it will be referred to as Simulation from within SIMULINK.

5) **Save the drawing in a file** under the directory `c:\winsim` (or any other directory you create). It is highly recommended that the controllers and the associated `.m` files reside in a separate directory from MATLAB or WinCon. It is also good practice to start the file name with the characters `wc_` in order to find them easily later on.

6) **Change the MATLAB data directory** Switch to the MATLAB Command Window ([Ctrl Esc]) and type `cd c:\winsim` (or the directory you chose in part 5). This ensures that all generated code and associated files are stored in that directory. If you do not change directory from inside MATLAB, the generated code will be added to the MATLAB directory and may create unnecessary confusion.

7) **Generate and Build Realtime Code** Switch back to the SIMULINK drawing ([Ctrl Esc] or [Alt Tab]) and select Code and Generate and Build Realtime. This starts the compilation of the diagram in to C code that can be linked with WinCon. The output file for this operation has the extension `.wcl`. This means it is a SIMULINK controller that can link with WinCon.
8) Start the Controller  From the SIMULINK drawing, select Simulation and Start. This starts WinCon, loads the controller for which you just generated code and starts the realtime controller. Note that this operation also starts a SIMULINK simulation which is automatically paused. You can also start and stop the simulation by typing start and stop from within the MATLAB Command Window (MCW). The start command automatically pauses the simulation but keeps the controller running. You may find this a more convenient method of starting the controller from MATLAB. Note that these commands start and stop the controller in the block diagram in the last window that you had clicked on before going to the MATLAB command window. Make sure you issue the command after you have clicked on the window of the controller you want to start. If WinCon is already up and the controller you want loaded, you can start the controller from any window by entering [Alt Pause] on the keyboard.

To stop the controller, you can either

- Stop the controller from any Window with the [Pause] key on the keyboard (this is the fastest and easiest way)
- Stop Simulation in the SIMULINK Window [Ctrl T] or issue stop from the MCW
- Stop the controller from the WinCon Main Window (WMW), click on the Traffic Light Button, or select File-Stop from the pull down menu.

You can also use the buttons that are in winlib (Figure 3.1). You can drag copies of these buttons into the SIMULINK diagram itself. Double clicking on these buttons will start and stop the controller.

The following table lists all the possible methods of starting and stopping the controller:

<table>
<thead>
<tr>
<th>From where</th>
<th>How</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>From SIMULINK Menu of Diagram</td>
<td>[Ctrl T] or Simulation Menu</td>
<td>SIMULINK up, starts WinCon, loads diagram RTC</td>
</tr>
<tr>
<td>From SIMULINK Diagram itself</td>
<td>Double click Start button</td>
<td>SIMULINK up, (Obtain Stop and Start from winlib)</td>
</tr>
<tr>
<td>From MATLAB Command line</td>
<td>Start, Stop</td>
<td>Last SIMULINK diagram that was active</td>
</tr>
<tr>
<td>From WinCon</td>
<td>Traffic light button Or File/Run Menu item</td>
<td>WinCon loaded</td>
</tr>
<tr>
<td>From Anywhere</td>
<td>[Alt pause], [Pause]</td>
<td>WinCon loaded</td>
</tr>
</tbody>
</table>
Figure 3.5 WinCon Main Window
9) Plotting realtime and saving data

"Scope" data selected in SIMULINK may be plotted in realtime by clicking on the Scopes/Workspace button in the WinCon main window. Select the variable you want to plot and click OK. If you want more than one variable plotted on the same graph (eg reference and output) click on the first variable then hold down the [Ctrl] key and click the left mouse button on the other variables you wish to plot. Once you are satisfied with the variable list click [OK].

Once the plot is showing you can alter some of the plot parameters such as Update Frequency, Buffer duration, axes etc... You can also select if you want plot windows to be a child of WinCon or be treated as an independent window accessible from the Windows Task List (Ctrl Esc ). This is done using the Plot / Are Children menu in the WinCon Main Window.

If you would like to open another plot window, you may do so using the Scopes/Workspace button or by clicking on File - New in any active plot Window. From an active plot Window you may switch to the WinCon main window or any other plot that is in the background using the Windows Task List (Ctrl Esc ) in the Plot Window.

If you want to change the variables being plotted in a given Plot Window, click on File/Variables in the Plot Window and select the new variables you want to plot.

You can also save the drawn data in two formats: (.m and .mat). Select File from the Plot Window and Save. Select the directory (we highly recommend that you place the data files in a different directory eg c:\winsim\data ) Select the type of file you want to save from the selection list. Caution: If you select a " .m " type file for output, make sure you do not use the same file name as your controller as you will overwrite your controller diagram with the data file!

Enter a filename (no extension) and click [OK]. The " .m " file will contain an ascii file of the variables you plotted in WinCon as well as a MATLAB plot statement. You can then switch to MATLAB and enter the file name. This will automatically plot the data you saved. If the file is saved as a " .mat " file, it will be in binary format and you will have to load it into MATLAB using the MATLAB load command. The data will consist of individual vectors with the Scope names you had chosen. If the variable name has a blank in it, the blank will be replaced with an underscore(_). There will also be a variable holding the time vector for the data named T. Both of these plot features are very useful if you want to perform parameter estimation or make hard copies of the plot to include in a report.
In summary, the following features are available from a Plot window menu:

**Under File**
- **New** Open a new plot window
- **Save** Saves the plotted data in a .m or .mat file
- **Variables** Select which Scope variables you want to plot
- **Close** Closes the plot window

**Under Update**
- **Fixed Interval or Realtime**: Selects whether you want to plot in realtime or plot just one trace and hold the trace on the plot (single sweep).
- **Buffer (Tb)**: Is the duration of the data buffer for the plot. This is a circular buffer of duration Tb. Only the last Tb seconds of data are available for plotting and saving.
- **Frequency (Pf)**: The sampling period of the data buffer. The fastest frequency possible is the actual sampling frequency of the controller. Pf may be larger than the actual sampling period (Ts) thus allowing decimation. You can then save a longer duration of data by compromising resolution. The maximum number of points per trace is 16000.

**Under Axes**
- **Auto Scale or Fixed**: Select whether or not the scale of the y axis is adjusted to fit the last sweep maximum and minimum. In Fixed mode, you select the scale.
- **Time (Tp)**: This is the duration of the displayed plot along the time axis. Tp is always less than or equal to the buffer duration Tb. If Tp is less than Tb and you are plotting in Fixed mode, then a scroll bar appears under the plot that lets you scroll though the data over the last Tb seconds.
- **Grid**: Self-explanatory

**Under Window**
- **Legend**: Self-explanatory
- **Main Window**: Takes you back to the WinCon main Window. A handy feature to help you navigate through many open windows.
- **Other plot windows list**: Takes you to another open plot window. Also helps you navigate through a maze of plot windows.

Furthermore, if you want to access the plotting facilities straight from the SIMULINK diagram, you can drag and drop the [plot] button from winlib (Figure 1.1) into the SIMULINK diagram. Double clicking on this box is just like clicking on the [Scopes/Workspace] button in WinCon.
WINCON SETUP & OPERATION

10) Changing controller parameters

If WinCon and SIMULINK are up, and the loaded controller in WinCon is the same as the diagram in
SIMULINK, then parameters changes in the diagram automatically take effect in the running controller. It
is usually good practice to stop the controller in SIMULINK, change the parameter and then start it
again. This is especially true if you are changing filter parameters drastically and the system you are
controlling is sensitive to parameter changes. If the diagram is altered in any way (eg, cutting connections,
adding blocks) you will need to regenerate realtime code as in step 7.

11) Saving and Running a WinCon Project

You may save a running WinCon controller as an independent project, including the status of all plots you
had opened. Do this from the WinCon main window and select File/Save as. Select a directory and
filename then click OK. You can then load this new project file (.prj extension) into WinCon in the future.
You can start the loaded project from WinCon by entering [Alt Pause] or clicking on the Traffic Light
Button (TLB) to the right of the Time display in WinCon. Starting the controller from WinCon, will start the
controller that was originally saved during Realtime Code Generation. If you want to use a controller with
different parameters, Stop the controller using the pause or TLB. Click the To SIMULINK button. This
starts MATLAB, and loads the SIMULINK diagram of the controller you have loaded into WinCon. You can
then Start from SIMULINK which will now start the WinCon Controller and will allow you to change
controller parameters. Once you are satisfied with controller performance, it is advisable to generate the
final realtime code so that the project you eventually save will contain the desired parameters. This way
you can start WinCon with a project that contains your "final" controller.

As in any application, practice makes perfect. Try out a few simple systems (our examples) first before
you start a serious project. Learn the sequence of operations, you'll eventually become an expert!

12) Notes on Step size
In WinCon, you may select the source of the timer interrupt that controls the
sampling frequency. For sampling frequencies below 1KHz, you may select the Windows System clock.
This is done by selecting Clock, in the I/O Card option of WinCon (on the screen, top right). If you want to
sample at a higher frequency, you have to use an external hardware clock tied to an interrupt line on the
PC. MultiQ supports this option. If you have other cards, a WinCon driver must be written for that card.
Contact Quanser Consulting in order to have a Clock driver written for your card. The card must be
equipped with a programmable realtime clock that can be tied to interrupt lines.

Be careful when selecting the step size. Make sure that it is not too small. Read the section on Maximum
Sampling Frequency under Hints and Troubleshooting at the end of the manual.
4.0 Examples

4.1 Simple loopback

This is an example that everyone should try. Not only do you practice the use of WinCon but you also test your A/D, D/A board. The example is a loopback system. You need to wire the output of channel #0 on the D/A subsystem of your board to the input of channel #0 on the A/D subsystem on your board. This is shown in Figure 4.1.1.

The "controller" will simply apply a sine function to the D/A output and will measure the voltage from the A/D. No other hardware is required for this example.

1) Start MATLAB for windows and type cd c:\winsim. Type ws-lpbk This loads the diagram shown in Figure 4.1.2. You could have generated this diagram yourself using SIMULINK.

2) If you have a Multi board you can skip to step 4

3) If you do not have a Multi board installed in your system, switch to the MATLAB Command Window and type doslib. Double click on the card you have from the list of cards available in doslib. Copy one A/D block and one D/A block to the ws_lpbk diagram. Delete the Multi A/D block in the ws_lpbk diagram and replace it with your board's A/D block. Delete the Multi D/A block and replace it with the D/A block for your card. Make sure the drivers are configured for the correct base address and channel. You can select the A/D board you are using from the doslib or winlib Simulink blocks.

4) Select Code from the menu and click on Generate and Build Realtime. This generates the realtime code for the diagram. Wait until the compilation is complete. A DOS window is automatically opened and upon completion the message:

* * * * * Make of ws_lpbk.wcl is complete * * * * *

is printed in the DOS Window. You can then close the DOS window.

5) Click on Simulation/Start in the ws_lpbk SIMULINK window. This starts WinCon, loads the compiled controller and starts running it at the sampling frequency specified in Real Time Options in SIMULINK. Alternatively, you could switch to the MATLAB Command Window and type start to start the controller and automatically pause the simulation. (See section 3.8).

6) Switch to WinCon using [Alt Tab]

7) Click on Scopes/Workspace. The names of scopes defined in ws_lpbk will appear in a list. Plot both variable on the same graph by selecting [Click] Command then [Ctrl Click] on Measured and then select

LPBK 1
[OK]. Note that the commanded voltage has an amplitude of 10 volts. However the saturation block limits it to 5 volts. Therefore the real output to the D/A is a clipped sinusoid at 5 Volts. This voltage is measured by the A/D (due to loopback wire) and displayed under the variable Measured. The plot will display a 10 V peak sine wave and a clipped sine wave. The command and the measured value should otherwise superimpose. If we apply a voltage smaller than 5 Volts p-p at the command, then the measured and command would (should) match exactly.

8) Click To SIMULINK from WinCon. This takes you to SIMULINK. Stop the controller [Ctrl T]. Click on the signal generator and change the source to a triangular wave. Keep the Plot Window of WinCon open so you can monitor the changes you are making. Start the Controller [Ctrl T] and note that the change you made took effect in WinCon by observing the plot. Change the signal generator parameters and observe that you are changing parameters in realtime without interruption in control!

9) Select a signal source different from the original sine wave (eg. triangular), start the controller and switch to WinCon. Plot the variables as before. With the plot active, save this "controller" as a project by selecting File - Save as from the WinCon main window and select the filename as c:\winsim\ws_lpbk

10) Click on To SIMULINK. Stop the controller (or issue stop from MATLAB). Save the changed diagram (with a triangular wave as signal source). Close SIMULINK. Close MATLAB. Close WinCon.

11) Open WinCon from scratch. From File/Open load c:\winsim\ws_lpbk.prj. This is the project you saved in step 9. Start the controller directly from WinCon. What are the plots showing? Triangular or Sine waves? It will be sine waves because the compiled code in step 4 was using a sine wave! Stop the controller using the Pause button. Click on To SIMULINK. This starts MATLAB and loads ws_lpbk.m diagram into SIMULINK. Click on Simulation - Start. Switch to WinCon. What do the plots show now? A triangular wave because you just loaded the parameters from the SIMULINK diagram into the running controller! This is where you have to be cautious. WinCon runs realtime generated code that can only be changed via SIMULINK or re-compiling.

Figure 4.1.2 SIMULINK diagram of loopback "controller"
4.2 PID Controller

This example demonstrates a PID controller that positions the output of the Servo plant shown in Figure 4.2.1. Start MATLAB from Windows, change directory to c:\winsim and type, ws_svr_r This brings up the diagram of the controller shown in Figure 4.2.2. (Note this example is shown in Win95).

The controlled output of the Servo Plant is measured using the A/D channel #0. The command to the system is generated from the Command Generation block. The difference between the command and the actual angle is the error signal which is multiplied by a proportional gain Kp. The error is also integrated and multiplied by a gain Ki. The measured variable (eg motor angle) is also differentiated using a high pass filter and multiplied by a gain Kd. The three gains are the PID gains of the controller. The sum of these terms is applied to the D/A output channel #0. The signal from the Command Generation block could be anything you want. In this example it is a signal generator applying a square wave of +/- 45 degrees. You may replace this block with another A/D channel for example which has a potentiometer attached to it. The controlled system will track the signal generated by the command generation block. The PID gains in the Figure could be constants or variables defined in the MATLAB Command Window. If you enter variable names in the PID gains (or in any other block for that matter) they must have a value assigned to them before you can generate the realtime code. Once you generate the code, the controller (.wcl extension file) will have these gain values hard coded into it. Changing the values of the variables in the MATLAB command window will not immediately change the value in a running controller. You will have to either stop and start the simulation or, open the dialogue box for the block which uses the variable name init and then click [OK]. This can be done while the simulation is running.

To start this controller perform the following operations:

1) Start MATLAB in Windows (if it is not already up. Do not start multiple copies)
2) type cd c:\winsim in the MATLAB Command Window
3) Load ws_svr_r by typing ws_svr_r in the MATLAB Command Window. This loads the SIMULINK drawing shown in Figure 4.2.1
4) Click on Code. Select Generate and Build Realtime from the Pull Down Menu. This starts the real time code generator and compiles the code to be linked with WinCon.
5) Once the code has been generated, you will see a DOS window with the message:

* * * * * Make of ws_svr_r.wcl is complete * * * * *

You can close the DOS window to minimize clutter. MATLAB does not close the DOS windows automatically.

6) Click on Simulation/Start.

7) The system starts WinCon, configures it and starts running the controller in realtime.

8) You can now plot and change gains as you please. You can save the project in WinCon so that the
controller can be started from WinCon (without SIMULINK) and run from WinCon. Figure 4.2.3 shows the response of the output to the square wave input when the derivative gain is zero. Note the overshoot and ringing. Figure 4.2.4 shows the response when the derivative gain is set to .002 (while the controller runs). Note the improved behaviour.
4.3 Inverted Pendulum

This example covers the Inverted Pendulum IP-01 shown in figure 4.3.1. The system is wired as shown in Figure 4.3.2.

The inverted pendulum has two sensors, one for the cart position and one for the inverted pendulum angle. These sensors are attached to the MultiQ A/D inputs channel 0 & channel 1 respectively using the Quanser Quick Connect Module. The system is driven by a DC motor which is powered via a power amplifier on the power module PA-0103. The input to the power amplifier is obtained from the D/A channel #0 on the MultiQ board.

The required controller is a state feedback controller that feeds back the following voltage to the motor:

\[
V_m = [k_1 \ k_2 \ k_3 \ k_4 \ k_5]\begin{bmatrix} e \\ \dot{e} \\ \alpha \\ \dot{\alpha} \\ \int e \end{bmatrix}
\]

The control system design is beyond the scope of this manual. For details see "A comprehensive and modular laboratory for control system design and implementation" by Quanser Consulting.

Load the controller ws_pend into SIMULINK.

The diagram shown in Figure 4.3.3 will appear. The measurements are voltages from the sensors and are measured using A/D channel #0 and A/D channel #1. These are applied to bias removal blocks and then fed to a calibration and differentiation block. The output of this block results in the states \([x \ \dot{x} \ \alpha \ \dot{\alpha}]\) A signal generator generates the commanded cart position which is subtracted from the actual cart position to obtain the cart position error \(e\). These are then fed to the state feedback controller which computes the integral of the error and applies the desired gains \(k_i\) to \(k_s\) to obtain the applied voltage to the motor. This
values is then sent to the D/A channel #0 on the MultiQ to drive the motor through the power amplifier

4.3.1 Bias removal blocks
The two bias removal blocks are "grouped" SIMULINK blocks. Viewing one of these blocks (double click on the block) reveals the structure shown in Figure 4.3.4 This block has two inputs. In_1 is the voltage measured from the sensor while In_2 is from the clock which keeps track of time from the moment the simulation starts. The clock controls the position of a SIMULINK switch. As soon as the time is greater than 0.1 seconds (set in the dialogue box of the switch), the output of the switch is the value applied at the first line. In the first 10 mseconds of the simulation, the output of the switch is the input applied at line 3. So, for the first 10 msec., the output of the block is (In_1 - In_1) = 0. When the switch trips over, the output of the switch is maintained to the last value before it switched over. Thus, the block defines the zero position of the device to be the position it was in at t = 10 msec. The purpose for this block, is so the user can hold the pendulum upright and the cart at x = 0. The user then starts the controller which for the first 10 msec does not do anything but measure the bias on the sensors. It then uses the measured bias to obtain the actual angle and cart positions.

4.3.2 Calibration amd Differentiation block
This block converts the cart position voltage and pendulum angle voltage to the appropriate units. The cart sensors measures 8 cm per volt while the pendulum angle sensor measures 14.5 degrees per volt. The voltages are first low-pass filtered and then multiplied by the appropriate calibration constants as shown in Figure 4.3.5 After converting the voltages to the desired units, the variables are applied to Scopes, which will be available in WinCon for plotting. The two signals are then fed to limited differentiators to obtain the cart velocity and pendulum angular velocity. Note that we use limited differentiators rather the "pure" derivatives to reduce noise.

4.3.3 State Feedback Block
The last block before the output block is the state feedback controller which implements the feedback gains to obtain the voltage that will be applied to the controller. The system requires five gains which are implemented as "sliders" in SIMULINK. While the controller is running you can click on the sliders to change the gains during operation. Note that the integrator is a "limited integrator" block which we suggest you always use instead of a regular unlimited integrator. This way, if the integrator reaches the maximum value it will stop integrating and will immediately start integrating in the opposite direction when the input signal changes sign. This is a more effective method of implementing an integrator in a real system. Note that the integrator state is automatically reset whenever the controller is stopped and re-started. The output of this block is fed directly to the MultiQ D/A block. The channel of the D/A is selected in the D/A dialogue box.

4.3.4 Safety Stop As a safety feature, the controller is automatically turned off if the pendulum angle exceeds 10 degrees in either direction.
Figure 4.3.3 SIMULINK controller of Inverted pendulum

Figure 4.3.4

Figure 4.3.5
4.3.4 Running the controller

To start this controller perform the following operations:

1) Start Matlab in Windows (If it is not already up. Do not start multiple copies)
2) type cd c:\winsome in the Matlab Command Window
3) Load ws_pend by typing ws_pend in the Matlab Command Window. This loads the SIMULINK drawing shown in Figure 4.3.3
4) Click on Code. Select Generate and Build Realtime from the Pull Down Menu. This starts the real time code generator and compiles the code to be linked with WinCon.
5) Once the code has been generated, you will see a DOS window with the message:

* * * * * Make of ws_pend.wcl is complete * * * * *

You can close the DOS window to minimize clutter. Matlab does not close the DOS windows automatically.

6) Hold the pendulum upright and the cart in the middle of the track.
7) Click on Simulation/Start.
8) The system starts WinCon, configures it and starts running the controller in realtime.
9) You can now plot and change gains as you please. You can save the project in WinCon so that the entire controller can be started from WinCon (without SIMULINK) and run from WinCon.

Figure 4.3.7 Plots obtained in realtime from WinCon

Figure 4.3.7 shows the WinCon plots that are obtained. The variables plotted are the signal generator, the cart position and the pendulum angle. Note the non-minimum phase behaviour in cart position.
4.4 Linear flexible joint frequency response

In some cases, you may want to evaluate the frequency response of a system. This example demonstrates the linear flexible joint system (LFJC) and its controller performance using a sine-sweep signal. The LFJC consists of the Inverted pendulum cart (IP-01) coupled to a load cart via a spring as shown in Figure 4.4.1. The input to the system is the voltage applied to the pendulum cart while the output is the position of the load cart. Note that the load cart (on the right) is not powered. The controller block diagram is shown in Figure 4.4.2. It consists of a state feedback controller that feeds back the positions of the two carts and their respective velocities. The commanded position is subtracted from both cart positions in order to maintain the length of the spring at rest. The interesting feature in this system is that you can change the gains from the feedback cart as shown in Figure 4.4.3 which is an expansion of the block named Control. By setting the two gains from the load cart to zero, we are not using output feedback and positioning the output cart by simply positioning the input cart and letting the spring do the rest. A better controller is obtained by feeding back the position and velocity of the output cart as well (non-zero gains). In order to evaluate the controller, we could apply square wave inputs to the command and look at the overshoot and damping. In this example however we will apply a sine wave of varying frequency and observe the resonance in the system (almost a bode plot.. but not quite).

![Figure 4.4.1 Hardware for flexible joint experiment](image)

The Command is generated from the Chirp block, which when unmasked shows the diagram in Figure 4.4.4. A sawtooth signal is applied to a MATLAB function block which generates a sine function whose frequency depends on the value of the sawtooth signal. The sawtooth is held at zero for the first and last five seconds of the period. The rate of change of the chirp signal is selected to be unity per second and is thus changing like time. The reason we use a sawtooth signal instead of the Time block is that we want to repeat the sine sweep every 30 seconds. Figure 4.4.5 shows the input command (chirp signal) and load cart position when the feedback gains from the load cart are set to zero. Note the resonance in the system. Using LQR designed gains for the system, we can dampen the response to obtain the plots shown in Figure 4.4.6. The FFT's of these signals are compared in Figures 4.4.7 and 4.4.8. All of the plots
Figure 4.4.2 Controller for flexible joint

Figure 4.4.3 Chirp generator

are MATLAB plots obtained from data saved during the controller execution.

Figure 4.4.4 Feedback gains
Figure 4.4.4 Chirp signal and load cart position when $K_2 = K_4 = 0$

Figure 4.4.6 Chirp signal and cart output when $K_2$ and $K_4$ have been designed using LQR

Figure 4.4.7 FFT of signals in Figure 4.4.5

Figure 4.4.8 FFT of signals in Figure 4.4.6
4.4 3 DOF Helicopter

The 3 DOF helicopter experiment shown in Figure 4.4.1 is a prototype of a new Quanser consulting experiment under development. The preliminary controller is shown in Figure 4.4.1 and the associated blocks are described below. The controlled variables are the Pitch, Travel and Elevation of the helicopter which are measured via encoders on the apparatus. The outputs of the controller are voltages applied to two motors, namely the Front motor and the Back motor, through D/A channels #0 & #1. The system is commanded via a joystick which generates two analog voltages measured via A/D channel #0 and A/D channel #1.

Figure 4.4.1 3DOF helicopter controller (file ws_heli3.m)

3D Heli - 1
**4.4.1 Joystick block** The joystick block measures two analog voltages and after filtering and calibration, generates an elevation command and a travel rate command. Note that the elevation command is generated by integrating the signal, while the Travel rate command does not integrate the signal. A deadband is applied to the signals in order to eliminate jitter.

![Diagram of Joystick block](image)

*Figure 4.4.2 Joystick block of helicopter Controller*

**4.4.3 Elevation, Travel and Pitch blocks**

These blocks measure the three controlled variables and low pass filter them as well as generate their derivatives using a limited differentiator (high pass filter). One of the blocks is shown expanded in Figure 4.4.3.

The encoder measurement is obtained from the MultiQ board, which is then multiplied by the calibration constant. The block also applies the measured value to a “scope” which will be accessible from WinCon.

![Diagram of Elevation block](image)

*Figure 4.4.3 Expanded Elevation block*
4.4.4 Controller blocks

These blocks generate the voltages that will be applied to the two motors. Each block is a controller using one of the measured variables and its command. Note that there is no command for the pitch axis. The controller tries to maintain the pitch horizontal. The travel rate command, causes disturbances in pitch, which in turn result in travel.

The voltages generated from the control blocks are then sent to summers which distribute the voltages to the two motors in the system. Note that this is a MIMO system but the controllers are designed as SISO systems.

Figure 4.4.7 shows the response of the Elevation axis to a commanded elevation derived from the joystick input.

Figure 4.4.8 shows the travel rate response to the commanded travel rate. The variation in pitch during this manoeuvre is shown in Figure 4.4.9.

In order to run the controller follow the steps given in the previous examples.

---

3D Heli - 3
Figure 4.4.7 Elevation response to an elevation command

Figure 4.4.8 Travel rate response to an travel rate command

Figure 4.4.9 Pitch reaction to travel rate command shown in Figure 4.4.8
5.0 Hints and Troubleshooting

5.1 Discrete and multirate blocks

Discrete time transfer functions have a Sample time parameter which must be entered when the block is defined. If the sample time is shorter than or equal to the Step size selected under Realtime Code Options, then the block is executed at every Step (size). If the Sample time is longer than the Step size, then you have defined a multirate system. The discrete block will then be evaluated at an integer multiple of the Step size closest to the Sample time you have defined when you generated the realtime code. This means that if you change the Sample time of a discrete block during a simulation, it will not take effect. The reason for this is that you have already generated code which fixes the Sample time for that block. Also note that the coefficients computed for a discrete transfer function, must be based on the Sample time you chose and not the Step size, [eg converting from continuous time to discrete use c2dm(nmu, den, Sample_Time) and not c2dm(nmu, den, Step_size)]. Otherwise, the coefficients will not be compatible with the actual implementation. Normally you would select the Sample time of a discrete block to be the same as the Step size you selected for Real-time options unless you want a multirate system.

5.2 Parameter changes and the message

The Simulink model has been structurally changed since the real-time code for WinCon was generated. Re-generate the real-time code.

If you change connections in the diagram or modify the number of blocks, you will need to re-compile the code. This is also true if you change the dimensions of a matrix or vector which is being used in a SIMULINK block. For example, if you are using a numerator for a transfer function num = [1.2 3.1] you decide to change it to num = [3.1] in the MATLAB Command Window, this is interpreted as a change in the block diagram. You may try to change it to num = [0 3.1] but you will still get the message that a change has occurred. This is because SIMULINK detects "0" as a no connection and a "1" as a connection. For this reason, if you want to change a value to zero or one without re-compiling, you should use a "very small value" for a zero. For example, changing the value of num to [0.000001 3.1] will result in practically the same filter but will not cause a re-compile message to arise. The same principle applies for a value of 1.

5.3 Changing a variable in the MATLAB Command Window and reflecting the change into the running SIMULINK controller

Suppose you are using a MATLAB variable named A in a SIMULINK block. A value must be assigned to the variable before you can generate realtime code. You assign the value of A from the MATLAB Command Window. Once you generate the Real-time code and you are running the controller, you can change the value of A in the MATLAB Command Window. The value does not change in...
SIMULINK/WinCon until you either

a) while the controller is running, open the dialogue box for the block in which the variable A is referenced
and close it.

b) Stop and Start the controller. The new variable value will be automatically updated in the
SIMULINK/WinCon controller without opening the dialogue box. The reason the value changed in
MATLAB is not automatically updated in SIMULINK is a SIMULINK specific characteristic.

5.4 Summing junction

Summing junction signs cannot be changed once the code has been generated. This is SIMULINK
specific characteristic. Do not open a Summing junction dialogue box while running code. This stops the
controller and switches Simulation Options to Normal. Always switch it back to External.

5.5 Controller does not start

If you click Start and the controller does not start, check the External switch under Simulation-Options,
also make sure you have configured the Realtime Code Options correctly as well as the External File. See
section 3.4.

5.6 Integration methods

The choice of integration method for continuous time blocks is made from the Code - Real-time Options
dialogue window. The choices under Simulation-parameters will not change integration methods of
Real-time Code. Once you compile with a specific integration method, you can only change it by
re-compiling. We suggest RK3 since it performs better than Euler but generates less code than RK5. If you
choose "None", then the outputs of continuous time blocks will be zero.

5.7 WinCon Configuration

WinCon should always be configured with zero Inputs/Outputs/References (Configure from WinCon Main
Window) while running SIMULINK controllers. This option is only for custom written controllers from
Quanser Consulting.

5.8 STOP Control/Simulation block

The STOP block from SIMULINK/Sinks does not work under external mode. If you want to stop the
controller use STOP WinCon. Obtain STOP WinCon by typing winlib in the MATLAB command Window
and selecting SOURCES.

5.9 Random Noise blocks

These blocks are very useful for parameter estimation experiments. The block from SIMULINK/Sources
does not work in external mode. Use the two choices from the SOURCES under winlib.

5.10 Other compilers

To date we support Watcom V10.0a or higher. For other compilers please contact Quanser Consulting.
We will be happy to assist you in creating the correct templates and makefiles for your compiler.

5.11 CRASH!

If the system returns with a Windows error message. Close everything down and reboot. It is a fact of life that crashes sometimes occur (the less often the better of course... but they do occur).

Who has not seen the cryptic message:

"An error occurred in your application if you choose to ignore ..."

It is a worthwhile habit to save all your work before re-compiling! WinCon has been rigorously tested and we hope that we have ironed out the bugs. If there is a consistent crash occurring please contact us with details. We will attempt to resolve the problem expeditiously. If the controller itself crashes, it is likely that you have selected a sampling period that is too small for the controller/computer combination you have. Start with a slow sampling period then move faster if necessary.

During realtime operation, please do not be hasty in changing windows or clicking the mouse buttons. If the hourglass symbol appears (system busy), wait until it is removed from the screen before you start another operation.

5.12 Maximum sampling frequency

Realtime controllers run in the background. Apart from obtaining realtime control, you want your computer to perform other tasks while the controller runs. How fast can the controller run such that you can still use another application in the foreground? There is no precise answer to this question. It will depend on the complexity of the controller and the speed of the processor you are using. Furthermore if your controller is too complex, you may not achieve the sampling speed you specified. If the operations to be performed (computation delay) take longer than one sample period, the next sample period will be ignored! Note that computational delay from one sample to the next cannot be assumed to be constant. It depends on the actual values being operated upon and if you have a nonlinear controller it depends on the statements being executed. So how can you verify that the controller is sampling at the rate you specify?

Here are a few methods:

a) The simplest method of determining if you are sampling too fast is to observe behaviour of the foreground task. For example, if the mouse cursor is not reacting as fast as it usually does, graphics is not updating quickly or the realtime display is skipping seconds, then the controller has taken over the processor and there is little time left in the foreground to perform other operations. This is usually a good indication that you should reduce the sampling rate.

b) A simple and relatively precise method would be to compare the realtime display in WinCon with a hand held digital chronometer. Start the controller and the chronometer at the same time. Compare the WinCon displayed time with the chronometer. If after ten minutes of operation, the controller and the chronometer are in synch (within 1 second due to delays in starting the controller and the discrepancy between your two hands), then the controller is running at the sample speed without skipping beats. If there is a discrepancy, then you are sampling too fast. The delay between the two times can be used to approximate how many beats were skipped and you can then estimate the average sampling period. As a rule of thumb, you should let 25% of processor time be available for your foreground tasks. For example, if
after 10 minutes of running a 1 Khz controller the WinCon clock was 30 seconds behind the chronometer then the controller skipped approximately 30,000 samples. That means you acquired only 570,000 samples in 10 minutes and the average sampling period was 1.053 mseconds. The average sampling frequency was 949Hz. You should limit your sampling frequency to .75*949 = 711 Hz (T_s = .0014 sec.) in order to allow enough time in the foreground to perform other tasks. Running the controller at this speed would also guarantee that no beats are missed.

c ) If you want an accurate measure, you can dedicate one of your D/A channels to switch its output between +/-1 volts at every sample. You can then examine the output of the D/A on the oscilloscope and see if you are attaining the speed you want. If you are sampling faster than the time it takes to execute the controller, the output will not be a square wave of the frequency you expect (half the sampling frequency since you switch signs at every sample).

If your system requires faster sampling rates than can be achieved on the PC, you may want to consider

1) a faster computer
2) Running in DOS
3) a dedicated controller (ie a DSP).

The natural question to ask is if one can predict the maximum sample rate for a given controller. This depends on the code that has been generated and the data acquisition board.

Our benchmark is that the following operations (inverted pendulum, example 4.3) can be performed at 3 kHz leaving little time (around 5%) for other tasks on a Pentium 90 MHz.

2 x A/D acquisitions
4 x 1st order continuous time filters
1 x signal generator
6 x scalar gain
7 x additions
3 x scopes
1 x realtime clock

One can then estimate that if you double the above complexity, then the fastest rate possible is around 1500 Hz if you use the same computer. Note that the data acquisition blocks typically take longer to execute than the other blocks. So if you maintain the same I/O channels and add some other blocks to the above, you may still be able to achieve around 3 KHz... it all depends on complexity.
Predicting the maximum sample rate accurately is usually not possible. The safest method to obtain this measure for a given controller is to try the controller without connecting the plant. You then keep increasing sampling frequency until there is considerable slowing of the tasks in the foreground. If the computer hangs while performing this test, you should reboot the system. The reason the computer would hang is that the sample rate was so high that there was no time left to interact with the user!

5.13 FUZZY Controllers

The file c:\matlab\toolbox\fuzzy\fuzzy\ffuis.c does not work with the realtime workshop. You will need to download an update for this file from the Mathworks website at: http://www.mathworks.com.

Fuzzy controllers will require much more computational power than a state feedback controller. Sampling periods must be selected carefully. Our experience indicates that a 2 input, 1 output FUZZY controller with 5 membership functions at each input, 9 rules and 5 membership functions at the output can only run at 200 Hz. Sample rates faster than that will not function properly.

5.14 Disk Space Make sure there are at least 10MBytes of free hard disk space on your system.

5.15 Networked systems WinCon is a realtime system and network operations will interfere with the performance. WinCon is designed for a standalone system and Quanser Consulting does not guarantee performance on networked systems.

5.16 Versions!

Much too often, all software changes versions (that is a drag). Type ver in the Matlab command window. You will get a display as shown below. These are the versions we have been using. If you are using earlier versions there is no guarantee things will work. Later versions should normally cause no problems.

```matlab
ver
MATLAB Version 4.2c.1
MATLAB Site Identification Number: ******
No Contents.m file for c:\winsim
No Contents.m file for c:\matlab\codegen\rt\wincon\devices
No Contents.m file for c:\matlab\codegen\rt\dos\devices
No Contents.m file for c:\matlab\codegen\rt\dos\quanser
MATLAB Toolbox Version 4.2a 25-Jul-94
SIMULINK model analysis and construction functions. Version 1.3c 15-August-94
SIMULINK demonstrations and samples. Version 1.3c 15-August-94
SIMULINK block library. Version 1.3c 15-August-94
SystemBuild 3.0 model import into SIMULINK. Version 1.0a 09-Mar-94
Real-Time Workshop Version 1.1c 25-May-95
Control System Toolbox. Version 3.0b 3-Mar-93
Nonlinear Control Design Toolbox. Version 1.0 5-Nov-93
Signal Processing Toolbox. Version 3.0b 10-Jan-94
User Interface Utilities. Version 1.3 19-Oct-95
System Identification Toolbox. Version 4.0 30-May-95
Fuzzy Logic Toolbox. Version 1.0, 1-19-95
Fuzzy Logic Toolbox Demos. Version 1.0, 1-19-95
```
B.2 MultiQ board manual

Instruction manual, 21 pages

MULTIQ™
8 Analog to Digital Converters
8 Digital to Analog Converters
6 Quadrature input decoders/counters
8 Digital inputs
8 Digital outputs
3 Realtime clocks

PROGRAMMING MANUAL

Quanser Consulting
CAUTION

The flat ribbon cable connectors are keyed and must be inserted into the MULTIQ board and the TERMINAL board with the correct orientation. Furthermore, the short cable should be used for the Analog signals and the long cable should be used for the digital signals and encoders.

CAUTION

![Diagram]

- Cable #2 (long)
- Cable #1 (short)

- MultiQ board
- Terminal board

Cable #1: Analog signals
Cable #2: Digital signals and encoders
No twists in the cables!

The board is static electricity sensitive. Be very careful with electrical discharge. Always touch a ground plane BEFORE you handle the board.

CAUTION

Some of the wiring you perform to the terminal board carries POWER. Be sure your wiring is correct as applying power incorrectly may either damage the device you are connecting to, the board or your computer!

NEED HELP?
CALL US AT (905) 527 5208 or FAX your question to (905) 570 1906
OR EMAIL TO: QUANSER@NETACCESS.ON.CA

- 71 -
1.0 General description

The MULTIQ is a general purpose data acquisition and control board which has 8 single ended analog inputs, 8 analog outputs, 8 bits of digital input, 8 bits of digital output, 3 programmable timers and up to 6 encoder inputs decoded in quadrature. Interrupts can be generated by either of the three clocks, one digital input line and the end of conversion from the A/D.

The system is accessed through the PC bus and is addressable via 16 consecutive memory mapped locations which are selected through a DIP switch located on the board.

2.0 Principles of operation

2.0.1 Terminal board

Turn the PC off and insert the MultiQ into a bus slot in your PC. Connect the cables as shown on the front page of this manual. The terminal board supplied with the board is shown in Figure 2.

![Figure 1 Block diagram of MULTIQ](image)

![Figure 2 Terminal board](image)
All wiring to the board is performed through the terminal board. Analog inputs and outputs are connected via RCA connectors. Digital I/O is via 16 pin headers and Encoder inputs are through 5 Pin Din (Stereo) connectors or via 10 pin headers.

2.1 Analog to digital conversion

The A/D of the MULTIQ is a single ended bipolar signed 13 bit binary (12 bit plus sign) A/D. You can perform a conversion on one of 8 channels by selecting the channel and starting a conversion. the EOC_I (end of conversion interrupt) bit in the STATUS REGISTER indicates that the data is ready and can be read. The data is read by issuing 2 consecutive 8 bit reads from the AD_DATA register.

The data returned is two 8 bit words which must be combined to result in a 16 bit signed word. 5 volts input maps to 0xFFFF while 0 volts maps to 0x0 and -5 Volts maps to 0x0000.

2.1.1 Wiring to the A/D

All inputs to the A/D multiplexer are single ended in the range +/- 5 Volts and should be wired via the RCA connectors on the terminal board labelled Analog Inputs.

2.1.2 Signal conditioning of the Analog Input signals

If you wish to low pass filter the input signal before data acquisition you may do so by populating the area labelled Signal Conditioning shown in Figure 2. The circuit for each input channel is shown in Figure 3. It is highly recommended that you attach a capacitor at least to the input of each A/D. This is simply done by soldering a capacitor into the appropriate holes. The factory configuration is Ra = short, Rb = open and C = open. The choice of the component values depends on the output impedance of the sensor and the sampling frequency of your data acquisition program. Typically you select a cutoff frequency to be less than half the sampling frequency (or even smaller).

![Signal conditioning circuit for Analog input signals](image)

**Figure 3** Signal conditioning circuit for Analog input signals
2.2 Analog output

The digital to analog (D/A) converters are 12 bit unsigned binary. An input of -5 volts maps to 0x000, 0 volts to 0x3FF and 5 volts maps to 0xFFF. Your program should write a 12 bit number (0 to 4095) to the appropriate register and should latch the data. The analog outputs change when the data is latched.

2.3 Encoder Inputs

The board can be equipped with up to six encoder decoders. (Models -2E, 4E and 6E). The encoders data is decoded in quadrature and used to increment or decrement a 24 bit counter. With 24 bits, you can obtain 16,777,215 counts. With a 2000 line encoder in quadrature, this results in 8000 counts per revolution and 2097 revolutions can be measured without overflowing the counters. Higher counts can be handled by software.

2.3.1 Wiring to the Encoder inputs.

Each encoder input is equipped with one 5 pin DIN socket and one 10 pin header on the terminal board. You can use either of these to connect an encoder to the board. The connectors supply +5V and GND to bias the encoders and receives an 'A' channel and a 'B' channel from the encoder. You must use 5 Volt output encoders only. Figure 4 shows the pin definition for the 5 pin DIN connector as well as the 10 pin header of the encoders inputs and the method of wiring.

![Figure 4 Encoder connections to the 5 pin DIN connector or the 10 pin header.](image)

2.4 Digital inputs

The board can read 8 digital input lines mapped to one I/O address. The digital input is normally high ('1') and results in a low ('0') when the line is pulled to GND. Digital input line #0 can be tied to an interrupt using the jumpers supplied.

2.5 Digital outputs

The board can control 8 individual digital outputs mapped to one I/O address. Writing a '0' to the appropriate bit results in zero volts (TTL LOW) at the output while writing a '1' results in 5 Volts (TTL HIGH).

2.6 Realtime clocks

The board is equipped with three independent programmable clock timers. Each timer can be programmed to run at a frequency between 2 MHz Hz and 30.52 Hz. The principle of operation is to write a divisor (N) to the desired clock and the output frequency will be 2.0/N MHz. (N) is a 16 bit integer value between 2 and 65535 (0xFFFF). The output of any of the three clock can be tied to an interrupt line using a jumper on the board. The outputs are available on the terminal board for monitoring or triggering external devices.
3.0 PROGRAMMING

3.1 Base address selection

The base address is selected by using the dip switches (SW2) on the board. Factory configuration is 0x320. make sure there are no other devices on that address and up to Base+0xF (0x32F factory configuration).

<table>
<thead>
<tr>
<th>Base Address</th>
</tr>
</thead>
<tbody>
<tr>
<td>B</td>
</tr>
<tr>
<td>0</td>
</tr>
</tbody>
</table>

3.1.1 Factory configuration

<table>
<thead>
<tr>
<th>Base Address</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
</tr>
<tr>
<td>0</td>
</tr>
</tbody>
</table>

3.2 Board Registers The table below shows the registers on the MULTIQ. Each register is described in its appropriate section.

<table>
<thead>
<tr>
<th>Base +</th>
<th>Read</th>
<th>Write</th>
<th>Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>DIGIN_PORT</td>
<td>DIGOUT_PORT</td>
<td>8 Bit</td>
</tr>
<tr>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td>DAC_DATA</td>
<td>16bit</td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>AD_DATA</td>
<td>AD_CS</td>
<td>16 bit write 8 bit read</td>
</tr>
<tr>
<td>5</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>STATUS</td>
<td>CONTROL</td>
<td>16 bit</td>
</tr>
<tr>
<td>7</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>N/A</td>
<td>CLK_DATA</td>
<td>8 bit</td>
</tr>
<tr>
<td>9</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>A</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>B</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>C</td>
<td>ENC_1</td>
<td>ENC_1</td>
<td>16 bit</td>
</tr>
<tr>
<td>D</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>E</td>
<td>ENC_2</td>
<td>ENC_2</td>
<td>16 bit</td>
</tr>
<tr>
<td>F</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

3.2.1 Digital port : Base + 0 (DIGIN_PORT & DIGOUT_PORT)
3.2.1.1 Write (DIGOUT_PORT)

This is the Digital output port. A 16 bit write to this port outputs the 16 bit data to the digital output header on the terminal board. Eg. writing a 0x00e5 (binary 11100101) results in bits 0,2,5,6,7 to go high.

<table>
<thead>
<tr>
<th>DIGIN_PORT Write</th>
</tr>
</thead>
<tbody>
<tr>
<td>15</td>
</tr>
<tr>
<td>N/A</td>
</tr>
</tbody>
</table>

3.2.1.2 Read (DIGIN_PORT)

This is the Digital input port. A 16 bit read from this board returns the digital levels at the header labelled Digital input on the terminal board. The inputs are tied high and a read with nothing connected to the header results in reading a 0xFFFF. A returned value of 0xFFa6 (last 8 bits 10100110) means that bits 0,3,4 and 6 have been pulled low by an external device (for example a switch). Note that the high word is always ones.

<table>
<thead>
<tr>
<th>DIGIN_PORT Read</th>
</tr>
</thead>
<tbody>
<tr>
<td>15</td>
</tr>
<tr>
<td>N/A</td>
</tr>
</tbody>
</table>

3.2.2 D/A data Port : Base + 2 (DAC_DATA)

3.2.2.1 Write (DAC_DATA)

A write to this port sets up the Data for the D/A output. The data should be written to bits (DO11 to DO0). A value of 0 puts out -5 Volts, a value of 0xFF puts out 0 Volts and 0xFFFF puts out +5 Volts. The output channel is selected by writing to bits (DA2 DA1 DA0) of the CONTROL REGISTER and the output is latched when a (11) is written to bits (LD0 LD1) of the CONTROL REGISTER.

<table>
<thead>
<tr>
<th>DAC_DATA Write</th>
</tr>
</thead>
<tbody>
<tr>
<td>15</td>
</tr>
<tr>
<td>NA</td>
</tr>
</tbody>
</table>

3.2.2.2 Read (NOT APPLICABLE)
3.2.3 A/D Register: Base + 4 (AD_CS and AD_DATA)

3.2.3.1 Write (AD_CS)

A write to this register (any data) initiates a conversion after the A/D has been properly set up using the CONTROL REGISTER. The program must wait for (EOC) to go high in the STATUS REGISTER before initiating a conversion.

3.2.3.1 Read (AD_DATA)

This is an 8 bit read register and contains the high byte on the first read and the low byte on the second read. The structure of the two 8 bit reads is shown below:

<table>
<thead>
<tr>
<th>AD_DATA Read (First time)</th>
</tr>
</thead>
<tbody>
<tr>
<td>7</td>
</tr>
<tr>
<td>SIGN</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>AD_DATA Read (Second time)</th>
</tr>
</thead>
<tbody>
<tr>
<td>7</td>
</tr>
<tr>
<td>AI</td>
</tr>
</tbody>
</table>

To convert the data to a voltage the two bytes should be combined into a 16 bit word as follows:

integer_data = (high_byte<<8)|(low_byte&0xFF);
volts = integer_data * 5.0/4096;

where (<<) is the shift left operator, (l) is bitwise 'OR' and (&) is bitwise 'AND'. This means mask off the left 8 bits of the low byte just in case there is extraneous noise, and then merge it with the high byte shifted left 8 bits. This results in a number between (-4096) and (4095). These are mapped to -5V and +5V.

3.2.4 CONTROL REGISTER: Base + 6 (STATUS & CONTROL)

3.2.4.1 Write (CONTROL)

<table>
<thead>
<tr>
<th>CONTROL REGISTER Write</th>
</tr>
</thead>
<tbody>
<tr>
<td>X</td>
</tr>
<tr>
<td>15</td>
</tr>
</tbody>
</table>

Bits 0-2 (DA0 DA1 DA2): Select the D/A channel number. eg writing a 0x03 selects channel D/A ch3
Bits 0-1 (RC0 RC1): Select the realtime clock register
Bits 3-5 (A0 A1 A2): Select the analog input channel you want to Multiplex to the A/D
Bit 6: (MX) Enable the 8 channel multiplexer
Bit 7: (AZ) Enable Auto Zero on the A/D
Bit 8: (CAL) Enable Autocalibration on the A/D
Bit 9: (S/H) Disable Sample and Hold on the A/D Keep this bit high all the time
Bit 10 (CLK): Select base clock frequency for the A/D. 1 is 4 MHz, 0 is 2 MHz. (Always use 4 Mhz). Keep this bit high all the time
Bit 11-12(LD0,LD1): Latch data to the selected D/A channel when both bits are set high.
3.2.4.2 Read (STATUS)

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th>EOC_I</th>
<th>EOC</th>
<th>CT1</th>
<th>CT2</th>
<th>CT0</th>
</tr>
</thead>
<tbody>
<tr>
<td>15</td>
<td>14</td>
<td>13</td>
<td>12</td>
<td>11</td>
<td>10</td>
<td>9</td>
<td>8</td>
<td>7</td>
<td>6</td>
<td>5</td>
<td>4</td>
<td>3</td>
</tr>
</tbody>
</table>

Bits 0-2 (CT0, CT1, CT2): Counter timeout states for the three timers.
Bit 3: (EOC) End of conversions on the A/D. Goes high when A/D is ready for another conversion.
Bit 4: (EOC_I) End of conversion Interrupt. Goes high when A/D conversion is complete.

3.2.5 Clock data register: base + 8 CLK_DATA

3.2.5.1 Write (CLK_DATA)

This a 8 bit write only register that accesses any of the four registers on the realtime clock chip. The register (CLK_0 to CLK_4, see section on programming) is selected by writing to bits (RC1 RC0) of the CONTROL REGISTER and then writing the data to CLK_DATA.

<p>| | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>7</td>
<td>6</td>
<td>5</td>
<td>4</td>
<td>3</td>
<td>2</td>
</tr>
</tbody>
</table>

3.2.5.2 Read (NOT APPLICABLE)
3.2.6 ENCODER REGISTERS Base + 0xC and Base + 0xD (ENC1 & ENC_2)

3.2.6.1 Write (ENC_1)

Writing to this register selects which of the 6 encoder counter values you want to read next. The encoder number is specified in bits (E2 E1 E0).

<table>
<thead>
<tr>
<th>ENC_1 Write</th>
</tr>
</thead>
<tbody>
<tr>
<td>15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0</td>
</tr>
<tr>
<td>NA NA NA NA NA NA NA NA NA NA NA EN EN EN EN</td>
</tr>
</tbody>
</table>

Writing to bits (R1 R0) will reset the encoders to a zero count. To reset an even numbered encoder (0,2,4) write the encoder number into bits (E2 E1 E0) and write a ‘1’ to bit R0. To reset an odd numbered encoder (1,3,5) write the encoder number to bits (E2 E1 E0) and write a ‘1’ to bit R1.

Writing to ENC_2 is not applicable

3.2.6.1 Read (ENC_1 and ENC_2)

A 16 bit read from ENC_1 returns the low 16 bits of the 24 bit counter selected by the write described above. A 16 bit read from ENC_2 returns the high eight bits of the 24 bit counter located in the low eight bits of the data.

<table>
<thead>
<tr>
<th>ENC_1 Read</th>
</tr>
</thead>
<tbody>
<tr>
<td>15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0</td>
</tr>
<tr>
<td>EN EN EN EN EN EN EN EN EN EN EN EN EN EN</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>ENC_2 Read</th>
</tr>
</thead>
<tbody>
<tr>
<td>15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0</td>
</tr>
<tr>
<td>NA NA NA NA NA NA NA NA NA NA NA EN EN EN EN</td>
</tr>
</tbody>
</table>
4.0 SAMPLE PROGRAMS

The following are sample programs written in Turbo C and can be used by your main program. These functions are included in the file `multiqdrv`. The file also contains the definitions of the register locations based on a base address 0x320 as shown below:

```c
#define base_port 0x320
#define digin_port base_port + 0x00
#define digout_port base_port + 0x00
#define dac_cs base_port + 0x02
#define ad_cs base_port + 0x04
#define status_reg base_port + 0x06
#define control_reg base_port + 0x06
#define clk_reg base_port + 0x08
#define enc_reg1 base_port + 0x0c
#define enc_reg2 base_port + 0x0e
#define AD_SH 0x200 /* active low */
#define AD_AUTOCL 0x100 /* active high */
#define AD_AUTOZ 0x80 /* active high */
#define AD_MUX_EN 0x40 /* active high */
#define AD_CLOCK_4M 0x400 /* high = 4 MHz */

/* IMPORTANT */
/* sample and hold disabled to prevent extraneous sampling */
/* and fix the clock speed to 4 MHz */
#define CONTROL_MUST (AD_SH | AD_CLOCK_4M)

unsigned int control_word = CONTROL_MUST;
```

4.1 D/A operation

- Write the analog output channel number to bits (DA2 DA1 DA0) of CONTROL along with a (11) to LD1 and LD0 of the CONTROL REGISTER
- Write the actual data to DAC_DATA (16 bit write lowest 12 bits carry the data).
- release the latch by writing a (00) to both bits (LD1 LD0) of the CONTROL REGISTER.

4.1.1 Sample C function

```c
void daout(int ch, int ivalue)
{
    outport(control_reg, 0x1800 | ch | CONTROL_MUST);
    outport(dac_es,ivalue);
    outport(control_reg, CONTROL_MUST);
}
```

4.1.2 Reset the D/A outputs

```c
void reset_da(void)
{
    int zero_v;
```
zero_v = vtoi(0.0); /* see this function below */
daout(0,zero_v);
daout(1,zero_v);
daout(2,zero_v);
daout(3,zero_v);
daout(4,zero_v);
daout(5,zero_v);
daout(6,zero_v);
daout(7,zero_v);
}

4.1.3 Voltage to integer conversion

This function is used to convert a desired voltage (in volts) to the appropriate integer value for the D/A.

```c
int vtoi(float v)
{
    return(ceil( v*2048/5.0 +2047));
}
```

4.2 A/D OPERATIONS

4.2.1 Calibrating the A/D

This can be performed once only at the start of a program. Once calibrated, the offset and gain are used for all subsequent measurements. To calibrate:

1) write a '1' to bit 'CAL' and to 'S/H' of CONTROL REGISTER
2) write a '1' to 'S/H' of CONTROL REGISTER
3) wait for EOC to go high in STATUS REGISTER

4.2.1.1 Sample C function

```c
void reset_ad(void)
{
    /* start calibration */
    outport(control_reg, AD_AUTOCAL | CONTROL_MUST);
    outport(control_reg, CONTROL_MUST);
    while((inport(status_reg)&0x08) == 0x00);
}
```

4.2.2 Acquiring a sample

1) select the channel and write it to control register bits (A2 A1 A0) along with a ‘1’ to (EN) , a ‘1’ to (S/H) and a ‘1’ to (CLK) bits of the CONTROL REGISTER. Also write a ‘1’ to bit (AZ) if you want auto zero before the sample. Note that autozero takes longer and is not normally necessary.
2) wait until (EOC) in STATUS REGISTER goes high.
3) Initiate a conversion by a write to AD_CS (any value)
4) wait until EOC_1 in STATUS REGISTER goes high
5) read high byte from AD_DATA
6) read low_byte from AD_DATA
4.2.2.1 Sample C Function

```c
int adin(int ch)
{
    unsigned int hb, lb;
    int toolong, maxcnt;
    maxcnt = 30;
    nosound();
    control_word = CONTROL_MUST | AD_MUX_EN | (ch<<3); /* select channel and enable mux start S/H*/

    /* use the next line instead of above line if you want to auto zero before every sample */
    /*control_word = CONTROL_MUST | AD_AUTOZ | AD_MUX_EN | (ch<<3);*/
    /* NOTE THAT IT IS SLOWER WITH AUTO ZERO */

    outport(control_reg, control_word);
    toolong = 0;
    while( (inport(status_reg)&0x8) == 0x00 ) && (toolong < maxcnt) toolong++;
    if(toolong>=maxcnt) sound(400);
    outportb(ad_cs, 0);
    while( (inport(status_reg)&0x10) == 0x00 );
    hb = inport(ad_cs) & 0xff;
    lb = inport(ad_cs) & 0xff;
    outport(control_reg, CONTROL_MUST);
    return ( (hb<<8) | lb );
}
```

Note the limited wait loop:

```c
toolong = 0;
while( (inport(status_reg)&0x8) == 0x00 ) && (toolong < maxcnt) toolong++;
if(toolong>=maxcnt) sound(400);
```

which ensures that the wait for EOC is left if it takes too long. If this happens, an error has occurred on the A/D Chip (National Semiconductor ADC1251) during a conversion and the chip is not ready for a conversion after sufficient waiting. This is not usually necessary but is good practice. If this error occurs, the computer will generate a sound until issuing 'nosound()' from the calling C program. If you do not want the sound to go on just delete the 'sound(400)' statement.

4.2.3 Integer to voltage conversion

The following function converts from an integer value read by the A/D to a floating point value in volts.

```c
float itov(int iv)
{
    return(iv*5/4095.);
}
```

4.3 Digital input operation:

- Read a 16 bit word from DIGIN_PORT

4.3.1 Sample C Function

```c
int digin(void)
```
4.4 Digital output operation

- Write a 16 bit word to DIGOUT_PORT

4.4.1 Sample C Function

```c
void digout(int dig_value)
{
  outport(digout_port,dig_value);
}
```

4.5 Encoder operations

4.5.1 Encoder reset

- write to ENC1 the encoder channel you want to reset.
- write to bit R0 of ENC1 a '1' if the channel number is even (along with the channel number)
- write to bit R1 of ENC1 a '1' if the channel number is odd (along with the channel number)

4.5.1.1 Sample C function

```c
void enc_reset(int ch)
{
  outport(enc_reg1,ch);
  if( (ch==0) || (ch==2) || (ch==4) ) outport(enc_reg1,((ch&0x07)<<0x8));
  if( (ch==1) || (ch==3) || (ch==5) ) outport(enc_reg1,((ch&0x07)<<0x10));
}
```

4.5.2 Encoder read

- write to ENC.1 the channel number you want to read into the lower 3 bits(E2 E1 E0)
- read from ENC.1 the low byte (16 bits)
- read from ENC.2 the high byte (16 bit read, mask off the 8 most significant)
- merge the two data to obtain a 'long signed int'

4.5.2.1 Sample C function

```c
long int enc_in(int ch)
{
  unsigned int low_word, high_word;
  unsigned long result;

  outport(enc_reg1,ch);
  low_word = inport(enc_reg1);
  high_word = inport(enc_reg2);
  result = high_word & 0xff;
  if(result & 0x80) result = result | 0xff00; /*convert to signed 32 bit*/
  result = (result << 16) | low_word;
  return ((long) result);
}
```
4.6 Clock operations

The three clocks are imbedded in a single integrated circuit (INTEL 82C54). Four registers located on the clock chip are addressed via bits (RC1 RC0) of the MULTIQ's CONTROL REGISTER. In order to write a desired value to a specific clock register, first write to (RC1 RC0) of the CONTROL REGISTER the value for the register you want to access and then write the desired value to the CLOCK DATA REGISTER (Base + 8).

<table>
<thead>
<tr>
<th>RC1</th>
<th>RC0</th>
<th>Clock IC register</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>CLOCK 0 DATA REGISTER</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>CLOCK 1 DATA REGISTER</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>CLOCK 2 DATA REGISTER</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>CLOCK COMMAND REGISTER</td>
</tr>
</tbody>
</table>

The CLOCK COMMAND REGISTER has the following bits:

<table>
<thead>
<tr>
<th>CLOCK COMMAND REGISTER</th>
</tr>
</thead>
<tbody>
<tr>
<td>7 6 5 4 3 2 1 0</td>
</tr>
<tr>
<td>SC1 SC0 RW1 RW2 M2 M1 M0 BC</td>
</tr>
</tbody>
</table>

The clock you want to perform an operation on is selected using bits (SC1 SC0) and the mode of operation is selected using bits (M2 M1 M0).

In order to program a specific clock to run at a given frequency, you must first select a divider for the clock. For example, if you want to run at a frequency F, the divisor is obtained by using the equation:

\[ \text{DIV} = \text{CEIL}(2e6 / \text{Freq}) \]

where 2e6 Hz is the base frequency for all three clocks. This will be a 16 bit integer value (0 to 65535). Note that if Freq is smaller than (2e6/65535) the clock will actually run much faster than you expect!

Next you need to write to the CLOCK COMMAND REGISTER the clock number into (SC1 SC0) and select mode2 into (M2 M1 M0) (Baud rate Generator). You do this by first writing to the CONTROL REGISTER bits (RC1 RC0) a (1 1) indicating you will be writing to the CLOCK COMMAND REGISTER next and then you write the desired data.

After you select the clock number and the mode of operation into the CLOCK CONTROL REGISTER, write to the CLOCK DATA REGISTER (bits RC1 RC0 in the CONTROL REGISTER) the low byte and then the high byte of the divisor DIV. At this point, the clock you selected will start running at the frequency you specified.

4.6.1 Sample C functions

```c
void clockdiv(int clk_num, int div_value) {
    unsigned int lb, hb;
    lb = div_value & 0xff;
    hb = (div_value & 0xff00) >> 8;
```
control_word = 3 | CONTROL_MUST;
outport(control_reg,control_word); /* select register 3 of RTC */
outportb(clk_reg,((clk_num<<6)|0x34));
control_word = clk_num | CONTROL_MUST;
outport(control_reg,control_word);
outportb(clk_reg,lb);
outportb(clk_reg,lb);
}

/* this function sets the clock frequency of the desired clock. It calls clockdiv() */

void set_clk_freq(int clk_num, float clk_freq)
{
  float base_freq = 2000000;
  int divider;
  divider = ceil(base_freq/clk_freq);
  clockdiv(clk_num, divider);
}

The states of the three clock can be monitored through bits (CT2 CT1 and CT0) of the STATUS REGISTER.

5.0 Tying to interrupts

You may wire the following lines to some of the interrupt lines on the PC bus.

<table>
<thead>
<tr>
<th>Bit</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>CT0</td>
<td>Clock Timer 0 overflow</td>
</tr>
<tr>
<td>CT1</td>
<td>Clock Timer 1 overflow</td>
</tr>
<tr>
<td>CT2</td>
<td>Clock timer 3 overflow</td>
</tr>
<tr>
<td>EOC_I</td>
<td>End of conversion interrupt</td>
</tr>
<tr>
<td>DI0</td>
<td>Digital input bit 0</td>
</tr>
</tbody>
</table>
The interrupts lines that can be tied to are

<table>
<thead>
<tr>
<th>Interrupt#</th>
<th>Normally used by (on standard PC)</th>
<th>Vector address</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>COM1: Serial port</td>
<td>0x8</td>
</tr>
<tr>
<td>5</td>
<td>PC Fixed disk controller</td>
<td>0xD</td>
</tr>
<tr>
<td>7</td>
<td>LPT1: Printer</td>
<td>0xF</td>
</tr>
<tr>
<td>9</td>
<td>RESERVED</td>
<td>RESERVED</td>
</tr>
<tr>
<td>10</td>
<td>UNUSED</td>
<td>0x72</td>
</tr>
<tr>
<td>11</td>
<td>UNUSED</td>
<td>0x73</td>
</tr>
<tr>
<td>12</td>
<td>UNUSED</td>
<td>0x74</td>
</tr>
<tr>
<td>15</td>
<td>UNUSED</td>
<td>0x77</td>
</tr>
</tbody>
</table>

The interrupt lines should be physically connected on the board using the jumpers provided. You should be certain that no other device is tied to the interrupt line you want to use.

5.1 Writing interrupt service routines

The following is a short explanation on how to write interrupt service routines on an IBM PC compatible system. More detailed information can be obtained from any good book on PC architecture and the C compiler you are using.

An interrupt service routine (ISR) is initiated every time a specified interrupt line associated with the ISR goes high. The interrupt mask register on the PC is located at address 0x21. It is an eight bit port and in order to activate a certain interrupt line, you must write a 0 to the associated bit in the interrupt mask register. For example if you want to allow interrupts only from lines 2 and 7 you must write a (01111011) or (0x7b) to memory location 0x21.

Each interrupt lines causes a jump to the address located in the vector table shown above. For example if you want a function:

```
extern void interrupt far newtimer(void);
```

to be executed every time interrupt #5 occurs, you set it up in the following manner:

```c
disable(); /* disable interrupts */
cidtimer = getvect(0xd); /* save old isr address, see vector in column 3 of table above */
setvect(0xd,newtimer); /* setup the new isr */
int_mask = inportb(0x21); /* get the present mask */
outportb(0x21,int_mask&0xdf); /* write out a new mask that sets bit 5 to 0 */
enable(); /* enable interrupts, at this point newtimer is active and will be executed every time line 3 goes high */
```
Now what happens inside the isr is also very important. The ISR should have the following structure:

```c
extern void interrupt far newtimer(void)
{
    asm fsave data87 /* assembly code to save floating point processor status */
    _clear87(); /* clear the floating point processor */
    disable(); /* disable other interrupts */

    /* here is where you write your code */
    /* this will be executed every time the interrupt occurs */

    asm frestore data87 /* restore floating point processor status */
    outportb(Ox20, Ox65); /* acknowledges interrupt to 8259 */
    enable();
}
```
data87 must be declared globally as

```c
char *data87[94];
```

### 6.0 Testing your board

The programs 'test mq.c', 'test_int.c' and the driver file 'multiq.drv' are supplied with the board. In order to compile these programs you need Turbo C and Turbo assembler. Executable versions are also supplied.

- make a directory MULTIQ on your hard drive
- copy all files to that directory
- Run the program **test mq**

- The program is interactive and you can plot selected data in realtime. You can select the parameters associated with the letters in brackets [ ]. For example entering the letter [v] you will be prompted to enter a voltage which will be output to D/A channel specified by hitting [o]. The program is interrupt driven and all variables are monitored on the screen in realtime. The A/D channel associated with the letter [i] can be plotted in realtime as well as the encoder input associated with the letter [e]. Entering [z] will reset the encoder selected. You can output a digital word to the digital output port by entering [d] followed by a hexadecimal number. You can select the data you want to plot in realtime by hitting [v]. Hit 'x' to exit realtime plotting. Selecting [t] allows you to alter the duration of the x axis of the realtime plot.

[i]: A/D input channel number  
[o]: D/A output channel number  
[v]: output volts  
[d]: Digital word out  
[e]: Encoder channel  
[z]: Encoder reset  
[F]: Sampling Frequency(Hz)  
[T]: X axis time duration(Sec)  
[V]: Y axis variable:  
[R]: Realtime plot  
[Q]: Quit program

The program **test mq** uses the IBM PC SOFTWARE INTERRUPT and changes the speed of the realtime clock used for time of day. Run this program to test all the functions on the board except the realtime clocks.
Reset the time of day from DOS if necessary.

The program `test_int` uses the MULTIQ CLOCK #1 for hardware interrupts on interrupt line #5. In order to run this program you should install the jumper labelled CTC1! to pin interrupt 5. (FACTORY CONFIGURATION)

6.1 Compiling the source code

If you need to change anything in the source code of the above programs, you will need to recompile them. Use the following lines to obtain new executable code.

**Turbo C users**

```c

tcc -r -B -f87 -ml %1 graphics.lib
```

**Borland C users**

```c

tcc -r -B -f87 -ml %1 graphics.lib
```

where %1 is the name of the program.

Note Turbo assembler (TASM) should be in the PATH as well as (TC\BIN) or (BC\BIN). The file EGAVGA.BGI must be present in the directory.

6.2 Execution speed

The program `speed.c` tests the speed of the various operations on the board using the drivers listed above. The speeds are obtained on a 66MHz PC486 and may vary with the processor you use. The speeds do not solely depend on the board but also on execution time of the instructions in the drivers. For example, a single A/D conversion is performed in only 8 μseconds but it really takes 19.2 μseconds to actually acquire the data into the program. The extra time is due to bus access (input, output wait loops).

The results from `speed.c` are tabulated below. These are calculated by obtaining an average over one million consecutive operations using the driver functions given above.

<table>
<thead>
<tr>
<th>Function</th>
<th>Execution speed in μ seconds</th>
</tr>
</thead>
<tbody>
<tr>
<td>Digital input (16 bit)</td>
<td>2</td>
</tr>
<tr>
<td>Digital output (16 bit)</td>
<td>2</td>
</tr>
<tr>
<td>Encoder read (24 bit)</td>
<td>5.5</td>
</tr>
<tr>
<td>Analog to digital conversion (13 bit)</td>
<td>19.2</td>
</tr>
<tr>
<td>Digital to analog conversion (12 bit)</td>
<td>5.0</td>
</tr>
</tbody>
</table>

The above information is important when determining what is the maximum sampling frequency you can set in an ISR. Suppose you would like to sample all 8 analog input channels, output to all 8 channels, read 6 encoders and perform 16 bit digital I/O in a single interrupt service routine. These would take approximately (2+2+5.5x6+19.2x8+5x8) = 230.6 microseconds. Therefore the ISR should be called at a frequency slower than 3.8 KHz. If you want to perform calculations in an ISR (which you typically would for a controller), then the time for calculations should also be taken into consideration. Assuming the calculations you are performing take another 230 microseconds, then the ISR should execute slower than 1.9 KHz. You would
also like to have some time left over for foreground jobs. Let's say 50% of processor time left for foreground operations (plotting, user interaction, etc), then the ISR should be set to a maximum of 950 Hz. This is the suggested maximum sampling frequency when the board is being used at full capacity. Of course, the less channels are used and the simpler the controller, the faster you can set the speed of the ISR.

7.0 Selecting interrupt sources

Select the interrupt source using the jumpers at the headers labelled Interrupts. USE ONE INTERRUPT JUMPER PER HEADER.

1) Placing a jumper at the header labelled A/D will cause an interrupt from EOC_I to occur at the line to which the jumper is attached. ie if the jumper is attached between A/D and the pin labelled '5', an EOC_I will cause an interrupt number 5 to occur.

2) Placing a jumper at the header labelled CTC0 will cause an interrupt from CLOCK 0 to occur at the line to which the jumper is attached. ie if the jumper is attached between CTC0 and the pin labelled '5', an interrupt number 5 will occur at the frequency at which CLOCK 0 is operating.

3) Placing a jumper at the header labelled CTC1 will cause an interrupt from CLOCK 1 to occur at the line to which the jumper is attached. ie if the jumper is attached between CTC1 and the pin labelled '5', an interrupt number 5 will occur at the frequency at which CLOCK 1 is operating. THIS IS FACTORY CONFIGURATION.

4) Placing a jumper at the header labelled CTC2 will cause an interrupt from CLOCK 2 OR FROM Digital Input #0 to occur at the line to which the jumper is attached. The source depends on the jumper labelled CTC2INT. If the jumper is between pins (12) then the source is clock 2, if the jumper is between (23) then the source is DIN0. With the CTC2INT jumper located at (12) and one jumper at (CTC2,5) an interrupt number 5 is generated at the speed of CLOCK2. With the CTC2INT jumper located at (23) and one jumper at (CTC2,5) an interrupt number 5 is generated every time the digital input number 0 is pulsed from LOW to HIGH to LOW.
Adjust trimpot until Tp5 is +5.00 volts.

Tp5 –

header for cable
digital inputs and encoders

MULTIQ TH
Quanser Consulting

select source

CT2INT CT1 DINO

header for analog inputs
cable #1

Factory configuration
jumpers between CTC1 and INT5

Select interrupt sources

Select base address

select interrupt line

END

8080

8080

CTC0

CTC1

A/B

INT5

Select base address

Factory configuration
jumpers between CTC1 and INT5
B.3 Manual: Dynaserv DD Servo-Actuator DM/SD series

Instruction manual, 22 pages

Instruction Manual

DYNASERV

DD Servo-Actuator
DM / SD Series
(New Driver Model)

YOKOGAWA
Yokogawa Precision Corporation

IM A301-E
INTRODUCTION

Thank you for purchasing our DYNASERV DD Servo-Actuator (New Driver Model).
The DYNASERV is a high torque, high velocity, highly accurate outer rotor type servo-actuator
which can be used in a wide range of field applications related to FA (Factory Automation),
including industrial robots, indexes, etc.

This instruction manual covers the DM/SD series. Be sure to read this instruction manual
prior to operating the DYNASERV.

CAUTIONS

- Copying or the reproduction by any means of all or any part of the contents of this manual
  without permission is strictly prohibited.

- Yokogawa Precision Corp. reserve the right to change the contents of this manual without
  prior notification.

- While every effort has been made to ensure accuracy in the preparation of this manual, if
  you should however notice any discrepancies, errors or omissions, kindly contact your dealer
  or the authorized service personnel of Yokogawa Precision Corp. or its authorized agency.

- Yokogawa Precision Corp. shall bear no responsibility for indirect or consequential damages
  such as, but not so as to limit the foregoing, the loss of profit, or the loss of production,
  caused by the use of our products in accordance with this manual.
**Warning on Installation and Operation**

1. Never install the motor with the rotor fixed and the stator set free for rotation.
2. Ensure that the power is switched off when removing the side panel of the driver for jumper setting, etc. Dangerously high voltage is present inside the unit.
3. The motor rotates at high speed with high torque. Beware of the rotating radius of the load when operating the motor with the load installed.
4. Ensure adequate grounding at the ground terminal.
5. When installing a load to the rotor of the motor, allow a space of 1mm or more between the top surface of the motor and the surface of the load in order to maintain the proper alignment of the surfaces. Never apply any force or press fit any materials into the center hole. (See the figure below.)

![Center hole](image)

6. Never touch the bolts (indicated by arrow in the figure) which fix the bottom part of the rotor of the motor. (See the figure on the right.) Loosening or tightening these bolts may change the electrical commutation angle, and may result in faulty rotation.
7. Materials easily affected by magnetism must never be brought close to the motor as the surface of the motor is magnetized.
8. Install the motor in an appropriate location as the motor is not dust proof, watertight or oil proof.
9. If the motor is used with oscillating rotation movements with a small angle (50° or less), then carry out a running-in operation with back-and-forth movement about 10 times, each move exceeding an angle of 90° at least. The running-in operation must be carried out every 10,000 times of back and forth oscillation movements in order to ensure proper lubrication of the bearings.
10. Compatibility of the motor with the driver or vice versa when they are of the same model is possible only when they are of the same type. (i.e. when the motor code is DM1□□□□□□×1, and the driver code is SD□□□□□□, the □□□□□□ of the motor and the driver shall be the same.)
11. Never disassemble or modify the motor or the driver. When such disassembling or modification is required, consult Yokogawa Precision Corp. or its authorized agency.

**Warning on Installation and Operation**

12. If the motor is placed on the floor or the like as shown below when carrying or installing the DYNASERV, the cable is bent by the weight of the motor and this bending may cut the conductor wire. When placing the motor, be sure to use a supporting base which protects the cable from being bent.

- The minimum bending radius shall be 50mm or more when installing the motor with the cable being bent. Do not apply bending force repeatedly to the cable when it is used.
- The cable specifications do not include application with a robot.

![Minimum bending radius R 50mm or more](image)

13. Appropriate centering and alignment must be carried out when connecting the motor to a load. The shaft metal of the motor may get damaged if the centering offset remains 10μm or more.

![Centering offset: 10μm or less](image)

14. Never carry out a withstanding voltage test. Carrying out this test ever accidentally may damage the circuits. When such withstanding voltage tests are required, consult Yokogawa Precision Corp. or its authorized agency.
1. PRODUCT OUTLINE

1.1 DYNASERV, DM/SD Series

The DYNASERV series are high speed, high torque, highly accurate outer rotor type servo actuators which embody the measurement technology and control technology developed during Yokogawa Electric Co.'s long experience in the field.

The DYNASERV is composed of a motor section incorporating an encoder, and a driver section.

There are two DYNASERV mode types—the 4 models of A Series with an output torque of 50-200Nm, and 4 models of B Series with an output torque of 15-60Nm. The A and B Series have an outer diameter of 204mm and 160mm respectively and a cylindrical hole of 68mm and 25mm diameter respectively.

There is a driver model for each motor.

The drivers are available in types so that they can correspond to motor types. The drivers are available also with 100-115V power supply type and 200-230V power supply type.

1.2 Standard Product Configuration

The standard product set consists of the following components. When after unpacked, ensure that the product corresponds to the correct model, and also ensure that the types and quantities of standard accessories are also correct.

![Diagram](image)

### Table 1.1 DYNASERV Models DM/SD Series

<table>
<thead>
<tr>
<th>Motor Model No.</th>
<th>Driver Model No.</th>
<th>Max. Torque (N.m)</th>
<th>Rated Output (W)</th>
<th>Rated Rotation Speed (rpm)</th>
</tr>
</thead>
<tbody>
<tr>
<td>DM1200A50-1</td>
<td>SD1200A50-1</td>
<td>200</td>
<td>820</td>
<td>1</td>
</tr>
<tr>
<td>DM1100A50-1</td>
<td>SD1100A50-1</td>
<td>150</td>
<td>820</td>
<td>1</td>
</tr>
<tr>
<td>DM1000A50-1</td>
<td>SD1000A50-1</td>
<td>100</td>
<td>820</td>
<td>1</td>
</tr>
<tr>
<td>DM1600B50-1</td>
<td>SD1600B50-1</td>
<td>40</td>
<td>220</td>
<td>1.5</td>
</tr>
<tr>
<td>DM1500B50-1</td>
<td>SD1500B50-1</td>
<td>35</td>
<td>220</td>
<td>1.5</td>
</tr>
<tr>
<td>DM1000B50-1</td>
<td>SD1000B50-1</td>
<td>20</td>
<td>220</td>
<td>1.5</td>
</tr>
<tr>
<td>DM1012B50-1</td>
<td>SD1012B50-1</td>
<td>15</td>
<td>125</td>
<td>1.5</td>
</tr>
</tbody>
</table>

1.3 Model and Specification Code

The DYNASERV, DM/SD Series motor and driver Model Nos., specification code of the rating nameplate are as shown in the following.

(1) Motor Section and Specification Code

- Motor series name
- Design version
- Maximum output torque (Nm with 3 digits)
- Motor Model No./outer diameter
  (A series: dia. 10", B series: dia. 6")
- Standard export model
- External appearance (0: Standard)
- With compatibilities

(2) Driver Section and Specification Code

- Driver series name
- Design version
- Maximum output torque (Nm with 3 digits)
- Motor Model No./outer diameter
  (A series: dia. 10", B series: dia. 6")
- Standard export model
- Compatibility (2: compatible with the former SD driver)
- Power voltage (1: 100-115V power requirements, 2: 200-230V power requirements)
- Built-in interface board type (S: Serial pulse)
- Mechanical resonance filter (1: With equalizer filter, 0: Without filter)

Note: The motor and the driver are compatible within the same model type. For compatibility, the upper five digits of the motor code type (DM1200A50) and the driver code type (SD1200A50) shall be the same.
2. FUNCTIONAL DESCRIPTION

2.1 Motor Section

2.2 Driver Section

2.3 Driver Panel Surface

* : Both GND terminals are connected.

Figure 2.3 Name and Exploration of the Controls and Terminals on the Driver Panel
3. PREPARATION FOR OPERATION

3.1 Initial Setting

(1) Setting of the Jumper Switches in the Driver Box

![Figure 3.1 Setting of the Jumper Switches in the Driver Box](image)

Certain jumpers, switches and variable resistors within the driver box may need to be set by the customer. However, prior to shipment, they are set as shown on the next page. See the figure above for their locations.

To remove the side plate of the driver box, unscrew the 4-screws shown in the figure above.

**CAUTION**

However, prior to commencing any operation, always turn OFF the power. Further, never touch the high-voltage generation section, even with the power turned OFF.

For setting and adjustment procedures, see the following pages. Never touch the switches and variable resistors other than those specified.

(2) Jumper Settings Done Prior to Shipment

- **<JP1> Jumper**
  - **MODE**: See the next page
  - **CALIB**: See the next page
  - **RATE#1**: Position command pulse multiplying factor setting
  - **RATE#2**: Position command pulse multiplying factor setting
  - **UD/AB**: With jumper / A/B-phase, Without jumper / Up/Down pulse
  - **VFFH**: Velocity feed forward amount setting (Note 1)
  - **VFFM**: Velocity feed forward amount setting (Note 1)
  - **VFFL**: Velocity feed forward amount setting (Note 1)
  - **GAIN H**: DC gain magnification setting (Note 2)

- **<JP2> Jumper**
  - **I**: Velocity I type control
  - **P**: Velocity P type control
  - **100**: Velocity detection filter (Hz) selection (Open when a mechanical resonance filter is installed)
  - **200**: Velocity detection filter (Hz) selection (Open when a mechanical resonance filter is installed)
  - **PV**: Mode selection
  - **VEL**: Velocity input
  - **TORQ**: Torque input
  - **ALM**: Open for standard models
  - **TLIM**: Open for standard models

**Note**: [**S**] indicates setting prior to shipment.

<table>
<thead>
<tr>
<th>Velocity Feed Forward Amount (%)</th>
<th>VFFH</th>
<th>VFFM</th>
<th>VFFL</th>
</tr>
</thead>
<tbody>
<tr>
<td>Shorted</td>
<td>Shorted</td>
<td>Shorted</td>
<td>100</td>
</tr>
<tr>
<td>Shorted</td>
<td>Shorted</td>
<td>Open</td>
<td>95</td>
</tr>
<tr>
<td>Shorted</td>
<td>Open</td>
<td>Shorted</td>
<td>90</td>
</tr>
<tr>
<td>Shorted</td>
<td>Open</td>
<td>Open</td>
<td>85</td>
</tr>
<tr>
<td>Open</td>
<td>Shorted</td>
<td>Shorted</td>
<td>80</td>
</tr>
<tr>
<td>Open</td>
<td>Shorted</td>
<td>Open</td>
<td>75</td>
</tr>
<tr>
<td>Open</td>
<td>Open</td>
<td>Shorted</td>
<td>70</td>
</tr>
<tr>
<td>Open</td>
<td>Open</td>
<td>Open</td>
<td>65</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Type</th>
<th>Gain Magnification</th>
</tr>
</thead>
<tbody>
<tr>
<td>With jumper</td>
<td>DC Gain x 12</td>
</tr>
<tr>
<td>Without jumper</td>
<td>DC Gain x 2</td>
</tr>
</tbody>
</table>

(3) Switch, Volume Settings Done Prior to Shipment

<table>
<thead>
<tr>
<th>Switch Name</th>
<th>Volume Name</th>
<th>Setting Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>DC GAIN</td>
<td>Minimum position</td>
<td></td>
</tr>
<tr>
<td>AC GAIN</td>
<td>Minimum position</td>
<td></td>
</tr>
<tr>
<td>POSW</td>
<td>Set to &quot;0&quot;</td>
<td></td>
</tr>
<tr>
<td>fs</td>
<td>Set to &quot;0&quot;</td>
<td></td>
</tr>
<tr>
<td>I_LIM</td>
<td>Set to &quot;0&quot;</td>
<td></td>
</tr>
<tr>
<td>TEST</td>
<td>Set to &quot;OFF&quot;</td>
<td></td>
</tr>
</tbody>
</table>
3. Preparation for Operation

3.2 Control Mode Setting

(1) Control Mode Types

The following 6 control modes are available for the DYNAMOTER IIIM/SD Series.

- Position control mode
- Speed control mode
- Torque control mode

The table on the right shows the relationship between the switches and variable resistors related to the control mode and the jumper pin settings for each control mode.

### Table 3.1 List of Control Modes and Jumper Pin Switch Settings

<table>
<thead>
<tr>
<th>Section</th>
<th>Jumper Name</th>
<th>Switch Name</th>
<th>Position Control</th>
<th>Velocity Control</th>
<th>Torque Control Mode</th>
</tr>
</thead>
<tbody>
<tr>
<td>JP1</td>
<td>MODE</td>
<td>&lt;RATE1#1&gt;</td>
<td>Open</td>
<td>Open</td>
<td>Open</td>
</tr>
<tr>
<td></td>
<td>CALIB</td>
<td>&lt;RATE2#1&gt;</td>
<td>Open</td>
<td>Open</td>
<td>Open</td>
</tr>
<tr>
<td></td>
<td>RATE #1</td>
<td>&lt;RATE#2&gt;</td>
<td>Open</td>
<td>Open</td>
<td>Open</td>
</tr>
<tr>
<td></td>
<td>UD/AD</td>
<td>&lt;POSW&gt;</td>
<td>Open</td>
<td>Open</td>
<td>Open</td>
</tr>
<tr>
<td></td>
<td>VFDU</td>
<td>&lt;POSW1&gt;</td>
<td>Open</td>
<td>Open</td>
<td>Open</td>
</tr>
<tr>
<td></td>
<td>VFPF</td>
<td>&lt;POSW2&gt;</td>
<td>Open</td>
<td>Open</td>
<td>Open</td>
</tr>
<tr>
<td></td>
<td>DAIN H</td>
<td>&lt;POSW3&gt;</td>
<td>Open</td>
<td>Open</td>
<td>Shorted</td>
</tr>
<tr>
<td></td>
<td>L</td>
<td>&lt;POSW4&gt;</td>
<td>Open</td>
<td>Open</td>
<td>Open</td>
</tr>
<tr>
<td></td>
<td>P</td>
<td>&lt;POSW5&gt;</td>
<td>Open</td>
<td>Open</td>
<td>Shorted</td>
</tr>
<tr>
<td></td>
<td>100</td>
<td>&lt;POSW6&gt;</td>
<td>Open</td>
<td>Open</td>
<td>Open</td>
</tr>
<tr>
<td></td>
<td>200</td>
<td>&lt;POSW7&gt;</td>
<td>Open</td>
<td>Open</td>
<td>Shorted</td>
</tr>
<tr>
<td></td>
<td>PV</td>
<td>&lt;POSW8&gt;</td>
<td>Open</td>
<td>Open</td>
<td>Open</td>
</tr>
<tr>
<td></td>
<td>VIL</td>
<td>&lt;POSW9&gt;</td>
<td>Open</td>
<td>Open</td>
<td>Shorted</td>
</tr>
<tr>
<td></td>
<td>TOOL</td>
<td>&lt;POSW10&gt;</td>
<td>Open</td>
<td>Open</td>
<td>Shorted</td>
</tr>
<tr>
<td></td>
<td>DC GAIN</td>
<td>&lt;POSW11&gt;</td>
<td>Open</td>
<td>Open</td>
<td>X</td>
</tr>
<tr>
<td></td>
<td>AC GAIN</td>
<td>&lt;POSW12&gt;</td>
<td>Open</td>
<td>Open</td>
<td>X</td>
</tr>
<tr>
<td></td>
<td>POSW</td>
<td>&lt;POSW13&gt;</td>
<td>Open</td>
<td>Open</td>
<td>X</td>
</tr>
<tr>
<td></td>
<td>L</td>
<td>&lt;POSW14&gt;</td>
<td>Open</td>
<td>Open</td>
<td>X</td>
</tr>
<tr>
<td></td>
<td>TEST</td>
<td>&lt;POSW15&gt;</td>
<td>Open</td>
<td>Open</td>
<td>X</td>
</tr>
</tbody>
</table>

Note: [ ] Validity, When the set value exerts an influence on motor operation.

(2) Feedback Pulse and Position Command Pulse Settings / JP1

The servo driver receives a signal from the encoder built into the motor, then outputs an A/B phase or UP/DOWN pulse signal to a higher-level controller. Jumper pins related to the feedback pulse signal are <RATE#/1 to 2> and <UD/AB>.

In addition, the position command pulse signal multiplication factor is determined by the setting of <RATE#/1 to 2>.

### Table 3.2 Setting of the Positioning Completion Width

<table>
<thead>
<tr>
<th>(POSW) Switch Setting</th>
<th>Positioning Completion Width (Unit: Pulse/#)</th>
<th>&lt;POSW&gt; Signal Setting</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1 H</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>5 H</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>10 L</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>50 L</td>
<td>3</td>
</tr>
<tr>
<td>4</td>
<td>200 L</td>
<td>4</td>
</tr>
<tr>
<td>5</td>
<td>4 H</td>
<td>5</td>
</tr>
<tr>
<td>6</td>
<td>10 L</td>
<td>6</td>
</tr>
<tr>
<td>7</td>
<td>40 L</td>
<td>7</td>
</tr>
<tr>
<td>8</td>
<td>200 L</td>
<td>8</td>
</tr>
<tr>
<td>9</td>
<td>80 L</td>
<td>9</td>
</tr>
<tr>
<td>10</td>
<td>400 L</td>
<td>10</td>
</tr>
<tr>
<td>11</td>
<td>8 H</td>
<td>11</td>
</tr>
<tr>
<td>12</td>
<td>50 L</td>
<td>12</td>
</tr>
<tr>
<td>13</td>
<td>100 L</td>
<td>13</td>
</tr>
<tr>
<td>14</td>
<td>400 L</td>
<td>14</td>
</tr>
<tr>
<td>15</td>
<td>800 L</td>
<td>15</td>
</tr>
</tbody>
</table>

R: 1 pulse = Unit’s resolution

(3) Velocity Signal Filter Setting / JP2

These jumpers are used to select the velocity signal filter cut-off frequency. The cut-off frequency is set to 100Hz with <200> shorted, and it is set to 200Hz with <200> shorted. However, when the resonance filter is connected, these jumpers must be kept open.

(4) Origin Pulse Output Signal Setting / z+, z- (Pin#43, 44)

This setting determines the duration of the reference zero (origin) position is achieved by equally dividing the positions of one rotation of the motor (for series A it is 100 and for series B it is 50). When the reference zero position is detected, the following potential signal will be output. Upon detecting the reference zero position, the potential of the detection signal changes from H to L when the motor is rotating in the CW (clockwise) direction, and it changes from L to H when the motor is rotating in CCW (counterclockwise) direction. The direction CW or CCW in these cases are defined as direction of rotation upon viewing the motor from the lead side.

(5) Positioning Completion Width Setting / S1

When positioning in the position control mode is completed, the CN1 connector CODN signal is set to ON. This positioning completion width can be selected by the [POSW] switch. The positioning completion width can be selected by the [POSW] switch on the front panel.

The table on the right shows the relationship between the [POSW] switches and the positioning completion width.

At the same time, when setting the position completion width using <POSW 0, 1> signal, set the [POSW] switch in 4 steps as shown in the table. With a combination of H and L of the <POSW> signals, the same selection as the [POSW] switch can be carried out.
3. Preparation for Operation

(6) Mechanical Resonance Filter (Notch Type) Adjustment (Optional)

The following explains the adjustment procedure when the mechanical resonance filter (notch type) is installed as an option. The board of the filter is located as shown below just inside the square cut-out on the side panel. The controls <Q1> and <Q2> on the board are used to set the notch frequencies at the first stage and the second stage respectively. The frequencies can be set within the range from 150Hz to 1.5kHz (the frequencies are factory-set to 1.5kHz when shipped).

Use the controls <Q1> and <Q2> to change the setting of the Q values. The Q values can be set within a range from 0.5 to 2.5 (0 to 20kΩ) (the Q values are factory-set to 2.5 at the time of shipping). The offset voltage shall be readjusted when the Q value has been changed. This voltage is to be adjusted using adjustment controls so that the voltage difference between <TP1> and <TP3> becomes 50mV or less.

The first-order delay filter is also located on this board. The frequencies can be selected from 20/80Hz, 30/120Hz and 40/160Hz, using a jumper. In addition, using an appropriate pair of C and R, a desired filter frequency can be set. The frequencies of the first-order delay filter are factory-set to 20/80Hz at the time of shipping.

![Notch Filter Board Layout](image)

**Figure 3.2 Mechanical Resonance Filter (Notch Type) Adjustment (Optional)**

3.3 External Wiring

(1) External Connection Outline Diagram

![External Connection Outline Diagram](image)

**Figure 3.3 External Connection Outline Diagram**

(2) Connection between the Motor and the Driver

![Connection between the Motor and the Driver](image)

**Figure 3.4 Connection between the Motor and the Driver**
(3) Typical Wiring Example (In the Position Control Mode)

100-115V/200-230V AC +10% to 15% 50/60Hz

Note: Prepare the 5V power supply on the user's side. Carry out complete noise rejection treatment in the position command pulse input section.

Typical Wiring Example:

Driver

- LINE (MIN)
- LINE (CONT)
- GND

Motor

- VA
- VB
- VC
- GND

Motor Cable

A
B
C

Encoder Cable

Encoder Section

Shield

Shaded wire treatment

See figure 3.6 on the next page.

Figure 3.5 Typical Wiring Example

(4) Connection to External Controller
(CN1 terminal I/O signal connection and external signal processing)

Controller

- 74SP or equivalent
- 74HC or equivalent

Driver

- CN1
- R

[Diagram of wiring connections]

- Insert R=1.2kΩ only at 5V < +V < 24V +V=24V
- Approx. 4mA (Max. drive current 15mA)
- Because the pulses are of high velocity, be sure to use a differential line driver/receiver.

Figure 3.6 Connection to External Connector
### Preparation for Operation

#### 3. Preparation for Operation

**Positioning completion signal**

#### List of Interface

**a) Input**

<table>
<thead>
<tr>
<th>Signal Name</th>
<th>Pin No.</th>
<th>Meaning</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>FN 3</td>
<td>1 (2)</td>
<td>Compliance setting (Serve motor setting)</td>
<td>(See Note 1)</td>
</tr>
<tr>
<td>FN 4</td>
<td>2 (4)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>FN 5</td>
<td>3 (6)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>FN 7</td>
<td>7 (8)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>POSW 1</td>
<td>9 (10)</td>
<td>Positioning completion pulse width</td>
<td>(See Note 1)</td>
</tr>
<tr>
<td>POSW 2</td>
<td>11 (12)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SGN+</td>
<td>19</td>
<td>Rotating direction command</td>
<td></td>
</tr>
<tr>
<td>SGN-</td>
<td>19</td>
<td></td>
<td></td>
</tr>
<tr>
<td>E/JST</td>
<td>21 (22)</td>
<td>Integral capacitor reset</td>
<td></td>
</tr>
<tr>
<td>BRGON</td>
<td>23 (24)</td>
<td>Servo ON</td>
<td></td>
</tr>
<tr>
<td>IAC'T/FACT</td>
<td>25 (26)</td>
<td>Integral / Proportional action selection</td>
<td></td>
</tr>
<tr>
<td>TLM/TFF</td>
<td>31 (32)</td>
<td>Torque Limit / Torque feed forward</td>
<td>For input of torque limit and torque feed forward (option)</td>
</tr>
<tr>
<td>GAIN T</td>
<td>37 (38)</td>
<td>Gain selection</td>
<td>(See Note 3)</td>
</tr>
<tr>
<td>GAIN N</td>
<td>33 (34)</td>
<td>Gain selection</td>
<td>(See Note 3)</td>
</tr>
<tr>
<td>TST</td>
<td>39 (40)</td>
<td>CPU reset</td>
<td></td>
</tr>
<tr>
<td>PUL+ / PULS</td>
<td>45</td>
<td>Position command pulse</td>
<td>Driver position command pulse signal</td>
</tr>
<tr>
<td>VIN</td>
<td>49</td>
<td>Velocity command input</td>
<td>Torque command input</td>
</tr>
<tr>
<td>A/GND</td>
<td>60</td>
<td>Analog input GND</td>
<td>Velocity input terminal analog GND</td>
</tr>
</tbody>
</table>

**Note:** (1) indicates VCC signal input's terminal.

**Note:** (2) / (3): FN 5 to 6 and POSW 1, 2 are logically wired in the "OR" configuration with the rotary switch and jumper pin (JP1/GAIN H).

**Note:** External controller, set this rotary switch to the "0" position and GAIN H to open.

**b) Output**

<table>
<thead>
<tr>
<th>Signal Name</th>
<th>Pin No.</th>
<th>Meaning</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>A+ / U+</td>
<td>13</td>
<td>Position feedback pulse signal</td>
<td>Pulse signal to indicate the motor rotating position. Either A+ / A- phase or UP/DOWN phase pulse can be selected by the jumper on the board.</td>
</tr>
<tr>
<td>A- / U-</td>
<td>14</td>
<td></td>
<td></td>
</tr>
<tr>
<td>B+ / D+</td>
<td>29</td>
<td></td>
<td></td>
</tr>
<tr>
<td>B- / D-</td>
<td>30</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SDY</td>
<td>35 (15)</td>
<td>Servo ready</td>
<td>The motor is ready to operate with this signal set to L. This signal is set to the H level about 3 seconds after driver power-up.</td>
</tr>
<tr>
<td>VELMON</td>
<td>17 (15)</td>
<td>Velocity monitoring</td>
<td>Signal for monitoring the number of motor revolutions to output positive voltage for CW rotation and negative voltage for CCW rotation. Velocity detection sensitivity is as shown on the following table. (See Note 3) Velocity detection sensitivity is not guaranteed for the number of revolutions in the range exceeding 7.5V.</td>
</tr>
</tbody>
</table>

**Note:** (1) indicates GND signal output.

---

### List of Output Interface

**a) List of Output Interface (27)**

<table>
<thead>
<tr>
<th>Signal Name</th>
<th>Pin No.</th>
<th>Meaning</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>CON</td>
<td>27 (28)</td>
<td>Positioning completion signal</td>
<td>This signal is set to L when the deviation counter value becomes less than the POSW switch set value.</td>
</tr>
<tr>
<td>OVER</td>
<td>41 (42)</td>
<td>Deviation counter overflow or overspeed</td>
<td>Deviation counter overflow signal is output only in the position control mode, and this signal is set to L when the deviation counter value exceeds 32767. The overspeed signal is set to L when feedback pulse output frequency becomes greater than about 3MHz.</td>
</tr>
<tr>
<td>Z+</td>
<td>43</td>
<td>Origin pulse</td>
<td>Signal for detecting the original positions obtained by equally dividing 1 revolution of the motor (100 for the A series and 60 for the B series), and changes from H to L during CW rotation and from L to H during CCW rotation.</td>
</tr>
<tr>
<td>Z-</td>
<td>44</td>
<td></td>
<td></td>
</tr>
<tr>
<td>OVL</td>
<td>47 (48)</td>
<td>Overload</td>
<td>Set to H during overloading, it simultaneously reduces motor current automatically to 1/2.</td>
</tr>
</tbody>
</table>

**Note:** (1) indicates GND signal output.

---

### List of Input Interface

**Table 3.3 List of Input Interface**

<table>
<thead>
<tr>
<th>Signal Name</th>
<th>Pin No.</th>
<th>Meaning</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>POSW 1</td>
<td>9 (10)</td>
<td>Positioning completion pulse width</td>
<td>(See Note 1)</td>
</tr>
<tr>
<td>POSW 2</td>
<td>11 (12)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Note:** (1) indicates VCC signal input's terminal.

**Note:** (2) / (3): FN 5 to 6 and POSW 1, 2 are logically wired in the "OR" configuration with the rotary switch and jumper pin (JP1/GAIN H).

**Note:** External controller, set this rotary switch to the "0" position and GAIN H to open.

---

### List of Output Interface

**Table 3.4 List of Output Interface (27)**

<table>
<thead>
<tr>
<th>Signal Name</th>
<th>Pin No.</th>
<th>Meaning</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>FN 2</td>
<td>H</td>
<td>H</td>
<td>H</td>
</tr>
<tr>
<td>FN 3</td>
<td>H</td>
<td>H</td>
<td>L</td>
</tr>
<tr>
<td>FN 4</td>
<td>H</td>
<td>H</td>
<td>H</td>
</tr>
<tr>
<td>FN 5</td>
<td>H</td>
<td>L</td>
<td>L</td>
</tr>
<tr>
<td>FN 6</td>
<td>H</td>
<td>L</td>
<td>H</td>
</tr>
<tr>
<td>FN 7</td>
<td>H</td>
<td>H</td>
<td>H</td>
</tr>
<tr>
<td>FN 8</td>
<td>H</td>
<td>H</td>
<td>L</td>
</tr>
<tr>
<td>FN 9</td>
<td>H</td>
<td>H</td>
<td>L</td>
</tr>
<tr>
<td>FN 10</td>
<td>H</td>
<td>L</td>
<td>H</td>
</tr>
<tr>
<td>FN 11</td>
<td>H</td>
<td>L</td>
<td>L</td>
</tr>
<tr>
<td>FN 12</td>
<td>H</td>
<td>L</td>
<td>L</td>
</tr>
<tr>
<td>FN 13</td>
<td>H</td>
<td>H</td>
<td>H</td>
</tr>
<tr>
<td>FN 14</td>
<td>H</td>
<td>H</td>
<td>L</td>
</tr>
<tr>
<td>FN 15</td>
<td>H</td>
<td>L</td>
<td>L</td>
</tr>
<tr>
<td>FN 16</td>
<td>H</td>
<td>L</td>
<td>L</td>
</tr>
</tbody>
</table>

**Note:** *: The product of this GAIN value and the variable resistor position (0.5 to 5.5) becomes the total gain.

---

**Table 3.4 List of Output Interface (27)**

<table>
<thead>
<tr>
<th>Signal Name</th>
<th>Pin No.</th>
<th>Meaning</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>A+ / U+</td>
<td>13</td>
<td>Position feedback pulse signal</td>
<td>Pulse signal to indicate the motor rotating position. Either A+ / A- phase or UP/DOWN phase pulse can be selected by the jumper on the board.</td>
</tr>
<tr>
<td>A- / U-</td>
<td>14</td>
<td></td>
<td></td>
</tr>
<tr>
<td>B+ / D+</td>
<td>29</td>
<td></td>
<td></td>
</tr>
<tr>
<td>B- / D-</td>
<td>30</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SDY</td>
<td>35 (15)</td>
<td>Servo ready</td>
<td>The motor is ready to operate with this signal set to L. This signal is set to the H level about 3 seconds after driver power-up.</td>
</tr>
<tr>
<td>VELMON</td>
<td>17 (15)</td>
<td>Velocity monitoring</td>
<td>Signal for monitoring the number of motor revolutions to output positive voltage for CW rotation and negative voltage for CCW rotation. Velocity detection sensitivity is as shown on the following table. (See Note 3) Velocity detection sensitivity is not guaranteed for the number of revolutions in the range exceeding 7.5V.</td>
</tr>
</tbody>
</table>

**Note:** (1) indicates GND signal output.
3.4 Installation

When the product is delivered, first check the product type and Model No. as well as for the presence or absence of accessories and for the exact combination of the motor and the driver.

(1) Motor-section Mounting

The motor-section can be mounted either vertically or horizontally. However, incorrect mounting and unsuitable mounting location may shorten the motor service life and cause trouble. Therefore, always observe the following.

a) Installation Location

The motor section is designed for indoor use. Therefore, the installation location must be such that:
- There are no corrosive and explosive gases.
- Ambient temperature is between 0 and 45°C
- Dust concentration is low, with adequate air ventilation and low humidity.

Note: The DYNASERV is not drip proof or oil proof, so it should be covered by a suitable drip proof and oil proof cover.

b) Mechanical Coupling

- When coupling a load with the motor rotor section, make sure there is a clearance of more than 1mm between the motor upper surface and the load.
- Secure the motor rotor and stator by tightening the setscrew with torques of less than the following values as given below.
- Motor base levelness deviation must be maintained less than 0.01mm.

![Diagram of Motor Section Mounting](image)

Rack Mounting (L-shaped angle brackets)

(a) Correct Installation

(b) Incorrect Installation

Figure 3.7 Tightening Torque

(2) Driver Section Mounting

The standard driver is designed for rack mounting.

a) Installation Location

- When there is a heat generation source near the installation location, ensure that temperature does not exceed 50°C in the approximately of the driver by providing an appropriate heat shield or cover, etc.
- When there is a vibration generating source near to the driver then mount the driver on the rack with appropriate vibration insulators.
- Further the installation must be at a location when the humidity is low, and when the surrounding environment is free from high temperature, dust, metal powders and corrosive gases.

b) Mounting Procedure

- Normally, the driver is rack mounted (L-shaped angle brackets) with its driver panel facing forward and its top and bottom surfaces horizontal. However, it may be mounted with its driver panel facing upward. Always avoid mounting it with its panel surface facing sideways or upside down. (See the figure(b) below.)
- Mount the driver using 4-screw holes at the top and bottom of the driver panel.

![Diagram of Driver Section Mounting](image)

More than 25mm

Figure 3.8 Driver Section Mounting
3. Preparation for Operation

3.5 Wiring Cables

(1) Cable Sizes and Rated Currents

<table>
<thead>
<tr>
<th>Table 3.5 Cable Sizes and Rated Currents</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Input</strong></td>
</tr>
<tr>
<td>D/A converter supply cable</td>
</tr>
<tr>
<td>Current (A)</td>
</tr>
<tr>
<td>Cable size</td>
</tr>
<tr>
<td>Motor cable</td>
</tr>
<tr>
<td>Current (A)</td>
</tr>
<tr>
<td>Cable size</td>
</tr>
<tr>
<td>Jumper wire</td>
</tr>
<tr>
<td>Current (A)</td>
</tr>
<tr>
<td>Cable size</td>
</tr>
<tr>
<td>Interface cable</td>
</tr>
<tr>
<td>Current (A)</td>
</tr>
<tr>
<td>Cable size</td>
</tr>
<tr>
<td>Encoder cable</td>
</tr>
<tr>
<td>Current (A)</td>
</tr>
<tr>
<td>Cable size</td>
</tr>
<tr>
<td>Grounding</td>
</tr>
<tr>
<td>Current (A)</td>
</tr>
<tr>
<td>Cable size</td>
</tr>
</tbody>
</table>

Notes:
1. Current values: r.m.s. of rated currents
2. Cable size: Cross-sectional area in mm²
3. Cross-sectional area of conductor marked with HV: More than 0.5mm² tinned-plated twisted wire
4. Current rating of the terminal used for CN1 and CN2: Less than 64, 16mm or less.
5. Current rating obtained under the condition that ambient temperature is 40°C and the rated current flows through 3 bundled leadwires.
6. HV: Heat resistant polyvinyl chloride tinned wire maintains insulation resistance up to an operation temperature of 70°C

4. OPERATION CAUTIONS

4.1 Input and Output Signal Cautions

(1) Position Command Pulse Input Signal (PULSE±)

This is a drive position command pulse signal. The pulse signal uses positive switching logic with a minimum pulse width of 150ns.

(2) Motor Rotating Direction Command Input Signal (SIGN±)

A signal indicating the motor rotation. The motor rotates in the CW direction with this signal set to H and CCW direction with this signal set to L. Timing of this signal with respect to the positioning command pulse signal at the output is as shown below.

4.2 More than 150ns

Note:
- The pulse should be set to active H.
- This means that current does not flow through the driver photo-coupler when the pulse is not output.

(3) Velocity Command Input (VIN)

An analog input signal is used as the motor rotating velocity command value. The maximum velocity in the CW direction at +6V, and the maximum velocity in the CCW direction at -6V. (DR Std' Series) (In the -6V to +6V, input range, input impedance is 100kΩ.)

(4) Velocity Monitoring Output (VELMON)

Motor analog velocity monitoring output
- Output voltage: At maximum velocity +6V (CW)
- At maximum velocity -6V (CCW) (output impedance is 1kΩ)

(5) A/B Phase, UP/DOWN Pulse Output Signals (A/±, B/±)

Pulse signals to indicate the motor position. The following 2 pulse output status can be selected by jumpers on the controller board.
4. Operation Cautions

a) A/B Phase Output Pulse
   The following pulse signal is output with the jumper <UD/AB> on the controller board shorted.

<table>
<thead>
<tr>
<th></th>
<th>CCW Direction (counterclockwise)</th>
<th>CW Direction (clockwise)</th>
</tr>
</thead>
<tbody>
<tr>
<td>A-phase</td>
<td>750kHz Max.</td>
<td>750kHz Max.</td>
</tr>
<tr>
<td>B-phase</td>
<td>90°</td>
<td>90°</td>
</tr>
</tbody>
</table>

b) UP/DOWN Output Pulse
   The following pulse signal is output with the jumper <UD/AB> on the controller board opened.

<table>
<thead>
<tr>
<th></th>
<th>CCW Direction (counterclockwise)</th>
<th>CW Direction (clockwise)</th>
</tr>
</thead>
<tbody>
<tr>
<td>UP-pulse signal</td>
<td>3MHz Max.</td>
<td></td>
</tr>
<tr>
<td>DOWN-pulse signal</td>
<td>3MHz Max.</td>
<td></td>
</tr>
</tbody>
</table>

Note: Because PULSE, SIGN E, A/DE, and E± are high speed pulse signals, a differential PC interface is required.

4.2 Power ON/OFF

Kindly pay attention to the following when the power is turned ON.
1) When turning ON the main and control circuit power supplies, turn them ON simultaneously or turn ON the control circuit power first.
2) When turning them OFF, turn them OFF simultaneously (including after instantaneous power failure), or turn OFF the main circuit power first.
3) Inrush current in both the main and control power circuits is about 35A (300VAC Source) and 20A (100VAC Source) peak. The motor is set to the servo status about 200ms after SRVON is set to L.
4) The motor is set to the servo status about 200ms after SRVON is set to L.
5) When the main power circuit is active, RDY = H indicates driver trouble. Therefore, use a sequence circuit to turn OFF the main power circuit at RDY = H. However, after the control and main circuit power supplies are turned ON, the RDY = H condition is maintained for up to 3 seconds. Therefore, hold the power-ON signal for more than 3 seconds.

5. Control Mode and Adjustment

5.1 Position Control Mode Adjustment

In the position control mode, motor positioning control is performed according to the command position sent from the higher-level controller. Two control methods are available in the velocity control mode: the I-PD type control system is selected with the CN1 connector <I.ACT/P.ACT> signal set to H, and the P type control system, with the same signal set to L. Usually, the I-PD type control system is selected in the positioning mode of operation.

1) I-PD Type Position Control
   This method uses position integral feedback and is suitable for highly accurate positioning. A stable control characteristic is also achieved even under load variation. In this mode, the adjustment of <fe switch>, <I.LIM switch> and <DC gain adjustment control> becomes necessary.

a) <fe Switch>
   The 1 to 16Hz position control system band is selected from a scale of 0 to P. However, in this case CN1 connectors FN 0 to FN 3 must all be set to H.

b) <I.LIM Switch>
   This prevents the wind-up phenomenon by limiting the output of the digital integrator during software servo computation. The larger the switch No., the larger the limited value. The smaller the limited value, the shorter the settling time. However, if the limited value becomes too small, the motor output torque is also limited. Therefore, it can be said that it is better to make the switch value large within the non-wind-up range. The fine adjustment is performed during the acceleration/deceleration operation.

c) <DC Gain Adjustment Control>
   The combination of driver CN1 connector GAIN H to L signals results in an adjustment range of from 0.5 to 120 times. The DC gain should be as large as possible. When there is a change in inertia adjust the gain so that it becomes optimum at the maximum load.

(2) P Type Position Control
   Positioning accuracy is not high because proportional control is used for positioning feedback. The velocity controls which can be set for simultaneous selection are P and I types, and they can be set with a jumper.
   With the P type velocity control (P-P type), a torque output which is proportional to the positioning error is obtained, and compliance control is possible. In this control mode, only <fe switch> and <DC gain control> are to be adjusted.
   With the I type velocity control, a high tact positioning can be achieved. In this control mode, the amount of velocity feed forward is to be adjusted with a jumper in addition to <fe switch>, <DC gain control> and <AC gain control>.
(3) Position Control System Adjustment Procedure (See the Following Figure.)

The position control system can be adjusted in the test mode. Turning ON the test switch at the front of the driver generates a 2.5Hz square-wave position command signal inside the driver to output the motor position to the POSN signal terminals. At this time, ensure that the motor exhibits reciprocal action at very small rotating angles.

1. The adjustment procedure for I-PD type position control in the test mode is as follows.

   Step 1: Connect an oscilloscope to the <POSN> signal terminals.
   Step 2: Set the CN1 connector <SERVO> signal to L. At this time, set the <TEST switch> to [OFF].
   Step 3: Set the <TEST switch> at the front of the driver to ON.
   Step 4: Adjust the <fc switch>. Its variable range is from 1 to 16Hz and it should be set to about 10Hz (scale graduation: 9) under normal load conditions. Set the <ILM switch> to a large value within the range where there is no hunting. Select the <GAIN H> to L signal so that they match the load condition.
   Fine adjustment is done by the <DC gain adjustment control>.
   Perform the above adjustments such that the POSN signal becomes a square wave.

   Step 5: Set the <TEST switch> at the front of the driver to OFF.
   Step 6: Set the CN1 connector <SERVO> signal to H.

2. The adjustment procedure for PI type position control in the test mode is as follows.

   Step 1: Connect an oscilloscope to the <POSN> signal terminals.
   Step 2: Set the CN1 connector <SERVO> signal to L. At this time, set the <TEST switch> to [OFF].
   Step 3: Set the <TEST switch> at the front of the driver to ON.
   Step 4: Adjust the <fc switch>. Its variable range is from 1 to 16Hz and it should be set to about 10Hz (scale graduation: 9) under normal load conditions. Set the <AC gain control> to a large value within the range where there is no hunting. Fine adjustment is done by the <DC gain control>.
   Perform the above adjustments such that the POSN signal becomes a square wave.

   Step 5: Set the <TEST switch> at the front of the driver to OFF.
   Step 6: Set the CN1 connector <SERVO> signal to H.

(4) Procedure for Adjustment without Measuring Instruments

The preceding section demonstrates the procedure for performing adjustments while monitoring the waveform; this section demonstrates an adjustment procedure that does not use any measuring instruments. These adjustment methods are valid only in the case of the position control mode (I-PD type, the setting at the time of shipping).

1) Calculate or otherwise verify the load inertia. In order to make use of this adjustment method, the load inertia must be known accurately. At this time, calculate the load multiple (K) by dividing the load inertia (JL: kgm² units) by the motor (DYNASERV) rotor inertia (Jm).

2) Set the <TEST switch> on the driver front panel to [ON].

3) Take the computed load multiple and refer to the tables of adjustment settings for the individual DYNASERV models (see pages 22 through 23). For example, suppose that K is (15) for a DM1300A; thus the "5" range applies for this case. Next, follow this row to the right for the setting values.

4) First, look at the value in <DC gain> "Column 1". Because the value is (25), turn the <DC gain> control to (25).

When the value for either A or B series is within range 1 or 2 (DC gain to be set is 5 or less), change the DC gain switching signal to <H> before carrying out the setting.

5) Similarly, take the "Column 1" values for <fc> and <ILM> in the same row, and set their respective controls to those values.

6) When the above settings have been completed, set the <TEST switch> to [OFF] to complete the adjustments.

Note: The GAIN value for signal selection shown below is multiplied by the DC GAIN level value to obtain the total gain.

<table>
<thead>
<tr>
<th>Range</th>
<th>Load Multiple K</th>
<th>Setting Value</th>
<th>DC GAIN</th>
<th>Setting Value</th>
<th>Load Multiplc K</th>
<th>Setting Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>SD1000AS</td>
<td>SD1000AS</td>
<td>SD1000AS</td>
<td>DC GAIN</td>
<td>fc</td>
<td>DC GAIN</td>
<td>fc</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>3</td>
<td>2</td>
<td>3</td>
<td>16</td>
<td>3</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>3</td>
<td>2</td>
<td>4</td>
<td>3</td>
<td>6</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>4</td>
<td>3</td>
<td>4</td>
<td>3</td>
<td>6</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>3</td>
<td>6</td>
</tr>
<tr>
<td>5</td>
<td>5</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>3</td>
<td>6</td>
</tr>
<tr>
<td>6</td>
<td>6</td>
<td>5</td>
<td>4</td>
<td>4</td>
<td>3</td>
<td>6</td>
</tr>
<tr>
<td>7</td>
<td>7</td>
<td>5</td>
<td>5</td>
<td>4</td>
<td>3</td>
<td>6</td>
</tr>
<tr>
<td>8</td>
<td>8</td>
<td>6</td>
<td>5</td>
<td>5</td>
<td>3</td>
<td>6</td>
</tr>
<tr>
<td>9</td>
<td>9</td>
<td>7</td>
<td>5</td>
<td>5</td>
<td>3</td>
<td>6</td>
</tr>
</tbody>
</table>

Table 5.1(a) Setting of the DYNASERV Controls (for A Series)
5. Control Mode and Adjustment

5.2 Velocity Control Mode Adjustment

In the velocity control mode, the motor rotating angle is controlled so as to correspond to the velocity command voltage (-10V to +10V) from the higher-level controller. The two control methods can be selected in the velocity control mode.

The following table shows the relationship between velocity command voltage and motor velocity.

<table>
<thead>
<tr>
<th>Model</th>
<th>Velocity/Impulse Voltage (rpm/V)</th>
</tr>
</thead>
<tbody>
<tr>
<td>DM1015B to DM1060B</td>
<td>1/10</td>
</tr>
<tr>
<td>DM1000A to DM1000A</td>
<td>1/10</td>
</tr>
</tbody>
</table>

(1) PI Type Velocity Control

The use of integral/proportional action in velocity control achieves smooth, disturbance-resistant control. This is the same control mode used in the conventional DC/AC servo motor control. In this control mode, only the two <DC gain> and <AC gain> adjustment controls are adjusted.

a) <DC Gain>

The combination of the driver CN1 connector GAIN 0 to 2 signals results in an adjustment range of from 0.5 to 120 times.

b) <AC Gain>

Velocity loop band damping is adjusted.

(2) P Type Velocity Control

Since velocity control is effective only in proportional action, response is fast but is strongly influenced by disturbances in the controlled motor. In this control mode, only the <DC gain> variable resistor at the front of the driver is adjusted. While in this velocity mode, the test switch becomes invalid.

(3) Adjustment of Velocity Control System

Adjustment of velocity control system can be carried out in the test mode.

By turning the test switch on the front of the driver to ON, applies a 2.5Hz square waveform signal to the speed input in the driver, and the motor starts moving back and forth movements, repeatedly, at a small rotating angle. Under this condition, observe the <VEL signal> at the front panel on an oscilloscope, and adjust <DC gain> and <AC gain> so that <VEL signal> becomes an optimum waveform as shown in the figure below.

5.3 Torque Control Mode Adjustment

In the torque control mode, current flows through the motor corresponding to the current command voltage (-8V to +8V) from the higher-level controller. Motor output torque depends on the current. Therefore, torque is 0 at 0V of command voltage, and the maximum torque is produced at 8V.

Note: When desiring of using the torque control mode, carefully plan and design the velocity & position control loops and a proper interlocking system such that, the final control system meets the exact specifications of the application.
6. MAINTENANCE AND INSPECTION

6.1 Motor Section

Only simple daily checks need be carried out on the Motor section. Check for noise or excessive vibration which is not normal. Never disassemble the Motor section. If the condition of the Motor section is not normal after 20,000 hours of use or after five years from the installation, replace the Motor section together with the Driver section. This time duration may change depending on the environmental and operating conditions where the Motor is used.

6.2 Driver Section

There is no need of daily maintenance for the Driver part. However, clean the Driver part periodically to prevent it from poor insulation caused by accumulated dust.

7. TROUBLESHOOTING AND MEASURES

7.1 Motor Trouble and Measures

Whenever any abnormal condition occurs while operating the motor, check the LED display on the front panel of the driver. Take appropriate countermeasures as shown below if the cause of the abnormal condition is determinable by the indication of the LED display.

When the motor does not function normally, even after the following measures have been taken, immediately cease operation and contact the Yokogawa Precision Corp. or it's authorized agent.

<table>
<thead>
<tr>
<th>Troubleshooting and Measures (3/3)</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Trouble</th>
<th>Estimated Cause</th>
<th>Inspected Item</th>
<th>Measured Item</th>
<th>Page for Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>The motor is not servo locked.</td>
<td>No AC power is fed.</td>
<td>Wiring inspection</td>
<td>Apply the specific AC power</td>
<td>Pages 10, 11</td>
</tr>
<tr>
<td></td>
<td>The servo ON (SRVON) terminal is set to L.</td>
<td>Inspection</td>
<td>Set to L.</td>
<td>Page 12</td>
</tr>
<tr>
<td></td>
<td>The CPU reset (RST) terminal is set to L.</td>
<td>Inspection</td>
<td>Set to H.</td>
<td>Page 12</td>
</tr>
<tr>
<td></td>
<td>Sr, L, LIM, DC gain is small.</td>
<td>Inspection</td>
<td>To be adjusted to an appropriate value.</td>
<td>Pages 21 to 24</td>
</tr>
<tr>
<td>The motor does not start.</td>
<td>Incorrect external wiring.</td>
<td>Inspect wiring.</td>
<td>Re-wire correctly by referring to the connection diagram.</td>
<td>Pages 10, 11</td>
</tr>
<tr>
<td></td>
<td>Sr, L, LIM, DC gain is small.</td>
<td>Inspection</td>
<td>To be adjusted to an appropriate value.</td>
<td>Pages 21 to 24</td>
</tr>
<tr>
<td>Motor rotation is unstable.</td>
<td>Imperfect connection</td>
<td>Check the connection of each phase of A, B, C and GND.</td>
<td>Re-wire correctly by referring to the connection diagram.</td>
<td>Pages 10, 11</td>
</tr>
<tr>
<td></td>
<td>The motor and driver combination is inappropriate.</td>
<td>Check the combination Nos. on the nameplate.</td>
<td>If the combination is incorrect, return the pair to YPC or it's authorized agent.</td>
<td>Page 2</td>
</tr>
</tbody>
</table>

Table 7.1 Motor Troubleshooting and Measures (2/2)

<table>
<thead>
<tr>
<th>Troubleshooting and Measures (2/2)</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Trouble</th>
<th>Estimated Cause</th>
<th>Inspected Item</th>
<th>Measured Item</th>
<th>Page for Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>The motor overheat.</td>
<td>Ambient temperature is high.</td>
<td>Check to see if ambient temperature is more than 40°C</td>
<td>Lower the ambient temperature to below 45°C</td>
<td></td>
</tr>
<tr>
<td></td>
<td>The motor is overloaded.</td>
<td>Operate the motor under no load.</td>
<td>When starting the motor, lighten the load.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Incorrect mounting</td>
<td>Loosen the screw.</td>
<td>Tighten the screw.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Bearing trouble</td>
<td>Check for sound and vibration near the bearing.</td>
<td>Motor replacement (Contact us.)</td>
<td></td>
</tr>
<tr>
<td>Motor runs out of control.</td>
<td>Incorrect motor/driver combination</td>
<td>Check the combination Nos. on the nameplate.</td>
<td>If the combination is incorrect, return the pair to YPC or it's authorized agent.</td>
<td>Page 2</td>
</tr>
<tr>
<td></td>
<td>Inappropriate jumper setting</td>
<td>Inspection</td>
<td>Perform correct jumper setting.</td>
<td>Pages 2 to 9</td>
</tr>
<tr>
<td></td>
<td>Imperfect connection</td>
<td>Check motor/encorder connection.</td>
<td>Refer correctly by referring to the connection diagram.</td>
<td>Pages 10, 11</td>
</tr>
<tr>
<td>Position is displaced.</td>
<td>Incorrect A:B phase and UID-pulse jumper, selection</td>
<td>To be inspected.</td>
<td>Pages 5 to 9</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Command pulse width is not as specified.</td>
<td>Check the command pulse width.</td>
<td>Pages 10, 26</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Feedback pulse rate and receive circuit response speed are not as specified.</td>
<td>Check the feedback pulse rate (3MHz Max.) and receive circuit response speed</td>
<td>Pages 19, 20</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Both ends of the feedback pulse transmission cable should not be connected to the earth.</td>
<td>To be inspected. If so, connect the driver to AGND and the controller to SG.</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

---

Table 7.1 Motor Troubleshooting and Measures (1/2)
7. Troubleshooting and Measures

7.2 List of LED Display

A seven-segment LED is mounted on the front panel of the driver to display the normal/abnormal status of the motor and driver. Display details are as shown in the following tables.

<table>
<thead>
<tr>
<th>LED Display</th>
<th>Status</th>
<th>Display Details with TEST Switch ON (Serial display of 4-digit hexadecimal notation)</th>
<th>Measures/Pages for Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>Normal status</td>
<td>No detail display</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Speed over</td>
<td>No detail display</td>
<td>See pages 28 to 30</td>
<td></td>
</tr>
<tr>
<td>RAM error</td>
<td>Indiscriminate</td>
<td>Reparation required</td>
<td></td>
</tr>
<tr>
<td>Encoder error</td>
<td></td>
<td>See pages 28 to 30</td>
<td></td>
</tr>
<tr>
<td>Encoder error</td>
<td></td>
<td>No detail display</td>
<td></td>
</tr>
<tr>
<td>Error on power supply</td>
<td>Indiscriminate</td>
<td>Reparation required</td>
<td></td>
</tr>
<tr>
<td>Counter overflow</td>
<td>No detail display</td>
<td>See pages 28 to 30</td>
<td></td>
</tr>
<tr>
<td>ROM error</td>
<td>ROM checksum code 4 digits</td>
<td>Reparation required</td>
<td></td>
</tr>
<tr>
<td>Abnormal main power supply</td>
<td>No detail display</td>
<td>See pages 28 to 30</td>
<td></td>
</tr>
<tr>
<td>Power status of driver</td>
<td>No detail display</td>
<td>Release reset</td>
<td></td>
</tr>
<tr>
<td>CPU error</td>
<td></td>
<td>Reparation required</td>
<td></td>
</tr>
<tr>
<td>Amplifier error</td>
<td></td>
<td>See pages 28 to 30</td>
<td></td>
</tr>
<tr>
<td>Overload error</td>
<td>No detail display</td>
<td>See pages 28 to 30</td>
<td></td>
</tr>
</tbody>
</table>

Note: Consult Yokogawa Precision Corp. or its authorized agency when repairation is required.

7.3 Procedure for Error Correction

(1) Encoder Error

- Encoder error
  - Check the encoder cable connection
  - LED detail display
  - Others
  - Power on again

(2) Over Speed

- Speed over
  - Command input correct?
  - YES
  - KM
  - Consult Yokogawa Precision Corp.
  - Power on again
  - NO
  - Correction
7. Troubleshooting and Measures

(3) Over Count

Over count  

- Already tuned?  
  - YES  
  - NO  

- Command input, correct?  
  - YES  
  - NO  

- Mismatch between models of motor and driver?  
  - YES  
  - NO  

- Excessive external force, applied to motor's rotor?  
  - YES  
  - NO  

  Consult Yokogawa Precision Corp.  

- Power on again

(4) Abnormal Main Power Supply

Abnormal main power supply  

- Power cable connection OK?  
  - YES  
  - NO  

  Correction  

- Power voltage, too low?  
  - YES  
  - NO  

  Consult Yokogawa Precision Corp.  

- Power on again

(5) Amplifier Error

Amplifier error  

- Over voltage  
  - LED detail display  
  - Over current  

- Power supply voltage, too high?  
  - YES  
  - NO  

- Motor cable connection, OK?  
  - YES  
  - NO  

- Operation duty, too high?  
  - YES  
  - NO  

  Correction  

- Motor part, grounded?  
  - YES  
  - NO  

  Consult Yokogawa Precision Corp.  

- Power on again

(6) Over Load

Over load  

- Motor's rotor, locked?  
  - YES  
  - NO  

- Operation duty, too high?  
  - YES  
  - NO  

  Power on again  

- Driver installation environment  
  - Without specification ranges  
    - WITHIN SPECIFICATION RANGES  
    - Correction  

- Make duty low, and reduce the load  
  - NO  

  Consult Yokogawa Precision Corp.  

- Power on again

- Power on again
8. OTHERS

8.1 Standard Specification

8.2 Torque vs Velocity Characteristic

(3) Environmental Specification

<table>
<thead>
<tr>
<th></th>
<th>Motor Section</th>
<th>Drive Section</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ambient operating conditions</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Temperature</td>
<td>0 to 40°C</td>
<td>0 to 50°C</td>
<td></td>
</tr>
<tr>
<td>Humidity</td>
<td>20 to 85% R.H</td>
<td>20 to 90% R.H</td>
<td>Non-condensing</td>
</tr>
<tr>
<td>Storage conditions</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Temperature</td>
<td>-20 to 55°C</td>
<td>-30 to 70°C</td>
<td></td>
</tr>
<tr>
<td>Humidity</td>
<td>20 to 65% R.H</td>
<td>20 to 90% R.H</td>
<td>Non-condensing</td>
</tr>
<tr>
<td>Operating environment</td>
<td>No corrosive gases, Dust-free environment</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
8.3 Dimensional Outline Drawing

(1) Motor (A Series)

(2) Motor (B Series)

(3) Driver
8.4 Driver Block Diagram
B.4 Manual: BE-A/B Type Dynamic Brake

Instruction manual, 10 pages
Introduction

Thank you for purchasing our DYNASERV-dedicated dynamic brake. This generation type brake was developed for DYNASERV DD motors and is of simple construction, making it easy to assemble into industrial machinery such as robots with built-in DYNASERV and hence, suitable for a wide range of applications.

This instruction manual describes those items that are considered to be necessary when using this brake so that its functions and usage cautions are fully understood prior to commencing operation.

Cautions

- Copying of part or all of the contents of this manual is strictly prohibited.
- The contents of this manual may be subject to change without notice.
- This manual is prepared carefully, but if any mistakes and/or omissions are found, please contact our sales or service representative immediately.
- Any damage or indirect damage due to our unintentional mistakes as a result of operation in accordance with this instruction manual may not our responsibility.
This negative action type electric power generation brake was developed especially for DYNASERV and has the following features.

- Simple construction ... No mechanical devices are required in the motor section, and efficient torque control is achieved simply by connecting this brake between the motor and driver sections.
- Power failure compensation ... Even during power failure, the same control torque as that at power-ON is available, through the use of a built-in power failure compensator.
- Both velocity change and capacitor types are available. Both velocity change and capacitor versions of this brake are available. The former is suitable for either high or low velocity applications while the latter is for use in the high-velocity area, making the model range suitable for a wide range of applications.
- Maintenance free

2. Operational Cautions

(1) Because the brake was designed especially for use with DYNASERV, it may not display its specified performance when used with other motors.

(2) When the brake has been activated, do not attempt to rotate the stopped motor by force, as doing so may overheat the internal resistor.

(3) When coupling this brake to a DYNASERV, do not mistake the connection of the A, B and C phases and GND, especially in the motor and driver sections, as doing so may stop DYNASERV from operating normally.

(4) When this brake is operated repeatedly, operate it at minimum intervals of 1 minute, otherwise the internal resistor may overheat.

(5) Both 100 V AC and 200 V AC power supply voltages are available. Always check to make sure that the correct one is being used.
(6) When installing the brake board, separate it from the case by more than 10 mm or keep it away from other boards more than 90 mm when the other board is laid on top of the parts installation side of the brake board. (See the figure below.) Also, if necessary, install a shielded plate as shown in the following figure.

### 3. Specifications

#### 3.1 Common Item and Environmental

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Structure</td>
<td>Printed board type</td>
</tr>
<tr>
<td>Dimensions (L x W x H) [mm]</td>
<td>210 x 170 x 55 (Max.)</td>
</tr>
<tr>
<td>Weight (g)</td>
<td>1,000</td>
</tr>
<tr>
<td>Operating Temperature [°C]</td>
<td>0~50</td>
</tr>
<tr>
<td>Operating Humidity [%]</td>
<td>20~90RH (Non condensing)</td>
</tr>
<tr>
<td>Storage Temperature [°C]</td>
<td>-20~85</td>
</tr>
<tr>
<td>Storage Humidity [%]</td>
<td>20~90RH (Non condensing)</td>
</tr>
<tr>
<td>Atmosphere</td>
<td>No corrosive gases. Dust-free atmosphere</td>
</tr>
<tr>
<td>Power source</td>
<td>100/200VAC+10%~15% 50/60Hz</td>
</tr>
<tr>
<td>Power consumption [W]</td>
<td>10 (Max.)</td>
</tr>
</tbody>
</table>

#### 3.2 Braking specification

<table>
<thead>
<tr>
<th>Model</th>
<th>Applied DYNASERV Model</th>
<th>Load cond. (Jn x 30) [kg-m²]</th>
<th>Speed cond. (Max.) [rps]</th>
<th>Braking angle [°] (Typical)</th>
<th>Speed changing type</th>
<th>Capacitor type</th>
</tr>
</thead>
<tbody>
<tr>
<td>BE1015B</td>
<td>DM1015B</td>
<td>0.372</td>
<td>2.4</td>
<td>356</td>
<td>J,</td>
<td>356</td>
</tr>
<tr>
<td></td>
<td>DR1015B</td>
<td>0.451</td>
<td>3.0</td>
<td>278</td>
<td>J</td>
<td>278</td>
</tr>
<tr>
<td>BE1030B</td>
<td>DM1030B</td>
<td>0.465</td>
<td>2.4</td>
<td>278</td>
<td>J</td>
<td>278</td>
</tr>
<tr>
<td></td>
<td>DR1030B</td>
<td>0.546</td>
<td>3.0</td>
<td>247</td>
<td>J</td>
<td>247</td>
</tr>
<tr>
<td>BE1045B</td>
<td>DM1045B</td>
<td>0.589</td>
<td>2.4</td>
<td>418</td>
<td>J</td>
<td>418</td>
</tr>
<tr>
<td></td>
<td>DR1045B</td>
<td>0.606</td>
<td>2.4</td>
<td>216</td>
<td>J</td>
<td>216</td>
</tr>
<tr>
<td>BE1060B</td>
<td>DM1060B</td>
<td>0.650</td>
<td>2.4</td>
<td>231</td>
<td>J</td>
<td>231</td>
</tr>
<tr>
<td></td>
<td>DR1060B</td>
<td>0.650</td>
<td>2.4</td>
<td>369</td>
<td>J</td>
<td>369</td>
</tr>
<tr>
<td>BE1075B</td>
<td>DM1075B</td>
<td>0.597</td>
<td>2.4</td>
<td>216</td>
<td>J</td>
<td>216</td>
</tr>
<tr>
<td></td>
<td>DR1075B</td>
<td>0.670</td>
<td>2.4</td>
<td>231</td>
<td>J</td>
<td>231</td>
</tr>
<tr>
<td>BE1050A</td>
<td>DM1050A</td>
<td>2.087</td>
<td>1.2</td>
<td>226</td>
<td>J</td>
<td>226</td>
</tr>
<tr>
<td></td>
<td>DR1050A</td>
<td>5.540</td>
<td>1.8</td>
<td>665</td>
<td>J</td>
<td>665</td>
</tr>
<tr>
<td>BE1100A</td>
<td>DM1100A</td>
<td>6.698</td>
<td>1.2</td>
<td>158</td>
<td>J</td>
<td>158</td>
</tr>
<tr>
<td></td>
<td>DR1100A</td>
<td>6.200</td>
<td>1.2</td>
<td>256</td>
<td>J</td>
<td>256</td>
</tr>
<tr>
<td>BE1150A</td>
<td>DM1150A</td>
<td>4.402</td>
<td>1.2</td>
<td>244</td>
<td>J</td>
<td>244</td>
</tr>
<tr>
<td></td>
<td>DR1150A</td>
<td>7.130</td>
<td>1.8</td>
<td>157</td>
<td>J</td>
<td>157</td>
</tr>
<tr>
<td>BE1200A</td>
<td>DM1200A</td>
<td>5.177</td>
<td>1.2</td>
<td>244</td>
<td>J</td>
<td>244</td>
</tr>
<tr>
<td></td>
<td>DR1200A</td>
<td>8.835</td>
<td>1.8</td>
<td>157</td>
<td>J</td>
<td>157</td>
</tr>
<tr>
<td>BE1300A</td>
<td>DR1300A</td>
<td>10.540</td>
<td>1.0</td>
<td>304</td>
<td>J</td>
<td>304</td>
</tr>
<tr>
<td>BE1400A</td>
<td>DR1400A</td>
<td>12.400</td>
<td>0.8</td>
<td>96</td>
<td>J</td>
<td>96</td>
</tr>
<tr>
<td>BE170E</td>
<td>DR170E</td>
<td>2.535</td>
<td>2.4</td>
<td>60</td>
<td>J</td>
<td>60</td>
</tr>
<tr>
<td>BE100E</td>
<td>DR100E</td>
<td>3.100</td>
<td>2.4</td>
<td>627</td>
<td>J</td>
<td>627</td>
</tr>
<tr>
<td>BE130E</td>
<td>DR130E</td>
<td>3.675</td>
<td>1.2</td>
<td>174</td>
<td>J</td>
<td>174</td>
</tr>
<tr>
<td>BE160E</td>
<td>DR160E</td>
<td>4.340</td>
<td>1.2</td>
<td>174</td>
<td>J</td>
<td>174</td>
</tr>
<tr>
<td>BE220E</td>
<td>DR220E</td>
<td>5.270</td>
<td>1.2</td>
<td>157</td>
<td>J</td>
<td>157</td>
</tr>
<tr>
<td>BE250E</td>
<td>DR250E</td>
<td>5.735</td>
<td>1.2</td>
<td>145</td>
<td>J</td>
<td>145</td>
</tr>
</tbody>
</table>

(Note) Jn : Rotor inertia of DYNASERV
4. Model No.

Product Model No. has the following meaning.

<table>
<thead>
<tr>
<th>Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Power supply voltage 100 V AC</td>
</tr>
<tr>
<td>2</td>
<td>Power supply voltage 200 V AC</td>
</tr>
<tr>
<td>V</td>
<td>Velocity changing type</td>
</tr>
<tr>
<td>C</td>
<td>Capacitor type</td>
</tr>
<tr>
<td></td>
<td>Motor series name (Alphabet)</td>
</tr>
<tr>
<td></td>
<td>4 digits corresponding to motor type</td>
</tr>
<tr>
<td></td>
<td>Electronic brake</td>
</tr>
</tbody>
</table>

Example: BE1200A-V2
 Electronic brake
 Corresponding to the 1200 motor type
 A Series
 Velocity changing type
 Power supply voltage: 200 V AC

5. Product Configuration

This brake consists of the following device and accessories when it is purchased.

<table>
<thead>
<tr>
<th>Name</th>
<th>Q'ty</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mainframe</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>Standard</td>
<td></td>
<td></td>
</tr>
<tr>
<td>accessories</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Connector</td>
<td>1</td>
<td>Type No. 5051-04 (Made by MOLEX)</td>
</tr>
<tr>
<td>Terminals</td>
<td>4</td>
<td>Type No. 5150TL (Made by MOLEX)</td>
</tr>
</tbody>
</table>

6. Dimensional Outline Drawing and Mounting Diagram (Unit: mm)

Figure 6-1 Dimensional Outline Drawing of Dynamic Brake

Figure 6-2 Mounting Diagram
7. Interface

Figure 7-1 Interface Connection Diagram

<table>
<thead>
<tr>
<th>Connector No.</th>
<th>Signal Name</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>CH1-1</td>
<td>VCM</td>
<td>Motor C-phase</td>
</tr>
<tr>
<td>2</td>
<td>VDE</td>
<td>Motor D-phase</td>
</tr>
<tr>
<td>3</td>
<td>VAM</td>
<td>Motor A-phase</td>
</tr>
<tr>
<td>4</td>
<td>FG</td>
<td>Frame grounding</td>
</tr>
<tr>
<td>5</td>
<td>VCD</td>
<td>Driver C-phase</td>
</tr>
<tr>
<td>6</td>
<td>VBD</td>
<td>Driver B-phase</td>
</tr>
<tr>
<td>7</td>
<td>VAD</td>
<td>Driver A-phase</td>
</tr>
<tr>
<td>8</td>
<td>FG</td>
<td>Frame grounding</td>
</tr>
<tr>
<td>9</td>
<td>GND</td>
<td>Grounding</td>
</tr>
<tr>
<td>10</td>
<td>AC</td>
<td>AC input</td>
</tr>
<tr>
<td>11</td>
<td>AC</td>
<td>AC input</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Connector No.</th>
<th>Signal Name</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>CN2-1</td>
<td>Vcc</td>
<td>Power supply voltage</td>
</tr>
<tr>
<td>2</td>
<td>BRON</td>
<td>Brake ON</td>
</tr>
<tr>
<td>3</td>
<td>Vcc</td>
<td>Power supply voltage</td>
</tr>
<tr>
<td>4</td>
<td>SERVO</td>
<td>Servo ON</td>
</tr>
</tbody>
</table>

Vcc = +5V to +12V
8. Wiring

Notes: Cables used for wiring related to the brake in the above wiring diagram should be as follows.

1. Same specifications as those of the power cable from the motor: Current capacity 20 A (A Series), or 15 A (B series)
   Cable size: HIV 2.0 mm² or more
   Length: less than 30 m (10 + 20)
2. Same as in 1
3. Current capacity: More than 100 mA DC
   Twisted pair collectively shielded wire (core cross section: More than 0.2 mm², zinc plated, twisted soft copper wire)
   Length: Less than 30 m
4. Current capacity: More than 1A AC

To avoid miss operation by electrical noise, wiring should take care of as follows:
(1) To insert surge current absorb circuit, when used solenoid, relay, and other magnetic switch on line or closed.
(2) To insert noise filter on power source line and input signal line, when existing high frequency noise generated source on line or closed.

Figure 8-1 Wiring Diagram
9. Circuit Configuration and Operation

(1) Circuit configuration

![Circuit Configuration Block Diagram](image)

Note: In the capacitor type circuit configuration, there is no "Relay-2", but the motor output is directly connected to the capacitor circuit from "Relay-1".

(2) Operation

- Velocity changing type
  
  Both high and low-velocity types are effective. Greater braking torque is obtained with the coil shorted than that of the capacitor type at low velocity. It is possible automatically to select the capacitor mode and the short mode by detecting the velocity. If the <BRON> and <SV> signal from the user unit is set to "H" or the power supply is suspended, "Relay-1" is turned OFF and the motor is connected to the capacitor mode of "Relay-2". Next, as shown in Figure 9-2, braking torque in the capacitor mode becomes smaller than in the short mode, so "Relay-2" is turned OFF to short the motor.

- Capacitor type
  
  This is effective in the high-velocity region. The effect of motor coil inductance becomes large at high velocity, and therefore it is restricted by a capacitor which converts rotating energy to thermal energy. Operation is the same as that of the velocity changing type, but no short mode is available.

![Generated Torque Diagram](image)

From the above, when selecting the brake, refer to the following table.

<table>
<thead>
<tr>
<th>Motor rotation speed [rps]</th>
<th>Braking</th>
</tr>
</thead>
<tbody>
<tr>
<td>A Series</td>
<td>B Series</td>
</tr>
<tr>
<td>0 to 1.2</td>
<td>0 to 2.4</td>
</tr>
<tr>
<td>0.25 to 1.2</td>
<td>0.4 to 2.4</td>
</tr>
<tr>
<td>0 to 0.25</td>
<td>0 to 0.4</td>
</tr>
</tbody>
</table>

(3) Interface circuit configuration

The CN2 terminal on this brake consists of a photocoupler as shown in the following diagram. Inputting <VCC> activates the circuit, but the "H" or "L" input signal in this case has the following meanings.

- "H": <VCC> level voltage (VCC = +5 V to +12 V)
- "L": <GND> level voltage

Note: Other pins @ and @ on the CN2 terminal have the same circuit configuration.
1. Operation Procedure

(1) Preparation
   Turn OFF the driver power of DYNASERV in use.
   Remove the power cable from the motor.

(2) Connection
   Connect and wire the brake between the motor and the
   DYNASERV driver in accordance with the connection diagram.

(3) Operation procedure
   ① Prior to turning ON the DYNASERV power, turn ON the
      brake power.
   ② Disconnect the connection of the brake CN2 terminal, or
      set the <BRON> and <SRVON> signal to "H".
   ③ Check to see if the brake is activated in this status.
   ④ Next, connect the brake's CN2 terminal, then set the
      <BRON> and <SRVON> signals to "L".
   ⑤ Check to see if the brake is released in this status.
   ⑥ Turn ON the DYNASERV's drier power.
      Set the <SRVON> signal to "L" and the mode to the test
      mode, then check to see if the brake is activated nor-
      mally.
   ⑦ If the above operation causes no abnormality, the brake is
      operating normally.

(4) Signal and operation status

<table>
<thead>
<tr>
<th>&lt;BRON&gt;</th>
<th>&lt;SRVON&gt;</th>
<th>&lt;Relay 1&gt;</th>
<th>Connection with motor</th>
<th>Brake power</th>
</tr>
</thead>
<tbody>
<tr>
<td>-</td>
<td>-</td>
<td>OFF</td>
<td>Short</td>
<td>OFF</td>
</tr>
<tr>
<td>L</td>
<td>L</td>
<td>ON</td>
<td>Driver</td>
<td>ON</td>
</tr>
<tr>
<td>L</td>
<td>H</td>
<td>ON</td>
<td>Driver</td>
<td>ON*</td>
</tr>
<tr>
<td>H</td>
<td>H</td>
<td>OFF</td>
<td>Capacitor. SHORT</td>
<td>ON**</td>
</tr>
</tbody>
</table>

Note: *: In this status, a robot can be programmed directly.
**: When the power is turned from ON to OFF, this
   means a power failure has occurred, but operation
   is the same as in the power-ON status due to power
   failure compensation.

(Note) The relay chatters for few second only after the power source is turned off.
However this is not trouble by back up capacity of power circuit capacitor.
## 11. Trouble and Measures

<table>
<thead>
<tr>
<th>Trouble</th>
<th>Probable cause</th>
<th>Inspecting items</th>
<th>Measures</th>
</tr>
</thead>
<tbody>
<tr>
<td>The motor does not start</td>
<td>Incorrect connection</td>
<td>Check the connection of the motor's A, B and C phases and GND.</td>
<td>Make sure that the wiring is correct by referring to the wiring diagram.</td>
</tr>
<tr>
<td></td>
<td>Incorrect power</td>
<td>Check the wiring.</td>
<td>Make sure that the wiring is correct by referring to the wiring diagram.</td>
</tr>
<tr>
<td></td>
<td>supply wiring</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Fuse burnt out</td>
<td>Check the fuse.</td>
<td>Replace the board if the fuse is burnt out.</td>
</tr>
<tr>
<td></td>
<td>in the brake circuit</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Motor rotation is unstable</td>
<td>Incorrect connection</td>
<td>Check the connection of the motor's A, B and C phases and GND.</td>
<td>Repair the faulty section.</td>
</tr>
<tr>
<td>The brake is not activated</td>
<td>Incorrect signal wiring</td>
<td>Check the wiring.</td>
<td>Make sure that the wiring is correct by referring to the wiring diagram.</td>
</tr>
</tbody>
</table>

Note: If it is assumed that the brake is faulty, the board itself should be replaced. Therefore, do not replace or repair any parts on the board.

If the brake fails or is damaged in the normal operating status due to defect attributed to faulty manufacture, within one year of the date of purchase, it will be repaired free of charge.
B.5 Cabinet manual

Instruction manual, 10 pages
GEMEENSCHAPPELIJKE TECHNISCHE DIENST

GROEP ELEKTRONICA / ENERGIETECHNIEK

ONDERWERP : Dynaserv Actuators

OPDRACHTGEVER : J.J.F.J. Garenfeld

TELEFOON : 2824

ONTWERP :

TELEFOON :

REALISATIE : B. Viveen

DOCUMENTATIE : B. Viveen

INHOUD :
**KLANT**

**PROJEKTNAMEN** : SERVO-SYSTEEM

**TEKENINGNUMMER** : GTD E/E

**OPDRACHTGEVER** :

<table>
<thead>
<tr>
<th>Fabrikant (firma)</th>
<th>Pad (zonder \EPLAN4\P)</th>
<th>Hoogste paginanummer</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>3506/001/23/2301</td>
<td>4</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Projektname</th>
<th>Fabrikaat</th>
<th>Type</th>
<th>Installatieplaats</th>
<th>Projektleider</th>
<th>Bijzonderheden</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Vervaardigd op</th>
<th>Hoogste paginanummer</th>
</tr>
</thead>
<tbody>
<tr>
<td>27. NOV. 1997</td>
<td>4</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Laatste wijziging</th>
<th>Aantal pagina's</th>
</tr>
</thead>
<tbody>
<tr>
<td>27. NOV. 1997</td>
<td>4</td>
</tr>
</tbody>
</table>
VOEDING
3×F+N+PE

ZIE OVERZICHT-SCHAHEMA

SERVO-SYSTEEM

HOOFDSTROOFSCHAHEMA
<table>
<thead>
<tr>
<th>Fabrikant</th>
<th>Artikelnummer</th>
<th>Bestelnummer</th>
<th>Omschrijving</th>
<th>P/a/p</th>
<th>Aantal</th>
<th>FL/eenh. excl. BTW</th>
<th>Korting %</th>
<th>BTW 17,5 %</th>
<th>Totaal</th>
</tr>
</thead>
<tbody>
<tr>
<td>Klockner Moel</td>
<td>TO-2-0293/E</td>
<td>688 820</td>
<td>Hoofdsch. 3 polig + nul front 20A</td>
<td>91</td>
<td>2.3</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>91</td>
<td>2.4</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Telemecanique</td>
<td>GB2-C006</td>
<td>645 226</td>
<td>Stuurstrom automaat 1A</td>
<td>FOB</td>
<td>2.6</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Klockner Moel</td>
<td>GODL-11/10</td>
<td>699 546</td>
<td>Aan uit drukker met lampje</td>
<td>SI</td>
<td>2.6</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Klockner Moel</td>
<td>BK11</td>
<td>698 647</td>
<td>Drukknop element 1mAak 1verbreek</td>
<td>SI</td>
<td>2.6</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Klockner Moel</td>
<td>EF</td>
<td>698 951</td>
<td>Lamphouder BK element</td>
<td>SI</td>
<td>2.6</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Klockner Moel</td>
<td>GIL 220K</td>
<td>469 296</td>
<td>Lampje 220V</td>
<td>SI</td>
<td>2.6</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Klockner Moel</td>
<td>APV</td>
<td>698 852</td>
<td>Nood uit knop ontgrendeling door trekken</td>
<td>SI</td>
<td>2.6</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Klockner Moel</td>
<td>BK11</td>
<td>698 647</td>
<td>Drukknop element 1mAak 1verbreek</td>
<td>SI</td>
<td>2.6</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Klockner Moel</td>
<td>FAK-R/T/V/KC11/1</td>
<td>669 184</td>
<td>Nood uit ext. ontgrendeling door trekken</td>
<td>SI</td>
<td>2.6</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Paraaf besteller... Paraaf bedrijfsbureau... Totaal FL...
<table>
<thead>
<tr>
<th>Fabrikant</th>
<th>Artikelnummer</th>
<th>Bestelnummer</th>
<th>Omschrijving</th>
<th>Honderdpaalcode</th>
<th>Aantal</th>
<th>FL/eeh excl. BTW</th>
<th>Korting %</th>
<th>Korting FL</th>
<th>BTW 17.5 %</th>
<th>Totaal</th>
</tr>
</thead>
<tbody>
<tr>
<td>Klockner Moel</td>
<td>OIL 00 AM</td>
<td>697 623</td>
<td>hoofdmagn. schak. 3P/5, 5KW/spoelsp.220V</td>
<td>KM</td>
<td>2.6</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Klockner Moel</td>
<td>11 DIL M</td>
<td>697 821</td>
<td>hulpcontacten DIL 0.1.2 M 1M1V</td>
<td>KM</td>
<td>2.7</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>A1</td>
<td>2.13</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>M1</td>
<td>2.13</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>A2</td>
<td>2.15</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>M2</td>
<td>2.15</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>A3</td>
<td>2.17</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>M3</td>
<td>2.17</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Paraaf Besteller:...........      Paraaf Bedrijfsbureau:...........  Totaal FL:...........
### Aansluiting 50 pol. Amphenol connector

<table>
<thead>
<tr>
<th>Nummer</th>
<th>Kleur</th>
<th>Nummer</th>
<th>Kleur</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>ROOD</td>
<td>27</td>
<td>ROOD/BLAUW</td>
</tr>
<tr>
<td>2</td>
<td>BLAUW</td>
<td>28</td>
<td>GRIJS/ROSE</td>
</tr>
<tr>
<td>3</td>
<td>ROSE</td>
<td>29</td>
<td>ZWART</td>
</tr>
<tr>
<td>4</td>
<td>GRIJS</td>
<td>30</td>
<td>VIOLET</td>
</tr>
<tr>
<td>5</td>
<td>GEEL</td>
<td>31</td>
<td>WIT</td>
</tr>
<tr>
<td>6</td>
<td>GROEN</td>
<td>32</td>
<td>BRUIN</td>
</tr>
<tr>
<td>7</td>
<td>BRUIN</td>
<td>33</td>
<td>ROSE</td>
</tr>
<tr>
<td>8</td>
<td>WIT</td>
<td>34</td>
<td>GRIJS</td>
</tr>
<tr>
<td>9</td>
<td>ROSE/GEEL</td>
<td>35</td>
<td>GEEL</td>
</tr>
<tr>
<td>10</td>
<td>ROSE/GROEN</td>
<td>36</td>
<td>GROEN</td>
</tr>
<tr>
<td>11</td>
<td>GRIJS/GEEL</td>
<td>37</td>
<td>ZWART/ROOD</td>
</tr>
<tr>
<td>12</td>
<td>GRIJS/GROEN</td>
<td>38</td>
<td>ZWART/BLAUW</td>
</tr>
<tr>
<td>13</td>
<td>ZWART/WIT</td>
<td>39</td>
<td>ROOD/GRIJS</td>
</tr>
<tr>
<td>14</td>
<td>ZWART/BRUIN</td>
<td>40</td>
<td>ROOD/ROSE</td>
</tr>
<tr>
<td>15</td>
<td>ROOD/WIT</td>
<td>41</td>
<td>ZWART/ROSE</td>
</tr>
<tr>
<td>16</td>
<td>ROOD/BRUIN</td>
<td>42</td>
<td>ZWART/GRIJS</td>
</tr>
<tr>
<td>17</td>
<td>BLAUW/WIT</td>
<td>43</td>
<td>ZWART/GEEL</td>
</tr>
<tr>
<td>18</td>
<td>BLAUW/BRUIN (afscherming)</td>
<td>44</td>
<td>BLAUW/GRIJS</td>
</tr>
<tr>
<td>19</td>
<td>ROSE/WIT</td>
<td>45</td>
<td>ZWART/GEEL</td>
</tr>
<tr>
<td>20</td>
<td>ROSE/BRUIN</td>
<td>46</td>
<td>ZWART/GROEN</td>
</tr>
<tr>
<td>21</td>
<td>GRIJS/WIT</td>
<td>47</td>
<td>ROOD/GEEL</td>
</tr>
<tr>
<td>22</td>
<td>GRIJS/BRUIN</td>
<td>48</td>
<td>ROOD/GROEN</td>
</tr>
<tr>
<td>23</td>
<td>GEEL/WIT</td>
<td>49</td>
<td>BLAUW/GEEL</td>
</tr>
<tr>
<td>24</td>
<td>GEEL/BRUIN</td>
<td>50</td>
<td>BLAUW/GROEN</td>
</tr>
<tr>
<td>25</td>
<td>GROEN/WIT</td>
<td></td>
<td></td>
</tr>
<tr>
<td>26</td>
<td>GROEN/BRUIN</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
B.6 Power sliprings

SM 140-2

Features:
- Heavy Duty design for industrial use
- Long life time
- 2/4/6/12 rings
- max. 15 Amps. per ring
- custom-made versions available

Technical Data:

Dimensions: see drawing
No. of rings: 2/4/6/12
Max. operating current: 15 A (at 50 V DC)
Operating voltage: nom. 50 V DC
max. 220 V DC* (the max. value is dependant on the actual operating current)
Circuit resistance: \( \leq 0.1 \) ohm
Insulation resistance at 500 V DC: \( 10^6 \) M
Dielectric strength: 1000 V\text{eff} (60s)
Rotation speed: max. 250 rpm
Operating temp.: 0 to 65°C
Protection: rotor: IP 00
terminals: IP 00

* If different data is required please use our specification sheet!
* according to insulation group of "VDE"
1) others on request
B.7 Signal sliprings

<table>
<thead>
<tr>
<th>NO. OF RINGS</th>
<th>A</th>
<th>B</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>.85</td>
<td>1.75</td>
</tr>
<tr>
<td>12</td>
<td>1.65</td>
<td>2.50</td>
</tr>
</tbody>
</table>
B.8 Conceptual design

The RRR-robot conceptual design drawing in this Section was used as a basis for the manufacturing of the various non-standard components by the CTD. Note that some dimensions were slightly altered during the manufacturing.