

Force Sensor User Guide

UMI-R3-510



Force Sensor User Guide

Rev.	Revision History	Date
001	Original Issue	96-06
002	F3, Rapl-3 Upgrade	98-06
003	C500c update	99-06
004	General update	99-10
004a	CSA certification information added	99-12

Copyright © 2000 CRS Robotics Corporation

RAPL-3, RAPL-II, and RAPL are trademarks of CRS Robotics Corporation and may be used to describe only CRS Robotics products.

All brand names and product names used in this guide are trademarks, registered trademarks, or trade names of their respective holders.

The information in this document is subject to change without notice.

CRS Robotics Corporation makes no warranty of any kind with regard to this material, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. CRS Robotics Corporation assumes no responsibility for any errors that may appear in this document. CRS Robotics Corporation makes no commitment to update nor to keep current the information contained in this document.

CRS Robotics Corporation software products shall remain the property of CRS Robotics Corporation.

Additional copies of this guide, or other CRS Robotics literature, may be obtained from the Sales Department or from your distributor.

P R E F A C E

About This Guide

This user guide accompanies the force sensor option for CRS robot systems. It contains general information, specifications, installation instructions, startup procedures, and basic operation instructions for the force sensor. It is designed to complement the more general manufacturer's guide for the ATI Industrial Automation F/T Sensor.

Before using or installing the force sensor, you should be familiar with all safety procedures contained in the manufacturer's guide for the ATI Industrial Automation F/T Sensor. This ATI guide is included as part of the force sensor kit.

Who Uses This Guide

This installation guide is intended for those who have attended a CRS Robotics robot system training course and already have some basic familiarity with RAPL-3 and Robcomm3.

How to Use This Guide



Throughout this user's guide warnings are marked by a "!" symbol in the left margin. Failure to comply with these warnings can result in system errors, memory loss, or damage to the robot and its surroundings

This guide is task based and uses navigational aids to help you quickly find the topics and information you need. If a technical term is not familiar to you, refer to the Glossary.

Before attempting to follow instructions or examples in a section, read the entire section first.

This guide consists of the following chapters:

- **Introducing the Force Sensor** describes the force sensor and the available operating modes.
- **Installing the Force Sensor** explains how to install the force sensor and set up communication between the force sensor and your PC.
- **RAPL-3 Library Commands** defines all available force sensor commands in the RAPL-3 library.
- **Troubleshooting** provides tips for correcting common problems with your force sensor configuration.

Throughout this manual, Imperial units are used as the standard units of measurement, followed by the metric equivalent in brackets. For example, "The ATI sensor can sense forces of up to 15 pounds (67 Newton's)..."

For More Information

Additional information is available in the following documents

- Installation and Operations Manual for Stand-Alone F/T Sensor Systems (*ATI Industrial Automation*)
- F3 Robot Arm for C500C Controller User Guide (*UMI-R3-310*)
- A465 Robot Arm User Guide for C500C Controller (*UMI-33-465-A*)
- A255 Robot Arm User Guide for C500C Controller (*UMI-33-255-A*)
- C500C Controller User's Guide (*UMI-33-CON*)
- RAPL-3 Programming Manual (*UMI-R3-210*)
- Robcomm3 User Guide (*UMI-UM-RC1*)
- Robcomm3 for Windows Reference Manual (*UMI-23-504*)

Additional copies of this manual, or other CRS Robotics literature, may be obtained from the Customer Support Group.

Training courses are offered at our facility in Burlington, Ontario, Canada, or can be conducted at your facility. For additional information contact the Customer Support Group.

CRS Robotics Corporation
5344 John Lucas Drive
Burlington, Ontario
L7L 6A6
Canada

Telephone: (905) 332-2000
Facsimile: (905) 332-1114

Contents

CHAPTER 1	1
Introducing the Force Sensor	1
The Force Sensor	2
Range of Measurable Forces and Torques	3
Operating Modes	3
CHAPTER 2	5
Installing the Force Sensor	5
Installation Requirements	6
Installing the Force Sensor	7
Testing the Force Sensor	10
Calibrating the Force Sensor	11
CHAPTER 3	13
RAPL-3 Library Commands	13
Operating commands	14
Operating Parameters	19
Detailed Examples	23
CHAPTER 4	31
Troubleshooting	31
Incorrect or Inconsistent Force Data	32
Inaccurate Sensor Readings	32
Application Cannot Communicate With Sensor	33
APPENDIX A	35
Sensor Electronics Configuration	35
APPENDIX B	37
Force Sensor Specifications	37
Glossary	G-1
Index	I-1

Introducing the Force Sensor

This chapter introduces the force sensor for CRS Robotics A255, A465, and F3 arms, and discusses the operating modes available. It contains the following sections:

- The Force Sensor
- Range of Measurable Forces and Torques
- Operating Modes

The Force Sensor

The force sensor is an arm option for CRS Robotics A255, A465, and F3 robot systems. It enables a robot system to detect forces and torques when the robot system is in contact with an object. When used with an A465 or F3 arm, the sensor can be additionally configured to provide *force control* information to the robot system. This information allows the trajectory planner to move the arm to actively compensate for applied forces and torques.



Figure 1: Typical force sensor (ATI Industrial Automation F/T sensor)

How the Force Sensor Works

The force sensor mounts directly on the tool flange of the arm and has a flange for attaching end effectors. When an attached end effector comes into contact with an object, pressure-sensitive devices inside the force sensor measure the applied forces and torques, and transfer this information to the controller. The force sensor controller digitizes the signal and transforms it into the forces and torques which are sent to the controller.

The force sensor is controlled from a computer via RAPL-3 library commands.

Range of Measurable Forces and Torques

The force sensor detects forces and torques in six degrees of freedom, i.e. about the X, Y, and Z axes of the sensor. It is sensitive within the ranges shown in the following table:

Table 1: *Measurable Forces and Torques*

Detected Property	Range of Sensitivity	
Force (X-direction)	0 to 15 pounds	0 to 67 Newton's
Force (Y-direction)	0 to 15 pounds	0 to 67 Newton's
Force (Z-direction)	0 to 50 pounds	0 to 222 Newton's
Torque (X-direction)	0 to 50 inch-pounds	0 to 5650 Newton-mm
Torque (Y-direction)	0 to 50 inch-pounds	0 to 5650 Newton-mm
Torque (Z-direction)	0 to 50 inch-pounds	0 to 5650 Newton-mm

Operating Modes

The force sensor operates in two modes: *force sampling* mode and *force control* mode.

Force sampling mode

In force sampling mode, the sensor provides information to the robot controller, but plays no part in controlling the action of the robot arm. Force sampling information is useful in applications such as measuring part weights, or detecting overload conditions.

Note: When the force sensor is used with an A255 arm, this is the only available operating mode.

Force control mode

In force control mode, the sensor provides information critical to the trajectory planner software that moves the robot arm. This information is used to actively control exerted force or maintain positional control of the arm. Force control mode is available on the A465 and F3 robot arms only.

Installing the Force Sensor

Before installing the force sensor, ensure that you are familiar with the manufacturer's instructions that came with the ATI Industrial Automation F/T sensor. Review all safety procedures relating to the installation process.

The following sections describe how to install, test, and calibrate the force sensor.

- Installation Requirements
- Installing the Force Sensor
- Testing the Force Sensor
- Calibrating the Force Sensor

Installation Requirements

To install the force sensor, you need a robot system which is already installed and configured. The robot system must contain the following components:

- F3, A465, or A255 arm
- C500C controller
- Computer terminal

Consult the guides that came with your robot system for installation and configuration instructions.

Required Software

Verify that the following software components are installed on your robot system:

- Robcomm3 for Windows
- RAPL-3, version 2.4.1113 or higher

Required Tools

You will need the following tools to install the force sensor:

- Metric and Imperial Hex (Allen) keys
- Philips screwdriver

Force Sensor Kit

The force sensor kit includes the following:

- Installation and Operations Manual for Stand-Alone F/T Sensor Systems
- Force sensor (also called an *F/T transducer*)
- Sensor controller (also called a *stand-alone F/T controller*)
- Sensor controller cable
- Power cable
- 9-pin null modem cable
- Screws and servogripper flange in a small plastic bag
- Floppy disk containing RAPL-3 force sensor utilities

If any parts are missing, please contact your distributor.

Installing the Force Sensor

Before installing the force sensor, you should familiarize yourself with the safety procedures in the manufacturer's instructions that came with the ATI Industrial Automation F/T sensor. You should also verify that:

- The arm is mounted on a level surface and is properly calibrated for gravity compensation and force tracking.
- You can easily reach the tool flange and upper arm.
- End effectors are removed from the tool flange.

Aligning the Tool Flange

Before installing the force sensor, you must align the tool flange by placing the arm in the ready position.

To align the A255 or A465 tool flange:

- 1 Turn on the controller.
- 2 Turn on power to the arm.
- 3 Home the arm.
- 4 Move the arm to its ready position. Ensure that the tool flange is aligned so that its keyway lies horizontally and joint #6 is in the middle of its travel. If necessary, move the arm with the teach pendant.
- 5 Turn off power to the arm.
- 6 Turn off power to the controller.

The tool flange is now in the correct position for mounting the force sensor.

To align the F3 tool flange:

- 1 Turn on the controller.
- 2 Turn on power to the arm.
- 3 Move the arm to the ready position.
- 4 Turn off power to the arm.
- 5 Turn off power to the controller.

The tool flange is now in the correct position for mounting the force sensor.

Mounting the Force Sensor

When mounting the force sensor, take care not to damage the internal electronics which are exposed.

To mount the force sensor:

- 1 Remove the sensor's rear plate by removing the eight M3 flat head screws on the rear plate of the sensor.
- 2 Carefully align the plate to the tool flange's 3 mm pin hole.

Note: If the arm is in the ready position, the plate's exposed pin should be uppermost. If the arm is upside down, the pin is low on the flange.

- 3 Attach the sensor plate to the tool flange, using four 10-24 flat head cap screws.
- 4 Re-attach the sensor body to its rear plate. Note the pin hole orientation. Tighten the eight screws securely.

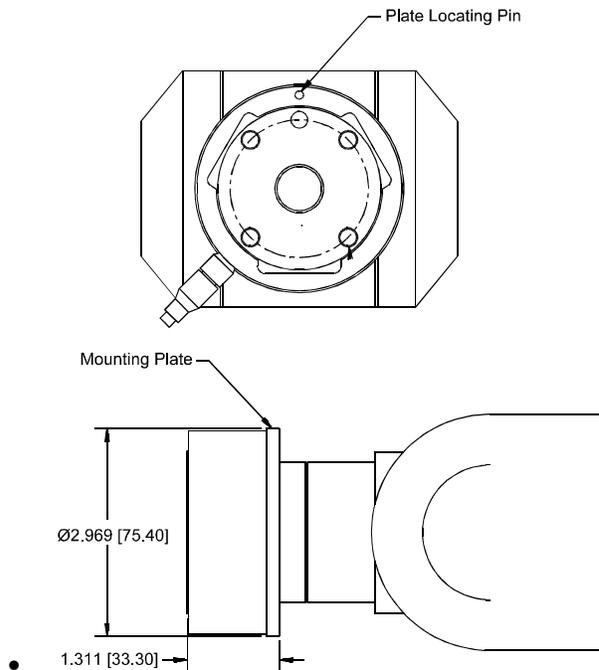


Figure 2: ATI FT 15/50 Force sensor mounted on arm

Connecting the Force Sensor Cables

Cables are used to connect the force sensor to the rest of the robot system. Always make sure that there is enough extra cable to allow for the full range of robot motion.

To connect the cables:

- 1 Place the force sensor controller near the C500C controller.
- 2 Plug the AC power cable into the force sensor controller.
- 3 Using the sensor controller cable, connect the force sensor to the force sensor controller.



Warning! *The SYSIO DB-25 connector on the C500C controller is not a serial communication port. Connecting a serial device to the SYSIO connector will result in damage to the controller and the serial device.*

- 4 Using the null modem interface cable, connect the force sensor controller to the 9-pin MCE (serial) port at the rear of the C500C controller.

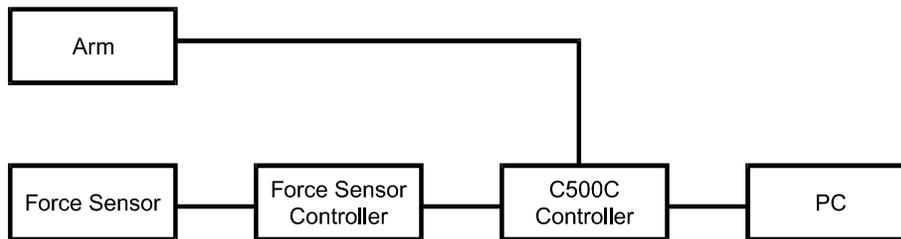


Figure 3: Cable connections between the force sensor and the robot system.

- 5 Allow extra slack in the cables to relieve strain at the following locations:
 - Between the arm elbow and the sensor body
 - At the arm shoulder
 - Near the base of the robot

Testing the Force Sensor

This procedure uses the sample applications on the floppy disk supplied with your force sensor kit to check that the force sensor is responding normally. These applications are explained in detail in the section entitled “Detailed Examples” on page 23.

To test the force sensor:

- 1 Turn on the robot system.
- 2 Using Robcomm3, transfer the force sensor example files from the disk to the `\app\test` directory on the C500C controller.
- 3 At the Robcomm3 system prompt, type

```
$ ash test
```
- 4 Compile and run `fon.r3` by typing the following:

```
test> fon.r
test> fon
```
- 5 The program attempts to initialize and activate the force sensor.
 - If you see the message “Unable to communicate with force sensor”, the C500C controller could not communicate with the force sensor. Shut down the C500C controller and the force sensor controller and check all cable connections, then repeat the test procedure. If the error persists, consult the Troubleshooting section at the end of this guide.
 - If the test runs correctly, compile and run `force.r3`. The sample application creates a dynamic display of sensed force and torque data.
- 6 While examining the data displayed in the Robcomm3 window, press against the sensor with your hand.
 - If the data change in response to the applied load, the sensor is functioning correctly.
 - If the data do not change, check all cable connections and repeat the test. If the error persists, consult the Troubleshooting section at the end of this guide.

Calibrating the Force Sensor

The example program `fcal.r3` can be used to automatically calibrate the force sensor to nullify the gravitational force readings of the current payload. For more detail on `fcal.r3`, see “Calibrating the Sensor to Compensate for Gravity” on p.26.

To perform the calibration:

- 1 Verify that the tool used with the force sensor is installed in the tool flange.

- 2 At the ash prompt, enter the appropriate tool transform by typing:

```
test> tool( xc, yc, zc, yaw, pitch, roll)
```

where x_c , y_c , and z_c , are the Cartesian coordinates of the center of the tool with respect to the default origin.

Note: The RAPL-3 library command `tool_set` can also be used to specify the tool transform. See the RAPL-3 Language Reference Guide for more detail.

- 3 If you have not already done so, transfer the example files to the C500C controller.

- 4 Compile and run `fcal.r3` by typing:

```
test> r3c fcal.r3
```

```
test> fcal
```

- 5 The program automatically calibrates the force sensor.



Caution! *Axes must be calibrated in the order +X, -Z, +Y. Calibrating axes in a different order results in an inaccurate calibration.*

If you prefer not to use the automated procedure, you can calibrate each force sensor axis individually with the `force_calibrate()` RAPL-3 library command. This and other RAPL-3 library commands are explained in the following chapter.

RAPL-3 Library Commands

The force sensor is programmed and operated using RAPL-3 library commands. This chapter provides sensor commands, variables, and sample applications that can easily be adapted to suit your needs.

Operating Commands

The following RAPL-3 commands allow you to develop programs which interact with the force sensor.

- **force_enable** initializes force parameters and sets default scaling, threshold values, and the force sensor frame type.
- **force_calibrate** performs a calibration to compensate for the gravitational force or bias that is sensed as the result of a tool or payload.
- **force_frame_set** specifies the frame of reference in which data is represented.
- **linklen_set** is used to adjust the link length value for an axis to account for the length of the force sensor.
- **online** activates or deactivates RAPL-3 online mode and force tracking.
- **force_param_set** sets the force sensor parameters.
- **force_param_get** obtains the current force sensor parameters.
- **force_data_get** retrieves all six elements of force data.
- **tool_set** defines the origin and orientation of the tool coordinate system.

force_enable

Description	The <code>force_enable</code> command installs the force sensor system by initializing all force parameters. It is also used to set default scaling, threshold values, and the force sensor frame type to <code>SENSOR_FR</code> .
Syntax	<code>command force_enable(int mode)</code>
Parameters	<code>mode: ON</code> turns on sensor sampling <code>OFF</code> turns off sensor sampling
Returns	<code>integer ≥ 0</code> Success <code>integer < 0</code> Failure

force_calibrate

Description	This command allows compensation for the gravitational force or bias sensed as a result of a tool or payload. Gravity must be nullified in order for the robot to exert a specified force in any degree of freedom at any orientation to gravity. Force calibration is especially important for unbiased force tracking along compound curves. Calibrate commands can also be used to weigh the robot payload.	
Syntax	<code>command force_calibrate(int <i>coordinate</i>)</code>	
Parameters	coordinate:	.X_WORLD sensor axis aligned to world +X direction. .Y_WORLD sensor axis aligned to world +Y direction. .Z_WORLD sensor axis aligned to world +Z direction.
Returns	integer ≥ 0 Success integer < 0 Failure	
Notes	<p>The end effector must be installed when performing the calibration commands. Three separate calibrations must be performed with the sensor flange x-axis aligned to real world +X, -Z, and +Y axes, <i>in that order</i>. For the complete sequence refer to the <code>fc1.r3</code> example program on page 26.</p> <p>You can check the calibration for your particular application with a weight scale. Compare the forces displayed on your scale with the force read in by displaying force parameter [6]. Example program <code>force.r3</code> on page 24 can be used as a template for creating a program to display this parameter. Gravity compensation is normally effective within four ounces or less. However, certain conditions can alter this effective accuracy. Refer to Troubleshooting the Force Sensor on page 31.</p>	

force_frame_set

Description	This command specifies the frame of reference in which force sensor data is represented.	
Syntax	<code>command force_frame_set(int <i>frame</i>)</code>	
Parameters	frame:	SENSOR_FR Forces and torques are resolved in the sensor frame of reference. WORLD_FR Forces are resolved in the world frame of reference. Torques cannot be meaningfully represented in the world frame of reference and are resolved in the tool frame of reference instead. TOOL_FR Forces and torques are resolved about the tool tip. When using this frame of reference it is particularly important for the tool transform as well as the link length offset to be set correctly. The tool transform is set with the <code>tool_set()</code> command, link length offset is set with <code>linklen_set()</code> .
Returns	integer ≥ 0 Success integer < 0 Failure	

linklen_set

Description	This command sets the link length for an axis.
Syntax	<code>command linklen_set(int link, float length)</code>
Parameters	<code>link:</code> An integer value for the link being initialized <code>length:</code> An integer value for the length of the link.
Returns	<code>integer ≥ 0</code> Success <code>integer < 0</code> Failure
Notes	The <code>linklen_set</code> command adjusts to the new location of the tool flange since the force sensor is now considered to be a part of the robot structure. For the ATI sensor, the link should be set to a value of 4.2638 inches if working in Imperial units and 108.3 millimeters if working in Metric.

online

Description	This command activates or deactivates online mode and force tracking. Force tracking cannot be activated when online mode is set to OFF and is automatically disabled when a non-straight line move is issued to the robot.
Syntax	<code>command online(int option)</code>
Parameters	<code>option:</code> ON initiates online mode OFF stops online mode 4 Enables force tracking. The robot memory receives tracking offsets in all directions. 5 Disables force tracking.
Returns	<code>integer ≥ 0</code> Success <code>integer < 0</code> Failure
Notes	<code>online</code> mode requires all subsequent motion commands to be executed through a path queue eight motions in depth and causes the robot to move through locations without stopping. The sensor tracking function enters positional offsets into the trajectory planner. These offsets depend on the forces sensed, the particular gain values chosen, and the frame of reference being used. Since straight-line mode must be commanded for tracking, any non-straight line move will turn tracking off. CTRL+Z or the <code>halt()</code> command can also be used to turn force tracking off. The recommended way to begin tracking is to issue a null straight-line move (for example, <code>wxs 0</code>). This technique is shown at the end of the <code>track.r3</code> example program, later in this chapter.

force_param_set

Description	This command sets the force sensor parameters. The complete list of these parameters can be found later in this chapter.
Syntax	<code>command force_param_set(int <i>index</i>, float <i>value</i>)</code>
Parameters	index: Specifies the desired location within the force sensor array. value: Specifies the desired value of the force sensor variable.
Returns	<code>integer ≥ 0</code> Success <code>integer < 0</code> Failure

force_param_get

Description	This command gets a force sensor parameter. The complete list of these parameters can be found later in this chapter.
Syntax	<code>command force_param_get(int <i>index</i>, float <i>value</i>)</code>
Parameters	index: Specifies the desired location within the force sensor array. value: Specifies the desired value of the force sensor variable.
Returns	<code>integer ≥ 0</code> Success <code>integer < 0</code> Failure

force_data_get

Description	This command retrieves all six elements of force data resolved into the chosen frame of reference.
Syntax	<code>command force_data_get(var float[6] <i>data</i>)</code>
Parameters	data: An empty 6-element array that will be populated with all six elements of force data.
Returns	<code>integer ≥ 0</code> Success <code>integer < 0</code> Failure

tool_set

Description	<p>Sets a tool transform, a redefinition of the origin point and the orientation of the tool coordinate system.</p> <p>The default origin is the center of the surface of the mechanical interface (tool flange).</p> <p>The <code>tool_set()</code> command has the capacity for a 6 degree-of-freedom transformation. The origin can be re-defined by translational coordinates: x, y, and z. The orientation can be re-defined by rotational coordinates: yaw, pitch, and roll. A <code>cloc</code> data type is used which requires an integer constant flag followed by float constant coordinates.</p>
Syntax	<code>command tool_set(var cloc toolloc)</code>
Parameters	<p><code>toolloc</code>: the transform with x, y, z, yaw, pitch, roll information</p> <p><code>flag</code>: this attribute is unused when specifying tool locations</p> <p><code>x</code>: the distance along the X axis, in current units</p> <p><code>y</code>: the distance along the Y axis, in current units</p> <p><code>z</code>: the distance along the Z axis, in current units</p> <p><code>yaw</code>: the rotation around the Z axis, in degrees</p> <p><code>pitch</code>: the rotation around the Y axis, in degrees</p> <p><code>roll</code>: the rotation around the X axis, in degrees</p> <p><code>e1</code>: this attribute is unused when specifying tool locations</p> <p><code>e2</code>: this attribute is unused when specifying tool locations</p>
Returns	<p><code>integer ≥ 0</code> Success</p> <p><code>integer < 0</code> Failure</p>
Example	<pre>cloc mytool mytool = build_cloc(0, 2.0, 0.0, 3.0, 0.0, 0.0, 0.0, 0.0, 0.0) tool_set(mytool) ;; for a tool with a tool centre-point 2.0 units along the X axis ;; and 3.0 units along the Z axis from the default origin</pre>
Notes	<p>When defining a tool transform, some of the standard <code>cloc</code> attributes are ignored. See the commands <code>base_set()</code> and <code>build_cloc()</code> in the RAPL-3 manual for more detail on using <code>cloc</code> variables.</p>

Operating Parameters

The force sensor communicates with the robot program through an array of parameters which register sensed forces and regulate the relationship between sensed forces and the forces exerted by the robot. The array contains 62 elements (0 through 61) and only becomes active when the `force_enable(ON)` command is executed.

To operate in force control mode, the variables must specify setpoints and gains for each degree of freedom. Setpoints specify the desired or target exerted forces. Gains specify positional corrections to be made according to deviations from the setpoints.

Gains are usually established during the running of the application and it is typical to change them often. A qualified application programmer must select a range of gains that provide suitable arm performance while maintaining a safe operational state. Refer to the Read or Write Parameters section later in this chapter.

In the descriptions below, the term force implies both force and torque data. Thus a sub-array represented by the symbols F_x , F_y , F_z , T_x , T_y , and T_z indicates force and torque data corresponding to X, Y, Z, Yaw, Pitch, and Roll. Figure 4 illustrates the positive directions of these forces and torques according to the right hand rule.

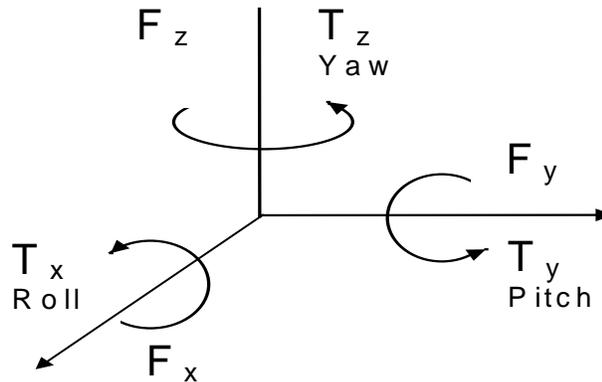


Figure 4: Directions of forces and torques

Read-Only Parameters

Read-only parameters are provided by sensor readings only and not by user input.

Resolved Forces/ Torques (ounces and ounce-inches)

Parameters [0..5] (F_x , F_y , F_z , T_x , T_y , T_z)

Raw values of sensed forces and torques are resolved into the appropriate reference frame (world, tool, or sensor), as defined by the command:

```
force_frame_set(SENSOR_FR|WORLD_FR|TOOL_FR)
```

Resultant Force (ounces)Parameter [6] (F_r)

The value of the resultant sensed force is equivalent to the maximum resolved force in the current coordinate system, as calculated by:

$$F_r = \sqrt{(F_x^2 + F_y^2 + F_z^2)}$$

Resultant Torque (ounce-inches)Parameter [7] (T_r)

The value of the resultant sensed torque is equivalent to the maximum resolved torque in the current coordinate system, as calculated by the following equation:

$$T_r = \sqrt{(T_x^2 + T_y^2 + T_z^2)}$$

Tracking Offset Registers (inches or mm and degrees)

Parameters [8..15](x, y, z, Yaw, Pitch, Roll)

The positional tracking offset registers are values that provide data on the amount of positional correction that the software force loop is exerting to the robot trajectory in order to maintain the force setpoints.

Note: These are read-only values. Do not alter them.

Maximum Sensed Forces (ounces and ounce-inches)Parameters [16..23] (F_x, F_y, F_z, T_x, T_y, T_z, F_r, T_r)

This data registers the maximum absolute values of forces, torques, and resultants that have been sensed since the activation of the interface. You can re-set the registers by writing a zero to these values. The maxima will continue from that point onwards.

Read or Write Parameters

Read or Write parameters are provided by force sensor readings and/or by user input.

Proportional Control Gains (ounces and ounce-inches)

Parameters [24..29] $(F_x, F_y, F_z, T_x, T_y, T_z)$

Proportional control gains determine the sensitivity of the force control software loop and affect the force exerted by the robot arm. They are positional corrections based on sensed deviations from the desired force setpoints. Refer to the next page. The units of these gains are in ounces per inch (or Newton's per millimeter) for the forces and radians per ounce-inch (or radians per Newton-millimeter) for torques.

The default gain values are set to provide a reasonable starting point for most applications.

Table 2: *Default Gain Settings*

	Proportional Gains		Differential Gains	
	Force	Torque	Force	Torque
Imperial	0.00008	0.000005	0.000004	0.000003
Metric	0.01	0.00002	0.0004	0.0003

Force gains are in units of inches per oz (millimeter per Newton) and torque gains are in units of radians per ounce-inch (radians per Newton-millimeter)

When programming an application, start with these recommended values and gradually decrease or increase them to a level of acceptable performance and stability.



Warning! *Start with the default gain settings and only modify gains for degrees of freedom required to perform the task. "Stiff" contact between the robot and objects in its environment can result in deteriorating stability and loss of control.*

Differential Control Gains (ounces and ounce-inches)

Parameters [30..35] $(F_x, F_y, F_z, T_x, T_y, T_z)$

Differential control gains determine a correction factor based on changes in sensed forces and torques. This correction provides a faster response to transient force conditions. The units of these gains are in inches per ounce-seconds for forces, and radians per ounce-inch-seconds for torques. Typically, differential gains are an order of magnitude smaller than the proportional gains. See default settings above.

Threshold Forces (ounces and ounce-inches)Parameters [36..41] $(F_x, F_y, F_z, T_x, T_y, T_z)$

The threshold values for sensed forces are the lower bound values below which the resolved forces or torques will be ignored during force tracking. Although the sensor is able to detect changes in force of as little as 0.2 oz, in practice noise, mechanical vibrations, and other disturbances prevent accuracy below the default threshold values of 1.0 oz (0.5 n). It may be necessary to increase the threshold value if working with particularly heavy payloads, or in particularly rough applications.

Note: Actual force readings can always be read by the user regardless of the threshold value.

Force Setpoints (ounces and ounce-inches)Parameters [42..49] $(F_x, F_y, F_z, T_x, T_y, T_z, F_r, T_r)$

The force setpoints are the desired forces and torques to be exerted by the robot in contacting an object. Setpoints are used to determine positional offsets according to the force sensor's tracking algorithm. The setpoints remain unchanged during normal operation until the sensor interface is disabled. Turning force tracking off (e.g. wx 0) will not affect this data, so force control can be temporarily "suspended" without losing the setpoint values.

Force Sensor Data Scaling (ounces and ounce-inches)Parameters [50..55] $(F_x, F_y, F_z, T_x, T_y, T_z)$

The force sensor data scaling provides the default units of measure for the type of force sensor you use. Normally, this data is automatically established when you install the force sensor. If another sensor body is used (i.e., a metric variant) these values can be changed. In this case, it is recommended that the values be changed only once, immediately after the initialization of the force sensor interface.

Note: Do not change these parameters unless you are using a non-standard sensor.

Limit Clamp Parameters (ounces and ounce-inches)Parameters [56..61] $(F_x, F_y, F_z, T_x, T_y, T_z)$

The limit clamp settings limit the positional corrections made by the robot arm. During operation, the force sensor generates tracking offset registers which feed to the command generator.

Note: Refer to the Read-Only Parameters section earlier in this chapter.

Limit clamp parameters limit the amount of offset that the command generator can command. When the command generator accumulates more offset than is specified by these parameters, the robot stops adapting to the sensed force, and the exerted force will deviate from the programmed force setpoint.

Detailed Examples

The following examples illustrate and explain typical RAPL-3 programs needed to operate the force sensor. The programs initialize the force sensor interface, perform typical force control procedures, and remove the sensor interface. All of these programs are included on the force sensor distribution disk.

Note: None of the force-related commands are available through the applications shell (ash). It is recommended that you become familiar with these examples and modify the programs for use with your specific application.

Table 2: *RAPL-3 programs for force sensor operation*

Program	Purpose
<code>fon.r3</code>	Initializes force sampling mode
<code>force.r3</code>	Displays current sensor readings
<code>track.r3</code>	Initializes force control mode
<code>fcal.r3</code>	Calibrates the force sensor.
<code>guard.r3</code>	Performs a guarded move
<code>guide.r3</code>	Forces a guided teach
<code>foff.r3</code>	Turns off the force sensor

Sampling Forces and Torques

These programs are used for sampling forces and torques, but do not control the robot arm. To disengage the sensor interface, refer to the `foff.r3` program in the next section.

Initializing Force Sampling Mode

The `fon.r3` program initializes the force sensor software and then turns on the sensor sampling interface.

```
main      ;; program name: fon
    force_enable(ON)      ;; initialize & turns on sampling
    ctl_rel()
end main
```

Displaying Current Sensor Readings

The `force.r3` program provides a dynamic display on the terminal screen of the currently sensed force and torque data as well as a convenient debugging tool for testing the sensor.

```
main          ;; force sensor display routine

    int i
    int j
    float[6] params
    float[6] s
    float[6] pmax= {-1000,-1000,-1000,-1000,-1000,-1000}
    float[6] pmin = {1000,1000,1000,1000,1000,1000}
    float[6] avg= {0,0,0,0,0,0}
    for i = 0 to 250
        force_data_get(params)
        for j = 0 to 5
            if pmax[j] < params[j]
                pmax[j] = params[j]
            end if
            if pmin[j] > params[j]
                pmin[j] = params[j]
            end if
            avg[j] += params[j]
        end for
        printf("{6.1f}, {6.1f}, {6.1f},",params[0], params[1], params[2])
        printf(" {6.1f}, {6.1f}, {6.1f}\n", params[3],params[4],params[5])
    end for
    for j = 0 to 5
        s[j] = pmax[j] - pmin[j]
        avg[j] = avg[j] / 250.0
    end for
    printf("Max vals {5.1f}, {5.1f}, {5.1f}",pmax[0],pmax[1],pmax[2])
    printf("{5.1f}, {5.1f}, {5.1f}\n",pmax[3],pmax[4],pmax[5])
    printf("Avg was: {5.1f}, {5.1f}, {5.1f}, {5.1f}, {5.1f},
    {5.1f}\n",avg[0],avg[1],avg[2],avg[3],avg[4],avg[5])
    printf("Min vals {5.1f}, {5.1f}, {5.1f}",pmin[0],pmin[1],pmin[2])
    printf("{5.1f}, {5.1f}, {5.1f}\n",pmin[3],pmin[4],pmin[5])
    printf("Spread was {4.1f}, {4.1f}, {4.1f}, {4.1f}, {4.1f},
    {4.1f}\n",s[0],s[1],s[2],s[3],s[4],s[5])
    ctl_rel()
end main
```

Controlling Forces and Torques

These examples illustrate how to program a typical force control application. The programs should be executed in the order in which they appear below.



Warning! *Start with the default gain settings and only modify gains for degrees of freedom required to perform the task. “Stiff” contact between the robot and objects in its environment can result in deteriorating stability and loss of control.*

Initializing Force Control Mode

The `track.r3` program initializes the sensor for force control mode. The program chooses the tool coordinate system, which is generally the most useful. It begins a Cartesian motion mode by issuing a null straight line move, `wxs(0)`. This program should be called once only at the beginning the application.

```
main                                ;; program name: track
    force_enable(ON)
    force_frame_set(TOOL_FR)
    online(ON)                        ;; enable online mode
    online(4)                          ;; engage force tracking mode
    wxs(0)
    ctl_rel()
end main
```

Calibrating the Sensor to Compensate for Gravity

The `fc1.r3` program calibrates the force sensor to nullify the gravitational force readings or bias of the current payload.

Note: The tool used with the force sensor must be installed before calibration.

The order of calibration in this program is not arbitrary. It is important that the sensor axes be calibrated in the correct order. The X-axis must be calibrated first, then the Z-axis, and finally the Y-axis. Deviation from this sequence will result in unreliable force data. Note the use of the `linklen_set()` command to compensate for the extension length of the force sensor apparatus installed on the arm. When the `fc1.r3` program is completed, the sensor works in a gravity compensated mode.

```
main      ;; program name: fc1
          .define NEW_METRIC_LINK_4  (108.3)
          .define NEW_IMPERIAL_LINK_4  (4.2638)

          int units
          units_get(units)
          if(units == UNITS_METRIC)
              linklen_set(4,NEW_METRIC_LINK_4)
          else
              linklen_set(4,NEW_IMPERIAL_LINK_4)
          end if
          force_enable(ON)
          speed(30)
          ready()
          finish()
          delay(1500)
          force_calibrate(WORLD_X)
          joint(5,-90)
          finish()
          delay(1500)
          force_calibrate(WORLD_Z)
          joint(4,90)
          finish()
          delay(1500)
          force_calibrate(WORLD_Y)
          force_frame_set(TOOL_FR)
          ctl_rel()
end main
```

Performing a Guarded Move to Contact a Part

The program `guard.r3` performs a guarded move, which is a motion that proceeds in a position controlled fashion but upon detection of a contact force, reverts to a force control mode. A force control program turns on tracking and sets gains. It contains a software loop that tracks sensed force and relates it to the forces and torques the robot arm will need to exert.

The example program below causes the arm to move in the world x direction until a force of six Newton's is sensed. It then moves in the world y direction being compliant in that direction.



Warning! Ensure that the robot workcell is free from obstructions before running this program.

```
main
;; Note: This program is designed to run in
;; metric mode. If the robot is being
;; operated in imperial units be sure to
;; make the appropriate changes
float fparam
int j

force_frame_set(WORLD_FR)
;; set up FY gains. Note that all torque gains are set to zero
;; except the world Y direction gains

fparam = 0

for j = 0 to 6
    force_param_set(24+j,fparam)
    force_param_set(30+j,fparam)
end for

fparam = 0.01
force_param_set(25,fparam)
fparam =0.00002
force_param_set(31,fparam)

speed (6)
ready()
joint(1,-45)
joint(2,20)
joint(3,20)
joint(5,-40)
finish()
wxs(-150)
finish()
delay(500)
```

```
online(ON)
online(4)
speed(6)
wxs(300)
fparam = 0.0

;; wait until contact force is encountered
while ( fabs(fparam) < 6 )
    force_param_get(0,fparam)
end while

;; stop the robot, move over a bit and engage tracking mode
halt()
wxs(-1)
finish()
online(ON)
online(4)
speed(2)
wys(-10)
ctl_rel()

end main
```

Forcing a Guided Teach

The `guide.r3` program implements a force-guided teach mode where the robot attempts, by moving appropriately, to zero out any external forces sensed. This program assumes that the force sensor has already been calibrated.

```
main                ;; program name: guide
                   ;; assumes sensor is calibrated

    int i
    float fparam, param1, param2

    ;; move the robot to a suitable starting position
    speed(20)
    ready()
    joint(2,20)
    joint(3,20)
    joint(5,-40)
    finish()

    ;; set up the Tool frame and engage tracking
    force_frame_set(TOOL_FR)
    online(ON)
    online(4)
    wxs(0)

    printf("Begin Force Guided Teach \n")

    ctl_rel()

end main
```

Turning off the Force Sensor

The `foff.r3` program turns the sensor off and disengages the interface from the robot controller. It can be called when there is no further use for the force sensor.

```
main                ;; program name: foff
                   ;; disengage tracking mode
    online(5)
    online(OFF)      ;; disable online mode
    force_enable(OFF)
    ctl_rel()

end main
```

CHAPTER 4

Troubleshooting

Force sensor problems can result from incorrect installation, calibration, and communications or from faulty equipment. Diagnosis requires a computer (PC) with Robcomm software.

Incorrect or Inconsistent Force Data

If the force sensor delivers incorrect or inconsistent data, monitor the raw sensor data and check sensor body orientation according to the following steps:

- 1 Run Robcomm and choose terminal mode from the utilities menu.
- 2 At the system prompt (\$), compile and execute `fon.r3`.
- 3 Press Enter to enable force sensing.
- 4 Compile and execute `force.r3`.
- 5 Press Enter. You will see a dynamic display of current force and torque data originating from the sensor.
- 6 If bad data values are present, perform the following checks:
 - Change the frame of reference to `SENSOR_FR` and run the `force.r3` program again. The current sensor data will be displayed in its raw (untranslated) form, using the sensor frame of reference.
 - Check the orientation of the sensor body on the robot flange. If it is not correct, then the sensor data displayed by the `force.r3` program may appear in directions that are not consistent with the installation.

Inaccurate Sensor Readings

Force sensor readings should be accurate to within four ounces. Inaccurate sensor readings may be caused by the following sources of interference.

Hoses or electrical conductors under strain

During the calibration, a false reading may result if hoses or electrical conductors come into contact with the sensor during calibration.

Verify that all cables or hoses are strain-relieved and properly supported, and be sure that they do not bind during the travel of the payload.

Incorrect or unspecified tool transform

Ensure that a correct tool transform has been specified. If the correct tool transform has not been set, sensed torques will be incorrect.

You can specify a tool transform at the ash prompt by typing:

```
test> tool( xc, yc, zc, yaw, pitch, roll)
```

where x_c , y_c , and z_c , are the Cartesian coordinates of the center of the tool with respect to the default origin.

Electrical noise

Check the system for sources of electrical noise. Re-routing the cable between the sensor and the force controller box may decrease the effect of electrical interference. Sensors have varying characteristics of electrical noise immunity. Consult the sensor manufacturer's specifications.

Improperly matched force sensor and sensor controller

Be sure that the sensor body is matched to the sensor controller unit. Verify that the serial number on the inside of the force sensor matches the serial number on the back of the force controller box. These devices have calibration data which is important for accurate measurement, although it is possible to operate an unmatched set, it is not recommended.

Changes in temperature

Changes in ambient temperature can affect the accuracy of measurement of the sensor. Consult the sensor manufacturer's specifications.

Application Cannot Communicate With Sensor

You will receive a communications error if an application cannot communicate with the sensor. Use the following steps if you are having difficulty establishing communication with the sensor:

- 1 Ensure that the force controller is plugged in, turned ON, and cables connecting the sensor to the controller and the controller to the C500c are both present and undamaged.
- 2 Verify that the cable connecting the force controller box and the C500c robot controller is a NULL (straight through) cable.
- 3 Ensure that the green power light on the front of the force controller box is ON and the red health light is OFF.
- 4 Verify that the version of CROS on the controller supports the force control option. This can be checked by bringing up the terminal window and typing the command `crosver` from the command prompt. The version number should be at least 2.4.1113. If the version number is less than this, consult CRS Customer Support to obtain a more recent release.

Sensor Electronics Configuration

The electronics configuration of the sensor controller is factory preset according to the exact specifications for your force sensor. Under normal sensor operation, there is no need to change any of these settings.

However, if the sensor becomes damaged it may be necessary to re-configure the sensor controller communication settings.

Note: These settings are part of the sensor controller unit and are not part of the robot controller.

Communication settings only need to be re-configured once to correct a force sensor problem. Refer to the configuration requirements and code descriptions in the manufacturer's programming guide for your sensor. Always be sure to use the correct sensor with its matching controller unit and cabling. Mixing these items will result in reduced accuracy. Re-configuration should be done only by a qualified service technician.

Sensor RS232 Communication Settings

This sensor uses an RS232 serial interface to communicate with the robot controller at 38400 baud. The communication parameters are programmed into the sensor controller unit with the following macro:

```
ZC 0,"CD B;SM 0;SB"
```

The macro sets the following parameters:

CD B Binary data communication enabled

SM 0 No overload error messages allowed

SB A single, initial sensor bias

Note: For normal operation, these settings generally do not need to be altered.

If the sensor electronics have suffered a configuration memory loss, consult your manufacturer's sensor installation guide for information on how to re-program these settings.

Force Sensor Specifications

Force Sensor interfaced to robot

Degrees of freedom

The force sensor can detect forces through all six degrees of robot motion.

Rated sensing range

Force

- X, Y 15 pounds (66.7 Newtons)
- Z 50 pounds (222 Newtons)

Torques

- X, Y, Z 50 pound-inches (5650 Newton-millimeters)

Resolution

Force 0.2 ounces (0.055 Newtons)

Torques 0.2 ounce-inches (1.4 Newton-millimeters)

Precision

Precision is defined by the accuracy with which the calibrated sensor measures applied forces and nullifies its weight and the weight of any fixed payload.

Force: 1.0 ounce (0.275 Newton's)

Torques: 1.0 ounce-inches (7.1 Newton-millimeters)

Sensor Controller Unit

Measurement units

- *Force* ounces or Newtons
- *Torque* ounce-inches or Newton-millimeters

Force sensor update

The sensor controller updates at a rate of 100 Hertz (Hz)

Available Coordinate Systems

- World Frame
- Tool Frame
- Sensor Frame

Modes of Operation

- Force Sensing
- Force Controlling

Glossary

Adaptive Control

A software loop that tracks sensed force and relates it to the forces and torques the robot arm will need to exert.

Bias

The gravitational force resulting from the robot's end effector or payload which must be compensated for in order for the robot to exert a specified force in any degree of freedom at any orientation of gravity. Bias is compensated for by the Sensor Force Calibrate commands.

Controller

The computerized device that moves the robot arm by providing the necessary control signals. During the running of an application, these signals are issued by the execution of a robot software program.

Degree of freedom (DOF)

The types of independent motions in which the robot can move its end effector (gripper, tool), as defined by the axes of motion. For example, the A465 robot has six degrees of freedom and can move to co-ordinates specified by: X, Y, Z, Yaw, Pitch, and Roll.

End effector

A device attached to the tool flange that is used to perform the robot's task, such as a gripper, dispenser, buffing wheel, or spray head.

Force

Drive or resistance that acts to push an object toward a different location (as indicated by X, Y, and Z). The force sensor measures force in ounces.

Gains

Variables that specify positional corrections to regulate the sensitivity of force control. Proportional gains are made according to deviations from the force setpoints. Differential gains are made according to changes in sensed forces and torques.

Moment

See Torque.

Orientation

Cartesian rotational coordinates (referred to as Yaw, Pitch, and Roll) that specify the spatial attitudes of a robot's end effector.

Position

Cartesian rectilinear coordinates (referred to as X, Y, and Z) that specify a location or directions in space.

Positional Offset

The distance the arm must move to contact an object with the force specified by the force setpoint in that direction.

RAPL (Robot and Automation Programming Language)

An automation-oriented, line-structured language of commands, similar to BASIC, used for robot applications and communication.

Robcomm (Robot Communication Software)

A Windows or DOS based computer software program that enables the transfer, editing, and debugging of robot application programs and support data; enables robot operation; and provides utilities such as terminal emulation and memory checks.

Real world reference frame

A world Cartesian coordinate system, which has the Z-axis pointing away from gravity. The default or (zero base offset) world reference frame.

Sensor frame of reference

The coordinate system having its origin at the center of the sensor flange. To be useful, the raw data that is sensed here must be translated to the robot's tool or the world coordinate system.

Setpoints

Variables that specify the desired or target force to be exerted by the robot in each degree of freedom.

Torque

Moment that acts to twist an object toward a different orientation (as indicated by Yaw, Pitch, and Roll). The force sensor measures torques in ounce-inches.

Index

A

accuracy, sensor, 32
ash, 23
ATI FT 15/50 Force Sensor, 8

B

bias, 15, 26

C

calibration, force, 11, 26
cartesian motion mode, 25
code descriptions, 35
command generator, 22
communication, troubleshooting, 33
configuration, sensor electronics, 35
contact force, 27
control loop. *See* software loop
coordinate system
 resolution to, 19
 tool, 25
correction factor, 21
corrections, inconsistent, 32

D

data scaling, 22
debugging tool, 24
degree of freedom, 15
desired, 2
diagnostics. *See* troubleshooting
differential control gains. *See* gains
display sensed forces. *See* forces: display current values

E

examples, 23
exerted force, 19

F

fcf.r3 program, 11, 26
foff.r3 program, 29
fon.r3, 23, 32
fon.r3 program, 23
force

 current, 24
 data, 18, 19
 display current values, 24
 exerted, 27
 interface variables, 19
 maximum sensed, 20
 resolved, 22
 resultant, 20
 sampling, 23
 sensed, 19
 setpoints. *See* setpoints
 tracking. *See* tracking
force control mode, 3, 16, 18, 19, 25, 27
 initializing, 25
 suspension of, 22
force sampling mode, 3
 initializing, 23
force sensor data scaling, 22
force sensor parameters, 17
force tracking. *See* tracking
force.r3, 24, 32
force.r3 program, 24
forces
 sensed, 21
forces, threshold. *See* threshold forces
frame of reference, 15

G

gains
 differential control, 21
 proportional control, 21
gravity, compensating for. *See* calibration, force
guard.r3 program, 27
guarded move, 27

I

inaccurate readings, 32
initialize
 force control mode, 25
 sensor program, 23
interface
 turning off, 29
 variables, 19. *See* force:interface variables

L

limit clamp settings, 22

M

maximum sensed force, 20
 re-setting, 20
measuring part weights. *See* force sampling mode
memory loss, 35
moment
 current, 24
 data, 18, 19
 display current values, 24
 exerted, 27
 maximum sensed, 20
 resultant, 20
 sampling, 23
 sensed, 19, 21
motion trajectory planner, 16

N

null straight line move, 16, 25

O

Operating Modes, 37

P

path queue, 16
payload, 11, 15, 26
positional correction, 21, 22. *See also* tracking
positional tracking. *See* tracking
program
 fcal.r3, calibrate for gravity, 11, 26
 foff.r3, turn off sampling, 29
 fon.r3, initialize force sampling, 23
 force.r3, display current forces, 24
 guard.r3, guarded move, 27
 track.r3, initialize force control, 25
programs, example, 23
proportional control gains. *See* gains

R

readings, inaccurate, 32

read-only arrays, 19, 21
real world reference frame, 11, 15
resolved forces, 19, 22
resolved moments, 19
resultant force, 20
resultant moment, 20
RS232 serial interface, 35

S

sensed
 deviations, 21
 forces, 19, 21
 moments, 19
sensor accuracy, 32
sensor, different type, 22
setpoints, 21, 22
software loop, 3, 20, 21, 27
specifications, force sensor, 37
stability, loss of, 21, 25
suspension of force control, 22

T

testing the sensor, 24
threshold forces, 22
track.r3 program, 25
tracking, 16, 22, 27
 offsets, 16, 20, 22
trajectory, 20
transient force conditions. *See* gains, differential control
troubleshooting, 31
typical force control procedures. *See* examples

V

variables. *See* force:interface variables

W

weighing payload. *See* payload