Specialty Plants

User Manual

Quanser Qball-X4

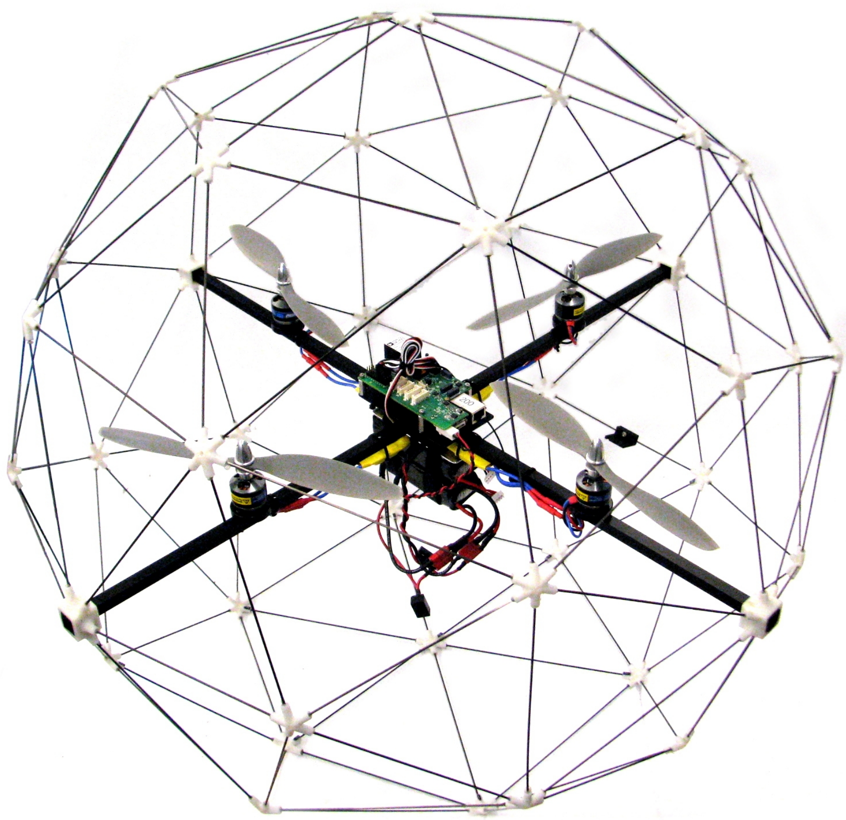


Table of Contents

1. Introduction

The Quanser Qball-X4 (Figure 1) is an innovative rotary wing vehicle platform suitable for a wide variety of UAV research applications. The Qball-X4 is a quadrotor helicopter design propelled by four motors fitted with 10-inch propellers. The entire quadrotor is enclosed within a protective carbon fiber cage (Patent Pending). The Qball-X4's proprietary design ensures safe operation as well as opens the possibilities for a variety of novel applications. The protective cage is a crucial feature since this unmanned vehicle was designed for use in an indoor laboratory, where there are typically many close-range hazards (including other vehicles). The cage gives the Qball-X4 a decisive advantage over other vehicles that would suffer significant damage if contact occurs between the vehicle and an obstacle.

To measure on-board sensors and drive the motors, the Qball-X4 utilizes Quanser's on-board avionics data acquisition card (DAQ), the HiQ, and the embedded Gumstix computer. The HiQ DAQ is a high-resolution inertial measurement unit (IMU) and avionics input/output (I/O) card designed to accommodate a wide variety of research applications. QUARC, Quanser's real-time control software, allows researchers and developers to rapidly develop and test controllers on actual hardware through a MATLAB Simulink interface. QUARC's open-architecture hardware and extensive Simulink blockset provides users with powerful controls development tools. QUARC can target the Gumstix embedded computer, automatically generating code and executing controllers on-board the vehicle. During flights, while the controller is executing on the Gumstix, users can tune parameters in real-time and observe sensor measurements from a host ground station computer (PC or laptop).

The interface to the Qball-X4 is MATLAB Simulink with QUARC. The controllers are developed in Simulink with QUARC on the host computer, and these models are downloaded and compiled into executables on the target (Gumstix [2]) seamlessly. A diagram of this configuration is shown in Figure 2.

Section 2 outlines operator warnings found throughout this manual, Section 3 goes through the prerequisites, and Section 4 lists various documents that are referenced in this manual. The general system description, component nomenclature, specifications, and model parameters are all given in Section 5. Section 6 goes into detail on how to setup the Qball-X4. Lastly, Section 8 contains a troubleshooting guide.

2. Operator Warnings

This symbol marks specific safety warnings and operating procedures that are important for the safety of the Qball-X4 and users. Read these warnings carefully. The Qball-X4 is a powerful and potentially dangerous vehicle if used improperly. Always follow safe operating procedures when using the Qball-X4. Quanser is not responsible for damages and injury resulting from improper or unsafe use of the Qball-X4. Before connecting batteries or attempting to run the Qball-X4, be sure to read this document and become familiar with the safety features and operating procedures of the Qball-X4.

⚠ When handling the Qball-X4, always make sure there are no models running and the power is turned off. It is recommended that users wear safety goggles to protect the eyes.

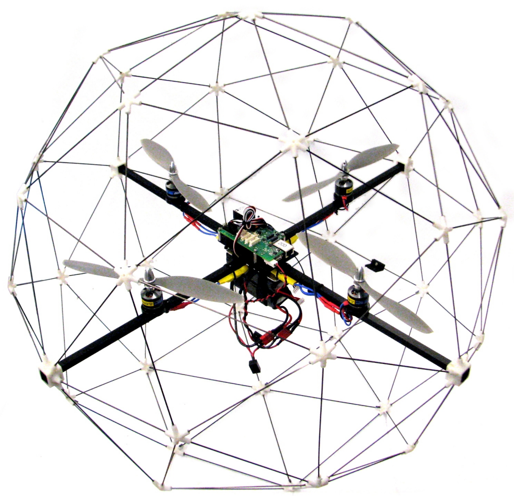


Figure 1: Quanser Qball-X4

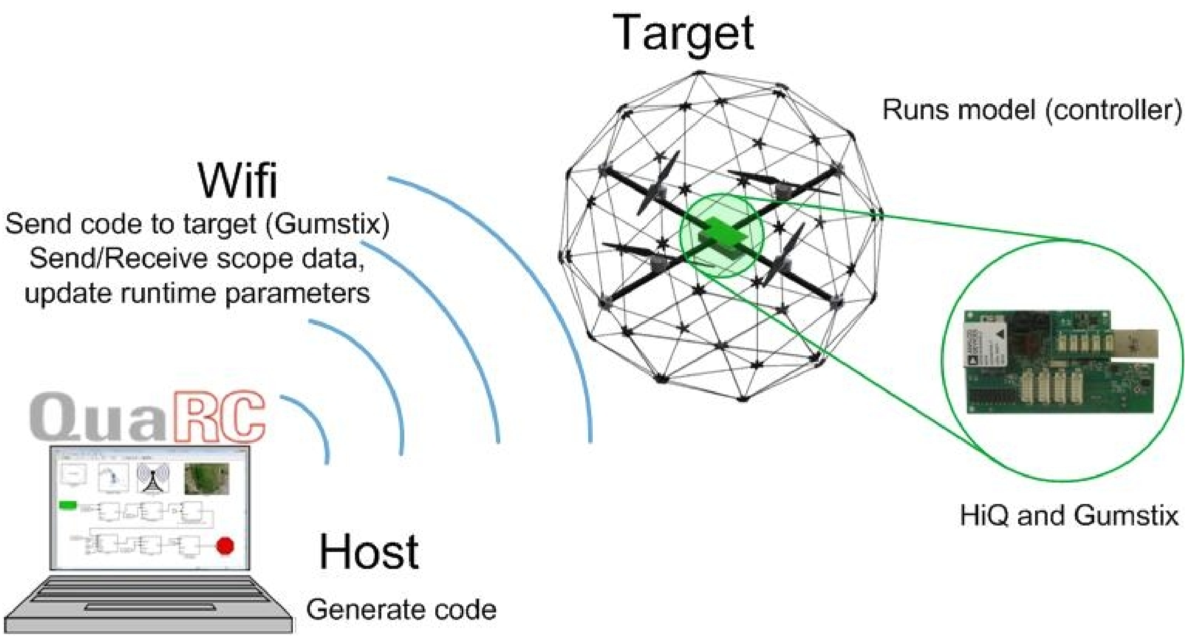


Figure 2: Communication Hierarchy

3. Prerequisites

To successfully operate the Qball-X4, the prerequisites are:

i) To be familiar with the wiring and components of the Qball-X4.
ii) To have QUARC version 2.0 installed and properly licensed.
iii) To be familiar with using QUARC to control and monitor the vehicle in real-time, and in designing a controller through Simulink. See Reference [2] for more details.

4. References

[1] *Gumstix*: http://gumstix.com/
[2] *QUARC User Manual* (type `doc quarc` in Matlab to access)
[3] *Park 480 Brushless motor - 1020Kv:* http://hobbyhobby.com/store/product/68211/%22Park-480-Brushless-Outrunner-Motor%2C-1020Kv%22/
[4] *Propellers description and technical information:* http://www.rctoys.com/rc-products/APC-10-047-SF-CR.html
[5] *Hobbywing Flyfun-30A electronic speed controller manual:* http://www.hobbywing.com/uploadfiles/sx/file/Manual/HW-01-V4.pdf

5. System Hardware and Software Description

5.1. Main Components

To setup this experiment, the following hardware and software are required:

- **Qball-X4:** Qball-X4 as shown in Figure 1 above
- **HiQ:** QUARC aerial vehicle data acquisition card (DAQ).
- **Gumstix:** The QUARC target computer. An embedded, Linux-based system with QUARC runtime software installed [1]
- **Batteries:** Two 3-cell, 2500 mAh Lithium-Polymer batteries
- **Real-Time Control Software:** The QUARC-Simulink configuration, as detailed in Reference [2]

5.2. X4 Diagram

Figure 3 below is a basic diagram of the Qball-X4, showing the axes and angles. Note that the axes follow a right-hand rule with the X axis aligned with the front of the vehicle.

⚠ **The tail or back of the vehicle is marked with colored tape. When flying the vehicle it is common to orient the vehicle such that the tail is pointing towards the operator with the positive X axis pointing away from the operator.**

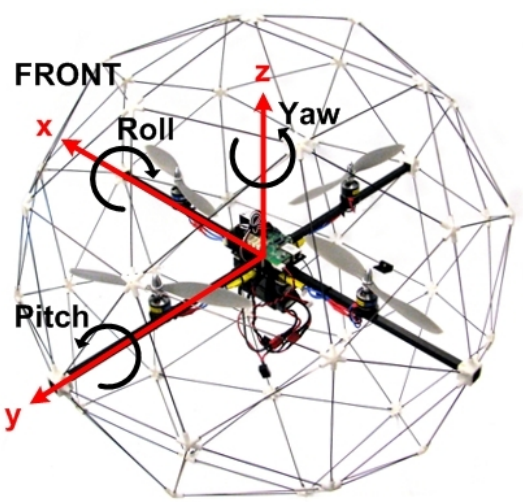


Figure 3: Qball-X4 axes and sign convention

5.3. Qball-X4 Components

The components comprising the Qball-X4 are labeled in Figures 4 to 12 and described in Table 1.

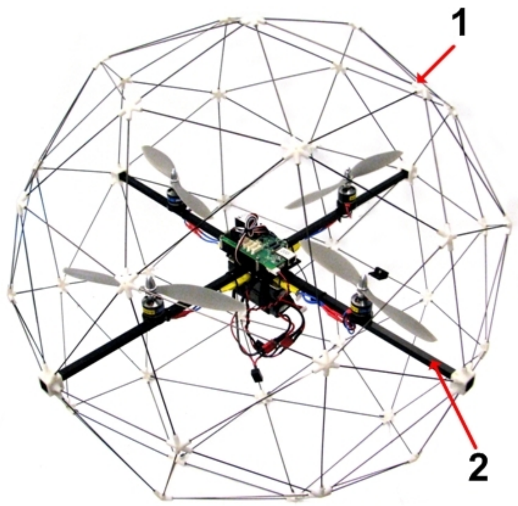


Figure 4: Qball-X4 cage and frame

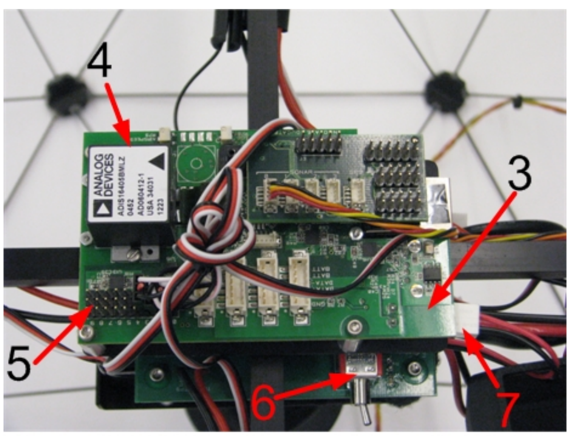


Figure 5: HiQ DAQ

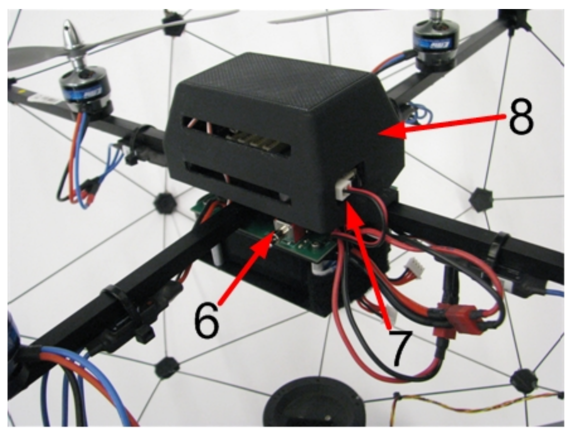


Figure 6: HiQ cover, power switch and connector

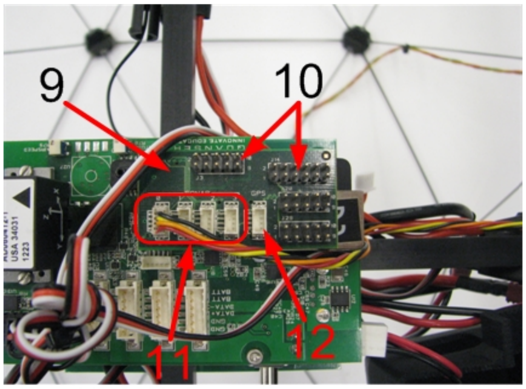


Figure 7: HiQ daughterboard

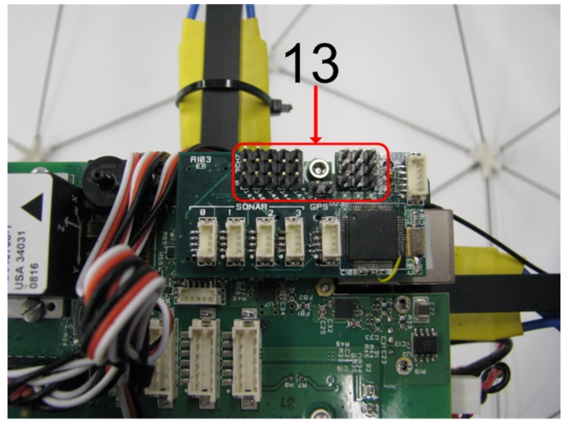


Figure 8: HiQ optional daughterboard with receiver inputs

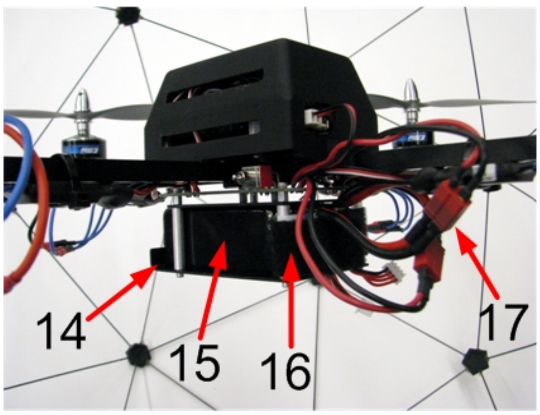


Figure 9: Battery compartment

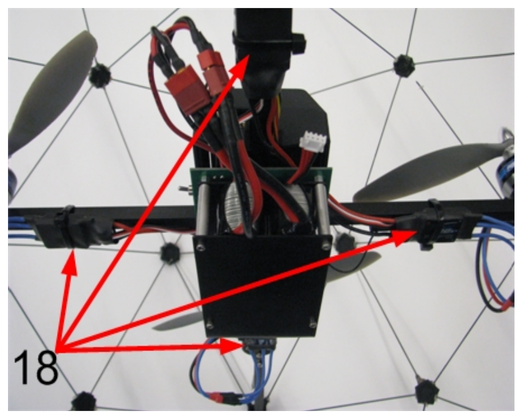


Figure 10: ESCs

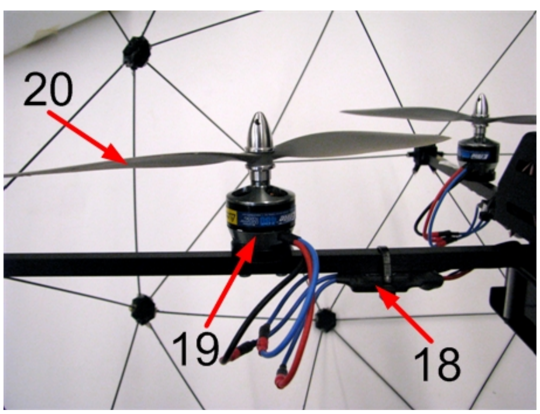


Figure 11: Motor, propeller and ESC

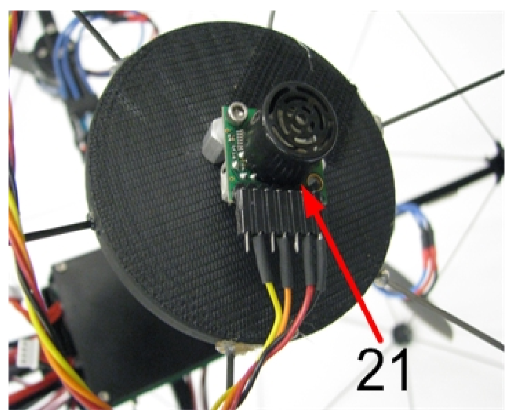


Figure 12: Sonar sensor

Figure 13: Qball joystick, type A

ID #	Description	ID #	Description
1	Qball-X4 protective cage	12	GPS serial input
2	Qball-X4 frame	13	Receiver inputs (optional daughterboard)
3	HiQ DAQ with Gumstix	14	Battery compartment
4	HiQ inertial measurement unit	15	LiPo batteries
5	HiQ servo PWM outputs	16	Velcro straps
6	Qball power switch	17	Battery connectors
7	HiQ power connector	18	ESCs
8	HiQ cover	19	Motor
9	HiQ daughterboard	20	Propeller
10	HiQ daughterboard breakout pins	21	Sonar sensor
11	Sonar inputs	22	Joystick

Table 1 Qball-X4 components

5.3.1. Qball-X4 frame

The Qball-X4 frame (#2 in Figure 4) is the crossbeam structure to which the Qball-X4 components are mounted including the HiQ DAQ, motors and speed controllers. The frame rests inside the Qball-X4 protective cage (#1 in Figure 4). The Qball-X4's protective cage is a carbon fiber structure designed to protect the frame, motors, propellers, and embedded control module (HiQ and Gumstix) during minor collisions. The cage is not intended to withstand large impacts or drops from heights greater than 2 meters.

⚠ **Do not pick up the Qball-X4 from the cage as this may stress the cage and cause damage. Instead, when transporting the Qball-X4 lift it from the ends of the frame as in Figure 14, using both hands to lift the frame from both sides.**

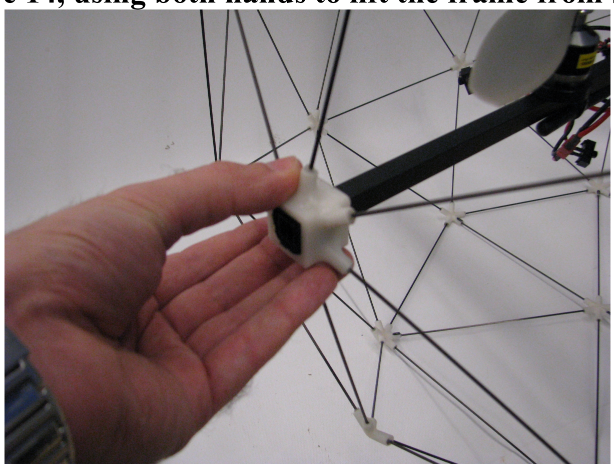


Figure 14: Pick up the Qball-X4 from both ends of the frame

5.3.2. HiQ DAQ

The HiQ DAQ is the Qball-X4's data acquisition board. Together with the Gumstix embedded computer, the HiQ controls the vehicle by reading on-board sensors and outputting motor commands. Each motor speed controller (#18 in Figure 10) is connected to a PWM servo output on the HiQ (#5 in Figure 5). There are 10 PWM servo output channels available on the HiQ and they are labeled 0 to 9 with the ground pins (black wire on the servo cable) located closest to the outer edge of the HiQ board. Each motor speed controller should be connected in a specific order for the provided Qball-X4 controllers to function. Table 2 lists the motors and their standard corresponding servo channels.

Motor	*Servo output channel*
Back	0
Front	1
Left	2
Right	3

Table 2: Motor servo channels

The HiQ may have an optional daughterboard that contains additional I/O such as receiver inputs (#13 in Figure 8), sonar inputs (#11 in Figure 7), and a TTL serial input used for a GPS receiver (#12 in Figure 7). If the Qball-X4 is provided with a sonar (#21 in Figure 12) it should be connected to sonar input channel 0 as labeled on the daughterboard.

5.3.3. Qball-X4 Power

The Qball-X4 uses two 3-cell 2500mAh LiPo batteries (#15 in Figure 9) to power the HiQ and motors. These batteries are housed in a battery compartment (#14 in Figure 9) beneath the Qball's cross frame and held in place using two velcro straps (#16 in Figure 9). The batteries should be stacked horizontally, inserted carefully into the battery compartment, and secured tightly with the velcro straps.

⚠ **Make sure the batteries are firmly connected before attempting to fly the Qball-X4.**

Secure the batteries to the frame before connecting the batteries to the Qball-X4 battery connectors (#17 in Figure 9) and always turn off the power using the Qball power switch (#6 in Figure 5) before changing batteries.

⚠ **LiPo batteries can be dangerous if charged improperly. Review the battery charging procedures and monitor battery levels frequently during flight. The 3-cell LiPo batteries can become damaged and unusable if discharged below 10 V. It is recommended that the batteries be fully charged once they reach 10 V or less.**

5.3.4. Qball-X4 Motors and Propellers

The Qball-X4 uses four E-Flite Park 480 (1020 Kv) motors [3] (#19 in Figure 11) fitted with paired counter-rotating APC 10x4.7 propellers [4] (#20 in Figure 11). The motors are mounted to the Qball-X4 frame along the X and Y axes and connected to the four speed controllers [5], which are also mounted on the frame. The motors and propellers are configured so that the front and back motors spin clockwise and the left and right motors spin counter-clockwise. The electronic speed controllers (ESCs) receive commands from the HiQ in the form of PWM outputs from 1ms (minimum throttle) to 2ms (maximum throttle). The ESCs used in the Qball-X4 are configured with the appropriate throttle range during assembly. It is important that the initial PWM outputs to the ESCs is the minimum throttle duty cycle value 0.05, otherwise you can enter the program mode and alter the ESC settings. Review the ESC's manual for instructions on changing ESC settings [5].

5.3.5. Qball Joystick

The Qball joystick is a critical component in operating the Qball. The joystick allows the operator to fly the Qball using the two stick controls for actuation of throttle (how much lift force is generated by the Qball motors), roll (rotating the Qball left or right to fly left/right), pitch (rotating the Qball forward or backward to fly forward/backward), and yaw (rotating the Qball about the vertical axis to change its direction or heading). Even when flying the Qball in autonomous modes the joystick is needed to initialize and enable the Qball and acts as a kill switch in the event the Qball controller goes unstable and must be stopped. The joystick comes with a label that will designate its type: TYPE A or TYPE B. Make sure that you use the corresponding TYPE A/B joystick models. If you are not sure which type of joystick you have, contact support@quanser.com for assistance. Section 6.5. describes how to use the joystick to fly the Qball using the provided Simulink models.

5.4. Qball-X4 Model

This section describes the dynamic model of the Qball-X4. The nonlinear models are described as well as linearized models for use in controller development. For the following discussion, the axes of the Qball-X4 vehicle are denoted *(x, y, z)* and are defined with respect to the vehicle as shown in Figure 3. Roll, pitch, and yaw are defined as the angles of rotation about the *x*, *y*, and *z* axis, respectively. The global workspace axes are denoted *(X, Y, Z)* and are defined with the same orientation as the Qball-X4 sitting upright on the ground.

5.4.1. Actuator Dynamics

The thrust generated by each propeller is modeled using the following first-order system

$$F = K\frac{\omega}{s+\omega}u \tag{1}$$

where u is the PWM input to the actuator, ω is the actuator bandwidth and K is a positive gain. These parameters were calculated and verified through experimental studies and are stated in Table 3. A state variable, v, will be used to represent the actuator dynamics, which is defined as follows,

$$v = \frac{\omega}{s+\omega} u . \qquad (2)$$

5.4.2. Roll/Pitch Model

Assuming that rotations about the *x* and *y* axes are decoupled, the motion in roll/pitch axis can be modeled as shown in Figure 15. As illustrated in this figure, two propellers contribute to the motion in each axis. The rotation around the center of gravity is produced by the difference in the generated thrust forces. From Eq. (2), let $u=\tilde{u}$, where $\tilde{u}$ is the control input for the pitch or roll dynamics that causes an increase or decrease in thrust force in the two pitch/roll motors shown in Figure 15 and such that the changes in force of each motor are opposite in direction so that the net result is a torque. E.g. the control signal is applied to increase the force in motor 1 and decrease the force in motor 2. This change in motor forces is what causes the resulting torque and roll or pitch dynamics, so we can ignore the net thrust force used to hover the Qball. The change in thrust generated by each motor can be calculated from Eq. (1). The roll/pitch angle, θ , can be formulated using the following dynamics

$$J\ddot{\theta} = \Delta F\, L \qquad (3)$$

where

$$J = J_{roll} = J_{pitch} \qquad (4)$$

are the rotational inertia of the device in roll and pitch axes and are given in Table 3. *L* is the distance between the propeller and the center of gravity, and

$$\Delta F = \Delta F_1 - \Delta F_2 \qquad (5)$$

represents the net change in the forces generated by the motors. Note that the difference in the forces is generated by the difference in the inputs to the motors, i.e.

$$\Delta u = 2\tilde{u} . \qquad (5)$$

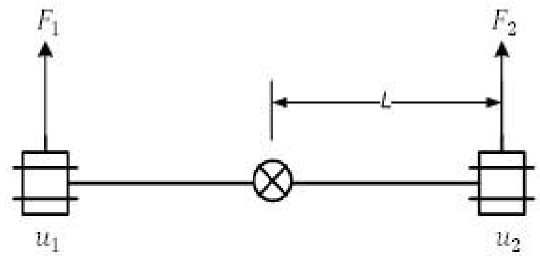


Figure 15: A model of the roll/pitch axis

By combining the dynamics of motion for the roll/pitch axis and the actuator dynamics for each propeller the following state space equations can be derived

$$\begin{bmatrix} \dot{\theta} \\ \ddot{\theta} \\ \dot{v} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & \frac{2KL}{J} \\ 0 & 0 & -\omega \end{bmatrix} \begin{bmatrix} \theta \\ \dot{\theta} \\ v \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \omega \end{bmatrix} \tilde{u}$$

To facilitate the use of an integrator in the feedback structure a fourth state can be added to the state vector, which is defined as follows

$$\dot{s} = \theta$$

After augmenting this state into the state vector, the system dynamics can be rewritten as

$$\begin{bmatrix} \dot{\theta} \\ \ddot{\theta} \\ \dot{v} \\ \dot{s} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & \frac{2KL}{J} & 0 \\ 0 & 0 & -\omega & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \theta \\ \dot{\theta} \\ v \\ s \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \omega \\ 0 \end{bmatrix} \tilde{u}$$

5.4.3. Height Model

The motion of the Qball-X4 in the vertical direction (along the Z axis) is affected by all the four propellers. The dynamic model of the Qball-X4 height can be written as

$$M \ddot{Z} = 4 F \cos(r) \cos(p) - M g$$

where F is the thrust generated by each propeller, M is the total mass of the device, Z is the height and r and p represent the roll and pitch angles, respectively. The total mass, M, is given in the Table 3. As expressed in this equation, if the roll and pitch angles are nonzero the overall thrust vector will not be perpendicular to the ground. Assuming that these angles are close to zero, the dynamics equations can be linearized to the following state space form

$$\begin{bmatrix} \dot{Z} \\ \ddot{Z} \\ \dot{v} \\ \dot{s} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & \frac{4K}{M} & 0 \\ 0 & 0 & -\omega & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} Z \\ \dot{Z} \\ v \\ s \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \omega \\ 0 \end{bmatrix} u + \begin{bmatrix} 0 \\ -g \\ 0 \\ 0 \end{bmatrix}$$

5.4.4. X-Y Position Model

The motion of the Qball-X4 along the X and Y axes is caused by the total thrust and by changing roll/pitch angles. Assuming that the yaw angle is zero the dynamics of motion in X and Y axes can be written as

$$M\ddot{X} = 4F\sin(p)$$
$$M\ddot{Y} = -4F\sin(r)$$

Assuming the roll and pitch angles are close to zero, the following linear state space equations can be derived for X and Y positions

$$\begin{bmatrix} \dot{X} \\ \ddot{X} \\ \dot{v} \\ \dot{s} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & \frac{4K}{M}p & 0 \\ 0 & 0 & -\omega & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} X \\ \dot{X} \\ v \\ s \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \omega \\ 0 \end{bmatrix} u,$$

$$\begin{bmatrix} \dot{Y} \\ \ddot{Y} \\ \dot{v} \\ \dot{s} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & \frac{-4K}{M}r & 0 \\ 0 & 0 & -\omega & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} Y \\ \dot{Y} \\ v \\ s \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \omega \\ 0 \end{bmatrix} u$$

5.4.5. Yaw Model

The torque generated by each motor, τ , is assumed to have the following relationship with respect to the PWM input, u

$$\tau = K_y u$$

where K_y is a positive gain and its value is given in Table 3. The motion in the yaw axis is caused by the difference between the torques exerted by the two clockwise and the two counter-clockwise rotating propellers. The model of the yaw axis is shown in Figure 16.

The motion in the yaw axis can be modeled using the following equation

$$J_y \ddot{\theta}_y = \Delta \tau$$

In this equation, θ_y is the yaw angle and J_y is the rotational inertia about the *z* axis, which is given in Table 3. The resultant torque of the motors, $\Delta \tau$, can be calculated from

$$\Delta \tau = \tau_1 + \tau_2 - \tau_3 - \tau_4$$

The yaw axis dynamics can be rewritten in the state-space form as

$$\begin{bmatrix} \dot{\theta}_y \\ \ddot{\theta}_y \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \theta_y \\ \dot{\theta}_y \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{K_y}{J_y} \end{bmatrix} \Delta \tau$$

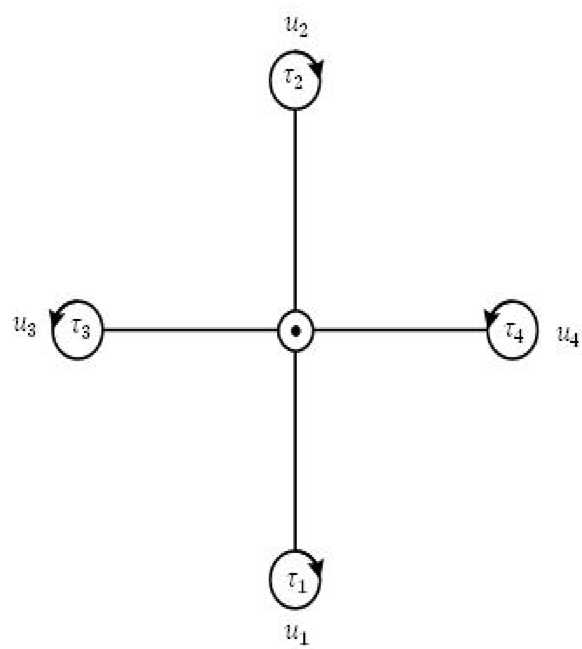


Figure 16: A model of the yaw axis with propeller direction of rotation shown.

Parameter	***Value***
K	120 N
ω	15 rad/sec
J_{roll}	0.03 kg. m^2
J_{pitch}	0.03 kg. m^2
M	1.4 kg
K_y	4 N.m
J_{yaw}	0.04 kg. m^2
L	0.2 m

Table 3: System parameters

6. System Setup

Section 6.1 describes setting up the vehicle hardware. Section 6.2 describes the Qball sensors and how they are accessed in QUARC. Sections 6.3 and 6.4 describe the procedures for configuring the wireless connection in order to communicate with the Qball. Finally, Sections 6.5 and 6.6 list the MATLAB Simulink files provided with the Qball and describe in detail the Qball controller.

6.1. Qball-X4 Vehicle Setup

1. First, check that all motors are securely fastened to the vehicle frame. Check that the propellers are firmly attached to the motors in the correct order: clockwise propellers (viewed from the top) on the front and back motors, counter-clockwise propellers on the left and right motors. Note that the back motor is indicated by a bright colored marking on the Qball-X4 frame.

   Check that the motors are firmly secured to the frame regularly (after every 2 hours of flight). Over time, vibrations in the frame may loosen the motor mounts. If a motor or mount feels loose, tighten it immediately.

   If a propeller is loose, use an allen key to remove the cap holding the propeller to the motor and ensure the propeller mounting shaft is pushed fully down onto the motor shaft. Replace the propeller on the mounting shaft and replace the motor cap and tighten it with an allen key. **Never change propellers or other components of the Qball-X4 with batteries connected.**

2. Install the batteries. Placing the Qball-X4 upsidedown so that it rests on the top of the cage. Align the two Qball-X4 batteries with the plate located on the bottom of the frame and secure the batteries tightly using the two velcro straps as shown in Figure 17. Connect the batteries to the battery connectors and place the Qball-X4 upright again so it rests on the bottom of the cage.

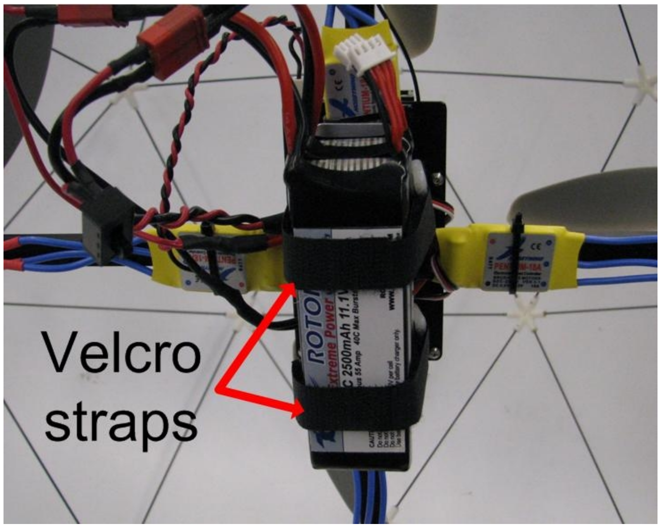


Figure 17: Batteries secured with velcro straps.

3. Power on the Qball-X4 using the two power switches connected to the battery cables (#11 in Figure 9). After 1 minute the Gumstix wireless module should be active. Connect to the GSAH ad-hoc network on the host PC (see section 6.3. Establishing Wireless Connection).

6.2. Qball-X4 Sensors

This section describes the blocks that are used to read the Qball-X4 sensors in Simulink and write outputs to the motors. The QUARC Hardware-In-the-Loop (HIL) blockset it used to communicate with Quanser data acquisition cards, including the HiQ and the Gumstix Verdex. For detailed information on the HIL blockset see the QUARC HIL user guide in the MATLAB help under QUARC Targets/User's Guide/Accessing Hardware. Table 4 lists the HIL blocks used to communicate with the Qball-X4's data acquisition hardware.

Block	***Description***
 HIL Initialize Qball-X4 HiQ (hiq_aero-0)	The HIL Initialize block selects the DAQ board and configures the board parameters. The HIL Initialize block is named via the *Board name* parameter, and all other HIL blocks reference the corresponding HIL Initialize through its name. For the HiQ, there is a board-specific option for selecting the gyroscope model installed on the HiQ. The gyro model is specified by typing “gyro_model=16405” in the HIL Initialize *Board-specific options* parameter or by selecting the model from the board-specific options dialog. The valid values for the gyro_model option are: 16350, 16360, 16400, and 16405. The default value if not specified is 16405. The gyroscope model number is found on the label of the gyroscope (#4 in Figure 5).
 HIL Read Write (Qball-X4 HiQ)	The HIL Read Write block is used to read sensor measurements from the HiQ and write motor commands to the four Qball-X4 motors. The inputs and outputs are specified with numeric channel numbers given in Table 5 and Table 6, respectively.
 HIL Set Property (Qball-X4 HiQ)	The HIL Set Property block is used to set certain board properties specific to the HiQ during model execution (note: property changes are not persistent across multiple model executions). The HIL Set Property block can be used to set integer, double or string type properties. The HiQ supports one integer property and one double property. It is recommended that users specify the property values internally in the block parameters and set the property at model start only (see help for the HIL Set Property block). The HiQ integer property is referenced by property code 128 and is used to set the gyroscope range. Valid values for this property are 75, 150, and 300, which correspond to gyroscope ranges of 75°/s, 150°/s, or 300°/s, respectively. The default gyroscope range is set to 75°/s. The HiQ double property is referenced by property code 128 and is used to set the final PWM output on all PWM output channels after the watchdog expires (typically after the model is stopped). For the Qball-X4, it is recommended that this property be set to a value of 0.05 (zero throttle) to ensure that the motors stop immediately in the event of a watchdog timeout. The default value for this property is 0.
 HIL Watchdog (Qball-X4 HiQ)	The HIL Watchdog block is used to set the timeout limit for the watchdog timer. For the HiQ board, if there is no PWM output command received for a consecutive period of time exceeding the watchdog timeout value then the HiQ watchdog will trigger, forcing the PWM outputs to 0 or to a value specified by the HIL Set Property block. The default timeout value for the HiQ watchdog is 50ms unless specified otherwise with this block. This block can be used to change the timeout value if 50ms is not suitable.

Table 4: HIL blocks used by the Qball-X4.

The HiQ provides several high-resolution avionics sensors, which are used to measure and control the stability of aerial vehicles. The I/O of the HiQ includes:

- 10 PWM outputs (servo motor outputs)
- 3-axis gyroscope, range configurable for ±75˚/s, ±150˚/s, or ±300˚/s, resolution 0.0125˚/s/LSB at a range setting of ±75˚/s
- 3-axis accelerometer, resolution 3.33 mg/LSB
- 6 analog inputs, 12-bit, +3.3V
- 3-axis magnetometer, 0.5 mGa/LSB
- 8 channel RF receiver input (optional)
- 4 Maxbotix sonar inputs
- 2 pressure sensors, absolute and relative pressure
- Input power 10-20V

In addition to the HiQ, the Gumstix Verdex provides the following I/O:

- 11 reconfigurable digital I/O
- 2 TTL serial ports
- Serial GPS input

Figure 18 shows the location of the I/O listed above on the HiQ DAQ. The HiQ I/O listed above is accessed using the QUARC HIL blockset. The Gumstix Verdex digital I/O is accessed through the QUARC HIL blockset, the serial I/O is accessed through the QUARC Stream blockset, and the serial GPS is accessed through the GPS NMEA block. For more information on accessing serial data see the QUARC help under QUARC Targets/User's Guide/Communications.

To initialize the HiQ board, a HIL Initialize block must be placed in the model. The HIL Initialize block is used to initialize a data acquisition card and setup the I/O parameters. In the HIL Initialize block, select the board type 'hiq_aero' to configure the HiQ DAQ and, if desired, enter a name in the Board name field as shown in Figure 19.

Next, to read and write from the HiQ, add a HIL Read Write block to the model (note that the HiQ is optimized for best performance when a single HIL Read Write block is used in a model, adding more HIL I/O blocks may reduce the performance, particularly the maximum sample rate). In the HIL Read Write block, select the board name corresponding to the board name given in the HIL Initialize block. The channels available for reading and writing for the HiQ are listed in Table 5 and Table 6 below. Enter the channel numbers to be read/written or use the browse buttons to open a channel selection dialog as shown in Figure

20.

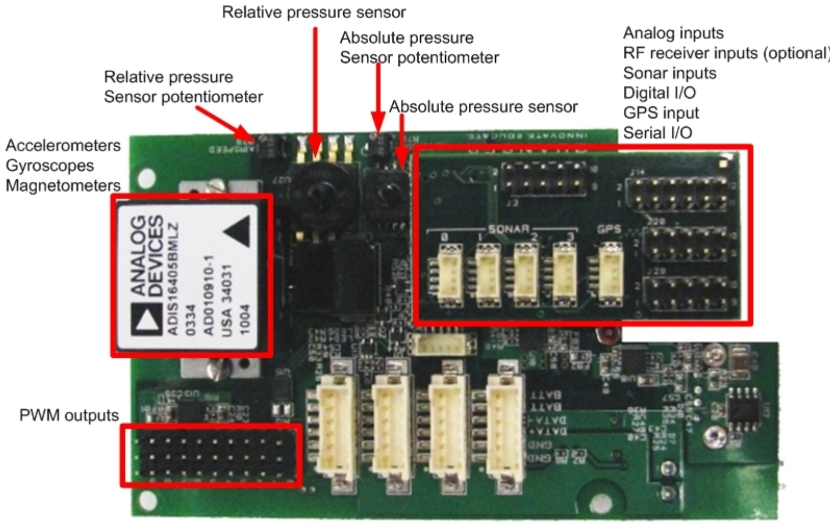


Figure 18: HiQ DAQ sensors

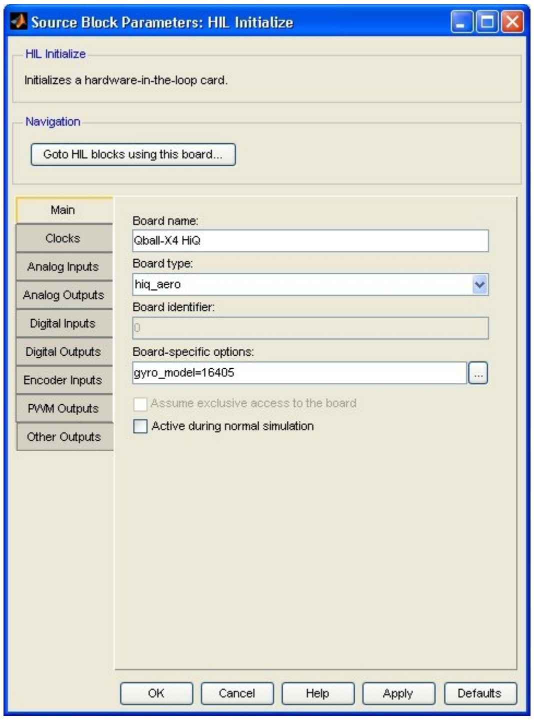


Figure 19: HIL Initialize block with the HiQ board selected.

Channel type	***Read channel numbers***	***Description***	***Units***
Analog	0-5	Analog inputs	V
Encoder	none	-	
Digital	none	-	
Other	3-6	Sonar inputs 0 to 3	m
	3000-3002	Gyroscope inputs: X, Y, Z axis	rad/s
	4000-4002	Accelerometer inputs: X, Y, Z axis	m/s^2
	8000-8002	Magnetometer inputs: X, Y, Z axis	Ga
	9000, 9001	Relative and absolute pressure sensors, respectively	Pa
	11000	Operating capacity (battery)	%
	11001	Battery level	V
	20000-20007	Receiver input channels	s

Table 5: HiQ input channels

Channel type	***Write channel numbers***	***Description***
Analog	none	-
PWM	0-9	Servo outputs 0 to 9
Digital	none	-
Other	none	-

Table 6: HiQ output channels

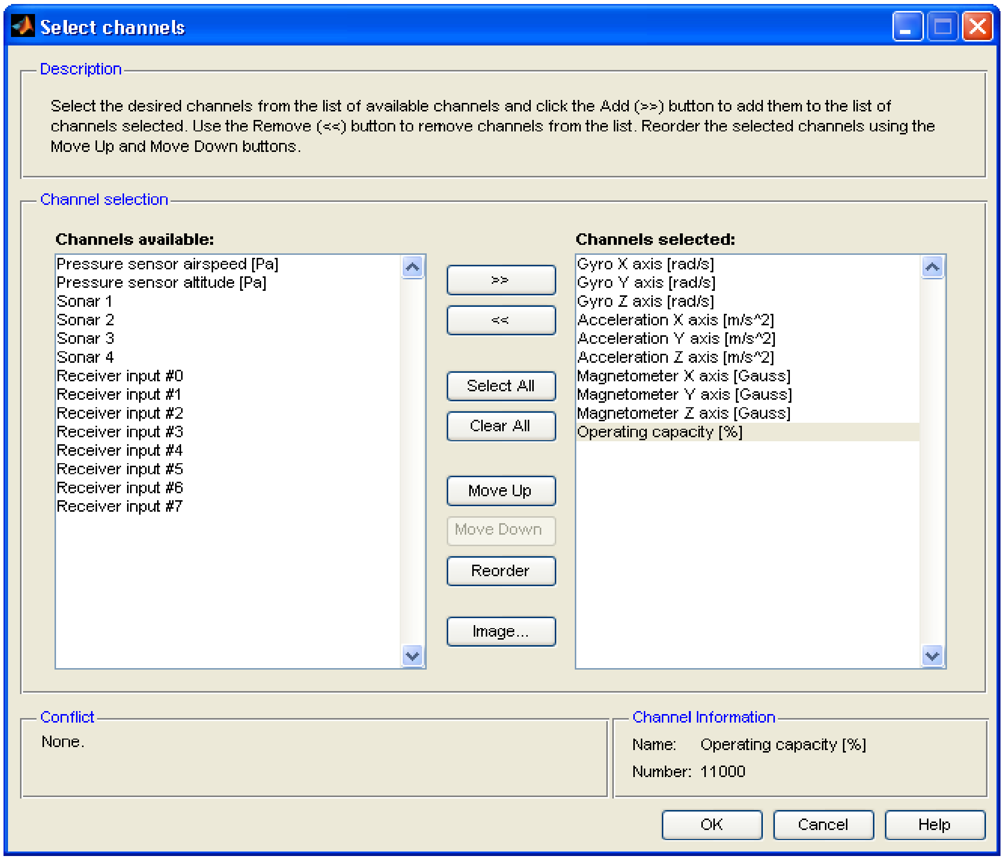


Figure 20: Channel selection dialog for the HiQ.

For the Qball-X4, PWM outputs 0-3 are used to command the four motors. The range of PWM output values is 0.05 to 0.10 (5% to 10% of a 20ms duty cycle), which corresponds to a 1ms to 2ms pulse, respectively. A command of 0.05 corresponds to zero throttle, which will cause the motors to stop. A command of 0.10 corresponds to full throttle.

The 3-axis gyroscope and accelerometer measurements are used to measure the Qball-X4 dynamics and orientation (roll, pitch and yaw). The magnetometer can be used as a digital compass to measure the Qball-X4 heading (yaw angle). These IMU inputs are crucial for controlling the flight of the Qball-X4. The Qball sonar sensor is the Maxbotix XL-Maxsonar EZ3, which measures distances between 20cm and 765cm with 1cm resolution. Objects between 0-20cm are ranged as 20cm. The sonar sensor is positioned at the bottom of the Qball and is used to measure the Qball height for closed-loop height control.

Note that the sonar works best over a hard surface which will reflect the ultrasonic signals. The sonar may not work over carpet or other surfaces that

will disperse the ultrasonic signals. Always test the sonar first by disabling the Qball motor outputs and lifting the Qball to see if the sonar is functioning as expected.

The HiQ is equipped with a relative and an absolute pressure sensor. Since atmospheric pressure varies by location, potentiometers are provided on the HiQ to adjust the pressure sensor inputs. Using a small screwdriver, adjust the potentiometers for the absolute and relative pressure sensors until the value read lies approximately in the middle of its range (the range can be found by tuning the potentiometers until the maximum and minimum values are reached). Thus, the pressure sensors will measure changes in pressure relative to the operating point set by the potentiometers.

The operating capacity input measures the battery capacity as a percentage (0-1) of the Qball-X4's input voltage operating range from 10V minimum to 20V maximum or as a direct voltage. Since the LiPo batteries used to power the Qball-X4 should be charged when they reach a voltage of no less than 10V, the operating capacity should be monitored. Figure 21 shows an example of how the operating capacity can be monitored so that a low battery warning will be displayed on the host PC if the operating capacity reaches 8% or less (corresponding to 10.8V or less) using the Show Message on Host block (found in the Simulink library under QUARC Targets\Sinks\Error Handling). Note that the HiQ input voltage measurement can be calculated according to the formula $V = 10x + 10$, where x is the operating capacity. Alternatively, the battery level input can be used to read the current battery level in volts.

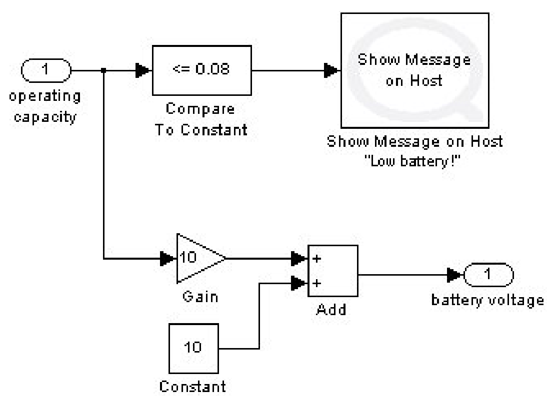


Figure 21: Monitoring the HiQ battery level.

Note that the HiQ operating capacity measures only the battery used to power the HiQ. It is recommended that the Qball-X4 batteries are always changed in pairs. Follow the directions of the charging system that is supplied to ensure the batteries are charged properly and safely.

The standard HiQ daughterboard provides several general purpose I/O channels for interfacing additional sensors. Figure 22 shows the HiQ daughterboard and its electrical pin layout. Table 7 lists the various I/O pins found on the HiQ daughterboard.

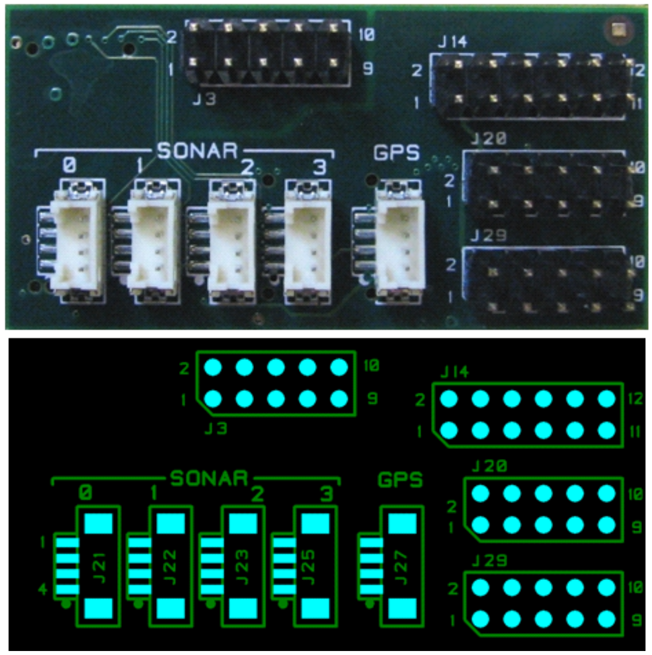


Figure 22: HiQ daughterboard and pin layout

J3	**1**	Analog GND		**J14**	**1**	GND
J3	**2**	+5V		**J14**	**2**	+3.3V
J3	**3**	Analog GND		**J14**	**3**	GND
J3	**4**	+3.3V		**J14**	**4**	PIC GPIO 0
J3	**5**	AI4		**J14**	**5**	GUM I2C SDA
J3	**6**	AI5		**J14**	**6**	GUM I2C SCL
J3	**7**	AI2		**J14**	**7**	GUM GPIO 63
J3	**8**	AI3		**J14**	**8**	GUM GPIO 65
J3	**9**	AI0		**J14**	**9**	GUM GPIO 62
J3	**10**	AI1		**J14**	**10**	GUM GPIO 59
				J14	**11**	PIC GPIO 1
				J14	**12**	GUM GPIO 64
J20	**1**	GND		**J29**	**1**	GND
J20	**2**	+3.3V		**J29**	**2**	+3.3V
J20	**3**	GUM FF RXD		**J29**	**3**	GND
J20	**4**	GUM IR TXD		**J29**	**4**	+4V
J20	**5**	GUM IR RXD		**J29**	**5**	GUM GPIO 58
J20	**6**	GUM FF CTS		**J29**	**6**	+5.5V
J20	**7**	GUM FF RTS		**J29**	**7**	GUM GPIO 60
J20	**8**	GUM BT RTS		**J29**	**8**	GUM GPIO 61
J20	**9**	GUM FF TXD		**J29**	**9**	GUM GPIO 66
J20	**10**	GUM BT CTS		**J29**	**10**	GUM GPIO 77
Sonar	**1**	GND		**GPS**	**1**	GND
Sonar	**2**	IC		**GPS**	**2**	GUM BT TXD
Sonar	**3**	TRIG		**GPS**	**3**	GUM BT RXD
Sonar	**4**	+3.3V		**GPS**	**4**	+4V

Table 7: HiQ daughterboard pin list

AI x : Analog Input channel x
PIC GPIO x : Microprocessor General Purpose Input/Output channel x
GUM GPIO x : Gumstix General Purpose I/O channel x (reconfigurable digital I/O)
GUM I2C SDA : Gumstix I2C data line
GUM I2C SCL : Gumstix I2C clock line
GUM xx RXD : Gumstix (FF/IR/BT) UART Receive Data
GUM xx TXD : Gumstix (FF/IR/BT) UART Transmit Data

GUM xx CTS : Gumstix (FF/IR/BT) UART Clear To Send
GUM xx RTS : Gumstix (FF/IR/BT) UART Request To Send
IC : Input Compare
TRIG : Trigger for sonars
Mating connector for sonar & GPS headers: Hirose DF13-4S-1.25C
Pins for sonar and GPS connectors: DF13-2630SCF

The Gumstix serial ports FF, BT, and IR are accessible through serial ports 0, 1, and 2, respectively. For more information on accessing serial data see the QUARC help under QUARC Targets/User's Guide/Communications.

The optional receiver daughterboard is shown in Figure 23 below. The sonar and GPS inputs provide the same function as the standard HiQ daughterboard. There are 8 receiver input channels that are PWM inputs configured to measure 50Hz PWM signals between 0.5ms and 2.5ms. Note the location of the ground pin for the receiver inputs and ensure that any receiver channels are connected with the corresponding orientation. The V+ pins are not powered, but are connected electrically to the external power line labeled EXT PWR +. The EXT PWR – pin is wired to the ground pins on the daughterboard. On its own, the daughterboard does not provide power to these pins, and it is not necessary to connect power to read receiver input signals. However, if you require power for the external power pins (e.g., to send power to the receiver), then a 3-pin servo cable can be connected from one of the HiQ's PWM output channels to one of the receiver daughterboard's PWM input channels (ensure that the cable connects properly the ground to ground and V+ to +6V). Doing so will effectively provide +6V to the EXT PWR pins as well as all of the daughterboard V+ pins. The HIL Read Other channels 20000 – 20007 read the PWM input channels 0 to 7 respectively in seconds.

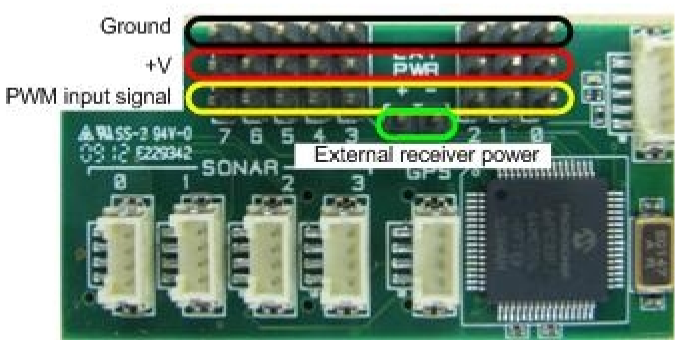


Figure 23: HiQ receiver daughterboard.

6.3. Establishing Wireless Connection

The Qball-X4 uses an ad-hoc peer-to-peer wireless TCP/IP connection for communicating with the host computer and/or other Quanser unmanned vehicles. The Qball-X4 package comes with a USB wireless adapter to setup the host computer with a wireless connection for use with the Qball-X4 and other Quanser unmanned vehicles. The Host PC and each of the vehicles must have unique IP addresses and the range of these addresses are defined below:

Host PC(s)	**182.168.1.10 to 182.168.1.19**
Quanser vehicles (Gumstix)	**182.168.1.20 to 182.168.1.254**

These steps outlined below for setting up the host computer wireless connection only need to be performed once.

1. After installing the USB wireless adapter that is provided, Windows should detect a network called GSAH, an “Unsecured Computer-to-Computer Network”. This is the peer-to-peer network used by the vehicle.
2. Open the **Status** of the wireless network, then click on Properties.
3. Under “This connection uses the following items:”, scroll down to Internet Protocol (TCP/IP), then double click on it.
4. Instead of obtaining an IP address of the computer automatically, enter the following:
   IP address: 182.168.1.10 (the Host PC wireless IP address should be set to an unused value between 182.168.1.10 and 182.168.1.19. For multiple host PCs, use different IP addresses within the valid range)
   Subnet mask: 255.255.0.0

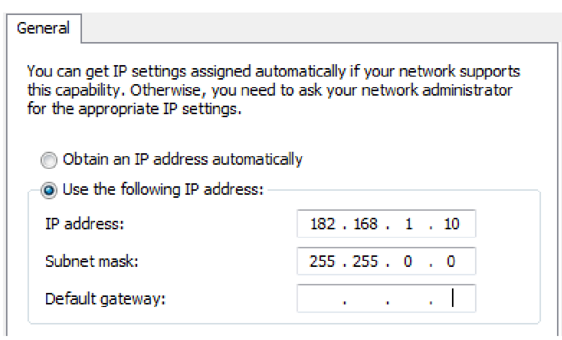


Figure 24: Wireless USB adapter settings

5. Connect to the GSAH network through the Windows network connections list.

6. If the Qball-X4 is powered on, the Qball-X4 can be pinged by typing “ping {IP of Qball-X4}” in the Run box in Windows (go to the Start menu and click Run). If the connection is successful you will see the ping replies in the command window. **Note: you may need to disable Windows firewall to establish a connection.**

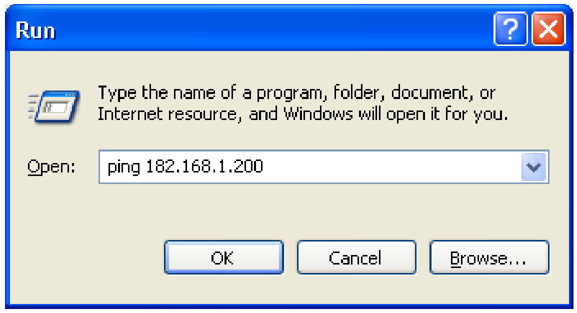


Figure 25: Pinging the Qball-X4

6.4. Configuring the model for the Qball target

Note: this section applies only to files that are run on the Gumstix target (i.e., on the Qball) such as qball_control_v2.mdl (see Table 8 in section 6.5. Simulink Files). Simulink should have a new menu item called QUARC once QUARC has been installed. The following steps are required to setup a new QUARC model for the Qball-X4:

1. Create a new Simulink model, or open an existing model to be run on the Gumstix.
2. Click on the QUARC menu, then select Options.
3. The System target file under Code Generation should be quarc_linux_verdex.tlc. Browse through the system target list to locate the proper file if necessary (Figure 26).

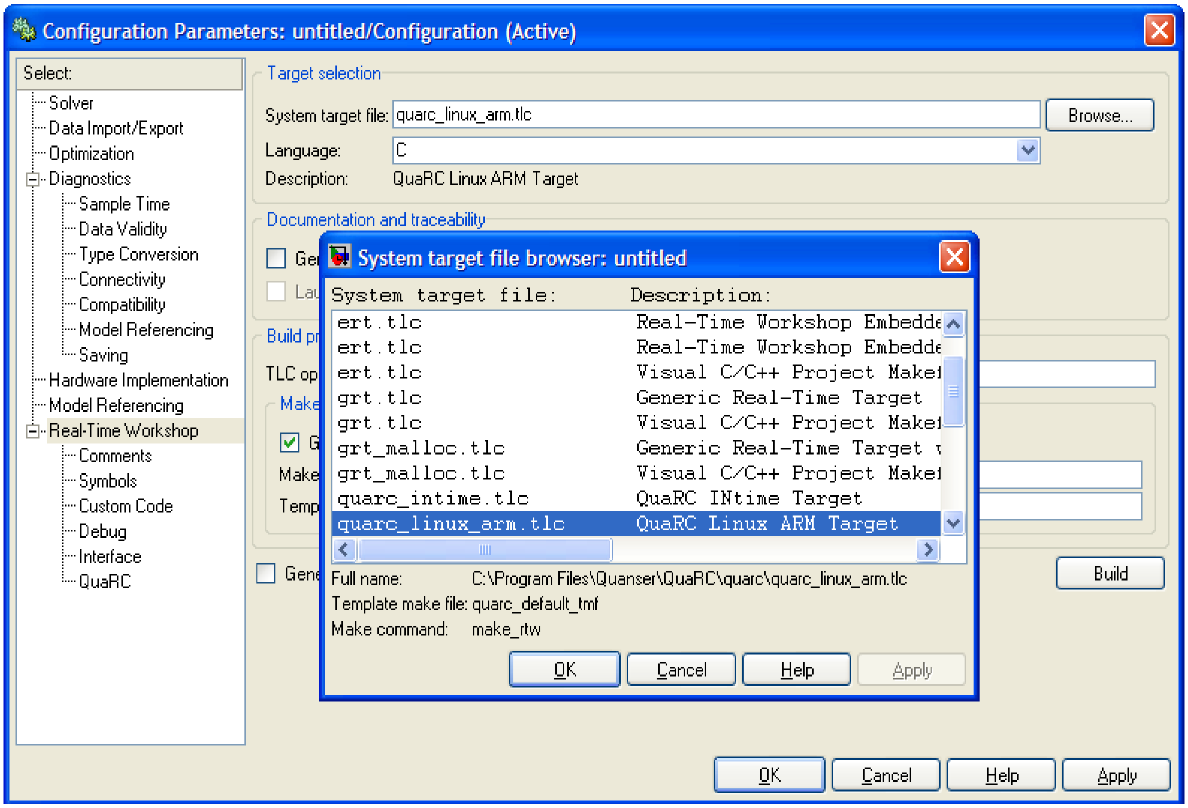


Figure 26 QUARC Option Menu.

4. In order to run the QUARC model on the target vehicle, the target's IP address must be specified. To setup the default target address for **all linux-verdex targets**, go to the QUARC menu and select Preferences. The Target type parameter should be set to linux_verdex. Replace the Default Model URI with the IP address of the desired target vehicle, e.g., “tcpip://182.168.1.200:17001” without quotes.

   Alternatively, to set the target address for the **current model only** open the model options under the QUARC/Options menu and choose Code Generation > Interface on the left hand pane. Under the MEX-file arguments, type '-w -d /tmp -uri %u', 'tcpip://{IP of Gumstix}:17001'. Include the single quotation marks (Figure 27). Replace {IP of Gumstix} with the IP of your Gumstix, e.g. 'tcpip://182.168.1.200:17001'.

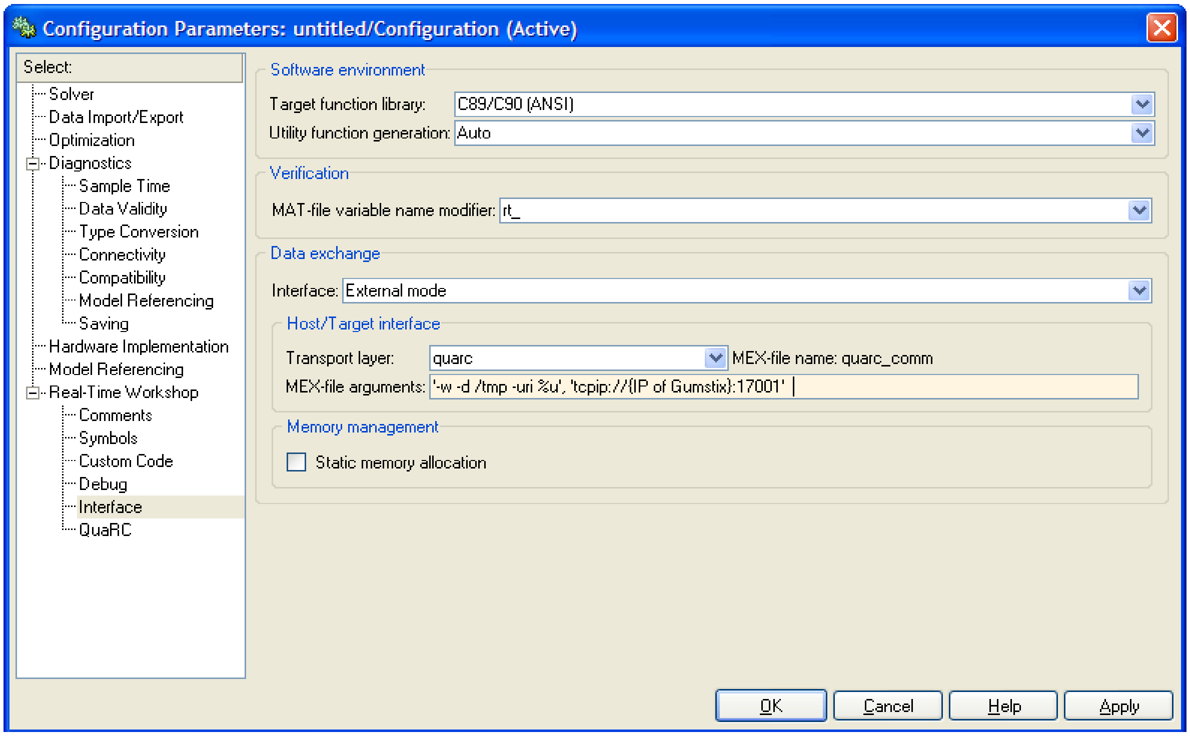


Figure 27: Configuring MEX-file arguments

5. Select "External" for simulation mode, instead of "Normal", which indicates that the model is to be run on the target machine (Gumstix) rather than simulating the model on the host machine.

6. The model is now ready to be compiled on the target vehicle. If the wireless connection to the vehicle has been established, a QUARC console can be opened to show additional messages and progress during model compilation by going to the menu item QUARC/Console for all. Building the model (QUARC/Build) will begin the code generation and compiling steps. Output from the compilation is shown in the QUARC console. This step may take a few minutes to complete.

6.5. Simulink Files

To operate the Qball there are several files needed. Table 8 lists the various files and their purpose. The joystick is an integral part of the Qball and must be used in every test flight. Even in closed-loop control modes, the joystick is still used as an enable/stop switch for safety reasons. The joystick has two control sticks. The left control is not spring activated along the vertical (up/down) direction since this controls the throttle. All other controls are spring activated. The mapping of the control sticks is shown below in Figure 28. The left stick controls the throttle and yaw (down->up is 0->100% throttle, left->right is rotate counter-clockwise -> clockwise about vertical axis) and the right stick controls pitch and roll (down->up is pitch backwards -> forwards, left-> right is roll left->right). The standard point-of-view is always with the Qball facing away from the operator, so he/she is viewing the Qball tail.

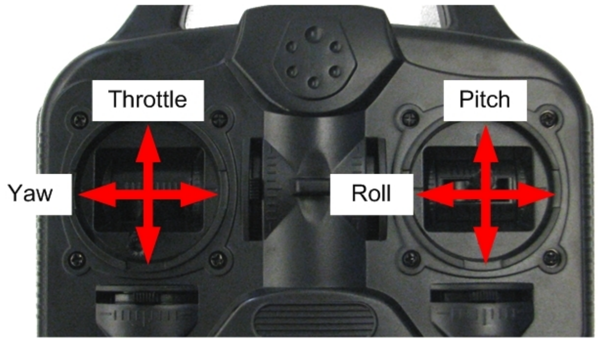


Figure 28: Joystick control mapping

Before running any Qball controllers, calibrate your joystick using the Windows game controller calibration. Build and run the appropriate Host_Joystick_xxx model for your configuration and then build and run the Host_Joystick_Test controller. This model connects to your joystick in the same way the Qball does over wifi and displays a 3D simulation (see Figure 29) of the joystick. Verify using this simulation that the joystick is responding and covering the correct range for each control stick. The Qball_X4_Joystick_Simulation_3D model can also be used to test the joystick operation with a Qball simulation (see Table 8 for a complete list of the files provided with the Qball).

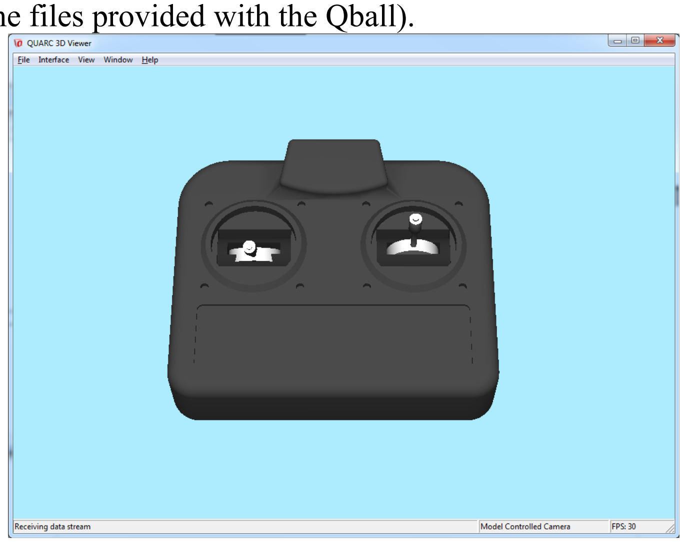


Figure 29: Host_Joystick_Test visualization

The Qball operates using a host-target structure. The host machine (ground station PC) runs one host model to stream joystick (and possibly localization) data to the target Qball. The target machine is the Qball, which executes the Qball's controller (qball_x4_control_v4.mdl). Various scripts are run upon opening the Qball controller model to initialize the controller parameters. If the MATLAB workspace is cleared and these parameters are no longer stored, simply run the setup_qball_x4.m script to reinstate the configuration parameters.

File name	*Description*
setup_qball_x4.m	A MATLAB script that is run whenever the Qball controller model is opened. This script runs several other scripts to initialize model and controller parameters. **NOTE: Make sure that you check the settings in this script and set the value of QBALL_MOTOR_TYPE to match the types of motors used on your Qball, otherwise the Qball may crash or fail to fly properly.**
filter_design.m	A script containing the properties of the complementary filter used to estimate the Qball's roll and pitch. This file is run by the setup_qball_x4.m script.
controller_design.m	A script used to compute the LQR controller gains used in stabilizing the Qball's orientation and position. This file is run by the setup_qball_x4.m script.
Host_Joystick_TYPE_A[_B].mdl	A Simulink model used on the host PC to stream joystick data to the Qball for joystick control. Use the model labeled for your joystick type.
Host_Joystick_TYPE_A[_B]_Optitrack_v4.mdl	A Simulink model used on the host PC to stream joystick and OptiTrack data to the Qball for joystick control and autonomous position control. Use the model labeled for your joystick type.
Host_Joystick_Test.mdl	Connects to a running host joystick model and displays the joystick stick controls in a 3D visualization. Use this model to verify that the joystick is properly calibrated and connected.
Qball_X4_Joystick_Simulation_3D.mdl	Simulates the Qball dynamics and displays the Qball using QUARC 3D visualization. This model connects to a running host joystick model and uses the joystick commands to fly the virtual Qball. Use this model to verify proper joystick operation and get familiar with flying the Qball simulation.
Qball_X4_Simulation_3D.mdl	This model also simulates the Qball dynamics but does not use a joystick for control. Position controllers are used to fly the Qball and track waypoints.

qball_x4_control_v4.mdl	This file contains the main flight controller for the Qball. This model is downloaded, compiled, and executed on the Qball UAV and controls the stability of the Qball orientation and, if localization information is available, the Qball position. Opening this file automatically runs the setup scripts above.

Table 8: Simulink files used to operate the Qball-X4.

Note that files highlighted in grey in Table 8 indicate that the model must be targeted and run on the quarc_linux_verdex (Gumstix) target; all other files are run on the Windows target. If you are using an older Gumstix target (quarc_linux_arm), change the target file to quarc_linux_arm.tlc under QUARC > Options > Code Generation.

6.6. The Qball-X4 Controller

The main Qball controller file contains several operating modes and various subsystems responsible for stabilizing the vehicle. Figure 30 illustrates the Qball controller's top level and the various subsystems contained in the model.

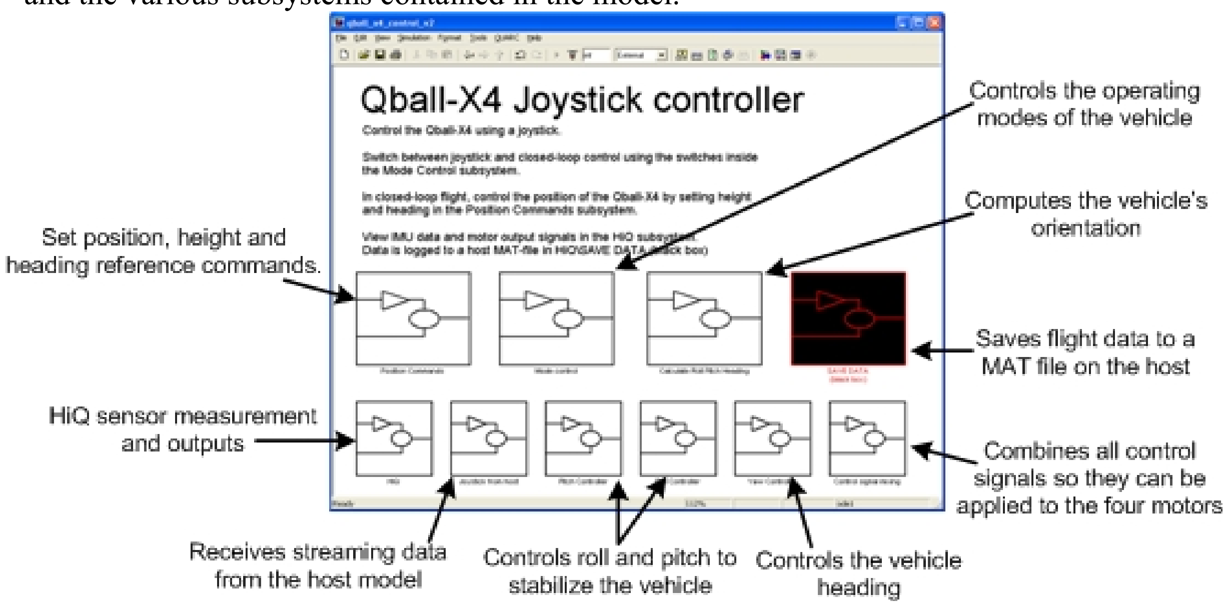
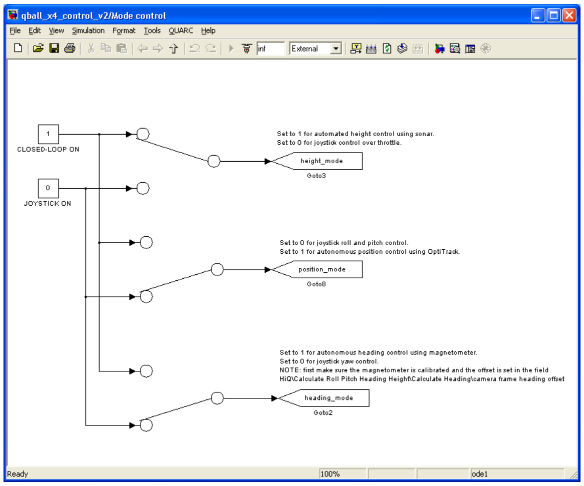


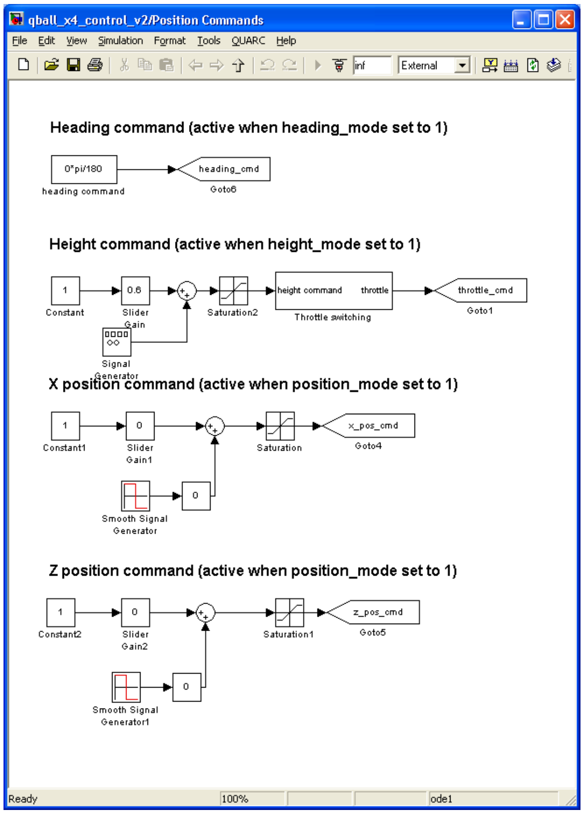
Figure 30: The main Qball-X4 controller subsystems.

The following table describes in more detail the subsystems used to command the Qball and change its operating modes.



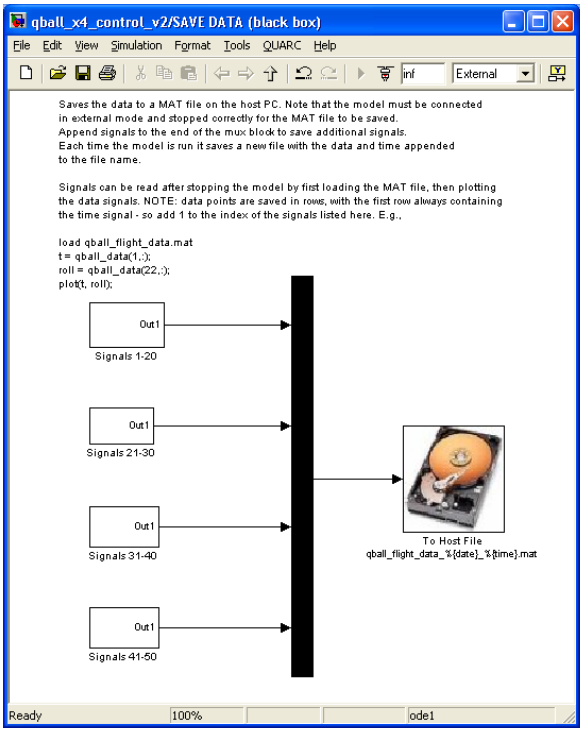
Mode control

This subsystem is used to change the operating mode of the Qball from open-loop joystick control to closed-loop autonomous control. There are 3 switches used to changes modes for height, position, and heading control. Operating modes should only be changed in between experiments when the model is not running or the system may go unstable. The closed-loop height mode allows the system to control the height using the on-board sonar sensor. The closed-loop position mode requires a localization system such as OptiTrack to control the horizontal position of the Qball. The closed-loop heading mode attempts to track the heading reference given in the Position Commands subsystem using OptiTrack localization. Joystick modes allow control of the throttle (height), roll and pitch (position), and yaw (heading) from the 4-channel joystick.



Position Commands

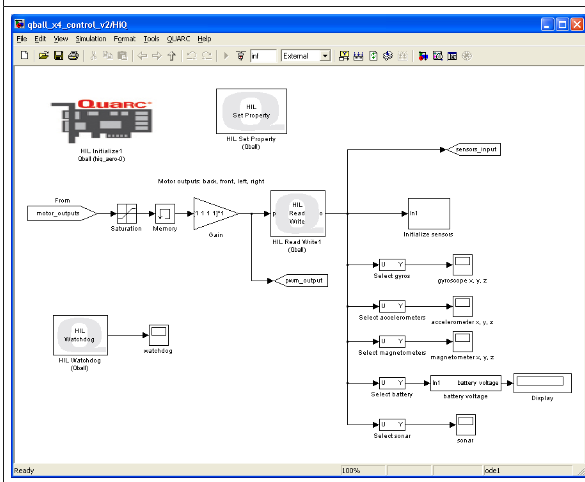
This subsystem contains the reference commands (set points) for the Qball heading, height, and horizontal position. These commands are the reference points used in closed-loop operating modes. If the Qball is operated in Joystick mode then these commands have no effect.



SAVE DATA (black box)

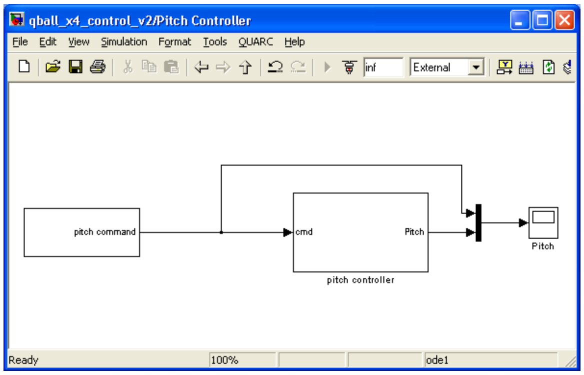
This subsystem collects data from the Qball model to save to a MAT file on the host using the QUARC To Host File block. See the help in QUARC for the To Host File block for more information.

Note that additional data from the Qball model can be added to the signals in order to save additional information as desired.



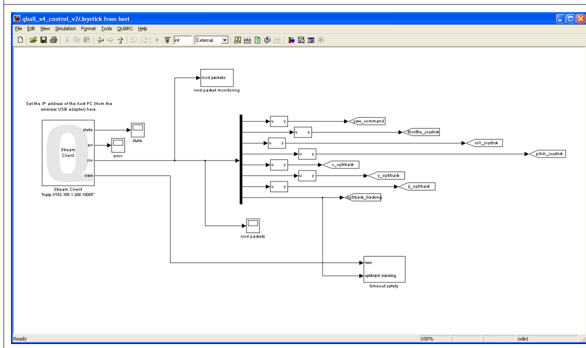
HiQ

The HiQ subsystem contains the Hardware-In-the-Loop (HIL) blocks used to configure the HiQ and read/write values. The gain block is used to enable or disable the Qball motors by multiplying the motor signals by 1 or 0, respectively. When testing the Qball sensors or handling the Qball, disable the motors for safety by multiplying the motor signals by 0 using the gain block.



Pitch/Roll Controller

These subsystems contain the reference commands (either from joystick inputs or from the Position Commands subsystem depending on the operating mode) and controller used to stabilize the roll and pitch of the Qball.



Joystick from host

This subsystem receives streaming data from the host model. **Ensure the IP address in the Stream Client URI matches the IP address of the host computer so that the Qball controller can connect to the host model.** The data packet contains joystick data and Optitrack localization data if available. If the operating mode is set to closed-loop position mode then this subsystem monitors the status of the localization tracking and issues a land if the tracking is invalid for an extended period of time. Also, a land is issued if for any reason there is no communications from the host for 1 consecutive second.

Note that even in closed-loop operating modes, the host model must still be connected and streaming joystick data to the Qball since the joystick acts as a safety switch. In closed-loop mode the joystick throttle disables the Qball flight when it is below 10% and enables takeoff and flight when the throttle is above 10%. This is a safety feature so that users can quickly disable the Qball during an experiment if desired. When disabling the Qball using the joystick, the Qball will attempt to land safely using the sonar height controller; after 4 seconds the motors will be turned off. Stopping the Qball controller directly through Simulink will force the motors to the final PWM output values set using the HIL Set Property block in the HiQ subsystem, which in this model is set to the minimum throttle value 0.05; this disables the motors immediately.

6.7. OptiTrack Localization System

When operating the Qball in closed-loop position or heading control mode, a localization system is required to provide pose feedback. In the supplied controllers, the Host_Joystick_Optitrack Simulink models stream the localization data from an OptiTrack system to the Qball.

The axes of the workspace are arranged in a specific orientation. With respect to the operator, the positive X axis points to the right, the positive Z axis points toward the operator and ground station, and the positive Y axis points upward with the origin located on the ground in the center of the workspace as shown in Figure 31. This is the workspace frame expected by the Qball controller. Typically, the workspace frame will undergo a transformation in the host model that interfaces with the localization system so that it conforms to this desired orientation.

In order to configure the localization system axes for the OptiTrack localization system, you must set the ground plane after calibrating the cameras. The ground plane setup for OptiTrack is shown in Figure 32. **Note: if you are using OptiTrack Motive, it is recommended to use version 1.5 Final as other versions may use different axis conventions.**

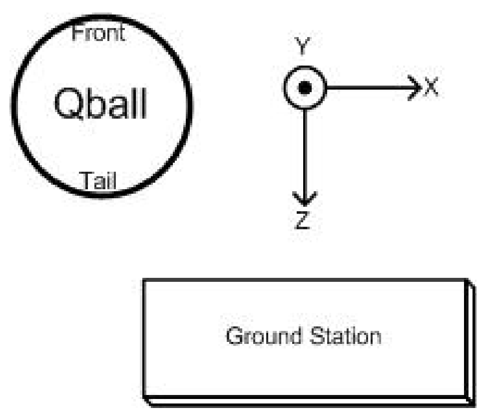


Figure 31: Localization system's coordinate frame.

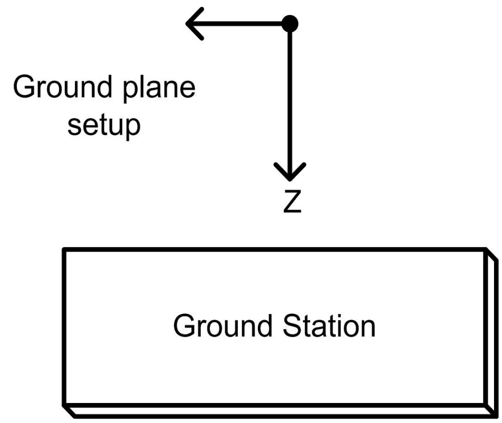


Figure 32: Ground plane setup for TrackingTools and Motive software packages.

QUARC supports several localization and tracking systems including: OptiTrack, Vicon, and PhoeniX Technologies. Make sure your chosen localization system is setup with the axes convention described above if you are using the provided closed-loop position controllers. Tracking multiple vehicles or objects requires more advanced configurations and is not covered in the supplied controllers but can be achieved by utilizing the tools provided.

The localization systems supported in QUARC typically track marker positions or trackable objects (also called rigid bodies), which are rigid geometric shapes made of 3 or more tracking markers. For operating the Qball in closed-loop modes, the OptiTrack localization system is used to provide measurements of the Qball position and orientation (heading). The closed-loop position controller assumes there is a trackable object placed on the top of the Qball's protective cage and configured such that the trackable pivot point (or center) corresponds to the position of the center of the Qball where the HiQ is located. The trackable also provides measurement of the rotation about the vertical Y axis to control the Qball heading. To setup the trackable on the Qball follow these steps:

1. Add tracking markers to the top of the Qball protective cage in a fixed, unique pattern, as shown in Figure 33. At least 3 markers must be used and additional markers provide better redundancy and robustness to occlusion.
2. Load the OptiTrack software and open the calibration file (requires that the system has already been calibrated – if not, refer to the OptiTrack Quick Start Guide).
3. Place the Qball in the workspace so that its markers are clearly visible and so that the Qball is oriented correctly with respect to the OptiTrack axes with its tail pointing in the +Z axis direction.
4. Open the Rigid Bodies view by clicking on `View > Rigid Body Properties`.

5. Using the mouse, select the markers corresponding to the Qball as shown in Figure 34.
6. Click on the Create From Selection button shown in Figure 34. The previously highlighted markers now show up as Rigid Body 1, as seen in Figure 36.
7. The pivot point of the rigid body must now be moved to the center of the Qball. Select the Qball rigid body in the left side Project Explorer pane. While the rigid body is selected, hold the CTRL key and select the top-most marker on the Qball and set it as the pivot point by right-clicking it and select Set Rigid Body Pivot Point, as shown in Figure 35. Next, select the Orientation tab in the Rigid Body view pane as shown in Figure 36. Under Translation, set the Y translation to -0.35 meters (negative 35 centimeters from the top point is the center of the Qball), as shown in Figure 37. Click Apply Translation. The pivot point will move from the top of the cage to the center.
8. Once the rigid bodies have been defined, they must be saved to a file (*.tra). This is accomplished by navigating to File > Export Rigid Bodies.

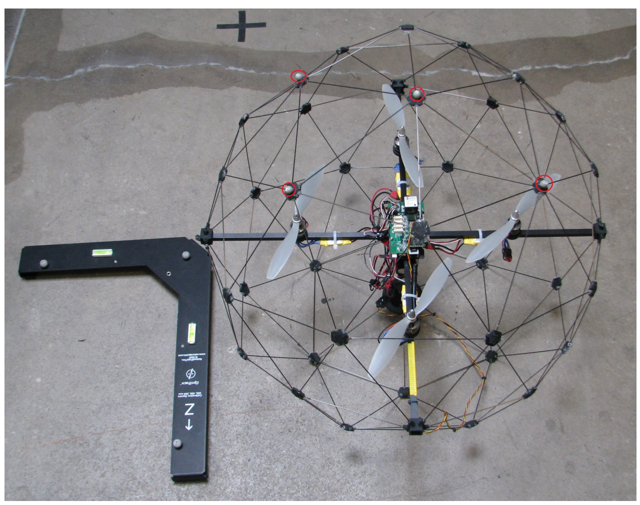


Figure 33: Qball markers placed for trackable.

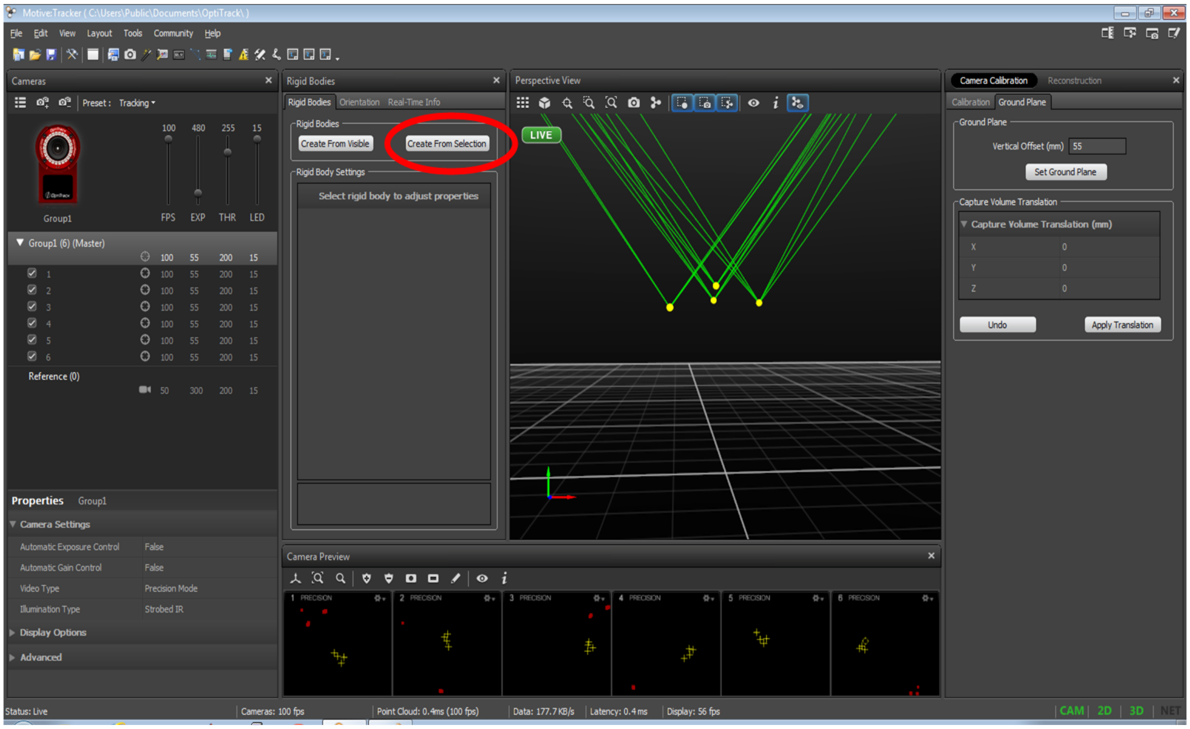


Figure 34: Select Qball markers to create a rigid body object.

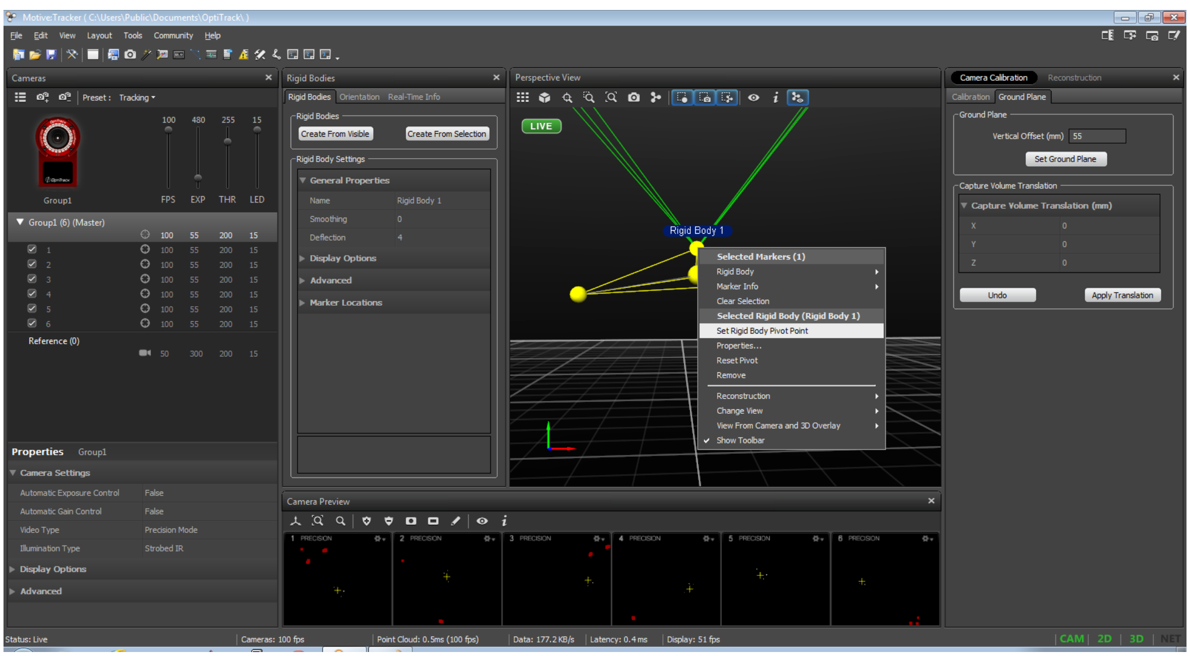


Figure 35: Set the new rigid body pivot point.

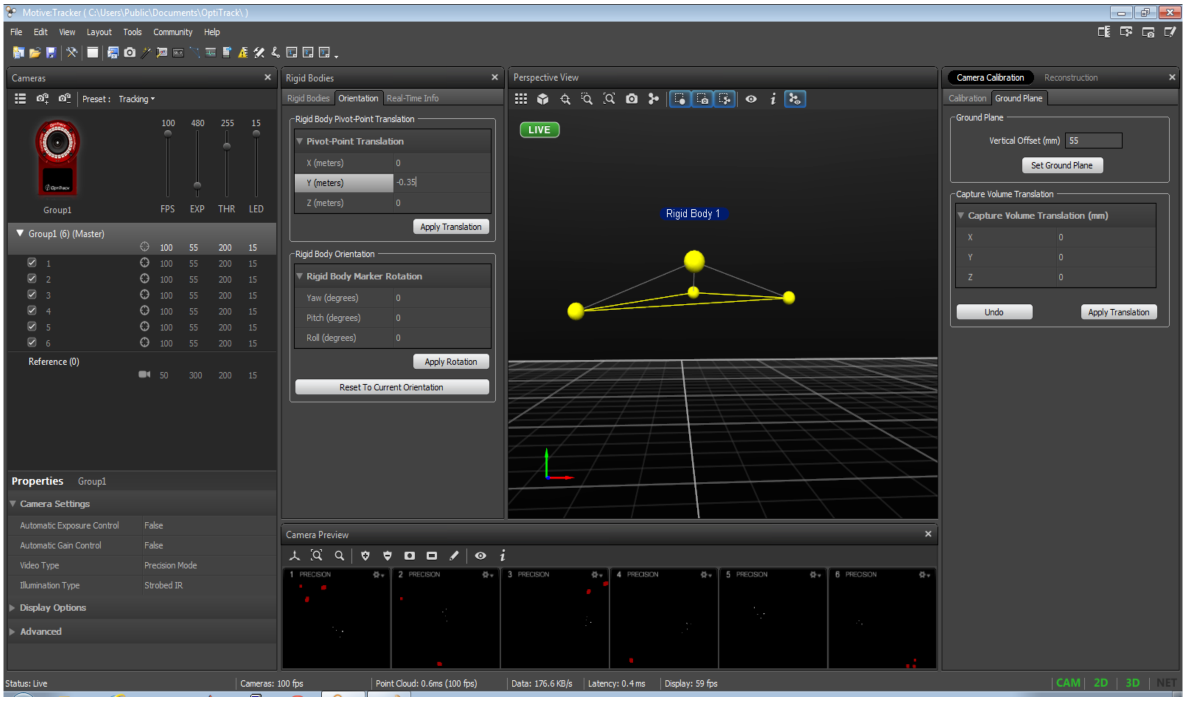


Figure 36: Apply a vertical translation to the Qball rigid body pivot point.

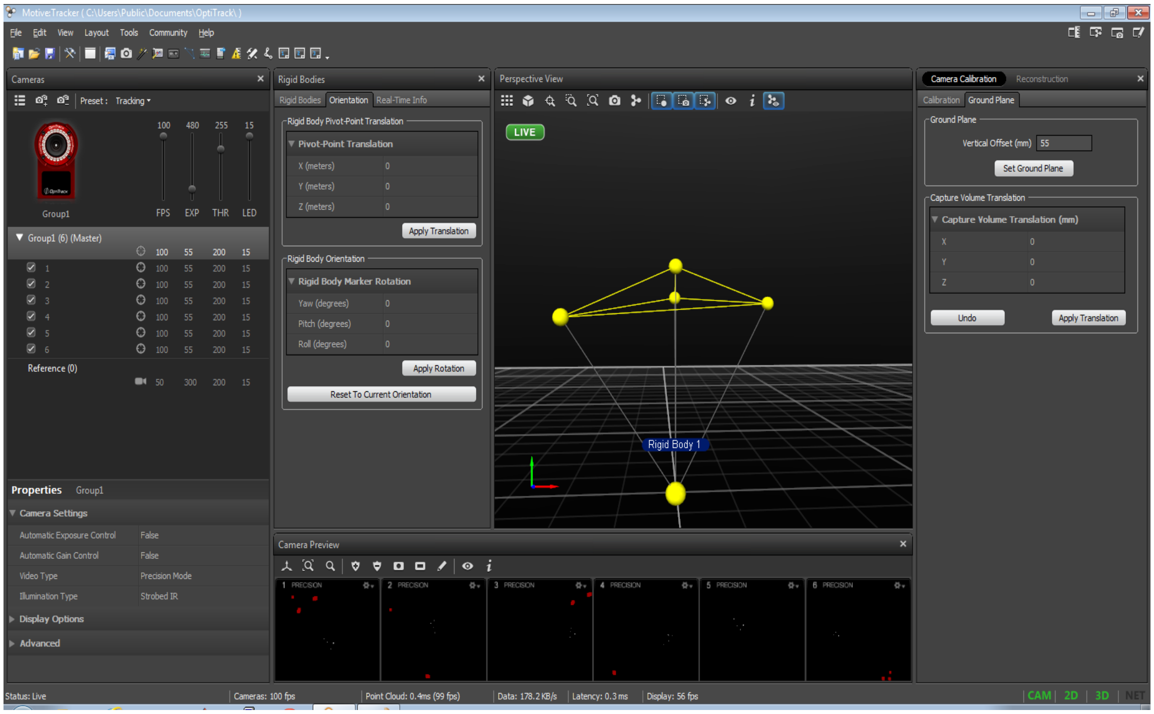


Figure 37: Qball rigid body pivot point moved to the center.

7. Charging Batteries

Before using any batteries, chargers/balancers, or power supplies, users must first read the manuals packaged with their equipment. Quanser supplies these guidelines for charging batteries but it is the users' responsibility to ensure they are operating their equipment safely and correctly. Quanser is not responsible for any damages resulting from use of batteries, power supplies, chargers, or balancers.

Before charging or using any batteries:

i) Read all instruction manuals for batteries, chargers, balancers, and power supplies
ii) Use and store system in a dry environment
iii) Do not charge under direct sunlight
iv) Do not charge battery when battery feels hot

v) Charge battery away from flammable objects
vi) Always be present when charging batteries and do not leave batteries connected to the chargers overnight.

7.1. Battery charging components

Figure 38 illustrates the hardware components supplied with the battery chargers.

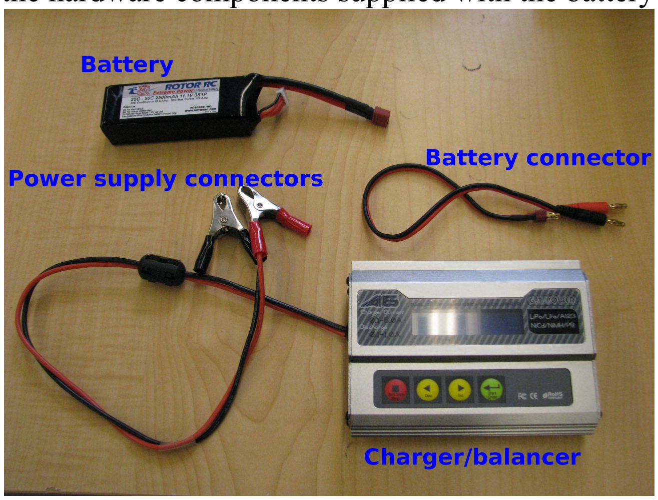


Figure 38: Battery charging components

The battery charger/balancer is supplied with either an individual power supply that connects to one charger or a shared power supply that can be connected to multiple chargers.

WARNING: Ensure that the power supply connectors do not touch at any time since this will cause a short circuit that may damage the charger and the battery.

The battery charger supplied with the system can typically charge various types of batteries. Ensure that the charger is set to charge LiPo type batteries since these are the batteries supplied with the Qball-X4. The charger contains a balancer that ensures even charging across cells within the battery. Figure 39 shows the output terminals of the charger connected to the battery cable and the balancer ports.

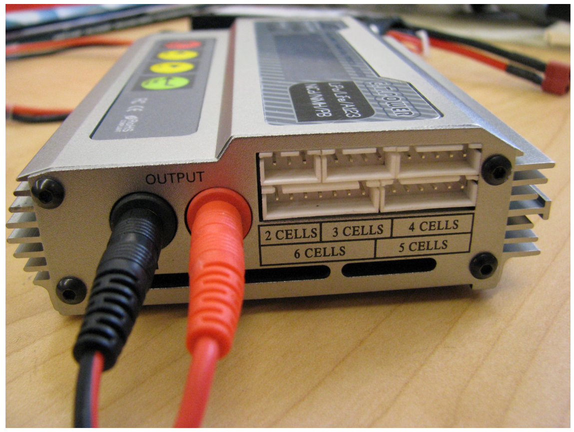


Figure 39: Battery charger output ports

7.2. Battery charging procedure

Setup procedure:

1. Ensure all prerequisites are met.
2. Connect the power supply to the charger.
3. Connect the battery connectors to the battery charger output terminals.

Charging procedure:

1. The correct number of cells and charge must be set. Set the charger to 3-cell LiPo battery, 11.1(3S), and a charge rate of 2.5 A (since the supplied batteries have a rating of 2500 mAh). Use the navigation keys to change the settings as described in the battery charger's user manual.
2. Connect the battery with the appropriate wires. Also, connect the balancer cable to the balancer. The connections are labeled according to the number of cells in the battery. Refer to your user manual for the battery details.
3. Start the charger according to the procedure described in the charger user manual. Typically this is done by holding the START/ENTER button until a beep is heard and if prompted to confirm press START again to begin charging.
4. Upon completing a full charge the charger should beep and display that the charge is complete.

8. Troubleshooting Guide

For any issue, the first and easiest troubleshooting solution on any electronic device is to reboot the device. Turn off the Qball-X4, then turn it back on again. For troubleshooting any problem with the Qball-X4, it is always a good idea to open the QUARC console in case additional information is printed to the console by going to the QUARC menu and clicking on “Console for all…”. The console must be opened after the Qball-X4 has booted and established a wifi connection. If the console is opened successfully it establishes a connection to the target and the console window has the title “QUARC Console for * at tcpip://182.168.1.xxx:17000”, where xxx corresponds to the IP address of the Qball-X4.

If you are still unable to resolve the issue after reading through this section, contact tech@quanser.com for further assistance.

8.1. The Qball has crashed! What should I do?

First, make sure that the model is stopped and the power is turned off. Do not approach the Qball if the model is still running or the propellers are turning. Upon stopping the Qball model, a saved data MAT-file is created on the host PC in the current directory. Make a backup of this saved file for review. If support is needed from Quanser, they will ask for this file so that the issue may be better diagnosed.

Load the saved file and review the data in MATLAB. The rows in the saved data file always begin with the time in row 1 and subsequent rows contain the data as ordered in the SAVE DATA subsystem in the Qball model (see 6.6. The Qball-X4 Controller).

Verify that the joystick is properly connected and calibrated by using the Host_Joystick_Test.mdl and the Qball_X4_Joystick_Simulation_3D.mdl controllers. First, calibrate your joystick in Windows using the game controller calibration procedure in Windows. Run the appropriate Host_Joystick_xxx model for your machine. Build and run the Host_Joystick_Test model and verify that the simulated joystick follows the same range of motions as the actual joystick. Next, build and run the Qball_X4_Joystick_Simulation_3D model and verify that the Qball is responding appropriately to throttle, roll, pitch, and yaw joystick commands.

8.2. The model fails to build/connect or the QUARC console does not successfully open.

1. Remove the Qball-X4 cover so that the HiQ is visible. Plug in the battery to the battery connector (Figure 9). Turn on the power switch and look at the bottom of the Gumstix (attached to the bottom of the HiQ) for the orange power LED. After approximately 30

seconds, a blue LED will flash to indicate the wifi module is powering on, and is attempting to connect to another computer on the ad-hoc wifi network. If the blue LED flashes and remains on, then the wifi module is functioning and is able to find another node on the ad-hoc wireless network. If the blue LED flashes and then turn off, the Gumstix is not able to detect other nodes (e.g., the host PC) on the ad-hoc wireless network. Check that the USB wifi adapter is inserted properly in the host PC and is configured according to the network configuration procedure outlined in this manual (Section 6.3. Establishing Wireless Connection). Verify that the host PC is connected to the wireless ad-hoc (GSAH) network and try to successfully ping the Gumstix by going to the Windows Start → Run and typing “ping 182.168.1.xxx” (without quotation marks), where xxx corresponds to the IP address of your vehicle. If the blue LED never flashes, the wireless antenna or wireless module may be disconnected. Turn off the power and verify that the Gumstix wireless module and antenna are properly connected. The wireless module shown in Figure 40 is located on the bottom side of the green Gumstix computer. Make sure the wireless module and antenna are secured and retry the above steps to establish a wireless connection.

Figure 40: Wifi Module

8.3. The Qball-X4 sensors are not being read correctly or they are stuck at some constant value.

1. Using the HIL Read block, output all possible channels. Check these outputs using scopes and displays, and determine if the problem lies with a particular sensor, or set of sensors, or if the issue is global across all sensors.

8.4. The Simulink model appears to run slowly (i.e., the simulation time runs slower than actual time), or the console displays the message "Sampling rate is too fast for base rate".

1. The maximum sample rate recommended for the Gumstix is 500 Hz (0.002 s). However, if there are complex calculations (such as image processing) performed within the model, then this could potentially limit the sample rate of the model. Try reducing the model sample rate in the menu QUARC\Options\Solver by increasing the "Fixed-step size (fundamental sample time)" parameter or change sample rates of blocks that take longer to run.

2. The HIL Read Write block should only be used once in a diagram. These blocks perform large data transfers between the HiQ and the controllers, so placing more than one of these blocks will cause multiple reads to be performed in the same sample instant, which is unnecessary. To achieve the optimal performance, use only one HIL Read Write block for the entire model.

3. To determine the execution time of blocks or subsystems within the model, use the Computation Time block found in the library under QUARC Targets\Sources\Time. This block outputs the computation time of a function call subsystem, measured using an independent high-resolution time source. Blocks can be placed inside a function call subsystem and connected to the Computation Time block to determine their execution time during each sample instant. This helps identify the bottlenecks in the model (blocks/subsystems with the highest execution time) and can identify blocks/subsystems whose computation time is greater than the sample time of the model. Try increasing the sample time of those blocks whose computation time is greater than the sample time of the model so that the blocks run in a slower rate thread.

8.5. Trying to start the Qball-X4 model results in the error "Unable to locate the dynamic link library or shared object."

1. This error indicates that the Qball-X4 driver is not found on the target. Make sure that the model target type is set to Linux Verdex by navigating to the QUARC menu QUARC\Options\Code Generation pane and changing the System target file to quarc_linux_verdex.tlc. Open a console through the QUARC menu QUARC\Console for all, and verify that the console window displays the target IP of your vehicle in the window title.

8.6. Building a model fails with the error "Not enough system resources are available to perform the operation." The hard disk is full on the Gumstix computer.

1. When several models are compiled, the disk space on the Gumstix may become full, and you will no longer have space to build models. Using the clean option in the QUARC menu under QUARC\Clean all will remove all generated code and compiled code for the current model, but this will only free up the space used by the current model. To view all models currently downloaded on the target select Manage target under the QUARC menu. The current model's target must be powered on and ready to accept a connection. The target information is displayed including all models that have been downloaded to the target. To clear all downloaded models select all the models in the list and click Remove. Note: this will only remove generated code from the target and will not delete the source models on the host PC.

8.7. Attempting to build or connect to the model results in the error message "The file could not be found".

This error could mean that there is an inconsistency in the version of QUARC installed on the host and on the target. Make sure that both host and target have the same version of QUARC installed.

This error may also indicate that an error has occurred with the uSD memory card inserted in the Gumstix. The uSD card is used to add more space to the QUARC build directory for storing compiled models. Check the Gumstix and make sure the uSD memory card is fully inserted in the Gumstix uSD card slot. To fully restore the build directory to its original state (WARNING: this will remove all generated code and compiled models from the Gumstix so the models will have to be recompiled):

1. Use PuTTY or other telnet client to log on to the Gumstix with user name "root" and password "quanser", without quotes.
2. Navigate to the build directory by typing the command "`cd /var/spool`".
3. Remove the old build directory by typing the command "`rm -rf quarc`".
4. Navigate to the uSD card: "`cd /media/card`".
5. If the quarc directory is present, remove the quarc directory from the uSD card: "`rm -rf quarc`".
6. Recreate a new quarc directory on the uSD memory card: "`mkdir quarc`".

7. Map the new build directory to the `/var/spool` directory by typing the command "`ln -s /media/card/quarc /var/spool/quarc`". Note the space between "`/media/card/quarc`" and "`/var/spool/quarc`".
8. The build directory is now ready and cleaned. Rebuild your model.