**Matlab results: example 1**

100 input-output samples.

Random input (thrust with and , angle uniform

Added gaussian noise with and as gyro noise (50 Hz).

Arbitrary initial guess.

theta true: theta init: theta optim: abs error

rod 1\_1: 1.732051 1.389267 1.732122 0.000071

rod 1\_2: 0.000000 -0.055294 -0.003793 0.003793

rod 1\_3: 0.000000 -0.806285 -0.020069 0.020069

rod 2\_1: -0.866025 -1.916043 -0.867871 0.001846

rod 2\_2: 1.224745 0.731097 1.213014 0.011731

rod 2\_3: 0.000000 0.631515 -0.002615 0.002615

pos true: pos init: pos optim: norm error:

pos\_1\_x: 0.000000 0.984122 0.028911 0.028912

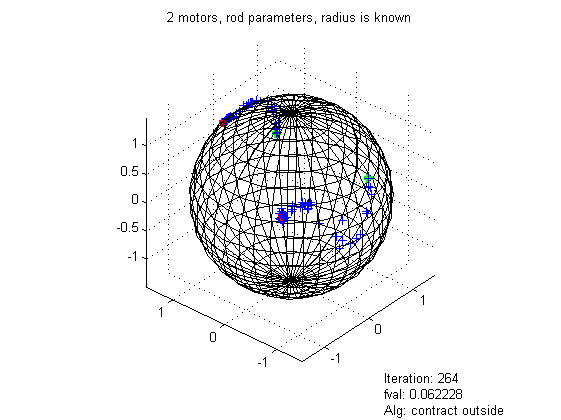
pos\_1\_y: 1.299038 1.125819 1.298819

pos\_1\_z: 0.750000 0.118479 0.749822

pos\_2\_x: -1.130534 0.256353 -1.130633 0.008862

pos\_2\_y: -0.799408 -1.272759 -0.804468

pos\_2\_z: 0.576923 0.751244 0.569649



Green: Initial points

Blue: Iteration and optimized points

Red: True points

**Matlab results: example 2**

100 input-output samples.

Random input (thrust with and , angle uniform

Added gaussian noise with and as gyro noise (50 Hz).

Arbitrary initial guess.

theta true: theta init: theta optim: abs error

quat 1\_1: 0.122395 0.130971 0.148879 0.026484

quat 1\_2: 0.698882 0.362740 0.787801 0.088919

quat 1\_3: 0.408393 0.545282 0.453791 0.045398

quat 1\_4: 0.574281 0.744268 0.641491 0.067210

radius: 1.500000 1.051434 1.195608 0.304392

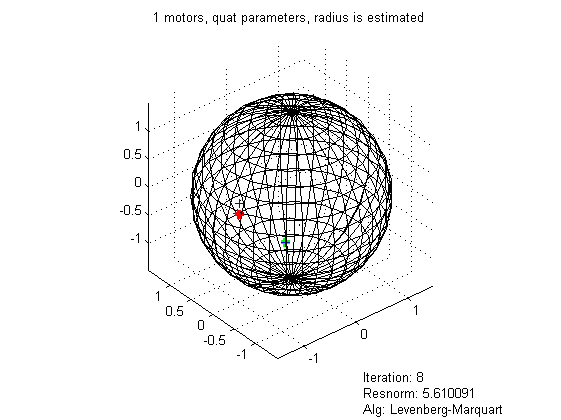
Note: Quaternion is not unit scaled (unconstraint optimization)

pos true: pos init: pos optim: norm error:

pos\_1\_x: -1.354019 -0.717901 -1.369994 0.035418

pos\_1\_y: -0.446978 -0.753515 -0.415630

pos\_1\_z: 0.465664 -0.149488 0.469733



Green: Initial points

Blue: Iteration and optimized points

Red: True points