```
# Stage model definitions for robots, sensors, etc.
#
#
#
# This file is loaded automatically by MobileSim to determine
# properties of models it creates; or it may be manually included
# into Stage world files.
#
# These settings are intended to represent typical values for the various
# models, but more importantly, agree with Aria's parameter files.
#
# Models form an inheritance tree. The base model for all robots is
# "pioneer".  The models "laser", "ranger" (sonar) and "position" (movable
# robot base) are built in to Stage.   To create a custom model definition,
# you can extend either the "pioneer" base model, or any existing model.
#
# Note, this file references laser "height". This feature has been removed
# from this version of MobileSim. It will be back.  If you want to re-enable it,
# edit the laser model source code in Stage.
#
# The shape of a robot model is given via positions of vertices [x y] of one or more
# polygons. These polygon shapes are translated into the center of the model,
# then scaled to the given size (meters), before being rendered or used for colision
# tests. Therefore, the scale or coordinate system origin of the polygon
# vertices does not matter, as long as the vertices are positioned correctly
# relative to each other.
# The Y axis is vertical positive up, X axis horizontal, positive to
# the right. The front of the robot will be along the +X axis.
# Give vertices in connected order, clockwise.
#
# For example, the vertex space is a 1x1 unit square:
#
#  Y
#  ^                              point 0 = (0.2, 0.8)
#  |                                    1 = (0.7, 0.8)
#(0,1)                 (1,1)           2 = (0.8, 0.6)
#  |                                    3 = (0.8, 0.4)
#  |   0 ------- 1                      4 = (0.7, 0.2)
```

```
#  |  /          \                    5 = (0.2, 0.2)
#  | 7            2                   6 = (0.1, 0.4)
#  | |            | Front             7 = (0.1, 0.6)
#  | 6            3
#  |  \          /
#  |   5 ------- 4
#  |
#(0,0)-----------------(1,0)--> X
#
#
# However, for most models here, the units used in the polygon shape
# are real robot dimensions (meters or mm), and then the size is also
# the real robot size (meters).
#
#
#
# All units are in meters and radians, except ranger (sonar) position angle which is
# degrees.
#



# Common parameters to all pioneers
define pioneer position (
  color "red"
  drive "diff"
  gui_nose 1                  # Indicate the front.
  gui_boundary 0              # Don't draw bounding box.
  obstacle_return 1           # Can hit things.
  laser_return 1              # Robot body seen by other lasers.
  ranger_return 1             # Seen by other sonar.
  blobfinder_return 1         # Seen by other blobfinders.
  fiducial_return 2           # Seen as "2" by other fiducial sensors.

  localization "odom"         # Change to "gps" to have impossibly perfect, global odometry
  localization_origin [0 0 0] # Start odometry at (0, 0, 0).
  odom_error [ 0.0075 0.0075 0.0075 ]   # Odometry error or slip in X, Y and Theta
                                # (Uniform random distribution), proportional
```

```
                                    # to velocity.

  # Used for position control commands (MOVE, HEAD):
  default_speed [1.5 0.0 1.3] # m(x), m(y), radians(theta)

  # Maximum limits:
  max_speed [2.0 0 1.74]   # m(x), m(y), radians(theta)

  # Acceleration:
  accel [0.3 0 1.74]        # m(x), m(y), radians(theta)
  decel [0.6 0 1.74]        # m(x), m(y), radians(theta)

  # Conversion factors for sending/recieving over client protocol
  pioneer_diffconv 0.0056
  pioneer_distconv 1.0
  pioneer_angleconv 0.001534
  pioneer_vel2div 20
  pioneer_velconv 1.0
  pioneer_rangeconv 1.0

  # Warn if no data received in this many ms. (Default is 2000 if omitted; 0 means to disable)
  # Note that this does not freeze motion as the real robot does, just displays a warning.
  #pioneer_watchdog 0
)

# SICK LMS-200 laser rangefinder configured for 32m range
define sicklms200 laser (
  range_min 0.0
  range_max 32.767
  samples 181
  fov 180.0
  color "LightBlue"
  size [0.155 0.15]
  #height 0.195 # not used.
  laser_beam_height 0.08 # approx, it actually can vary a few cm in reality.  but not used.
  laser_return 1
  ranger_return 1
  blobfinder_return 0
```

```
  fiducial_return 0
  noise 0.005 # Adds uniform random number to range value in [-0.005, 0.005] meters
  reading_angle_error 0.0007 # Adds uniform random number to angle of sample in [-0.0007,0.0007] radians

  # Rules for simulating some details of reflector detection:
  laser_return_rules 3

  # Change any reflector value greater than 1 into just 1, if it's more than 30
  # meters away:
  laser_return_rule[0].model_gt 1
  laser_return_rule[0].condition "outside_range"
  laser_return_rule[0].range 30
  laser_return_rule[0].detect 1

  # Change reflector values >1 into 1 if more than 90deg away:
  laser_return_rule[1].model_gt 1
  laser_return_rule[1].condition "outside_angle"
  laser_return_rule[1].angle 90
  laser_return_rule[1].detect 1

  # Change the specific reflector value 2 into 33 (which is the actual value the
  # real SICK returns to ARIA):
  laser_return_rule[2].model_eq 2
  laser_return_rule[2].detect 33
)

# SICK LMS-100 or LMS-111 laser configured for 1 degree resolution and 20 m
# range
define sicklms100 laser (
  range_min 0.005
  range_max 20
  samples 271
  fov 270.0
  color "LightBlue"
  size [0.1 0.1]
  laser_return 1
  ranger_return 1
  blobfinder_return 0
```

```
  fiducial_return 0
  noise 0.0075 # Adds uniform random number to range value in [-0.0075, 0.0075] meters
  reading_angle_error 0.0007 # Adds uniform random number to angle of sample in [-0.0007,0.0007] radians
)

# s300 laser
define sicks300 laser (
  range_min 0.0
  range_max 30.0
  samples 540
  fov 270.0
  color "yellow"
  size [0.102 0.105]
  laser_return 1
  ranger_return 1
  blobfinder_return 0
  fiducial_return 0
  noise 0.006
  reading_angle_error 0.0007
)

# Sonar array
define pioneerSonar ranger (
  sview [0.1 5.0 30]  # min (m), max (m), field of view (deg)
  ssize [0.01 0.04]
  laser_return 0
  blobfinder_return 0
  fiducial_return 0
  noise 0.0005   # sonar is pretty stable, actually

  # If we use projection_type "single", then the sonar is modeled
  # as a single ray projected from the center of the sonar positions.
  projection_type "single"

  # If we use projection_type "closest" then you can get slightly
  # more complex sonar behavior, which can be tuned with these parameters:
  #projection_type "closest"
  #projection_res 6      # Test a sensor's field of view at a resolution of 6 degrees
```

```
  #enable_throwaway 1
  #throwaway_thresh 0.4  # Test range delta to consider throwing reading away
  #throwaway_prob 0.8    # Probability of throwing a sensor reading away


  # These values are used in the config packet sent back to the client. How many
  # sonar sensors are simulated and their locations are specified separately
  # with the scount and spose properties (see individual model definitions
  # below).
  pioneer_hasfrontarray 1
  pioneer_hasreararray 1

  # This parameter acts as a crude stand-in for the delayed timing of
  # real sonar, which is not simulated -- the client program  will recieve
  # sonar values more slowly though the range data will not be old.
  # Note, if this value is too large (and the robot has many many sonar),
  # then an oversized  packet will be sent, which could crash the client
  # program.
  #pioneer_max_readings_per_packet 4
)


# Example model based on pioneerSonar where only 8 front sonar
# sensors are present on a P3/2 DX, AT or PeopleBot.
define frontP3Sonar pioneerSonar (
  pioneer_hasfrontarray 1
  pioneer_hasreararray 0
  scount 8
  spose[0] [0.024 0.119 50]
  spose[1] [0.058 0.078 30]
  spose[2] [0.077 0.027 10]
  spose[3] [0.077 -0.027 -10]
  spose[4] [0.058 -0.078 -30]
  spose[5] [0.024 -0.119 -50]
  spose[6] [-0.02 -0.136 -90]
  spose[7] [-0.191 -0.136 -90]
)
```

```
# Bumper array for DX (one half)
# For future use. Bumpers aren't implemented yet in MobileSim.
#define P3DXBumperRing bumpswitches (
#  color "black"
#  bumpcount 5
#  bumpsize [0.100 0.015]
#  bumppose[4] [0.23739 0 0]
#  laser_return 0
#  blobfinder_return 0
#  fiducial_return 0
#  ranger_return 0
#  #height 0.06
#)

# Bumper array for AT (one half)
#define P3ATBumperRing bumpswitches (
#  color "black"
#  bumpcount 5
#  bumpsize [0.100 0.015]
#  bumppose[4] [0.23739 0 0]
#  laser_return 0
#  blobfinder_return 0
#  fiducial_return 0
#  ranger_return 0
#  #height 0.06
#)

# Bumper array for PowerBot (one half)
#define powerbotBumperRing bumpswitches (
#  color "black"
#  bumpcount 5
#  bumpsize [0.100 0.015]
#  bumppose[4] [0.23739 0 0]
#  laser_return 0
#  blobfinder_return 0
#  fiducial_return 0
#  ranger_return 0
#  #height 0.06
```

```
#)

# Bumper array for patrolbot (all)
#define patrolbotBumpers bumpswitches (
#  color "grey"
#  bumpcount 12
#  bumpsize [0.090 0.010]
#  bumppose[3] [0.27 0.045 0.175]
#  laser_return 1
#  blobfinder_return 0
#  fiducial_return 0
#  ranger_return 1
#)


# Model for a amigo differential-drive robot base with sonar.
define amigo pioneer (
  pioneer_robot_subtype "amigo"

  # Speed profile:
  max_speed [0.5 0 0.87]
  accel [0.3 0 0.87]
  decel [0.75 0 2.5]

  # Body shape:
  origin [-0.01 0.0 0.0]
  size [0.33 0.279]
  polygons 1
  polygon[0].points 8
  polygon[0].point[0] [-0.1 0.165]
  polygon[0].point[1] [0.1 0.165]
  polygon[0].point[2] [0.1395 0.1]
  polygon[0].point[3] [0.1395 -0.1]
  polygon[0].point[4] [0.1 -0.165]
  polygon[0].point[5] [-0.1 -0.165]
  polygon[0].point[6] [-0.1395 -0.1]
  polygon[0].point[7] [-0.1395 0.1]
  polygon[0].filled 1
```

```
  # height off floor:
  #height 0.15 # Uncomment this to enable

  # client conversion factors
  pioneer_diffconv 0.011
  pioneer_distconv 0.5083
  pioneer_velconv 0.6154
  pioneer_angleconv 0.001534
  pioneer_vel2div 20

  # Sonar:
  pioneerSonar (
    scount 8
    spose[0] [0.07 0.1 90]
    spose[1] [0.12 0.075 41]
    spose[2] [0.144 0.03 15]
    spose[3] [0.144 -0.03 -15]
    spose[4] [0.12 -0.075 -41]
    spose[5] [0.07 -0.1 -90]
    spose[6] [-0.146 -0.058 -145]
    spose[7] [-0.146 0.058 145]
  )
)

define amigo-sh amigo (
  max_speed [0.75 0 0.87]
  pioneer_robot_subtype "amigo-sh"
  pioneer_diffconv 0.011
  pioneer_distconv 1
  pioneer_velconv 1
  pioneer_angleconv 0.001534
  pioneer_vel2div 20
)
```

```
# Model for a p2at differential-drive robot base with sonar.
define p2at pioneer (
  pioneer_robot_subtype "p2at"

  # Speed profile:
  max_speed [0.75 0 1.74]

  # Body shape:
  size [0.626 0.505]
  polygons 1
  polygon[0].points 8
  polygon[0].point[0] [-0.12 0.313]
  polygon[0].point[1] [0.12 0.313]
  polygon[0].point[2] [0.2525 0.12]
  polygon[0].point[3] [0.2525 -0.12]
  polygon[0].point[4] [0.12 -0.313]
  polygon[0].point[5] [-0.12 -0.313]
  polygon[0].point[6] [-0.2525 -0.12]
  polygon[0].point[7] [-0.2525 0.12]
  polygon[0].filled 1

  #height 0.27724 # Uncomment to enable

  # client protocol conversion factors
  pioneer_diffconv 0.0034
  pioneer_distconv 1.32
  pioneer_rangeconv 0.268

  # Sonar:
  pioneerSonar (
    scount 16
    spose[0] [0.147 0.136 90]
    spose[1] [0.193 0.119 50]
    spose[2] [0.227 0.079 30]
    spose[3] [0.245 0.027 10]
    spose[4] [0.245 -0.027 -10]
    spose[5] [0.227 -0.079 -30]
    spose[6] [0.193 -0.119 -50]
```

```
    spose[7] [0.147 -0.136 -90]
    spose[8] [-0.144 -0.136 -90]
    spose[9] [-0.189 -0.119 -130]
    spose[10] [-0.223 -0.079 -150]
    spose[11] [-0.241 -0.027 -170]
    spose[12] [-0.241 0.027 170]
    spose[13] [-0.223 0.079 150]
    spose[14] [-0.189 0.119 130]
    spose[15] [-0.144 0.136 90]
  )
  sicklms200( pose [0.16 0.007 0] )
)



# Model for a p2ce differential-drive robot base with sonar.
define p2ce pioneer (
  pioneer_robot_subtype "p2ce"

  # Speed profile:
  max_speed [0.75 0 1.74]

  # Body shape:
  origin [-0.04465 0.0 0.0]
  size [0.511 0.4]
  polygons 1
  polygon[0].points 8
  polygon[0].point[0] [-0.12 0.2555]
  polygon[0].point[1] [0.12 0.2555]
  polygon[0].point[2] [0.2 0.12]
  polygon[0].point[3] [0.2 -0.12]
  polygon[0].point[4] [0.12 -0.2555]
  polygon[0].point[5] [-0.12 -0.2555]
  polygon[0].point[6] [-0.2 -0.12]
  polygon[0].point[7] [-0.2 0.12]
  polygon[0].filled 1
```

```
  # Height of top plate from floor (m):
  #height 0.23711

  # client protocol conversion factors
  pioneer_diffconv 0.0057
  pioneer_distconv 0.826
  pioneer_rangeconv 0.268

  # Sonar:
  pioneerSonar (
    scount 16
    spose[0] [0.069 0.136 90]
    spose[1] [0.114 0.119 50]
    spose[2] [0.148 0.078 30]
    spose[3] [0.166 0.027 10]
    spose[4] [0.166 -0.027 -10]
    spose[5] [0.148 -0.078 -30]
    spose[6] [0.114 -0.119 -50]
    spose[7] [0.069 -0.136 -90]
    spose[8] [-0.157 -0.136 -90]
    spose[9] [-0.203 -0.119 -130]
    spose[10] [-0.237 -0.078 -150]
    spose[11] [-0.255 -0.027 -170]
    spose[12] [-0.255 0.027 170]
    spose[13] [-0.237 0.078 150]
    spose[14] [-0.203 0.119 130]
    spose[15] [-0.157 0.136 90]
  )
  sicklms200( pose [0 0 0] )
)



# Model for a p2d8 differential-drive robot base with sonar.
define p2d8 pioneer (
  pioneer_robot_subtype "p2d8"

  # Speed profile:
```

```
max_speed [0.75 0 1.74]

# Body shape:
origin [-0.04465 0.0 0.0]
size [0.511 0.4]
polygons 1
polygon[0].points 8
polygon[0].point[0] [-0.12 0.2555]
polygon[0].point[1] [0.12 0.2555]
polygon[0].point[2] [0.2 0.12]
polygon[0].point[3] [0.2 -0.12]
polygon[0].point[4] [0.12 -0.2555]
polygon[0].point[5] [-0.12 -0.2555]
polygon[0].point[6] [-0.2 -0.12]
polygon[0].point[7] [-0.2 0.12]
polygon[0].filled 1

# Height of top plate from floor (m):
#height 0.23711

# client protocol conversion factors
# same as base pioneer model

# Sonar:
pioneerSonar (
  scount 16
  spose[0] [0.069 0.136 90]
  spose[1] [0.114 0.119 50]
  spose[2] [0.148 0.078 30]
  spose[3] [0.166 0.027 10]
  spose[4] [0.166 -0.027 -10]
  spose[5] [0.148 -0.078 -30]
  spose[6] [0.114 -0.119 -50]
  spose[7] [0.069 -0.136 -90]
  spose[8] [-0.157 -0.136 -90]
  spose[9] [-0.203 -0.119 -130]
  spose[10] [-0.237 -0.078 -150]
  spose[11] [-0.255 -0.027 -170]
```

```
    spose[12] [-0.255 0.027 170]
    spose[13] [-0.237 0.078 150]
    spose[14] [-0.203 0.119 130]
    spose[15] [-0.157 0.136 90]
  )
  sicklms200( pose [0.018 0 0] )
)



# Model for a p2de differential-drive robot base with sonar.
define p2de pioneer (
  pioneer_robot_subtype "p2de"

  # Speed profile:
  max_speed [0.75 0 1.74]

  # Body shape:
  origin [-0.04465 0.0 0.0]
  size [0.511 0.4]
  polygons 1
  polygon[0].points 8
  polygon[0].point[0] [-0.12 0.2555]
  polygon[0].point[1] [0.12 0.2555]
  polygon[0].point[2] [0.2 0.12]
  polygon[0].point[3] [0.2 -0.12]
  polygon[0].point[4] [0.12 -0.2555]
  polygon[0].point[5] [-0.12 -0.2555]
  polygon[0].point[6] [-0.2 -0.12]
  polygon[0].point[7] [-0.2 0.12]
  polygon[0].filled 1

  #height 0.23711

  # client protocol conversion factors
  pioneer_rangeconv 0.268
  pioneer_distconv 0.969
```

```
  # Sonar:
  pioneerSonar (
    scount 16
    spose[0] [0.069 0.136 90]
    spose[1] [0.114 0.119 50]
    spose[2] [0.148 0.078 30]
    spose[3] [0.166 0.027 10]
    spose[4] [0.166 -0.027 -10]
    spose[5] [0.148 -0.078 -30]
    spose[6] [0.114 -0.119 -50]
    spose[7] [0.069 -0.136 -90]
    spose[8] [-0.157 -0.136 -90]
    spose[9] [-0.203 -0.119 -130]
    spose[10] [-0.237 -0.078 -150]
    spose[11] [-0.255 -0.027 -170]
    spose[12] [-0.255 0.027 170]
    spose[13] [-0.237 0.078 150]
    spose[14] [-0.203 0.119 130]
    spose[15] [-0.157 0.136 90]
  )
  sicklms200( pose [0.017 0.008 0] )
)


# Model for a p3at differential-drive robot base with sonar.
define p3at pioneer (
  pioneer_robot_subtype "p3at"

  # Speed profile:
  max_speed [0.6 0 0.75]

  # Body shape:
  size [0.626 0.505]
  polygons 1
  polygon[0].points 8
  polygon[0].point[0] [-0.18 0.313]
  polygon[0].point[1] [0.18 0.313]
```

```
  polygon[0].point[2] [0.2525 0.18]
  polygon[0].point[3] [0.2525 -0.18]
  polygon[0].point[4] [0.18 -0.313]
  polygon[0].point[5] [-0.18 -0.313]
  polygon[0].point[6] [-0.2525 -0.18]
  polygon[0].point[7] [-0.2525 0.18]
  polygon[0].filled 1

  #height 0.27724

  # client protocol conversion factors
  pioneer_diffconv 0.0034
  pioneer_distconv 0.465
  pioneer_angleconv 0.001534
  pioneer_velconv 1.0

  # Sonar:
  pioneerSonar (
    scount 16
    spose[0] [0.147 0.136 90]
    spose[1] [0.193 0.119 50]
    spose[2] [0.227 0.079 30]
    spose[3] [0.245 0.027 10]
    spose[4] [0.245 -0.027 -10]
    spose[5] [0.227 -0.079 -30]
    spose[6] [0.193 -0.119 -50]
    spose[7] [0.147 -0.136 -90]
    spose[8] [-0.144 -0.136 -90]
    spose[9] [-0.189 -0.119 -130]
    spose[10] [-0.223 -0.079 -150]
    spose[11] [-0.241 -0.027 -170]
    spose[12] [-0.241 0.027 170]
    spose[13] [-0.223 0.079 150]
    spose[14] [-0.189 0.119 130]
    spose[15] [-0.144 0.136 90]
  )
  sicklms200( pose [0.125 0 0] )
)
```

```
define p3at-sh p3at (
  pioneer_robot_subtype "p3at-sh"
  pioneer_distconv 1.0
  pioneer_diffconv 0.0034
  pioneer_velconv 1.0
)

define p3atiw p3at (
  pioneer_robot_subtype "p3atiw"
  size [0.626 0.49]
)

define p3atiw-sh p3at-sh (
  pioneer_robot_subtype "p3atiw-sh"
  size [0.626 0.49]
)



# Model for a p3dx differential-drive robot base with sonar.
define p3dx-nolaser pioneer (
  pioneer_robot_subtype "p3dx"

  # Speed profile:
  max_speed [1.0 0 1.74]

  # Body shape:
  origin [-0.04465 0.0 0.0]
  size [0.511 0.4]
  polygons 1
  polygon[0].points 8
  polygon[0].point[0] [-0.12 0.2555]
  polygon[0].point[1] [0.12 0.2555]
  polygon[0].point[2] [0.2 0.12]
  polygon[0].point[3] [0.2 -0.12]
  polygon[0].point[4] [0.12 -0.2555]
  polygon[0].point[5] [-0.12 -0.2555]
```

```
  polygon[0].point[6] [-0.2 -0.12]
  polygon[0].point[7] [-0.2 0.12]
  polygon[0].filled 1

  # Height of top plate from floor (m):
  #height 0.23711

  # client protocol conversion factors
  pioneer_distconv 0.485
  pioneer_diffconv 0.0056
  pioneer_velconv  1.0

  # Sonar:
  pioneerSonar (
    scount 16
    spose[0] [0.069 0.136 90]
    spose[1] [0.114 0.119 50]
    spose[2] [0.148 0.078 30]
    spose[3] [0.166 0.027 10]
    spose[4] [0.166 -0.027 -10]
    spose[5] [0.148 -0.078 -30]
    spose[6] [0.114 -0.119 -50]
    spose[7] [0.069 -0.136 -90]
    spose[8] [-0.157 -0.136 -90]
    spose[9] [-0.203 -0.119 -130]
    spose[10] [-0.237 -0.078 -150]
    spose[11] [-0.255 -0.027 -170]
    spose[12] [-0.255 0.027 170]
    spose[13] [-0.237 0.078 150]
    spose[14] [-0.203 0.119 130]
    spose[15] [-0.157 0.136 90]
  )
)

define p3dx p3dx-nolaser (
  sicklms200( pose [0 0 0] )
)
```

```
define p3dx-noerr p3dx (
  odom_error [0 0 0]
)

define p3dx-sh p3dx (
  pioneer_robot_subtype "p3dx-sh"
  pioneer_diffconv 0.0056
  pioneer_distconv 1.0
)

define p3dx-with-rear-laser p3dx (
  sicklms200(pose [-0.155 0 180])
)


# Model for a patrolbot differential-drive robot base with sonar.
define patrolbot-sh pioneer (
  color "grey"
  pioneer_robot_subtype "patrolbot-sh"

  # Speed profile:
  max_speed [2.0 0 3.49]

  # Body shape:
  size [0.521 0.435]
  polygons 1
  polygon[0].points 8
  polygon[0].point[0] [-0.13 0.2605]
  polygon[0].point[1] [0.13 0.2605]
  polygon[0].point[2] [0.2175 0.13]
  polygon[0].point[3] [0.2175 -0.13]
  polygon[0].point[4] [0.13 -0.2605]
  polygon[0].point[5] [-0.13 -0.2605]
  polygon[0].point[6] [-0.2175 -0.13]
  polygon[0].point[7] [-0.2175 0.13]
  polygon[0].filled 1

  #height 0.38
```

```
  # client protocol conversion factors
  pioneer_diffconv 0.0056
  pioneer_distconv 1.0
  pioneer_angleconv 0.001534

  # Sonar:
  pioneerSonar (
    scount 16
    spose[0] [0.083 0.229 90]
    spose[1] [0.169 0.202 55]
    spose[2] [0.232 0.134 30]
    spose[3] [0.263 0.046 10]
    spose[4] [0.263 -0.046 -10]
    spose[5] [0.232 -0.134 -30]
    spose[6] [0.169 -0.202 -55]
    spose[7] [0.083 -0.229 -90]
    spose[8] [-0.083 -0.229 -90]
    spose[9] [-0.169 -0.202 -125]
    spose[10] [-0.232 -0.134 -150]
    spose[11] [-0.263 -0.046 -170]
    spose[12] [-0.263 0.046 170]
    spose[13] [-0.232 0.134 150]
    spose[14] [-0.169 0.202 125]
    spose[15] [-0.083 0.229 90]
  )
  sicklms200(
    pose [0.037 0 0]

    # Mounted upside down, inside the body:
    reverse_scan 1
    laser_beam_height 0.115
    height_offset -0.20
  )
)

define patrolbot patrolbot-sh (
     # lets us say patrolbot as shorthand for patrolbot-sh
```

```
)


define mt400 patrolbot-sh (
  # It's similar to the older patrolbot-sh but with different batteries and
  # correct speed limits thta match firmware config
  pioneer_robot_subtype "mt400"
  pioneer_batterytype 2
  max_speed [2.2 0 8.72]
)

define mt400-500 mt400 (
  pioneer_batterytype 2
)

define mt400-600 mt400 (
  pioneer_batterytype 0
)

define mt400-260 mt400 (
  pioneer_batterytype 2
)

define mt400-261 mt400 (
  pioneer_batterytype 0
)

define researchPB mt400-600 (
  pioneer_robot_subtype "researchPB"
)

define research-patrolbot researchPB (
  # More descriptive name
)


# Model for a peoplebot differential-drive robot base with sonar.
define peoplebot-sh pioneer (
```

```
pioneer_robot_subtype "peoplebot-sh"
color "grey"

# Speed profile:
max_speed [0.75 0 1.74]

# Body shape:
origin [-0.04465 0.0 0.0]
size [0.513 0.425]
polygons 1
polygon[0].points 8
polygon[0].point[0] [-0.12 0.2565]
polygon[0].point[1] [0.12 0.2565]
polygon[0].point[2] [0.2 0.12]
polygon[0].point[3] [0.2 -0.12]
polygon[0].point[4] [0.12 -0.2565]
polygon[0].point[5] [-0.12 -0.2565]
polygon[0].point[6] [-0.2 -0.12]
polygon[0].point[7] [-0.2 0.12]
polygon[0].filled 1

# height of upper top plate from floor:
#height 1.115

# client protocol conversion factors
pioneer_diffconv 0.006
pioneer_distconv 1.0
pioneer_angleconv 0.001534

# Sonar:
pioneerSonar (
  scount 24
  spose[0] [0.069 0.136 90]
  spose[1] [0.114 0.119 50]
  spose[2] [0.148 0.078 30]
  spose[3] [0.166 0.027 10]
  spose[4] [0.166 -0.027 -10]
  spose[5] [0.148 -0.078 -30]
```

```
    spose[6] [0.114 -0.119 -50]
    spose[7] [0.069 -0.136 -90]
    spose[8] [-0.157 -0.136 -90]
    spose[9] [-0.203 -0.119 -130]
    spose[10] [-0.237 -0.078 -150]
    spose[11] [-0.255 -0.027 -170]
    spose[12] [-0.255 0.027 170]
    spose[13] [-0.237 0.078 150]
    spose[14] [-0.203 0.119 130]
    spose[15] [-0.157 0.136 90]
    spose[16] [-0.02 0.136 90]

    # Top sonar ring, same as bottom front in the 2D stage simulation:
    spose[17] [0.024 0.119 50]
    spose[18] [0.058 0.078 30]
    spose[19] [0.077 0.027 10]
    spose[20] [0.077 -0.027 -10]
    spose[21] [0.058 -0.078 -30]
    spose[22] [0.024 -0.119 -50]
    spose[23] [-0.02 -0.136 -90]
    spose[24] [-0.191 -0.136 -90]
  )
  sicklms200( pose [0.021 0 0] )
)


# just another name for peoplebot-sh:
define peoplebot peoplebot-sh (
)



# Model for a pion1x differential-drive robot base with sonar.
define pion1x pioneer (
  pioneer_robot_subtype "pion1x"
  color "blue"

  # Speed profile:
```

```
max_speed [0.4 0 1.74]
accel [0 0 0]
decel [0 0 0]

# Body shape:
origin [-0.05 0.0 0.0]
size [0.5 0.4]
polygons 1
polygon[0].points 8
polygon[0].point[0] [-0.175 0.25]
polygon[0].point[1] [0.175 0.25]
polygon[0].point[2] [0.2 0.175]
polygon[0].point[3] [0.2 -0.175]
polygon[0].point[4] [0.175 -0.25]
polygon[0].point[5] [-0.175 -0.25]
polygon[0].point[6] [-0.2 -0.175]
polygon[0].point[7] [-0.2 0.175]
polygon[0].filled 1

#height 0.22

# client protocol conversion factors
pioneer_diffconv 0.00333333
pioneer_distconv 0.05066
pioneer_angleconv 0.0061359
pioneer_vel2div 4.0
pioneer_velconv 2.5332
pioneer_rangeconv 0.1734

# Sonar:
pioneerSonar (
  scount 7
  sview [0.1 3.5 30]
  ssize [0.01 0.04]
  spose[0] [0.1 0.13 90]
  spose[1] [0.12 0.095 30]
  spose[2] [0.13 0.05 15]
  spose[3] [0.13 0 0]
```

```
    spose[4] [0.13 -0.05 -15]
    spose[5] [0.12 -0.095 -30]
    spose[6] [0.1 -0.13 -90]
  )
)


# Model for a pioneer1 at differential-drive robot base with sonar.
define pionat pioneer (
  pioneer_robot_subtype "pionat"
  color "blue"

  # Speed profile:
  max_speed [0.4 0 1.74]
  accel [0 0 0]
  decel [0 0 0]

  # Body shape:
  size [0.5 0.4]
  polygons 1
  polygon[0].points 8
  polygon[0].point[0] [-0.175 0.25]
  polygon[0].point[1] [0.175 0.25]
  polygon[0].point[2] [0.2 0.175]
  polygon[0].point[3] [0.2 -0.175]
  polygon[0].point[4] [0.175 -0.25]
  polygon[0].point[5] [-0.175 -0.25]
  polygon[0].point[6] [-0.2 -0.175]
  polygon[0].point[7] [-0.2 0.175]
  polygon[0].filled 1

  #height 0.22

  # client protocol conversion factors
  pioneer_diffconv 0.00333333
  pioneer_distconv 0.07
  pioneer_angleconv 0.0061359
  pioneer_rangeconv 0.1734
```

```
  pioneer_vel2div 4.0

  # Sonar:
  pioneerSonar (
    scount 7
    sview [0.1 3.5 30]
    ssize [0.01 0.04]
    spose[0] [0.1 0.1 90]
    spose[1] [0.12 0.08 30]
    spose[2] [0.13 0.04 15]
    spose[3] [0.13 0 0]
    spose[4] [0.13 -0.04 -15]
    spose[5] [0.12 -0.08 -30]
    spose[6] [0.1 -0.1 -90]
  )
)


# Model for a powerbot differential-drive robot base with sonar.
define powerbot pioneer (
  pioneer_robot_subtype "powerbot"
  color "yellow"

  # Speed profile:
  max_speed [2 0 1.74]

  # Body shape:
  origin [-0.073565 0.0 0.0]
  size [0.911 0.68]
  polygons 1
  polygon[0].points 8
  polygon[0].point[0] [-0.24 0.4555]
  polygon[0].point[1] [0.24 0.4555]
  polygon[0].point[2] [0.34 0.24]
  polygon[0].point[3] [0.34 -0.24]
  polygon[0].point[4] [0.24 -0.4555]
  polygon[0].point[5] [-0.24 -0.4555]
  polygon[0].point[6] [-0.34 -0.24]
```

```
polygon[0].point[7] [-0.34 0.24]
polygon[0].filled 1

#height 0.48396

# client protocol conversion factors
pioneer_diffconv 0.00373
pioneer_distconv 0.5813


# Sonar:
pioneerSonar (
  scount 32
  spose[0] [0.152 0.278 90]
  spose[1] [0.2 0.267 65]
  spose[2] [0.241 0.238 45]
  spose[3] [0.274 0.2 35]
  spose[4] [0.3 0.153 25]
  spose[5] [0.32 0.096 15]
  spose[6] [0.332 0.033 5]
  spose[7] [0.333 0 0333 0]
  sview[7] [5 5 0] # does not really exist but needs to return a reading (5m)
  spose[8] [0.332 -0.033 -5]
  spose[9] [0.32 -0.096 -15]
  spose[10] [0.3 -0.153 -25]
  spose[11] [0.274 -0.2 -35]
  spose[12] [0.241 -0.238 -45]
  spose[13] [0.2 -0.267 -65]
  spose[14] [0.152 -0.278 -90]
  spose[15] [0     -0.278 -90]
  sview[15] [5 5 0] # does not really exist but needs to return a reading (5m)
  spose[16] [-0.298 -0.278 -90]
  spose[17] [-0.347 -0.267 -115]
  spose[18] [-0.388 -0.238 -135]
  spose[19] [-0.42 -0.2 -145]
  spose[20] [-0.447 -0.153 -155]
  spose[21] [-0.467 -0.096 -165]
  spose[22] [-0.478 -0.033 -175]
```

```
    spose[23] [-0.478  0     -180]
    sview[23] [5 5 0] # does not really exist but needs to return a reading (5m)
    spose[24] [-0.478 0.033 175]
    spose[25] [-0.467 0.096 165]
    spose[26] [-0.447 0.153 155]
    spose[27] [-0.42 0.2 145]
    spose[28] [-0.388 0.238 135]
    spose[29] [-0.347 0.267 115]
    spose[30] [-0.298 0.278 90]
    spose[31] [0  0.278 90]
    sview[31] [5 5 0] # does not really exist but needs to return a reading (5m)
  )
  sicklms200(
    pose [0.251 0 0]
    height_offset -0.40811
      # Vertical position from top plate when mounted in
      # typical "lower" position (just above bumpers)
  )
)


# Model for a powerbot differential-drive robot base with sonar.
define powerbot-sh pioneer (
  pioneer_robot_subtype "powerbot-sh"

  color "yellow"
  # Speed profile:
  max_speed [2 0 1.74]

  # Body shape:
  origin [-0.073565 0.0 0.0]
  size [0.911 0.68]
  polygons 1
  polygon[0].points 8
  polygon[0].point[0] [-0.24 0.4555]
  polygon[0].point[1] [0.24 0.4555]
  polygon[0].point[2] [0.34 0.24]
  polygon[0].point[3] [0.34 -0.24]
```

```
polygon[0].point[4] [0.24 -0.4555]
polygon[0].point[5] [-0.24 -0.4555]
polygon[0].point[6] [-0.34 -0.24]
polygon[0].point[7] [-0.34 0.24]
polygon[0].filled 1

# client protocol conversion factors
pioneer_diffconv 0.00373
pioneer_distconv 1.0

# Sonar:
pioneerSonar (
  scount 32
  spose[0] [0.152 0.278 90]
  spose[1] [0.2 0.267 65]
  spose[2] [0.241 0.238 45]
  spose[3] [0.274 0.2 35]
  spose[4] [0.3 0.153 25]
  spose[5] [0.32 0.096 15]
  spose[6] [0.332 0.033 5]
  spose[7] [0.333 0 0333 0]
  sview[7] [5 5 0] # does not really exist but needs to return a reading (5m)
  spose[8] [0.332 -0.033 -5]
  spose[9] [0.32 -0.096 -15]
  spose[10] [0.3 -0.153 -25]
  spose[11] [0.274 -0.2 -35]
  spose[12] [0.241 -0.238 -45]
  spose[13] [0.2 -0.267 -65]
  spose[14] [0.152 -0.278 -90]
  spose[15] [0     -0.278 -90]
  sview[15] [5 5 0] # does not really exist but needs to return a reading (5m)
  spose[16] [-0.298 -0.278 -90]
  spose[17] [-0.347 -0.267 -115]
  spose[18] [-0.388 -0.238 -135]
  spose[19] [-0.42 -0.2 -145]
  spose[20] [-0.447 -0.153 -155]
  spose[21] [-0.467 -0.096 -165]
  spose[22] [-0.478 -0.033 -175]
```

```
    spose[23] [-0.478  0     -180]
    sview[23] [5 5 0] # does not really exist but needs to return a reading (5m)
    spose[24] [-0.478 0.033 175]
    spose[25] [-0.467 0.096 165]
    spose[26] [-0.447 0.153 155]
    spose[27] [-0.42 0.2 145]
    spose[28] [-0.388 0.238 135]
    spose[29] [-0.347 0.267 115]
    spose[30] [-0.298 0.278 90]
    spose[31] [0  0.278 90]
    sview[31] [5 5 0] # does not really exist but needs to return a reading
  )
  sicklms200(
    pose [0.251 0 0]

    # Vertical position from top plate when mounted in
    # typical "lower" position (just above bumpers):
    height_offset -0.40811

    # Mounted upside down, according to powerbot-sh.p:
    reverse_scan 1
    laser_beam_height 0.115
  )
)

# Seekur with one SICK mounted on front
define seekur pioneer (
  pioneer_robot_subtype "seekur"
  color "cornflower blue"
  drive "omni"
  max_speed [3 3 1.74]
  accel [0.5 0.5 2.0]
  decel [0.5 0.5 2.0]
  size [1.4 1.3]
  polygons 1
  polygon[0].points 8
  polygon[0].point[0] [-0.3 0.4]
  polygon[0].point[1] [0.3 0.4]
```

```
  polygon[0].point[2] [0.4 0.3]
  polygon[0].point[3] [0.4 -0.3]
  polygon[0].point[4] [0.3 -0.4]
  polygon[0].point[5] [-0.3 -0.4]
  polygon[0].point[6] [-0.4 -0.3]
  polygon[0].point[7] [-0.4 0.3]
  polygon[0].filled 1

  #height 1.0 #approx.

  pioneer_angleconv 0.001534
  pioneer_distconv 1.0
  pioneer_velconv 1.0
  pioneer_diffconv 0.0056


  sicklms200(
    pose [0.62 0.00 0]
    height_offset -0.42 #approx.
  )
)

define seekurjr pioneer (
  pioneer_robot_subtype "seekurjr"
  color "cornflower blue"
  gui_boundary 1
  max_speed [1.2 0 1.396]
  accel [0.5 0 2.0]
  decel [0.5 0 2.0]
  size [1.2 0.8]
  polygons 1
  polygon[0].points 8
  #polygon[0].point[0] [0 0]
  #polygon[0].point[1] [0 0]
  #polygon[0].point[2] [0 0]
  #polygon[0].point[3] [0 0]
  #polygon[0].point[4] [0 0]
  #polygon[0].point[5] [0 0]
```

```
  #polygon[0].point[6] [0 0]
  #polygon[0].point[7] [0 0]
  polygon[0].point[0] [-0.3 0.4]
  polygon[0].point[1] [0.3 0.4]
  polygon[0].point[2] [0.4 0.3]
  polygon[0].point[3] [0.4 -0.3]
  polygon[0].point[4] [0.3 -0.4]
  polygon[0].point[5] [-0.3 -0.4]
  polygon[0].point[6] [-0.4 -0.3]
  polygon[0].point[7] [-0.4 0.3]
  polygon[0].filled 1
  pioneer_angleconv 0.001534
  pioneer_distconv 1
  pioneer_velconv 1
  pioneer_diffconv 0.0056

  sicklms100 (
    pose [0.6 0 0]
  )

)

# This is a quick hack to have several differently colored Pioneers
# in the same simulation
define red_p3dx p3dx (
   color "red"
)
define blue_p3dx p3dx (
   color "blue"
)
define violet_p3dx p3dx (
   color "violet"
)
define green_p3dx p3dx (
   color "green"
)
define orange_p3dx p3dx (
   color "orange"
```

```
)

define p3at_with_gps p3at-sh (
  gps (
    pose [ -0.15 0.09 0 ]
    color "white"
  )
)
# P3AT-SH but with only 8 front sonar instead of 16
define p3at-frontsonaronly pioneer (
  pioneer_robot_subtype "p3at-sh"
  pioneer_distconv 1.0
  pioneer_diffconv 0.0034
  pioneer_velconv 1.0
  pioneer_angleconv 0.001534
  max_speed [0.6 0 0.75]
  size [0.626 0.505]
  polygons 1
  polygon[0].points 8
  polygon[0].point[0] [-0.18 0.313]
  polygon[0].point[1] [0.18 0.313]
  polygon[0].point[2] [0.2525 0.18]
  polygon[0].point[3] [0.2525 -0.18]
  polygon[0].point[4] [0.18 -0.313]
  polygon[0].point[5] [-0.18 -0.313]
  polygon[0].point[6] [-0.2525 -0.18]
  polygon[0].point[7] [-0.2525 0.18]
  polygon[0].filled 1
  pioneer_hasreararray 0
  pioneerSonar (
     scount 8
     spose[0] [0.147 0.136 90]
     spose[1] [0.193 0.119 50]
     spose[2] [0.227 0.079 30]
     spose[3] [0.245 0.027 10]
     spose[4] [0.245 -0.027 -10]
     spose[5] [0.227 -0.079 -30]
     spose[6] [0.193 -0.119 -50]
```

```
      spose[7] [0.147 -0.136 -90]
 )
)

define box model (
  size [0.5 0.5]
  gui_boundary 1
  color "yellow"
  polygons 1
  polygon[0].points 4
  polygon[0].point[0] [-0.5 -0.5]
  polygon[0].point[1] [-0.5 0.5]
  polygon[0].point[2] [0.5 0.5]
  polygon[0].point[3] [0.5 -0.5]
  polygon[0].filled 1
  obstacle_return 1
  laser_return 1
  ranger_return 1
)

define box-nolaser box (
  size [0.5 0.5]
  obstacle_return 1
  laser_return 0
  ranger_return 1
)

define box-nolaser-nosonar box (
  size [0.5 0.5]
  obstacle_return 1
  laser_return 0
  ranger_return 0
)

define box-reflector box (
  size [0.5 0.5]
  obstacle_return 1
  laser_return 2
```

```
  ranger_return 1
)



# Model for an mtx lynx/lx differential-drive robot base with lms300, sonar,
# bumps.
define mtx pioneer (
  color "grey"
  pioneer_robot_type "MTX"
  pioneer_robot_subtype "mtx"

  # Speed profile:
  max_speed [2.0 0 3.49]
  accel [0.5 0 1.74]
  decel [0.5 0 1.74]

  # Body shape:
  size [0.64 0.48]
  polygons 1
  polygon[0].points 12
  polygon[0].point[0]  [ 3223 -0668]
  polygon[0].point[1]  [ 2913 -1775]
  polygon[0].point[2]  [ 2075 -2408]
  polygon[0].point[3]  [-2075 -2408]
  polygon[0].point[4]  [-2913 -1775]
  polygon[0].point[5]  [-3223 -0668]
  polygon[0].point[6]  [-3223  0668]
  polygon[0].point[7]  [-2913  1775]
  polygon[0].point[8]  [-2075  2408]
  polygon[0].point[9]  [ 2075  2408]
  polygon[0].point[10] [ 2913  1775]
  polygon[0].point[11] [ 3223  0668]
  polygon[0].filled 1

  # client protocol conversion factors
  pioneer_diffconv 1.0
```

```
  pioneer_distconv 1.0
  pioneer_angleconv 0.001534

  # Battery type (determines behavior of battery test commands and data reported
  # in SIP)
  pioneer_batterytype 2

  # Sonar:
  ranger (
    laser_return 0
    blobfinder_return 0
    fiducial_return 0
    noise 0.0005
    projection_type "single"
    pioneer_hasfrontarray 1
    pioneer_hasreararray 1
    sview [0.1 4.335 50]    # XXX FOV is a complete guess
    ssize [0.01 0.021]
    scount 8      # XXX There might actually only be 6 (2 in front)
    spose[0] [0.0324 0.090 15.5]
    spose[1] [0.0331 0.0607 10.4]
    spose[2] [0.0331 -0.0607 -10.4]
    spose[3] [0.0324 -0.090 -15.5]
    spose[4] [-0.0316 -0.090 -164.24]
    spose[5] [-0.0325 -0.030 -174.73]
    spose[6] [-0.0325 0.030 174.73]
    spose[7] [-0.0316 0.090 164.24]
  )

# ARIA can't use simulated s300 yet  sicks300(
  sicklms100 (
    # Mounted upside down, inside front of the body:
    pose [0.267 0 0]
    reverse_scan 1
  )

)
```

```
define lynx mtx (
  pioneer_robot_subtype "lynx"
)

define lx mtx (
  pioneer_robot_subtype "lx"
)
```