

Odysseus Documentation

A. Giavaras

Contents

1	Modeling	2
1.1	Kinematics Model	2
2	State Estimation	3
2.1	Extended Kalman Filter	3
2.1.1	Ultrasound Sensor Model	3
2.2	EKF Simulation Verification	4
2.2.1	Test 1	4
2.2.2	Test 2	4
3	Software Architecture & Design	5

1 Modeling

This section discusses the modeling approach that Odisseus follows

1.1 Kinemtics Model

Odisseus is using the following kinematic model

$$\frac{dx}{dt} = v \cos(\theta) \quad (1)$$

$$\frac{dy}{dt} = v \sin(\theta) \quad (2)$$

$$\frac{d\theta}{dt} = \omega \quad (3)$$

where x, y are the coordinates of the reference point, θ is the yaw angle, v is the input velocity and ω is the input angular velocity of the robotic platform.

The state vector \mathbf{x} has three components; the x, y components of the reference point and the orientation or yaw angle θ . Mathematically, this is written as

$$\mathbf{x} = (x, y, \theta) \quad (4)$$

As mentioned previously, the velocity v is one of the inputs that is given to the system. Namely, it is calculated according to

$$v = \frac{v_L * R + v_R * R}{2} \quad (5)$$

where R is the wheels radius and v_R, v_L are the right and left wheels velocities respectively. Similarly the second input to the system is the angular velocity of the robot which is given by

$$\omega = \frac{\omega_L * R + \omega_R * R}{L} \quad (6)$$

where L is the axle length connecting the two motorized wheels. ω_L, ω_R are the angular velocities of the left and right wheels respectively.

This kinematic model is used as a motion model in the Extended Kalman Filter discussed in section 2.1. Concretely, the following discretized form is utilized

$$x_k = x_{k-1} + (\Delta t v_k + \mathbf{w}_{1,k}) \cos(\theta_{k-1} + \Delta t \omega_k + \mathbf{w}_{2,k}) \quad (7)$$

$$y_k = y_{k-1} + (\Delta t v_k + \mathbf{w}_{1,k}) \sin(\theta_{k-1} + \Delta t \omega_k + \mathbf{w}_{2,k}) \quad (8)$$

$$\theta_k = \theta_{k-1} + \Delta t \omega_k + \mathbf{w}_{2,k} \quad (9)$$

where Δt is the sampling rate and \mathbf{w} is an error vector such that

$$E[\mathbf{w}] = \mathbf{0} \quad (10)$$

2 State Estimation

This section discusses the state estimation algorithms implemented in Odisseus.

2.1 Extended Kalman Filter

The Extended Kalman Filter is a popular state estimation technique for non-linear systems. It is an extension of the very popular Kalman Filter (see https://en.wikipedia.org/wiki/Kalman_filter). Just like the original Kalman Filter algorithm, the EKF also has two steps namely predict and update. The main difference of EKF over Kalman Filter is that it introduces a linearization of the non-linear system. Overall the algorithm is as follows

$$\mathbf{h} = \begin{pmatrix} h_{sonar} \\ h_{camera} \\ h_{ir} \end{pmatrix} \quad (11)$$

Odisseus is equipped with the following three types of sensors

- Ultrasound sensor
- Camera sensor
- infrared sensor

2.1.1 Ultrasound Sensor Model

As mentioned previously \mathbf{h} represents a vector valued function and h_{sonar} is the modeled measurement from the sonar sensor. Odisseus is using the following model

$$h_{sonar}(\mathbf{x}, \mathbf{v}_{sonar}) = \sqrt{(x - x_o)^2 + (y - y_o)^2} + \mathbf{v}_{sonar} \quad (12)$$

where \mathbf{v}_{sonar} is the error vector associated with the sonar. x_o, y_o are the coordinates of the obstacle detected by the sensor.

2.2 EKF Simulation Verification

This section presents some simulation results that verify the EKF implementation on Odisseus

2.2.1 Test 1

In this test the following input data was used

$$R = \begin{pmatrix} 1.0 & 0.0 \\ 0.0 & 1.0 \end{pmatrix} \quad (13)$$

$$Q = \begin{pmatrix} 0.001 & 0.0 \\ 0.0 & 0.001 \end{pmatrix} \quad (14)$$

The motion model \mathbf{f} is according to equation 9 where the error vector \mathbf{w} was set to zero.

The observation model function \mathbf{h} was simply the identity function meaning returning the passed state vector

$$\mathbf{h}(\mathbf{x}, \mathbf{v}) = \mathbf{x} \quad (15)$$

The error vector \mathbf{v} was set to

$$\mathbf{v} = (0.0, 0.0) \quad (16)$$

Finally, the following data was used

1. $\Delta t = 0.5$
2. $R = 2.5cm$
3. $v_L = v_R = 50RPM$
4. $L = 15cm$

2.2.2 Test 2

The second simulation test uses equation 12 to model the sonar measurement

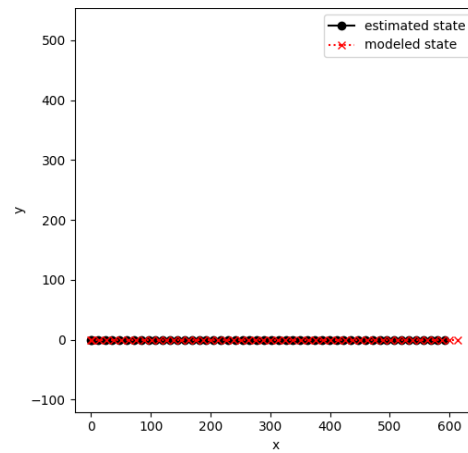


Fig. 1: Straight motion test 1.

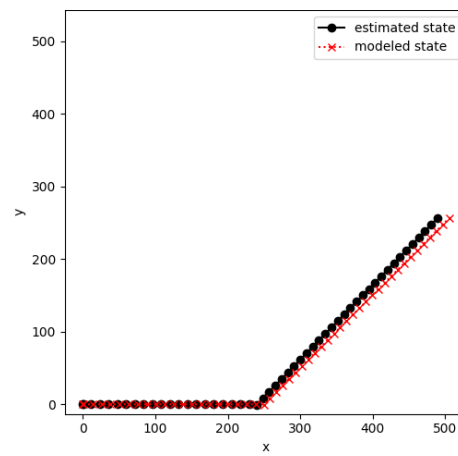


Fig. 2: Change direction test 1.

3 Software Architecture & Design

Odisseus is a multiprocess application. All its sensors as well as its motors run on a separate process. These processes are

- MasterProcess
- WebAppProcess
- CameraProcess
- IRProcess

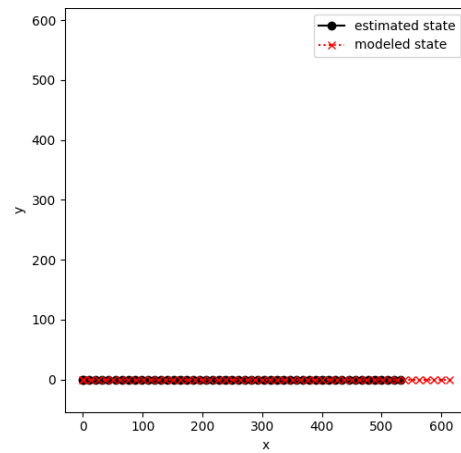


Fig. 3: Straight motion test 2.

- UltrasoundSensorProcess
- PropulsionProcess
- DecisionMakerProcess

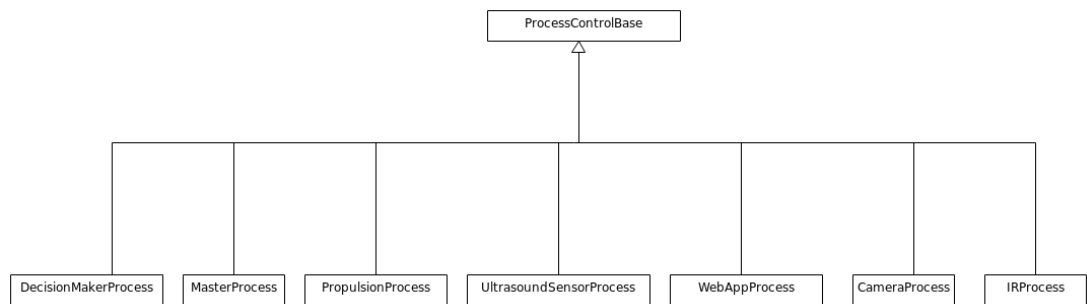


Fig. 4: Process inheritance diagram.

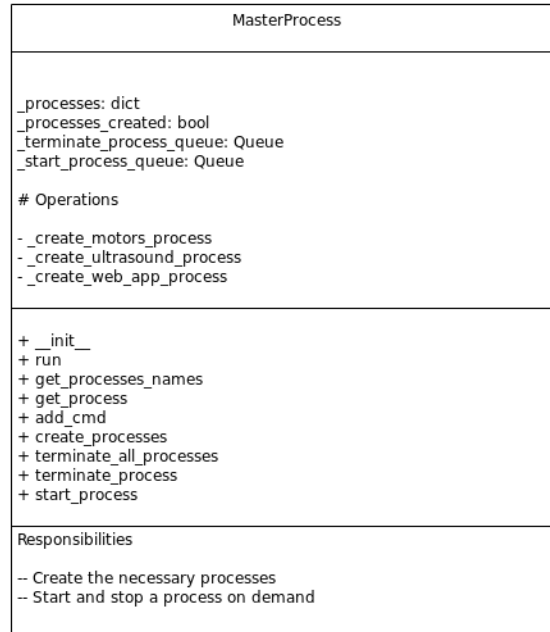


Fig. 5: MasterProcess.

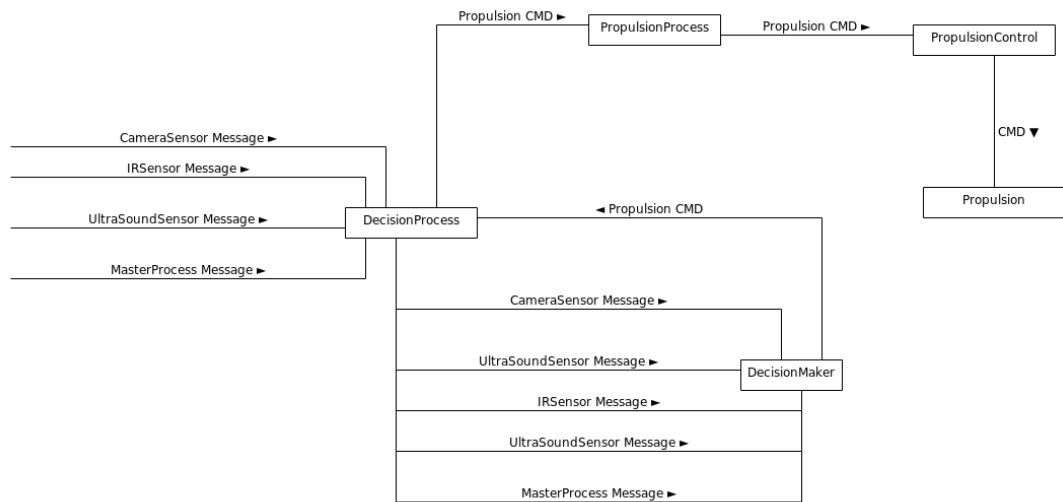


Fig. 6: Process messaging.