Enclosure methods for safe localisation

Quentin Brateau

ENSTA Bretagne · Agence Innovation Défense

Table of contents

State Equation

System defined by state equation

$$\begin{cases} \dot{x} = f(x,u) & (\textit{evolution}) \quad (1) \\ y = g(x,u) & (\textit{observation}) \quad (2) \end{cases}$$

State Equation

System defined by state equation

$$\begin{cases} \dot{x} = f(x,u) & (\textit{evolution}) \quad (1) \\ y = g(x,u) & (\textit{observation}) \quad (2) \end{cases}$$

Application

- Dynamic knowledge of the system
- Control the system
- Simulate the system

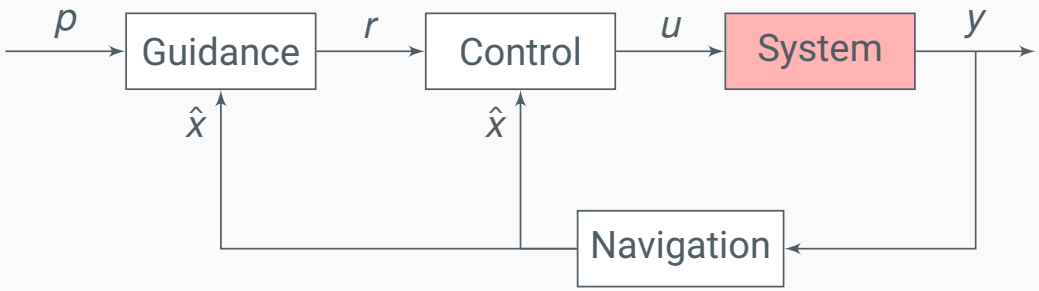


Figure 1: Guidance-Navigation-Control block diagram

System

System for which state x want to be controlled.

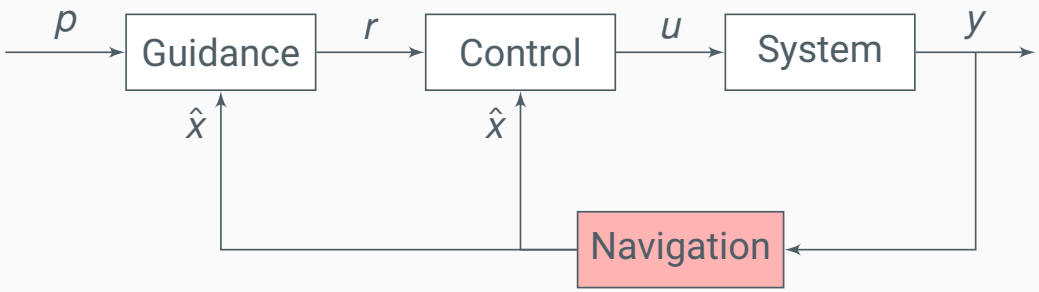


Figure 1: Guidance-Navigation-Control block diagram

Navigation

Estimate the state $\hat{x}$ of the system using its output y.

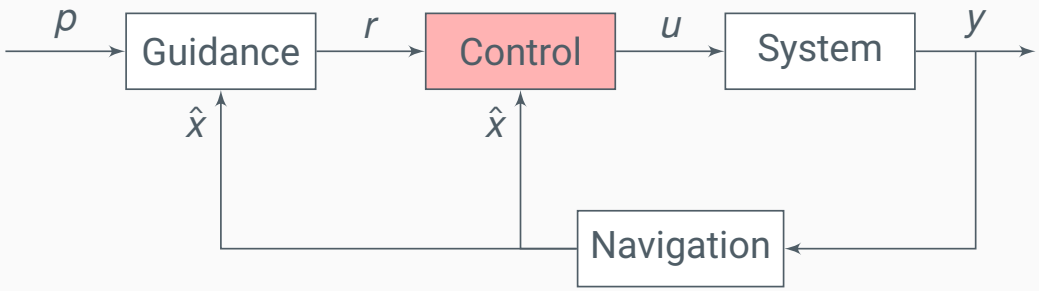


Figure 1: Guidance-Navigation-Control block diagram

Control

Control the system toward the reference r from the guidance using estimated state $\hat{x}$.

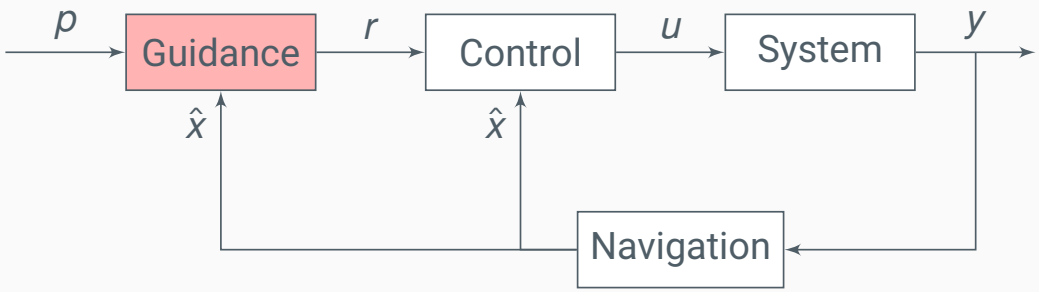


Figure 1: Guidance-Navigation-Control block diagram

Guidance

Guide the system toward a desired state using the estimated output $\hat{x}$ and some input parameters p.

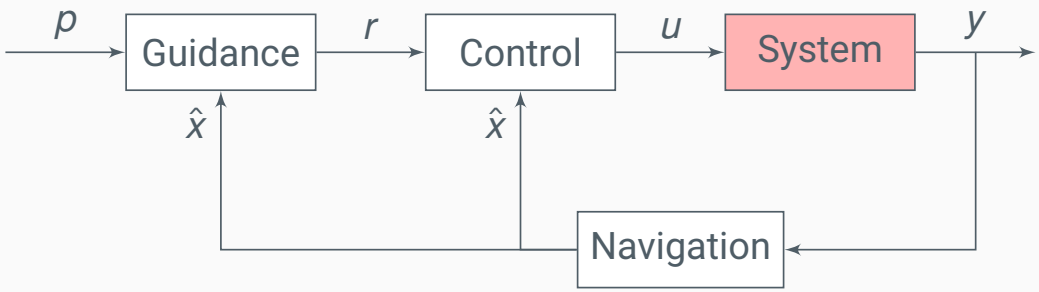


Figure 2: Guidance-Navigation-Control block diagram

Nonlinear systems

- Approximations in Navigation
- Approximations in Control

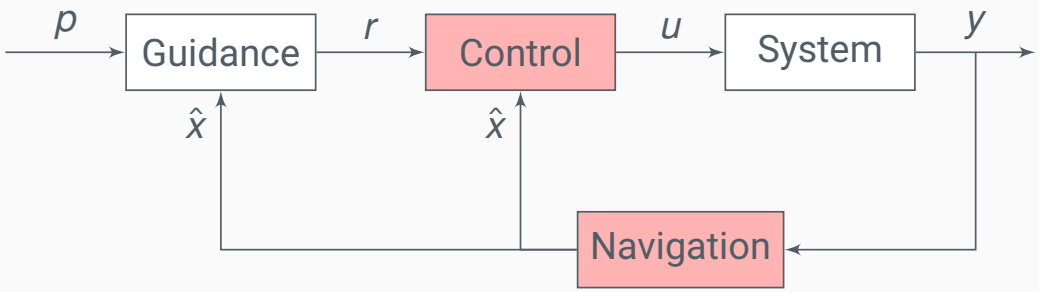


Figure 2: Guidance-Navigation-Control block diagram

Nonlinear systems

- Approximations in Navigation
- Approximations in Control

Set methods

- Deal with nonlinear equations
- Guaranteed results

Dead reckoning

- Proprioceptive informations (odometry, inertial, ...)
- Robot achieve dead reckoning

Dead reckoning

- Proprioceptive informations (odometry, inertial, ...)
- Robot achieve dead reckoning

Scope

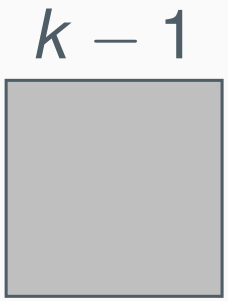
- Initial set enclosing robot's state
- Divergent set due to uncertainties

Dead reckoning

- Proprioceptive informations (odometry, inertial, ...)
- Robot achieve dead reckoning

Scope

- Initial set enclosing robot's state
- Divergent set due to uncertainties

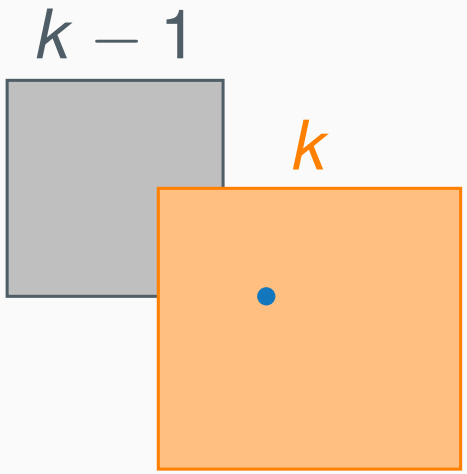


Dead reckoning

- Proprioceptive informations (odometry, inertial, ...)
- Robot achieve dead reckoning

Scope

- Initial set enclosing robot's state
- Divergent set due to uncertainties



Field

- Field (magnetic, acoustic, bathymetric, ...)
- Give information about robot possible state
- Can contract the enclosing robot state

Field

- Field (magnetic, acoustic, bathymetric, ...)
- Give information about robot possible state
- Can contract the enclosing robot state

Scope

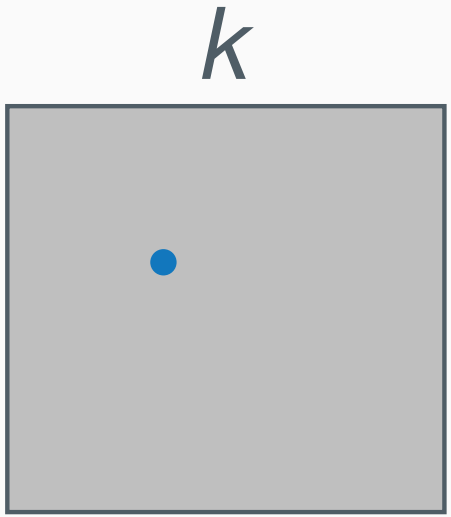
- Mapping known a priori
- Map generated by sensor or simulator

Field

- Field (magnetic, acoustic, bathymetric, ...)
- Give information about robot possible state
- Can contract the enclosing robot state

Scope

- Mapping known a priori
- Map generated by sensor or simulator

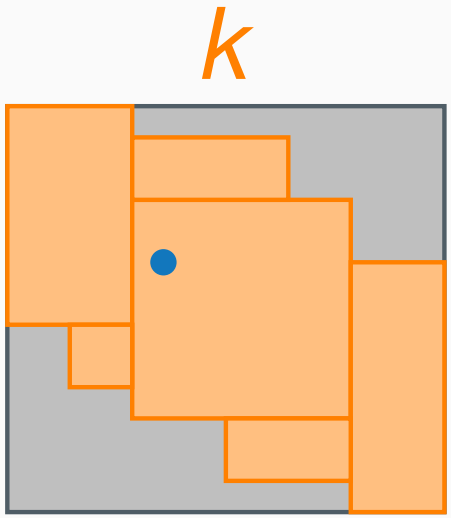


Field

- Field (magnetic, acoustic, bathymetric, ...)
- Give information about robot possible state
- Can contract the enclosing robot state

Scope

- Mapping known a priori
- Map generated by sensor or simulator



Prediction

- Using estimated robot state
- Generate prediction at next time step using enclosing evolution and control function

Prediction

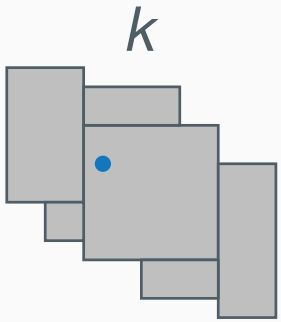
- Using estimated robot state
- Generate prediction at next time step using enclosing evolution and control function

Utility

- Check the viability of the control
- Danger zones
- Enclose the next estimated state

Prediction

- Using estimated robot state
- Generate prediction at next time step using enclosing evolution and control function



Utility

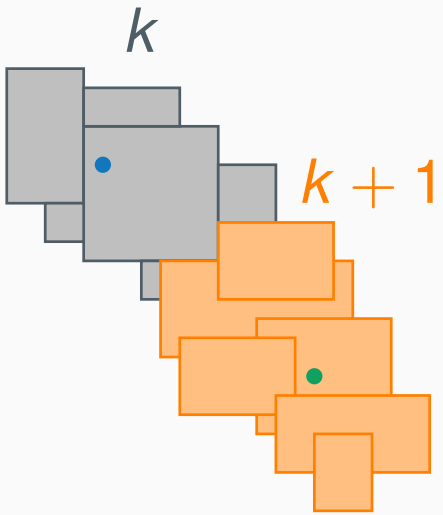
- Check the viability of the control
- Danger zones
- Enclose the next estimated state

Prediction

- Using estimated robot state
- Generate prediction at next time step using enclosing evolution and control function

Utility

- Check the viability of the control
- Danger zones
- Enclose the next estimated state

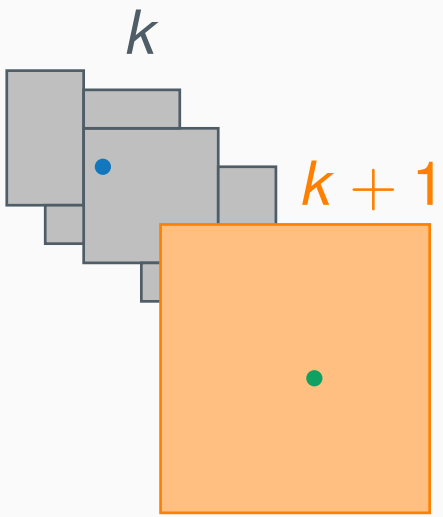


Prediction

- Using estimated robot state
- Generate prediction at next time step using enclosing evolution and control function

Utility

- Check the viability of the control
- Danger zones
- Enclose the next estimated state



Control

- Generate control using enclosing control function on estimated state
- Take the barycenter of control as inputs

Control

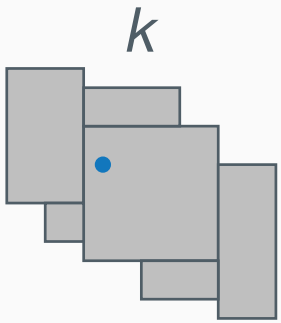
- Generate control using enclosing control function on estimated state
- Take the barycenter of control as inputs

Results

- Control roughly the robot
- Gives correct results

Control

- Generate control using enclosing control function on estimated state
- Take the barycenter of control as inputs

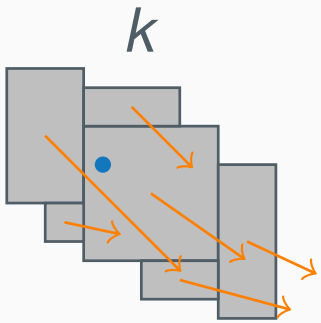


Results

- Control roughly the robot
- Gives correct results

Control

- Generate control using enclosing control function on estimated state
- Take the barycenter of control as inputs

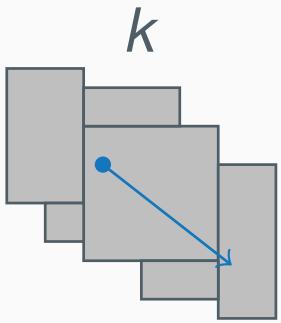


Results

- Control roughly the robot
- Gives correct results

Control

- Generate control using enclosing control function on estimated state
- Take the barycenter of control as inputs



Results

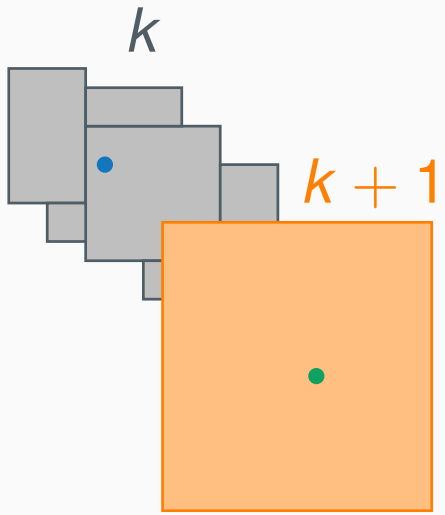
- Control roughly the robot
- Gives correct results

Control

- Generate control using enclosing control function on estimated state
- Take the barycenter of control as inputs

Results

- Control roughly the robot
- Gives correct results



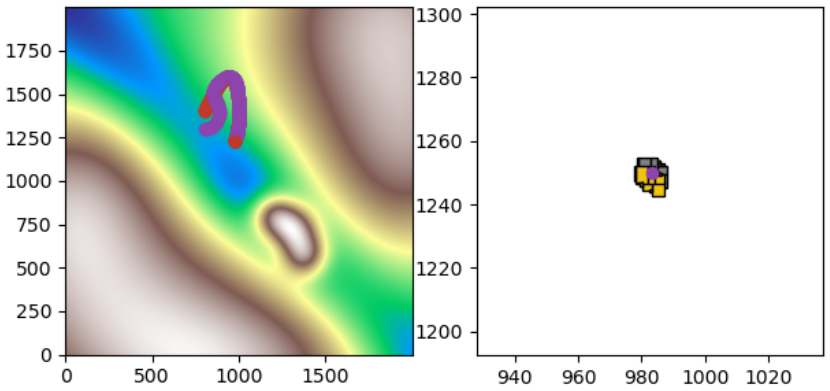
Scope of the example

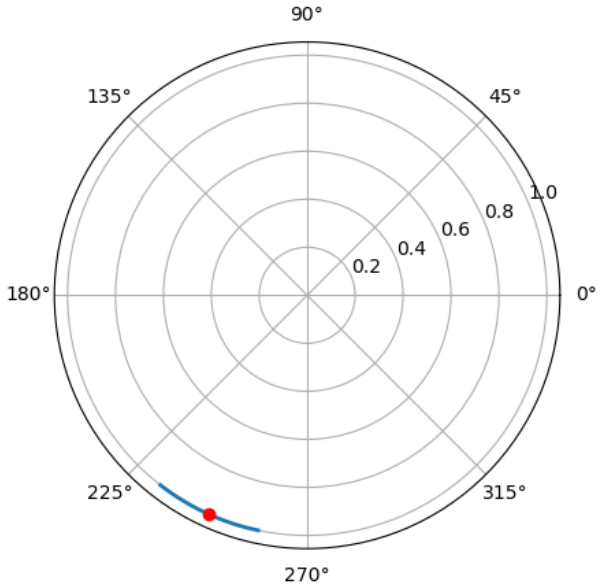
- Boat simulated with Dubins state equation
- Using simulated bathymetric map as measured field
- Control the boat to follow a Lissajous curve

Assumptions

- State estimation and control only using interval analysis
- Inertial drift set to 0.1 m per seconds
- Bathymetric measurements uncertainty set to 0.1 m

Example Simulation





Conclusion

- Works better in highly nonlinear fields
- Possibility to refine next state estimation by enclosing it in a sub-paving
- Still a simulation will be tried in guaranteed robot control experiments.