Navigation in the RCLL using Fawkes

Tobias Neumann
Navigation

**Global position**
Common frame during a whole game for all robots

**Driving**
Without colliding with other robots or the field

"His path-planning may be sub-optimal, but it's got flair."

http://wiki.ros.org/navigation
Wheel encoder and IMU
Calculate the driven path by the robot

Drift
Unbound over time

---

\(^1\)http://www.ros.org/reps/rep-0105.html
Wheel encoder and IMU
Calculate the driven path by the robot

Drift
Unbound over time

\footnote{http://www.ros.org/reps/rep-0105.html}
1. Navigation

2. Localisation

3. NavGraph (Global Planner)

4. Automatic Generation of the NavGraph

5. Colli (Local Planner)

6. Robotino (Motor)

7. How-to use the Pathplanner
1 Navigation

2 Localisation

3 NavGraph (Global Planner)

4 Automatic Generation of the NavGraph

5 Colli (Local Planner)

6 Robotino (Motor)

7 How-to use the Pathplanner
Localisation

AMCL

- Adaptive Monte Carlo Localisation
- Particle filter
- Validating laser against map

http://playerstage.sourceforge.net/doc/Player-2.0.0/player/group__driver__amcl.html
Localisation

AMCL

- Adaptive Monte Carlo Localisation
- Particle filter
- Validating laser against map

http://playerstage.sourceforge.net/doc/Player-2.0.0/player/group__driver__amcl.html
Localisation

AMCL

- Adaptive Monte Carlo Localisation
- Particle filter
- Validating laser against map

http://playerstage.sourceforge.net/doc/Player-2.0.0/player/group__driver__amcl.html
AMCL

- Adaptive Monte Carlo Localisation
- Particle filter
- Validating laser against map

http://playerstage.sourceforge.net/doc/Player-2.0.0/player/group__driver__amcl.html
Localisation

AMCL

- Adaptive Monte Carlo Localisation
- Particle filter
- Validating laser against map

http://playerstage.sourceforge.net/doc/Player-2.0.0/player/group__driver__amcl.html
Localisation

**AMCL**
- Adaptive Monte Carlo Localisation
- Particle filter
- Validating laser against map

**Map generation**
Manual drawn using GIMP
(pixel ≈ 5 cm)

---

Map RoboCup 2015, Hefei, China

![Diagram showing flow of localisation]
Path Planning

Map based planner → Sensor based planner → Motor

Global planner → Local planner → Odometry
Path Planning

- Map based planner
- Sensor based planner
- Motor
- Global planner
- Local planner
- Odometry
- NavGraph
- Colli
- Robotino
1 Navigation

2 Localisation

3 NavGraph (Global Planner)

4 Automatic Generation of the NavGraph

5 Colli (Local Planner)

6 Robotino (Motor)

7 How-to use the Pathplanner
Topological graph

Team Description RoboCup@Home; S.Schiffer, T.Niemueller, M.Doostdar, G.Lakemeyer; 2009
Topological graph

Team Description RoboCup@Home; S.Schiffer, T.Niemueller, M.Doostdar, G.Lakemeyer; 2009
```yaml
# Config file for NavGraph

- nodes
- edges
- properties

1. fawkes-robotino/fawkes/src/plugins/navgraph
2. fawkes-robotino/cfg/navgraph-basis.yaml
```
- A* on edges
- Current node used for local planner
- Go to next node if threshold is reached
- A* on edges
- Current node used for local planner
- Go to next node if threshold is reached
- A* on edges
- Current node used for local planner
- Go to next node if threshold is reached
A* on edges

Current node used for local planner

Go to next node if threshold is reached
A* on edges

Current node used for local planner

Go to next node if threshold is reached
A* on edges

Current node used for local planner

Go to next node if threshold is reached
A* on edges
- Current node used for local planner
- Go to next node if threshold is reached
1 Navigation

2 Localisation

3 NavGraph (Global Planner)

4 Automatic Generation of the NavGraph

5 Colli (Local Planner)

6 Robotino (Motor)

7 How-to use the Pathplanner
Basis graph

Agent -> navgraph-generator-mps -> navgraph-generator
Nodes are generated using a Voronoi diagram
Nodes are generated using a Voronoi diagram
Nodes are generated using a Voronoi diagram
Nodes are generated using a Voronoi diagram
Nodes are generated using a Voronoi diagram
Nodes are generated using a Voronoi diagram
- Nodes are generated using a Voronoi diagram
Nodes are generated using a Voronoi diagram
1 Navigation

2 Localisation

3 NavGraph (Global Planner)

4 Automatic Generation of the NavGraph

5 Colli (Local Planner)

6 Robotino (Motor)

7 How-to use the Pathplanner
Colli

1. Obstacle map
2. Planning
3. Drive Mode
4. Drive Realisation

1fawkes-robotino/fawkes/src/plugins/colli
Robust Collision Avoidance in Unknown Domestic Environments; S. Jacobs, A. Ferrein, S. Schiffer, D. Beck, and G. Lakemeyer; 2009

- Creates an obstacle map from laser data
Robust Collision Avoidance in Unknown Domestic Environments; S. Jacobs, A. Ferrein, S. Schiffer, D. Beck, and G. Lakemeyer; 2009

- Planing using A*
Robust Collision Avoidance in Unknown Domestic Environments; S. Jacobs, A. Ferrein, S. Schiffer, D. Beck, and G. Lakemeyer; 2009

- Ray tracing points on path for next collision free point
Using drive modes to calculate motor velocities
Simple linear change of the velocity
1 Navigation

2 Localisation

3 NavGraph (Global Planner)

4 Automatic Generation of the NavGraph

5 Colli (Local Planner)

6 Robotino (Motor)

7 How-to use the Pathplanner
Robotino Plugin

Plugin to communicate with the Robotino

- TransRotMessage(vx, vy, omega)
- Odometry

Furthermore

- RobotinoSensorInterface
- IMUInterface (Robotino 3 IMU)
- GripperInterface (Robotino gripper)

\(^1\)fawkes-robotino/fawkes/src/plugins/robotino
1 Navigation

2 Localisation

3 NavGraph (Global Planner)

4 Automatic Generation of the NavGraph

5 Colli (Local Planner)

6 Robotino (Motor)

7 How-to use the Pathplanner
Interfaces

Interface of the NavGraph

NavigatorInterface::<Pathplan>
PlaceGotoMessage(place=<node on NavGraph>)
Frame map

Interface of the Colli

NavigatorInterface::<Navigator>
CartesianGoto(x=<x>, y=<y>, orientation=<theta>)
Frame base_link

Interface of the Robotino

MotorInterface::<Robotino>
TransRotMessage(vx=<vx>, vy=<vy>, omega=<omega>)
Frame base_link

^1 fawkes-robotino/fawkes/src/libs/interfaces/NavigatorInterface.xml
^2 fawkes-robotino/fawkes/src/libs/interfaces/MotorInterface.xml
Path planner

- relgoto\{x=\langle x\rangle, y=\langle y\rangle, \text{ori}=\langle\theta\rangle\}
- goto\{x=\langle x\rangle, y=\langle y\rangle, \text{ori}=\langle\theta\rangle\}
- ppgoto\text{place}=\langle\text{node on NavGraph}\rangle\}

Direct path

- motor\_move\{x=\langle x\rangle, y=\langle y\rangle, \text{ori}=\langle\theta\rangle\}

---

1. https://trac.fawkesrobotics.org/wiki/Carologistics/skills/relgoto
2. https://trac.fawkesrobotics.org/wiki/Carologistics/skills/ppgoto
Summary

Localisation
Using AMCL with a hand drawn map

NavGraph
Based on topological graph
Will be generated automatically

Colli
Local planner
Hands on

Change map

- Using GIMP (pixel $\approx$ 5 cm)
- Adapt to setup in hall

Change **NavGraph**

- Add the new MPS into the NavGraph

Drive on **NavGraph**

- Using the skills to drive