Artificial intelligence in robotics 2019
—
Autonomous Navigation

Tom Krajník

FEL ČVUT

Nov 2019

Navigation

- The art of getting from one place to another, safely and efficiently.
- The process of monitoring and controlling the movement of a craft or vehicle from one place to another.
- The activity of accurately ascertaining one's position and planning and following a route.
- As stated in [1], to navigate, one must answer the following three questions:

"Where am I?", "Where am I going?", "How do I get there?"

Navigation

- The art of getting from one place to another, safely and efficiently.
- The process of monitoring and controlling the movement of a craft or vehicle from one place to another.
- The activity of accurately ascertaining one's position and planning and following a route.
- As stated in [1], to navigate, one must answer the following three questions:

Localisation, "Where am I going?", "How do I get there?"

Navigation

- The art of getting from one place to another, safely and efficiently.
- The process of monitoring and controlling the movement of a craft or vehicle from one place to another.
- The activity of accurately ascertaining one's position and planning and following a route.
- As stated in [1], to navigate, one must answer the following three questions:

Localisation, Mapping, "How do I get there?"

Navigation

- The art of getting from one place to another, safely and efficiently.
- The process of monitoring and controlling the movement of a craft or vehicle from one place to another.
- The activity of accurately ascertaining one's position and planning and following a route.
- As stated in [1], to navigate, one must answer the following three questions:

Localisation, Mapping, Motion planning

Lecture overview

Autonomous navigation in mobile robotics can be divided by the way it uses knowledge of the environment [2]:

1. Map-less navigation
   - unknown environments with known structure
   - road following, obstacle avoidance
   - observations translate to motion commands
2. Map-based navigation
   - known (un)structured environments
   - observations and map translate to motion commands
   - a typical intermediate step is localisation
3. Map-building-based navigation
   - observations and map translate to both commands and map update
   - (un)known, (un)structured environments
   - a typical intermediate step is localisation

Map-less navigation

Apriori known environment structure
Sensor data → motion command
Design considerations often based on [3]

Pseudo-random
- behaviour-based

Optical flow
- Lucas-Kanade

Line detection
- RANSAC, Hough

Road segmentation
- region grow, CNN

Map-less navigation

Apriori known environment structure
Sensor data → motion command
Design considerations often based on [3]

Pseudo-random
- **behaviour-based**

Optical flow
- Lucas-Kanade

Line detection
- RANSAC, Hough

Road segmentation
- region grow, CNN

Map-less navigation

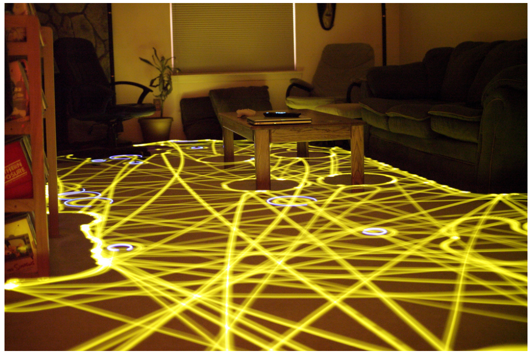
Apriori known environment structure
Sensor data → motion command
Design considerations often based on [3]

Pseudo-random
- **behaviour-based**

Optical flow
- Lucas-Kanade

Line detection
- RANSAC, Hough

Road segmentation
- region grow, CNN



Map-less navigation

Apriori known environment structure
Sensor data → motion command
Design considerations often based on [3]

Pseudo-random
- behaviour-based

Optical flow
- **Lucas-Kanade**

Line detection
- RANSAC, Hough

Road segmentation
- region grow, CNN

Video location: videos/flow

Map-less navigation

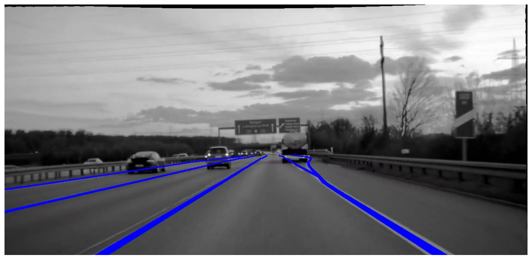
Apriori known environment structure
Sensor data → motion command
Design considerations often based on [3]

Pseudo-random
- behaviour-based

Optical flow
- Lucas-Kanade

Line detection
- **RANSAC, Hough**

Road segmentation
- region grow, CNN



Video location: videos/line-detect

Map-less navigation

Apriori known environment structure
Sensor data → motion command
Design considerations often based on [3]

Pseudo-random
- behaviour-based

Optical flow
- Lucas-Kanade

Line detection
- **RANSAC, Hough**

Road segmentation
- region grow, CNN

Map-less navigation

Apriori known environment structure
Sensor data → motion command
Design considerations often based on [3]

Pseudo-random
- behaviour-based

Optical flow
- Lucas-Kanade

Line detection
- RANSAC, Hough

Road segmentation
- **region grow, CNN**

Vid. loc.: videos/road-segmentation

Map-based navigation

Apriori known environment (pre-build map)
General: (observations, map) → motion command
Typical: (observations, map) → position → motion command

Topological map
- **map-less + direction**

Landmark map
- image features

Geometric map
- CAD, polygons

Occupancy grids
- 2d, 3d, OctoMap

Memory-based
- qualitative nav.

Video location: videos/lama

Map-based navigation

Apriori known environment (pre-build map)
General: (observations, map) → motion command
Typical: (observations, map) → position → motion command

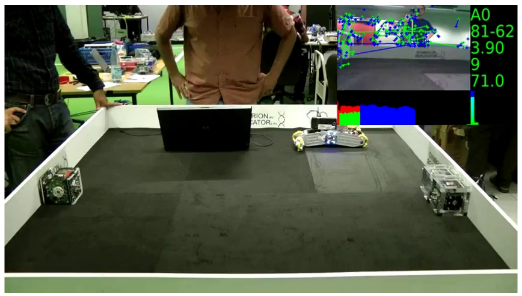
Topological map
- map-less + direction

Landmark map
- **image features**

Geometric map
- CAD, polygons

Occupancy grids
- 2d, 3d, OctoMap

Memory-based
- qualitative nav.



Video location: videos/repli

Map-based navigation

Apriori known environment (pre-build map)
General: (observations, map) → motion command
Typical: (observations, map) → position → motion command

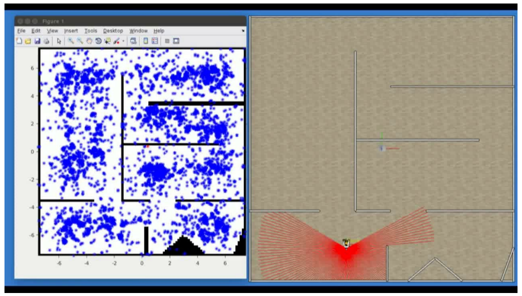
Topological map
- map-less + direction

Landmark map
- image features

Geometric map
- **CAD, polygons**

Occupancy grids
- 2d, 3d, OctoMap

Memory-based
- qualitative nav.



Video location: videos/mcl

Map-based navigation

Apriori known environment (pre-build map)
General: (observations, map) → motion command
Typical: (observations, map) → position → motion command

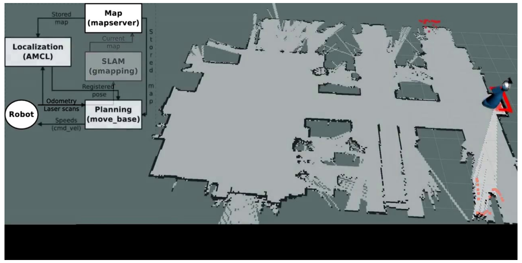
Topological map
- map-less + direction

Landmark map
- image features

Geometric map
- CAD, polygons

Occupancy grids
- **2d, 3d, OctoMap**

Memory-based
- qualitative nav.



Video location: videos/2dgrid

Map-based navigation

Apriori known environment (pre-build map)
General: (observations, map) → motion command
Typical: (observations, map) → position → motion command

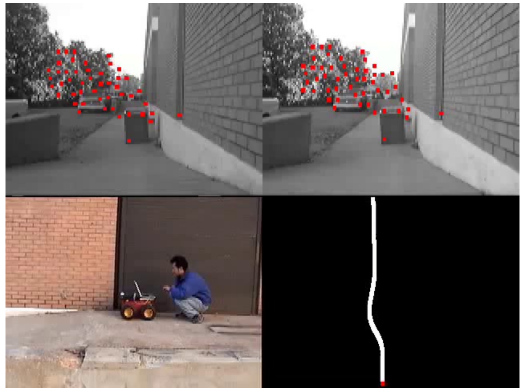
Topological map
- map-less + direction

Landmark map
- image features

Geometric map
- CAD, polygons

Occupancy grids
- 2d, 3d, OctoMap

Memory-based
- **qualitative nav.**



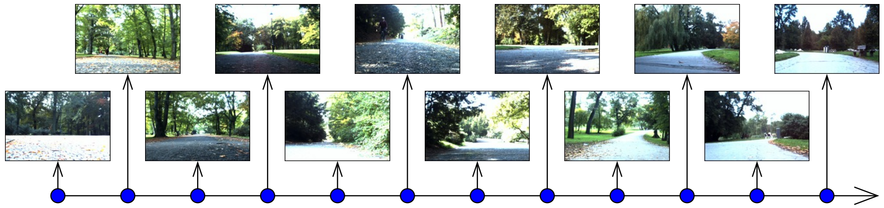
Video location: videos/qualitative

Example teach-and-repeat map-based navigation

- raw distance between frames,
- local snapshots of the environment,
- robot traverses approximately the same distance,
- steers according to what is sees **in front** of it.

Example teach-and-repeat map-based navigation

Image sequence indexed by position pics/along the learned path



Example teach-and-repeat map-based navigation

Images stored in a prior map

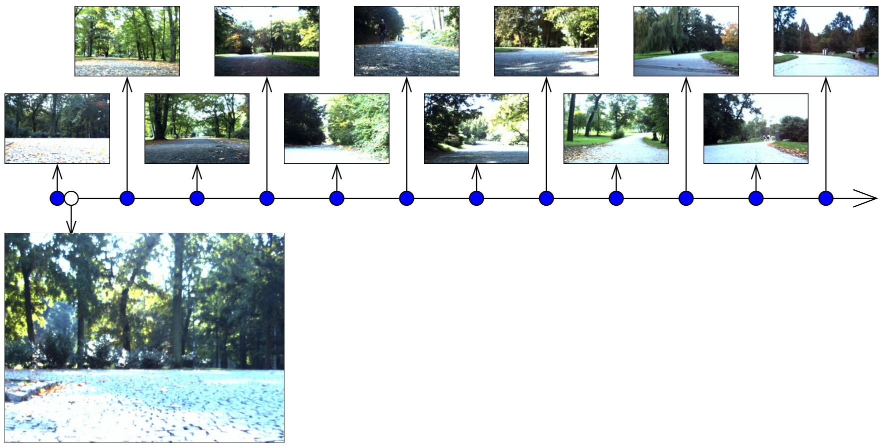


Image perceived by the robot during autonomous navigation

Example teach-and-repeat map-based navigation

Images stored in a prior map

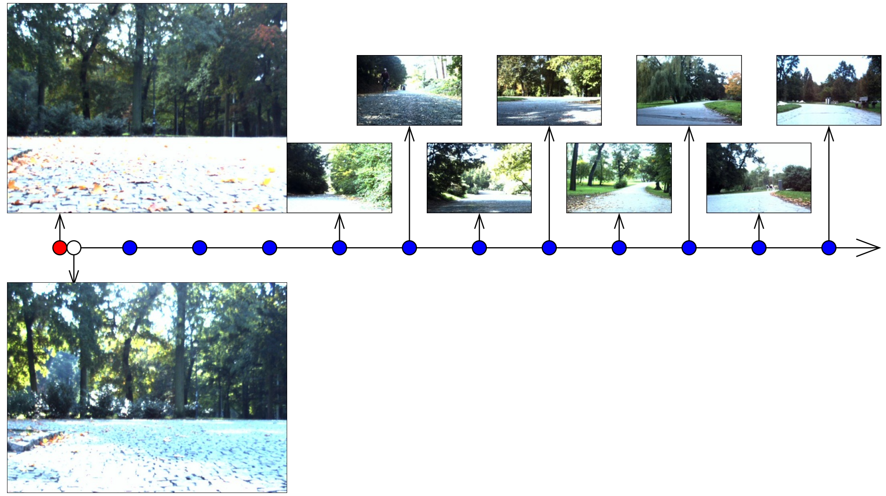


Image perceived by the robot during autonomous navigation

Example teach-and-repeat map-based navigation

Images stored in a prior map

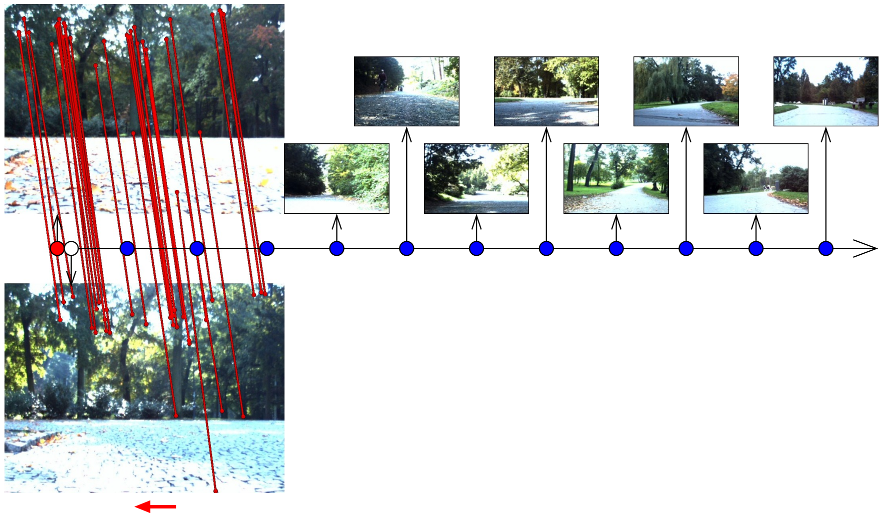


Image perceived by the robot during autonomous navigation

Example teach-and-repeat map-based navigation

Images stored in a prior map

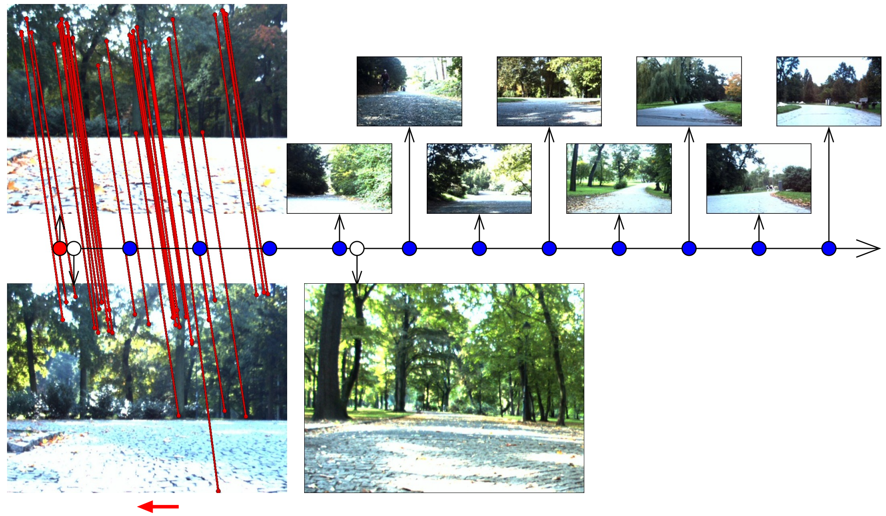


Image perceived by the robot during autonomous navigation

Example teach-and-repeat map-based navigation

Images stored in a prior map

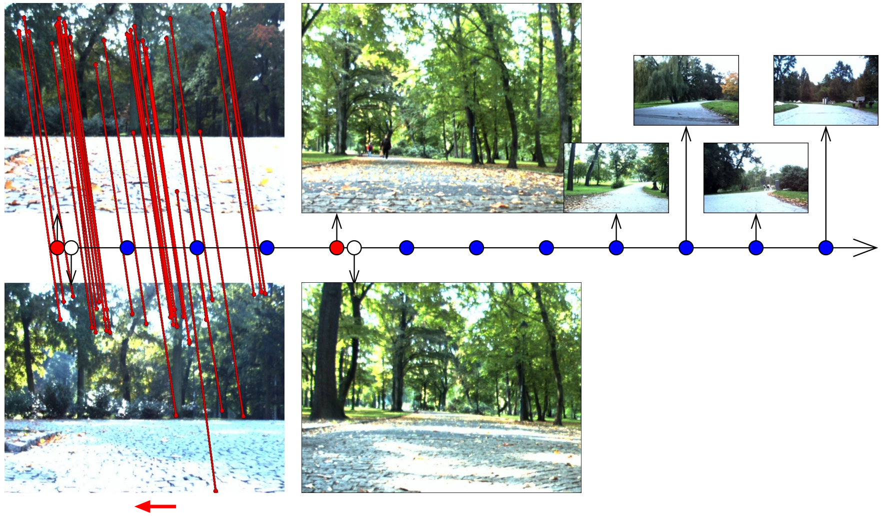


Image perceived by the robot during autonomous navigation

Example teach-and-repeat map-based navigation

Images stored in a prior map

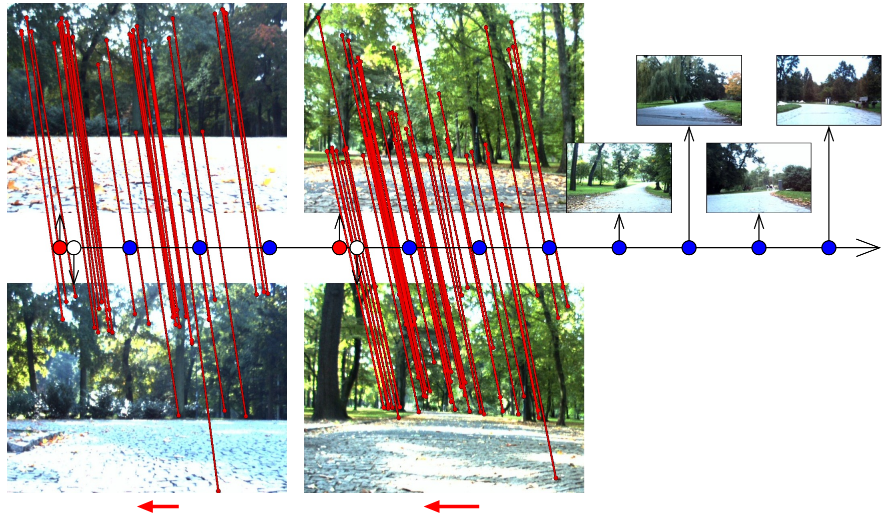


Image perceived by the robot during autonomous navigation

Example teach-and-repeat map-based navigation

Images stored in a prior map

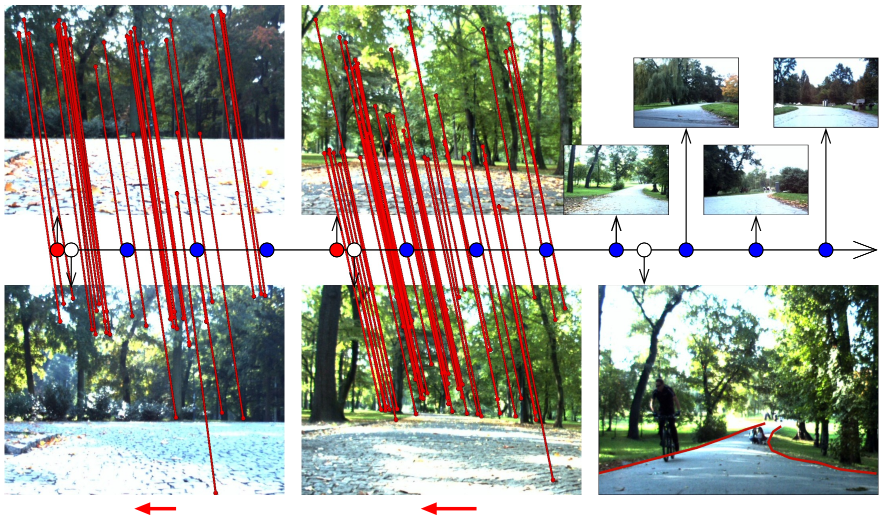


Image perceived by the robot during autonomous navigation

Example teach-and-repeat map-based navigation

Images stored in a prior map

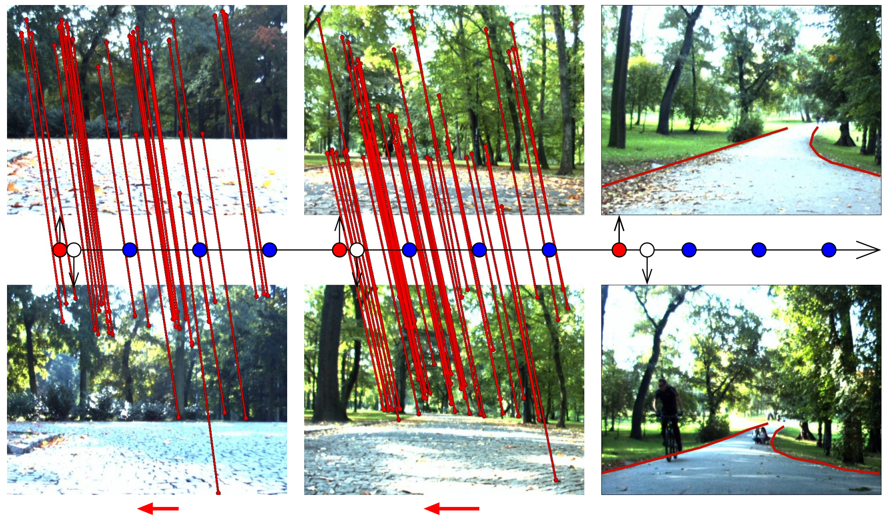


Image perceived by the robot during autonomous navigation

Example teach-and-repeat map-based navigation

Images stored in a prior map

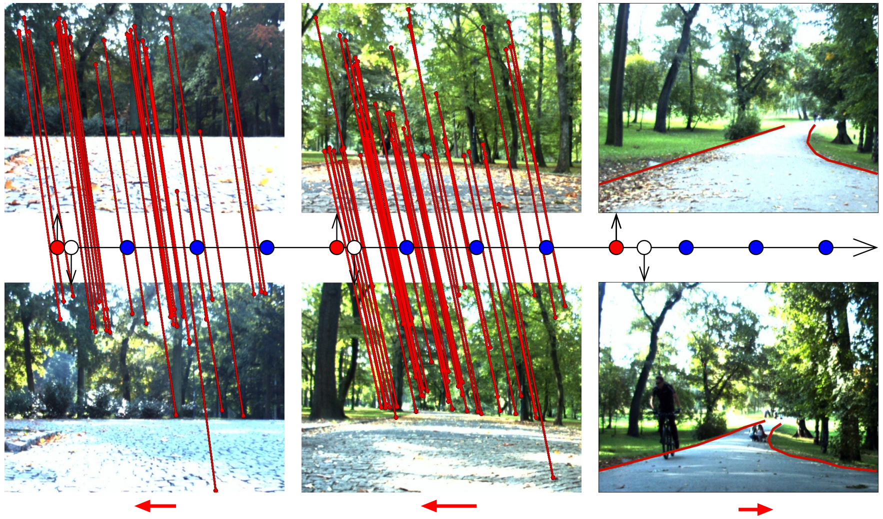
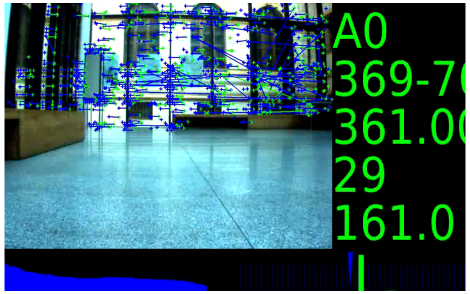


Image perceived by the robot during autonomous navigation

Example teach-and-repeat map-based navigation

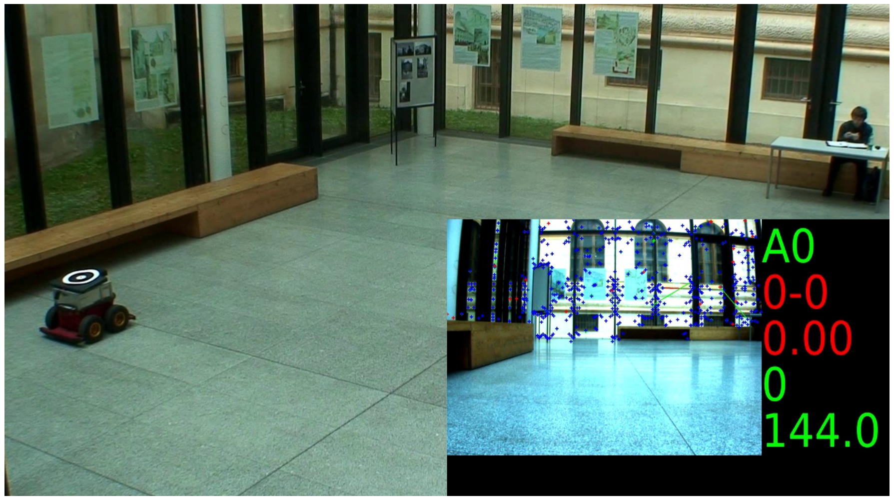
Autonomous navigation along a polygonal path
Discrete motion model

- Move forwards,
- get features from map,
- get features from image,
- establish matches,
- steer by histogram voting,
- stop and turn when odometry exceeds segment length.



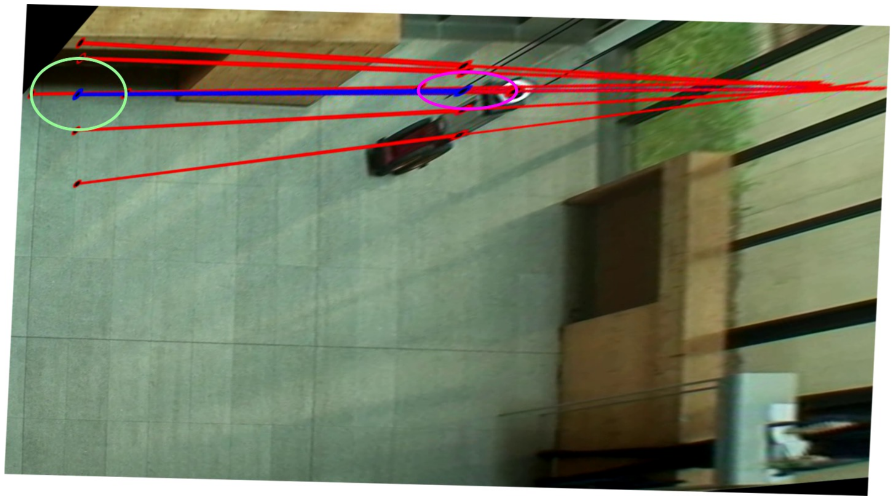
Video location: videos/navigate

Example map-based navigation - discrete error model



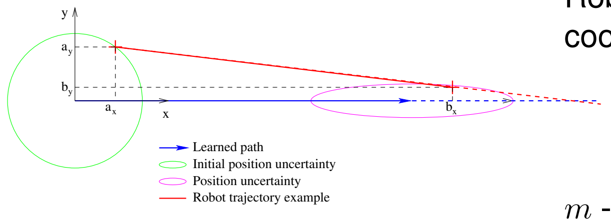
Video location: videos/segment

Example map-based navigation - discrete error model



Video location: videos/segment

Example map-based navigation - discrete error model

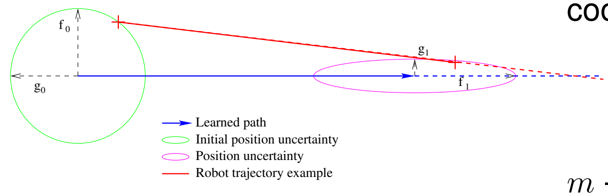


Robot position coordinates

$$b_x = a_x + s(1+\upsilon),$$
$$b_y = m a_y + \xi.$$

m - heading correction
ξ, υ - errors (odo+cam)

Example map-based navigation - discrete error model



Robot position coordinates

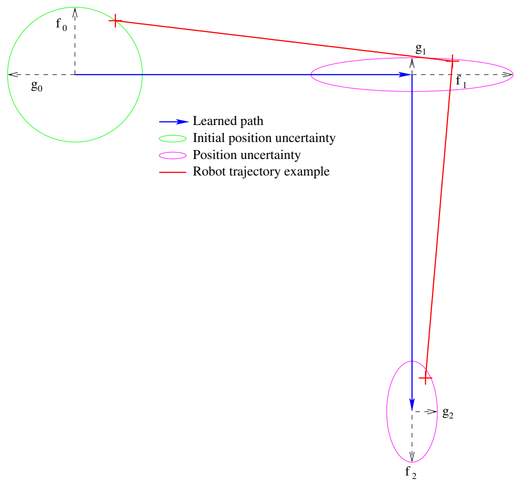
$$b_x = a_x + s(1+\upsilon),$$
$$b_y = m a_y + \xi.$$

m - heading correction
ξ, υ - errors (odo+cam)

Position error ellipse axes

$f_{i+1} = g_i + \upsilon,$
$g_{i+1} = m f_i + \xi$

Example map-based navigation - discrete error model



Robot position coordinates

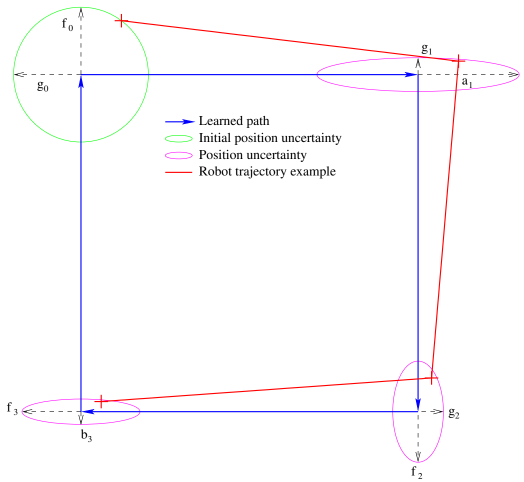
$$b_x = a_x + s(1+\upsilon),$$
$$b_y = m a_y + \xi.$$

m - heading correction
ξ, υ - errors (odo+cam)

Position error ellipse axes

$f_{i+1} = g_i + \upsilon,$
$g_{i+1} = m f_i + \xi$

Example map-based navigation - discrete error model



Robot position coordinates

$$b_x = a_x + s(1+\upsilon),$$
$$b_y = m a_y + \xi.$$

m - heading correction
ξ, υ - errors (odo+cam)

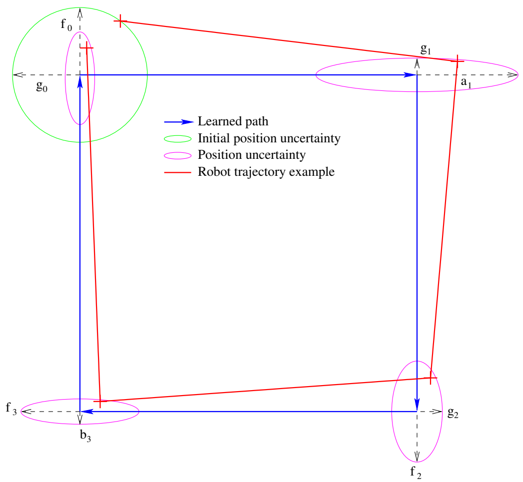
Position error ellipse axes

$f_{i+1} = g_i + \upsilon,$
$g_{i+1} = m f_i + \xi$

Convergence

$f_\infty = (\xi + \upsilon)/(1-m)$
f_∞, g_∞ finite if $\|m\| < 1$.

Example map-based navigation - discrete error model



Robot position coordinates

$$b_x = a_x + s(1+\upsilon),$$
$$b_y = m a_y + \xi.$$

m - heading correction
ξ, υ - errors (odo+cam)

Position error ellipse axes

$f_{i+1} = g_i + \upsilon,$
$g_{i+1} = m f_i + \xi$

Convergence

$f_\infty = (\xi + \upsilon)/(1-m)$
f_∞, g_∞ finite if $\|m\| < 1.$

Example map-based navigation - discrete error model

Robot position vector

$$\begin{pmatrix} b_x \\ b_y \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & m \end{pmatrix} \begin{pmatrix} a_x \\ a_y \end{pmatrix} + \begin{pmatrix} s + s\upsilon \\ \xi \end{pmatrix}$$

Becomes
$\mathbf{b} = \mathbf{R^T}(\mathbf{MRa} + \mathbf{s}) = \mathbf{Na} + \mathbf{t}$ for a segment with arbitrary azimuth $\mathbf{R}$

Position error covariance matrix
$\mathbf{A_{i+1}} = \mathbf{N_i A_i N_i^T} + \mathbf{T_i}$

Convergence
$\mathbf{A_\infty} = \breve{\mathbf{N}}\mathbf{A_\infty}\breve{\mathbf{N}} + \breve{\mathbf{T}}$ (Lyapunov equation) $\mathbf{A_\infty}$ exists and is finite iff $\|\breve{\mathbf{N}}\| < 1$
$\breve{\mathbf{N}} = \prod_{i=1}^{n} \mathbf{N_i} = \prod_{n=1}^{n} \mathbf{R_i^T N_i R_i}$

Robot position coordinates

$$b_x = a_x + s(1+\upsilon),$$
$$b_y = m a_y + \xi.$$

m - heading correction
ξ, υ - errors (odo+cam)

Position error ellipse axes

$f_{i+1} = g_i + \upsilon,$
$g_{i+1} = m f_i + \xi$

Convergence

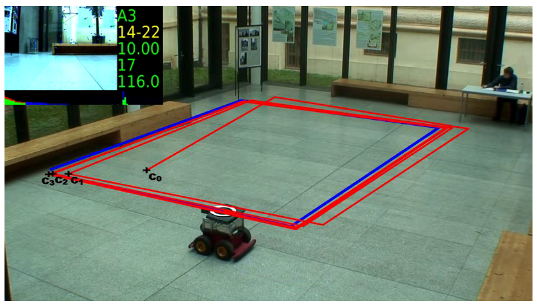
$f_\infty = (\xi + \upsilon)/(1-m)$
f_∞, g_∞ finite if $\|m\| < 1.$

Example map-based navigation - discrete error model

Experimental evaluation:

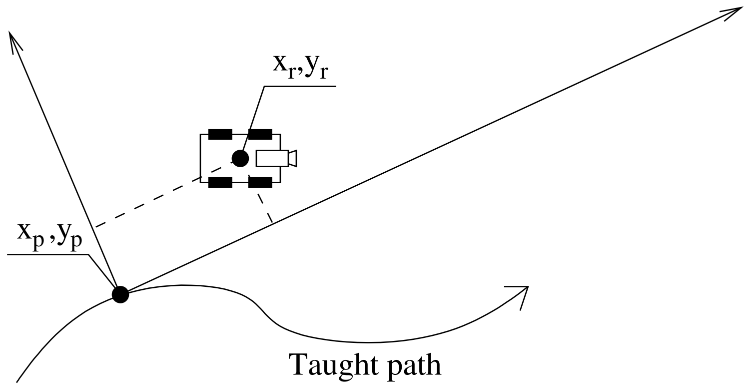
1. Teach a closed path,
2. displace at start,
3. traverse n times,
4. measure $\mathbf{c_i}$,
5. compute $\varepsilon_{acc}, \varepsilon_{acc}$.

$$\varepsilon_{acc} = \sqrt{\frac{1}{n-j}\sum_{i=j}^{n} \|\mathbf{c_i}\|^2}$$
$$\varepsilon_{rep} = \sqrt{\frac{1}{n-j}\sum_{i=j}^{n} \|\mathbf{c_i}, \mu\|^2}$$
$$\mu = \sum_{i=j}^{n} \mathbf{c_i}/(n-j)$$



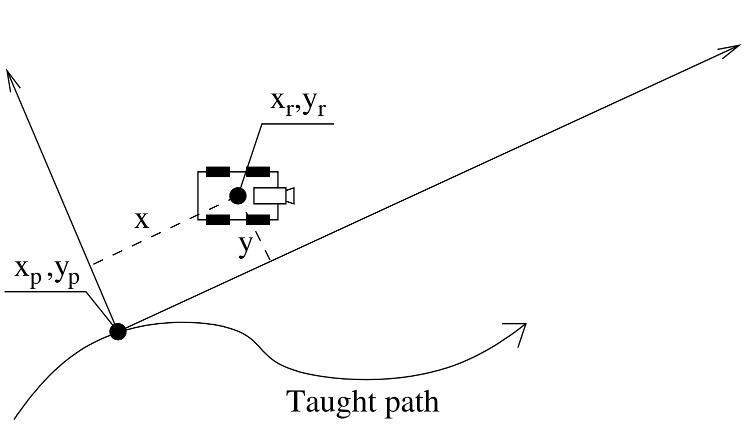
Video location: videos/converge

Example map-based navigation - continuous error model



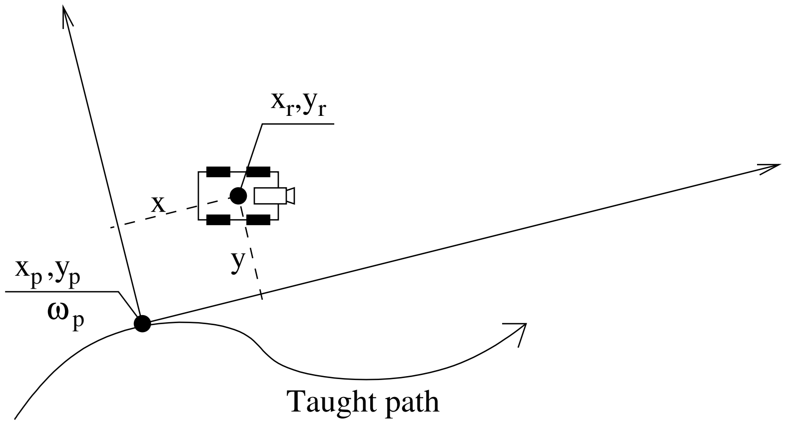
$$\dot{x} =$$
$$\dot{y} =$$

Example map-based navigation - continuous error model



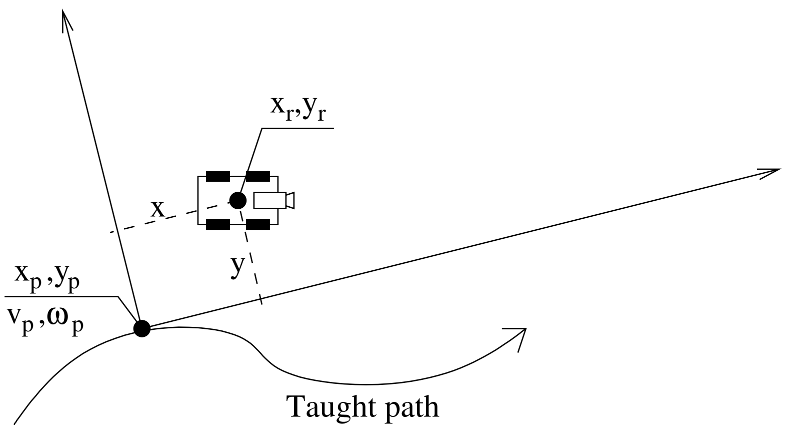
$$\begin{aligned}\dot{x} &= \\ \dot{y} &= \end{aligned}$$

Example map-based navigation - continuous error model



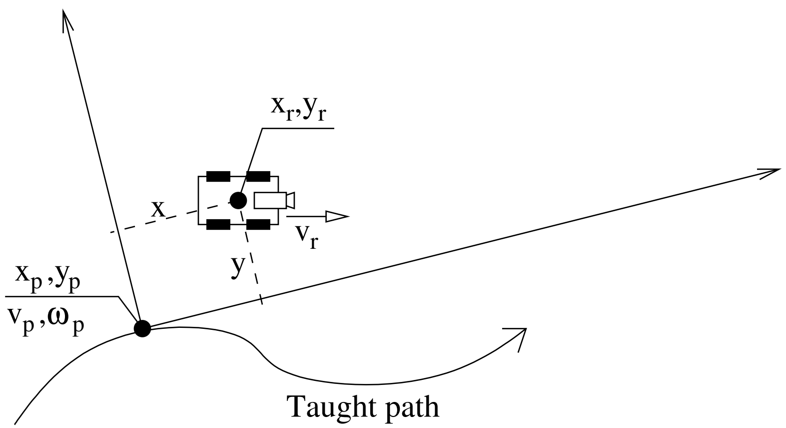
$$\begin{aligned}\dot{x} &= +\omega_p\, y \\ \dot{y} &= -\omega_p\, x\end{aligned}$$

Example map-based navigation - continuous error model



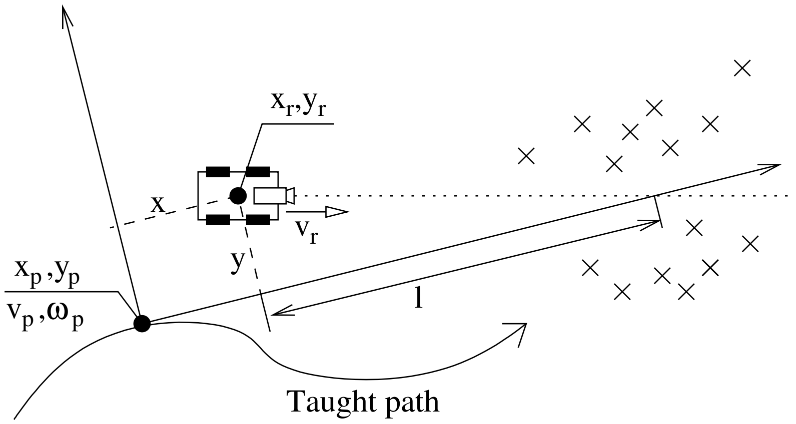
$$\begin{aligned}\dot{x} &= +\omega_p\, y \quad -v_p \\ \dot{y} &= -\omega_p\, x\end{aligned}$$

Example map-based navigation - continuous error model



$$\begin{aligned}\dot{x} &= +\omega_p\, y \quad -v_p \quad +v_r \\ \dot{y} &= -\omega_p\, x\end{aligned}$$

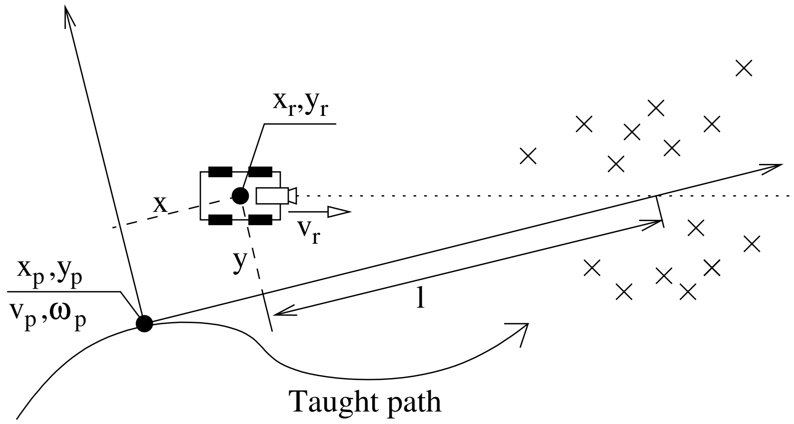
Example map-based navigation - continuous error model



$$
\begin{aligned}
\dot{x} &= +\omega_p\, y \quad -v_p \quad +v_r \\
\dot{y} &= -\omega_p\, x \quad\quad\quad -v_r\, y\, l^{-1}
\end{aligned}
$$

Tom Krajník Autonomous Navigation AIC@CTU

Example map-based navigation - continuous error model



$$
\begin{aligned}
\dot{x} &= +\omega_p\, y \quad -v_p \quad +v_r \quad\quad\quad +s_x \\
\dot{y} &= -\omega_p\, x \quad\quad\quad -v_r\, y\, l^{-1} \quad +s_y
\end{aligned}
$$

Krajnik, Faigl et al.: **Simple, Yet Stable Bearing only Navigation**. Journal of Field Robotics, 2010 12 / 15

Tom Krajník Autonomous Navigation AIC@CTU

Example map-based navigation - continuous error model

$$
\begin{aligned}
\dot{x} &= + \;\omega_p\, y \;-\; v_p \;+\; v_r \;+\; s_x \\
\dot{y} &= - \;\omega_p\, x \;-\; v_r\, y\, l^{-1} \;+\; s_y,
\end{aligned}
$$

Tom Krajník Autonomous Navigation AIC@CTU

Example map-based navigation - continuous error model

$$
\begin{aligned}
\dot{x} &= + \;\omega_p\, y \;-\; v_p \;+\; v_r \;+\; s_x \\
\dot{y} &= - \;\omega_p\, x \;-\; v_r\, y\, l^{-1} \;+\; s_y,
\end{aligned}
$$

Matrix form:

$$
\begin{pmatrix} \dot{x} \\ \dot{y} \end{pmatrix} = \begin{pmatrix} 0 & +\omega_p \\ -\omega_p & -v_r\, l^{-1} \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} s_x \\ s_y \end{pmatrix},
$$

Krajnik, Majer et al.: **Navigation without localisation: reliable teach and repeat based on ....** In IROS, 2018 13 / 15

Example map-based navigation - continuous error model

$$\begin{array}{rcrlrlrlrl} \dot{x} & = & + & \omega_p\, y & - & v_p & + & v_r & + & s_x \\ \dot{y} & = & - & \omega_p\, x & & & - & v_r\, y\, l^{-1} & + & s_y, \end{array}$$

Matrix form:

$$\begin{pmatrix} \dot{x} \\ \dot{y} \end{pmatrix} = \begin{pmatrix} 0 & +\omega_p \\ -\omega_p & -v_r\, l^{-1} \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} s_x \\ s_y \end{pmatrix},$$

Matrix eigenvalues:

$$\lambda^2 + \lambda \frac{v_r}{l} + \omega_p^2 = 0,$$

Tom Krajník Autonomous Navigation AIC@CTU

Example map-based navigation - continuous error model

$$\begin{array}{rcrlrlrlrl} \dot{x} & = & + & \omega_p\, y & - & v_p & + & v_r & + & s_x \\ \dot{y} & = & - & \omega_p\, x & & & - & v_r\, y\, l^{-1} & + & s_y, \end{array}$$

Matrix form:

$$\begin{pmatrix} \dot{x} \\ \dot{y} \end{pmatrix} = \begin{pmatrix} 0 & +\omega_p \\ -\omega_p & -v_r\, l^{-1} \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} s_x \\ s_y \end{pmatrix},$$

Matrix eigenvalues:

$$\lambda^2 + \lambda \frac{v_r}{l} + \omega_p^2 = 0,$$

$$v_r > 0, \quad l > 0 \implies Re(\lambda_{1,2}) < 0 \;\; \texttt{iff} \;\; \omega_p \neq 0.$$

Krajnik, Majer et al.: **Navigation without localisation: reliable teach and repeat based on ....** In IROS, 2018 13 / 15

Tom Krajník Autonomous Navigation AIC@CTU

Example map-based navigation - continuous error model

$$\begin{array}{rcrlrlrlrl} \dot{x} & = & + & \omega_p\, y & - & v_p & + & v_r & + & s_x \\ \dot{y} & = & - & \omega_p\, x & & & - & v_r\, y\, l^{-1} & + & s_y, \end{array}$$

Matrix form:

$$\begin{pmatrix} \dot{x} \\ \dot{y} \end{pmatrix} = \begin{pmatrix} 0 & +\omega_p \\ -\omega_p & -v_r\, l^{-1} \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} s_x \\ s_y \end{pmatrix},$$

Matrix eigenvalues:

$$\lambda^2 + \lambda \frac{v_r}{l} + \omega_p^2 = 0,$$

$$v_r > 0, \quad l > 0 \implies Re(\lambda_{1,2}) < 0 \;\; \texttt{iff} \;\; \omega_p \neq 0.$$

Position error decreases if path is not only a straight line.

Krajnik, Majer et al.: **Navigation without localisation: reliable teach and repeat based on ....** In IROS, 2018 13 / 15

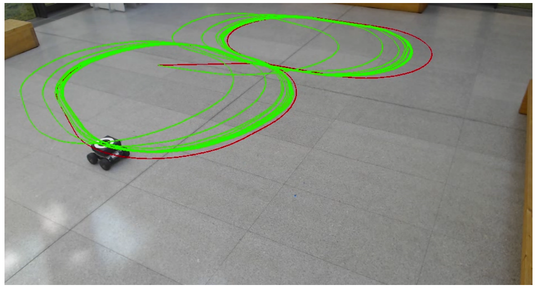
Tom Krajník Autonomous Navigation AIC@CTU

Example map-based navigation - continuous error model

Experimental evaluation similar to the discrete case

1. Teach a closed path,
2. displace at start,
3. traverse n times,
4. measure $\mathbf{c_i}$,
5. compute ε_{acc}, ε_{acc}.

$$\begin{array}{rcl} \varepsilon_{acc} & = & \sqrt{\frac{1}{n-j} \sum_{i=j}^{n} \|\mathbf{c_i}\|^2} \\ \varepsilon_{rep} & = & \sqrt{\frac{1}{n-j} \sum_{i=j}^{n} \|\mathbf{c_i}, \mu\|^2} \\ \mu & = & \sum_{i=j}^{n} \mathbf{c_i}/(n-j) \end{array}$$



Video location: videos/continuos

Lecture wrap up: what to remember

What to remember

- Mobile robot navigation type can be divided by the map usage, which assume different prior knowledge and thus handle the robot observations in different ways. [1]
- Although simple, mapless navigation is commonly used in commercially-successfull systems (Roomba, Tesla etc). [2]
- Map-based navigation typically uses localisation [3], but it is not necessary for teach-and-repeat systems [4].

References

- [1] DeSouza et al.: *Vision for mobile robot navigation: A survey.* IEEE PAMI, 2002
- [2] Brooks: *Intelligence without representation.* AI 1991
- [3] Filiat and Meyer: *Map-based navigation in mobile robots: A review..* Cog.Sys. Research 2003.
- [4] Krajnik et al.: *Navigation without localisation ...* In IROS 2018.