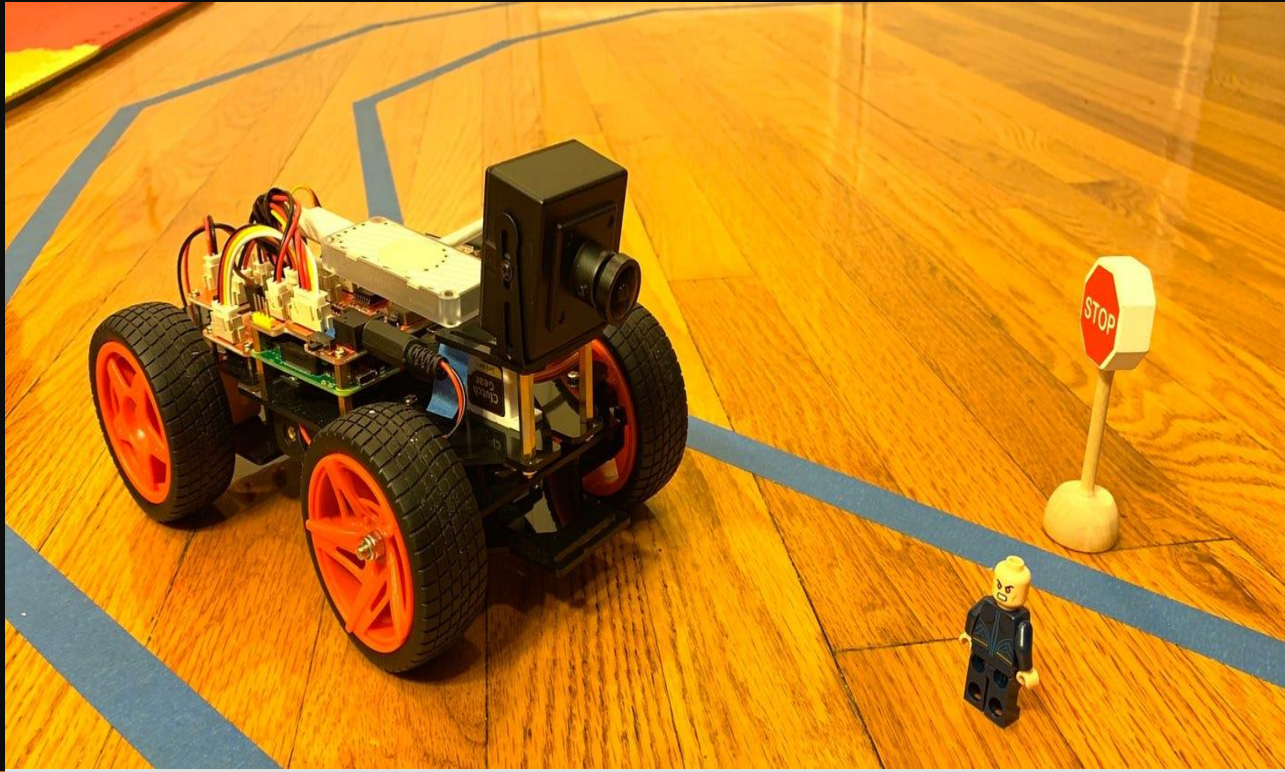
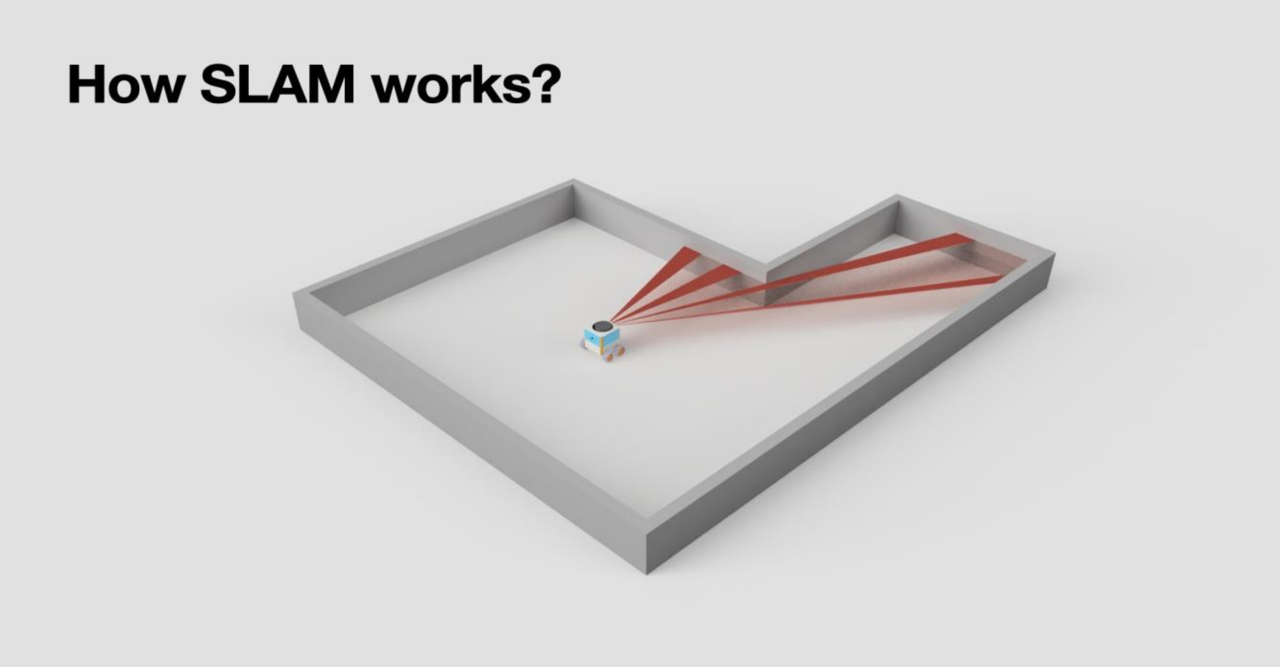
SLAM

Prepared By: Youssef Hindawi





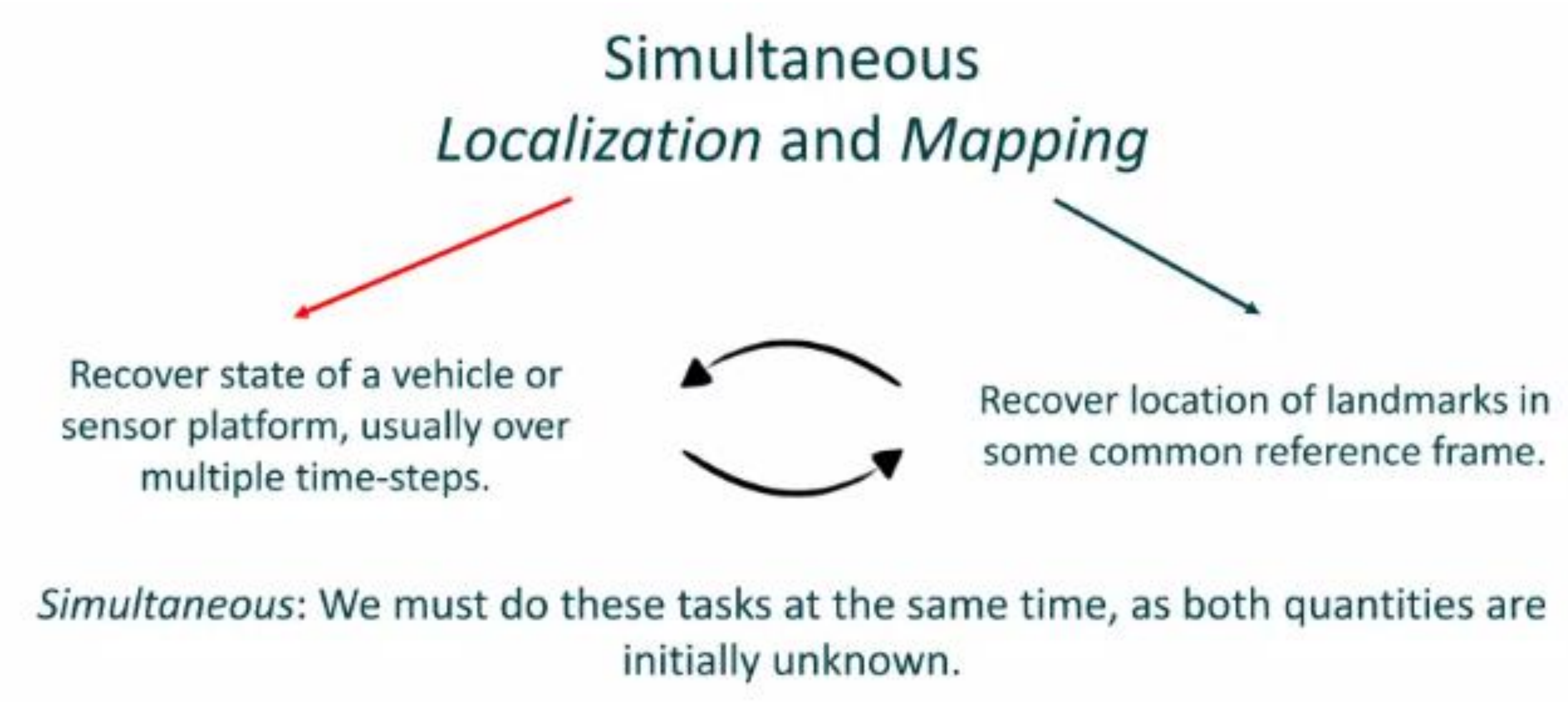
By the end of this Presentation, you will be able to:

- Types of SLAM
- Online SLAM & Full SLAM
- SLAM techniques
- Grid-Based FastSLAM
- Particles filter & Scan matching to solve SLAM challenges
- Experiment with Gmapping package
- Know Gmapping parameters and how to tune it

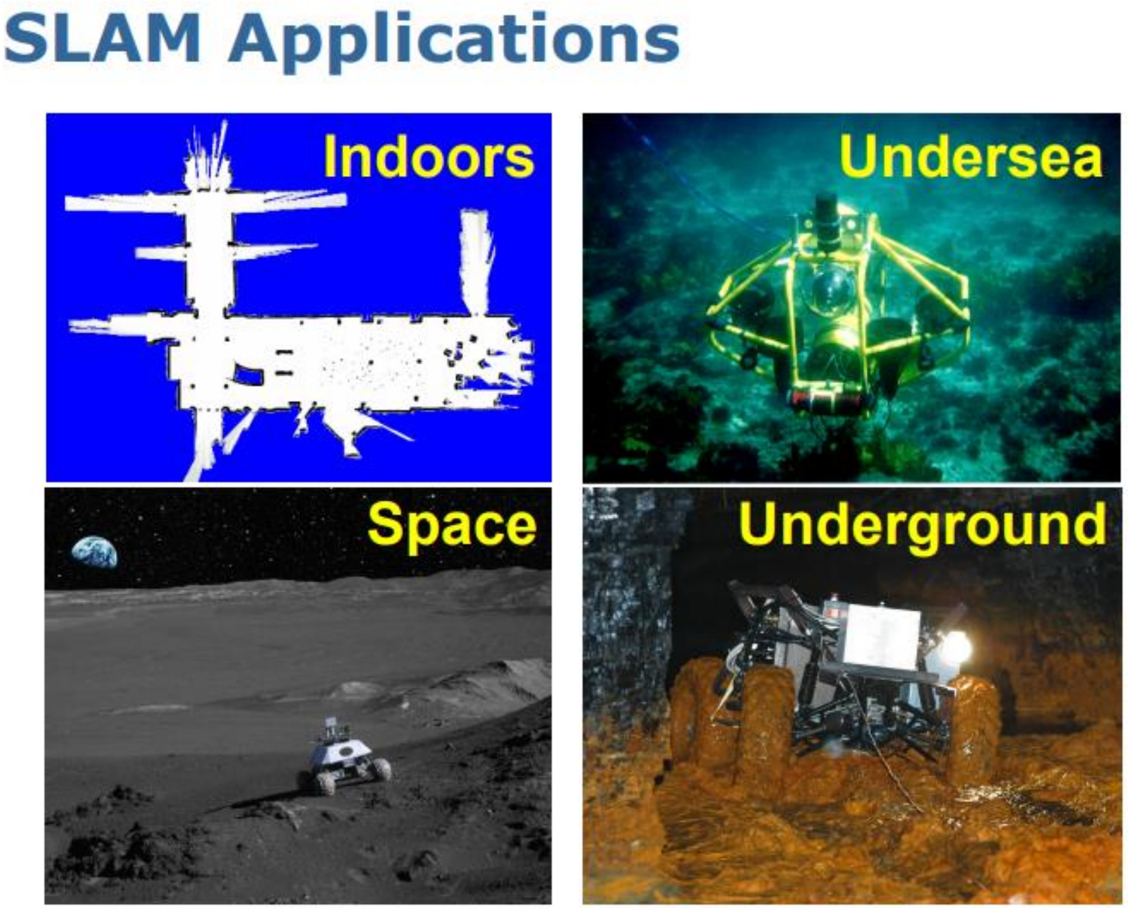
SLAM

- What is SLAM?
- Estimate the pose of a robot and the map of the environment at the same time
- SLAM is hard, because
  - a map is needed for localization and
  - a good pose estimate is needed for mapping
- Localization: inferring location given a map
- Mapping: inferring a map given locations
- SLAM: learning a map and locating the robot simultaneously

SLAM


Simultaneous
Localization and *Mapping*
Recover state of a vehicle or sensor platform, usually over multiple time-steps.
Recover location of landmarks in some common reference frame.
Simultaneous: We must do these tasks at the same time, as both quantities are initially unknown.

SLAM Applications



The SLAM Problem

- SLAM is considered a fundamental problems for robots to become truly autonomous
- Large variety of different SLAM approaches have been developed
- The majority uses probabilistic concepts

SLAM Techniques

- EKF SLAM
- FastSLAM
- Graph-based SLAM
- Topological SLAM (mainly place recognition)
- Scan Matching / Visual Odometry (only locally consistent maps)

Feature-Based SLAM

Given:

- The robot's controls

  $$\boldsymbol{U}_{1:k} = \{\boldsymbol{u}_1, \boldsymbol{u}_2, \ldots, \boldsymbol{u}_k\}$$

- Relative observations

  $$\boldsymbol{Z}_{1:k} = \{\boldsymbol{z}_1, \boldsymbol{z}_2, \ldots, \boldsymbol{z}_k\}$$

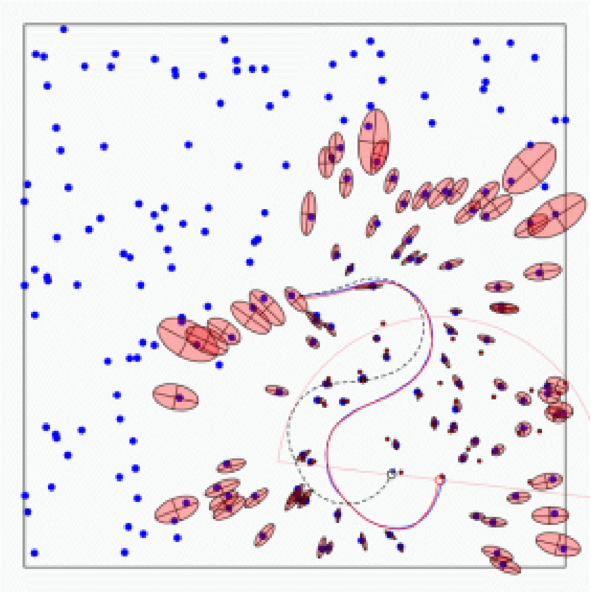
Wanted:

- Map of features

  $$\boldsymbol{m} = \{\boldsymbol{m}_1, \boldsymbol{m}_2, \ldots, \boldsymbol{m}_n\}$$

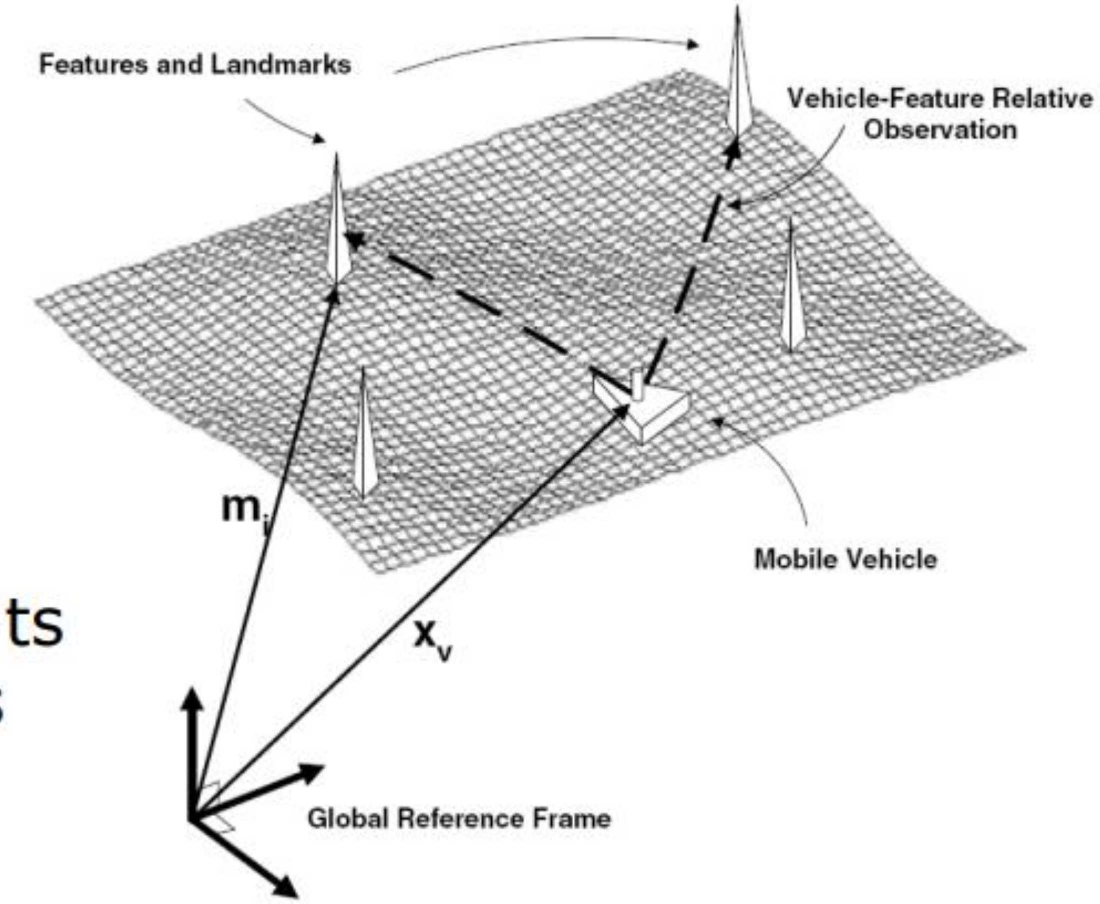
- Path of the robot

  $$\boldsymbol{X}_{1:k} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_k\}$$



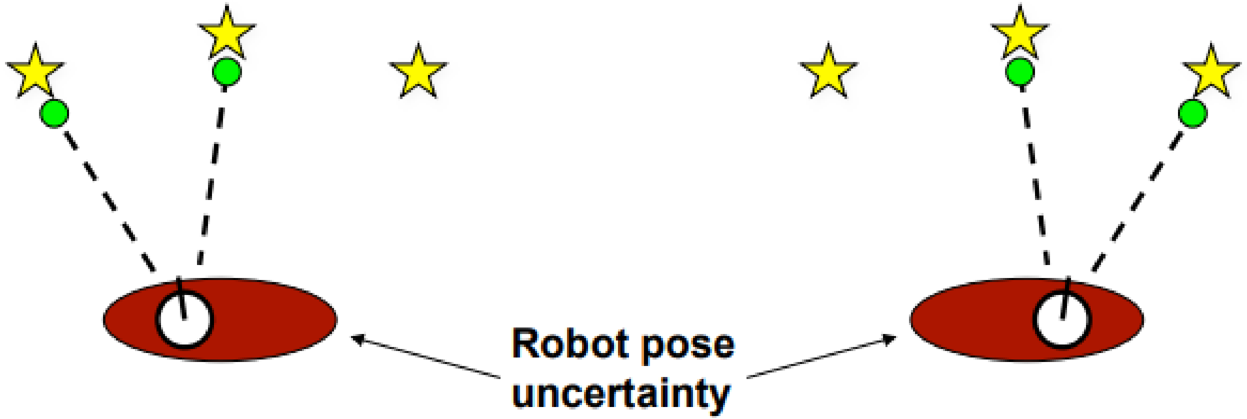
SLAM

- **Absolute** robot poses
- **Absolute** landmark positions
- But only **relative** measurements of landmarks

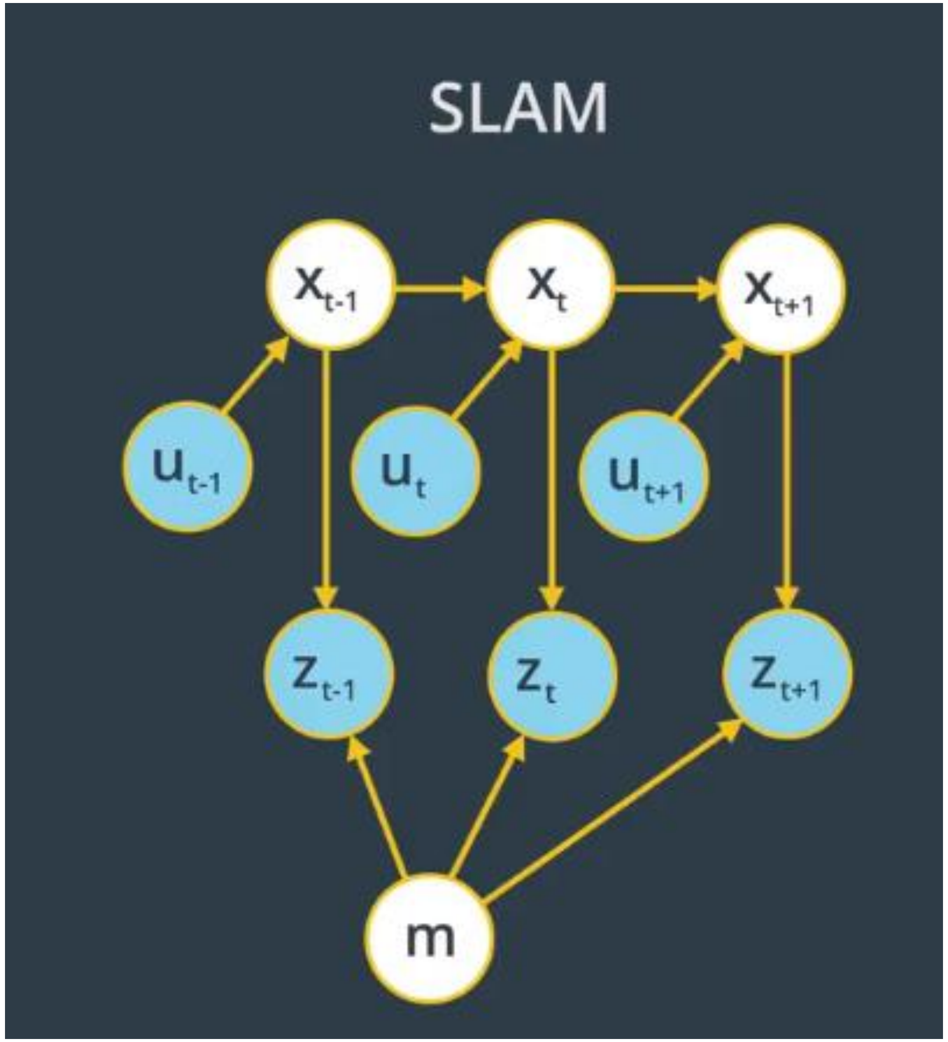


Why is SLAM a hard problem?

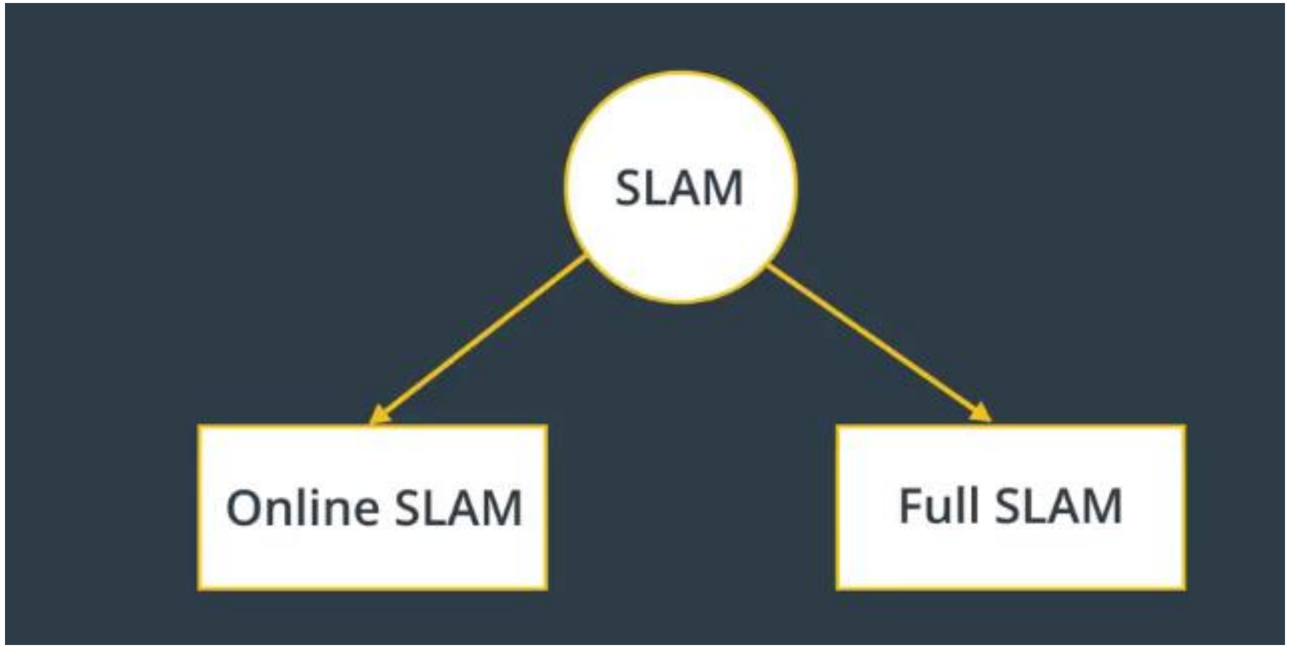
- The **mapping between observations and landmarks is unknown**
- Picking **wrong** data associations can have **catastrophic** consequences (divergence)



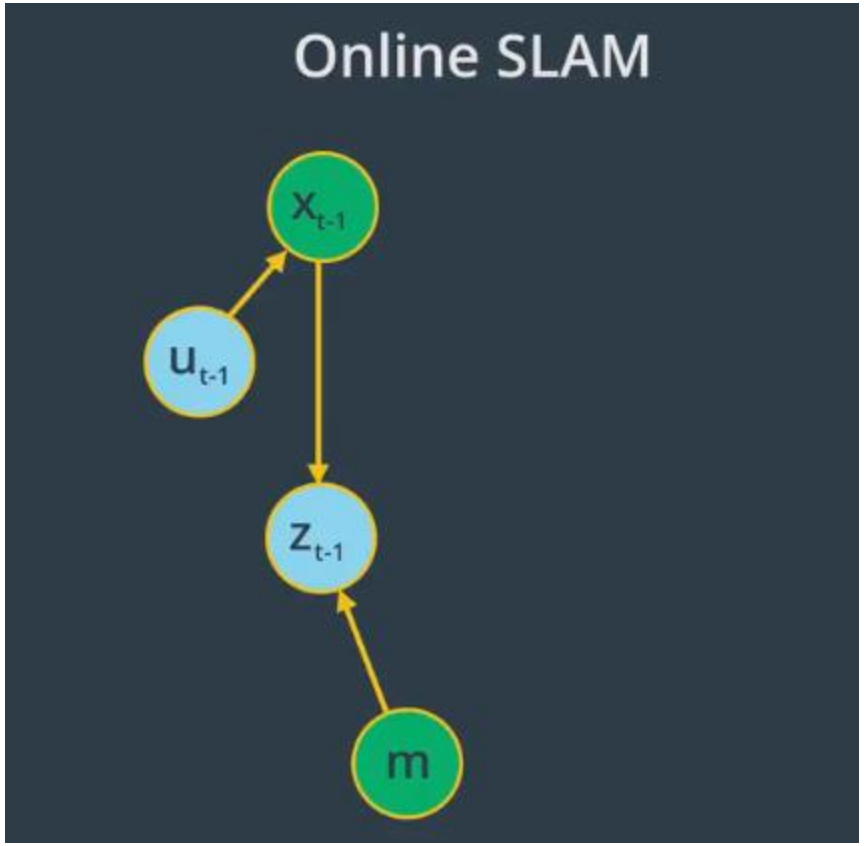
SLAM



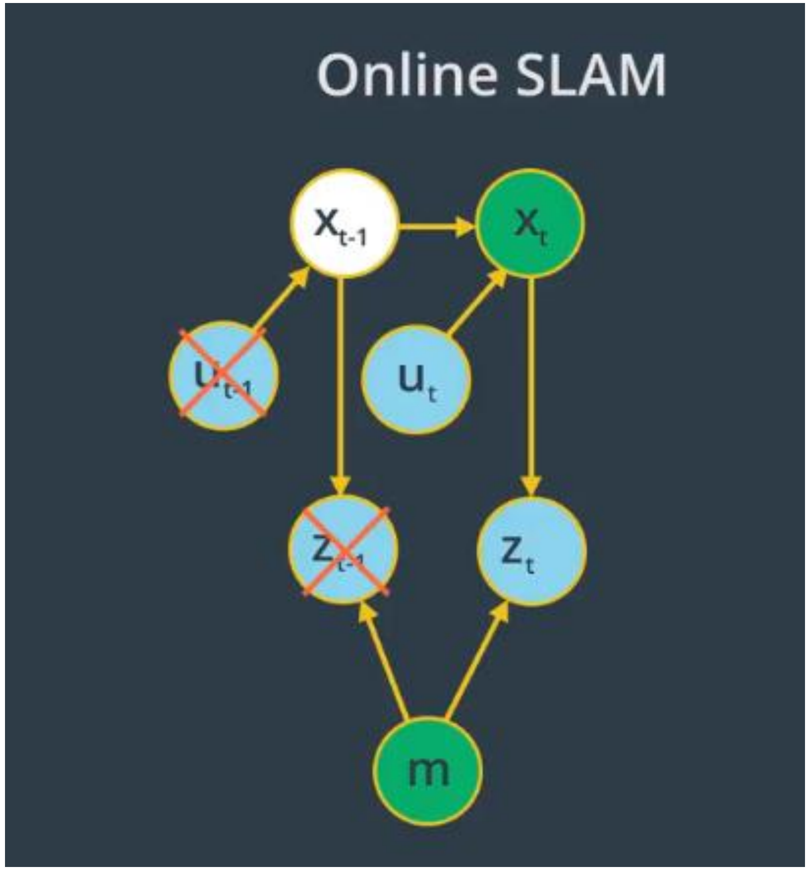
SLAM



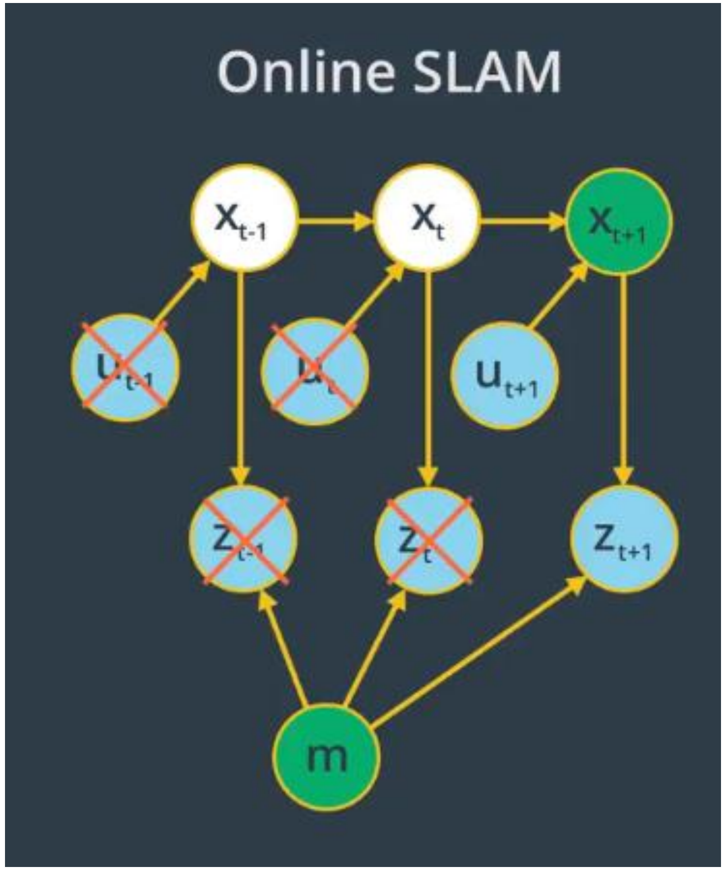
SLAM



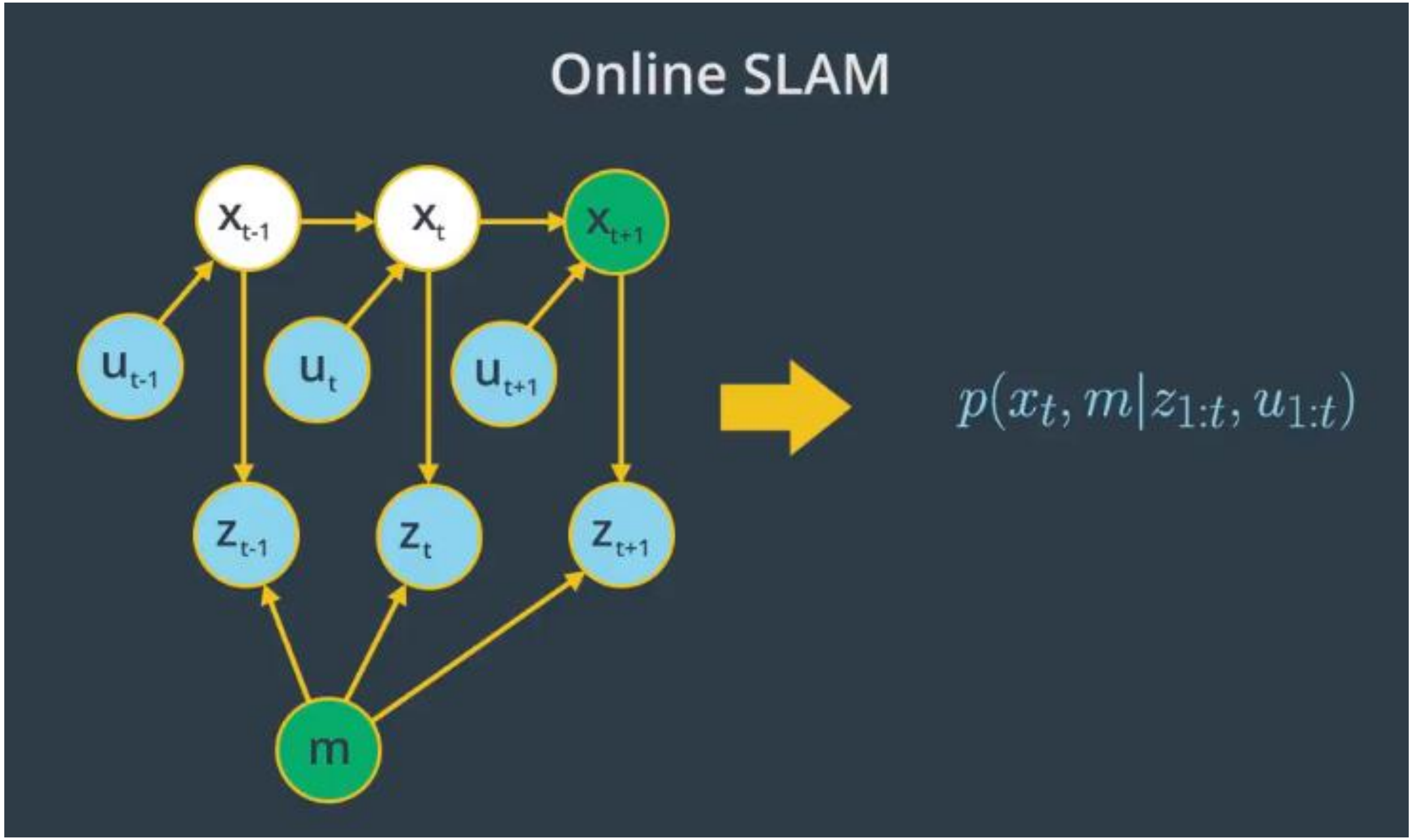
SLAM



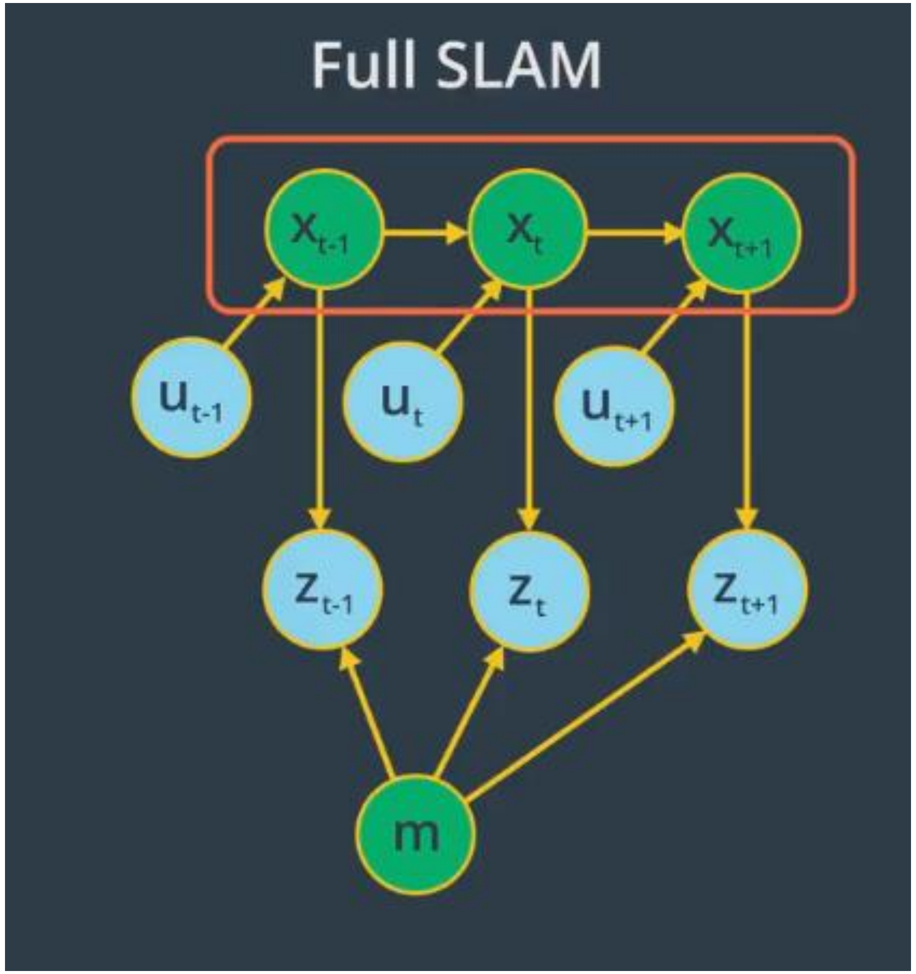
SLAM



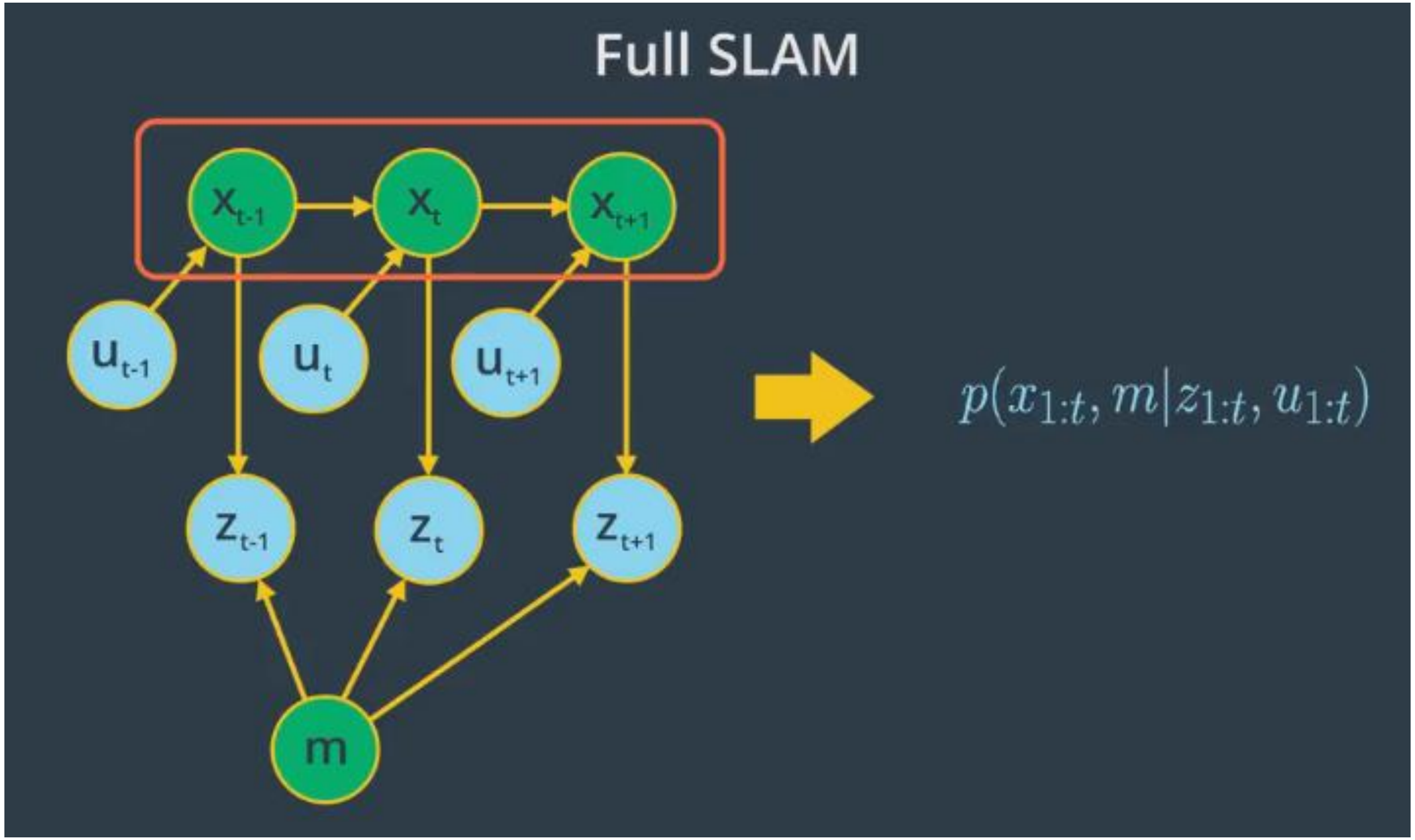
SLAM



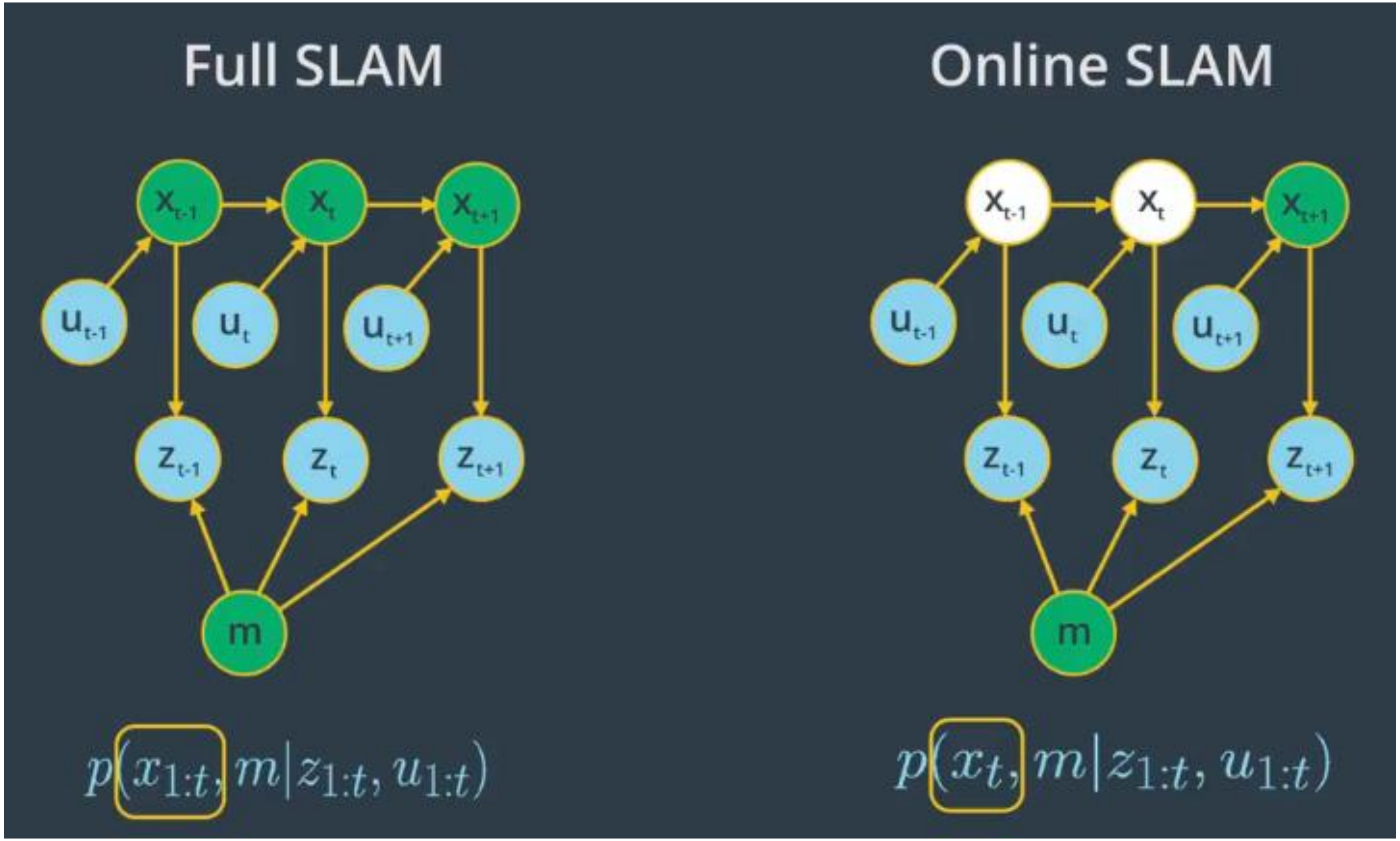
SLAM



SLAM

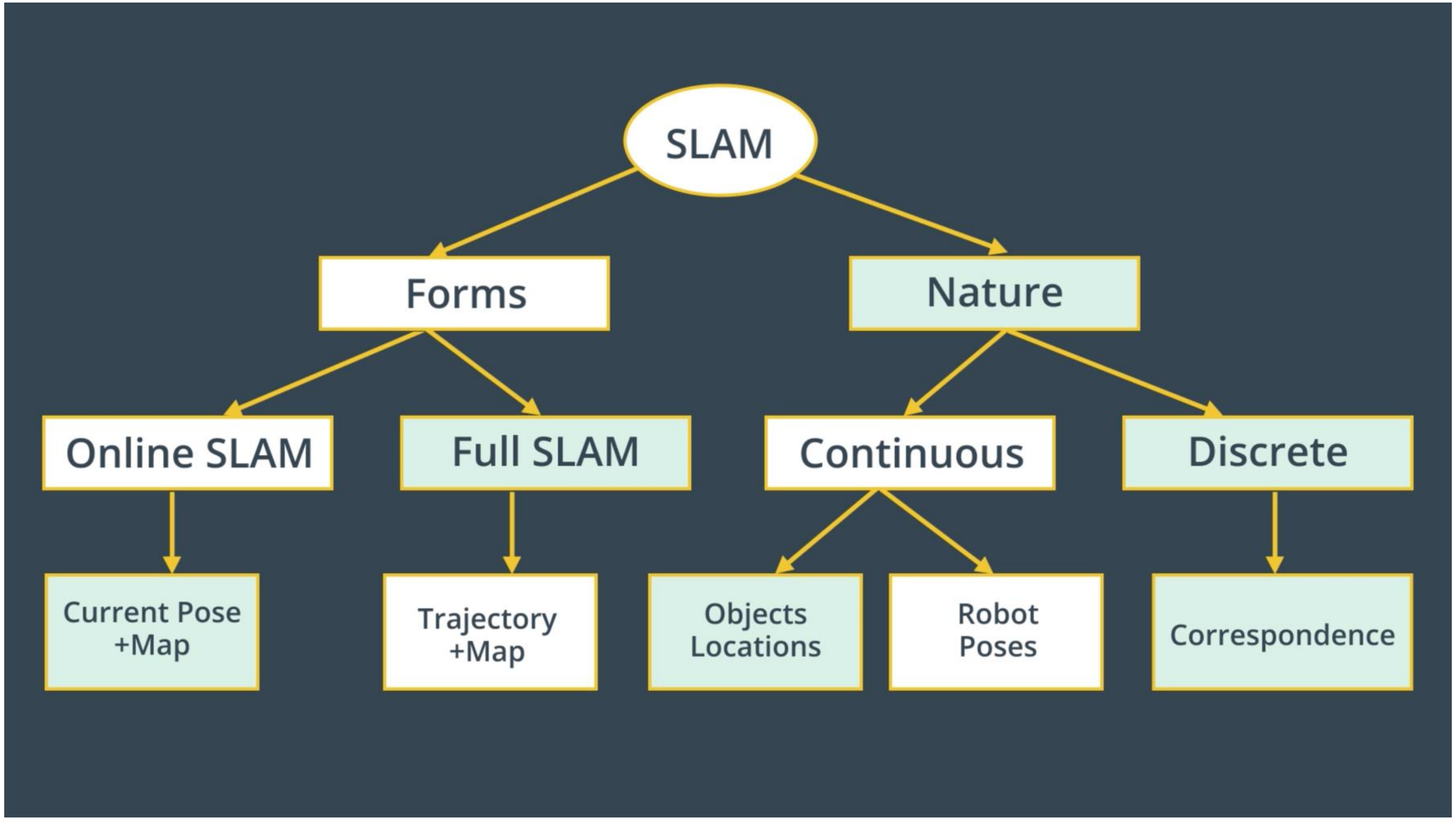


SLAM

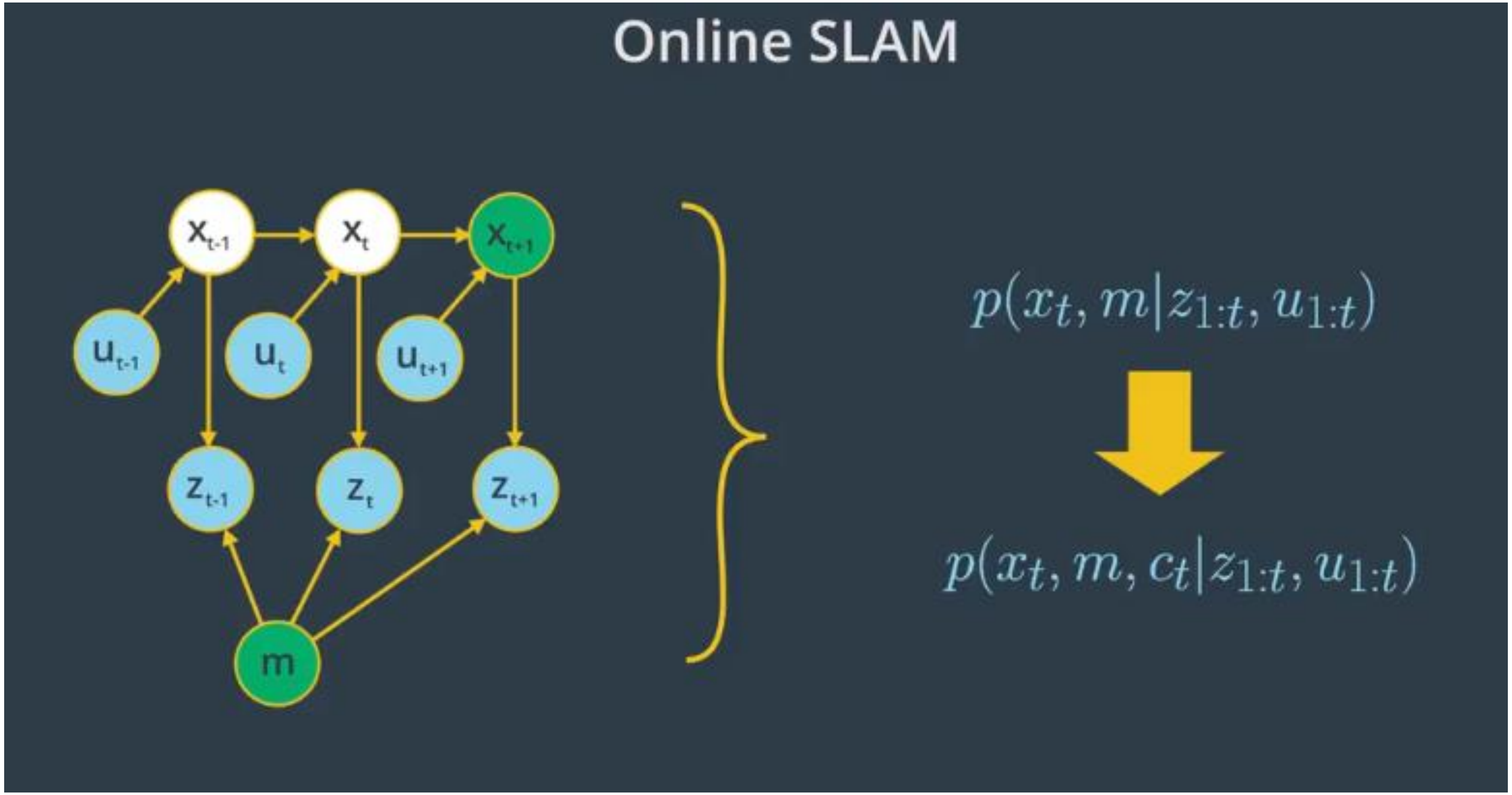


Full SLAM
Online SLAM
x_{t-1} x_t x_{t+1}
u_{t-1} u_t u_{t+1}
z_{t-1} z_t z_{t+1}
m
$p(\boxed{x_{1:t},} m|z_{1:t}, u_{1:t})$
$p(\boxed{x_t,} m|z_{1:t}, u_{1:t})$

SLAM

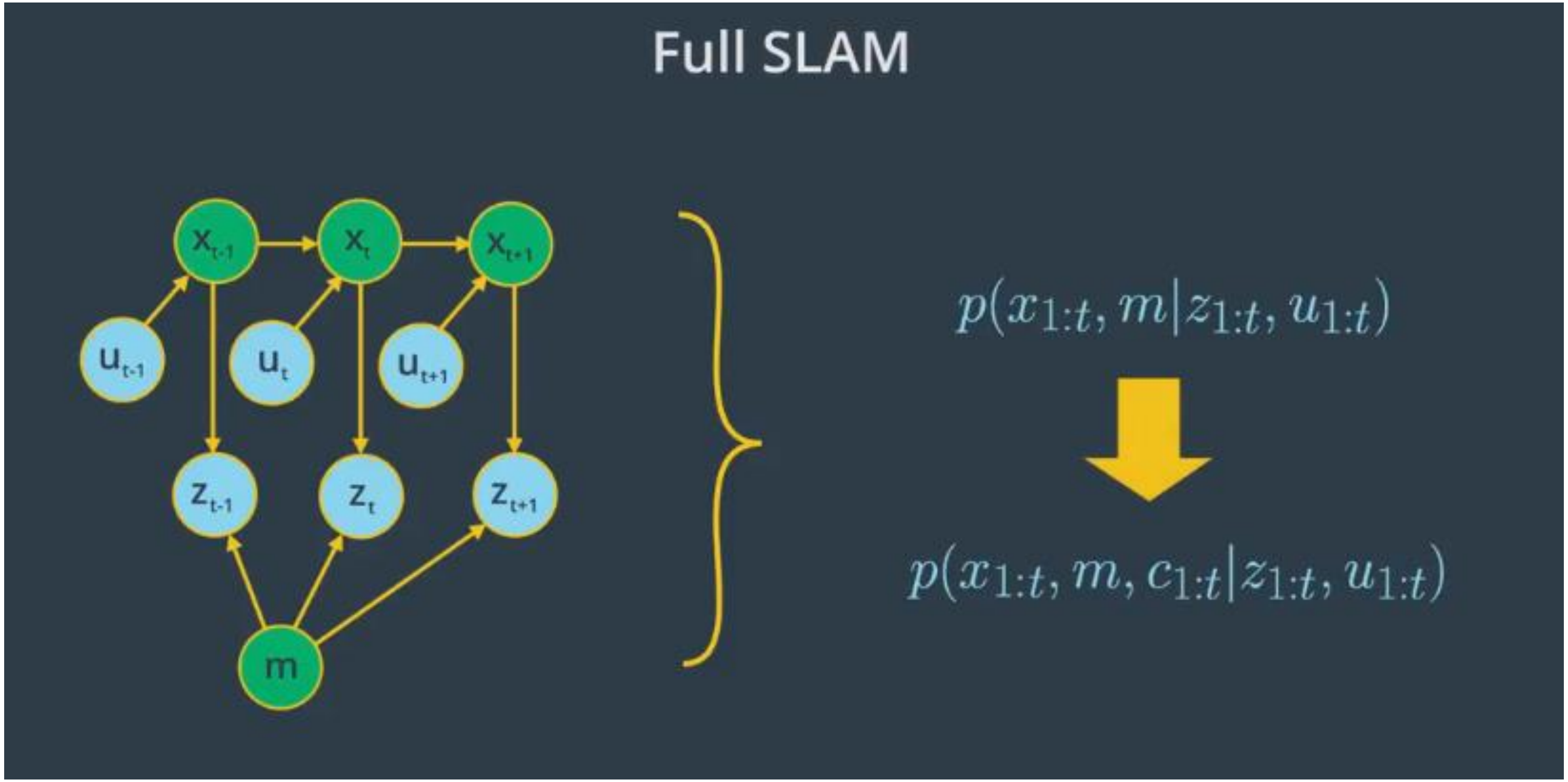

SLAM
Forms
Nature
Online SLAM
Full SLAM
Continuous
Discrete
Current Pose +Map
Trajectory +Map
Objects Locations
Robot Poses
Correspondence

SLAM



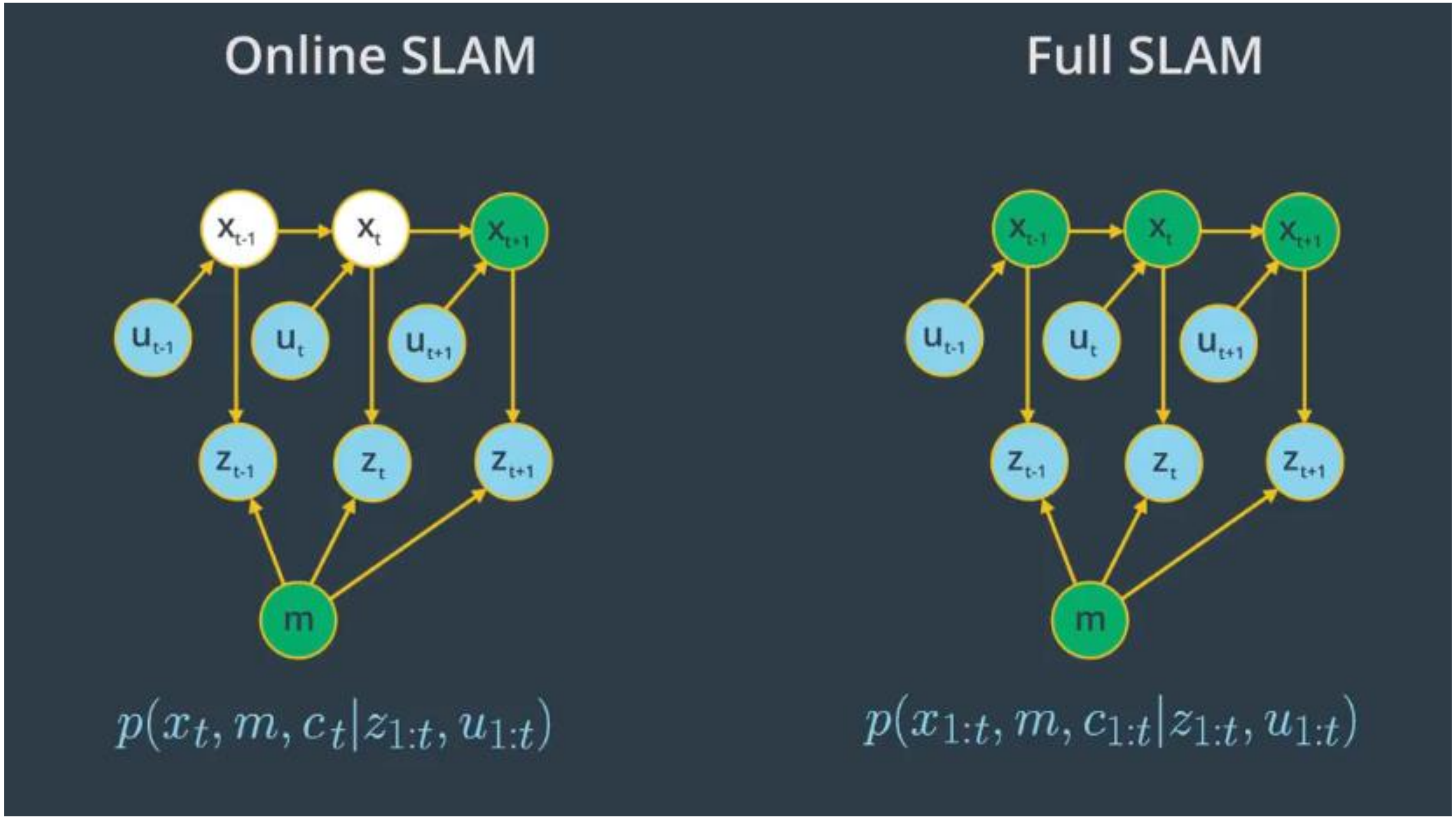
Online SLAM
x_{t-1}
x_t
x_{t+1}
u_{t-1}
u_t
u_{t+1}
z_{t-1}
z_t
z_{t+1}
m
$p(x_t, m | z_{1:t}, u_{1:t})$
$p(x_t, m, c_t | z_{1:t}, u_{1:t})$

SLAM



Full SLAM
x_{t-1}
x_t
x_{t+1}
u_{t-1}
u_t
u_{t+1}
z_{t-1}
z_t
z_{t+1}
m
$p(x_{1:t}, m | z_{1:t}, u_{1:t})$
$p(x_{1:t}, m, c_{1:t} | z_{1:t}, u_{1:t})$

SLAM



Online SLAM
Full SLAM
x_{t-1}
x_t
x_{t+1}
u_{t-1}
u_t
u_{t+1}
z_{t-1}
z_t
z_{t+1}
m
$p(x_t, m, c_t | z_{1:t}, u_{1:t})$
$p(x_{1:t}, m, c_{1:t} | z_{1:t}, u_{1:t})$

Grid-Based FastSLAM

- Factorization of the SLAM posterior

poses map observations & movements

$$p(x_{0:t}, m \mid z_{1:t}, u_{1:t})$$

Grid-Based FastSLAM

- Factorization of the SLAM posterior

poses map observations & movements

$$p(x_{0:t}, m \mid z_{1:t}, u_{1:t})$$

$$= p(x_{0:t} \mid z_{1:t}, u_{1:t})\, p(m \mid x_{1:t}, z_{1:t})$$

path posterior (particle filter)

map posterior (given the path)

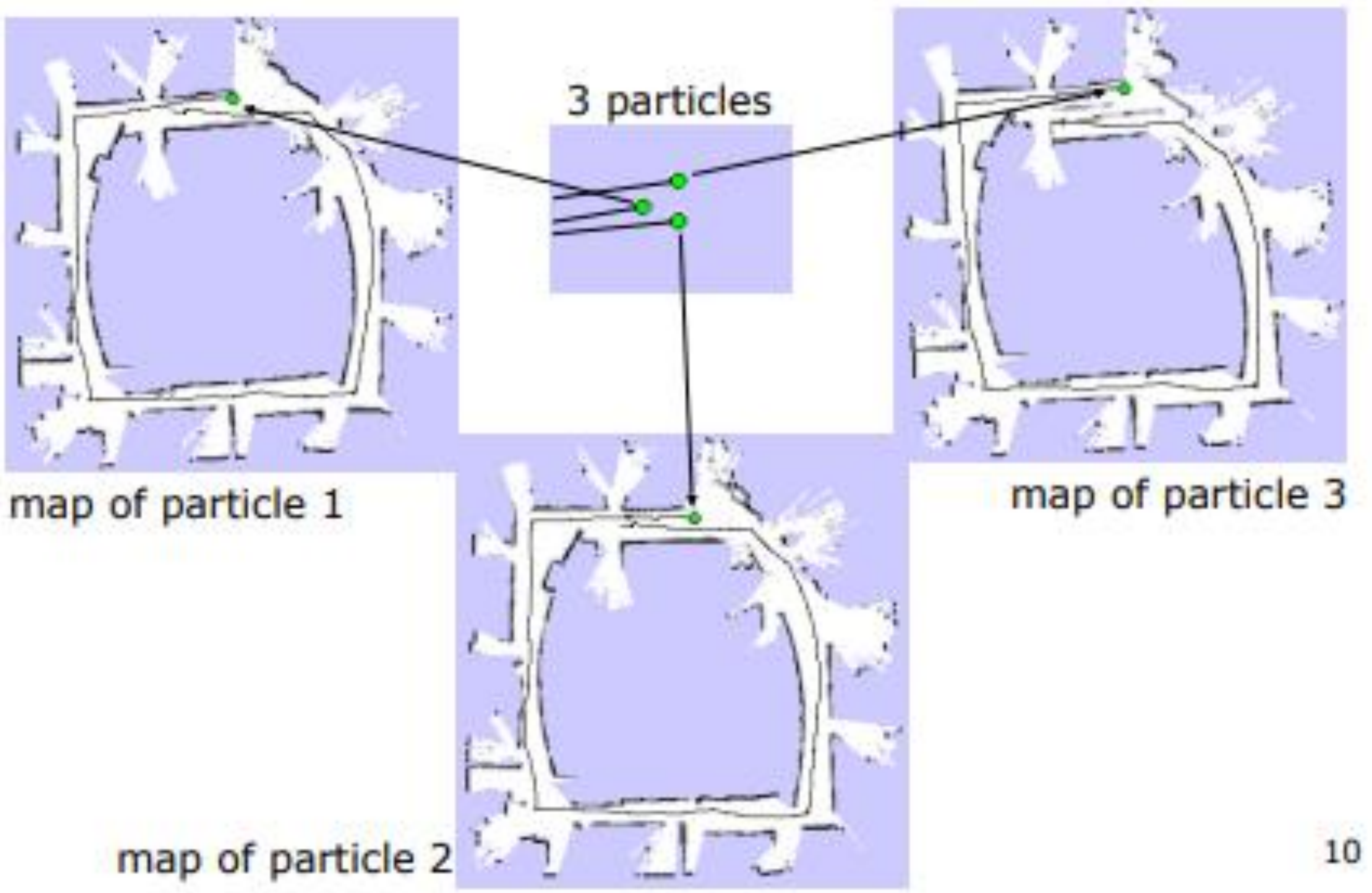
Grid-Based FastSLAM

- As with landmarks, the map depends on the poses of the robot during data acquisition
- If the poses are known, grid-based mapping is easy (“mapping with known poses”)

Grid-Based Mapping with RaoBlackwellized Particle Filters

- Each particle represents a possible trajectory of the robot
- Each particle maintains its own map
- Each particle updates it upon "mapping with known poses"

Particle Filter Example

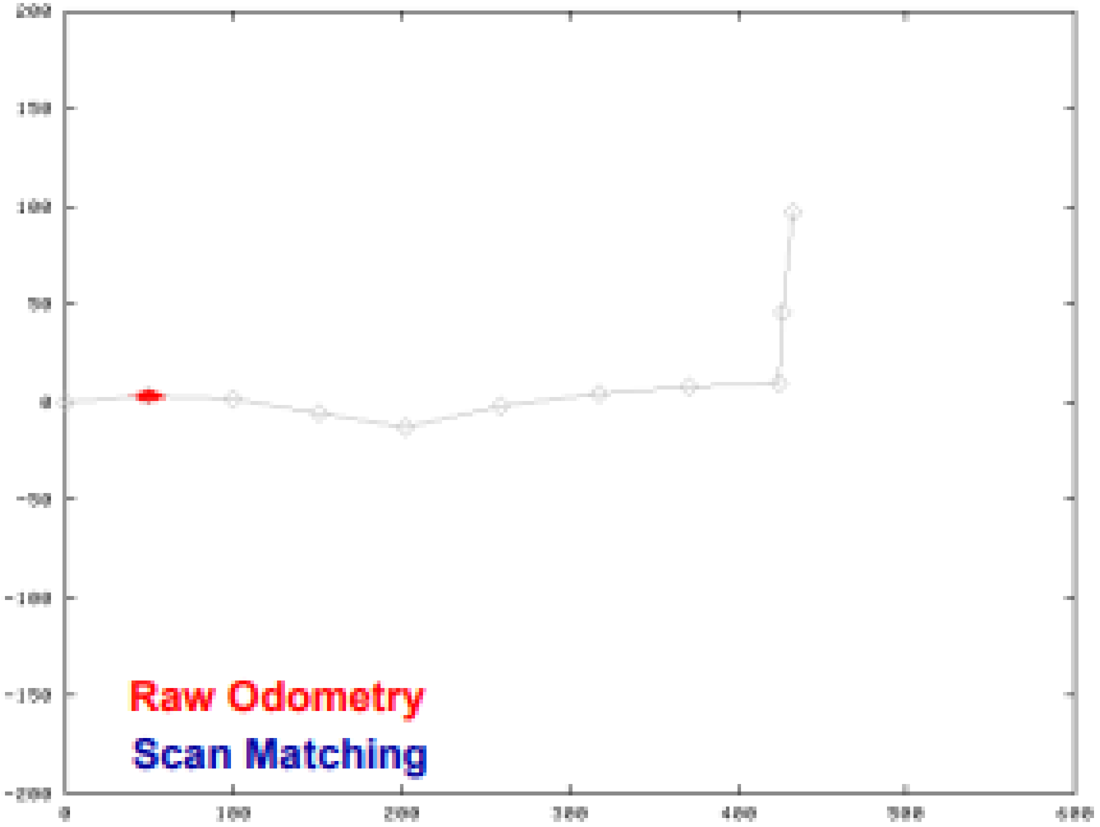


Pose Correction Using ScanMatching

Maximize the likelihood of the **current** pose and map relative to the **previous** pose and map

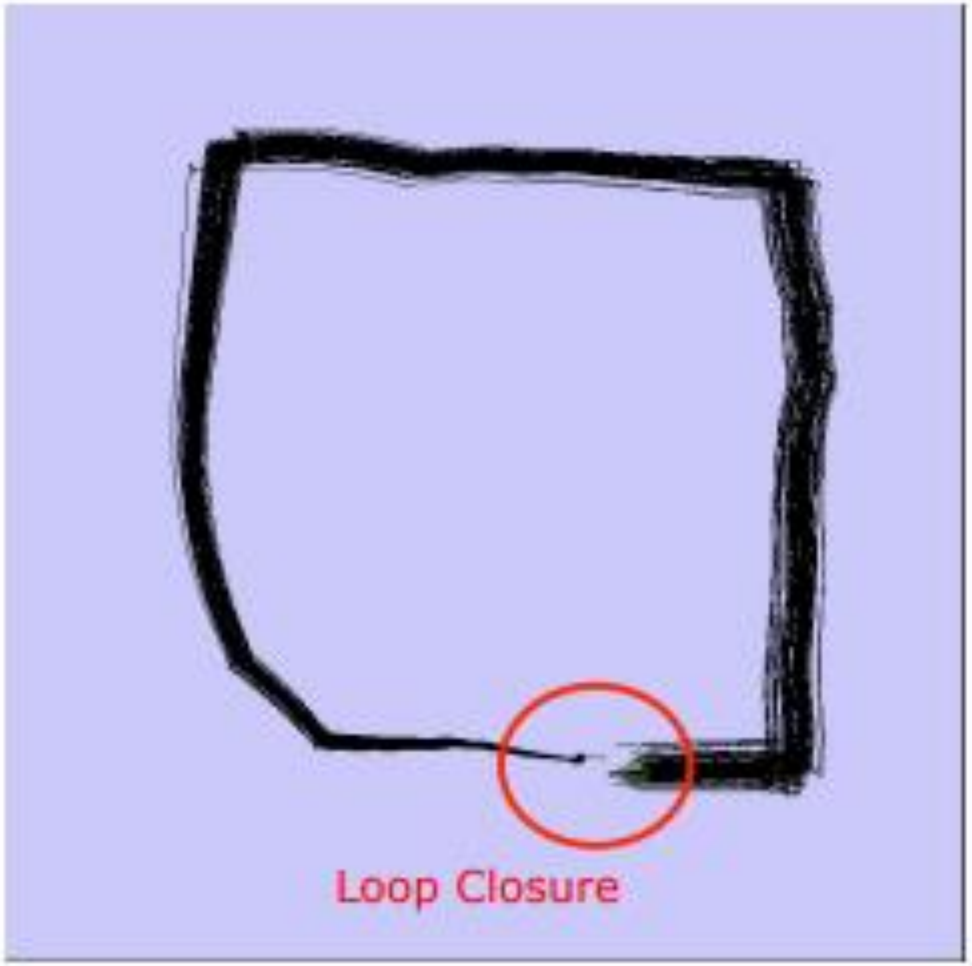
$$x_t^* = \underset{x_t}{\operatorname{argmax}} \left\{ p(\underbrace{z_t}_{\text{current measurement}} \mid x_t, \underbrace{m_{t-1}}_{\text{map constructed so far}})\, p(x_t \mid \underbrace{u_t, x_{t-1}^*}_{\text{robot motion}}) \right\}$$

Motion Model for Scan Matching



Courtesy: Dirk Hähnel

Grid-Based FastSLAM with Scan-Matching



Courtesy:
Dirk Hähnel

Grid-Based FastSLAM with Scan-Matching

Courtesy:
Dirk Hähnel

What we have done so far !

- Approach to SLAM that combines scan matching and FastSLAM
- Scan matching to generate virtual ‘high quality’ motion commands

- Can be seen as an ad-hoc solution to an improved proposal distribution

What's left

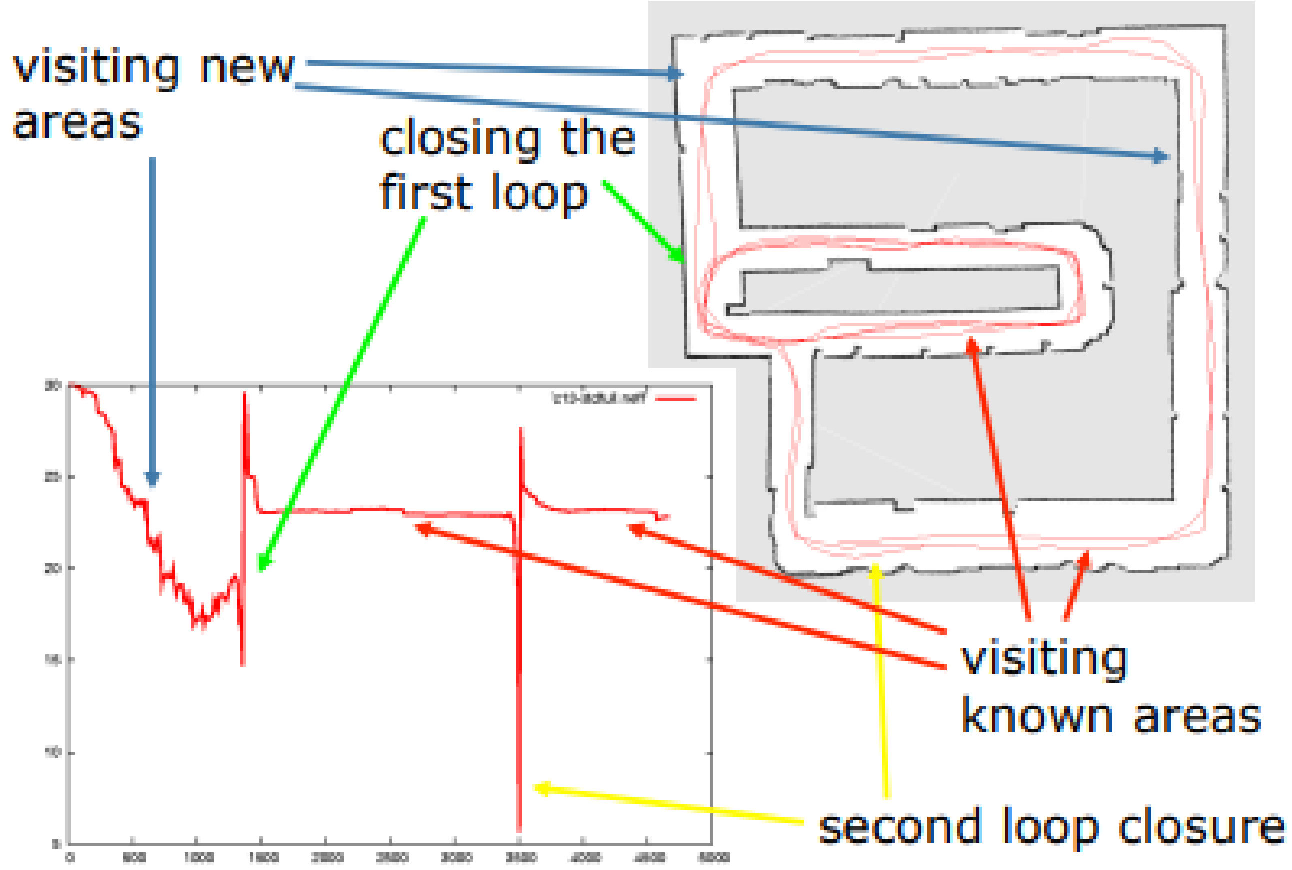
- Compute an improved proposal that considers the most recent observation

$$x_t^{[k]} \sim p(x_t \mid x_{1:t-1}^{[k]}, u_{1:t}, z_{1:t})$$

Goals:

- More precise sampling
- More accurate maps
- Less particles needed

Loop Closure



SLAM Practice

- **15 particles**
- four times faster than real-time P4, 2.8GHz
- 5cm resolution during scan matching
- 1cm resolution in final map

SLAM Practice

- **30 particles**
- 250x250m^2
- 1.75 km (odometry)
- 30cm resolution in final map

Review

- Types of SLAM
- Online SLAM & Full SLAM
- SLAM techniques
- Grid-Based FastSLAM
- Particles filter & Scan matching to solve SLAM challenges
- Experiment with Gmapping package
- Know Gmapping parameters and how to tune it

By the end of this session you should be able to:

- Know different types of SLAM
- Know the difference between online SLAM and full SLAM
- Know that Gmapping is based on Grid-Based FastSLAM
- Tune Gmapping parameters