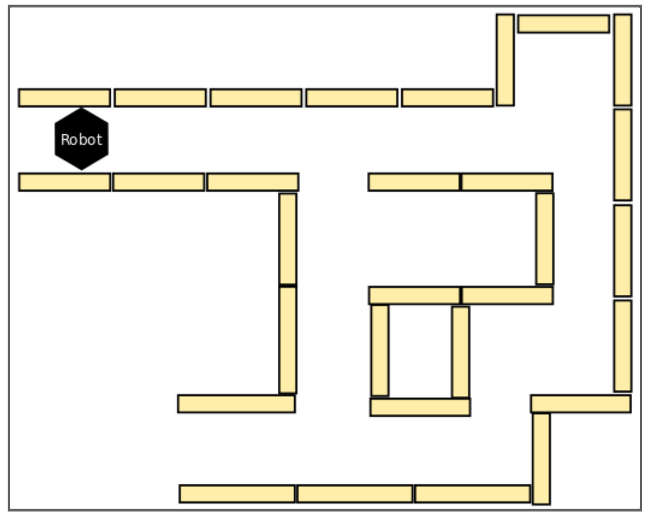
Embedded Motion Control

Best Practices in System Design for Robot Control



Nico Huebel & Herman Bruyninckx

Mechanical Engineering
KU Leuven & TU Eindhoven

Objectives of this lecture

Today's focus is on perception and software design for systems-of-systems.

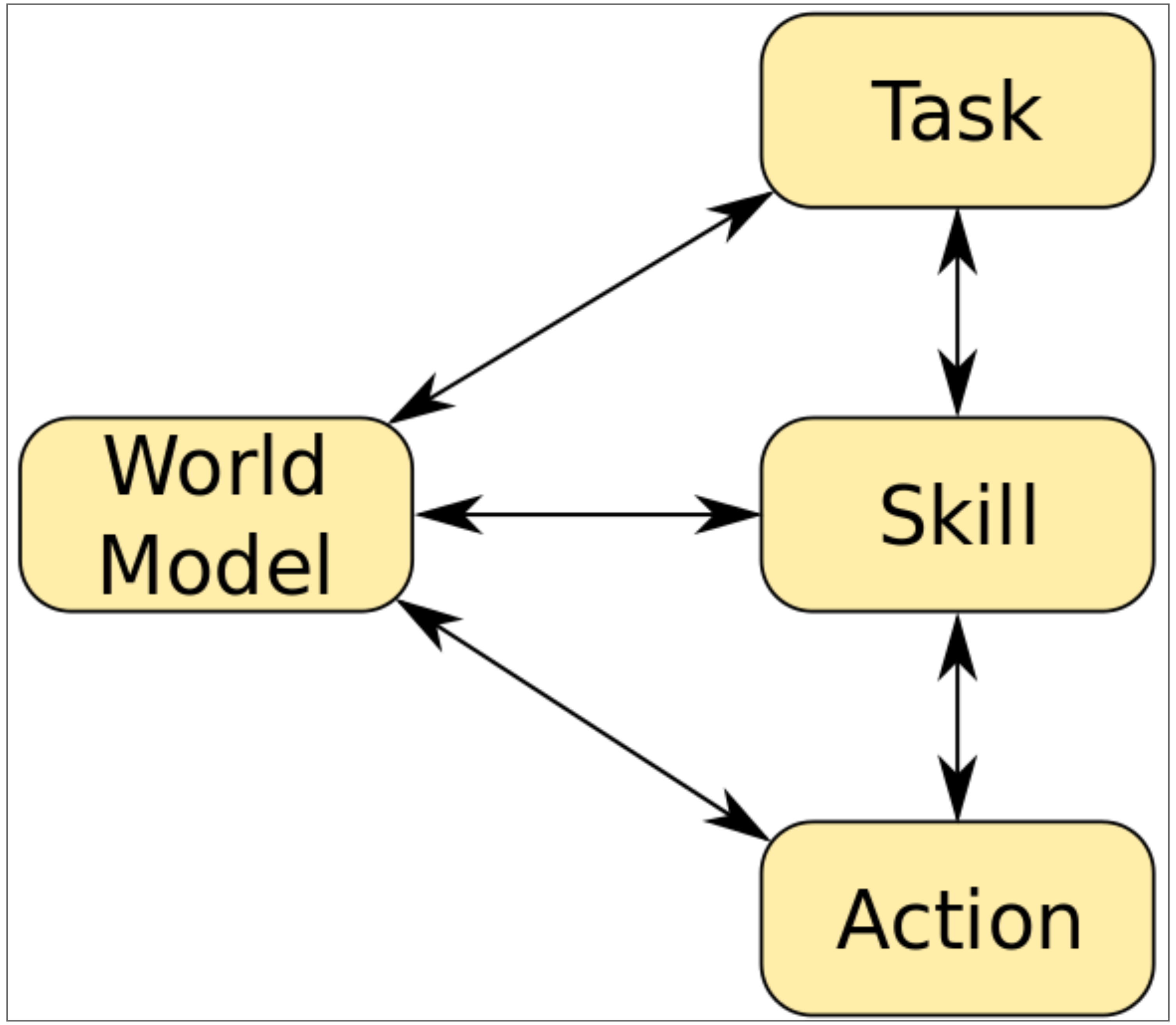
Task: navigate maze till you reach the exit

Skills: go straight, turn left , open door, sense exit, ...

Actions: Motion/Sensing actions like get laser scan, move motor, move platform, ...

⇒ representation and level of abstraction is your choice!

World Model brings perception, control, knowledge (task, objects, environment, robot) together.



A World Model for the maze challenge

Representation: type of tiles:

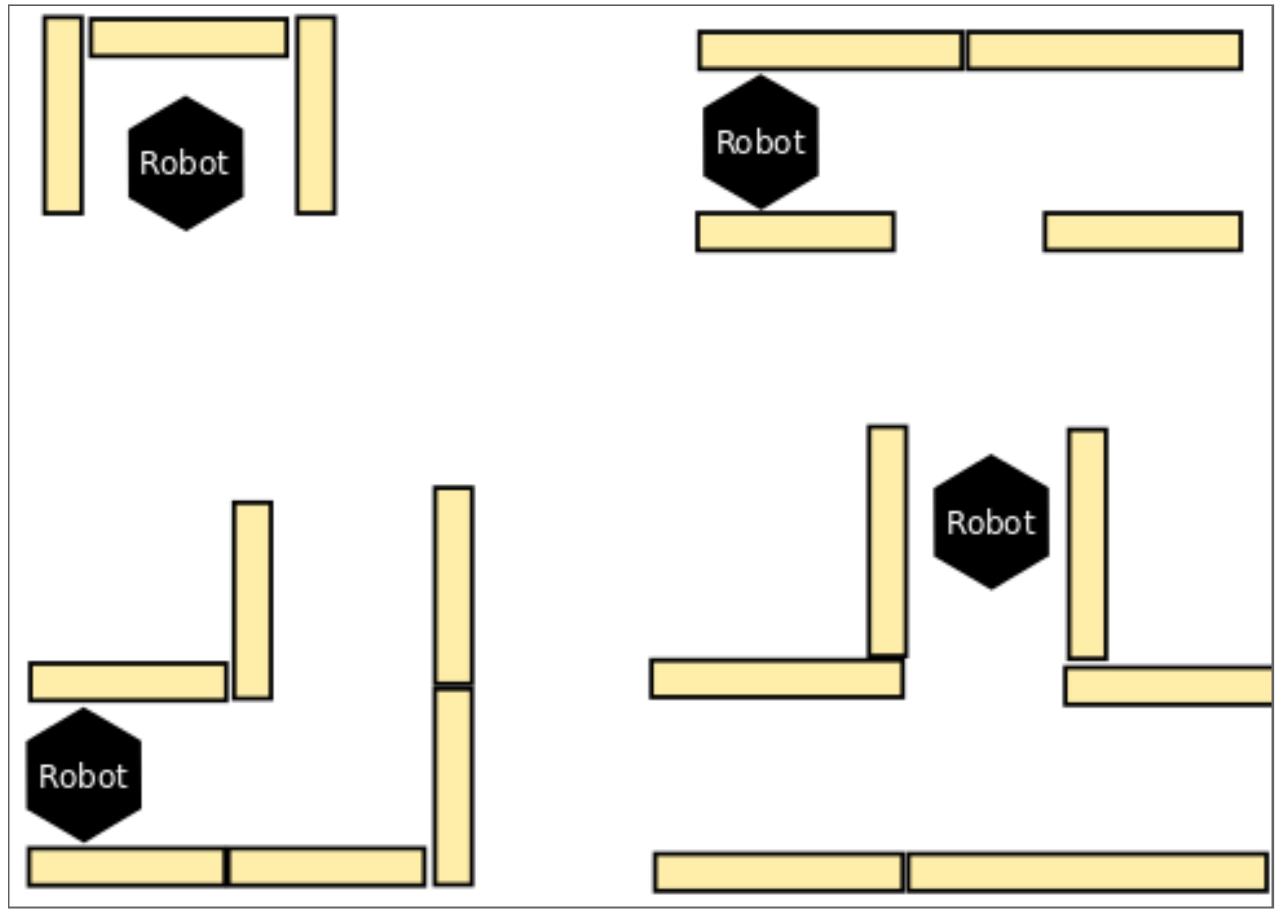
- left junction
- right junction
- door
- ...

How can these **semantic primitives** be used to improve perception?

- attach template
- specific algorithm (or configuration)
- ...

How are these semantic primitives connected to the **skills**:

- T-junction: turn left, turn right
- straight corridor: go straight, turn left (?)
- ...



How are these semantic primitives connected to **control**:

- drive straight until I am at a junction
- ...

A *semantic world model* connects and configures the layers using semantic representations (knowledge)!

Today's focus will be on the representation and estimation of the robot and environment state.

I want you to understand the *concepts* not only the equations.

I will represent *one* particular view on it based on the inclusion of semantic knowledge.

Questions so far?

Kalman Filter for Sensor Fusion (1/3)

Kalman Filter gives you an estimate of your system state *and* about the uncerteinty of this estimate.

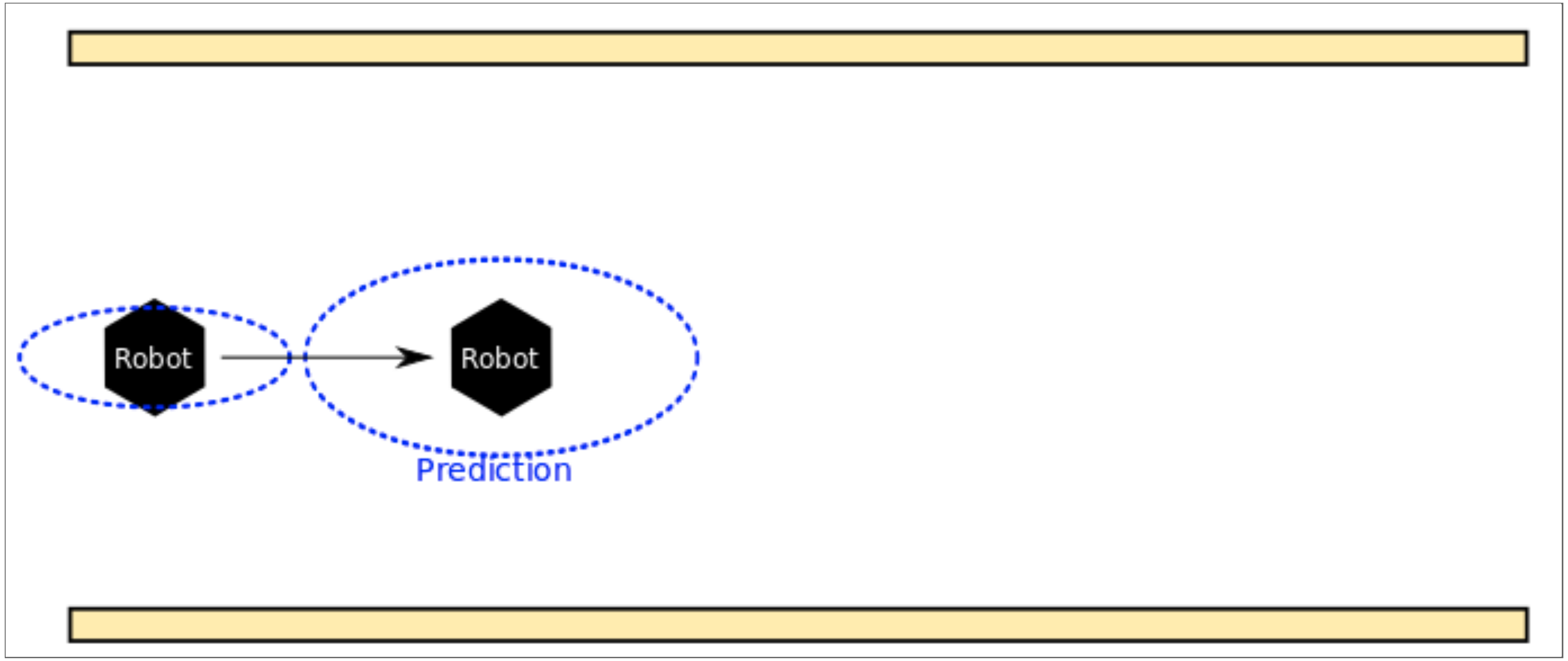
Maze challenge: Pico has encoders and laser scanner

- Encoders: fast, accumulates errors (digitalization, slip, ...)
- Laser scanner: slow, exteroceptive

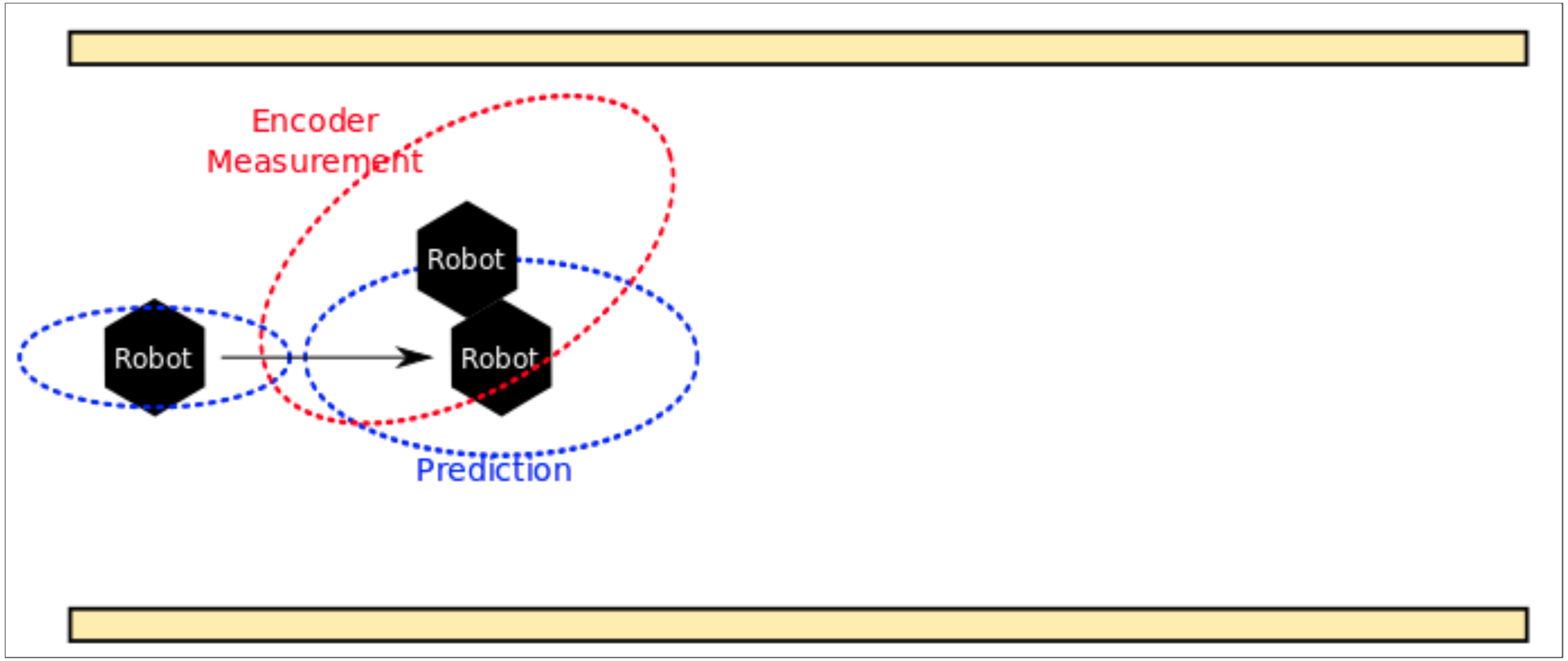
How can we use the measurements of both sensors to improve our state estimate?

Kalman Filter for Sensor Fusion (2/3)

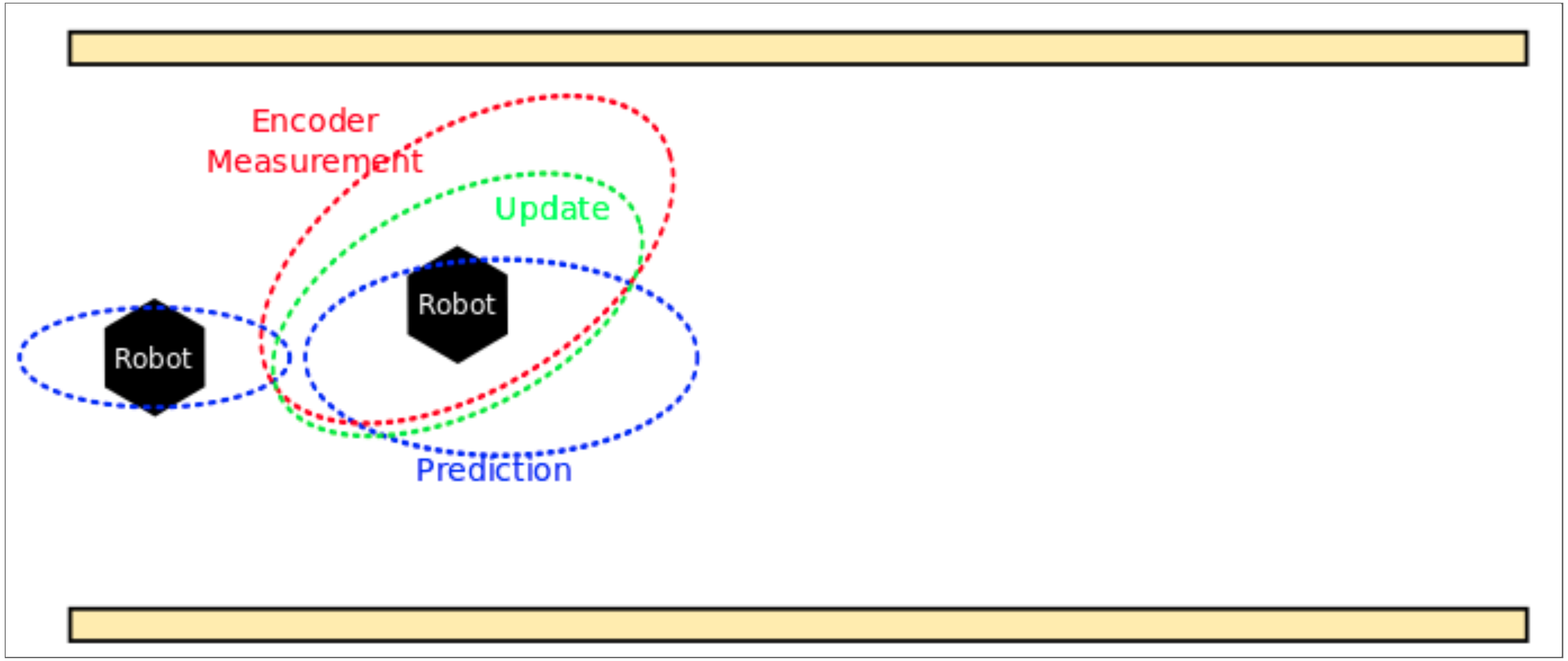
Kalman Filter for Sensor Fusion (2/3)



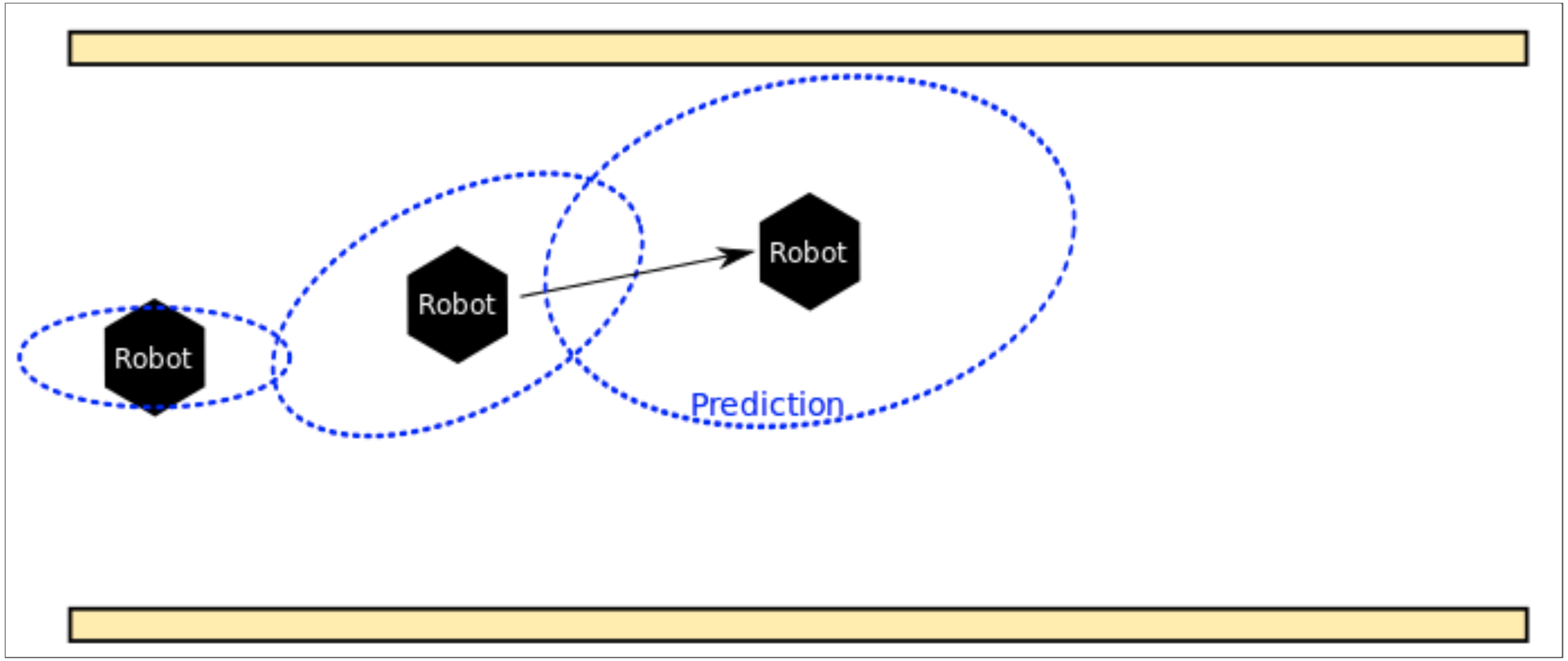
Kalman Filter for Sensor Fusion (2/3)



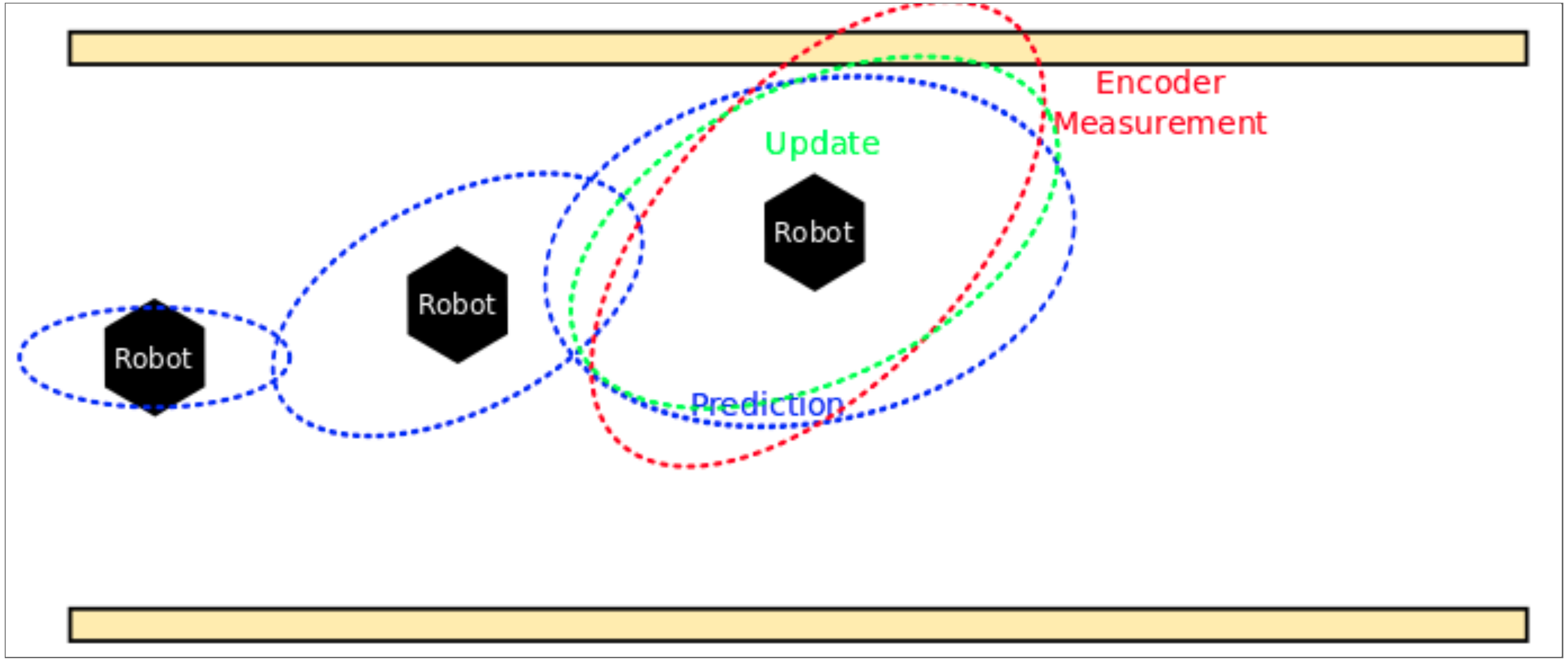
Kalman Filter for Sensor Fusion (2/3)



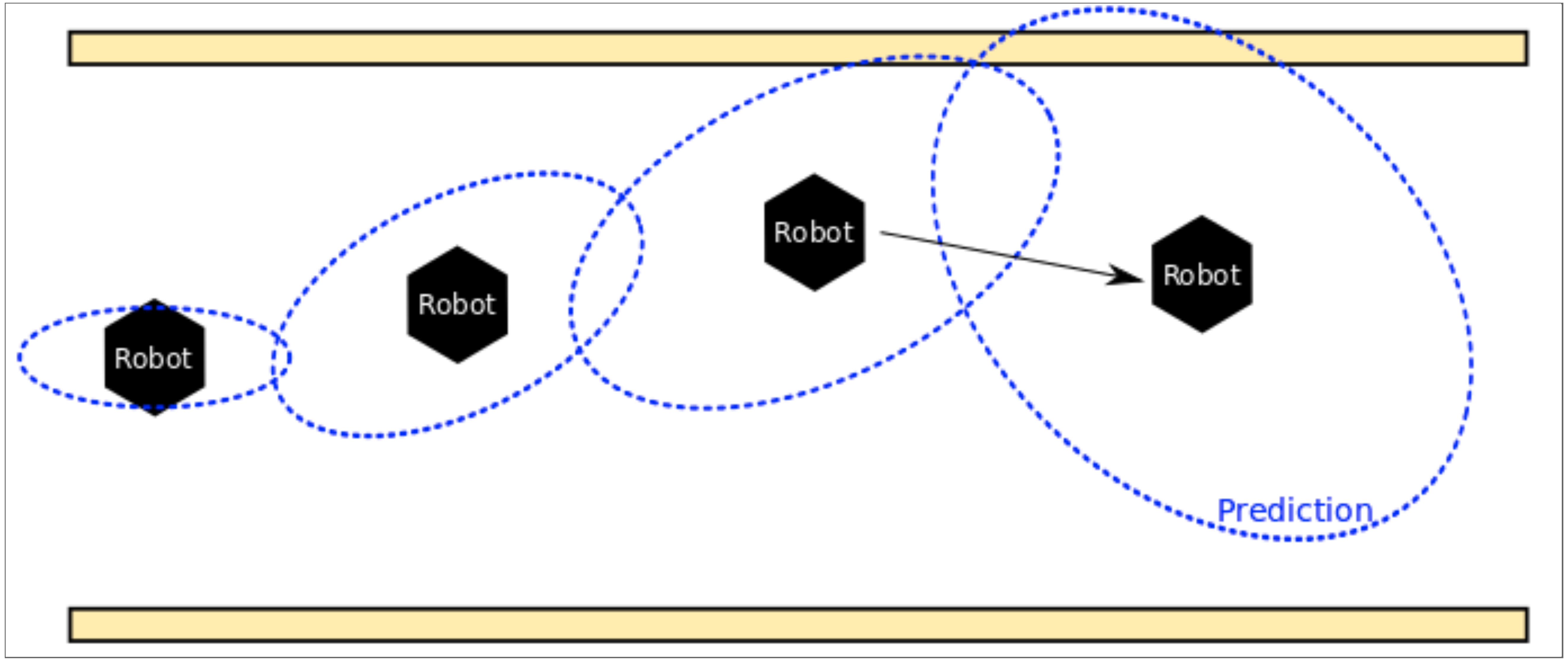
Kalman Filter for Sensor Fusion (2/3)



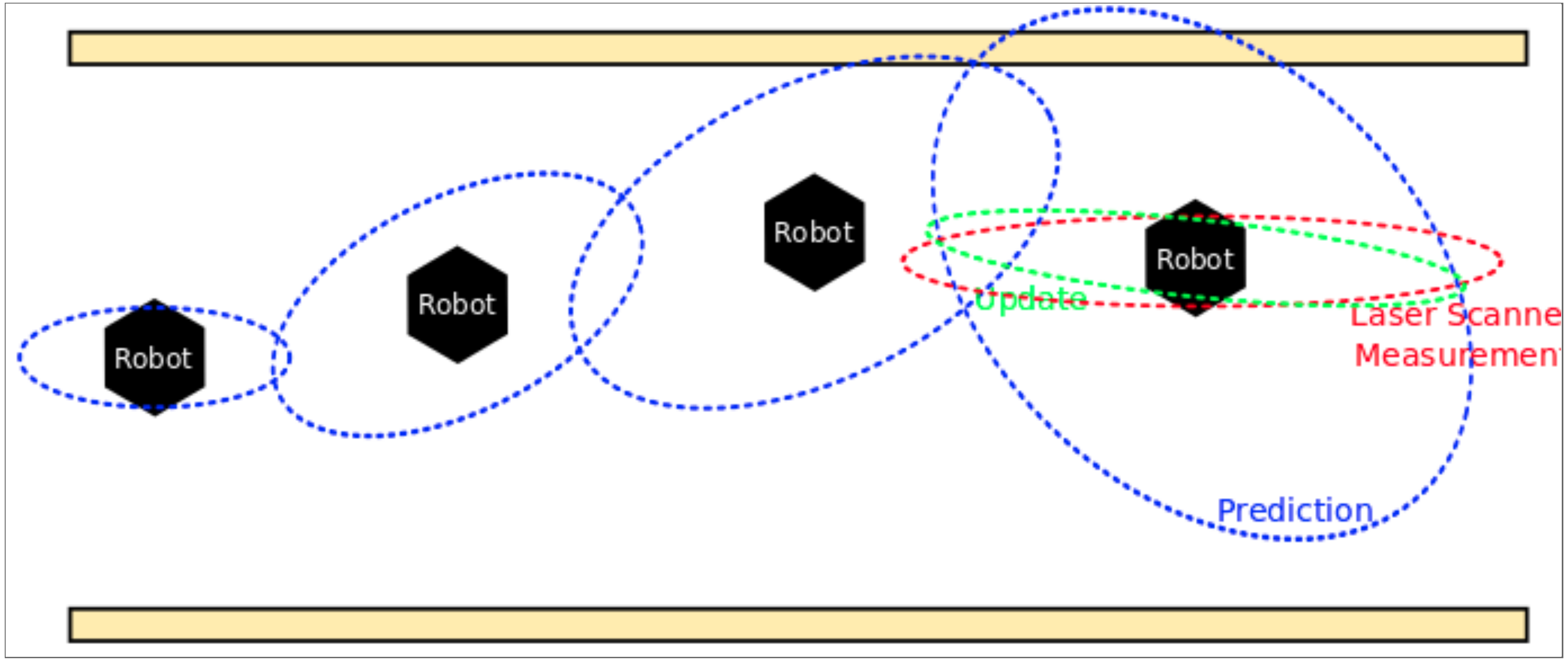
Kalman Filter for Sensor Fusion (2/3)



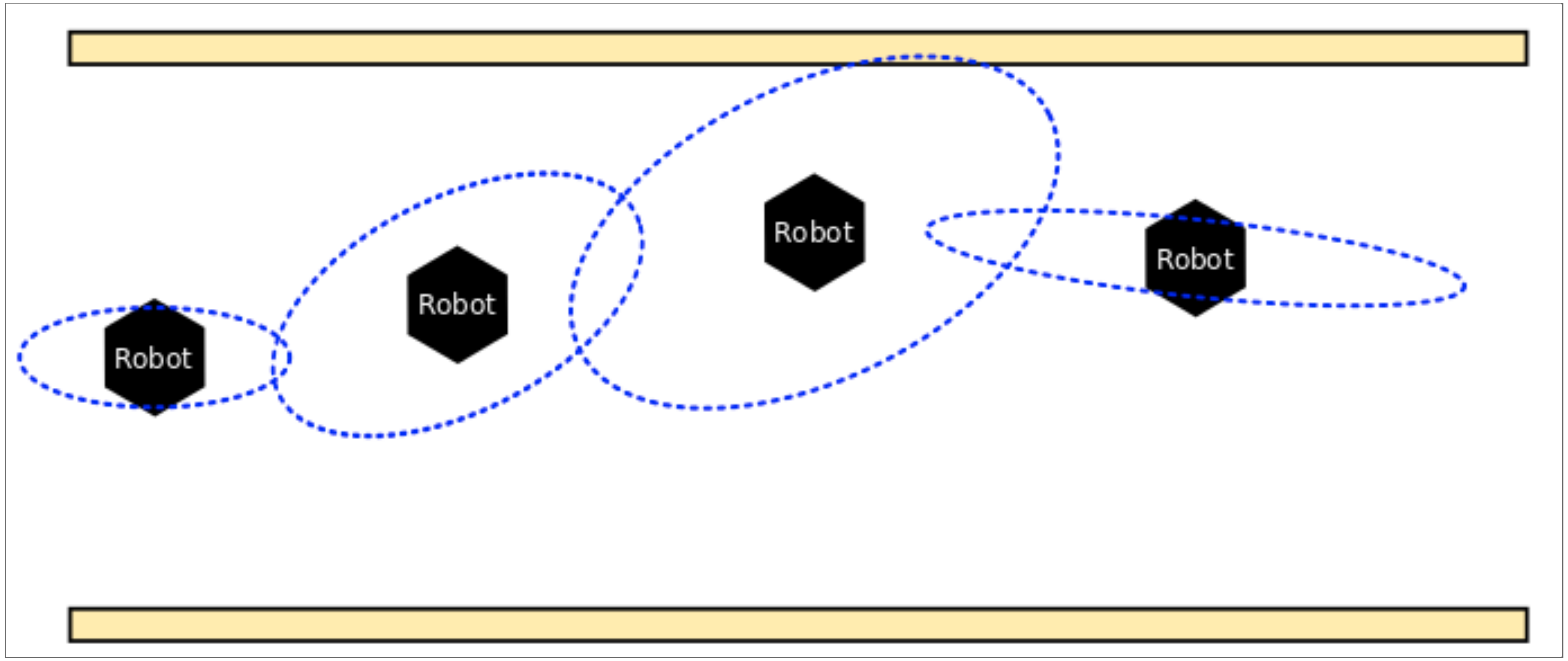
Kalman Filter for Sensor Fusion (2/3)



Kalman Filter for Sensor Fusion (2/3)



Kalman Filter for Sensor Fusion (2/3)



Kalman Filter for Sensor Fusion (3/3)

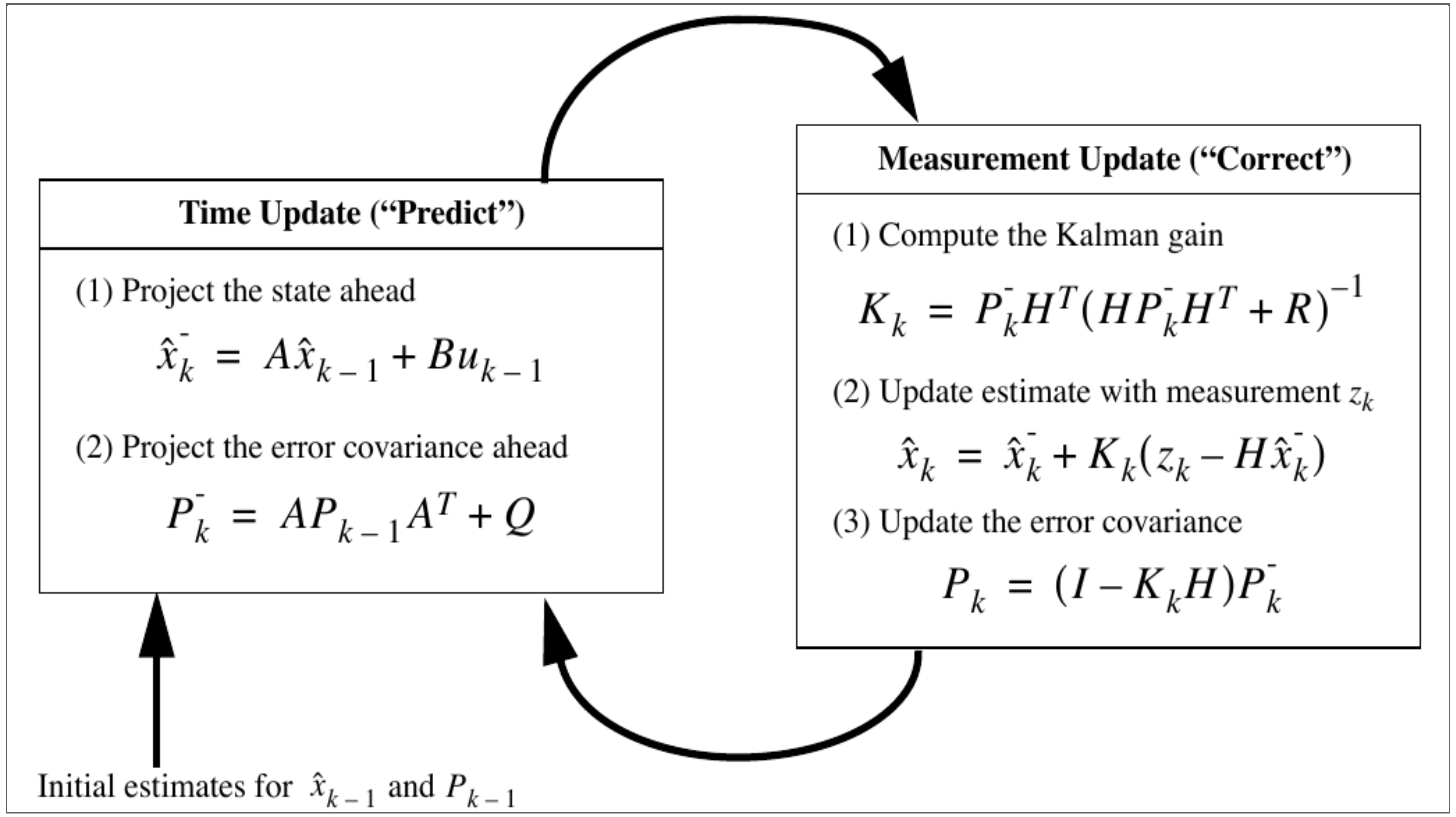
Process Model:

$$x_k = Ax_{k-1} + Bu_{k-1} + w_{k-1}$$

Measurement Model:

$$z_k = Hx_k + v_k$$

$$p(w) \sim N(0, Q),$$
$$p(v) \sim N(0, R).$$

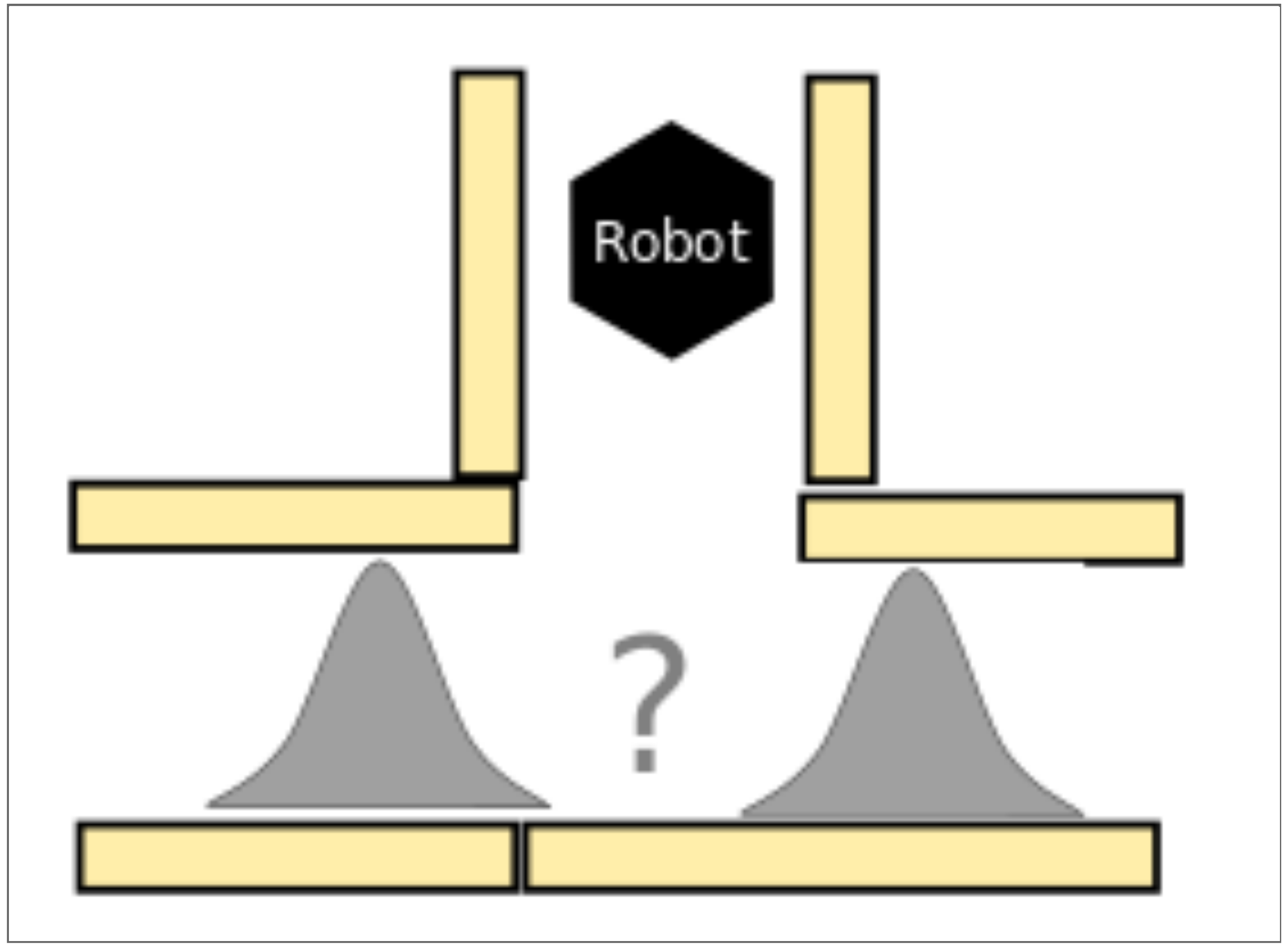


Taken from ***An Introduction to the Kalman Filter*** by Greg Welch and Gary Bishop

Shortcomings of the Kalman Filter

The Kalman Filter is widely used but has several limitations:

- Linear system
- Gaussian noise
- How to get the covariance matrices?
- Unimodal probability distribution



Where are the semantics?

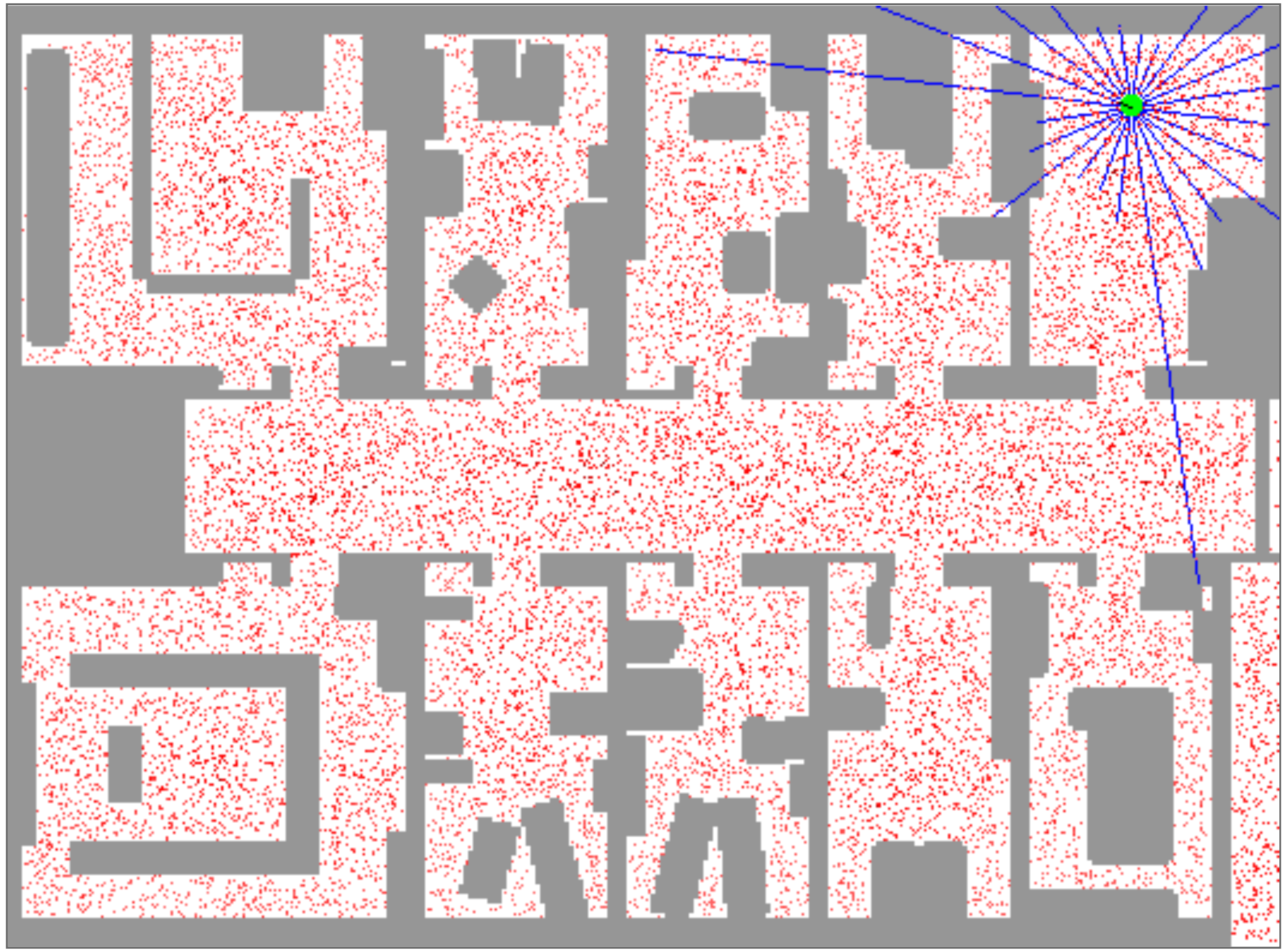
Particle Filters (1/2)

Particle Filter algorithms are also widely used and

- work with non-linear processes,
- approximate arbitrary distribution with samples,
- follow a similar (recursive) structure.

How do they work?

- Every particle represents a hypothesis for the state.
- Prediction: Put every particle through process equations.
- Update: Score particle.
- Resampling: Remove "bad" particles and add more in "good" regions.



Dieter Fox, University of Washington

Recap: A World Model for the maze challenge

Representation: type of tiles:

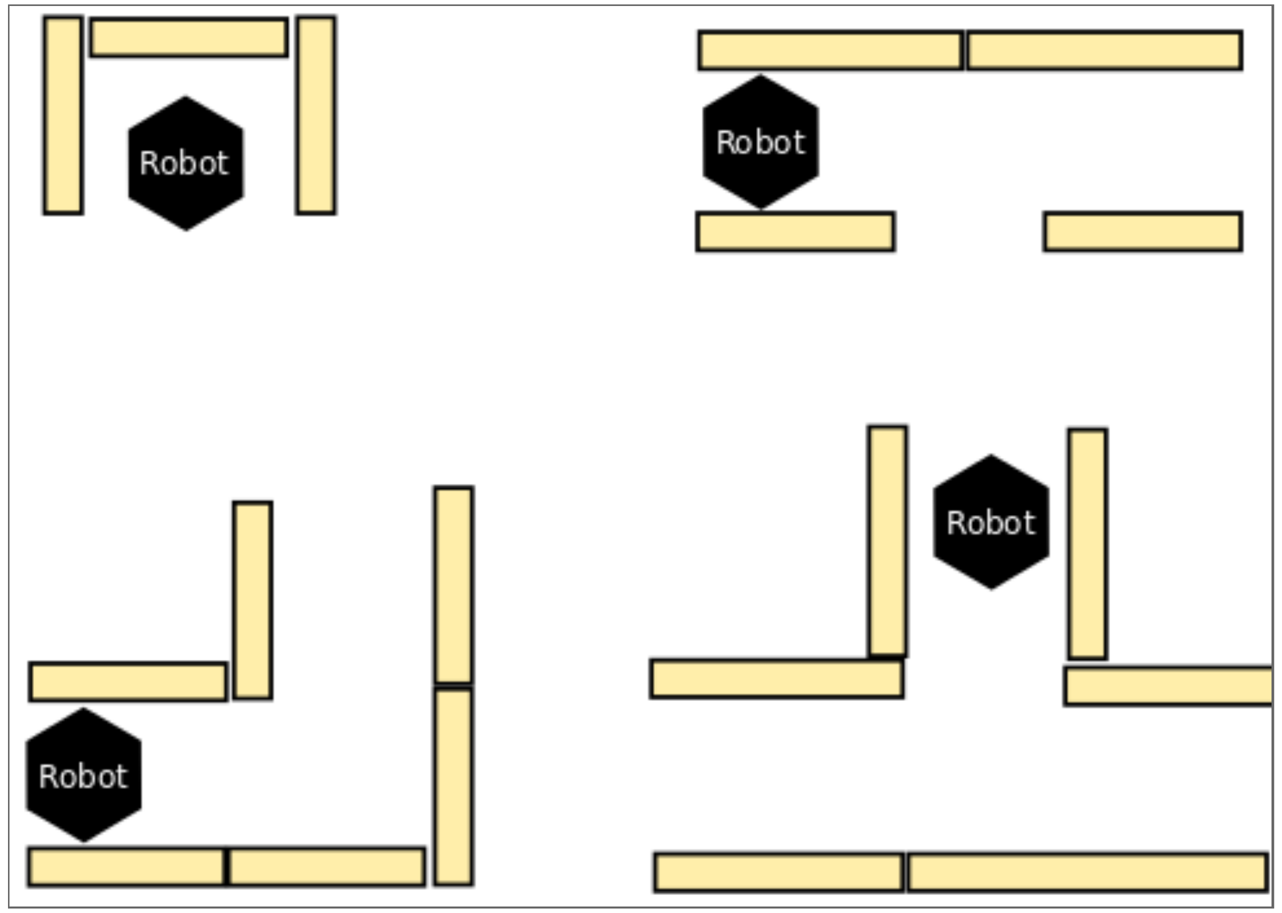
- left junction
- right junction
- door
- ...

How can these **semantic primitives** be used to improve perception?

- attach template
- specific algorithm (or configuration)
- ...

How are these semantic primitives connected to the **skills**:

- T-junction: turn left, turn right
- straight corridor: go straight, turn left (?)
- ...



How are these semantic primitives connected to **control**:

- drive straight until I am at a junction
- ...

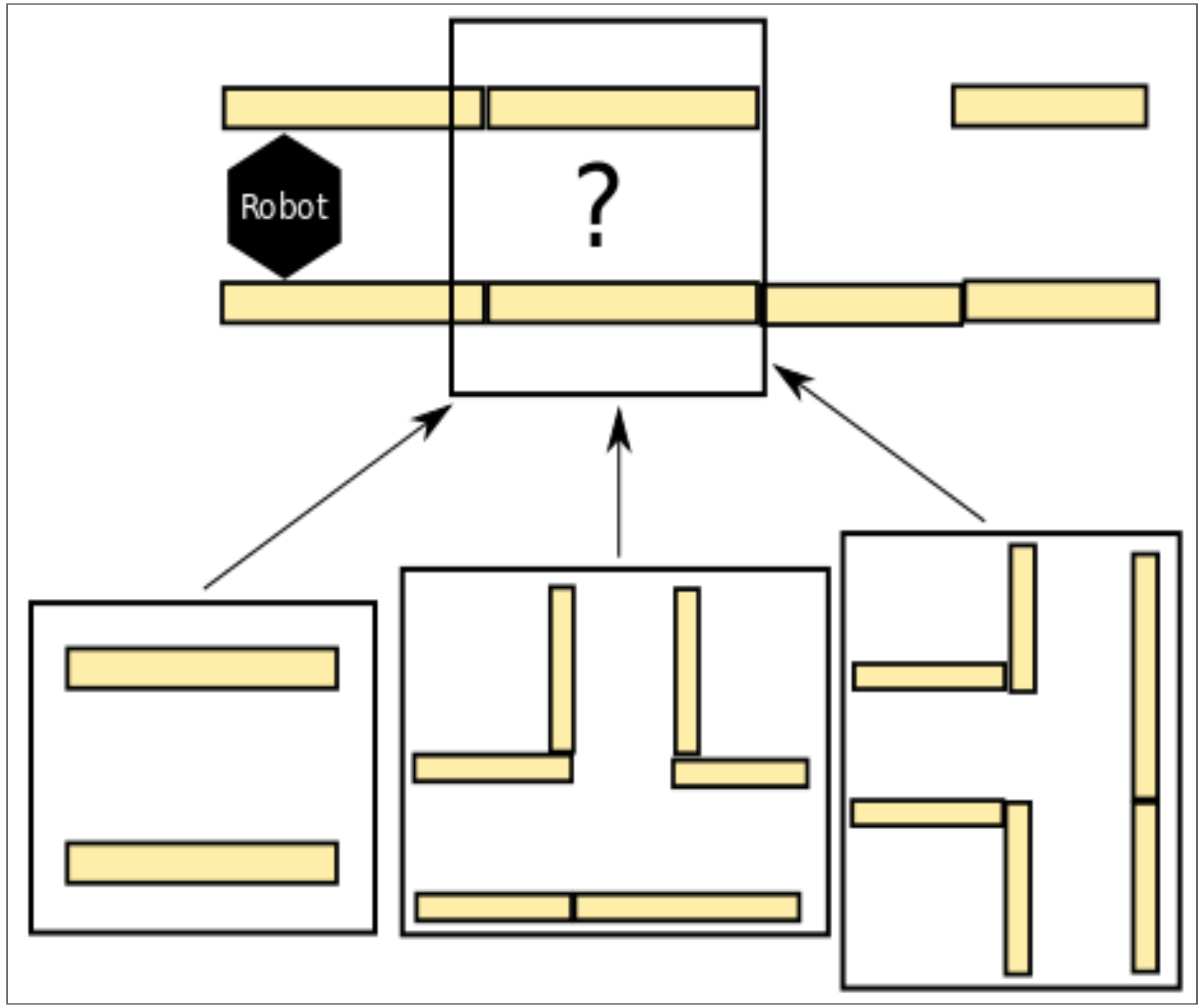
Particle Filters (2/2)

Where is the semantic knowledge?

- Particles can represent a hybrid state.
- Combine knowledge about the environment...
- ... while obtaining the parametrization of skills/actions.

What we really care about in the maze challange is what kind of semantic primitive is in front of us and how far is it away.

- Type of primitive gives us available skills.
- We only care about our relative distance to the primitive, not our absolute position in the maze!
- This distance can be used to parametrize skills and monitor their execution.
- Particle represents a hypothesis about the type of primitive (discrete) and its relative position (continous).



Conclusions

- Choosing good representations is key for a successful application.
- Always estimate what is useful for your application. That often requires some transformation/pre-processing.
- World Model brings everything together. So designing it should be a key effort.
- Kalman Filter gives you an optimal estimate *plus* uncertainty of that estimate. Use that.
- The prediction-update structure is common and very powerful.
- A particle filter can represent hybrid states and we can incorporate semantics.