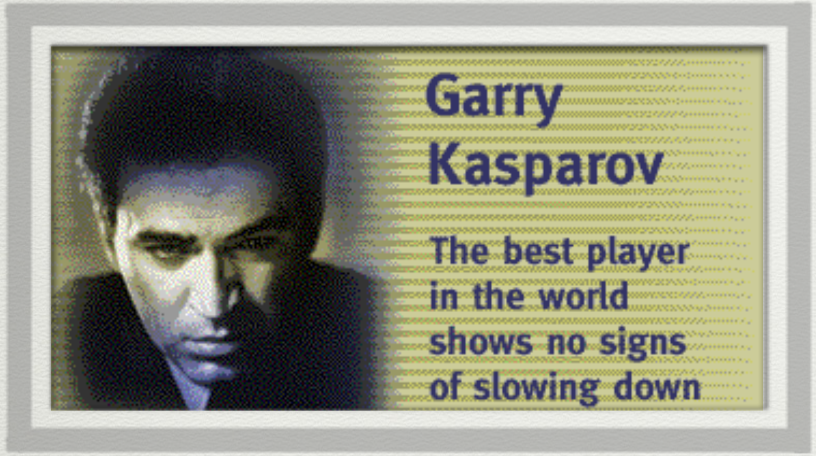
# BAYESIAN PROGRAMMING FOR ROBOTICS

Pierre Bessière

# INCOMPLETENESS AND UNCERTAINTY

# Who is the cleverest ?

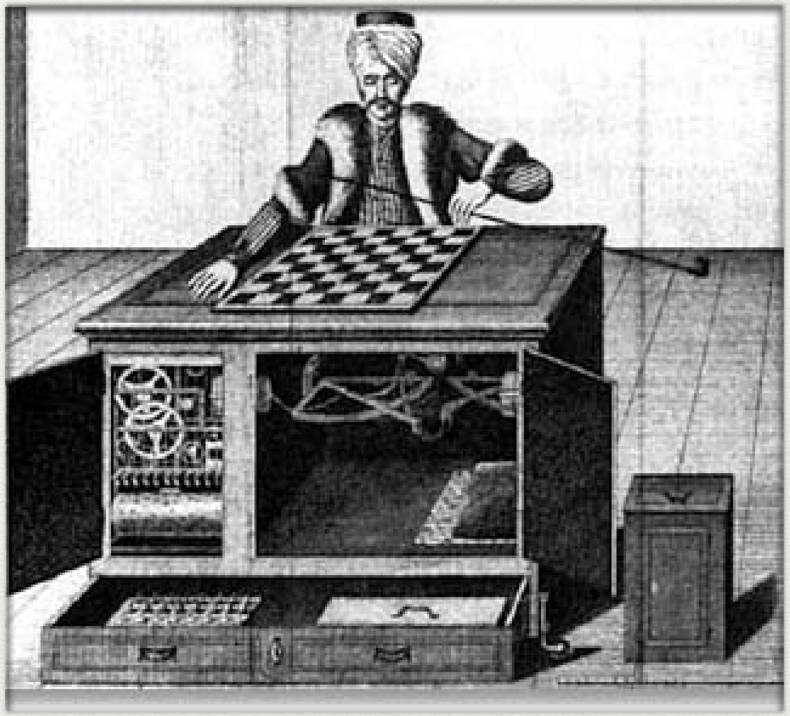
## Playing chess




# Who is the cleverest ?

# Who is the cleverest ?

## Playing with chess



Baron Wolfgang von Kempelen (1769)

# Content

- Probability as an extension of logic
- Bayesian programming and ProBT
- Robotics applications
  - How to learn reactive behaviors
  - Bayesian filter
  - Hierarchies of models
  - Coherence variables
- Toward dedicated hardware

# PROBABILITY AS AN EXTENSION OF LOGIC

# Probability for incomplete problem?

**Incompleteness**

Bayesian Learning

Preliminary Knowledge
+
Experimental Data
=
Probabilistic Representation

**Uncertainty**

Bayesian Inference

$$P(a) + P(\neg a) = 1$$

$$P(a \wedge b) = P(a)P(b \mid a) = P(b)P(a \mid b)$$

**Decision**

# Probability as an extension of logic

**Probabilistic inference and learning theory**, considered as a **model of reasoning**, is an alternative to logic to explain and understand perception, inference, decision, learning and action.

Uncertainty is not in things but in our head: uncertainty is a lack of knowledge
**Jacob Bernouilli**, Ars Conjectandi (Bernouili, 1713)

La théorie des probabilités n'est rien d'autre que le sens commun fait calcul.

**Marquis Pierre-Simon de Laplace**, Théorie analytique des probailités (Laplace 1812)

The actual science of logic is conversant at present only with things either certain, impossible, or entirely doubtful, none of which (fortunately) we have to reason on. Therefore the true logic for this world is the calculus of Probabilities, which takes account of the magnitude of the probability which is, or ought to be, in a reasonable man's mind .

**James Clerk Maxwell (1850)**

Randomness is just the measure of our ignorance.
To undertake any probability calculation, and even for this calculation to have a meaning, we have to admit, as a starting point, an hypothesis or a convention, that always comprises a certain amount of arbitrariness. In the choice of this convention, we can be guided only by the principle of sufficient reason. From this point of view, every sciences would just be unconscious applications of the calculus of probabilities. Condemning this calculus would be condemning the whole science.

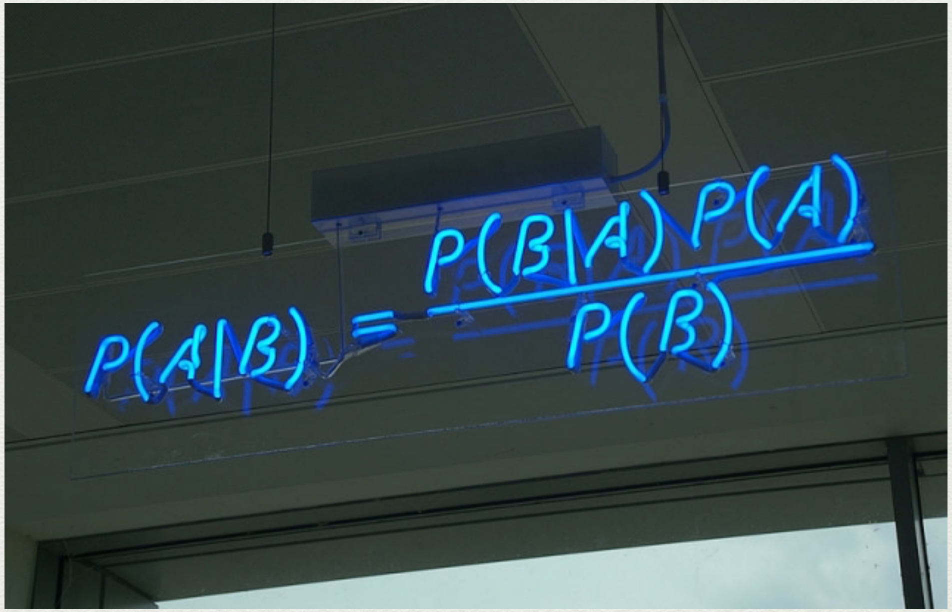
**Henri Poincaré**, La science et l'hypothèse (Poincaré, 1902)

By inference we mean simply: deductive reasoning whenever enough information is at hand to permit it; inductive or probabilistic reasoning when - as is almost invariably the case in real problems - all the necessary information is not available. Thus the topic of « Probability as Logic » is the optimal processing of **uncertain** and **incomplete** knowledge .

**E.T. Jaynes**, Probability theory theory: the logic of science (Jaynes, 2003)

# What rules ?

$$\sum_A P(A) = 1$$




$$P(A,B) = P(A)P(B \mid A) = P(B)P(A \mid B)$$

# Syllogisms ?

a: X is divisible by 9

b: X is divisible by 3

Modus Ponens

$$a \wedge [a \Rightarrow b] \mapsto b \qquad P(b \mid a) = 1$$

Modus Tollens

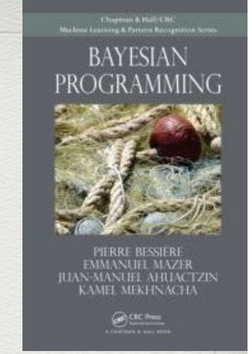
$$\neg b \wedge [a \Rightarrow b] \mapsto \neg a \qquad P(\neg a \mid \neg b) = 1$$

$$b \wedge [a \Rightarrow b] \mapsto ??? \qquad P(a \mid b) = \frac{P(a)P(b \mid a)}{P(b)} \geq P(a)$$

$$\neg a \wedge [a \Rightarrow b] \mapsto ??? \qquad P(b \mid \neg a) \leq P(b)$$

# BAYESIAN PROGRAMMING

# Bayesian Programming




**Bayesian Program**

**Description**

**Specification**

- Variables

$$S^0, \ldots\ldots, S^t, O^0, \ldots\ldots, O^t$$

- Decomposition

$$\mathbf{P}(S^0 \wedge \ldots\ldots \wedge S^t \wedge O^0 \wedge \ldots\ldots \wedge O^t) = \mathbf{P}(S^0) \times \mathbf{P}(O^0 \mid S^0) \times \prod_{i=2}^{t} [\mathbf{P}(S^i \mid S^{i-1}) \times \mathbf{P}(O^i \mid S^i)]$$

- Parametric Forms

$$\mathbf{P}(S^0) \equiv \mathbf{G}(S^0, \mu, \sigma)$$
$$\mathbf{P}(S^i \mid S^{i-1}) \equiv \mathbf{G}(S^i, A \bullet S^{i-1}, Q)$$
$$\mathbf{P}(O^i \mid S^i) \equiv \mathbf{G}(O^i, H \bullet S^i, R)$$

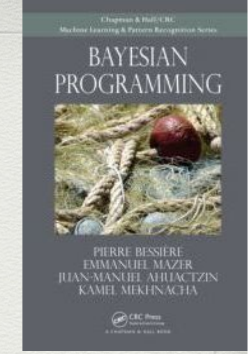
**Identification**

- Learning from instances

**Question**

$$\mathbf{P}(S^t \mid O^0 \wedge \ldots\ldots \wedge O^t)$$

# Bayesian Programming

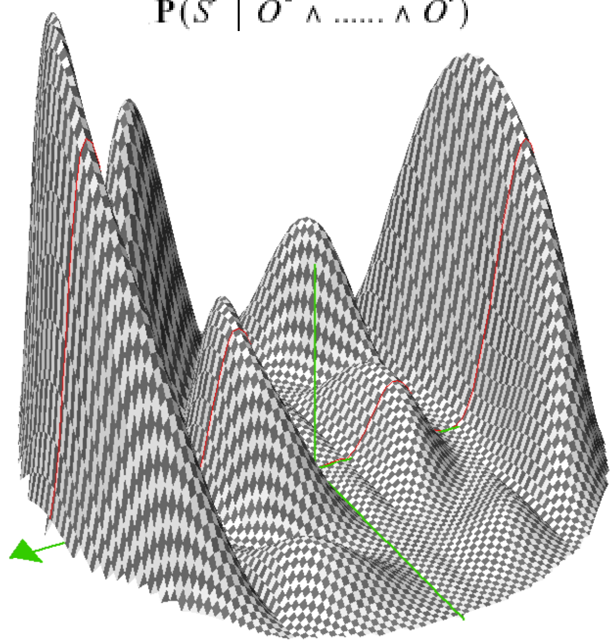



BAYESIAN PROGRAM

- DESCRIPTION
  - SPECIFICATION
    - VARIABLES: $S^0, \ldots$
    - DECOMPOSITION: $P(S^0 \ldots$ $\ldots S^i \mid S^{i-1}) \times P(O^i \mid S^i)]$
    - PARAMETRIC FORMS
  - IDENTIFICATION
    - LEARNING
- QUESTION: $P(S^t \mid O^0 \wedge \ldots\ldots \wedge O^t)$

```
main ()
{
//Variables
plFloat read_time;
plInteg
plFloat
p
p
p

//Par
//Const
plProbVa
p

/
p
p

//Const
plProbVa
plProb

 //Const
plKernel
plValues
t_and_i
Pt_id.p
t_and_i

Pt_id.p

//Lea
// P(ti
plJointD

                                    time)
plCndKernel Pid_t;
jd.ask(Pid_t,id,time);

//Read a time from the key board
cout<<"P(id,time)= "<<Pid_t<<"\n"
cout<<"Time? : ";
cin>>read_time;

//Getting P(id | time = read_time)
plKernel Pid_readTime;
```

$P(S^t \mid O^0 \wedge \ldots\ldots \wedge O^t)$



PROBAYES.COM

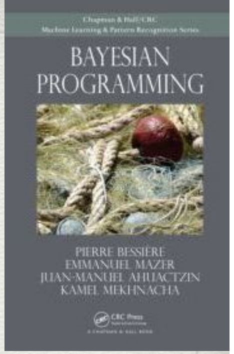
BAYESIAN-PROGRAMMING.ORG

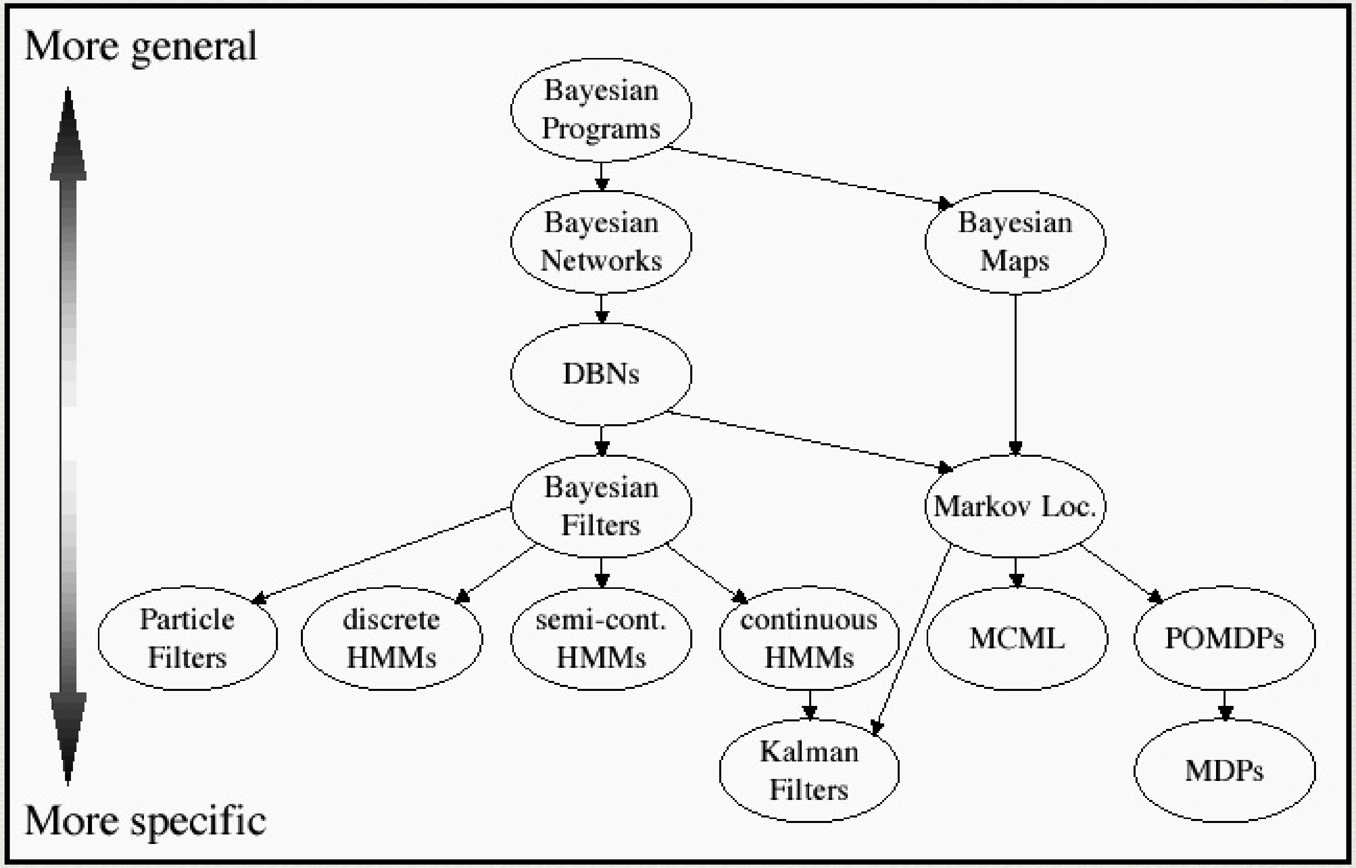
BAMBI



CNRS UPMC SORBONNE UNIVERSITÉS ISIR INSTITUT DES SYSTÈMES INTELLIGENTS ET DE ROBOTIQUE

# Bayesian Programming



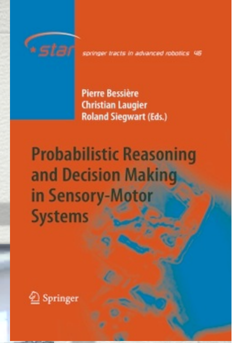

Equivalent to probabilistic factor graphs

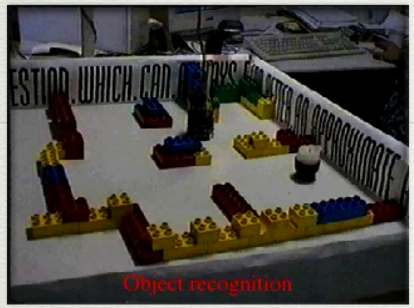
# Why is it called „programming“?

- Iteration: Filters
- Calling subroutines: Hierarchies
- Conditional statements
- Variable equality, assignation and matching: Coherence variables

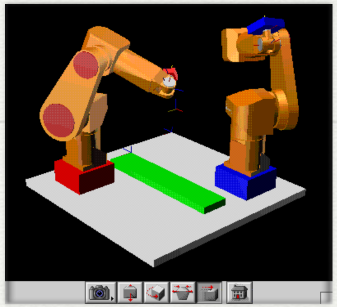
# APPLICATIONS IN ROBOTICS

# Robotics

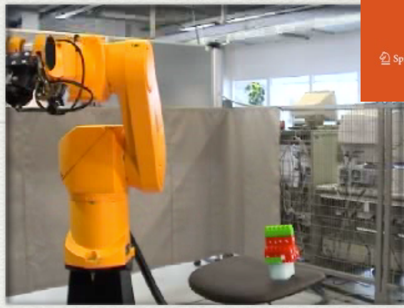



PhD Olivier Lebeltel (1999)
*Advanced Robotics* (2004)

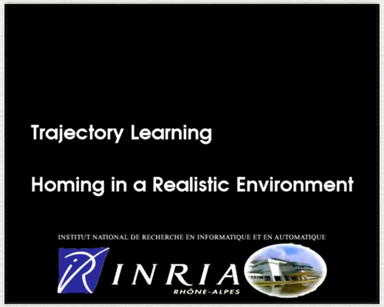


PhD Kamel Mekhnacha (1999)
*Advanced Robotics* (2001)

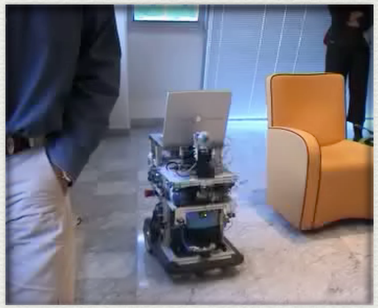


PhD Ruben Garcia (2003)

PhD Christophe Coué (2003)
*IJRR* (2006)




PhD Cédric Pradalier (2004)
*Robotics and Autonomous systems* (2005)



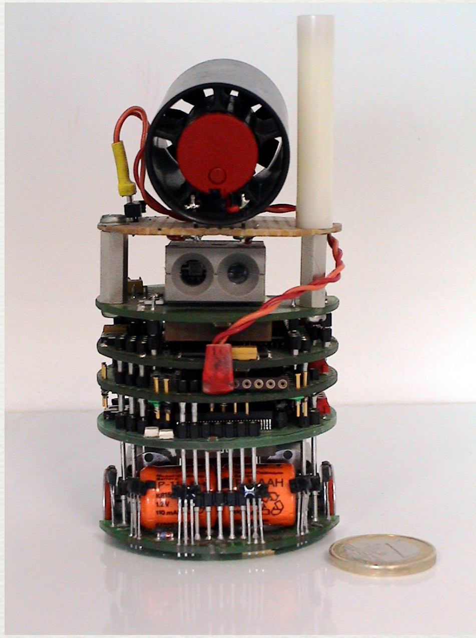
PhD Carla Koike (2005)


PhD Ronan Le Hy (2007)
*Robotics and Autonomous systems* (2004)

PhD Gabriel Synnaeve (2012)
*IEEE Tr. Computational Intelligence and AI in Games* (2015)

# LEARNING REACTIVE BEHAVIORS

# Learning reactive behaviors

- Avoiding Obstacle
- Contour Following
- Piano mover
- Phototaxy
- etc.

Lebeltel, O., Bessière, P., Diard, J. & Mazer, E. (2004) Bayesian Robot Programming; *Autonomous Robots*, Vol. 16, p. 49-79

Lebeltel, O. (1999) *Programmation bayésienne des robots*; Thèse INPG

# Pushing objects

Program

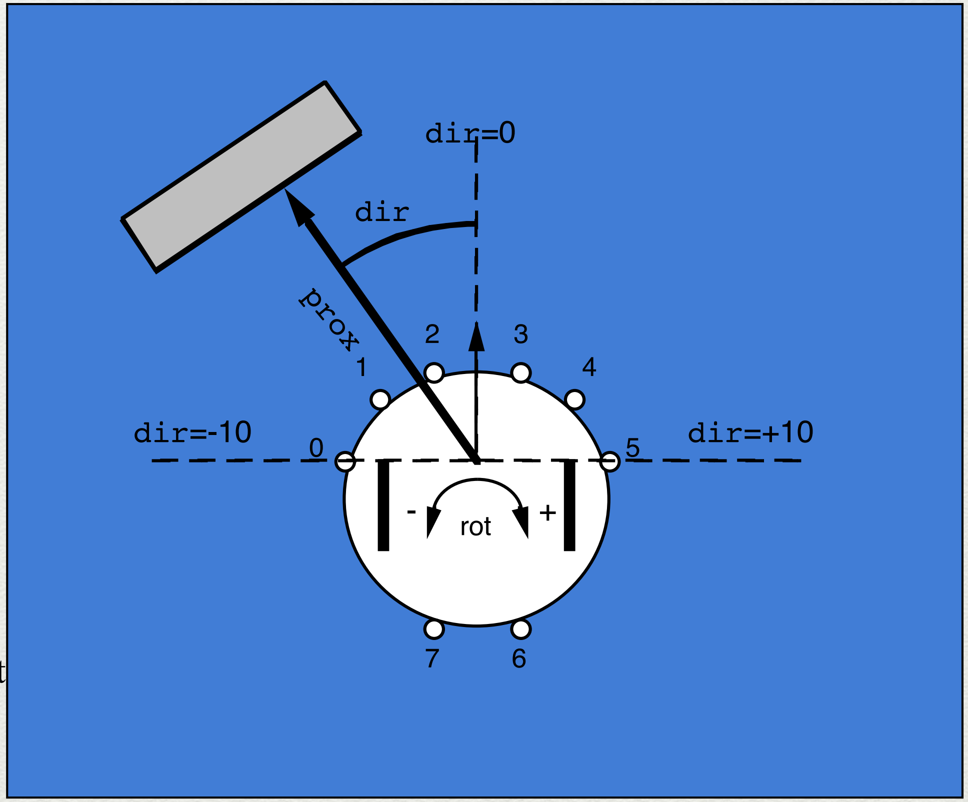
Description

Question

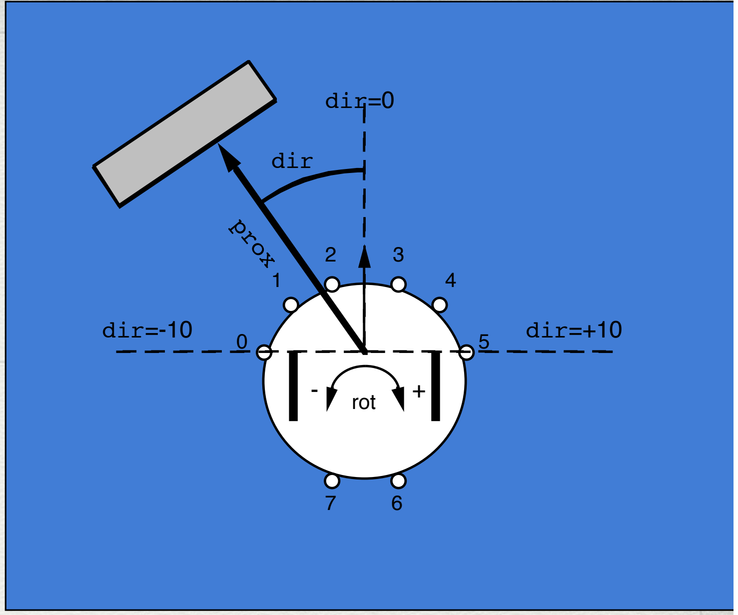
*Specification*

- Variables

Utilizat



# Pushing objects




Program

Description

*Specification*

- Variables

$$Dir, Prox, Vrot$$

- Decomposition

$$P(Dir, Prox, Vrot) = P(Dir)P(Prox)P(Vrot \mid Dir, Prox)$$

- Parametrical Forms

$$P(Dir, Prox) = Uniform$$

$$P(Vrot \mid Dir, Prox) = Gaussians$$

*Identification*

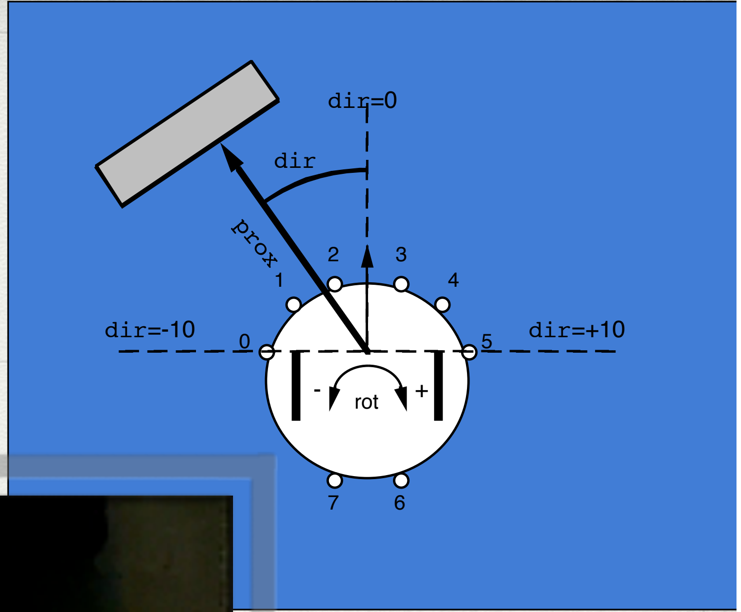
- Joystick Remote Control ➔ Experimental Data δ1

Question

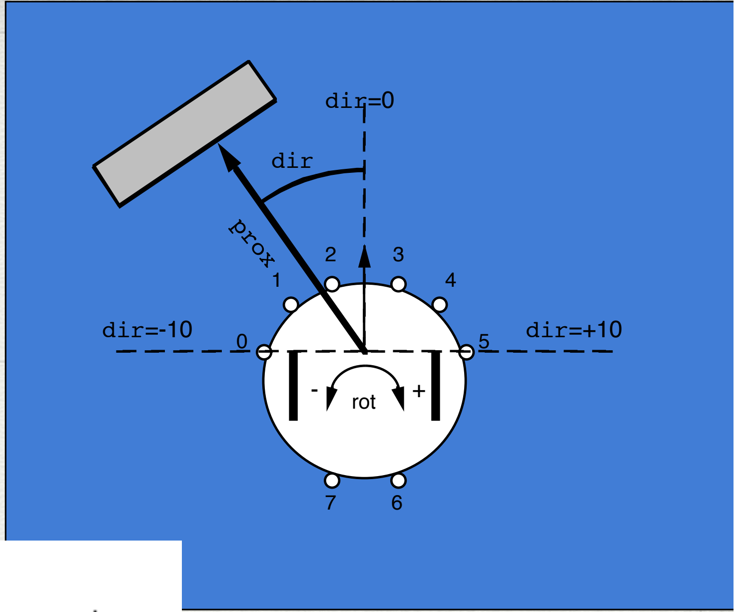
Utilization

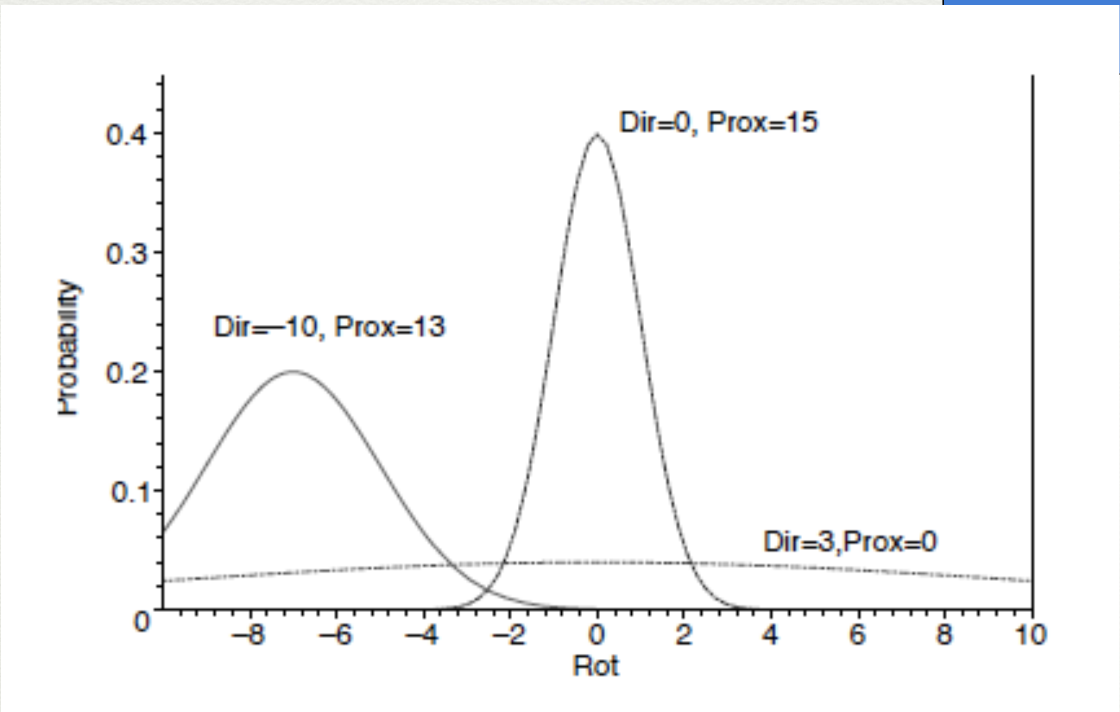
$$P(Vrot \mid [Dir = d][Prox = p])$$

# Pushing objects

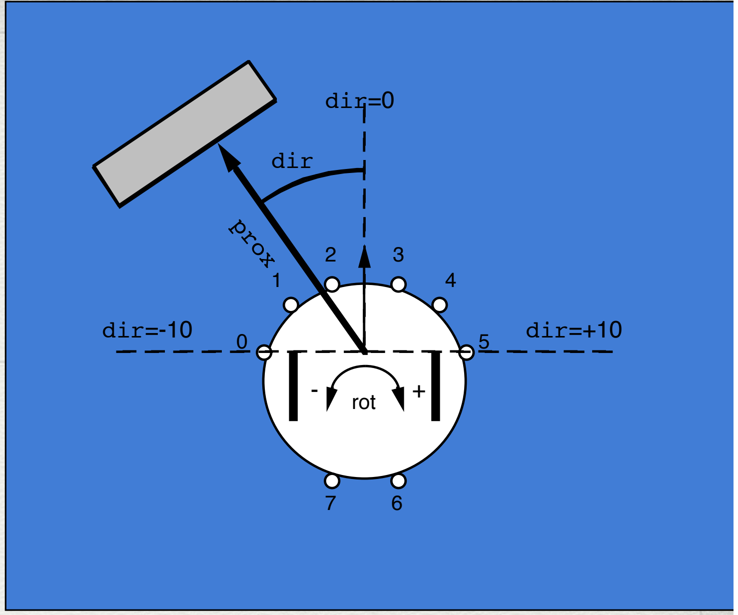



# Pushing objects



# Contour following




Program

Description

*Specification*

- Variables

$$Dir, Prox, Vrot$$

- Decomposition

$$P(Dir, Prox, Vrot) = P(Dir)P(Prox)P(Vrot \mid Dir, Prox)$$

- Parametrical Forms

$$P(Dir, Prox) = Uniform$$

$$P(Vrot \mid Dir, Prox) = Gaussians$$

*Identification*

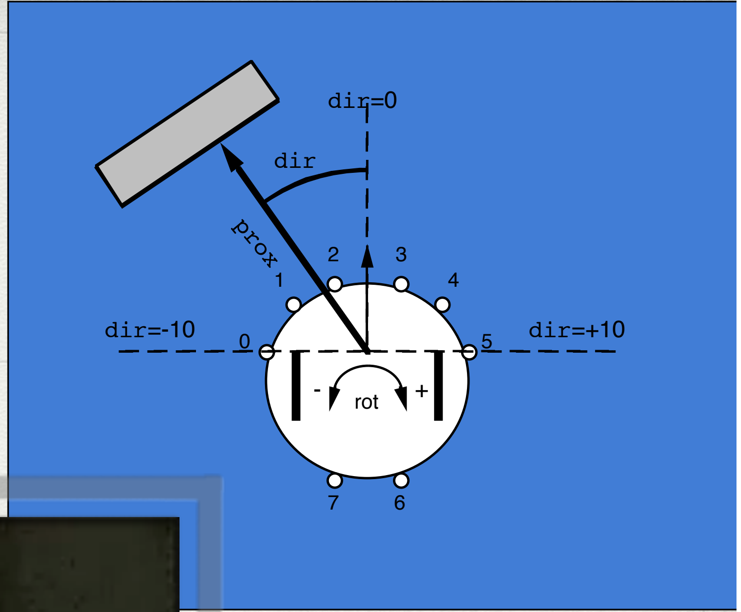
- Joystick Remote Control ➔ Experimental Data $\delta 2$

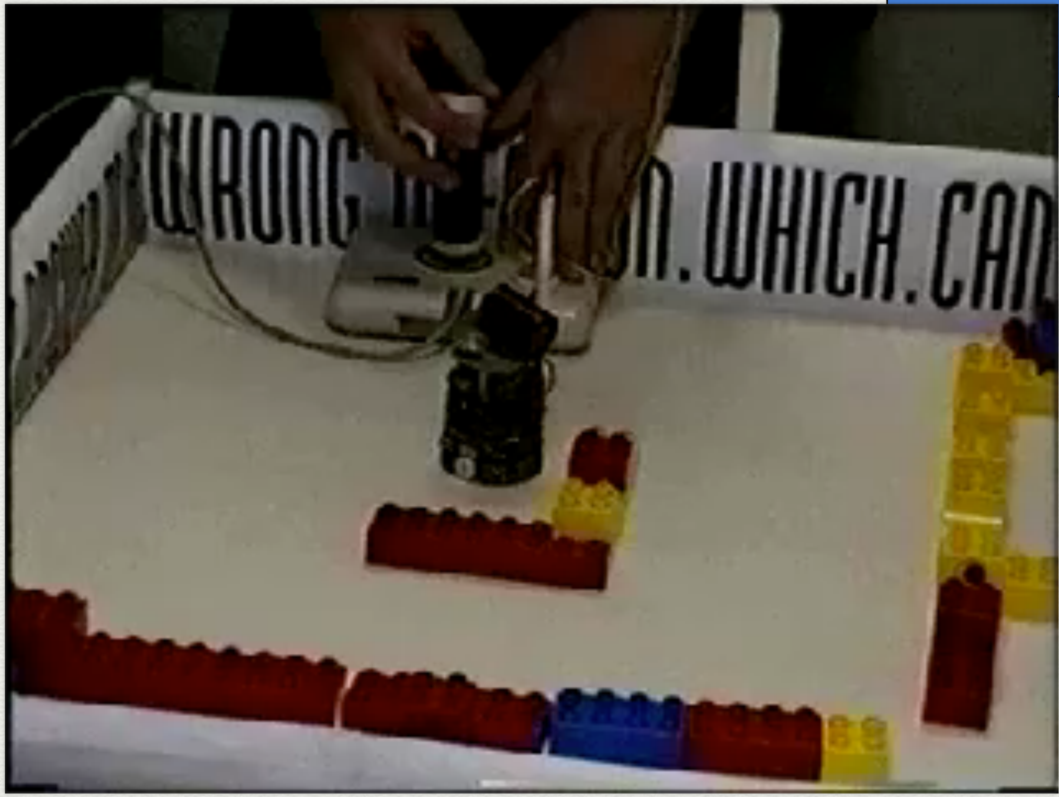
Question

Utilization

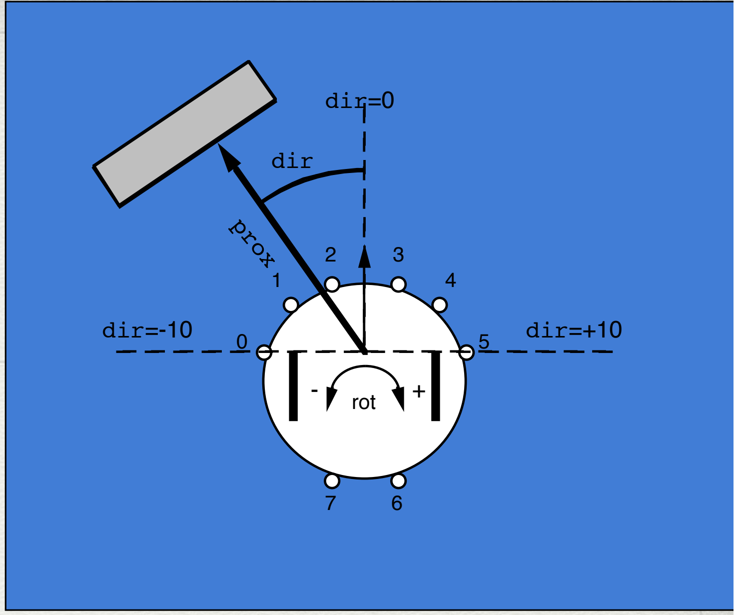
$$P(Vrot \mid [Dir = d][Prox = p])$$

# Contour following



# Pushing vs. following



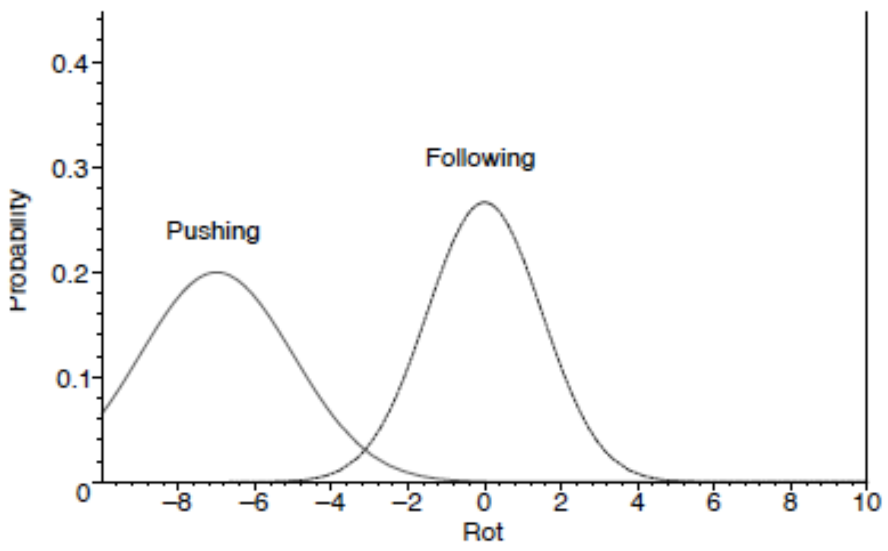




Figure 4.4: $\mathbf{P}(Rot \mid [Dir = -10] \wedge [Prox = 13])$

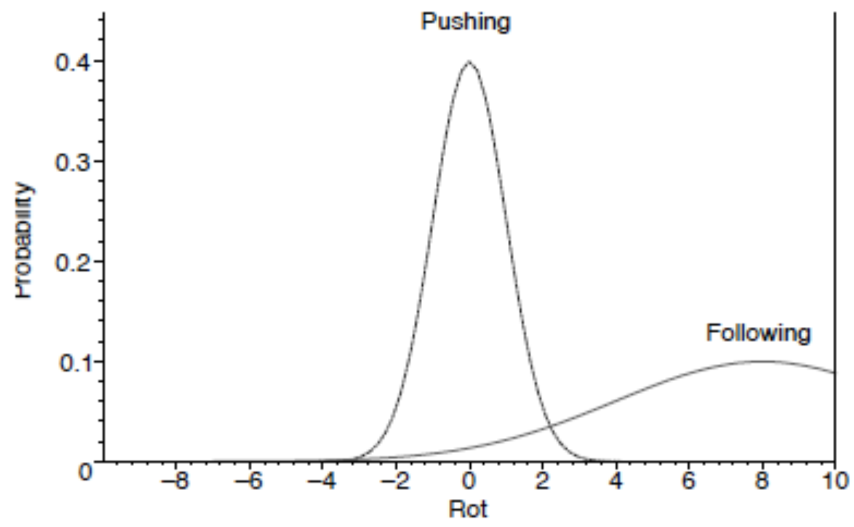



Figure 4.5: $\mathbf{P}(Rot \mid [Dir = 0] \wedge [Prox = 15])$

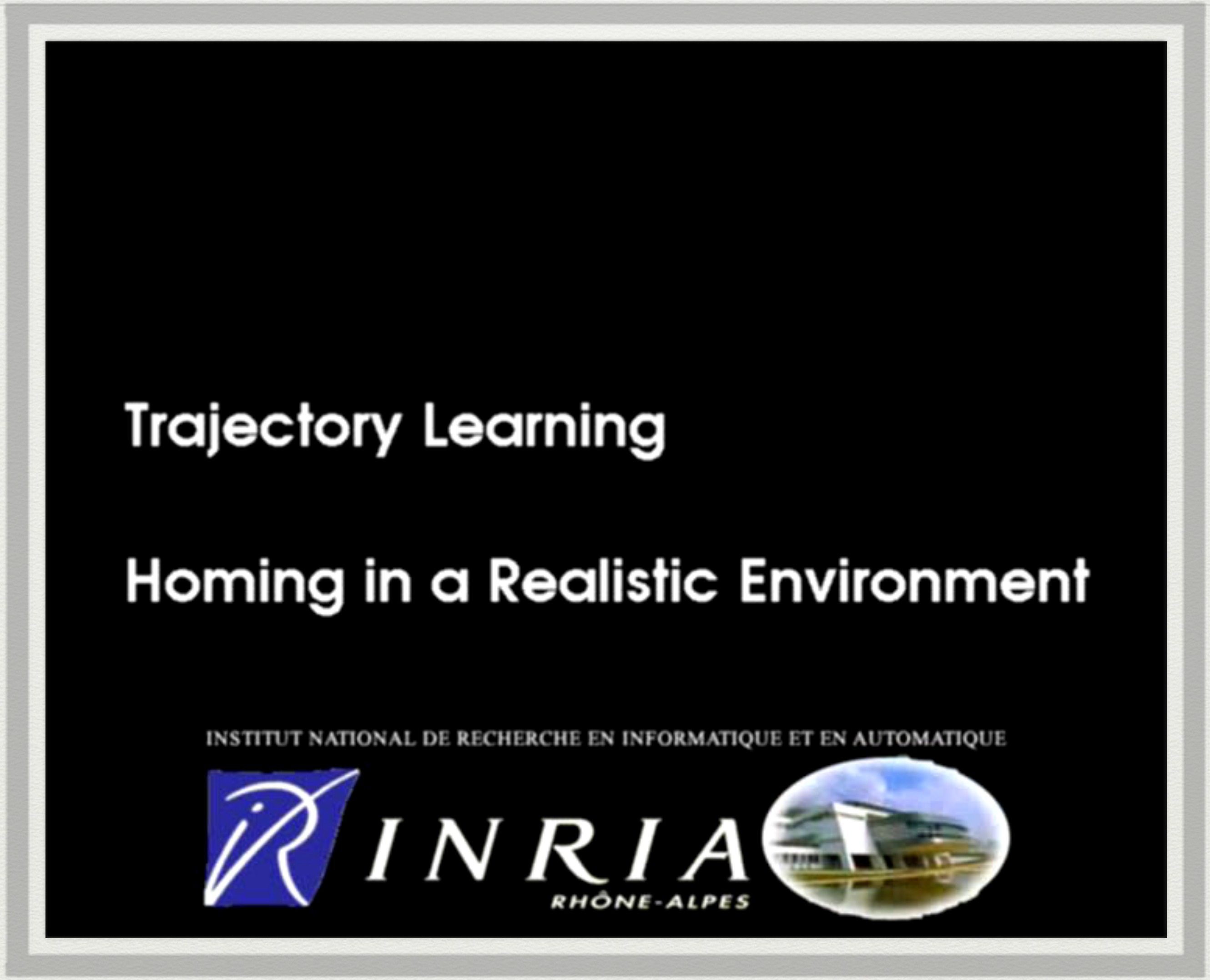
# Mobile Robot Navigation

PhD C. Pradalier (2004)

[Pradalier04]
[Pradalier05]

# Mobile Robot Navigation

PhD C. Pradalier (2004)



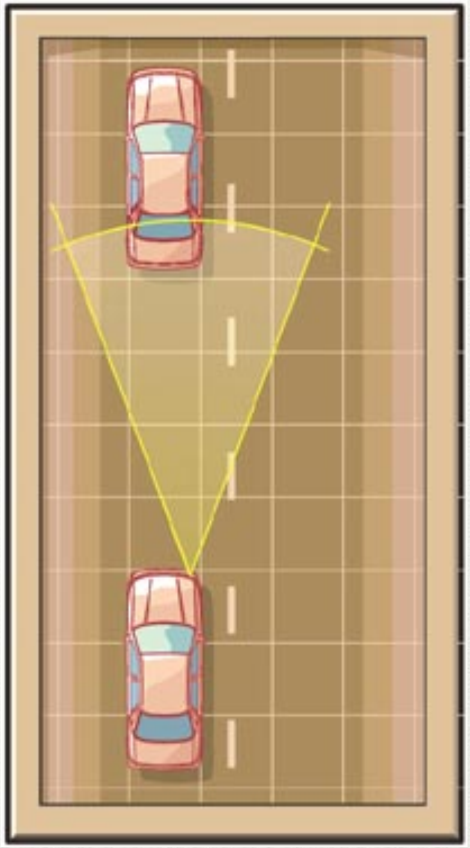

[Pradalier04]
[Pradalier05]

# BAYESIAN FILTERS

# Specification

$$
Pr\begin{cases}
Ds\begin{cases}
Sp\begin{cases}
Va:\begin{cases} S^t, \forall t \in [0,\ldots,T] : S^t \in D_S \\ O^t, \forall t \in [1,\ldots,T] : O^t \in D_O \end{cases} \\
Dc:\begin{cases} P\left(S^0 \wedge O^1, \ldots S^t \wedge O^t \ldots S^T \wedge O^T\right) = \\ P\left(S^0\right) \prod\limits_{t\in[1\ldots T]} \left(P\left(S^t \mid S^{t-1}\right) P\left(O^t \mid S^t\right)\right) \end{cases} \\
Fo:\begin{cases} P\left(S^0\right) = \text{Initial condition} \\ P\left(S^t \mid S^{t-1}\right) = \text{Transition Model} \\ P\left(O^t \mid S^t\right) = \text{Sensor Model} \end{cases}
\end{cases} \\
Id:
\end{cases} \\
Qu: P\left(S^T \mid o^1 \ldots o^T\right)
\end{cases}
$$

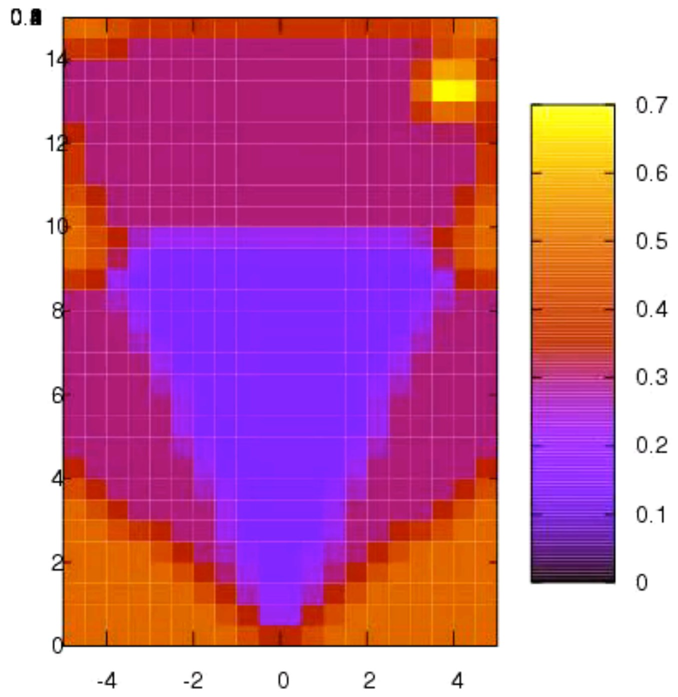
# Bayesian Occupency Filter (BOF) for ADAS

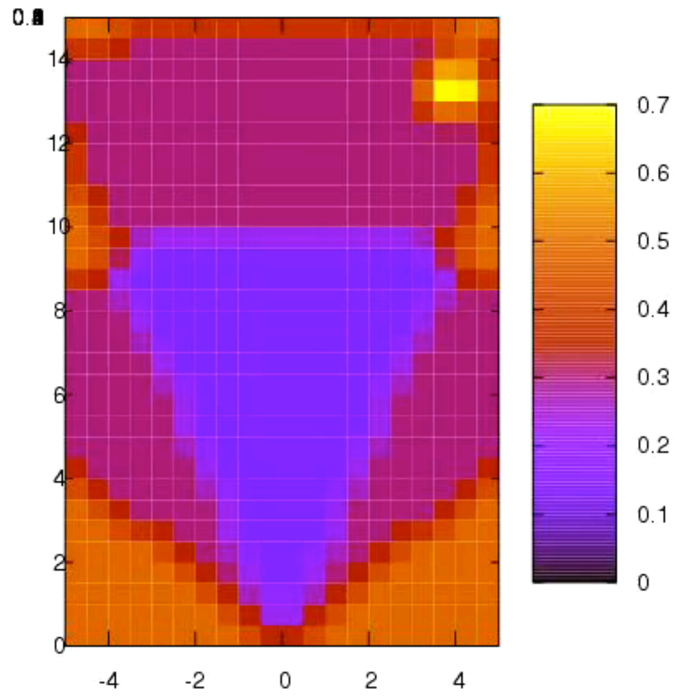


Coué, C., Pradalier, C., Laugier, C., Fraichard, T. & Bessière, P. (2006) Bayesian Programming multi-target tracking: an automotive application; *IJRR (International Journal of Robotic Research);* Vol. 25, N° 1, pp. 19-30

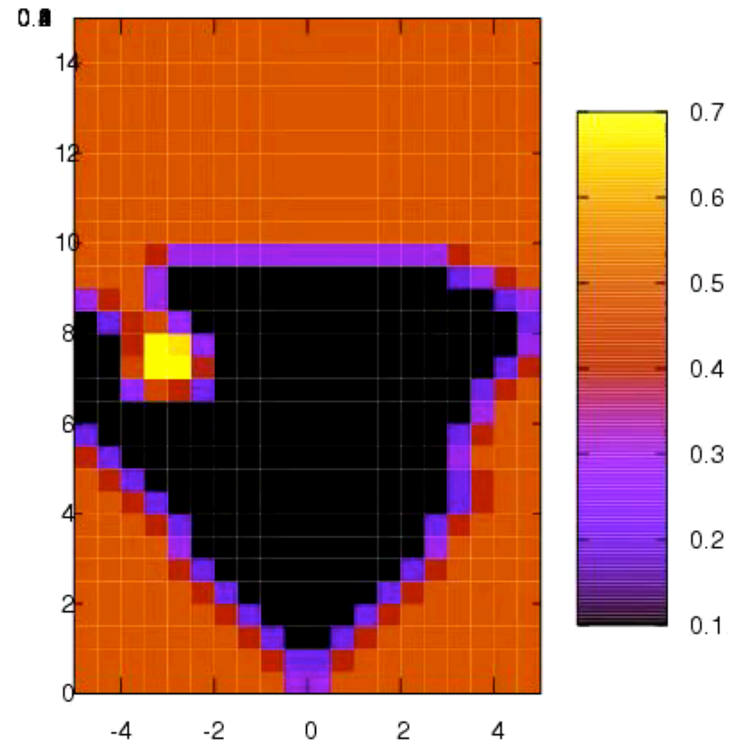
Coué, C. (2003) *Fusion d'information capteur pour l'aide à la conduite automobile*; PhDF thesis, INPG

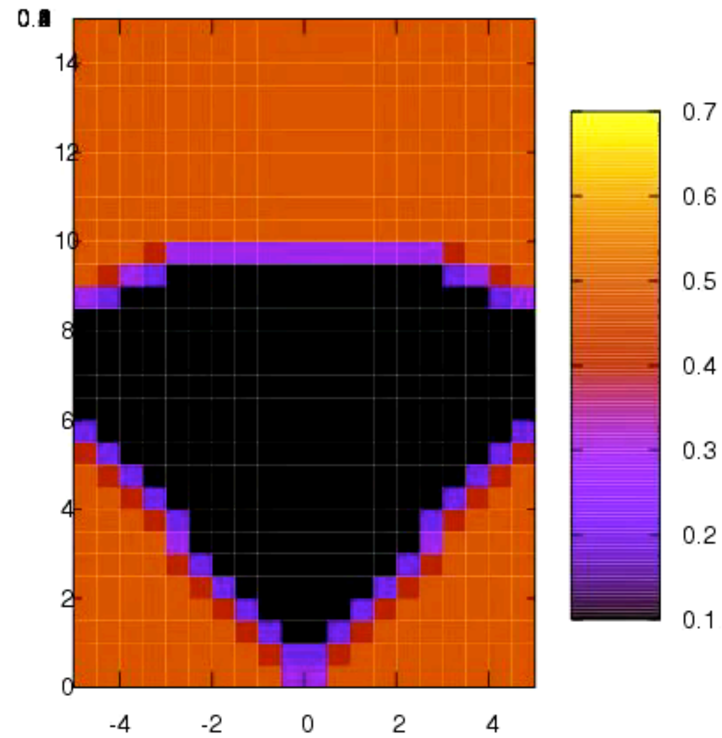
# Temporal filter



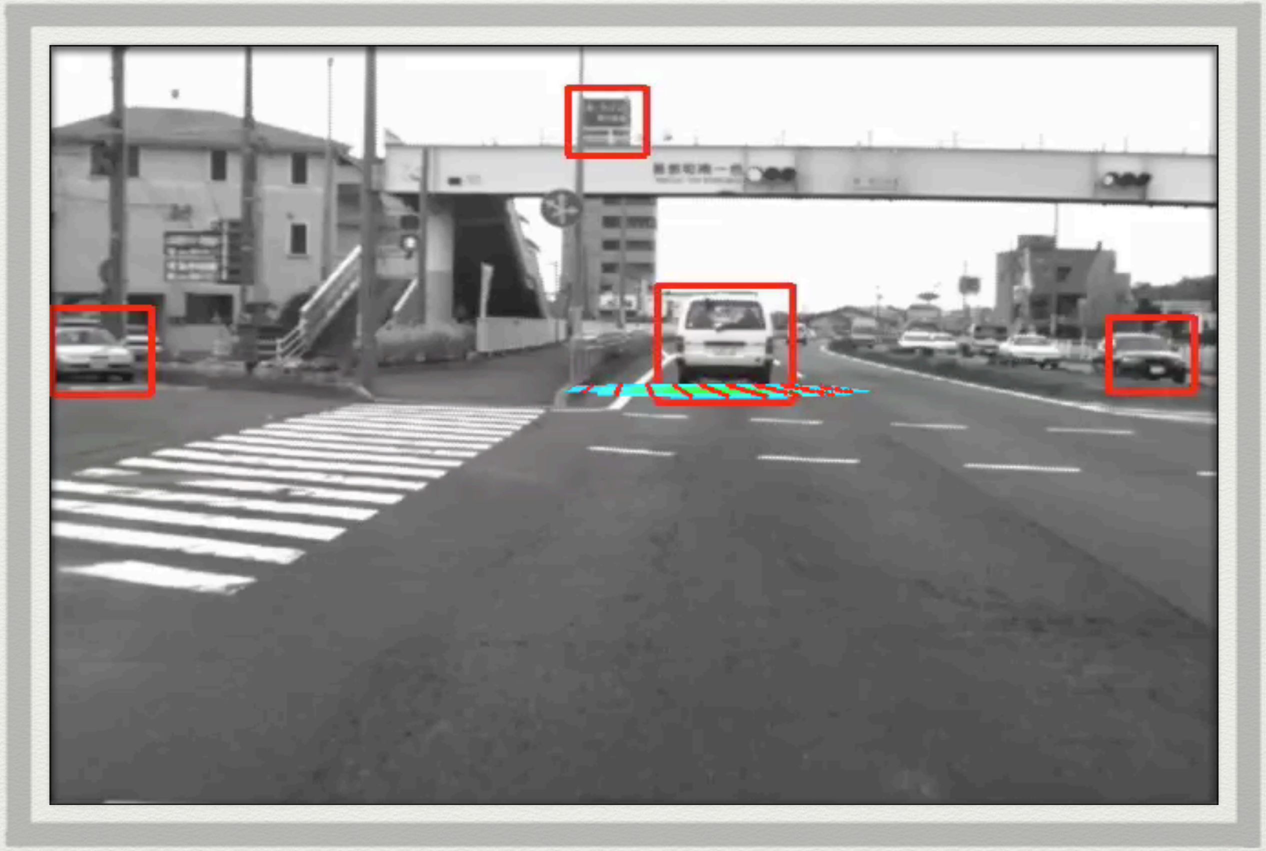



# Occultation



# Real time ADAS



http://www.probayes.com

# HIERARCHIES OF MODELS

# Calling subroutines

Program

Description

*Specification*

- Variables

$$X_1, X_2, \cdots X_N$$

- Decomposition

$$P(X_1, \cdots, X_N \mid \pi_1) = P(X_1 \mid \pi_1) P(X_2 \mid X_1, \pi_1) P(X_3 \mid \cancel{X_2}, X_1, \pi_1) \cdots$$

- Parametrical Forms

$$P\left(X_1 \mid \pi_1\right) = Uniform$$

$$P(X_2 \mid X_1, \pi_1) = Gaussians$$

$$\cancel{P(X_3 \mid X_1, \pi_1) = Histograms} \qquad P(X_3 \mid X_1, \pi_1) = P(X_3 \mid X_1, \pi_2)$$

*Identification*

- Leraning

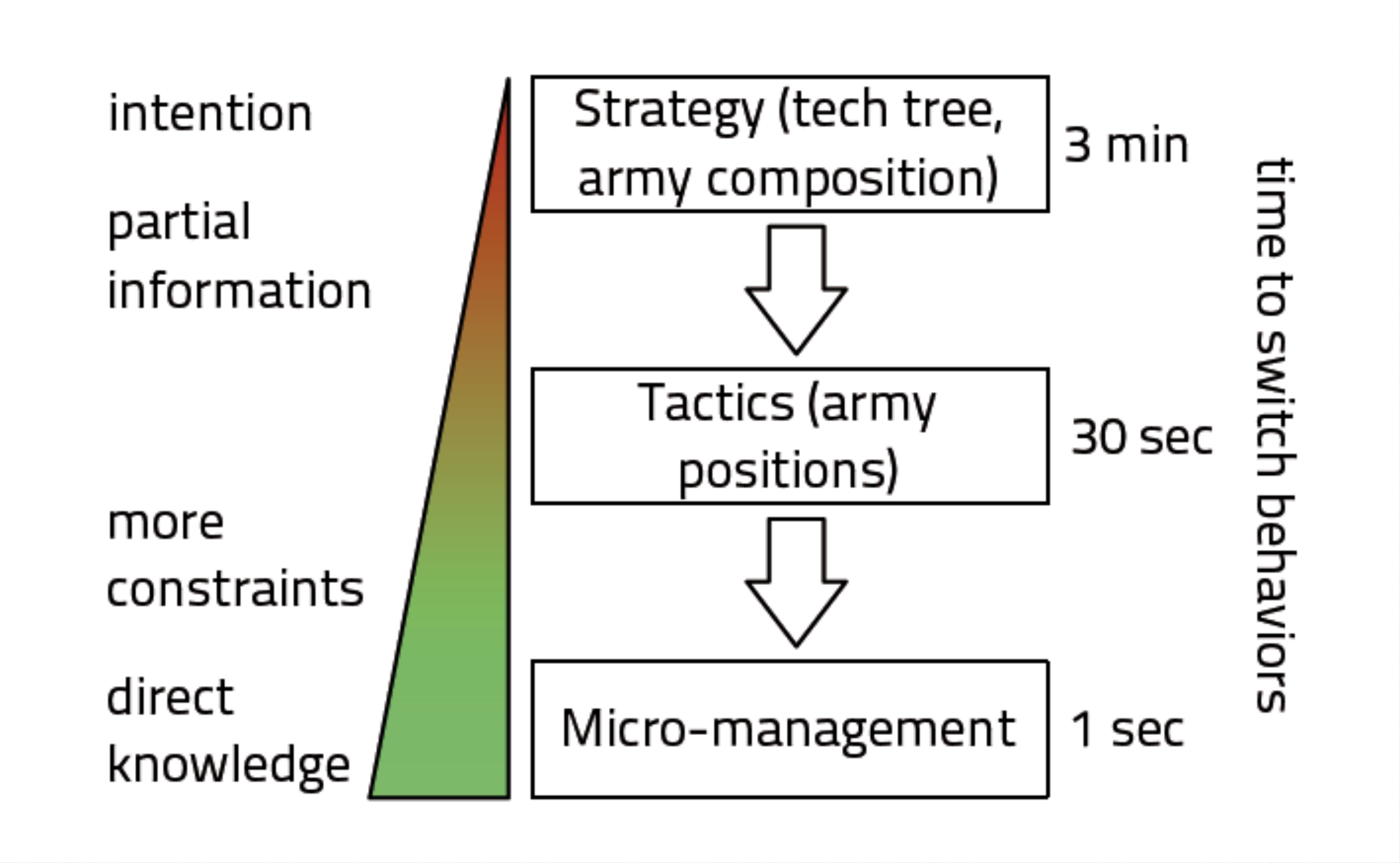
Question

Utilization

$$P(X_i, X_j \mid x_k, x_l, x_m, \pi_1)$$

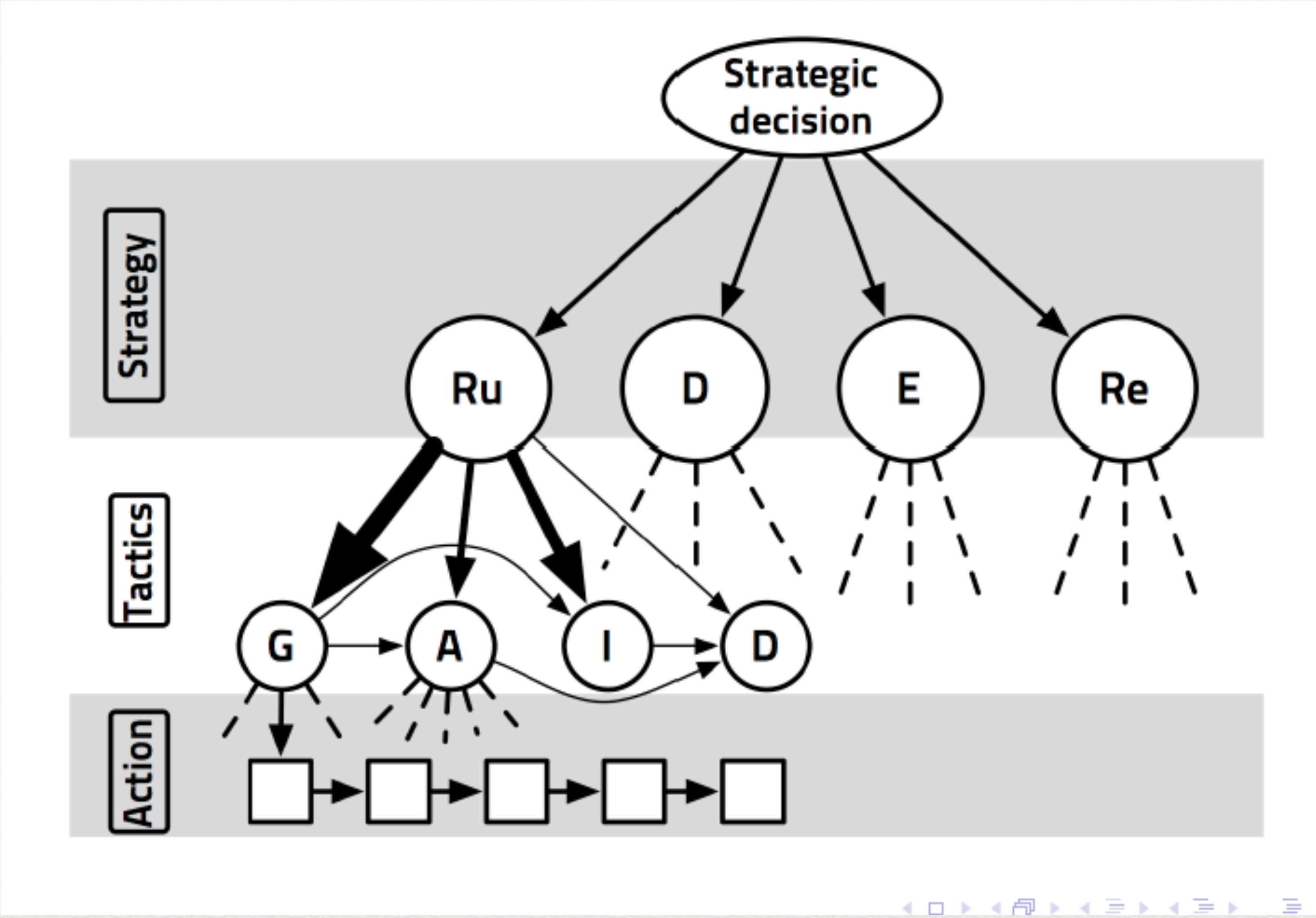
# StarCraft

# StarCraft

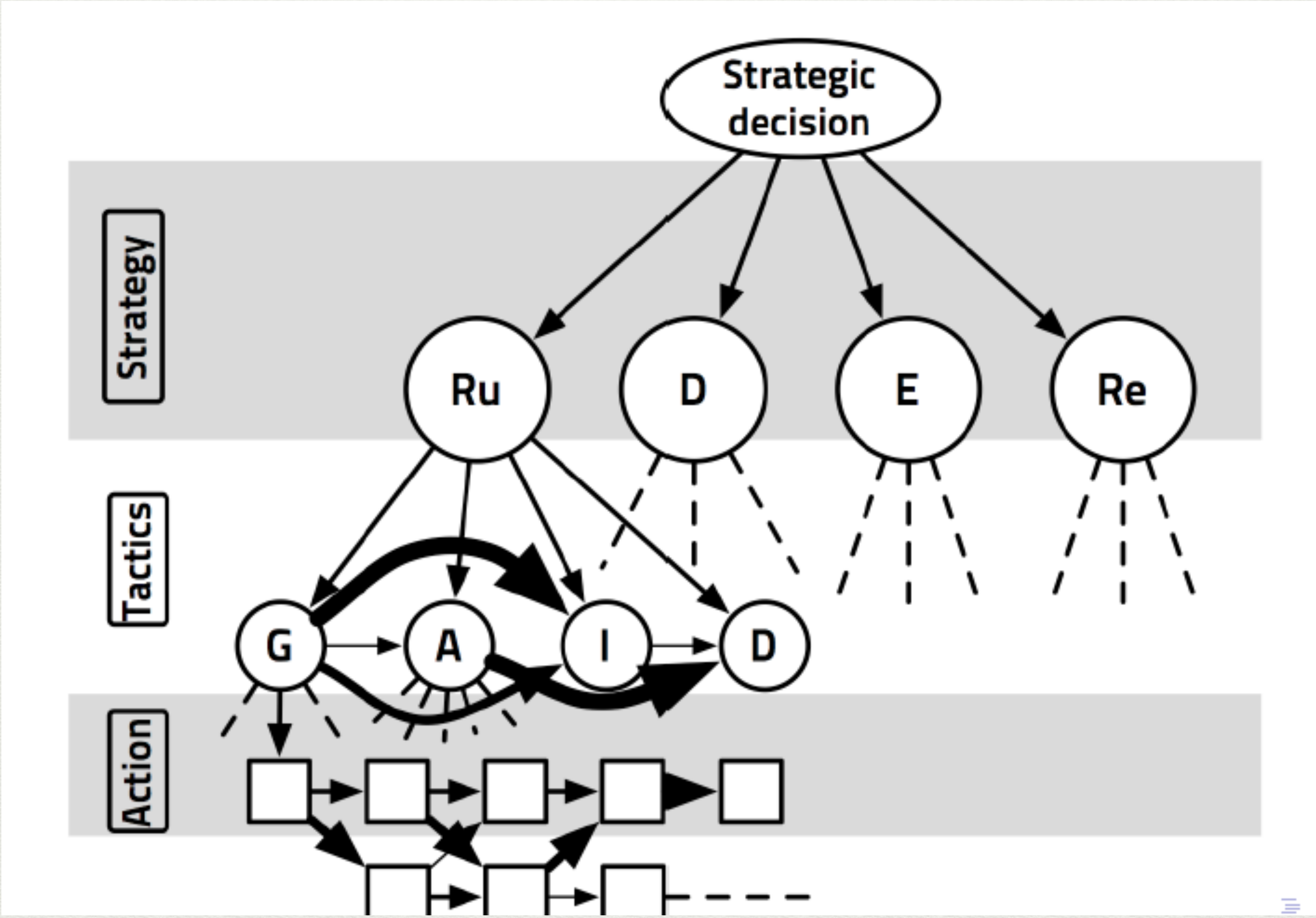


intention
partial information
more constraints
direct knowledge
Strategy (tech tree, army composition)
3 min
Tactics (army positions)
30 sec
Micro-management
1 sec
time to switch behaviors

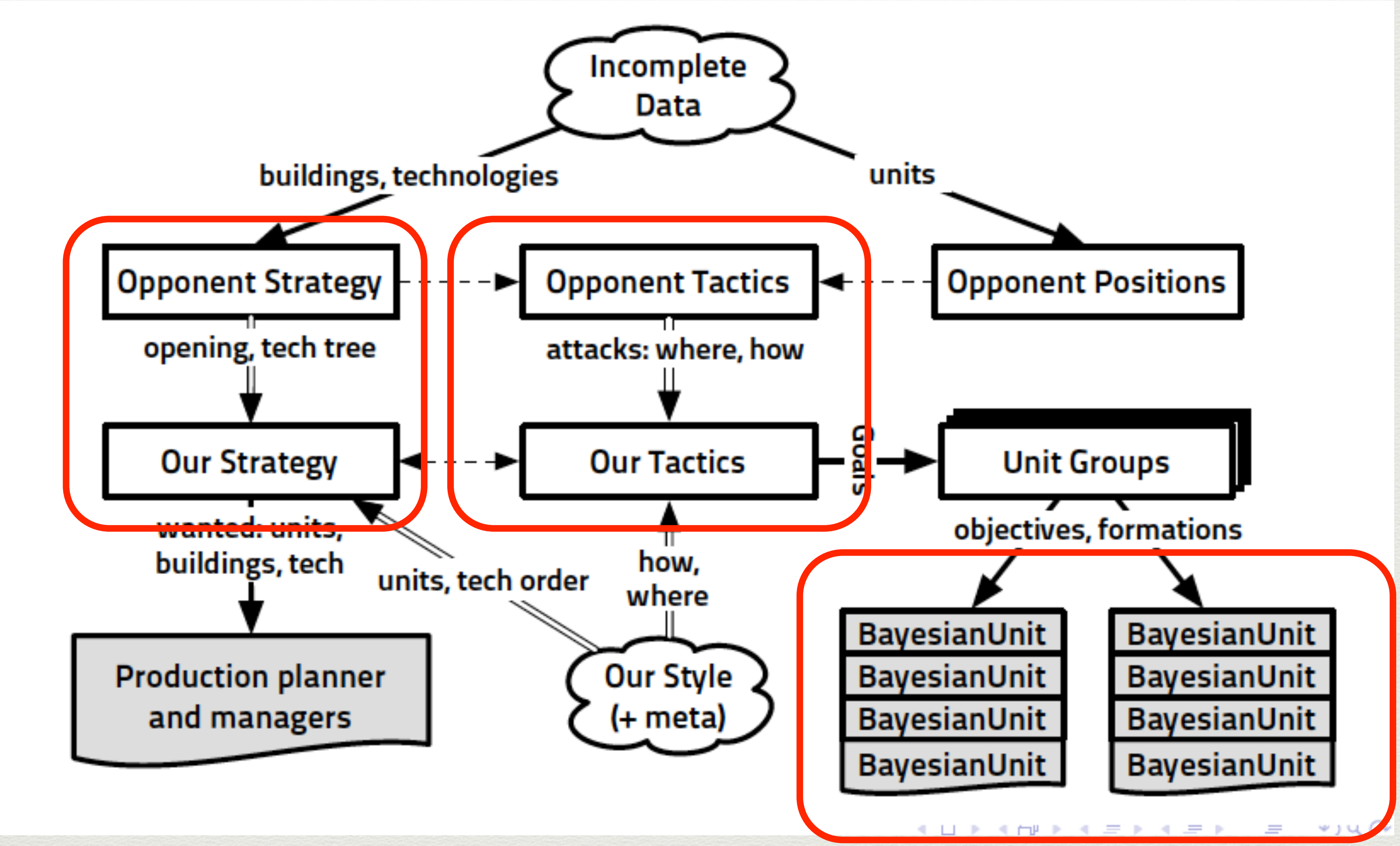
# StarCraft



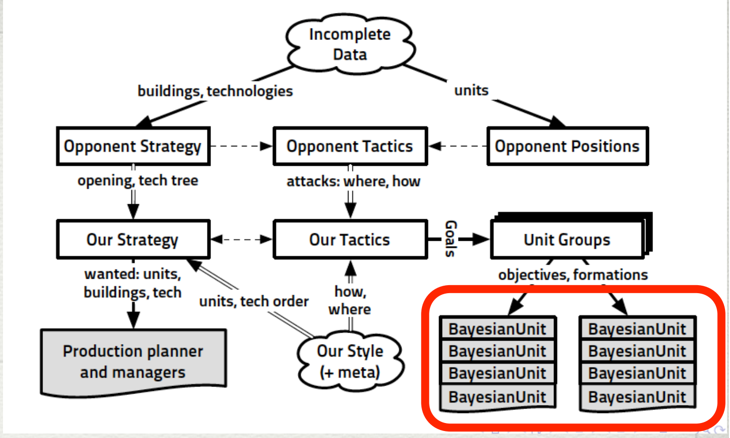
# StarCraft



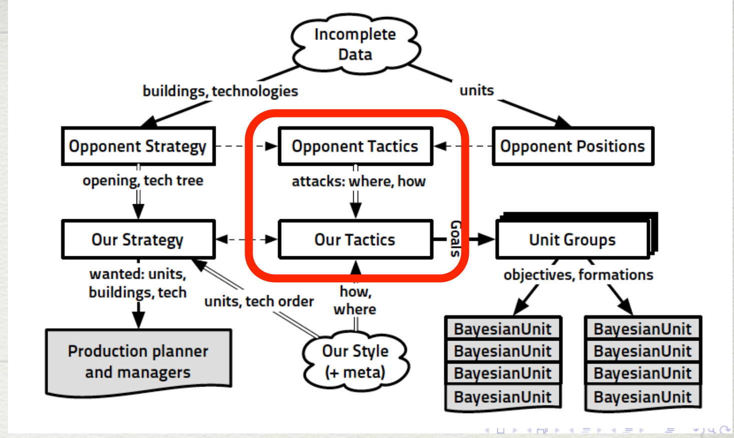
# StarCraft


Incomplete Data
buildings, technologies
units
Opponent Strategy
Opponent Tactics
Opponent Positions
opening, tech tree
attacks: where, how
Our Strategy
Our Tactics
Goals
Unit Groups
wanted: units, buildings, tech
units, tech order
how, where
objectives, formations
Production planner and managers
Our Style (+ meta)
BayesianUnit
BayesianUnit
BayesianUnit
BayesianUnit
BayesianUnit
BayesianUnit
BayesianUnit
BayesianUnit

# StarCraft


Incomplete Data
buildings, technologies
units
Opponent Strategy
Opponent Tactics
Opponent Positions
opening, tech tree
attacks: where, how
Our Strategy
Our Tactics
Goals
Unit Groups
wanted: units, buildings, tech
units, tech order
how, where
objectives, formations
Production planner and managers
Our Style (+ meta)
BayesianUnit
BayesianUnit
BayesianUnit
BayesianUnit
BayesianUnit
BayesianUnit
BayesianUnit
BayesianUnit

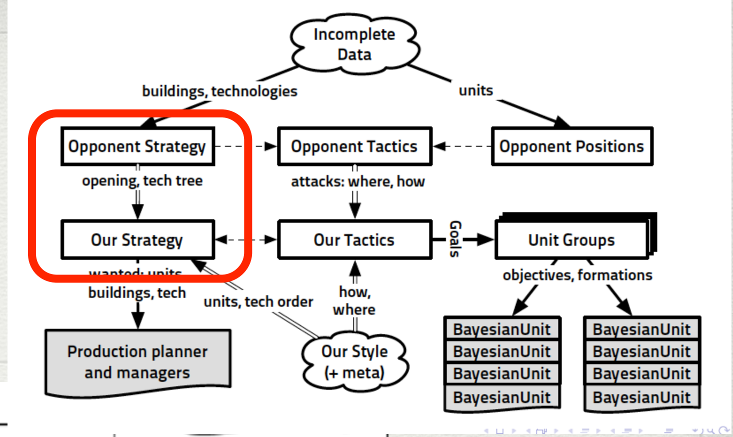

MENU

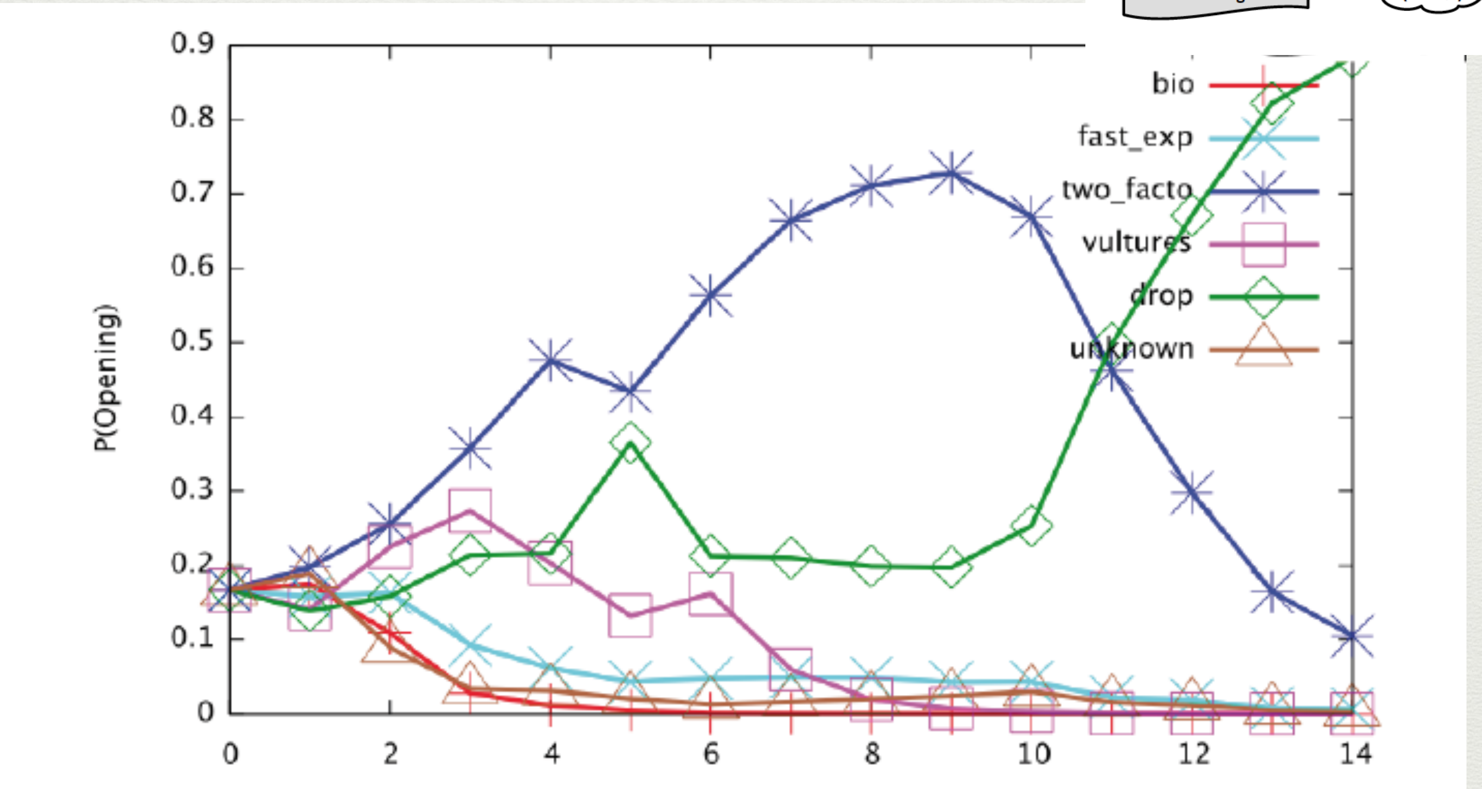

Incomplete Data
buildings, technologies
units
Opponent Strategy
Opponent Tactics
Opponent Positions
opening, tech tree
attacks: where, how
Our Strategy
Our Tactics
Goals
Unit Groups
wanted: units, buildings, tech
units, tech order
how, where
objectives, formations
Production planner and managers
Our Style (+ meta)
BayesianUnit
BayesianUnit
BayesianUnit
BayesianUnit
BayesianUnit
BayesianUnit
BayesianUnit
BayesianUnit

# StarCraft


http://www.8da.org
MENU

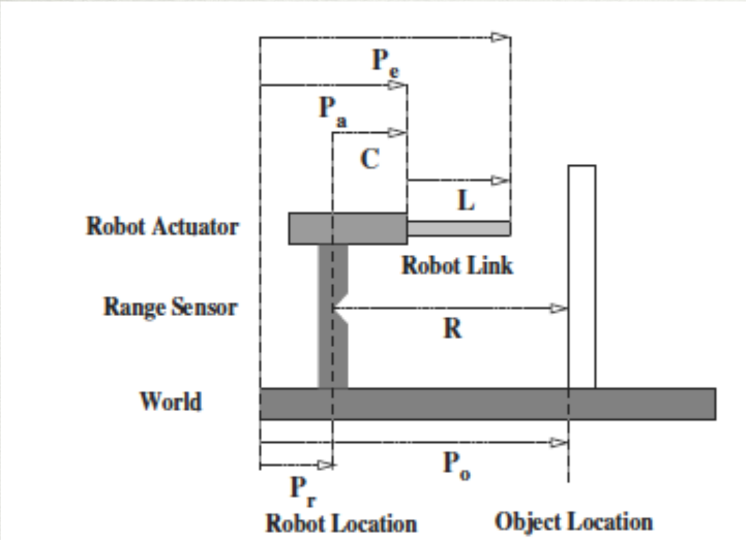
# StarCraft

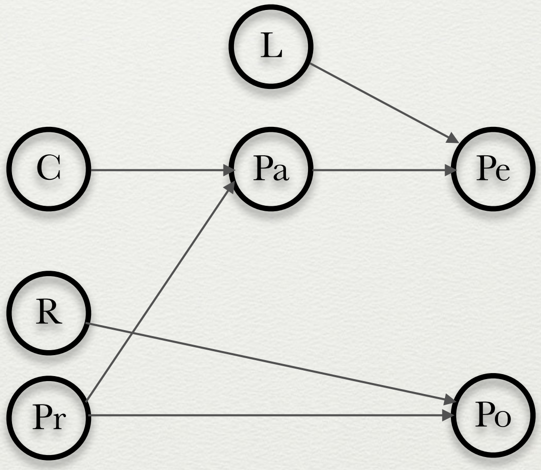

Incomplete Data
buildings, technologies
units
Opponent Strategy
Opponent Tactics
Opponent Positions
opening, tech tree
attacks: where, how
Our Strategy
Our Tactics
Goals
Unit Groups
buildings, tech
units, tech order
how, where
objectives, formations
Production planner and managers
Our Style (+ meta)
BayesianUnit

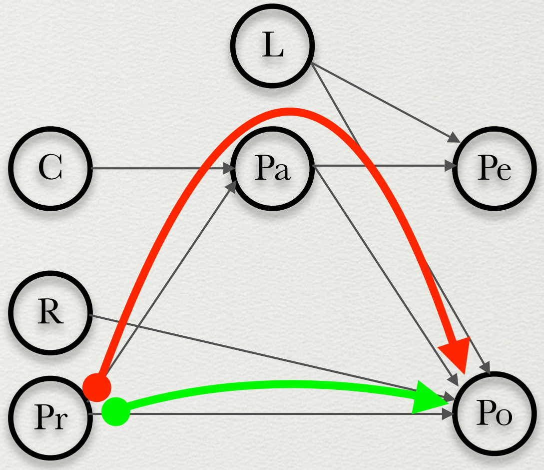

P(Opening)
bio
fast_exp
two_facto
vultures
drop
unknown
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
0 2 4 6 8 10 12 14

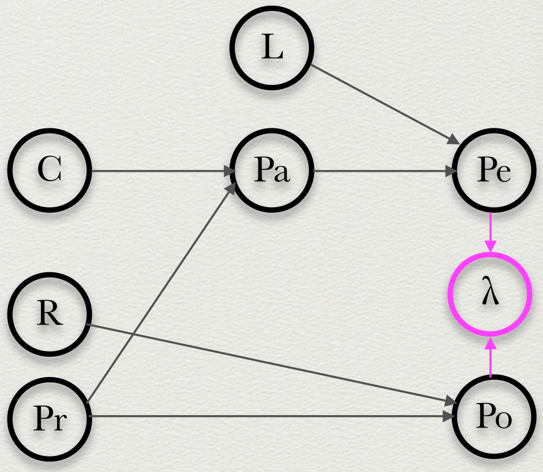
# COHERENCE VARIABLES TO DEAL WITH MULTI-INFERENCE PATHS AND SOFT EVIDENCE

# Multi inference paths

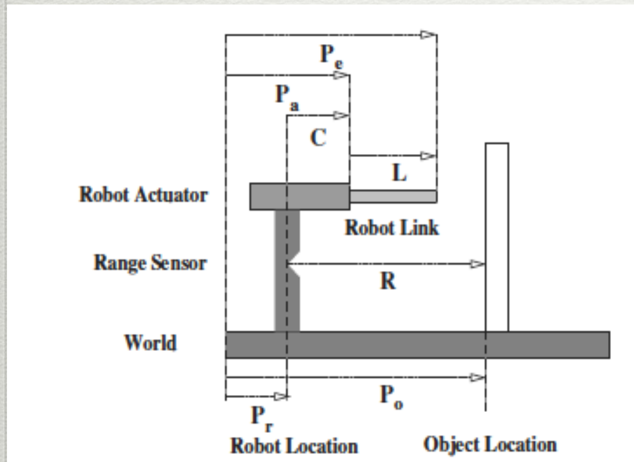


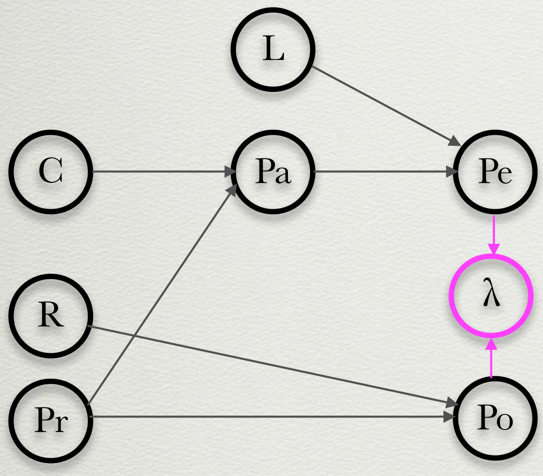



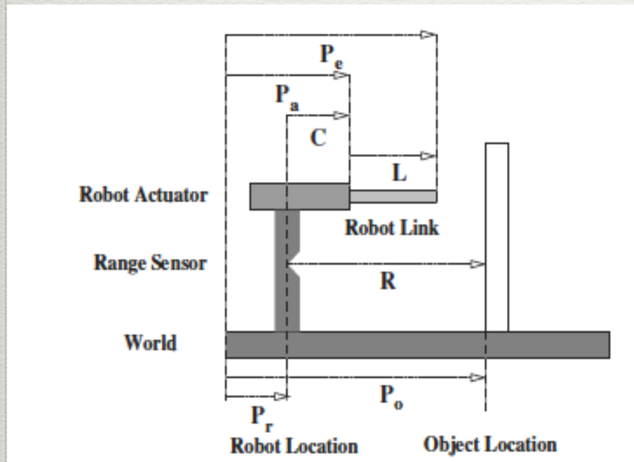
# Multi inference paths

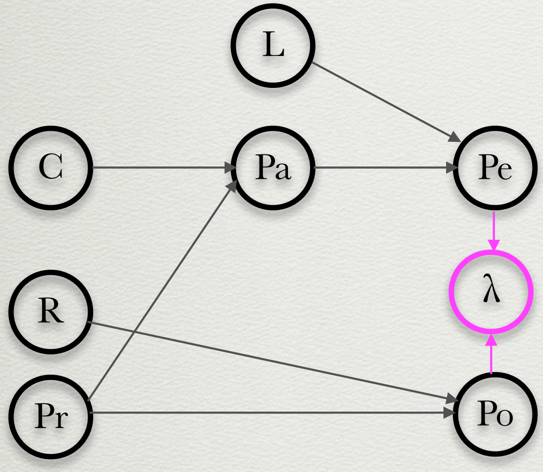



$$Pr\begin{cases} Ds\begin{cases} Sp(\pi)\begin{cases} Va: \\ P_r, R, P_o, C, P_a, L, P_e, \Lambda \\ Dc: \\ \begin{cases} & P(P_r\wedge\ldots\wedge\Lambda|\pi)\\ = & P(P_r|\pi)\times P(R|\pi)\times P(P_o|P_r\wedge R\wedge\pi)\\ & \times P(C|\pi)\times P(P_a|P_r\wedge C\wedge\pi)\\ & \times P(L|\pi)\times P(P_e|P_a\wedge L\wedge\pi)\\ & \times P(\Lambda|P_o\wedge P_e\wedge\pi)\end{cases}\\ Fo: \\ \begin{cases} P(P_o|P_r\wedge R\wedge\pi) = Normal(P_r, \epsilon_r)\\ P(C|\pi) = Uniform\\ P(P_a|P_r\wedge C\wedge\pi) = Normal(p_r + c, \epsilon_c)\\ P(L|\pi) = Normal(L_0, \epsilon_L)\\ P(P_e|P_a\wedge L\wedge\pi) = \delta_{p_e=p_a+l}\\ P(\Lambda|P_o\wedge P_e\wedge\pi) = \delta_{0\leq p_e-p_o\leq\epsilon_t}\end{cases}\end{cases}\\ Id \end{cases}\\ Qu: \\ P(C|r\wedge\lambda\wedge\pi)\\ P(P_o|r\wedge c\wedge\lambda\wedge\pi)\\ P(L|r\wedge c\wedge\lambda\wedge\pi)\end{cases}$$

# Multi inference paths




## Inverse kinematic :

$$P(C \mid r,\lambda)$$
$$\propto \sum_{P_r,P_o,P_a,L} \left[ P(P_r)P(P_o \mid P_r,r)P(C)P(P_a \mid P_r,C)P(L)P\left([P_e=P_o] \mid P_a,L\right) \right]$$

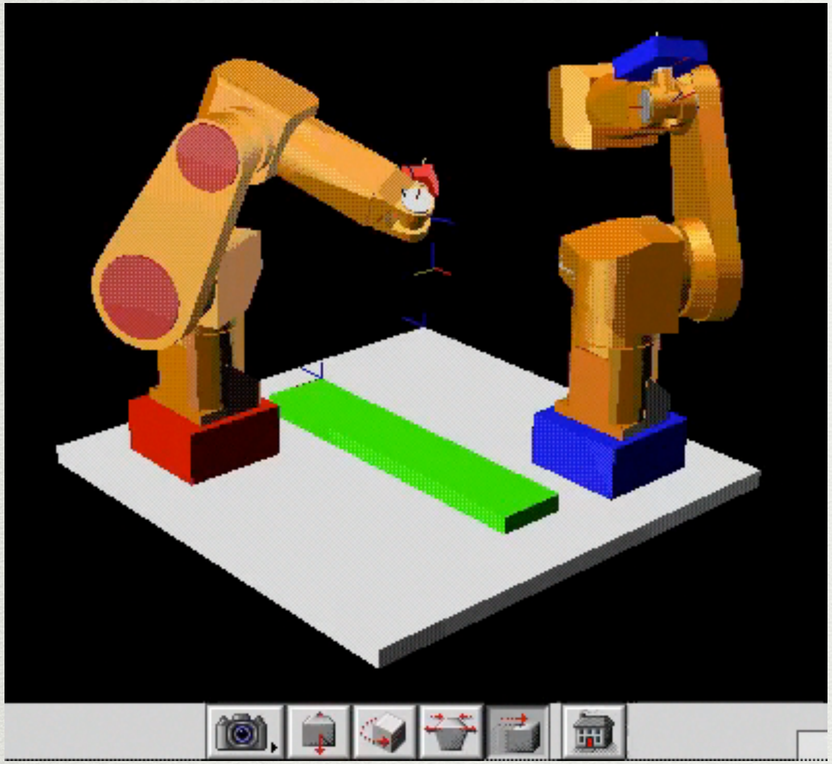
## Localization :

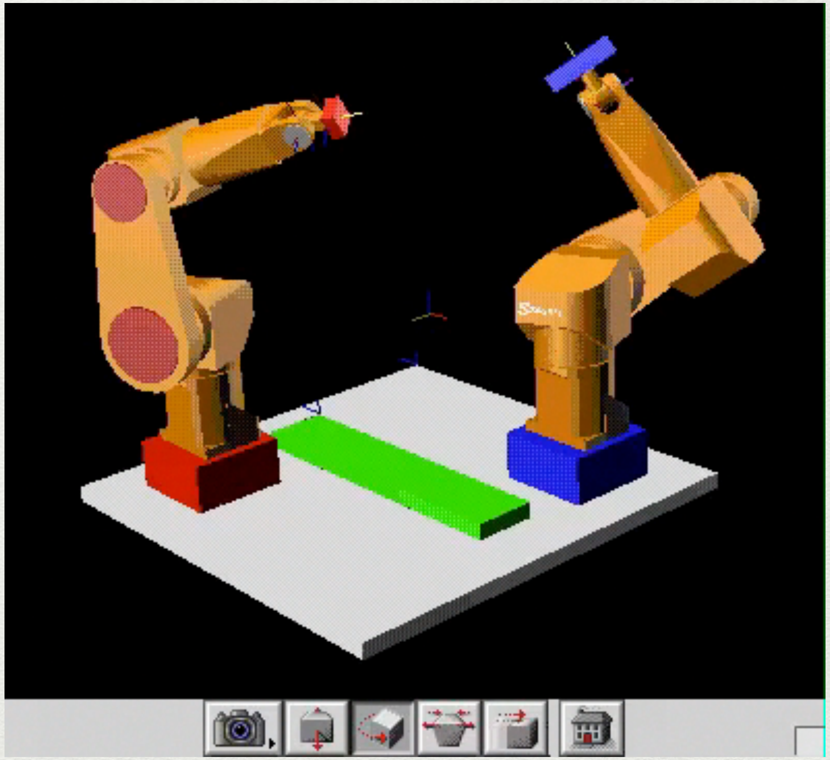
$$P(P_o \mid r,c,\lambda)$$
$$\propto \sum_{P_r,P_a,L} \left[ P(P_r)P(P_o \mid P_r,r)P(P_a \mid P_r,c)P(L)P\left([P_e=P_o] \mid P_a,L\right) \right]$$

## Calibration :

$$P(L \mid r,c,\lambda)$$
$$\propto \sum_{P_r,P_o,P_a} \left[ P(P_r)P(P_o \mid P_r,r)P(P_a \mid P_r,c)P(L)P\left([P_e=P_o] \mid P_a,L\right) \right]$$

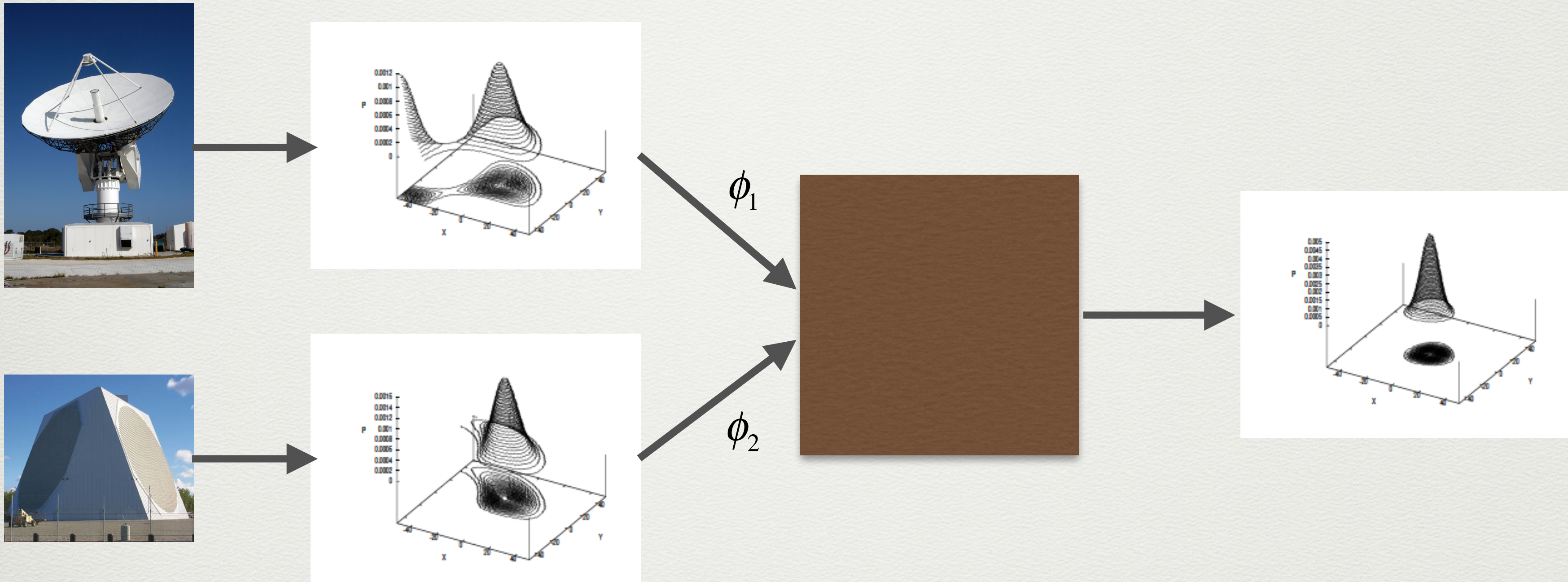
# Inverse kinematic





# Localization

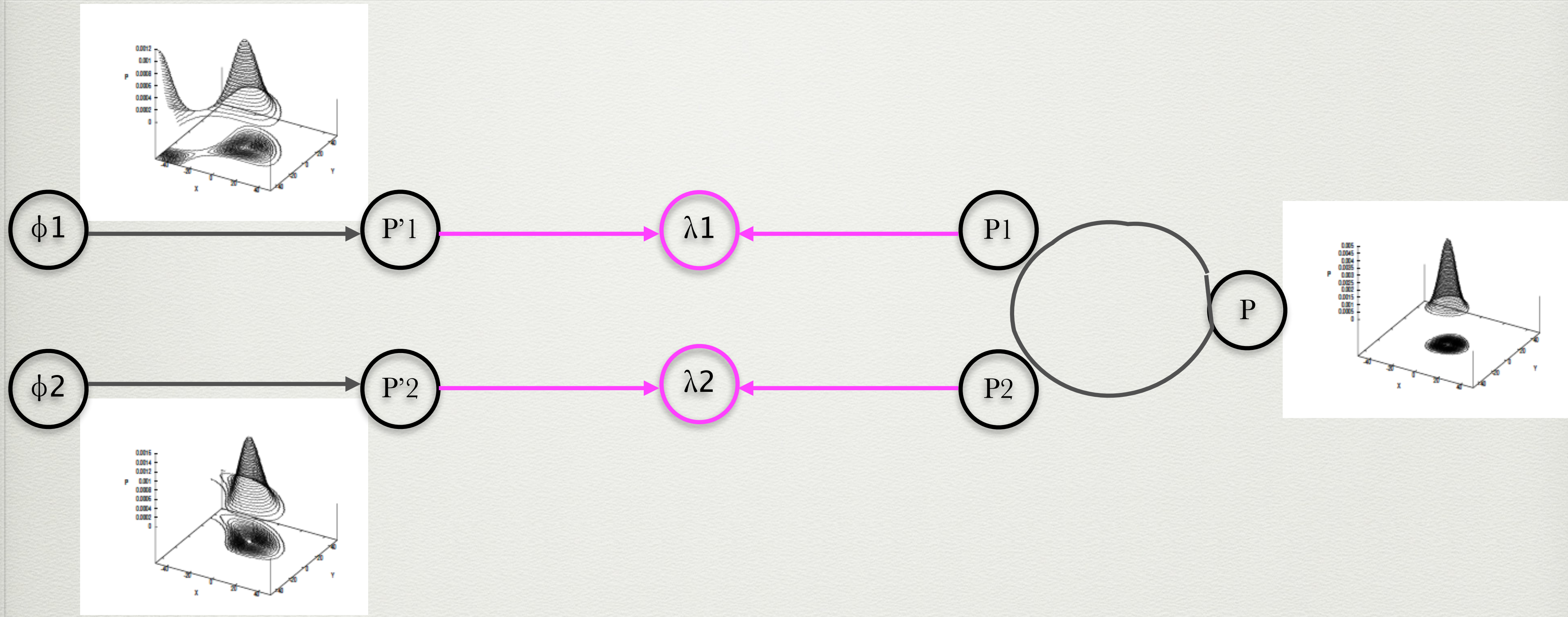
# Reasoning with soft evidence




$$P\left(X \wedge Y \mid x_1 \wedge y_1 \wedge x_2 \wedge y_2\right) \longrightarrow \cancel{P\left(X \wedge Y \mid P\left(X_1 \wedge Y_1\right) \wedge P\left(X_2 \wedge Y_2\right)\right)}$$

$$P\left(X \wedge Y \mid \phi_1 \wedge \phi_2\right)$$

# Reasoning with soft evidence

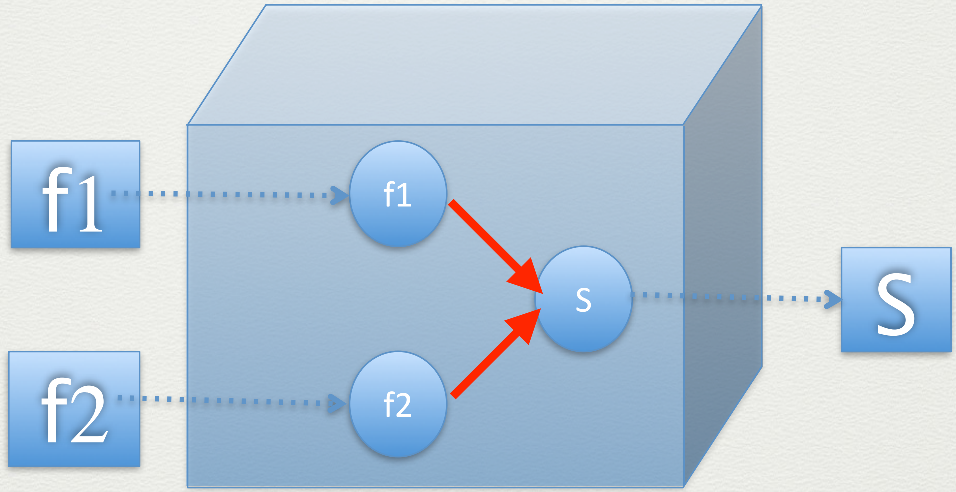



$$P(\Phi_1)P(P'_1 \mid \Phi_1)P(\Phi_2)P(P'_2 \mid \Phi_2)P(\lambda_1 \mid P_1, P'_1)P(\lambda_2 \mid P_2, P'_2)P(P_1, P_2, P)$$

$$P(P \mid \phi_1, \phi_2, \lambda_1, \lambda_2) \propto \sum_{P_1, P_2} \left[ P(P_1, P_2, P) P([P'_1 = P_1] \mid \phi_1) P([P'_2 = P_2] \mid \phi_2) \right]$$

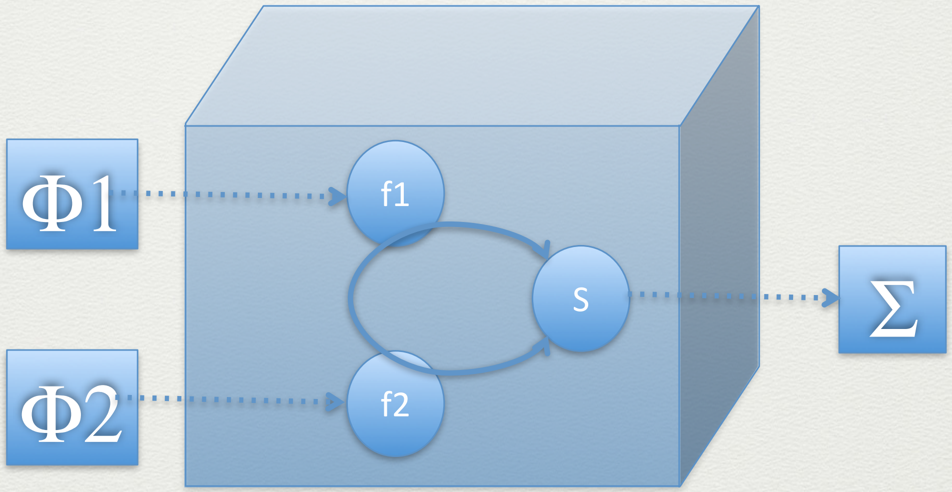
# TOWARD DEDICATED HARDWARE

# Boolean gates with 2 entries

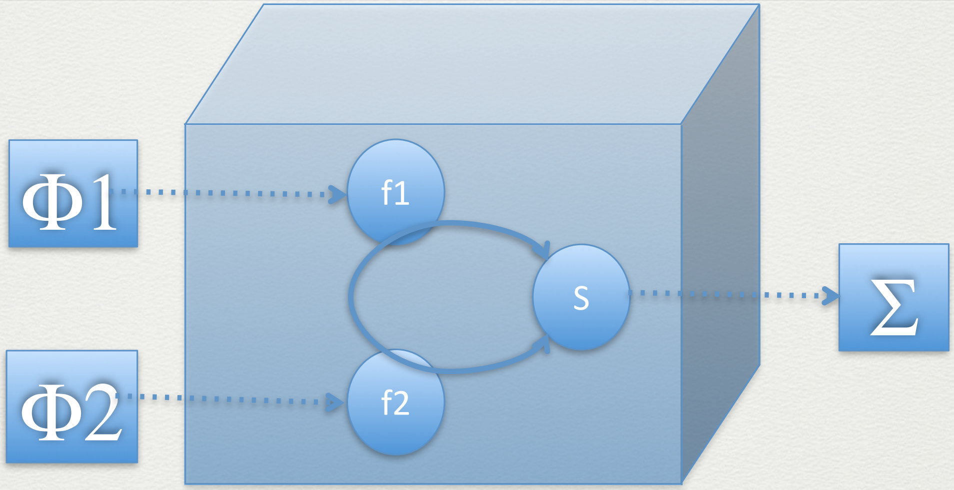



| f1 | f2 | S |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

# Bayesian gates with 2 entries



# Bayesian gates with 2 entries




$$P\left(\Phi_1,\Phi_2,F'_1,F'_2,F_1,F_2,S,\lambda_1,\lambda_2\right)$$
$$=P\left(\Phi_1\right)P\left(\Phi_2\right)P\left(F'_1\mid\Phi_1\right)P\left(F'_2\mid\Phi_2\right)P\left(F_1,F_2,S\right)P\left(\lambda_1\mid F_1,F'_1\right)P\left(\lambda_2\mid F_2,F'_2\right)$$

$$\Sigma=\frac{P\left(\left[S=1\right]\mid\phi_1,\phi_2,\lambda_1,\lambda_2\right)}{P\left(\left[S=0\right]\mid\phi_1,\phi_2,\lambda_1,\lambda_2\right)}$$

$$\Sigma=\frac{P(001)+P(011)\phi_2+P(101)\phi_1+P(111)\phi_1\phi_2}{P(000)+P(010)\phi_2+P(100)\phi_1+P(110)\phi_1\phi_2}$$

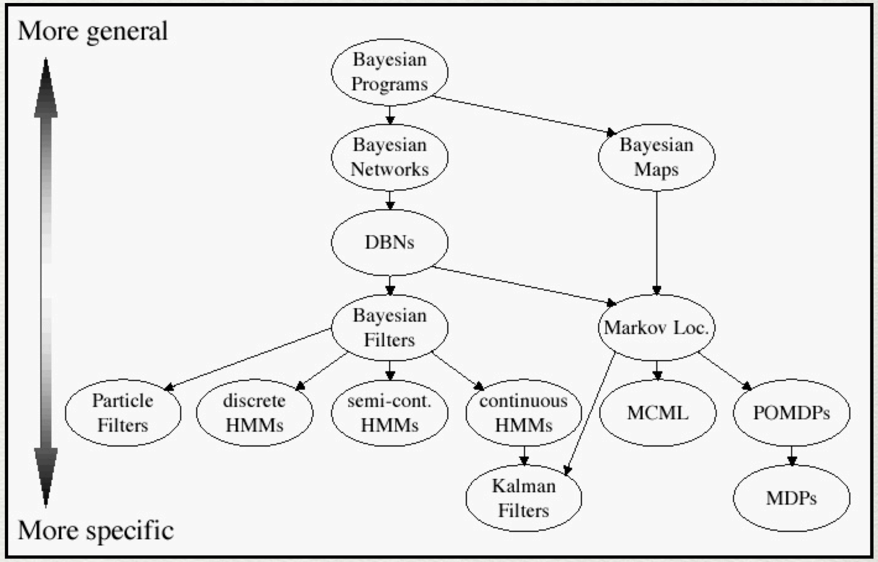
# Boolean gates with 2 entries

| f1 | f2 | S |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

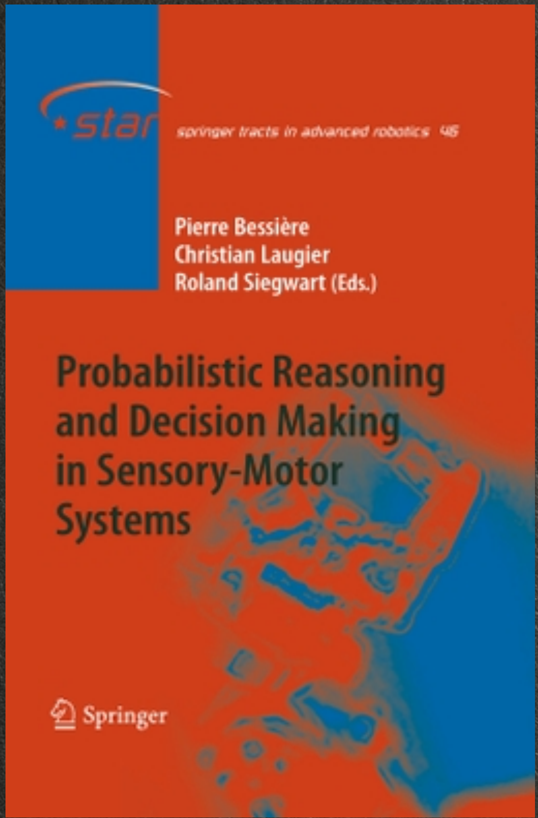
| F1 | F2 | S | P(F1,F2,S) |
|---|---|---|---|
| 0 | 0 | 0 | P(000) |
| 0 | 0 | 1 | P(001) |
| 0 | 1 | 0 | P(010) |
| 0 | 1 | 1 | P(011) |
| 1 | 0 | 0 | P(100) |
| 1 | 0 | 1 | P(101) |
| 1 | 1 | 0 | P(110) |
| 1 | 1 | 1 | P(111) |

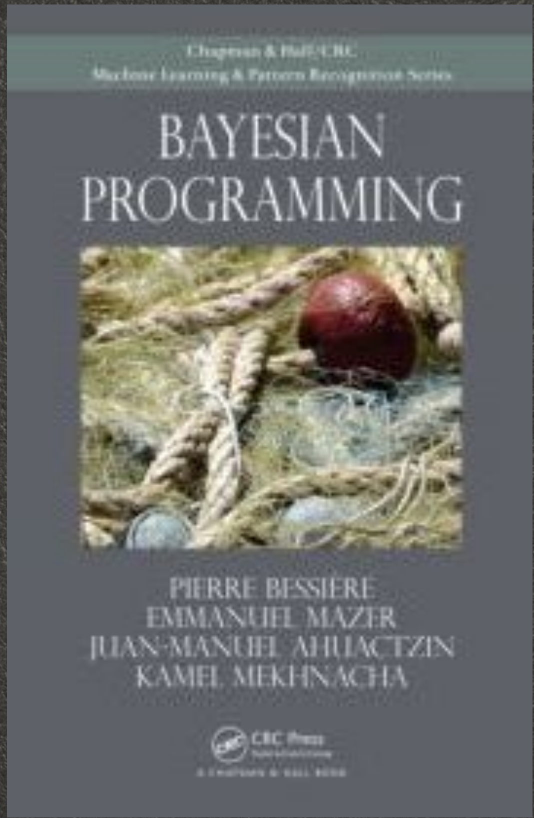
# Take home messages

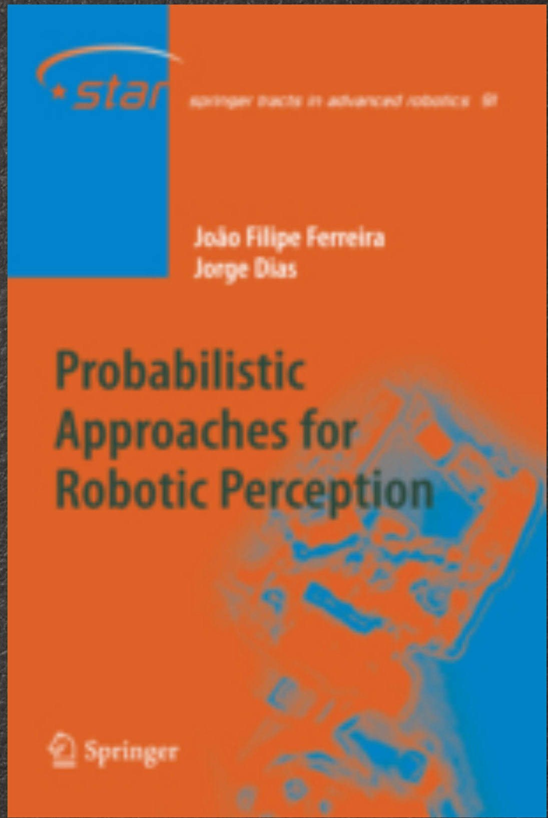
- A generic formalism
- A programming language
- An inference engine: ProBT
- Numerous robotics examples
- Toward dedicated hardware



# WANT TO KNOW MORE ?




Pierre.Bessire@upmc.fr