# Embodied moving-target seeking with prediction and planning

Noelia Oses (Fatronik-Tecnalia)
Matej Hoffmann (AI Lab, UZH)
Randal A. Koene (Fatronik-Tecnalia)

# Outline

Introduction

Building blocks:

- Forward model
- Inverse model
- Prey model

Models and experiments:

- Reactive model
- Prey prediction model
- Planning model

Conclusions and discussion

# Introduction

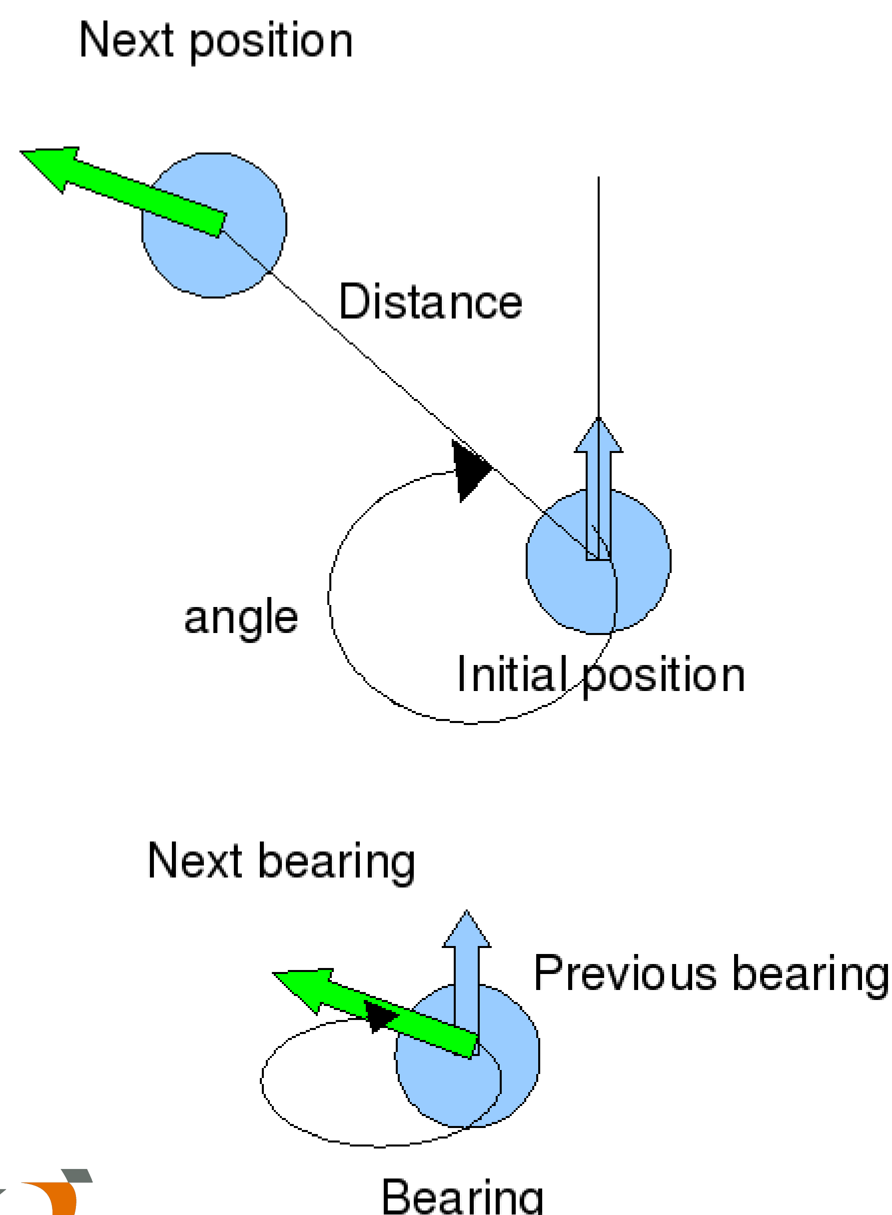
- Predator-prey scenario:
  - A mobile robot (hunter) needs to catch another mobile robot (prey)
  - Bio-inspired control: the problem has been solved in nature

# Introduction

- Hunter robot should learn to predict the consequences of its actions (forward model).
  - The actions are the different gaits applied for one time-step.
- Choosing an action to achieve a goal (inverse model)
- Prey model: hunter needs to learn how the prey moves to predict future prey positions.
- All models:
  - Robot-centered coordinates
  - No assumptions on the action space
  - Probabilistic

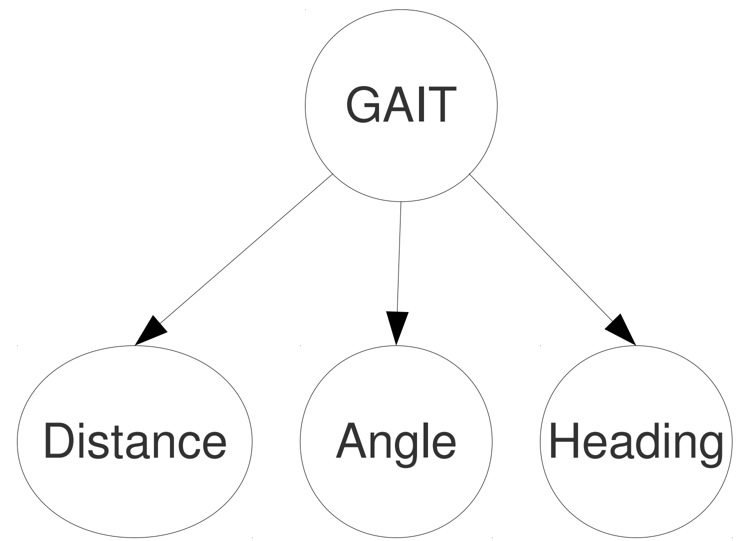
# Egocentric reference system

- Robot pose = location + bearing
- Location in polar coords centred at the robot's centre of mass
- Angles measured clockwise from robot's PA vector
- Bearing at t+Δt = angle that the robot's PA vector subtends with respect to the robot's PA vector at time t



# Forward model

- To predict the consequences (new relative pose) of actions (gaits).
- Naïve bayes classifier
- BN: powerful probabilistic framework to express the causal nature of a robot's control system.



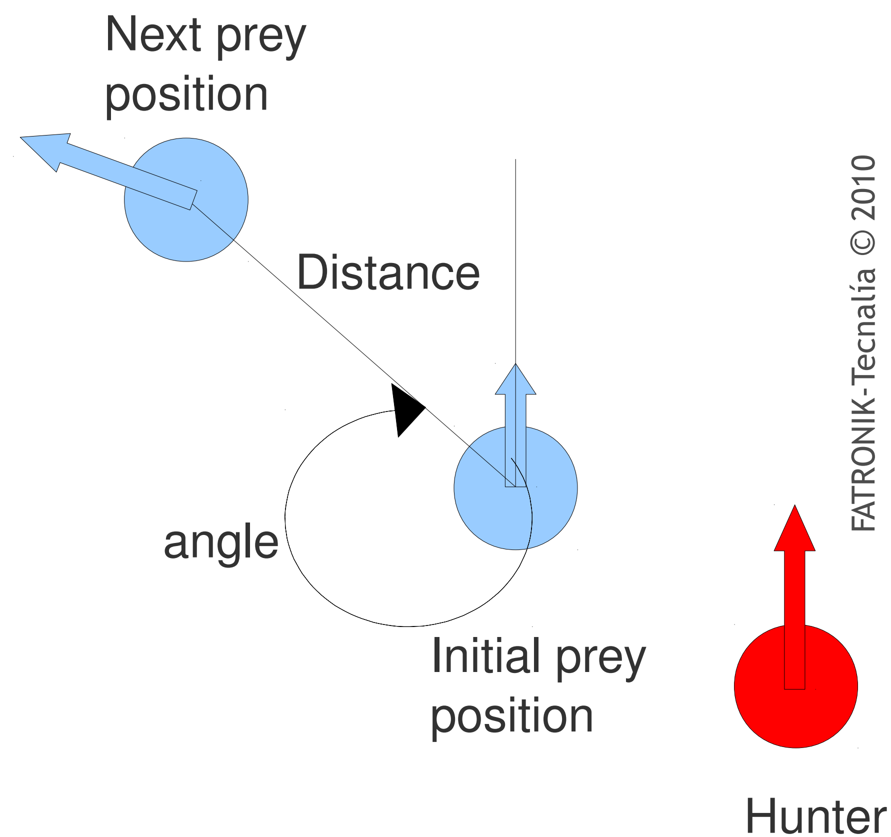

- BN parameters learned offline from motor-babbling data.

# Inverse model

- Goal: catch prey i.e. get to (next) prey's position i.e. get to position (distance,angle).
- Inverse model:
  - select action (gait) to achieve a desired goal.
  - given a target position (distance,angle) decide which gait to use
  - inference in the Bayesian network:
    - Obtain P( Gait | Dist=d, Angle=T ) from P(Dist=d|Gait) and P(Angle=T|Gait)

# Prey model

- Independent from the forward/inverse models of the hunter.
- Hunter learns a probabilistic transition model for the prey online.
- Transition = prey's pose at t+Δt with respect to prey's pose at t.
- Hunter uses this transition model to predict the prey's future positions.

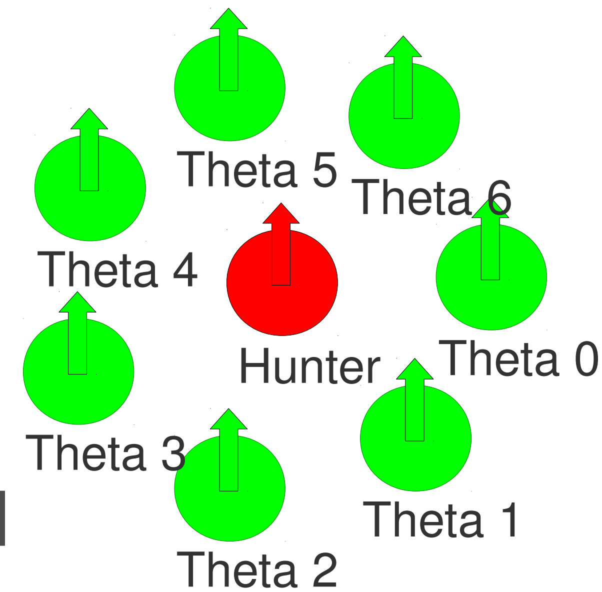


# Models

- Reactive model
  - Hunter: forward, inverse models
  - Prey's current pose
- Prey prediction model
  - Hunter: forward, inverse, prey models
  - Predicted prey pose at $t+\Delta t$
- Planning model
  - Hunter: forward, inverse, prey models + planning
  - Predicted prey pose several time-steps ahead

# Experiments

- Walled-in and open environments
- Seven initial states:
  - Prey at 5 bodies' distance
  - Angle: 0,1,2,3,4,5,6
  - Hunter and prey same heading
- Hunter: no obstacle avoidance
- Performance measure: simulated time elapsed until catch
- End: catch or one simulated minute
- 100 simulations per experiment
- Hunter: Khepera robot model with a set of 10 gaits.
- Prey: Khepera model with Cyberbotics' Webots' Braitenberg controller





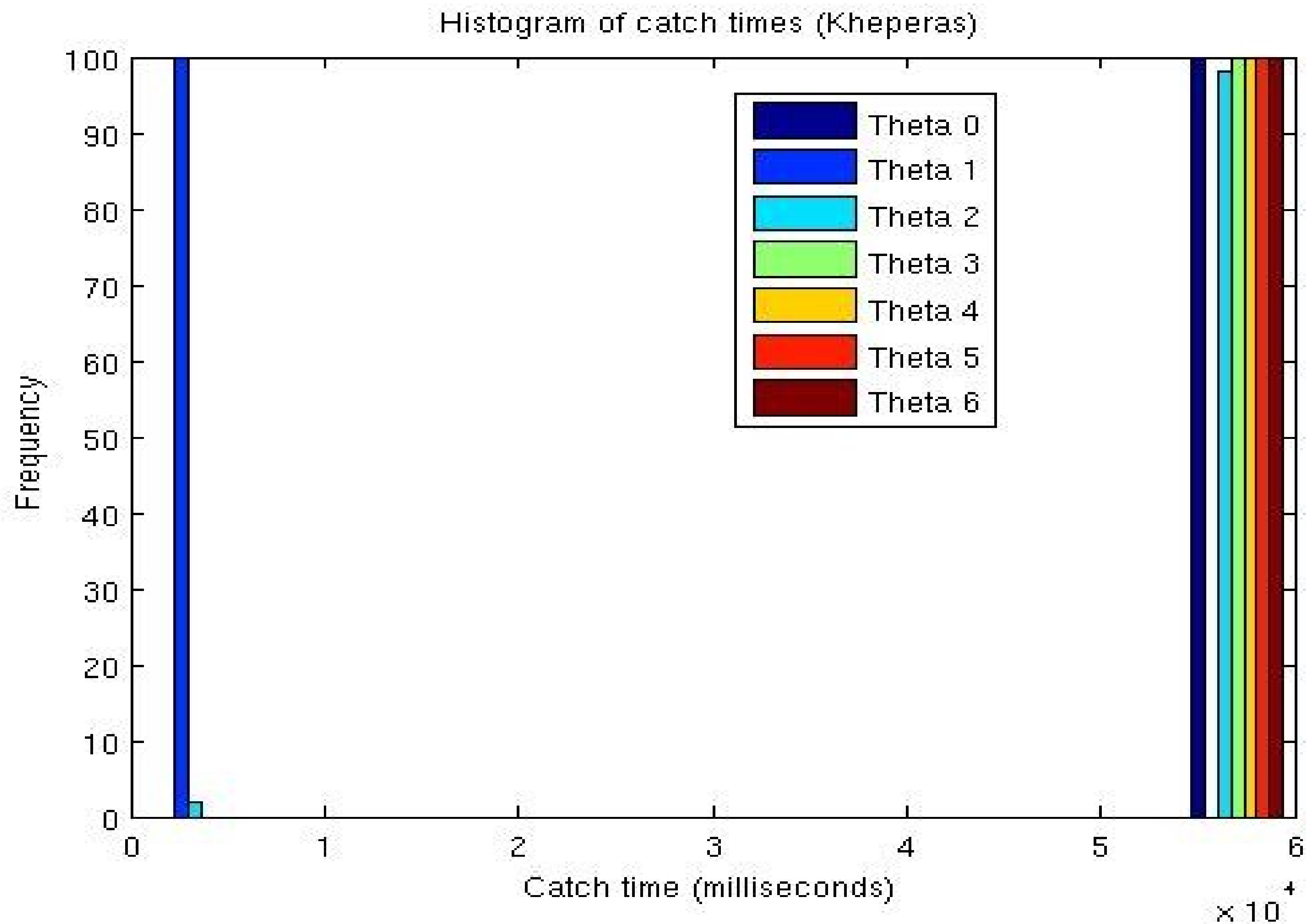

fatronik tecnalia

# Reactive model

- Hunter: forward, inverse models
- Prey's current pose
- Hunter applies gait determined by inverse model.
- Hunter catches the prey only in very concrete circumstances.
- It appears to follow prey around

fatronik tecnalia

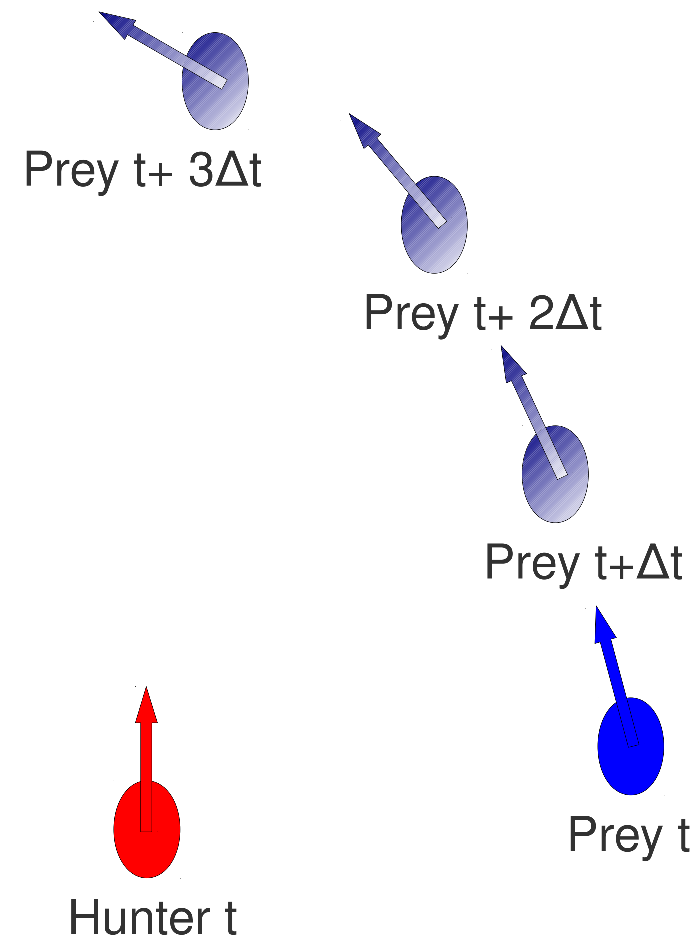
# Reactive model results




fatronik tecnalia

# Prey prediction model

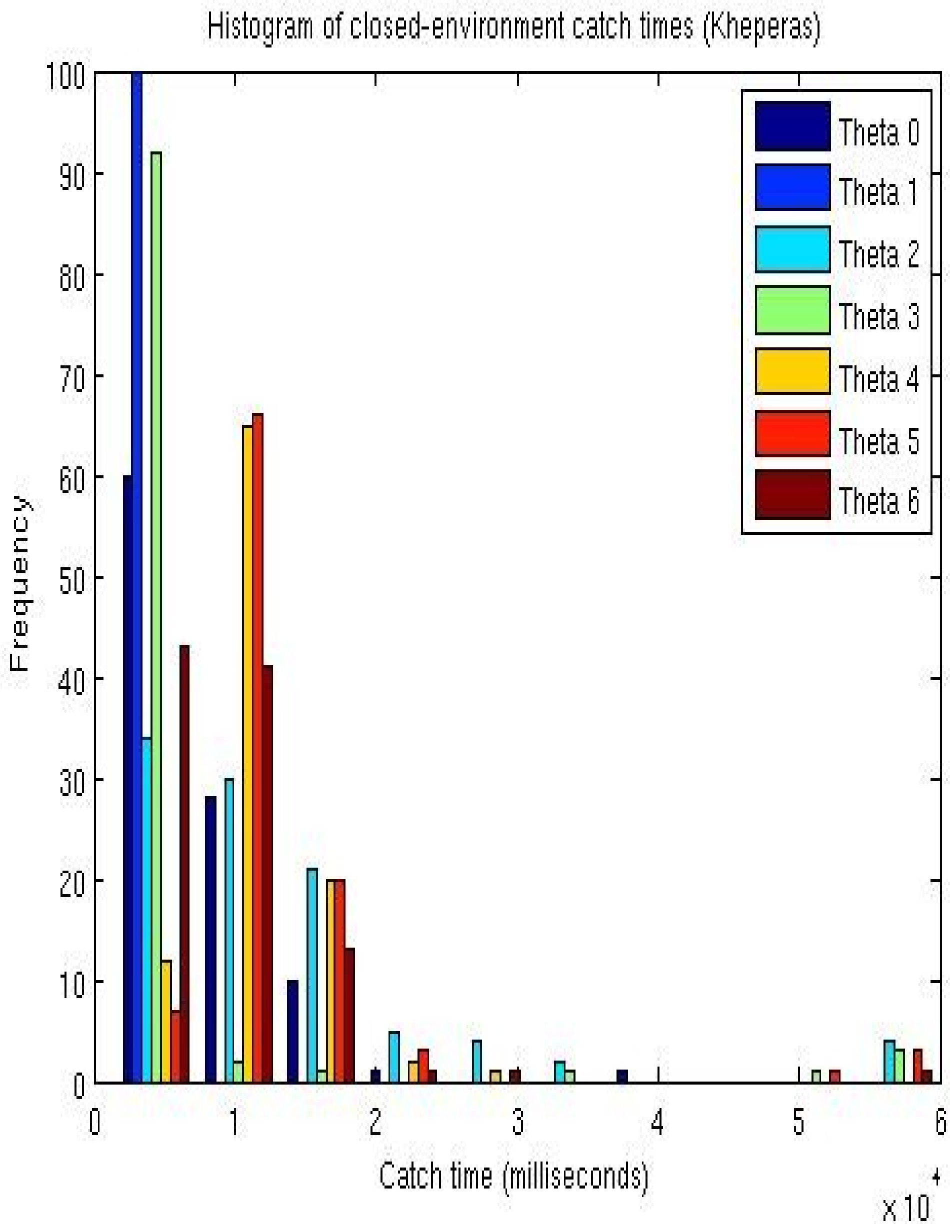
- Predicts prey's future position (sampling)
- feeds this to the inverse model and applies resulting gait
- Lookahead for prey's prediction depends on distance between hunter and prey

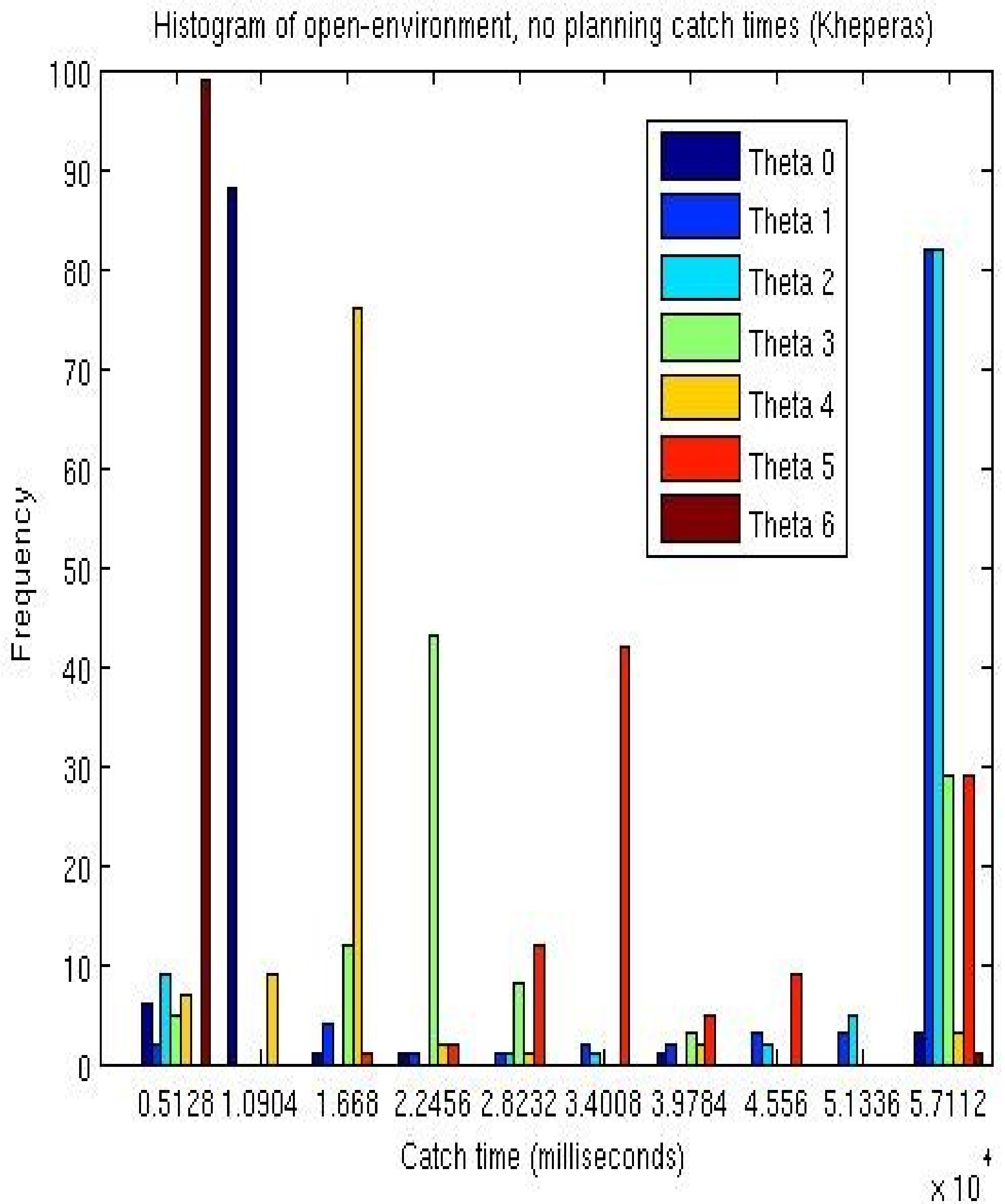


# Results of prediction model

- Closed environment
- Open environment

# Prey prediction results




tecnalia

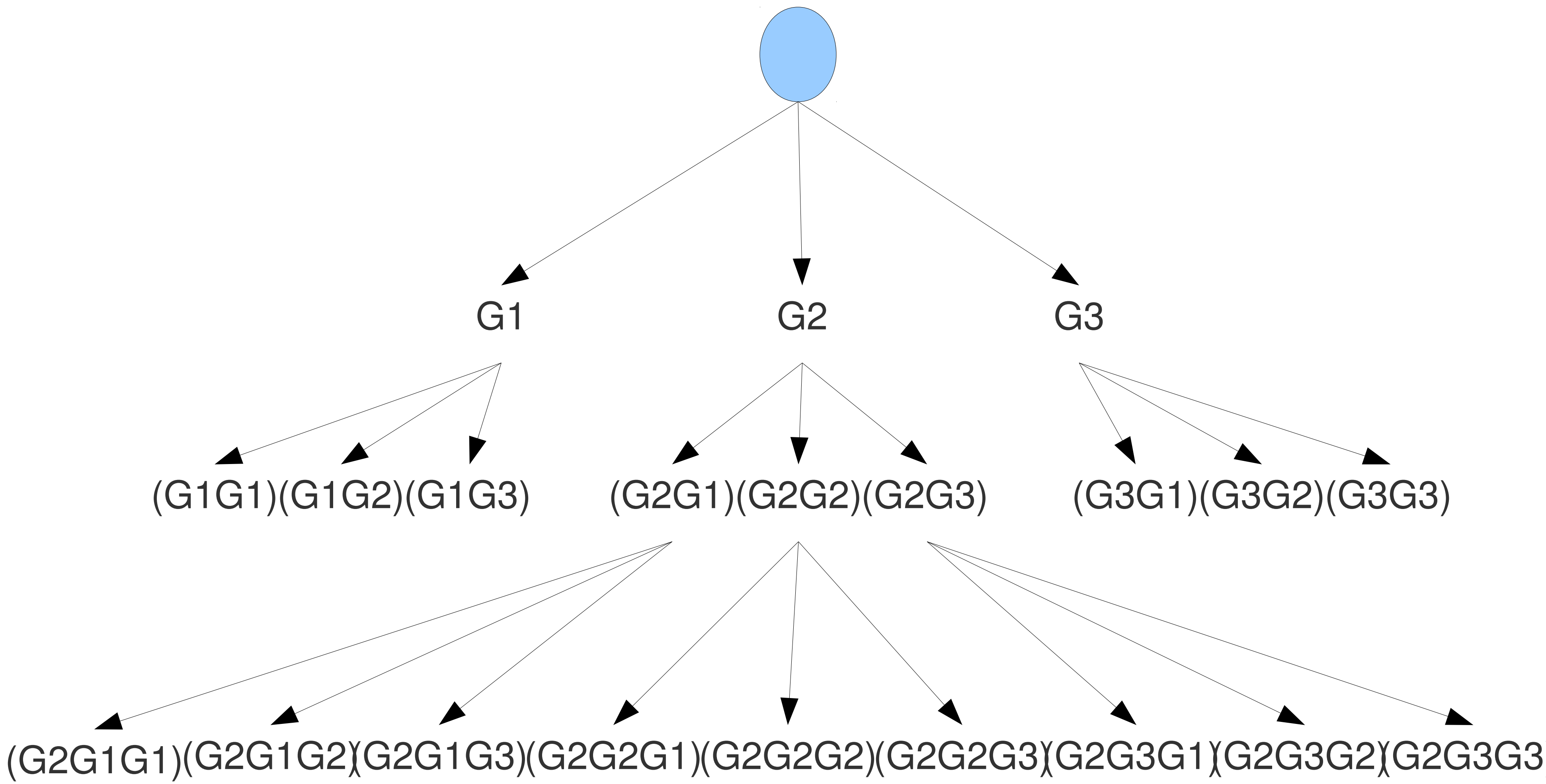
# Planning model

- Hunter: forward, inverse, prey models + planning
- Predicts several future positions of the prey at different time-steps
- Finds a sequence of gaits for the hunter so that it minimises the distance between hunter and prey

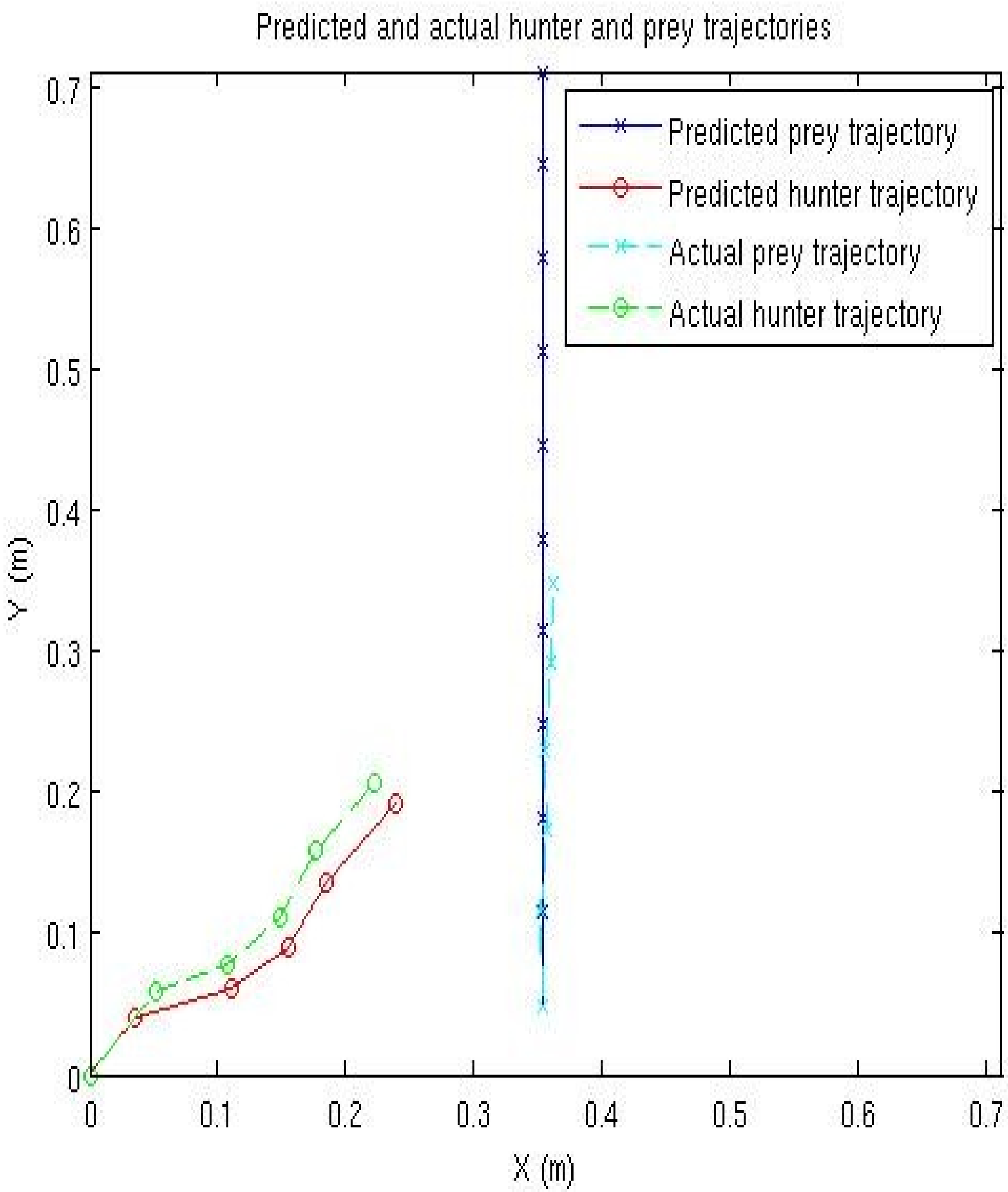
# Heuristic solution for planning

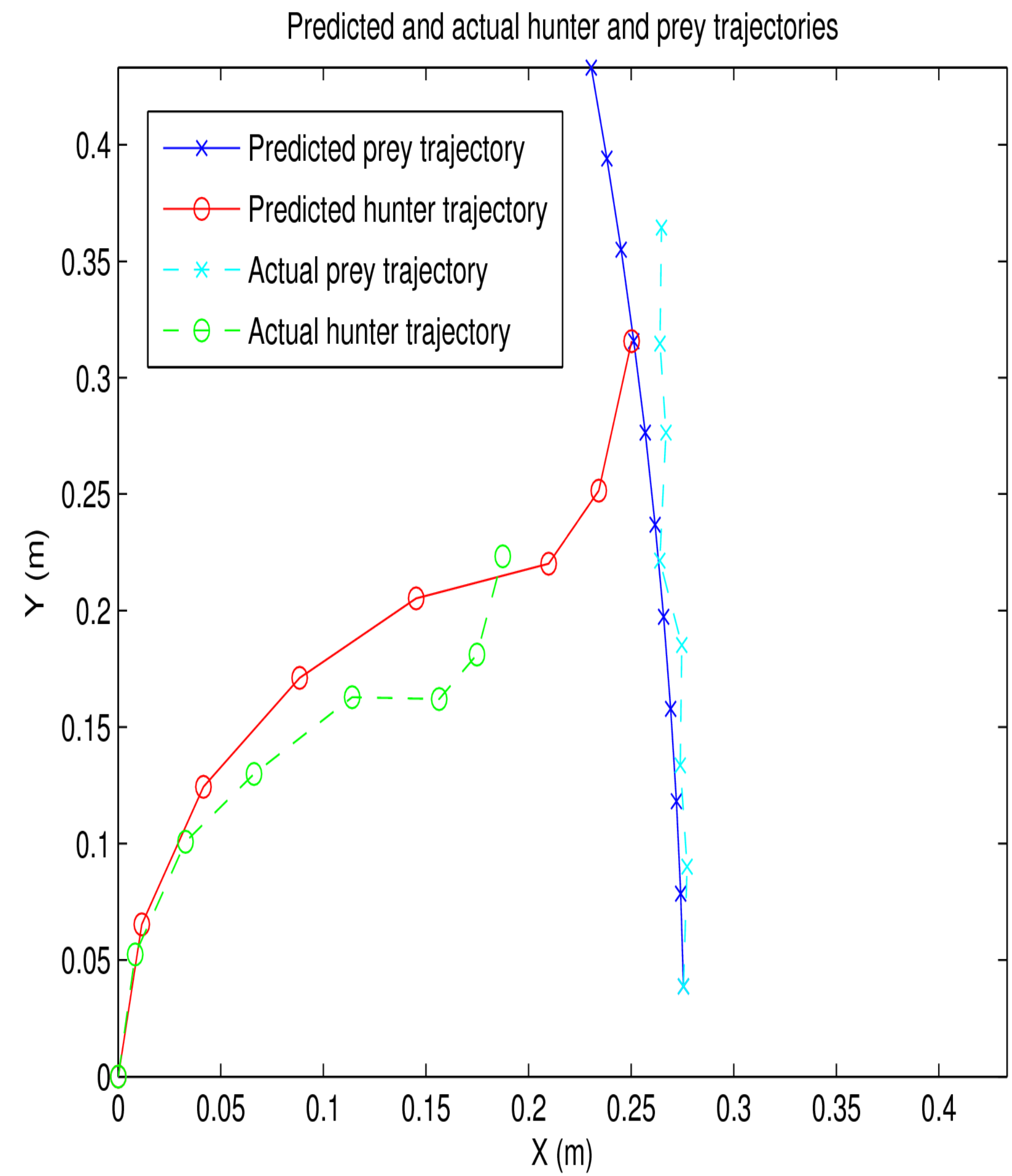
- Builds a search tree of the different gait combinations for the hunter
- For each possible sequence of gaits a hunter trajectory is sampled (instead of calculating the whole distribution)
- Node value: distance between hunter and prey
- Best-first search (expand the node with the best value first)
- Pruning: Eliminate gait sequences with more than one gait transition

# Heuristic solution



# Planning model examples

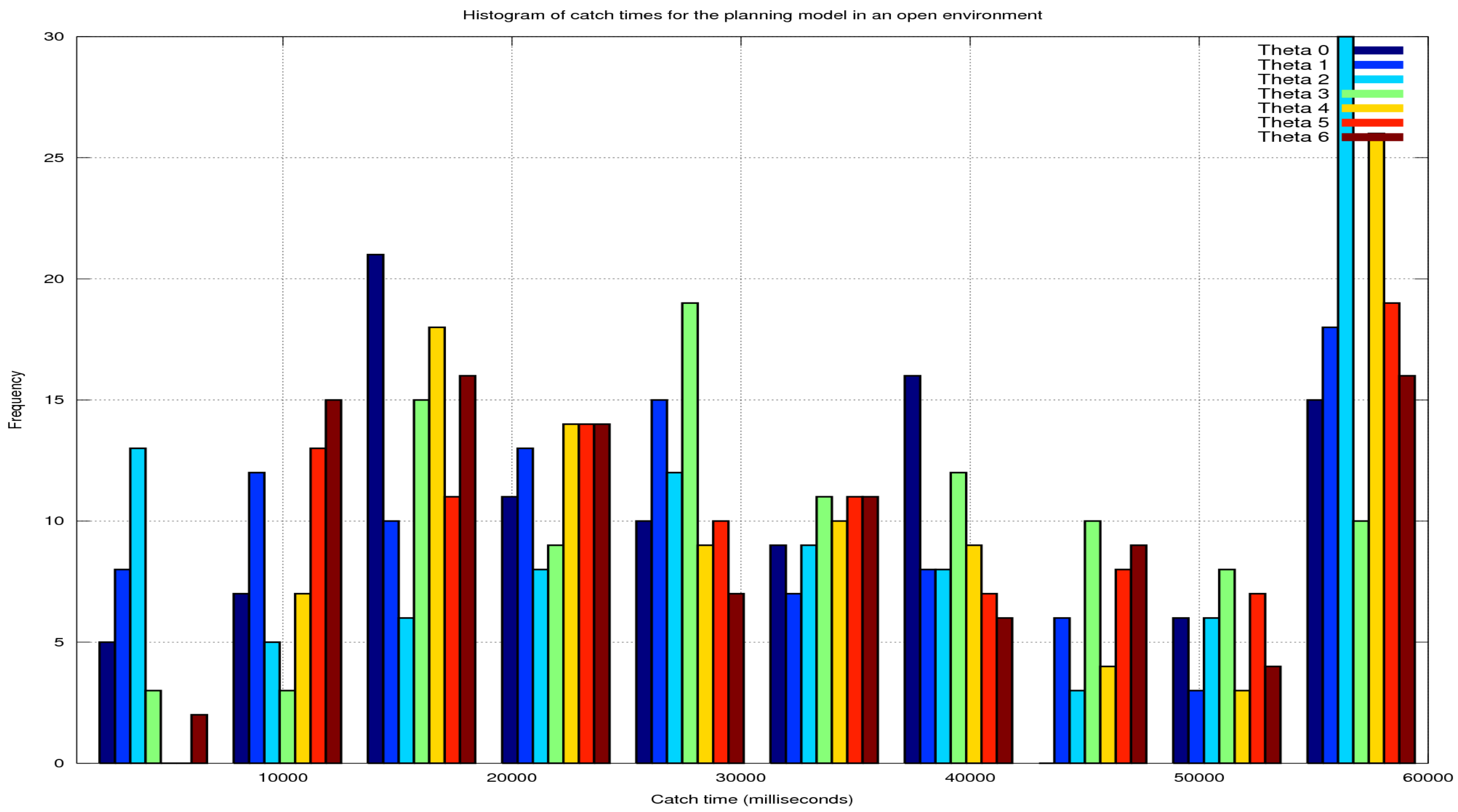



fatronik tecnalia

# Planning model

# Planning results



# Conclusions

- Bottom-up approach: only added cognitive capabilities as and when necessary
- Architecture properties:
  - Egocentric coordinate system
  - Arbitrary action repertoire (discrete action space)
  - Models learned *ab initio*
  - *Account for and plan with uncertainty*
- *Further work:*
  - *Extend work to legged platform*
  - *Adding real sensing of the prey*
  - *Studying various cost functions for the planning (e.g. energy consumption, computational complexity...)*

Thank you!



fatronik
tecnalia