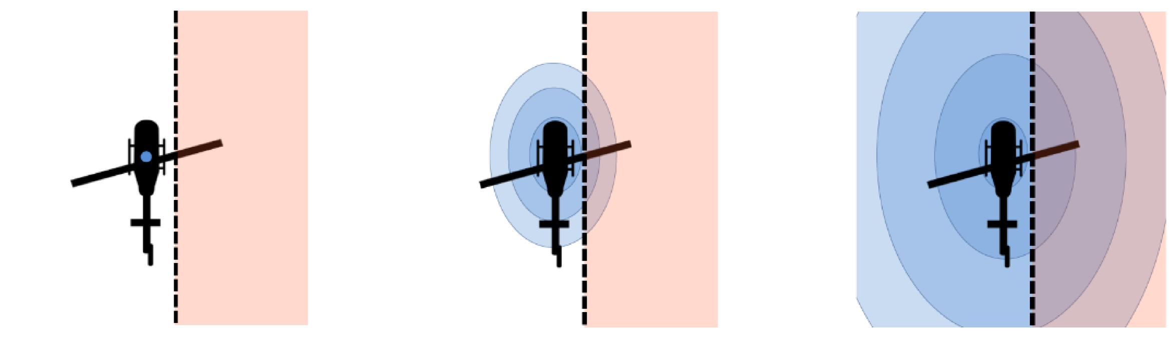
Spatio-Temporal Learning and Reasoning for Situation Awareness in Robotics

Mattias Tiger, Linköping University

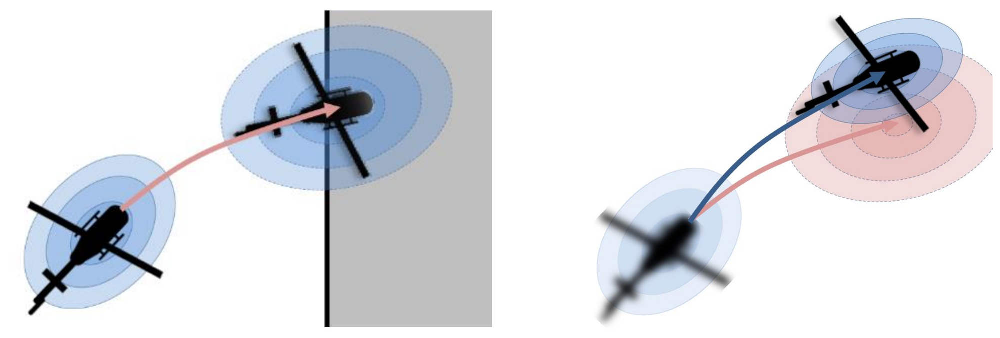
Awareness of entity motion is central to robot safety. This is true regardless of whether the robot is driving, flying or working alongside humans. It is important to recognize common motion patterns and to discover new ones. Furthermore, the important ability to anticipate what will happen next requires good predictions of future movement. It is also useful to be aware of how predictable other entities are, and to detect abnormal behaviors. Accurate awareness of the current situation is necessary for robust, efficient and safe decision making and control. Our focus is on trajectory-based approaches for continuous learning using probabilistic machine learning techniques, and on integrating probabilistic and logical stream reasoning.

Probabilistic and predictive stream reasoning for runtime verification

Logical constraints can be used to specify safe operating behaviors and assumptions about the environment. We have proposed P-MTL **[3]**, a metric temporal logic for stream reasoning (incremental reasoning over rapidly-changing information) over probabilistic and predicted states. This addresses two important problems in AI; integrating logical and probabilistic reasoning and integrating reasoning over observations and predictions. This allows a robot to explicitly reason about the uncertainty of the world, the expected change of the world and the quality of its observations and predictions.



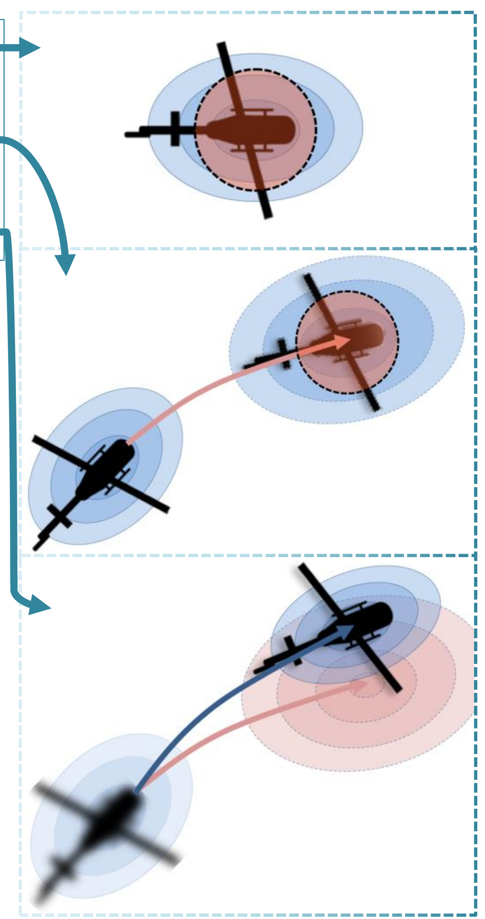
Reasoning over Uncertainty

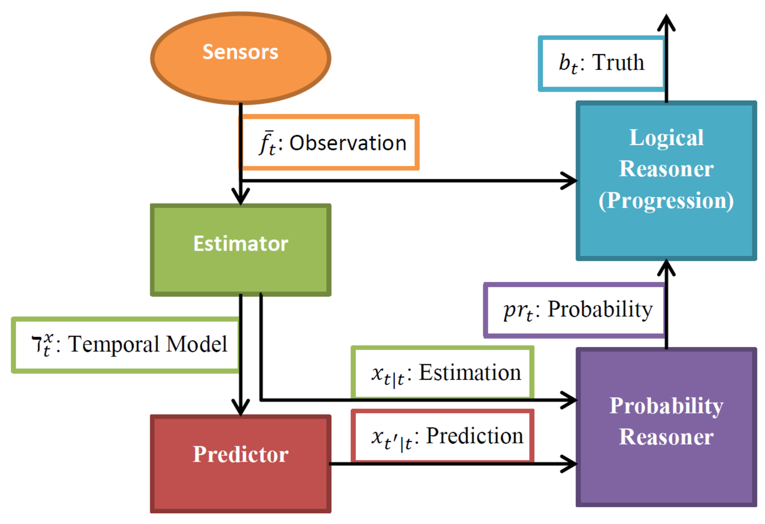


Reasoning over Predictions

Example formula for monitoring prerequisites for safe motion

$$\square\big(Pr(\mathit{insideRelative}(\bullet_{0|0}\text{Position}[\mathbf{uav1}], \mathbf{1mCircle})) > 0.99 \\ \wedge \\ Pr(\mathit{insideRelative}(\bullet_{3|0}\text{Position}[\mathbf{uav1}], \mathbf{1mCircle})) > 0.95 \\ \wedge \\ \bigcirc_3\ (\mathit{similarity}(\bullet_{0|0}\text{Position}[\mathbf{uav1}], \bullet_{0|-3}\text{Position}[\mathbf{uav1}]) > 0.5)\big)$$





Computational Environment

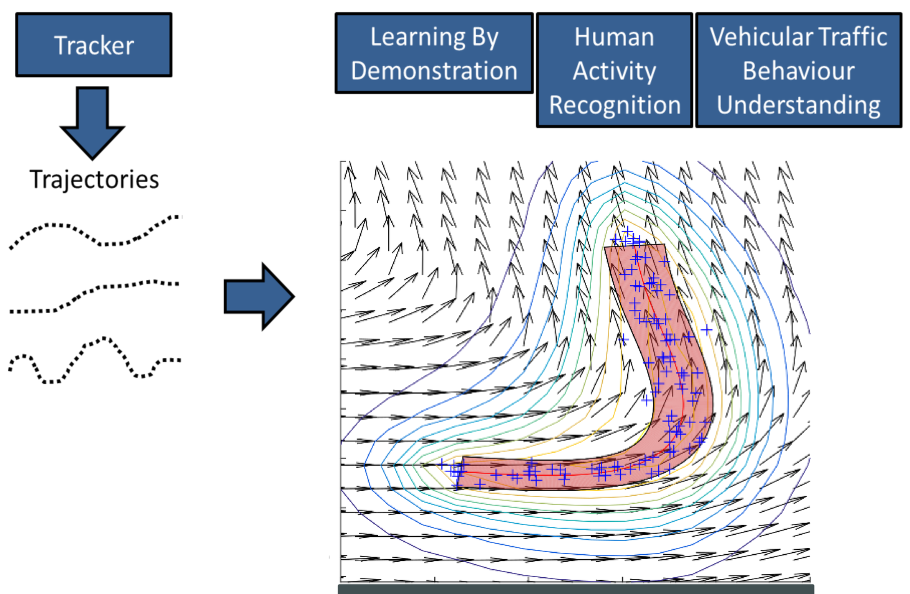
Learning and recognizing trajectory-based spatio-temporal activities such as motion patterns

We have proposed an online unsupervised framework that learns a probabilistic representation of observed spatio-temporal activities and their transitions from observed trajectories **[1]**. It can recognize activities and predict activity chains. Bayesian networks are used to model the transition graph and Gaussian processes (GP) to model atomic activities.

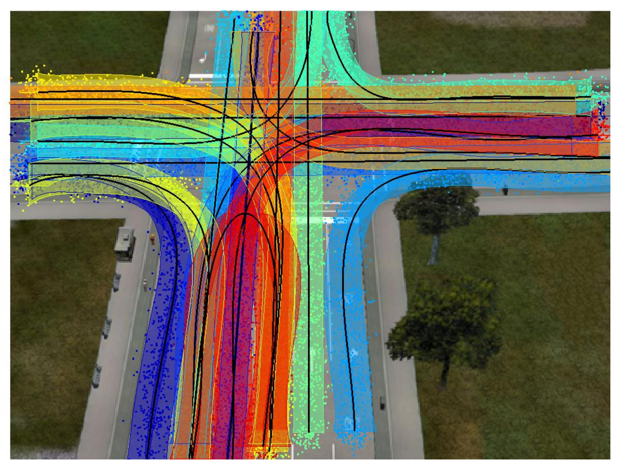
We have developed Gaussian process trajectory modelling tools for merging and separating serially and in parallel connected models. This includes an efficient technique for approximating mixtures-of-GPs and efficient online trajectory model learning using GPs **[2]**.

We have further proposed extended GP-based models and methodology to motion patterns in large-scale complex road structures **[4]**. Such topology necessitates sequential local models.

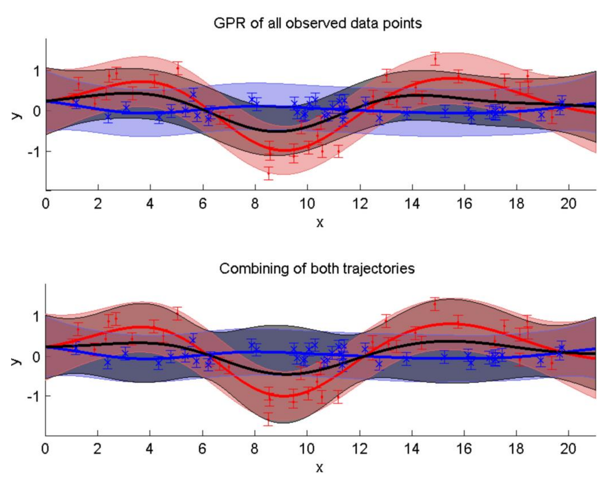
Motion pattern recognition includes:
clustering, classification, prediction, abnormality detection



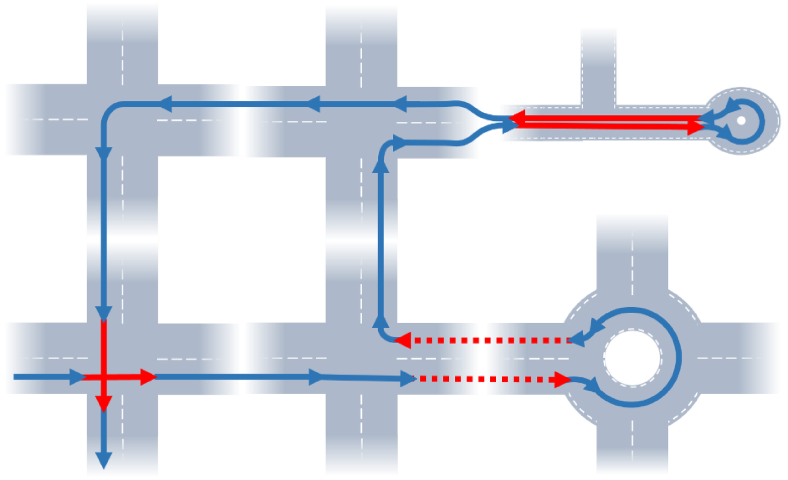
Motion Pattern (trajectory model)



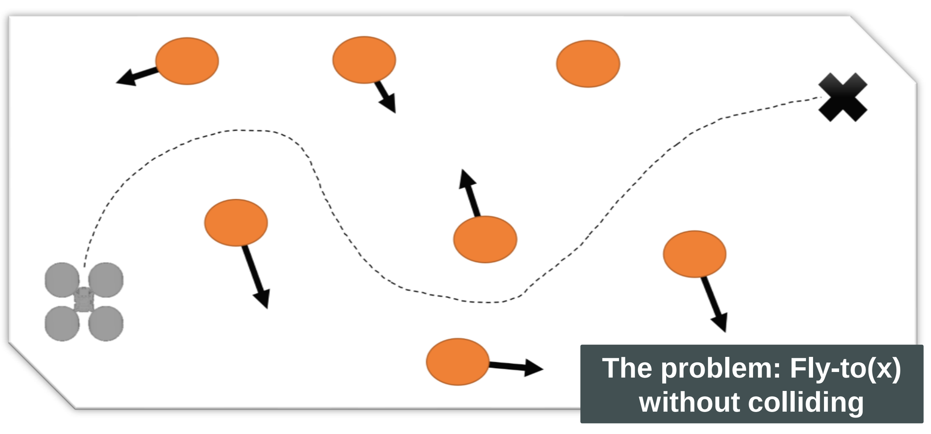
Common motion patterns in a crossing



Mixture-of-GPs approximation

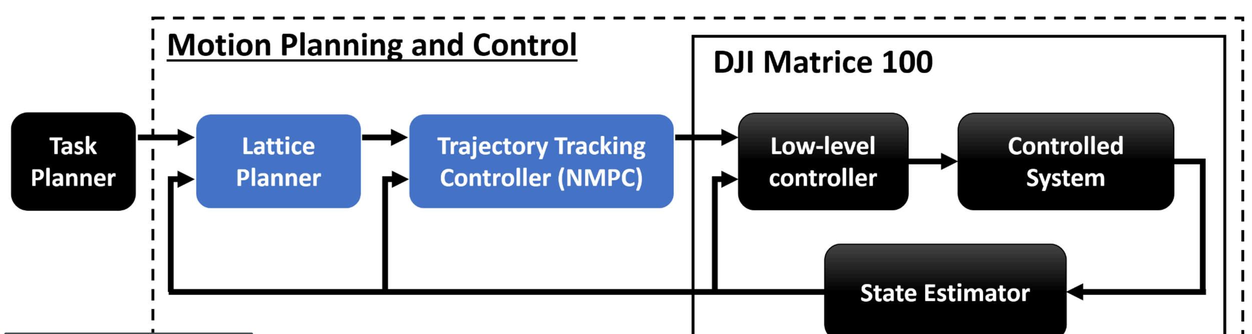


Large-scale modelling of normative motion patterns

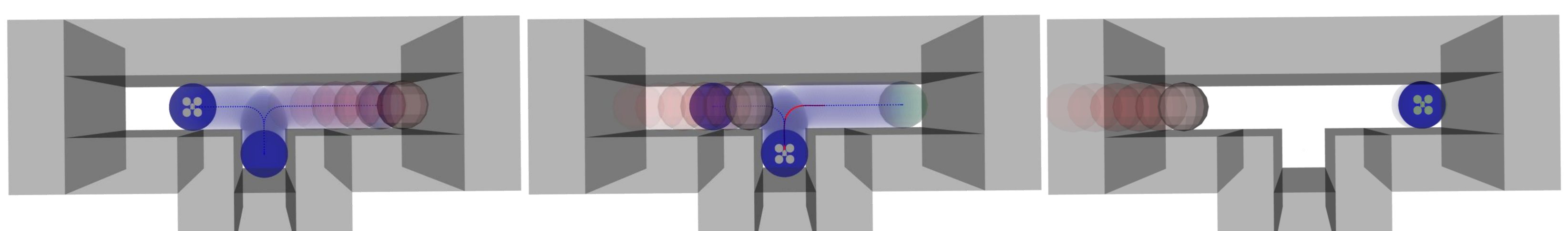


The problem: Fly-to(x) without colliding

DJI M100 UAV platform



System overview



Example where we plan to wait for an obstacle to pass us by

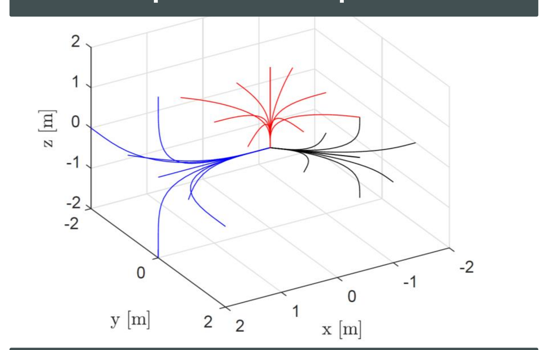
Trajectory-based motion planning and control in complex and dynamic environments

We have proposed **[5]** a principled solution to motion planning with dynamic obstacle avoidance by using a unified optimization-based motion planning and control architecture, where both layers use the system dynamics to generate and execute feasible trajectories in real-time.

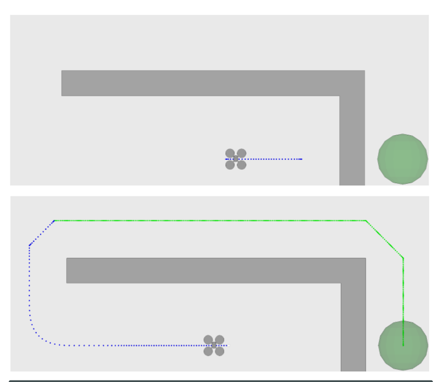
The plans are made with respect to time in several ways: In terms of predicted future movements of agents, time duration of actions and by the ability to plan to wait.

Motion primitives (full trajectories) are generated offline using numerical optimal control and then used in online graph-search for finding feasible and cost-efficient plans.

Example of motion primitives



Full trajectory planning and tracking
State: 3D position, 3D velocity, roll, pitch, thrust



Does not get stuck locally

[1] M. Tiger and F. Heintz, *Towards unsupervised learning, classification and prediction of activities in a stream-based framework*, SCAI 2015.
[2] M. Tiger and F. Heintz, *Online sparse Gaussian process regression for trajectory modeling*, FUSION 2015.
[3] M. Tiger and F. Heintz, *Stream Reasoning using Temporal Logic and Predictive Probabilistic State Models*, TIME 2016.
[4] M. Tiger and F. Heintz, *Gaussian Process Based Motion Pattern Recognition with Sequential Local Models*, IV 2018.
[5] O. Andersson, O. Ljungqvist, M. Tiger, D. Axehill, F. Heintz, *Receding-Horizon Lattice-Based Motion Planning with Dynamic Obstacle Avoidance*, CDC 2018

WALLENBERG AI, AUTONOMOUS SYSTEMS AND SOFTWARE PROGRAM