

POSTER CATALOGUE

WASP Winter Conference 2020

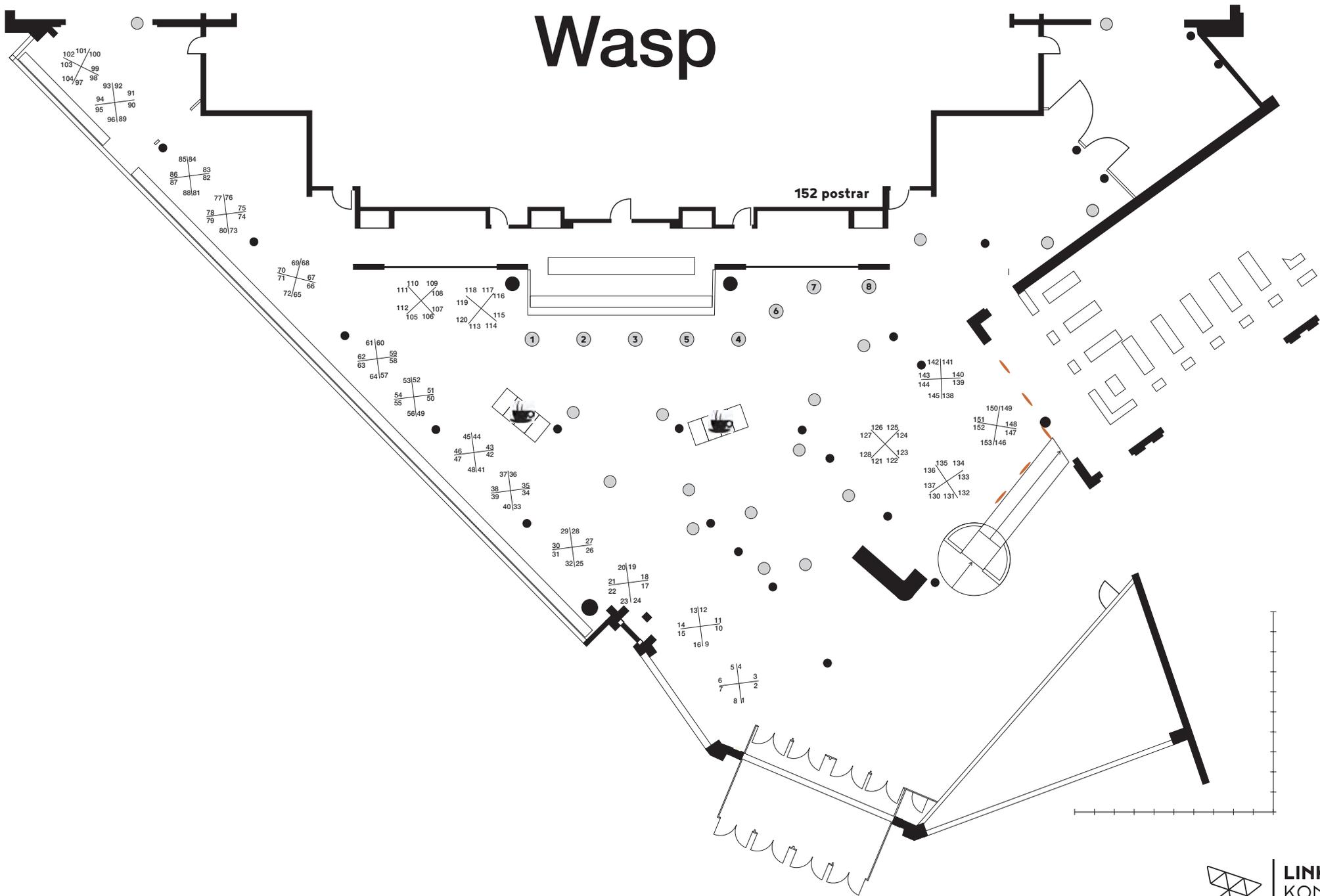
Poster Session 1

Tuesday 14 January 15.00-16.00

Poster no.	First name	Last name	Title of poster
1	David	Gillsjö	Efficient Merging of Maps and Detection of Changes
5	David	Mattos	Data-driven Continuous Evolution of Smart Systems
13	Emil	Brissman	Representation Learning and Visual Localization
17	Emilio	Jorge	Exploration and uncertainty in generative networks for reinforcement learning
21	Emir	Konuk	Towards more diverse and controllable generative models *
25	Erik	Jakobsson	Condition Monitoring for Autonomous Mining
29	Filip	Wikman	Deep Learning in Continuous Time *
33	Filippo	Vannella	Real-World Reinforcement Learning for mobile networks optimization
37	Francesca	Tombari	Homotopical decompositions of Vietoris-Rips complexes
41	Fredrik	Hagebring	Towards Planning and Scheduling of Unknown Systems using Simulation-based Active Learning *
45	Georg	Bökman	How to inject geometry into deep learning *
49	George	Osipov	Constraint Satisfaction Problems over Infinite Domain
53	Gizem	Çaylak	Deep Probabilistic Programming
57	Hanna	Hultin	Generative Models and Reinforcement Learning in Finance
61	Hannes	Eriksson	Epistemic Risk-Sensitive Reinforcement Learning *
65	Henrik	Ekström	Dynamics enhances the capacity of networks
69	-	-	-
73	Inês	de Miranda de Matos Lourenco	Learning Dynamical Systems
77	-	-	-
81	Jiexiong	Tang	Sparse2Dense: From direct sparse odometry to dense 3D reconstruction
85	Johan	Grönqvist	Adaptivity and Robustness with Machine Learning *
89	Johan	Fredin Haslum	Deep Learning for Drug Discovery
93	Johan	Wessen	Optimization Based Assembly Robot Programming *
97	Jonas	Nordlöf	Planning for minimum uncertainty
101	Jonas	Krook	Design and Formal Verification of a Safe Stop Supervisor for an Automated Vehicle
109	Jonatan	Vallin	Analysis of a Deep One Unit Residual Network
113	Jonathan	Styrud	Learning of control architectures for robot automation
117	Julian Alfredo	Mendez	Ethical Verification of AI Systems
121	Karl	Bengtsson Bernander	Robust learning of geometric equivariances
125	Kristoffer	Bergman	An Optimization-Based Receding Horizon Trajectory Planning Algorithm
129	Lars	Svensson	Traction Adaptive Motion Planning in Critical Situations
133	Linnea	Persson	Model predictive control for autonomous landings in a search and rescue scenario at sea
137	Magnus	Gyllenhammar	Models for efficient safety assurance for automated driving system
141	Martin	Andersson	Optimizing Neural Networks *
145	Matthias	Mayr	Parameterization of Behavior Trees for Industrial Assembly Tasks through Model-Based Reinforcement Learning *

*] Poster not available in this catalogue

Wasp





LUND
UNIVERSITY

EFFICIENT MERGING OF MAPS AND DETECTION OF CHANGES

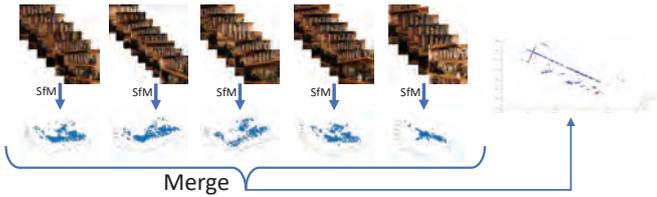
Gabrielle Flood, David Gillsjö, Anders Heyden, Kalle Åström

Centre for Mathematical Sciences, Lund University, Sweden



Motivation and Background

- We have developed a memory efficient method for merging of maps.
- It can also be used to detect changes in the map.
- The methods are tested on sensor data from both radio sensors and from cameras.



- Cheaper sensors have opened up for the possibility of using crowd sourced data.
- Individual estimates of a map can be obtained using structure form motion.
- A map is typically a set of 3D points, each consisting of a position and a feature vector.
- We denote the parameters which are optimized z , and the residuals r .
- The maximum likelihood estimate of z is found by minimizing the sum of squared residuals, $z^* = \text{argmin}_z r^T r$.

The Compressed Representation

When N separate maps from the same scene are obtained they can be merged to a single, more accurate map. We present a method which uses linearization to decrease the computational effort.

First, divide z into two parts: q which contains the parameters that exist in all the SfM sessions and s which are auxiliary parameters. Divide the Jacobian accordingly, s.t. J_a corresponds to q and J_b to s . Using $\Delta z = -(J^T)^{-1} J^T r$, we get

$$(J^T) \Delta z = \begin{bmatrix} U & W \\ W^T & V \end{bmatrix} \begin{bmatrix} \Delta q \\ \Delta s \end{bmatrix} = -J^T r,$$

which then gives

$$\frac{\partial s}{\partial q} = -V^{-1} W^T.$$

Using that $J = \partial r / \partial z$ we find

$$\Delta r = \begin{bmatrix} J_a & J_b \end{bmatrix} \begin{bmatrix} \Delta q \\ \Delta s \end{bmatrix} = \left(J_a + J_b \cdot \frac{\partial s}{\partial q} \right) \Delta q = J_q \Delta q.$$

Finally, expressing r as a function of Δq , and using a Taylor expansion give

$$r^T r \approx (r|_o + J_q|_o \Delta q)^T (r|_o + J_q|_o \Delta q) = r|_o^T r|_o + 2r|_o^T J_q|_o \Delta q + \Delta q^T J_q|_o^T J_q|_o \Delta q,$$

and this can be simplified to the compressed representation

$$r^T r \approx a + \Delta q^T R^T R \Delta q.$$

Merging Separate Maps

For the merge of two maps we get

$$a^{(*)} = \left((a^{(1)})^2 + (a^{(2)})^2 + (Mq^{(*)} - b)^T (Mq^{(*)} - b) \right)^{\frac{1}{2}},$$

with $M = [R^{(1)} \quad R^{(2)}]^T$ and $b = [R^{(1)}q^{(1)} \quad R^{(2)}q^{(2)}]^T$. Letting R^* come from a QR-decomposition of M gives a similar representation

$$(r^{(*)})^T r^{(*)} = (a^{(*)})^2 + (\Delta q^{(*)})^T (R^{(*)})^T R^{(*)} \Delta q^{(*)}.$$

Detection of Changes

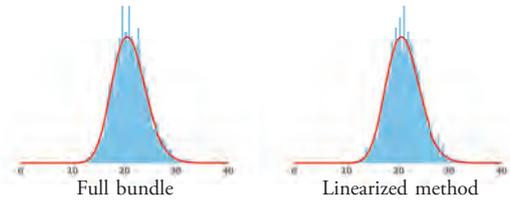
The representation can also be used to detect if changes have occurred in the map. When N map representations are merged, the differences in $a = r^T r$ will have the expected value

$$E[\bar{a}^2] = \sigma^2 (N - 1)(m\rho - \phi).$$

Furthermore, it can be seen as a sample from a Gamma distribution,

$$\bar{a} \sim \Gamma \left(\frac{1}{2\sigma^2}, (N - 1)(m\rho - \phi) \right).$$

A large deviation from this shows that the map has changed. The histograms below show \bar{a} computed for simulated data together with the assumed Gamma distribution.

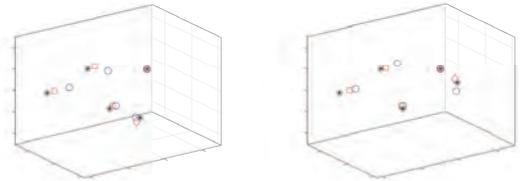


Results

The performance of the linearized method presented here was compared to the performance of one large bundle for merging of maps and to the use of a Kalman filter.

	n	100	1000	4000
Runtime [s]	Full bundle	0.19	3.8	54.7
	Linearized	$3.2 \cdot 10^{-4}$	$4.7 \cdot 10^{-4}$	$3.3 \cdot 10^{-4}$
	Kalman	$2.1 \cdot 10^{-4}$	$2.2 \cdot 10^{-4}$	$2.1 \cdot 10^{-4}$
$\ q^{(t)} - q \ $	Full bundle	0.11	$1.6 \cdot 10^{-2}$	$3.0 \cdot 10^{-3}$
	Linearized	0.11	$1.6 \cdot 10^{-2}$	$3.0 \cdot 10^{-3}$
	Kalman	0.12	$2.2 \cdot 10^{-2}$	$5.8 \cdot 10^{-3}$
$\frac{r^T r}{mn} = \frac{a^2}{mn}$	Full bundle	0.12	0.13	0.13
	Linearized	0.12	0.13	0.13

The method adjusts to changes faster than the Kalman filter in an experiment using UWB data.



The method can also be used to detect changes in SfM maps from image data.



Conclusions

- Our method is efficient and has a small memory footprint.
- It can be used to merge individual maps.
- It can also be used to detect changes in a map.

Data-driven Continuous Evolution of Smart Systems

David I. Mattos*, Jan Bosch*, Helena H. Olsson**

*Chalmers University of Technology, **Malmö University

Description:

The overall objective of this project is to analyze how to automate different types of experiments and how companies can support and optimize their systems through automated experiments. We explore this topic from the perspectives of the software architecture, the algorithms for the experiment execution and the experimentation process, and we focus on two main application domains: the online and the embedded systems domain.

Our optimization through automated experiments approach can be used in different configurations such as in offline and simulation, testing bed, hardware-in-the-loop, and deployed and live systems.

Background & Motivation

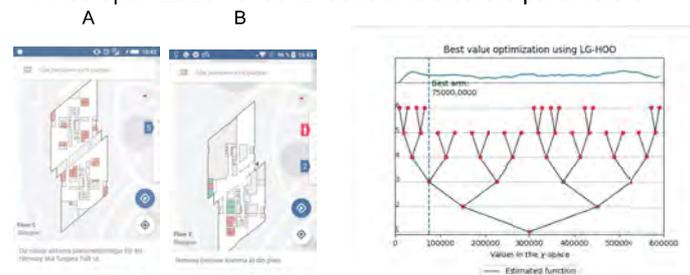
Online controlled experiments are one of the key techniques used to validate, optimize and incrementally deliver value in software systems. Automated experiments can help R&D organizations to lower the cost of each experiment iteration, run more experiments and run new experiments in scenarios that were previously not possible. In this context, we ask try to answer the questions:

- How do we optimize deployed complex systems to deliver more value to the customers and users?
- How to establish a trustworthy experimentation process?
- How do we architect an automated experimentation system?
- How can new machine learning algorithms help automated experiments?
- How can we run experiments in embedded systems and in systems that have high reliability requirements?

Sony Mobile Case Study

An integrated data-driven development of solution for running online controlled experiments and automated experiments for web, mobile, backend systems and distributed hardware.

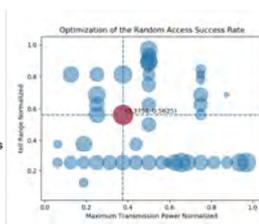
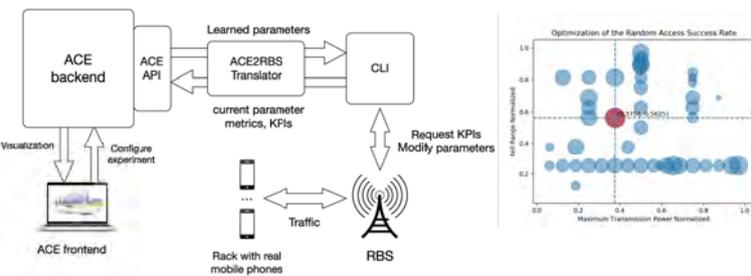
- A/B experiments for mobile user interface
- Automated experiments for live optimization of hyperparameter of algorithms
- Mixed optimization of continuous and discrete parameters



Ericsson Case Study

Automated optimization of software parameters in a LTE radio base station.

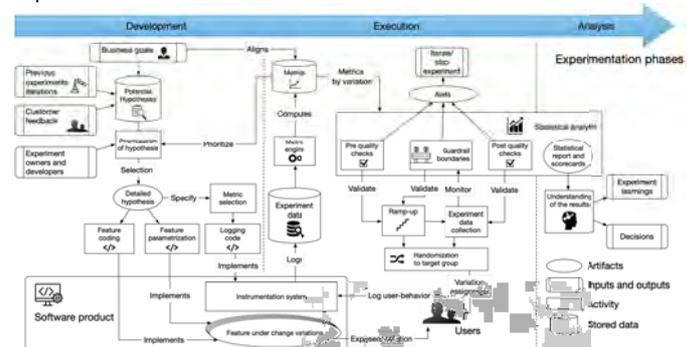
- How do we optimize a live radio base station to operators' business goals?
- How do we minimize the regret associated with the exploration of the parameter space?
- How do we add an optimization system to an existing radio base station without changing its software?



Microsoft Case Study

A trustworthy experimentation process for online controlled experiments.

- What are the activities of a trustworthy online controlled experimentation process?
- How do metrics evolve and influence the experiment and product evolution?



Project Publications

1. D. I. Mattos, J. Bosch, and H. H. Olsson, "Your System Gets Better Every Day You Use It: Towards Automated Continuous Experimentation," in the 43rd Euromicro Conference on Software Engineering and Advanced Applications (SEAA), 2017
2. D. I. Mattos, J. Bosch, and H. Holmström Olsson, "More for Less: Automated Experimentation in Software-Intensive Systems," in The 18th International Conference on Product-Focused Software Process Improvement, 2017
3. D. I. Mattos, J. Bosch, and H. H. Olsson, "Challenges and Strategies for Undertaking Continuous Experimentation to Embedded Systems: Industry and Research Perspectives," in Lecture Notes in Business Information Processing, vol. 314, 2018, pp. 277–292.
4. D. I. Mattos, E. Mårtensson, J. Bosch, and H. H. Olsson, "Optimization Experiments in the Continuous Space: The Limited Growth Optimistic Optimization Algorithm", in the Proceedings of the 10th International Symposium on Search-Based Software Engineering, 2018, pp. 1-15.
5. D. I. Mattos, P. Dmitriev, A. Fabijan, J. Bosch, and H. H. Olsson, "An Activity and Metric Model for Online Controlled Experiments" in the Proceedings of the 19th International Conference on Product-Focused Software Process Improvement, 2018, pp. 1-16
6. D. I. Mattos, J. Bosch, and H. H. Olsson, "Multi-armed bandits in the Wild: Common Pitfalls in Online Experiments" in submission to an international software engineering journal, 2018, pp.1-22.
7. D. I. Mattos, J. Bosch, H. H. Olsson, A. Dakkak, and K. Bergh, "Automated Optimization of Software Parameters in a Long Term Evolution Radio Base Station," in submission, 2018, pp. 1–8.
8. D. I. Mattos, E. Mårtensson, J. Bosch, and H. H. Olsson, "Optimization Experiments at Sony," to be submitted to an international software engineering journal, 2019

Representation Learning and Visual Localization

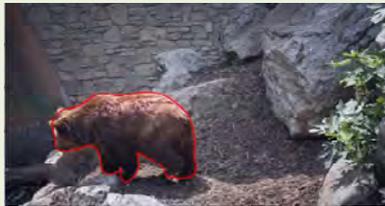
Emil Brissman^{1,2}, supervisors Michael Felsberg² and Per-Erik Forssén²
¹Saab Dynamics, ²Computer Vision Laboratory (CVL) Linköping University

Background & Motivation

- Vehicle, autonomous or semi-autonomous.
- Need vision during operation.
- Onboard resources, limited computational power.
- Try to avoid data overload.
- Mimic human visual attention.
- Choosing salient areas from the total visual scene.



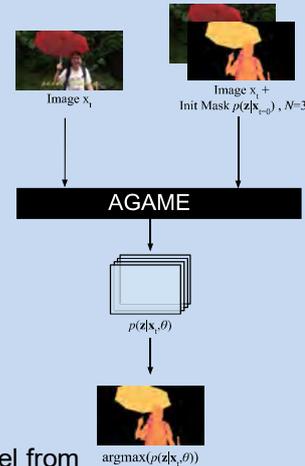
Multiple Object Segmentation



Contour. Image from DAVIS [2].

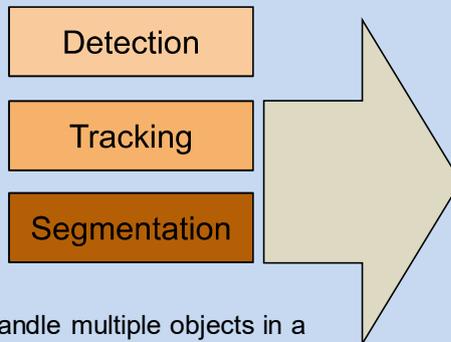
Methods & Preliminary Results

- So far focused on single object tracking in the image-plane.
- **Video Object Segmentation** is the task of tracking and segmenting one or multiple objects.
 - **Input:** image sequence and segmentation mask in frame #1, $t=1$
 - **Output:** segmentation masks for frames $t > 1$
 - **Assumptions:** ground truth segmentation given in first frame (semi-supervised)
 - **Challenge:** build an object model from the first frame, appearance changes, distractor objects (semantically similar).

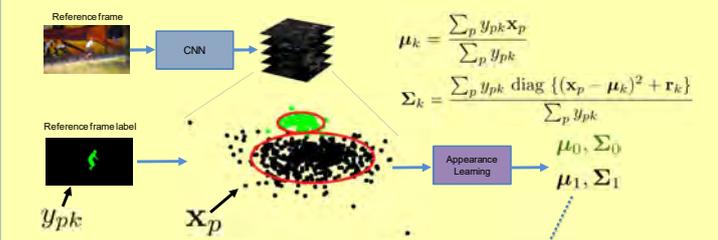


Research Goal & Questions

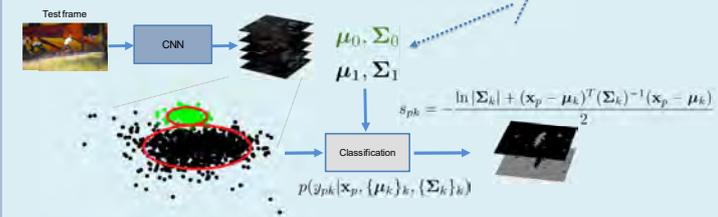
- Develop visual methods (CNN) for vehicle localization with single image input.
- What kind of image knowledge can be used?
- In order to estimate eg. risk of collision with other objects or ego-pose.
- For the segmentation task handle multiple objects in a single forward pass.



Generative Appearance Model: Learning

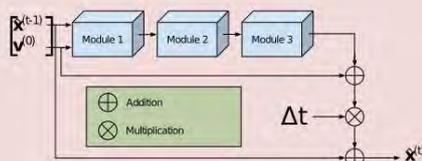
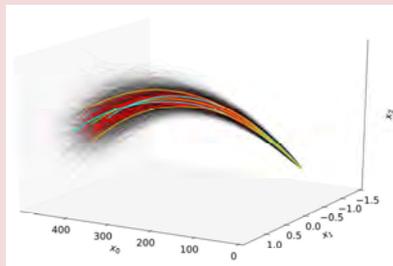


Generative Appearance Model: Classification



Uncertainty Estimation

- Work from other perspective.
- Learn a motion model from data.
- Synthetic data.
- Network uncertainty estimation.
- Estimating the predictive distribution of projectile paths.
- Predictive distribution estimated with MC-Dropout [3].



- Co-author on AGAME [1], presented at CVPR2019
 - The idea is to probabilistically model the feature distribution of the object and the background.
 - Feature vectors are explicitly classified via the probabilistic object model which is updated after each frame.

References

- [1] J. Johnander, M. Danelljan, E. Brissman, F. S. Khan, and M. Felsberg. A generative appearance model for end-to-end video object segmentation. In The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2019.
- [2] J. Pont-Tuset, F. Perazzi, S. Caelles, P. Arbelaez, A. Sorkine-Hornung, and L. Van Gool. The 2017 davis challenge on video object segmentation, 2017.
- [3] Gal and Z. Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. ICML, 2016.

Exploration and uncertainty in generative networks for reinforcement learning

Emilio Jorge

Chalmers University of Technology

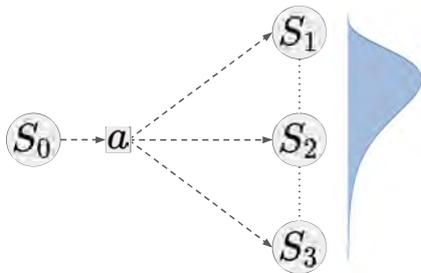
Why do we want know our uncertainty?

Safety

By avoiding what we do not know sufficiently about we can avoid behaviour that can lead to unsafe outcomes.

Exploration

In situations where safety is not an issue we can explore uncertain regions to search for potential rewarding outcomes that are still undiscovered.



What are good ways of representing uncertainty?

What do we want?

Generality. It is important for the models to be distributionally agnostic as transitions and rewards will very rarely belong to any friendly distributions with simple closed form.

Convergence. As more data is obtained it is important that the estimates converge to something that is close to the true model.

Efficiency. It needs to learn fast in terms of samples. It also needs to be able to scale reasonably with regards to computational complexity. It is especially useful if it has efficient ways of updating incrementally with data, as that is natural in reinforcement learning settings.

What might not be necessary?

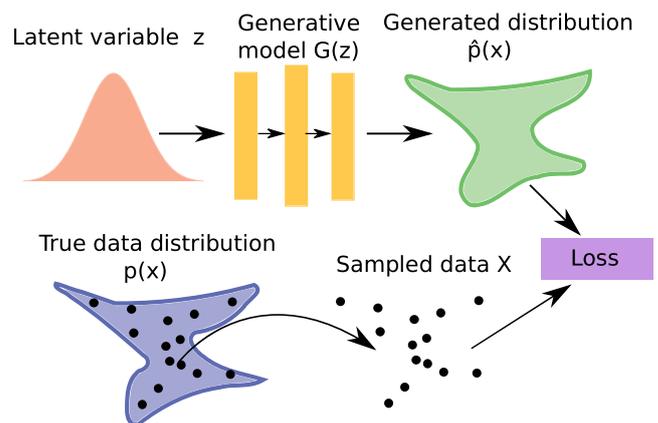
We might not always need explicit likelihoods (but it is nice to have), sampling is often enough. It might also be possible to relax some of the requirements mentioned above based on the desired application.

Generative models

Generative models allow us to approximately represent (and generate from) distribution $p(x)$ using sampled data. Popular methods are Generative Adversarial Networks (GAN) and Variational AutoEncoders (VAE).

How do they work?

They transform a sample from a latent variable z into a sample of the more complex distribution $\hat{p}(x)$ that aims to approximate the true distribution $p(x)$.



Research goals

We want to use generative models to reflect underlying uncertainty. By using this we aim to be able to create algorithms for reinforcement learning that can explore safer and/or in a more efficient way.



Any ideas or questions?

Come talk to me directly or email me at emilio.jorge@chalmers.se



CHALMERS
UNIVERSITY OF TECHNOLOGY

WASP | WALLENBERG AI,
AUTONOMOUS SYSTEMS
AND SOFTWARE PROGRAM

Condition Monitoring for Autonomous Mining

Erik Jakobsson

Erik Frisk, Robert Pettersson, Mattias Krylander
erik.jakobsson@epiroc.com



DESCRIPTION

Condition monitoring is the process of using measurement data from an asset to identify changes, which might indicate developing faults. Oil sampling, vibration analysis and thermography are common examples of techniques used. Prognostics extends condition monitoring to also take into account the change of condition over time, given a certain usage of the asset. The goal of prognostics is typically to determine the remaining useful life of the asset.

BACKGROUND & MOTIVATION



Mining often takes place in remote areas, and involves tasks both tedious and dangerous. It is also a predictive environment, with complete access control for personnel and vehicles. This makes it an ideal environment for using au-

tonomous machines. But how should one decide how to use the autonomous machines? Should one maximize the driving speed to maximize output? Should one drive safe to avoid breakdowns and failures? We claim that internal system awareness is key for a successful autonomous system. The vehicle needs to be able to monitor its internal state, predict its future state, and communicate this information to the planning system of the mine. Mining vehicles are characterized by relatively low volumes and high customization, which together with the difficulties of measuring and logging on mobile machines limits the usability of standard techniques.

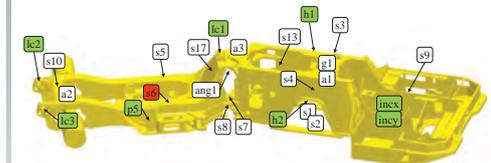
RESEARCH GOAL & QUESTION

A number of research questions define the area:

1. How can a rock drill be modelled and monitored in order to predict future failure? What data should be collected to maximize the information gathered without creating a too complex product?



2. Can a low number of non-dedicated sensors be used to monitor multiple components, for example points in a mine truck frame?

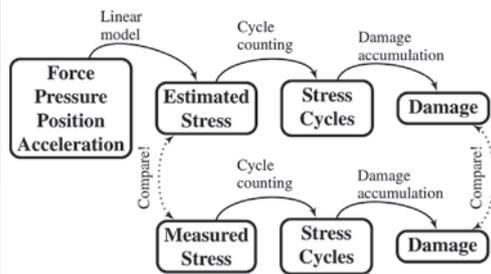


3. How should the collected measured data be used, together with models, in order to facilitate individual-based predictive maintenance for mining products?

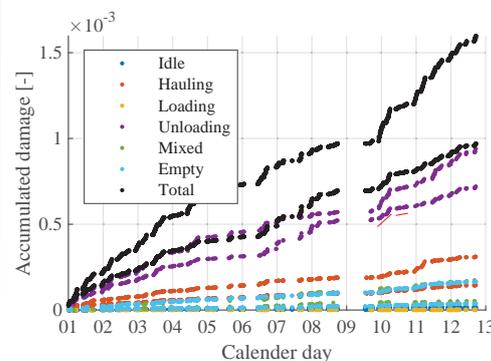
An important aspect of the research is to understand how predictive maintenance methods can be applied for products with a relatively low volume, high customization, and in a very harsh environment.

CONTRIBUTIONS

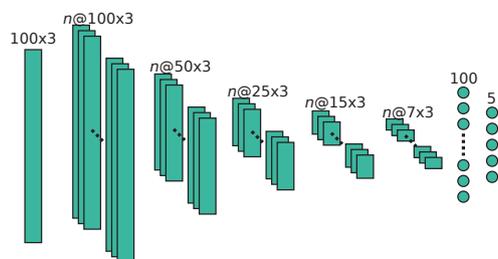
Mine Truck Fatigue Estimation On-board sensors and data driven models are used to estimate the condition of the frame of a mine truck, as seen in the following process:



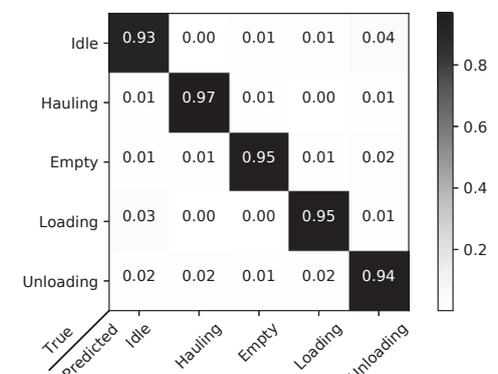
Using the model, differences in operating patterns can be seen. The figure below shows how one way of operating the vehicle causes almost 5 times as much damage.



Driving mode categorization The most recent work includes categorization of driving modes for a vehicle, where nothing but an accelerometer is available. The current approach is to use temporal convolution neural networks. The structure is inspired from research on Human Activity Recognition (HAR).



As seen below, the network is able to correctly classify tasks with roughly 95% accuracy.



ROADMAP & MILESTONES

- One conference paper accepted [1], two journal papers [2],[4] submitted and one conference paper submitted [3].
- Licentiate thesis presentation, December 2019.
- Ongoing and Future: Condition monitoring of hydraulic rock drills.

PUBLICATIONS

- [1] Jakobsson et al. "Data driven modeling and estimation of accumulated damage in mining vehicles using on-board sensors" published in *Proceedings of Annual Conference of the Prognostics and Health Management Society, St. Petersburg, Florida, USA, 2017*.
- [2] Jakobsson et al. "Fatigue Damage Monitoring and Prognostics for Mining Vehicles using Data Driven Models" submitted to the *International Journal of Prognostics and Health Management (IJPHM), 2019*.
- [3] Jakobsson et al. "Automated Usage Characterization of Mining Vehicles For Life Time Prediction" submitted to *IFAC World Congress, Berlin, 2020*.
- [4] Åstrand et al. "A System for Underground Road Condition Monitoring" submitted to *International Journal of Mining Science and Technology, 2019*.

Real-World Reinforcement Learning for mobile networks optimization

Filippo Vannella, Alexandre Proutiere, Jaeseong Jeong
vannella@kth.se, alepro@kth.se, jaeseong.jeong@ericsson.com



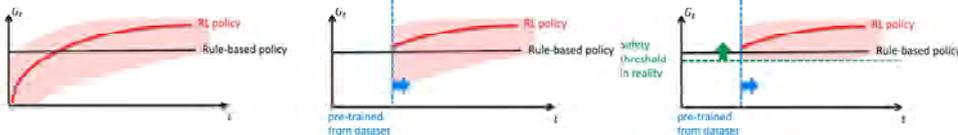
OVERVIEW

- Real-world Reinforcement Learning (RL):** In the context of RL an agent is required to interact with an unknown environment to collect samples of experience and learn from a feedback signal. Even though RL has proven its worth in a series of artificial domains, much of the research advances in RL are often hard to leverage in real-world systems due to many challenges:



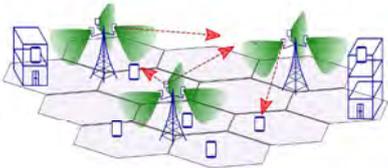
- Legacy Telecom automation:** Traditionally, automation in mobile networks relies on rule-based methods governed by heuristic domain knowledge. These technologies are however limited to the range of the given tasks and rather hard to be generalized to more complex environment where the observations are in a high dimensional space and include random noise in real world.
- Research goal:** actualize RL agents to successfully automate key use cases real-world mobile networks addressing challenges of Real-World RL.

METHOD: OFFLINE OFF-POLICY LEARNING & SAFE EXPLORATION



1. *RL uncontrolled exploration* inevitably degrades system performance VS rule-based policy having constant sub-optimal performance level.
2. *Pre-training* (offline) from \mathcal{D}_{π_0} to initialize policy π having performance at least as good as π_0 .
3. *Safe exploration* (online) allow to learn a policy π satisfyign a *policy safety constraint*.

USE CASE: REMOTE ELECTRICAL TILT (RET) OPTIMIZATION



- Remote Electrical Tilt (RET):** automation of antenna tilt angle in 4G LTE networks for Coverage and Capacity Optimization (CCO).

Contextual bandit setting:

1. Observe *context* $x_i \stackrel{i.i.d.}{\sim} \Pr(\mathcal{X})$. $x_i = [c_i, q_i]$ aggregated feature of *coverage* c_i and *quality* q_i Key Performance Indicators (KPIs).
2. Choose *action* $y_i \in \mathcal{Y} = \{-1, 0, 1\}$: down-tilt, no change, uptilt antenna.
3. Experience *loss* $\delta_i = \delta(x_i, y_i) = \max\{c_{i+1}, q_{i+1}\} - \max\{c_i, q_i\}$ hadcrafted measure of context variation.

- Baseline Dataset:** $\mathcal{D}_{\pi_0} = \{(x_i, y_i, \delta_i)\}_{i=1}^N$ derived from rule-based expert baseline policy π_0 .

- Objective:** Derive new policy $\pi \in \Pi$ from \mathcal{D}_{π_0} minimizing the expected risk:

$$R(\pi) = \mathbb{E}_{x \sim \Pr(\mathcal{X})} \mathbb{E}_{y \sim \pi(\cdot|x)} [\delta(x, y)]$$

Challenges:

1. *Sampling bias:* actions favored by π_0 will be over-represented in \mathcal{D}_{π_0} .
2. *Incompleteness:* feedback $\delta(x_i, y)$ for $y \neq y_i$ not chosen by π_0 are not available.

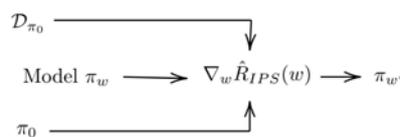
- Solution:** Inverse Propensity Score (IPS) risk.

$$R(\pi) = \mathbb{E}_{\pi}[\delta(x, y)] = \mathbb{E}_{\pi_b} \left[\delta(x, y) \frac{\pi(y|x)}{\pi_0(y|x)} \right]$$

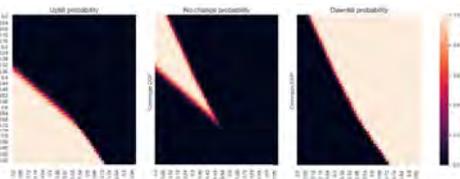
IPS corrects for distribution mismatch between *baseline policy* π_0 and *target policy* π .

- Training objective:** Monte-Carlo IPS risk estimator training objective on Artificial Neural Network (ANN) model $\pi_w(y|x) : \mathcal{X} \rightarrow \Pr(\mathcal{Y})$

$$\pi_{w^*} = \arg \min_w \frac{1}{N} \sum_{i=1}^N \delta_i \frac{\pi_w(y_i|x_i)}{\pi_0(y_i|x_i)}$$



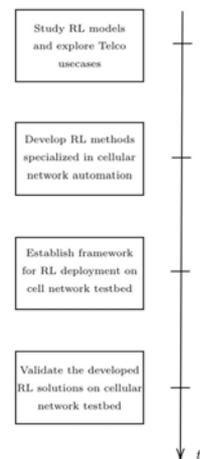
	$\hat{R}(\pi_0) \simeq -0.17$			$\hat{R}(\pi_w) \simeq -0.2$		
	$c 0$	$c 1$	$c 2$	$c 0$	$c 1$	$c 2$
$q 0$	0.08	0.02	-0.09	0.08	0.08	-0.2
$q 1$	0	0.06	-0.02	0.04	0.03	-0.07
$q 2$	-0.12	-0.09	-0.03	-0.25	-0.09	-0.22



RESEARCH GOAL & QUESTIONS

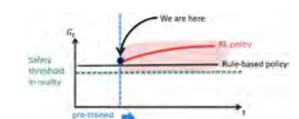
- How to address the ubiquitous reality gap in RL algorithm for the actual deployment of policies in real-life use cases? How this relates to RL algorithms specialized to the automation in cellular networks?
- What are the optimal techniques to pre-train a policy offline from observation collected to another policy (offline off-policy learning) and how to iteratively reduce the risk of the pre-trained policy?
- How to design safety constraint allowing online deployment of the pre-trained policy and safe exploration of the remaining state space?

MILESTONES



CONCLUSIONS & FUTURE WORK

- Promising generalization performance of offline off-policy Learning.



- Study and design safety constraints for RL online deployment.
- Extend IPS training to Markov Decision Process scenario.

1 Introduction

The use of algebraic topology is rapidly growing in understanding data. The general pipeline of TDA can be summarized by the following:

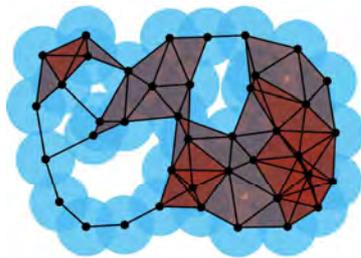
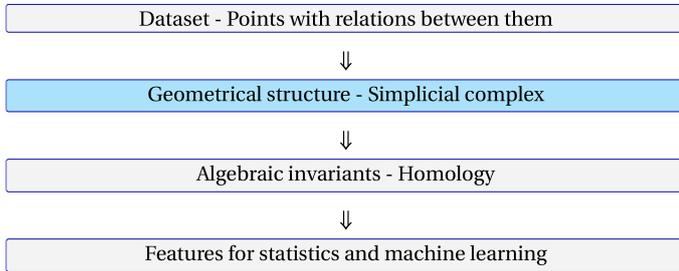


Figure 1: An example of a Vietoris-Rips complex, given some $r > 0$.

The computational challenges motivates the following type of question: given a decomposition of our data set $Z = X \cup Y$, what information can we recover about the Vietoris-Rips complex of Z from the component Vietoris-Rips complexes?

2 A general approach

Let K be a simplicial complex, $K_0 = X \cup Y$ the set of its vertices and $A = X \cap Y$. Let $K_X = K \cap X$ and $K_Y = K \cap Y$. One can easily notice that the union of K_X and K_Y does not give the initial simplicial complex K . A natural question that might arise is whether the following inclusion is a homotopy equivalence or not

$$K_X \cup K_Y \hookrightarrow K.$$



Figure 2: Example of a simplicial complex with high complexity. (Image courtesy of the authors of arXiv:1608.03520)

A special case of this problem occurs when a pseudo-metric space (Z, d) is considered. Fixing $r > 0$ and a covering of Z consisting in two subspaces X and Y , we get the inclusion

$$VR_r(X) \cup VR_r(Y) \hookrightarrow VR_r(Z).$$

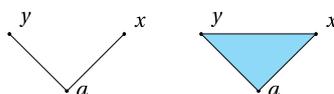


Figure 3: The two figures show a simplicial complex K (on the right) and $K_X \cup K_Y$ (on the left), where $X = \{x, a\}$ and $Y = \{y, a\}$.

3 Main result

We define the obstruction complex:

$$F(\sigma, A) := \{\mu \subset A \mid \mu \cup \sigma \in K\}.$$

Theorem. Let \mathcal{C} be a closed collection of simplicial sets. If, for every σ in $\{\sigma \in K \mid \sigma \cap X \neq \emptyset \text{ and } \sigma \cap Y \neq \emptyset \text{ and } \sigma \cap A = \emptyset\}$, the simplicial complex $F(\sigma, A)$ satisfies \mathcal{C} , then the homotopy fibers of $K_X \cup K_Y \subset K$ also satisfy \mathcal{C} .

Corollary. If, for every σ as above, the simplicial complex $F(\sigma, A)$ is contractible, then $K_X \cup K_Y \subset K$ is a weak equivalence.

We get a long exact sequence in the case when adding one vertex:

$$H_n(F(x, A)) \rightarrow H_n(K_A) \rightarrow H_n(K) \rightarrow H_{n-1}(F(x, A)) \rightarrow H_{n-1}(K_A)$$

and another when adding two vertices:

$$H_n(\Sigma F(x, y, A)) \rightarrow H_n(K_X \cup K_Y) \rightarrow H_n(K) \rightarrow H_{n-1}(\Sigma F(x, y, A)) \rightarrow H_{n-1}(K_X \cup K_Y).$$

These sequences give information about the global homology of K with respect to local information.

4 Examples

Consider the metric space $Z = \{x_1, x_2, a_1, a_2, y\}$, with the metric such that every two points of Z has distance 1 except for x_1, a_2 and x_2, a_1 having distance 1.1. Let $X = \{x_1, x_2, a_1, a_2\}$, $Y = \{y, a_1, a_2\}$ be a cover for Z . We can easily see that $VR_1(X) \cup VR_1(Y)$ has the homotopy type of S^1 , while $VR_1(Z)$ is contractible. This is due to the fact that $F(\sigma, A)$ is empty, hence non-contractible, when σ is the 2-simplex with vertices x_1, x_2 and y .

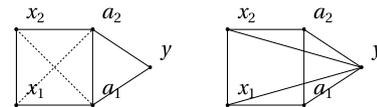


Figure 4: $K_X \cup K_Y$ on the left and K on the right. Notice that all the triangles in this example are filled, because K is a clique complex.

The following picture shows an example of a decomposition of $Z = \{x_1, x_2, y_1, y_2, a_{11}, a_{12}, a_{21}, a_{22}\}$ that has the same homology as the total simplicial complex up to degree 2, but different H_3 .

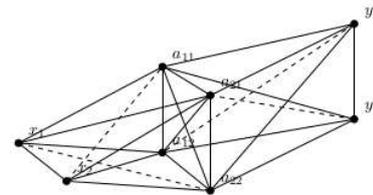


Figure 5: The figure represents a 2-dimensional visualization of the Vietoris-Rips complex $VR_r(X) \cup VR_r(Y)$. $VR_r(X \cup Y)$ is obtained by the above simplicial complex adding the simplex $\{x_1, x_2, y_1, y_2\}$

The metric is given by:

$$\begin{aligned} d(a_{11}, a_{21}) &= d(a_{11}, a_{12}) = d(a_{21}, a_{22}) = d(a_{12}, a_{22}) = 4, \\ d(a_{11}, a_{22}) &= d(a_{12}, a_{21}) = 6, \\ d(x_1, a_{11}) &= d(y_1, a_{21}) = d(x_2, a_{22}) = d(y_2, a_{12}) = 3, \\ d(x_1, a_{12}) &= d(y_1, a_{11}) = d(x_2, a_{21}) = d(y_2, a_{22}) = 5, \\ d(x_1, a_{21}) &= d(y_1, a_{22}) = d(x_2, a_{12}) = d(y_2, a_{11}) = 7, \\ d(x_1, a_{22}) &= d(y_1, a_{12}) = d(x_2, a_{11}) = d(y_2, a_{21}) = 9, \\ d(x_1, x_2) &= d(y_1, y_2) = 6, \\ d(x_1, y_1) &= d(x_1, y_2) = d(x_2, y_1) = d(x_2, y_2) = 8. \end{aligned}$$

As we have already noticed, the study of this problem for Vietoris-Rips complexes is actually a consequence of the same problem stated for generic simplicial complexes. Analogously, the conditions that we put on a metric space are just a translation of hypothesis on simplicial complexes.

5 References

- [1] W. Chachólski, A. Jin, M. Scolamiero, and F. Tombari. Homotopical decompositions of Vietoris-Rips complexes. Forthcoming, 2019.
- [2] Adamaszek et. al. On Homotopy Types of Vietoris-Rips Complexes of Metric Gluings. Proceedings of the 34th Symposium on Computational Geometry (2018), 3:1-3:15.

Constraint Satisfaction Problems over Infinite Domains

George Osipov

Background

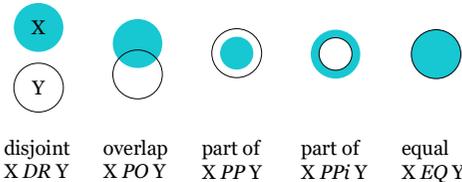
Constraint satisfaction problems (CSPs) appear in numerous domains of computer science: artificial intelligence, machine learning, computer vision, computational biology, etc. An instance of a CSP consists of a set of variables, a set of values that can be assigned to these variables (known as the *domain*) and a set of constraints - logical predicates over variables that impose restrictions on the possible assignments of values. To solve such a problem, every variable needs to be assigned a value so that all the constraints are satisfied.

The framework of constraint satisfaction allows to express many different problems. Despite its generality, CSP exhibits structural properties that can be exploited to determine its complexity. For CSPs over finite domain the recently proven dichotomy conjecture [1,2] states that all problems are either in P or NP-complete. Moreover, complexity of any finite-domain CSP can be checked by investigating the set of its solutions and the symmetries of this set. This line of research is usually referred to as algebraic approach to CSPs.

The dichotomy conjecture no longer holds for CSPs over infinite domain [3] and algebraic methods do not apply directly. Furthermore, problems no longer admit a brute-force algorithm that enumerates all solutions, so there is no exponential upper bound in the number of variables in general case. Only superexponential algorithms are known for some CSPs. Improvement is unlikely in some cases based on assumptions in computational complexity such as Exponential Time Hypothesis [4]. But in other cases, there is a large gap between the best known upper and lower bounds. Many problems that arise in the context of artificial intelligence when spatial and temporal reasoning is involved fall into this uncharted area. We are interested in developing mathematical tools to investigate them.

Spatial Reasoning

Region Connection Calculus (RCC) [5] is designed for qualitative spatial reasoning. Here we will concentrate on its simplest version - RCC-5. There are 5 basic relations in RCC-5:



A general constraint satisfaction problem for RCC-5 consists of a set of variables representing regions and constraints which are *disjunctions* of basic relations. To formalize this problem independently of the geometry of regions, one can think of variables ranging over subsets of natural numbers and relations being the corresponding set operations. Consider an example:

$$\begin{aligned} X \{PP, PPI\} Y \\ Y \{PP, PPI\} Z \\ X \{DR\} Z \end{aligned}$$

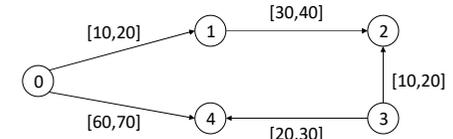
A possible solution here is $X = \{1\}$, $Y = \{1,2\}$, $Z = \{2\}$.

Best known lower bound: $O^*(c^n)$
Best known upper bound: $O^*(n!)$

Here n is the number of variables, c is some constant value. Notation O^* hides polynomial factors in n .

Temporal Reasoning

Temporal Networks proposed by Dechter et al. [6] are used to express temporal constraints. Time points are represented by vertices of a graph. Directed edges $X \rightarrow Y$ are labeled with intervals $[a, b]$ and impose the constraint $a \leq Y - X \leq b$. Here is an example of a temporal network:



Consistency of a temporal network can be checked in polynomial time in the size of the graph. However, a more general version of the problem which allows for disjunctive constraints admits a superexponential lower bound.

Best known lower bound: $2^{O(n \log n)}$
Best known upper bound: $2^{O(n^2)}$

References

- [1] Bulatov (2017) A dichotomy theorem for nonuniform CSPs.
- [2] Zhuk (2017) A proof of CSP dichotomy conjecture.
- [3] Jonsson P., Lagerkvist, V., Nordh, G. (2015) Constructing NP-intermediate problems by blowing holes with parameters of various properties.
- [4] Impagliazzo, R., Paturi, R., Zane, F. (2001) Which problems have strongly exponential complexity?
- [5] Cohn, A. G., Bennett, B., Gooday, J., Gotts, N. M. (1997) Qualitative spatial representation and reasoning with the region connection calculus.
- [6] Dechter, R., Meiri, I., Pearl, J. (1991). Temporal constraint networks.
- [7] Bodirsky, M., Jonsson, P. (2017) A model-theoretic view on qualitative constraint reasoning.

Deep Probabilistic Programming

Gizem Çaylak

Supervisor: David Broman

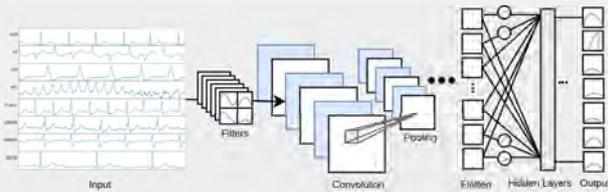
KTH Royal Institute of Technology, Sweden

WASP WALLBERG AI AUTONOMOUS SYSTEMS AND SOFTWARE PROGRAM

This work was partially supported by the Wallenberg AI, Autonomous Systems and Software Program (WASP) funded by the Knut and Alice Wallenberg Foundation

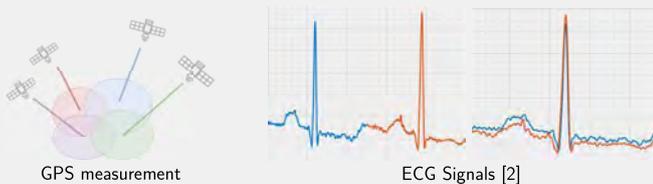
Introduction

- ▶ **Deep Learning** methods can learn very intricate functions by discovering multiple layers of representation.
- ▶ Many deep learning frameworks have been important in health care and provided many valuable insights, but most do not take advantage of probability theory to represent **uncertainty**.
- ▶ Detecting ECG abnormalities [1]
- ▶ For its use in a clinical setting, there is a need for propagation of uncertainty through the pipeline of classification for **risk assessment**.



Probabilistic Programming Languages

- ▶ **Uncertainty** comes in many forms.



- ▶ To handle varying forms of uncertainty, a fundamental approach is to use a probabilistic approach based on the probability theory which is a **mathematical language** to model uncertainty.

$$P(A | B) = \frac{P(B | A)P(A)}{P(B)}$$

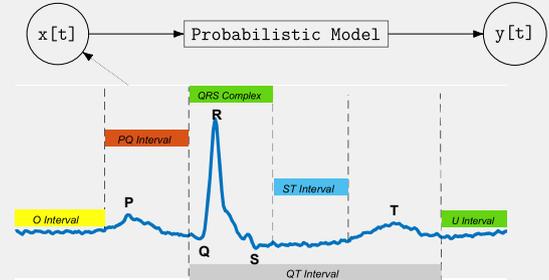
- ▶ The probabilistic programming languages aim to develop probabilistic systems which separate model and inference.
- ▶ The **focus** on this project will be on using a probabilistic programming language, e.g. **Pyro** [3], together with developing new neural networks on sequential data.

Motivation

- ▶ PPLs increase **expressiveness** via separating model and inference.
Benefit: The user can see the model clearly and avoid complex inference code.
- ▶ PPLs construct **confidence intervals** on estimates.
Benefit: In clinical settings where risk-assessment is critical, it is important to indicate how *confident* we are about our estimation.

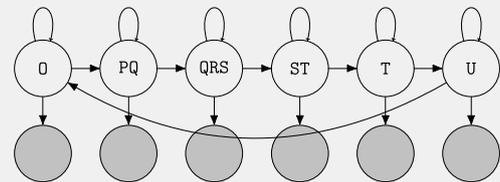
Research Problem

- ▶ Designing a model on sequential data with Pyro which enables probabilistic modeling to represent uncertainty.



Pyro Example

- ▶ Hidden markov model to represent full ECG segmentation.



- ▶ Supervised HMM model in Pyro:

```
def model(self, mini_batch, mini_batch_reversed, mini_batch_mask,
          mini_batch_seq_lengths, a_f=1.0):
    T_max = mini_batch.size(1) #the number of time steps
    pyro.module("dmm", self)
    # set x_prev = x_0 recursive conditioning in p(x_t | x_{t-1})
    x_prev = self.x_0.expand(mini_batch.size(0), self.x_0.size(0))
    # plate: each datapoint is conditionally independent of others
    with pyro.plate("x_minibatch", len(mini_batch)):
        for t in range(1, T_max + 1): # sample x and observed y's
            transition_prob = self.trans(x_prev) # p(x_t | x_{t-1})
            withoutine.scale(None, a_f):
                x_t = pyro.sample("x_%d" % t,
                                  dist.Categorical(transition_prob)
                                  .mask(mini_batch_mask[:, t - 1:t]).to_event(1))
            y_loc, y_scale = self.emitter(x_t)
            # normal distribution p(y_t|x_t)
            pyro.sample("obs_y_%d" % t, dist.Normal(y_loc, y_scale)
                       .mask(mini_batch_mask[:, t - 1:t])
                       .to_event(1), obs=mini_batch[:, t - 1, :])
            x_prev = x_t
```

Project

- ▶ This project is part of a collaboration work, **Deep Probabilistic Neural Networks for Survival Analysis**, between Gizem Çaylak and Assoc. Prof. David Broman at KTH Royal Institute of Technology and Prof. Thomas Schön and Asst. Prof. Niklas Wahlström and Daniel Gedon at Uppsala University.
- ▶ This work was partially supported by the Wallenberg AI, Autonomous Systems and Software Program (WASP) funded by the Knut and Alice Wallenberg Foundation.

References

1. A. H. Ribeiro, M. H. Ribeiro, G. M. M. Paixão, D. M. de Oliveira, P. R. Gomes, J. A. Canazart, M. P. S. Ferreira, C. R. Andersson, P. W. Macfarlane, W. M. Jr., T. B. Schön, and A. L. P. Ribeiro, "Automatic diagnosis of the short-duration 12-lead ECG using a deep neural network: the CODE study," *CoRR*, vol. abs/1904.01949, 2019.
2. P. Plawiak, "Ecg signals (1000 fragments)," <http://dx.doi.org/10.17632/7dybx7wyfn.3>, 2017.
3. E. Bingham, J. P. Chen, M. Jankowiak, F. Obermeyer, N. Pradhan, T. Karaletsos, R. Singh, P. Szerlip, P. Horsfall, and N. D. Goodman, "Pyro: Deep Universal Probabilistic Programming," *Journal of Machine Learning Research*, 2018.

Hanna Hultin, KTH/SEB, hhultin@kth.se

Advisor: Henrik Hult, KTH

Co-Advisor: Alexandre Proutiere, KTH

Industrial Advisor: Patrik Karlsson, SEB

Optimal Order Execution

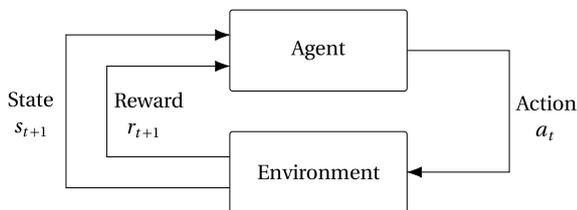
A common task within financial trading, is trying to buy or sell a large amount of assets at the best price available. However, it is typically not possible to fill the whole amount with just one big order, instead the large order is split into smaller suborders executed at different times.

How to split the large order into smaller ones is an optimal control problem where we have to take the following into consideration:

- **Cost:** Not enough available volume at best price
- **Market Impact:** Our orders will typically push the price in the wrong direction
- **Risk:** Independently of our own orders, something unrelated might happen that moves the price

Can we use machine learning to improve our order execution?

Reinforcement Learning for Order Execution

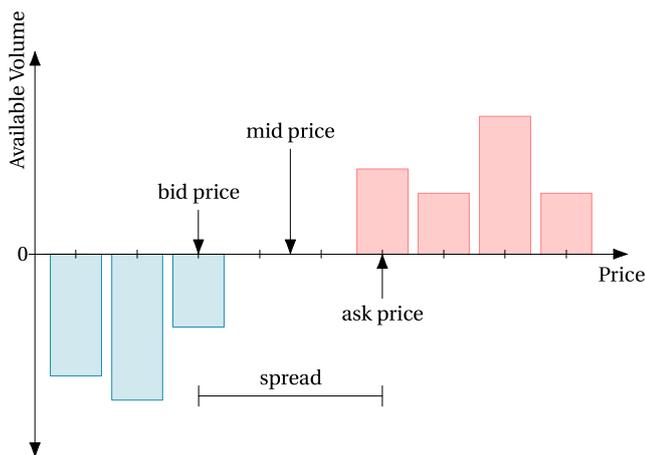


Problem:

Need to train the agent offline, but do not have access to realistic data which models market impact

Solution: Use generative models that learn from real data

Limit Order Book



- Limit Order Book (LOB) keeps track of all outstanding orders
- Can view LOB as high-dimensional time series, but usually very sparse
- Typically model the LOB with stochastic processes, but challenging identifying more complex dynamics
- Machine learning methods suitable for finding flexible and complex representations

Generative Models for Order Books

- First model based on Recurrent Neural Networks
- Models each change in the order book as an event
- Each event is specified by an **event type**, the **price level** of the change and the **volume** of the change
- The model is trained by maximizing the likelihood of training data

Modelling of Order Book Events

$$\textcircled{1} \mathbb{P}(e_t | h_t, OB_t)$$

- e_t - event type: Market Order/Limit Order/Cancellation
- h_t - memory of RNN
- OB_t - current state of order book

$$\textcircled{2} \mathbb{P}(l_t | h_t, OB_t, e_t)$$

- l_t - price level of the change

$$\textcircled{3} \mathbb{P}(v_t | h_t, OB_t, e_t, l_t)$$

- v_t - volume of the change

$$\textcircled{\times} \text{Joint probability:}$$

$$\mathbb{P}(e_t, l_t, v_t | h_t, OB_t) = \mathbb{P}(e_t | h_t, OB_t) \mathbb{P}(l_t | h_t, OB_t, e_t) \mathbb{P}(v_t | h_t, OB_t, e_t, l_t)$$

Evaluation of Model

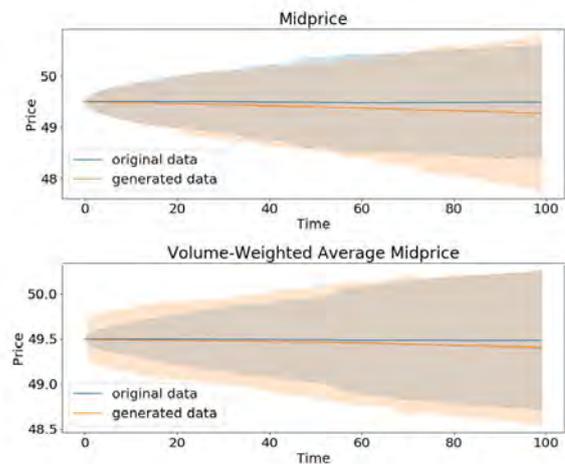


Figure 1: Mean and standard deviation for midprice and volume weighted average midprice

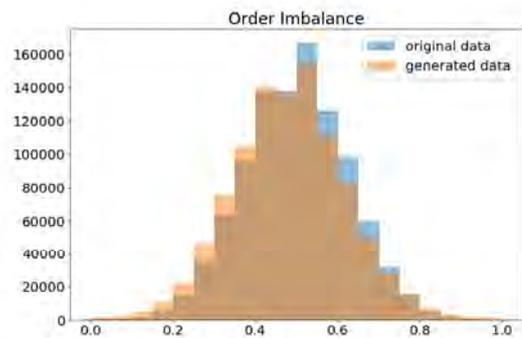
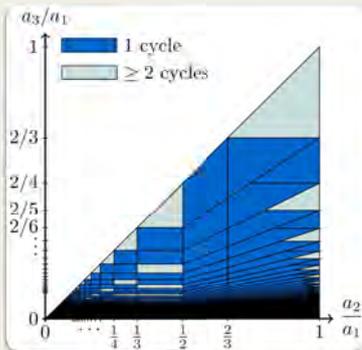


Figure 2: Histogram of Order Imbalance

Some directions for research to understand the functioning of the brain.



The parameter space, partitioned by the cycle(s) that the system can end up in given any initial condition.

Conclusion: Many different behaviours occur, infinitely many as the speed ratios approach zero. The dynamics imposed on the structure gave it a capacity to *encode information*. In the future we will study cycles and combinations of several similar graphs where we expect this

1. Particle flow dynamics on graphs: multiple phase-transitions

Dynamics: Consider particles flowing on the graph below. From O , each particle goes in the direction in which the next particle is the furthest distance away.

Parameters: The speeds, a_i , that the particles have on each outgoing edge.

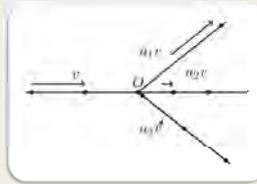
Initial conditions: The distances from O to the nearest particle in each outgoing edge at time $t = 0$.

Output: The sequence of chosen edges, which will be a *cycle*.

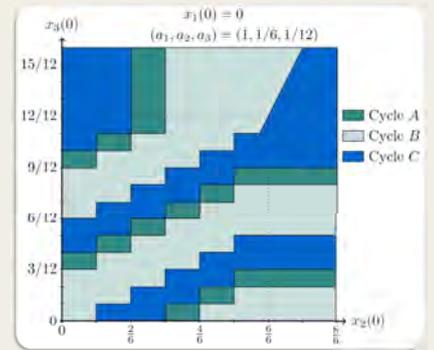
Partition...

- The parameter space by the different behaviours (left)
- Given a_i s, the space of initial conditions by the resulting cycle (right)

These are both highly non-trivial partitions!



The system in question. All particles move with the constant speed shown.



A hyperplane of the space of initial conditions for parameters that admit three cycles. It is partitioned by which cycle (A, B or C) the system will end up in given that initial condition.

2. The Hopfield model for auto-associative memory

The random model

Structure: A randomly diluted graph with $i \rightarrow j$ weight $W_{ij} = \frac{1}{N} \varepsilon_{ij} \sum_{\mu=1}^M \xi_i^{\mu} \xi_j^{\mu}$, where $\varepsilon_{ij} \in \{0,1\}$, i.i.d.

Parameters: M 'patterns' $\xi^1, \dots, \xi^M, \xi^i \in \{-1,1\}^N$, and the dilution probability.

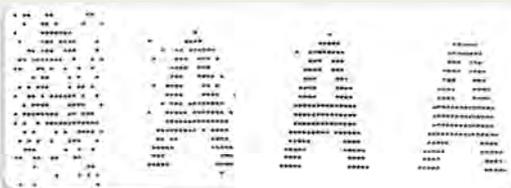
States: The configuration $\sigma(t) \in \{-1,1\}^N$ of 'spins' at time t .

Dynamics: The spin evolution

$$T: \sigma_i(t+1) = \text{sign} \left(\sum_{j=1}^N \sigma_j(t) W_{ij} \right)$$

The **fixed points** of T correspond to the patterns ξ_i and are the minima of the Hopfield Hamiltonian

$$H_N(\sigma) = - \sum_{i,j=1}^N W_{ij} \sigma_i \sigma_j.$$



(Left) The Hopfield model with a noisy input and resulting fixed point corresponding to a learned pattern.

(W.Kinzel)

The dilute model

Some (even many) weights W_{ij} can become small and contribute little. Randomly removing weights has been shown to preserve pattern retrievability surprisingly well. Better:

Remove the smallest weights!

- Study the structure of the remaining graph
- Is there a correlation between the data (ξ^i) and the structure?
- Are certain graphs more 'efficient' than others, i.e. large M but few non-zero W_{ij} ?
- Can the bounds of random dilution be improved with this method?
- We currently study different random structures

The goal: relate neuronal networks and (artificial) intelligence

1. Mathematical analysis of dynamics on diluted (random) networks
2. Find correlations between function and structure
3. Derive the structure from empirical results on the brain

3. Neuronal networks and graphs

The neuronal network in the brain is a directed graph:

- Neurons are nodes
- Axon-dendrite connections are directed edges between nodes

Neuroscientists are working hard to map these graphs, but what do we want that for? We need **dynamics** on the graph structure in order to understand its function! Models exist that generate similar graph structures (see right).

Two simple suggestions:

Cellular automata: a node is 'on' when at least n neighbours are 'on' and is 'off' otherwise.

- Long-term behaviour: all turn off / oscillatory / all (or a subset) turn on
- Certain parameters correspond to certain behaviours.

Phase transitions?

The Hopfield model (above) where only edges present in the graph can have non-zero weights.

- How do these graphs compare with the *random* model above?
- Are they more similar to any found in the *dilute* model?

Neuronal networks are evolutionarily adapted to perform their function well, so finding dynamical models where similar structures arise, and *outperform* purely random variants, can give new insight into that functionality!

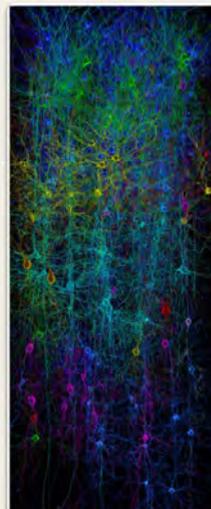
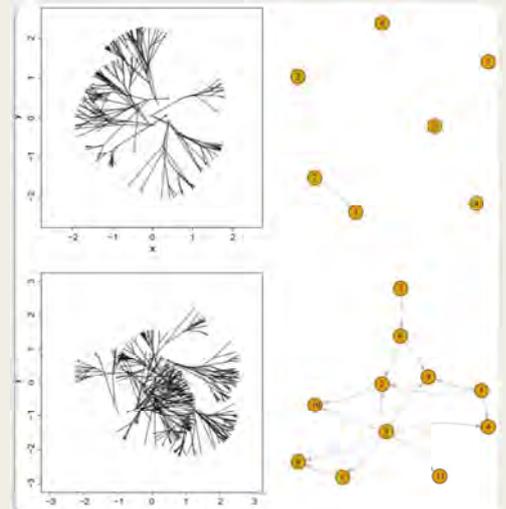


Illustration of a neuronal network. (Hill, Wang, Riachi, Shürmann, Markam)



Two simulations and the resulting graphs. (Ajazi, Chavez-Demoulin, Turova)

LEARNING DYNAMICAL SYSTEMS

Inês Lourenço

ineslo@kth.se

KTH Royal Institute of Technology

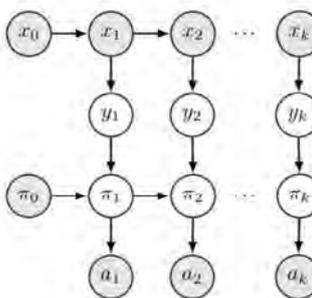


Description

My research topic is called Learning Dynamical Systems, and this can be applied to a broad area of problems. It provides backbone algorithms for digitalization of industry and society, such as core technology in autonomous systems with applications like smart buildings, self-driving vehicles, and self-learning robots.

I am currently looking at a wide range of problems structures related to the estimation of the belief (posterior distribution) of an agent on the state of the world, and the estimation of its state and sensor properties. The main application we are working with is the inverse filtering problem for counter-adversarial autonomous (CAA) systems.

Background & Motivation

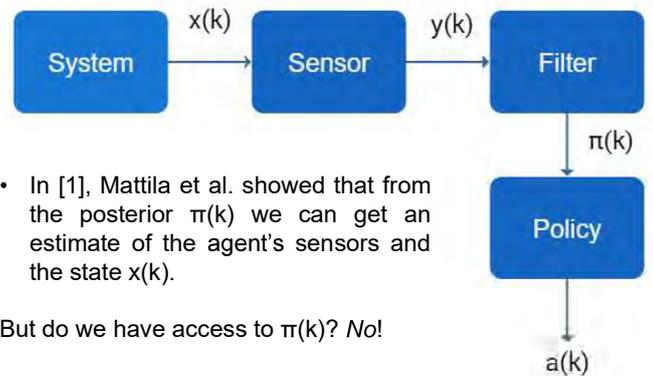


We consider the problem where an agent is modeled as a discrete-time hidden Markov model (HMM). At each time step, this agent:

- is in a state of the world $\mathbf{x}(k)$,
- receives an observation $\mathbf{y}(k)$
- from which it updates its posterior distribution $\boldsymbol{\pi}(k)$ (belief over the states of the world),
- and based on which it performs action $\mathbf{a}(k)$.

Our goal is to, based on our knowledge of the state of the world and the actions done by the agent, learn its internal variables such as the set information it has, and its posterior distribution and state. This setting is related to **inverse filtering** and is important for applications such as *social learning*, *inverse portfolio allocation* and *counter-adversarial systems*.

Research Goal & Questions

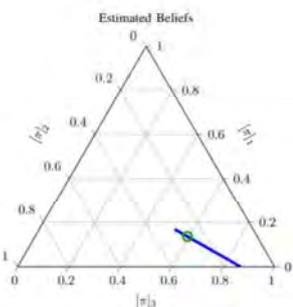


- In [1], Mattila et al. showed that from the posterior $\boldsymbol{\pi}(k)$ we can get an estimate of the agent's sensors and the state $\mathbf{x}(k)$.

But do we have access to $\boldsymbol{\pi}(k)$? No!

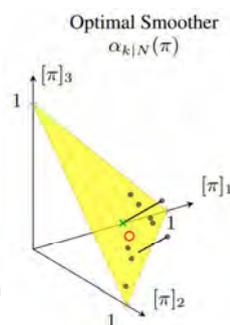
- So, given the state sequence $\mathbf{x}(0), \dots, \mathbf{x}(k)$ and actions $\mathbf{a}(1), \dots, \mathbf{a}(k)$, how to estimate the adversary's beliefs $\boldsymbol{\pi}(1), \dots, \boldsymbol{\pi}(k)$?

Methods & Preliminary Results



○ The agent's actual private belief
— The set Π_k

In [2] we show that we can obtain a set of beliefs $\Pi(k)$, consistent with the actions $\mathbf{a}(k)$ performed by the agent, as shown in blue in the figure on the left.

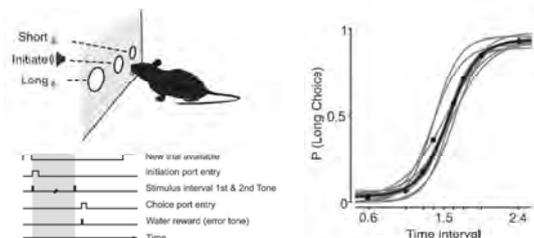


In a future paper we will show that we can get a better estimate of the set $\Pi(k)$ by using a smoother for the inverse filter (figure on the right).

Roadmap & Milestones

The goal is to later apply these inverse problems in an **inverse Reinforcement Learning** setup.

For example, if we have a neuroscientific theory, such as how temporal perception is coded in the brain, how can we test it in robots? By observing the agent's actions, can we infer its belief $\boldsymbol{\pi}(k)$ on the state of the world?



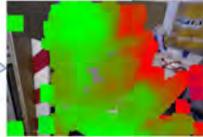
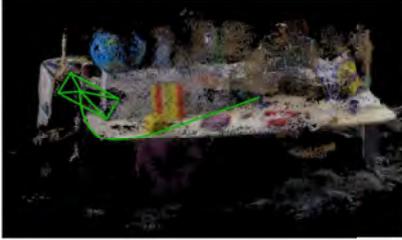
References

[1] R. Mattila, C. R. Rojas, V. Krishnamurthy, and B. Wahlberg, "Inverse filtering for hidden Markov models," in Advances in Neural Information Processing Systems (NIPS'17), pp. 4207–4216, 2017.
[2] R. Mattila, I. Lourenço, C. R. Rojas, V. Krishnamurthy, and B. Wahlberg, "Estimating private beliefs of Bayesian agents based on observed decisions," IEEE Control Systems Letters, vol. 3, pp. 523–528, July 2019.

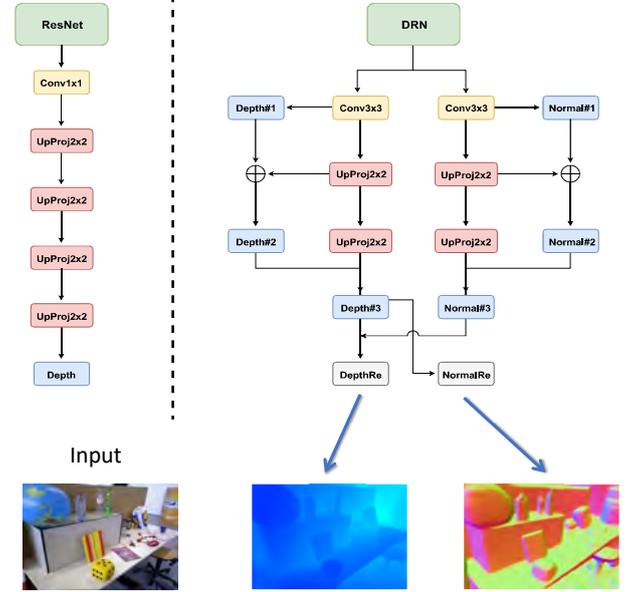


Introduction

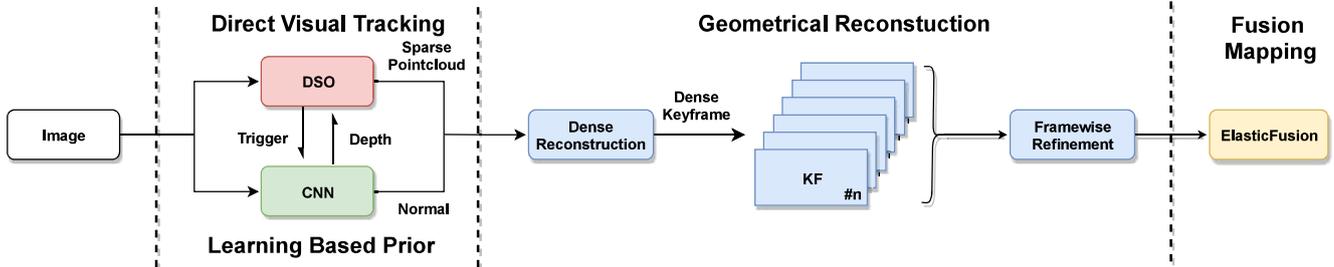
In this paper, we proposed a new deep learning based dense monocular SLAM method. The proposed framework constructs a dense 3D model via a sparse to dense mapping using learned surface normals. With single view learned depth estimation as prior for monocular visual odometry, we obtain both accurate positioning and high quality depth reconstruction.



Coupled Depth/Normal Estimation



System Overview



Quantitative

Absolute Trajectory Error

Datasets	S2D	DDSO	DSO	CNN-SLAM	LSD-B	LSD	ORB	Laina
TUM/seq1	0.071	0.552	1.221	0.542	1.717	1.826	1.206	0.809
TUM/seq2	0.078	0.203	0.123	0.243	0.106	0.436	0.495	1.337
TUM/seq3	0.072	0.335	0.648	0.214	0.037	0.937	0.733	0.724
ICL/office0	0.132	0.409	1.118	0.266	0.587	0.528	0.430	0.337
ICL/office1	0.131	0.155	0.633	0.157	0.790	0.768	0.780	0.218
ICL/office2	0.085	0.456	0.795	0.213	0.172	0.794	0.860	0.509
ICL/living0	0.137	0.143	0.404	0.196	0.894	0.516	0.493	0.230
ICL/living1	0.082	0.028	0.187	0.059	0.540	0.480	0.129	0.060
ICL/living2	0.045	0.162	0.668	0.323	0.211	0.667	0.663	0.380

Percentage of Correct Depth

Datasets	S2D	CNN-SLAM	LSD-B	LSD	ORB	Laina	REMODE
TUM/seq1	53.287	12.477	3.797	0.086	0.031	12.982	9.548
TUM/seq2	66.628	24.077	3.966	0.882	0.059	15.412	12.651
TUM/seq3	37.683	27.396	6.449	0.035	0.027	9.450	6.739
ICL/office0	27.445	19.410	0.603	0.335	0.018	17.194	4.479
ICL/office1	19.702	29.150	4.759	0.038	0.023	20.838	3.132
ICL/office2	27.059	37.226	1.435	0.078	0.040	30.639	16.708
ICL/living0	19.337	12.840	1.443	0.360	0.027	15.008	4.479
ICL/living1	25.090	13.038	3.030	0.057	0.021	11.449	2.427
ICL/living2	68.907	26.560	1.807	0.167	0.014	33.010	8.681

Experimental Results

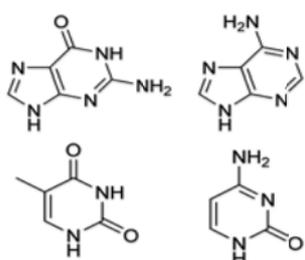
Qualitative



Description:

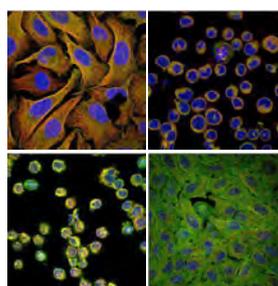
Developing new drugs is a long and costly process, both in terms of time and resources. Early stages of the process include High Throughput Screenings (HTS) which involves investigating the effect of applying a compound to a cell line and observing the response. Predicting the outcome of such experiments could speed-up the drug discovery process and provide new insight into the underlying biological processes. To achieve this, experimental data from previous HTS is repurposed and used to train Deep Learning models to predict drug activity over a number of different cell lines and targets.

Background & Motivation



Once a target that regulates a certain disease has been identified a compound that has an effect on the target is of interest. Finding such a compound often involves screening thousands of compounds. What if instead of screening thousands compounds we only screen the hundred that are most likely to have an effect based on or predictive model?

Research Goal & Questions

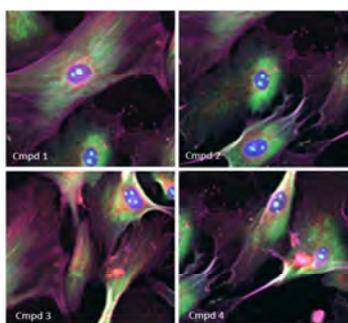


The dataset used in this project were not produced with machine learning in mind and presents a number of challenges.

- Sparse
- Diverse Origin
- Diverse Datatypes
- Dataset Size

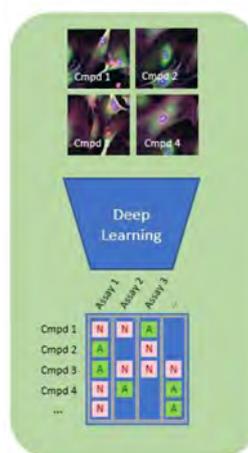
The goal of this project is to develop Deep Learning methods capable of performing well when applied to datasets exhibiting the challenges presented above.

Methods & Preliminary Results



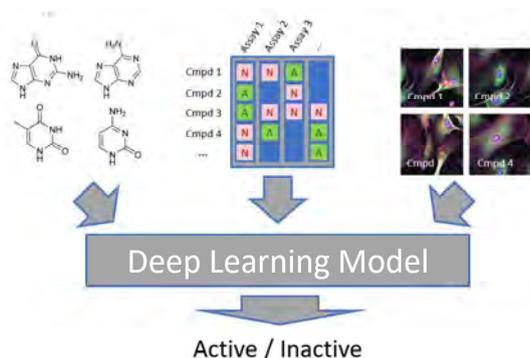
Using florescent microscopy images of thousands of compounds applied to a certain cell line, combined with activity data for the same compounds but different cell lines one can build a Deep Learning models capable of predicting drug activity. The approach is similar to a supervised learning binary classification task and has been shown to yield actionable results by other research groups which we have reproduced

Roadmap & Milestones



- Developing Semi-Supervised Learning approaches to improve current prediction power with sparse labels
- Exploring Multi-Modal Deep Learning approaches to draw information from diverse data sources
- Utilizing Explainable AI methods to find biological insights and relations found by the Deep Learning models

Combining multiple data sources



- Academic Advisors
 - Kevin Smith
 - Hossein Azizpour
- Industrial Supervisors
 - Erik Müllers
 - Johan Karlson
 - Karl-Johan Leuchowius

Planning for minimum uncertainty

Jonas Nordlöf (LiU/FOI) jonas.nordlof@foi.se

Supervisor: Daniel Axehill (LiU)

Co-supervisor: Gustaf Hendeby (LiU), Jonas Nygårds (FOI)

Abstract

By planning for minimum uncertainty, it is possible to improve the positioning accuracy of an autonomous system in difficult and GNSS-degraded environments. The project aims at incorporating information regarding landmark densities, as well as terrain properties, in order to plan a path that minimises the expected positioning uncertainty.

Machine learning (ML) will be studied for the purpose of estimating landmark densities based on ordinary maps. The project intends to evaluate the proposed approach on an autonomous system together with the Swedish Defence Research Agency (FOI) in Linköping.



Figure 1: Sensor platform for data collection and real-time execution of positioning algorithms.

Background

Traditional planning assumes a low and uniform uncertainty in the position estimate throughout the planning space. This assumption is not always true due to, e.g.:

- obstruction of sensor measurements;
- GNSS-degradation;
- lack of recognizable and static features; and
- uneven, unstable, or even moving ground surfaces.

All of these factors need to be considered to assure mission success in real-world scenarios. Solving this problem is often referred to as informative path planning (IPP).

Project goals

The main objectives of the project is to:

- jointly solve the task of state-estimation and action planning to predict positioning uncertainty;
- be able to plan the execution of a task in unstructured environments; and
- be able to model the effects of terrain properties on the motion of the system.

Key concepts

- No assumption of known landmarks are made.
- Expected information gain is maximized given knowledge of landmark densities.
- Aims at using ML for identifying beneficial vantage points during planning.
- Utilize ML for estimating landmark densities from pre-existing maps and sensor data.
- Incorporate negative information.

Preliminary results

Currently, the project has:

- developed theory to use virtual landmarks to describe the information potential in unseen areas;
- developed an algorithm which calculates an approximate solution to the IPP problem; and
- validated the IPP solution in a simulated environment.

Evaluation

To evaluate the performance of the developed methods in realistic scenarios, and to visualize the technology to different stakeholders, a proof-of-concept platform will be developed based upon a pre-existing positioning system.

Collaboration

The project is being funded by the division of C4ISR within the Swedish Defence Research Agency (FOI). The project will follow the Swedish armed forces Research and Tech (R&T) program for the area of autonomous systems and the area of sensors and signature management, which includes research on autonomous localization using sensor data. These R&T programs provide a wide set of international cooperation's through, among others, EU, EDA and NATO that will be utilized through out the project.

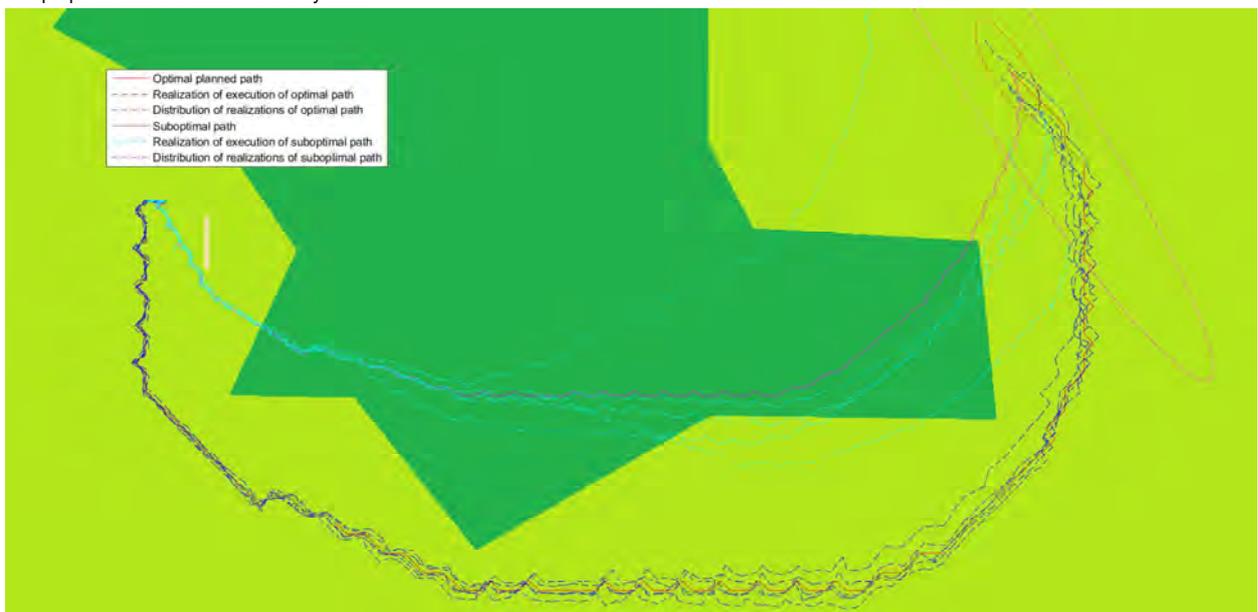


Figure 2: Planned path based on landmark densities plotted with an estimated trajectory with sampled landmarks. The dark green area has a lower landmark density, the algorithm therefore avoids the area and finds a path through the bright green area which has a higher landmark density. The realizations of the different paths are created by sampling landmarks from the densities and letting a simulated self-positioning autonomous system move along the path.

Design and Formal Verification of a Safe Stop Supervisor for an Automated Vehicle

Jonas Krook, Lars Svensson, Yuchao Li, Lei Feng, Martin Fabian
krookj@chalmers.se, larsvens@kth.se, yuchao@kth.se, lfeng@kth.se, fabian@chalmers.se



Questions for a safe stop supervisor:

- What safety benefits are achieved by formally verifying the software?
- What requirements can be proven? Which cannot be?
- How is nominal functionality included in the architecture and methods?

Safe transportation; can we STOP here?

The scenario considered is when an automated vehicle is parked in a spot at parking lot A, and it receives a transport mission where it needs to drive to and park in a goal spot at parking lot B. To do this, it first has to plan a path connecting the two parking lots via the road network. Then it needs to generate a path from a point in A to the road network. When it arrives at parking lot B it needs to construct a path from the road to the goal parking spot.

There is no driver so the vehicle itself needs to ensure safe driving, which means that the vehicle always should be able to reach a safe state while driving, if an error occurs. For driver assistance functions the safe state is usually to cede all control to the driver, but for automated vehicles we cannot let the vehicle continue to move uncontrolled. So the safe state has to include being stationary. However, the vehicle cannot stop just anywhere.

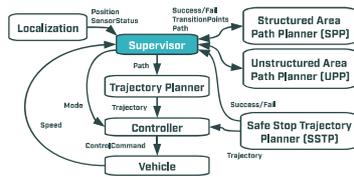


Fig. 1: System architecture.

Fig. 2: Example mission with paths and transition points.

One Supervisor to rule them all

To accomplish the transport mission, a supervisor brings together the two nominal path planners UPP and SPP, and makes sure that the SSTOP stops the vehicle in a safe spot (e.g. on the shoulder) if and only if an error occurs. The supervisor is implemented using model based design and integrated in a ROS environment.

The proposed supervisor can handle GPS sensor failures and path generation failures. Extensions for additional types of failures can be accommodated with relative ease.

Formal verification requires a model of the vehicle and the software, so a verification model of the supervisor is derived from its implementation. Only key aspects of the other software components in Fig. 1 are modelled. The vehicle is modelled as a standard discrete-time vehicular longitudinal dynamic system along the length of the path.

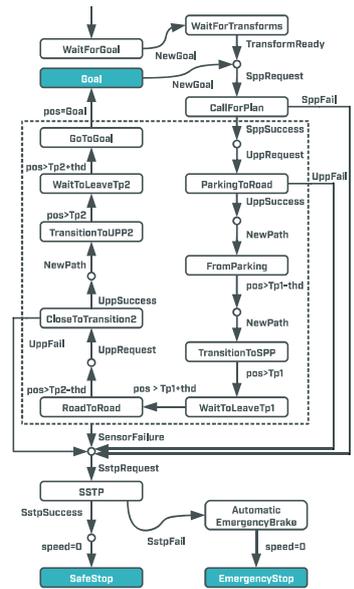


Fig. 3: The proposed supervisor.

Going for SPIN (What the model checker proves)

The model checker SPIN is used to formally prove the following requirements when the supervisor is interacting with the rest of the system:

- The supervisor and the four concurrent planners shall never stop at invalid end states.
- There is always a future state in which the vehicle is stationary.
- When arriving at the goal position, all paths must have been generated.
- If the vehicle passes the end point of a path then the next path must be known already.
- If the vehicle stops safely then a failure must have occurred and the SSTOP cannot have failed.
- If the vehicle stops by emergency braking then a failure must have occurred and the SSTOP must have failed.
- If a failure occurs then the vehicle must be stopped safely by SSTOP. If SSTOP fails then the automatic emergency brake must perform an emergency stop

SPIN also produces a counterexample when trying to prove that it is not possible to reach the goal, showing that the supervisor can successfully coordinate a transport mission.

Going for a spin

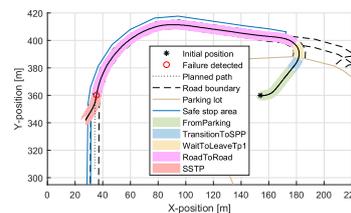


Fig. 4: Position and state during a simulation with a GPS failure.

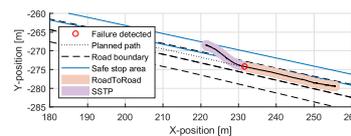


Fig. 5: Position and state during a drive with injected GPS failure.

Take-aways

- It is possible to formally verify the implementation and not only the design, giving more confidence of correct software.
- Implementation with verification in mind encourages safer coding.
- Only key abstract aspects of the nominal functionality need to be modelled for verification.
- It is difficult to formally verify requirements on nominal functionality with the chosen method.
- Verification requires modelling of software, which is manual and error prone.
- The current design may fall back on automatic emergency braking to stop. Future research should look at strategies that ensures the availability of SSTOP.



ANALYSIS OF A DEEP ONE UNIT RESIDUAL NETWORK

M.G. Larson, K. Larsson, J. Vallin

We consider a deep residual neural network with a simple form where each layer only contains one unit. In addition to the inner weights and bias each layer also has an outer weight vector which allows mapping into different dimensions. Such a network increases the approximation capacity with increasing depth and we show that this architecture can be used to solve classification problems in a robust and accurate way.

NETWORK ARCHITECTURE

We construct our network by stacking several residual layers of the form shown in Figure 2. One such layer maps the input vector $x_n \in \mathbb{R}^{d_n}$ to a vector $x_{n+1} \in \mathbb{R}^{d_{n+1}}$ by the piecewise linear mapping:

$$x_{n+1} = A_n x_n + a_n \text{ReLU}(b_n \cdot x_n + c_n) \equiv T_n(x_n),$$

where $A_n \in \mathbb{R}^{d_{n+1} \times d_n}$ is the identity if $d_n = d_{n+1}$, $a_n \in \mathbb{R}^{d_{n+1}}$, $b_n \in \mathbb{R}^{d_n}$ and $c_n \in \mathbb{R}$.

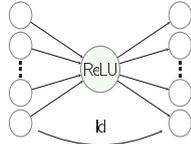


Figure 1: A residual layer with one unit.

We may restrict b_n to be a unit vector since the ReLU function allows factoring out positive constants. A network of depth N will be a function $F: \mathbb{R}^{d_1} \rightarrow \mathbb{R}^{d_N}$ constructed by a composition of N such mappings followed by an affine transformation $L: \mathbb{R}^{d_N} \rightarrow \mathbb{R}$.

$$F(x) = L \circ T_N \circ T_{N-1} \circ \dots \circ T_1(x).$$

We define the dimension of the network as the maximum of all d_1, \dots, d_N .

PROPERTIES OF THE MAPPING

Since b_n is a unit vector, the function $\rho_n(x) \equiv b_n \cdot x + c_n$ will be the signed distance from the hyperplane given by $b_n \cdot x + c_n = 0$. This defines the two half spaces:

- $H_n^- = \{x : \rho_n(x) \leq 0\}$,
- $H_n^+ = \{x : \rho_n(x) > 0\}$.

For $d_n = d_{n+1}$ the mapping $T_n: \mathbb{R}^{d_n} \rightarrow \mathbb{R}^{d_{n+1}}$ is the identity on H_n^- while points in H_n^+ will be mapped. Depending on the value of the quantity $a_n \cdot b_n$ the mapping T_n will have different properties:

- $a_n \cdot b_n > -1$, T_n is a bijection,
- $a_n \cdot b_n = -1$, T_n projects H_n^+ onto the hyperplane,
- $a_n \cdot b_n < -1$, T_n is a folding that maps H_n^+ on H_n^- .

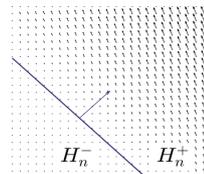


Figure 2: Visualization of a map $T_n: \mathbb{R}^2 \rightarrow \mathbb{R}^2$.

LOCALIZING STRUCTURES

Given a set of k hyperplanes in \mathbb{R}^d we show that a network of dimension $d+1$ and k layers can realize a function whose support is precisely $\bigcap_{i=1}^k H_i^+$. In other words, any d -dimensional convex polytope can be represented by a level set of a $(d+1)$ -dimensional network with sufficient depth.

Moreover, for $S^+ = \{x : F(x) > 0\}$, i.e. the positive level set of a network, we can obtain positive level sets corresponding to the following fundamental set of operations by adding one layer:

- Union: $S^+ \cup H_n^+$,
- Intersection: $S^+ \cap H_n^+$,
- Difference: $S^+ \setminus H_n^+$,
- Complement: $(S^+)^c$.

By iterating these operations more complex geometries can be constructed as well.

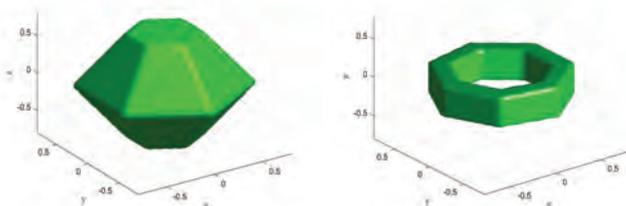


Figure 3: Two polyhedras (convex and non-convex) as levelsets of two different networks.

CLASSIFICATION PROBLEMS

A binary classification problem can be phrased as follows: given two sets of data points S_1 and S_2 belonging to two different classes, train a network such that it will classify unobserved points correctly. In most cases the data lives in a high dimensional space but a common hypothesis is that the data itself resides on some embedded low dimensional manifold embedded.

There are two different point of views to this problem. If the network is seen as a composition of mappings, the problem is to construct these mappings such that the datasets become linearly separable in the final coordinate system. In Figure 4 one can see how the spiral dataset evolves through the layers of a 3-dimensional network. In the final layer the datasets have been transformed such that they can be separated by a hyperplane.

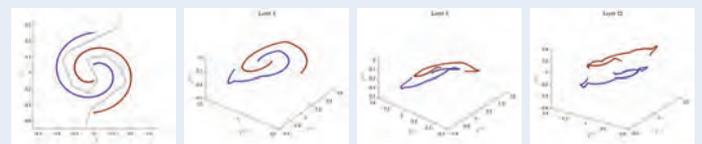


Figure 4: Evolution of the spiral dataset embedded in \mathbb{R}^3 .

Alternatively, if the network is viewed as a function F , the classification problem is solved if F has a level surface that separates the two sets. Then, a new observation is classified according to its position relative to the level surface. In Figure 5 two classification problems are shown where 4-dimensional networks were trained to solve the problems. The resulting green level surfaces separate the different classes in both cases.

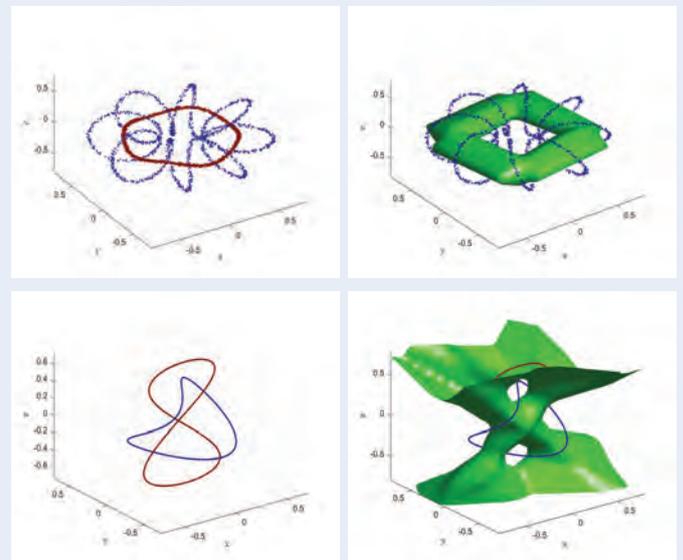


Figure 5: Two different datasets where the trained networks have constructed level surfaces that separate the datasets.

Learning of control architectures for robot automation

PRESENTER: Jonathan Styrud

INTRO

- The main obstacle holding robots back is the effort needed for **task programming**. When the program is generated automatically with show and tell like programming, robot automation will be **affordable for all**.
- This project will investigate methods for collaborative robots to autonomously learn control architectures such as behavior trees.

FULL PROJECT TITLE

Autonomous learning of control architectures for real-time industrial robot automation in a dynamic environment

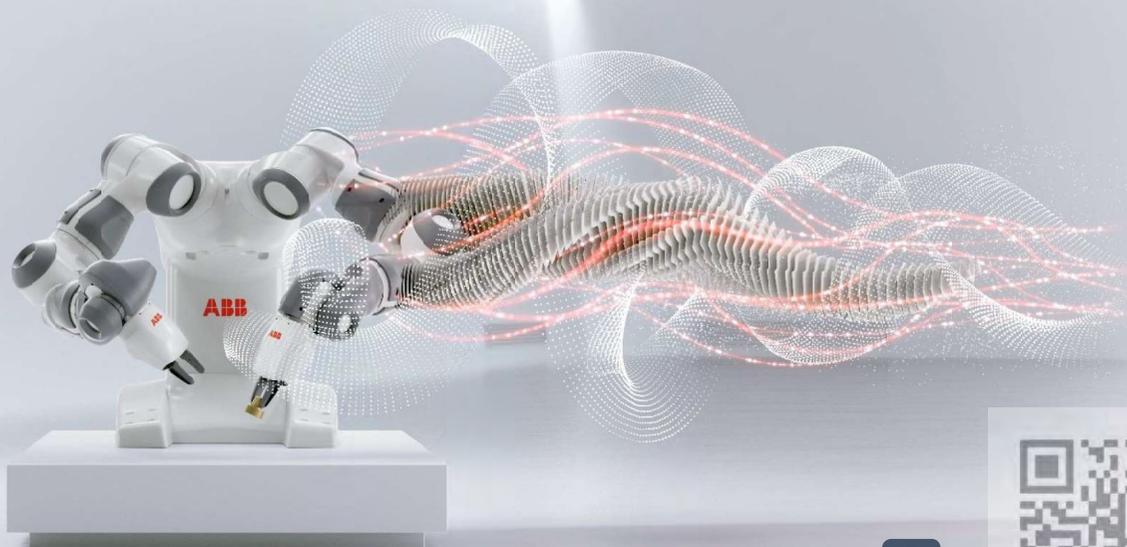
METHODOLOGY

- **Restrict scope**. Assume vision and grasping is solved, assume HMI interface exists, assume basic skills are in place, restrict to point-to-point manipulation etc.
- **Use simulation**, real world trials are too expensive.
- **Select control architecture** that is **modular** and **transparent**, such as **behavior trees**. Don't try to do pixel-to-torque.
- **Start small** and simple, and later expand with more complex tasks.

RESULTS

- Nothing to report so far!

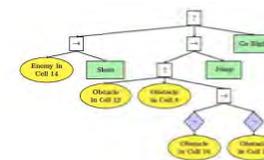
The main obstacle holding robots back is the effort needed for **task programming**. When the program is generated automatically, robot automation will be **affordable for all**.



Take a picture for contact details



Behavior tree example:



Overall picture:



Automatic planning is not feasible because it typically assumes:

- Perfect prediction of the effect of various actions
- Perfect knowledge of the system state
- Small upper bounds on complexity of the assembly tree



DESCRIPTION

This project seeks to develop methods to verify and monitor the ethical behavior of AI systems based on the observation of their input and output according to a continuously evolving societal optimum.

BACKGROUND & MOTIVATION

- Current Machine Learning approaches can be notoriously difficult to monitor due to their use of black box algorithms.
- Trust on autonomous systems depends on the possibility to verify and minimally control their behavior. [1]
- The development of theories and methods for control and verification of AI systems requires going beyond traditional methods:
 - These systems operate in open environments, which cannot be fully described nor defined a priori.
 - Opacity is inherent in open environments because the systems are developed and operated by different parties.
- AI systems must comply with societal values and ethical principles, such as fairness, non-discrimination, safety, and privacy.

RESEARCH GOAL & QUESTION

- Find expressive modelling languages that can be easily understood by humans, but generate tractable fragments for automated verification.
- Model ethical contexts using formal languages, possibly taking advantage of semantic reasoners [3] or properties of specific domains [4].

EXAMPLE

This is a minimalistic example:

- $\text{ethical_rule}(p_1, p_2, p_3) = p_1 \wedge (p_2 \vee p_3) \wedge \neg(p_2 \wedge p_3)$
- $\text{ethical_value} = v$
(an *acceptable value* according to society)
- $\text{black_box}(x_1, x_2, x_3) = w_1 \cdot x_1 \cdot (x_2 + x_3) \cdot (1 - w_2 \cdot x_2 \cdot x_3)$
- $\text{black_box_conf} = (w_1, w_2)$
- $\text{input_tr}(p_1, p_2, p_3) = (x_1, x_2, x_3)$
such that $x_i = 1$ if p_i , 0 otherwise
- $\text{output_tr}(x) = x \geq v$

	black box conf	ethical value	result
1.	(0.5, 1.0)	0.5	passes
2.	(0.5, 0.5)	0.5	fails
3.	(0.4, 1.0)	0.5	fails
4.	(0.4, 1.0)	0.4	passes

Failing cases

2. $\{(\text{true}, \text{true}, \text{true}) [0.5]\}$
(it should be < 0.5 to model **false**)
3. $\{(\text{true}, \text{false}, \text{true}) [0.4],$
 $(\text{true}, \text{true}, \text{false}) [0.4]\}$
(it should be ≥ 0.5 to model **true**)

METHODS & PRELIMINARY RESULTS

The following diagram shows how an AI system can be tested.

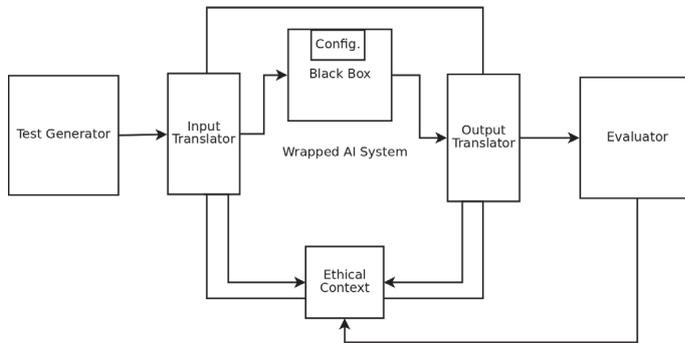


Figure 1: Test environment.

- **Ethical Context:** ethical norms written in a formal language, which can use, for example, *logical rules* and *numerical values*.
- **Black Box:** an opaque AI system.
- **Black Box Configuration:** a particular state of the AI system.
- **Input/Output Translators:** layers to communicate with the AI system.
- **Wrapped AI System:** AI system inside a glass box [2].
- **Test Generator:** generates tests to verify the AI system.
- **Evaluator:** verifies whether the AI system is ethical with respect to an ethical context.

ROADMAP & MILESTONES

1. Identify AI systems that require ethical monitoring or verification.
2. Find expressive formalisms for ethical contexts.
3. Design an input/output protocol to test the AI systems.
4. Develop a prototype for ethical verification.

BIBLIOGRAPHY

- [1] Virginia Dignum et al. *Ethics by Design: necessity or curse?*. Proceedings of the 2018 AAI/ACM Conference on AI, Ethics, and Society, 2018.
- [2] A. Aler Tubella, A. Theodorou, F. Dignum, and V. Dignum. *Governance by Glass box: implementing transparent moral bounds for AI behaviour*. IJCAI, 2019.
- [3] Julian Mendez. *jcel: A Modular Rule-based Reasoner*. ORE 2012.
- [4] Carsten Lutz. *Description Logics with Concrete Domains — A Survey*. Advances in Modal Logic. World Scientific Publishing Co. Pte. Ltd., 2002, vol 4.

Robust learning of geometric equivariances

Karl Bengtsson Bernander, Joakim Lindblad, Nataša Sladoje

Centre for Image Analysis, Department of Information Technology, Uppsala University, Sweden

Abstract

We extend convolutional neural networks (CNNs) as to provide rotation equivariance. We evaluate several methods that incorporates this property into the network architectures. We start by replicating and evaluating these methods on existing datasets. Our plan is to apply promising methods to our own dataset consisting of microscopy images of cells, weakly labelled as cancer or no cancer. We expect that incorporating rotation equivariance into CNNs will increase the expressive capacity without increasing the number of parameters, reducing overfitting. Also, since data augmentation can be reduced, misclassification due to interpolation artifacts should decrease.

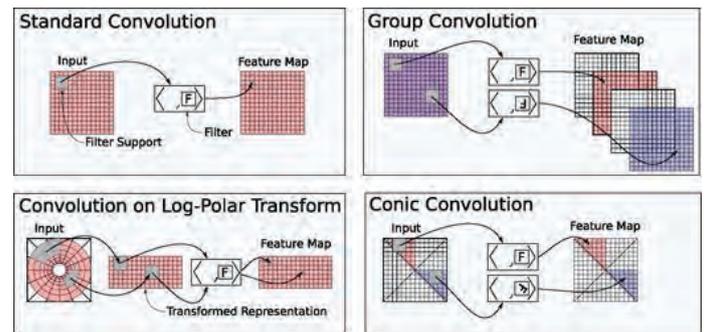
Existing work

One feature of standard convolutional neural networks (CNNs) is translational invariance: the result of convolving an input with a filter and then shifting the output is identical to shifting the input and then applying the convolution. We are interested in other equivariances, such as rotations and scaling. Recent works on rotation equivariance in CNNs include:

- Group-equivariant convolutional networks (G-CNNs) [3], using group-convolutions.
- Steerable filters [4], using linear combinations of a system of atomic filters to achieve arbitrary angular resolution w.r.t the sampled filter orientations.
- CFNet [1], using various filters at rotations in corresponding conic regions.
- Warped convolutions [5], transforming the image to the log-polar domain.

$$T_g \Phi(f) = \Phi(T_g f)$$

Rotational equivariance: filtering (Φ) an input f , then rotating (T_g), gives the same result as filtering on the rotated input.



Different convolution schemes [1]

Experiments and replications

We have begun the empirical part of the project by replicating some of the most promising methods for achieving rotational equivariance for CNNs [1], [3], [4]. As a reference point, we focus on the rotated MNIST dataset.

Classification accuracy on rotated MNIST for different CNN architectures

Method	Test error (%)
Standard CNN	5.03
G-CNN	2.28
CFNet	1.75
Steerable filters	0.71



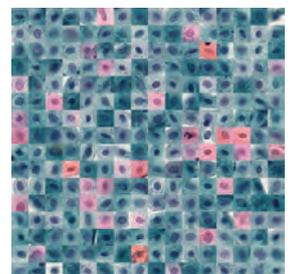
Samples from the rotated MNIST dataset

Further directions

We plan to apply promising methods to our own dataset, consisting of microscopy images of cells from the oral cavity. They are weakly labelled with cancer or no cancer.

The usage of rotation equivariant CNNs should increase accuracy by avoiding interpolation artifacts caused by data augmentation. We will also compare equivariances versus invariances, depending on texture and local shapes.

We are also interested in studying equivalence. Meaning, for different neural networks, we want to compare similarities in captured information content - that is, if there exists some map between them [2].



Microscopy image of cells from the oral cavity

References

1. Benjamin Chidester, Tianming Zhou, Minh N Do, Jian Ma. Rotation equivariant and invariant neural networks for microscopy image analysis. *Bioinformatics*, Volume 35, Issue 14, July 2019
2. Kaerl Lenc, Andrea Vivaldi. Understanding Image Representations by Measuring Their Equivariance and Equivalence. *International Journal of Computer Vision*, Volume 127, Issue 5, May 2019
3. T.S. Cohen, M. Welling. Group Equivariant Convolutional Networks. *Proceedings of the International Conference on Machine Learning (ICML)*, 2016
4. Weiler, Maurice, Hamprecht, Fred, Storath, Martin. Learning Steerable Filters for Rotation Equivariant CNNs. *Proceedings of the Conference on Computer Vision and Pattern Recognition*, 2018
5. Henriques J.F, Vedaldi A. Warped convolutions: efficient invariance to spatial transformations. *Proceedings of the International conference on Machine Learning*, 2017

An Optimization-Based Receding Horizon Trajectory Planning Algorithm

Kristoffer Bergman^{*†}, Oskar Ljungqvist^{*}, Torkel Glad^{*}, and Daniel Axehill^{*}

^{*}Linköping University, [†]Saab Dynamics

Motivation

Problem considered: Trajectory planning for nonlinear systems in unstructured environments.

Method: Combine a sampling-based motion planner and a receding horizon optimization algorithm.

Drawback of state-of-the-art: Does not guarantee feasibility beyond the planning horizon (recursive feasibility).

Main contribution: Use the solution from a *feasible* motion planner as terminal manifold to provide theoretical guarantees on feasibility during the entire planning horizon.

Problem formulation

Trajectory planning is an optimal control problem (OCP):

$$\begin{aligned} & \underset{\mathbf{u}(\cdot), t_f}{\text{minimize}} && J_{\text{tot}}(\mathbf{x}_0, \mathbf{u}(\cdot)) = \int_{t_0}^{t_f} \ell(\mathbf{x}(t), \mathbf{u}(t)) dt \\ & \text{subject to} && \mathbf{x}(t_0) = \mathbf{x}_0, \quad \mathbf{x}(t_f) = \mathbf{x}_f, \\ & && \dot{\mathbf{x}}(t) = f(\mathbf{x}(t), \mathbf{u}(t)), \\ & && \mathbf{x}(t) \in \mathcal{X}_{\text{free}}, \quad \mathbf{u}(t) \in \mathcal{U} \quad t \in [t_0, t_f]. \end{aligned} \quad (1)$$

- Computationally expensive to solve (or improve nominal solutions) for long horizon problems
- Idea: use receding horizon planning (RHP)
- Relies on a feasible nominal trajectory $(\bar{\mathbf{x}}(\cdot), \bar{\mathbf{u}}(\cdot), \bar{t}_f)$ computed by a motion planning algorithm
 - Used computationally to warm-start the RHP-step
 - Used theoretically for convergence guarantees

Receding horizon planning

- Introduce timing variable τ_k and cost-to-go function $\Psi_k(\tau_k)$

$$\begin{aligned} & \underset{\mathbf{u}_k(\cdot), \tau_k}{\text{minimize}} && J(\mathbf{x}_{\text{cur}}, \mathbf{u}_k(\cdot), \tau_k) = \Psi_k(\tau_k) + \int_{t_k}^{t_k+T} \ell(\mathbf{x}_k(t), \mathbf{u}_k(t)) dt \\ & \text{subject to} && \mathbf{x}_k(t_k) = \mathbf{x}_{\text{cur}}, \quad \mathbf{x}_k(t_k+T) = \bar{\mathbf{x}}_{k-1}(\tau_k) \\ & && \dot{\mathbf{x}}_k(t) = f(\mathbf{x}_k(t), \mathbf{u}_k(t)), \\ & && \mathbf{x}_k(t) \in \mathcal{X}_{\text{free}}, \quad t \in [t_k, t_k+T] \\ & && \mathbf{u}_k(t) \in \mathcal{U}. \end{aligned} \quad (2)$$

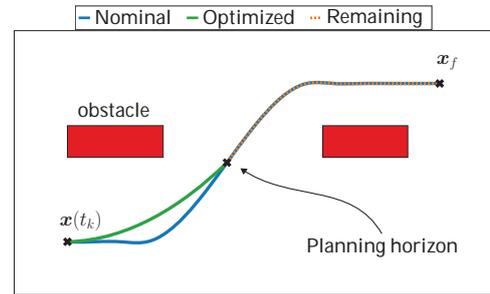
- Uses previous solution $\bar{\mathbf{x}}_{k-1}(\cdot)$ as “anchor” or terminal manifold
- Theoretical guarantees:
 1. Recursive feasibility
 2. Non-increasing objective function value $J_{\text{tot}}(\mathbf{x}_0, \mathbf{u}_k(\cdot))$
 3. Finite number of RHP iterations \rightarrow convergence to \mathbf{x}_f

Practical algorithm

- Practical issue: want to use piecewise continuous control inputs
- Nominal solution $\bar{\mathbf{x}}(\cdot)$ and cost-to-go function $\Psi_k(\cdot)$ not continuously differentiable \rightarrow (2) can not be solved using standard OCP interfaces
- Introducing minor adjustment to (2) such that standard direct optimal control methods can be used
- Theoretical guarantees still hold using this adjustment

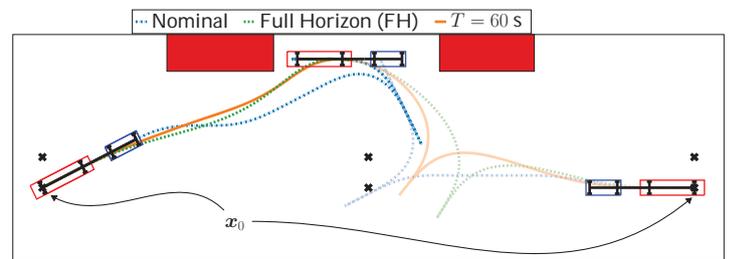
Acknowledgments

This work was partially supported by FFI/VINNOVA and the Wallenberg AI, Autonomous Systems and Software Program (WASP) funded by the Knut and Alice Wallenberg Foundation.



Simulation study

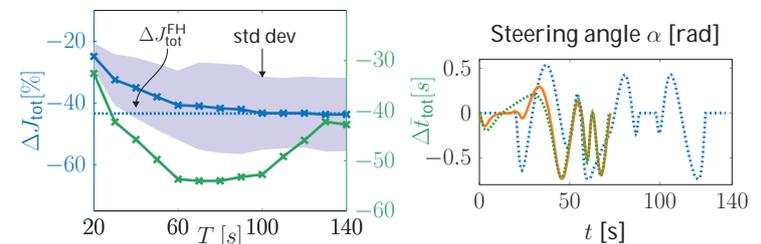
- A truck and trailer system in confined environments.
- A lattice-based motion planner used to compute nominal solutions.
- Compare performance using different planning horizons T .



Summary of results

- ΔJ_{tot} : Difference in objective function value.
- \bar{t}_{RHP} : Computation time per RHP iteration.
- $\Delta \bar{t}_{\text{lat}}$: Difference in latency time (i.e first RHP iteration)
- $\Delta \bar{t}_{\text{tot}}$: Difference in total time (execution time + computation time)

T [s]	20	40	60	80	120	Full horizon
ΔJ_{tot} [%]	-24.9	-35.2	-40.8	-41.7	-43.4	-43.7
\bar{t}_{RHP} [s]	0.09	0.29	0.77	2.0	10.4	17.0
$\Delta \bar{t}_{\text{lat}}$ [s]	0.35	0.73	2.0	3.5	12.3	17.0
$\Delta \bar{t}_{\text{tot}}$ [s]	-32.6	-45.7	-53.7	-54.0	-45.9	-44.1



Conclusions

- A new two-step trajectory planning algorithm introduced.
- Allows for trade-off between latency against solution quality.
- Nominal trajectory exploited for feasibility and convergence guarantees.
- Future work: Dynamic scenarios with online re-planning.

Manuscript available online at: <https://arxiv.org/abs/1912.05259>

Problem Formulation

As deployment of automated vehicles increases, so does the rate at which they are exposed to critical traffic situations. Such situations, e.g. a late detected pedestrian in the vehicle path, require operation at the handling limits in order to maximize the capacity to avoid an accident. Furthermore, the physical limitations of the vehicle typically vary in time due to local road and weather conditions. In this research project, we tackle the problem of motion planning and control at the limits of handling under time varying constraints, by adapting to local traction limitations. Details are provided in [1].

Real world example of the motion planning problem in a critical situation



https://en.wikipedia.org/wiki/Death_of_Elaine_Herzberg

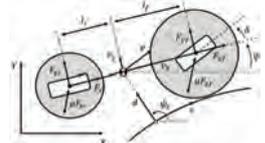
The corresponding optimal control problem. In our approach, locally varying traction information enter as time varying

$$\begin{aligned} \min_{u_0, \dots, u_{N-1}|t} & J(x_{k|t}, u_{k|t}) \\ \text{s.t.} & x_{k+1|t} = f(x_{k|t}, u_{k|t}), \\ & x_{k|t} \in \mathcal{X}_{k|t}, u_{k|t} \in \mathcal{U}_{k|t}(\mu_{k|t}), \\ & \forall k = 0, \dots, N-1, \\ & x_0|t = x_t, x_{N|t} \in \mathcal{X}_{k|t}, \end{aligned} \quad (1)$$



Proposed Method

In order to capture the effects of bounded tire forces, we use a dynamic vehicle model:



Where the states propagate as:

$$\begin{aligned} \dot{s} &= \frac{v_x \cos(\Delta\psi) - v_y \sin(\Delta\psi)}{1 - d\kappa_c}, \\ \dot{d} &= v_x \sin(\Delta\psi) + v_y \cos(\Delta\psi), \\ \Delta\dot{\psi} &= \dot{\psi} - \kappa_c \frac{v_x \cos(\Delta\psi) - v_y \sin(\Delta\psi)}{1 - d\kappa_c}, \\ \dot{\psi} &= \frac{1}{I_z} (l_f F_{yf} - l_r F_{yr}), \\ \dot{v}_x &= \frac{1}{m} (F_x - D_a v_x^2) - g \sin(\theta), \\ \dot{v}_y &= \frac{1}{m} (F_{yf} + F_{yr}) - v_x \dot{\psi} + g \sin \phi. \end{aligned}$$

with lateral and longitudinal tire forces as inputs. We model the locally varying traction limit as a time varying input constraint $U_{k|t}$ determined by

$$F_h \leq \mu F_z$$

Where F_z varies according to the pitch dynamics.

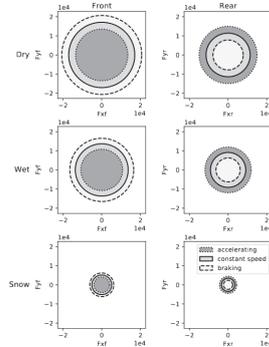


$$F_{zf} = \frac{1}{l_f + l_r} (m\dot{v}_x h - mgh \sin \theta + mgl_r \cos \theta)$$

$$F_{zr} = \frac{1}{l_f + l_r} (-m\dot{v}_x h + mgh \sin \theta + mgl_f \cos \theta)$$

and μ is identified online.

Here is an example of how the horizontal force limits vary in different conditions.



We introduce the time varying tire force bounds as a time varying constraint in the optimization (1).

This introduces further challenges in solving (1) in real time, which we address by augmenting RTI-SQP [2] with a trajectory rollout method [3]. The algorithm is summarized as

Algorithm 1 The SAA-RTI Algorithm

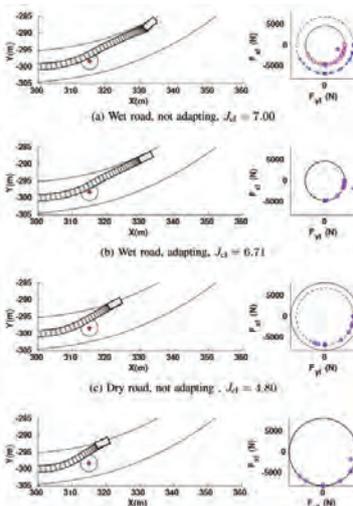
Input: $x_t, \mathcal{T}_{t-1}^*, M, O, \mu_{k|t}$
Output: \mathcal{T}_t^*
1: $\mathcal{T}_t^* \leftarrow$ shiftAndEnsureFeasibility(\mathcal{T}_{t-1}^*)
2: $\mathcal{U}_{k|t} \leftarrow$ computeAdaptiveConstraints($\mu_{k|t}$)
3: $S_t \leftarrow$ feasibleTrajectoryRollout($x_t, M, \mathcal{U}_{k|t}$)
4: **for** each trajectory \mathcal{T}_t^* in $[S_t, \mathcal{T}_{t-1}^*]$ **do**
5: $J(\mathcal{T}_t^*) \leftarrow$ evaluateCost(\mathcal{T}_t^*)
6: **end for**
7: $\mathcal{T}_t^* \leftarrow$ selectLowestCost(arg J(\mathcal{T}_t^*))
8: $A_{k|t}, B_{k|t} \leftarrow$ linearizeDynamicModel(\mathcal{T}_t^*)
9: $\hat{\mathcal{T}}_t \leftarrow$ computeStateConstraints(\mathcal{T}_t^*, O, M)
10: $\mathcal{T}_t^* = (x_{k|t}^*, u_{k|t}^*) \leftarrow$ opti($\hat{\mathcal{T}}_t, \mathcal{U}_{k|t}, A_{k|t}, B_{k|t}$)
11: **return** \mathcal{T}_t^*

and further elaborated in [1].

Results and Discussion

First, we compare traction adaptive trajectory planning with non-adaptive for an obstacle avoidance scenario. We make two observations:

1. When adapting to reduced traction, we observe improved control, because adapting avoids planning dynamically infeasible maneuvers
2. When adapting to improved traction, we observe a lower closed loop cost, because adapting enables full utilization of the available traction.



Closed loop trajectories for comparison between adaptive and non-adaptive trajectory planning and control. The vehicle is depicted in gray, a suddenly appearing obstacle in red. In the force plot to the right, blue crosses denote the commanded tire forces and magenta circles denote actual tire forces. The solid and dashed black lines represent the actual and assumed friction boundary respectively.

Results from Monte Carlo simulation of the critical obstacle avoidance scenario 1200 runs, with varied obstacle position, initial velocity and road geometry. J_{cl} denotes the average closed loop cost and P_{acc} the empirical accident probability over all runs.

Road Conditions	Strategy	J_{cl}	P_{acc}
wet road: $\mu_{act} = 0.55$	non-adaptive	5.33	42%
	adaptive	5.37	38%
dry road: $\mu_{act} = 0.95$	non-adaptive	3.84	13%
	adaptive	3.37	9%

Second, we generalize the results by means of Monte Carlo simulation. Results indicate that adaptation improves the vehicle's capacity to avoid accidents both at reduced and improved traction.

Third, we compare SAA-RTI similar methods, namely standard RTI-SQP [2] and state space sampling [3] (with MPC tracking). SAA-RTI represents an improvement in terms of optimality w.r.t physical capabilities and in terms of capacity to avoid local minima.



Comparison of closed loop trajectories between SAA-RTI (blue) and state space sampling with MPC tracking (orange)

Example of how SAA-RTI avoids local minima. Orange: converged SQP solution initialized left of obstacle. Blue: converged SQP solution initialized right of obstacle. Gray: Closed loop trajectory of the vehicle controlled by SAA-RTI

Conclusions and Future Work

Our results indicate that the concept of traction adaptation in motion planning:

- Increases the capacity to avoid accidents when adapting to both deteriorated and improved traction
- Enables optimality w.r.t time varying input constraints
- Reduces sensitivity to bad local minima

Next steps in this research project include full scale vehicle experiments and integration of online tire-road friction estimation.

References

- [1] L. Svensson, M. Bujarbaruah, N. Kapania, M. Tömgren. "Adaptive Trajectory Planning and Optimization at Limits of Handling". *arXiv preprint arXiv:1903.04240*, 2019
- [2] M. Diehl, H. G. Bock, and J. P. Schlöder, "A real-time iteration scheme for nonlinear optimization in optimal feedback control," *SIAM Journal on control and optimization*, vol. 43, no. 5, pp. 1714–1736, 2005.
- [3] M. Werling, S. Kammel, J. Ziegler, and L. Gröll, "Optimal trajectories for time-critical street scenarios using discretized terminal manifolds," *The International Journal of Robotics Research*, vol. 31, no. 3, pp. 346–359, 2012.
- [4] L. Svensson, L. Masson, N. Mohan, E. Ward, A. P. Brenden, L. Feng, and M. Tömgren, "Safe stop trajectory planning for highly automated vehicles: An optimal control problem formulation," in *2018 IEEE Intelligent Vehicles Symposium (IV)*, June 2018, pp. 517–522.

Model predictive control for autonomous landings in a search and rescue scenario at sea

Linnea Persson, KTH Royal Institute of Technology

Description

This research is about a Model Predictive Control approach for autonomous landing of a quadcopter on a cooperating ground vehicle. The landing maneuver is executed in a cooperative manner where both vehicles take actions to fulfill their common objective. The maneuver is designed to be feasible under a range of conditions, including scenarios where the boat is moving across the water or when it is subjected to disturbances such as waves and winds. During the landing, the vehicles must also consider various safety constraints for landing safely and efficiently.

Background & Motivation

The research is motivated by the large-scale demonstrator arena WARA-PS, which is equipped with autonomous boats and drones that collaborate to perform various tasks related to search and rescue at sea. Search and rescue missions benefit greatly from drones since they allow for an extended situation awareness and can work independently from the boats. However, drones tend to have a very limited battery time, and as such the range that they can reach is limited between charges. To extend the usage time, we wish to use the Unmanned Surface Vehicles as mobile charging stations, where drones can takeoff and land as necessary.

Research Goal & Questions

The objective of this work has been to show how MPC can be applied to autonomous cooperative maneuvers. The system must fulfill a set of safety constraints during the landing, as is illustrated here in the form of a spatial constraint to avoid protruding parts. The objective is to use algorithms that are applicable to real-life scenarios and to verify in real experiments.



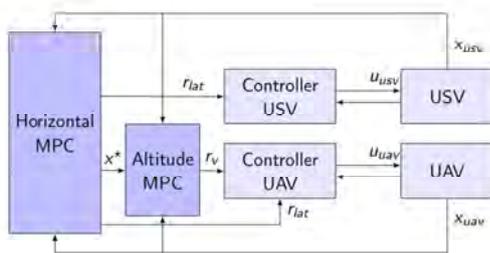
Methods & Preliminary Results

The algorithms are implemented both in hardware-in-the-loop simulations, where we demonstrate some of the different scenarios that the algorithm is expected to handle, as well as on a real drone landing on a virtual boat. For implementation on the real boat, we will look into different ways of doing relative positioning.

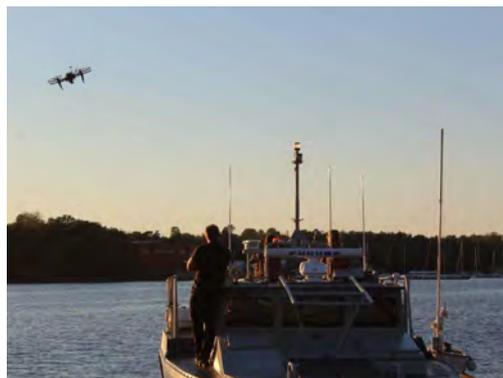
Current research

Current research directions include distributing the computations on the two vehicles and defining communication and update conditions for the cooperative trajectory. In recently submitted papers we perform landing experiments where we are able to increase the look-ahead time by several seconds only by distributing the computations, and an even further increase was achieved by combining an inner and outer MPC controller. This approach was also shown to make the system more robust to communication delays.

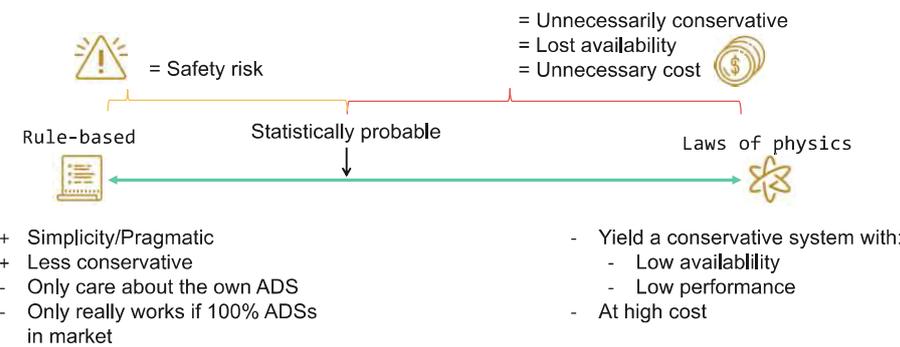
We are also working on defining shrinking horizon conditions that work in real-time while also being able to handle certain disturbances. The objective is to be able to adapt the prediction and the precision of the computed trajectories to the distance until the rendezvous.



The system consists of a DJI matrice 100 drone and a rebuilt CB90 boat from Saab Kockums.



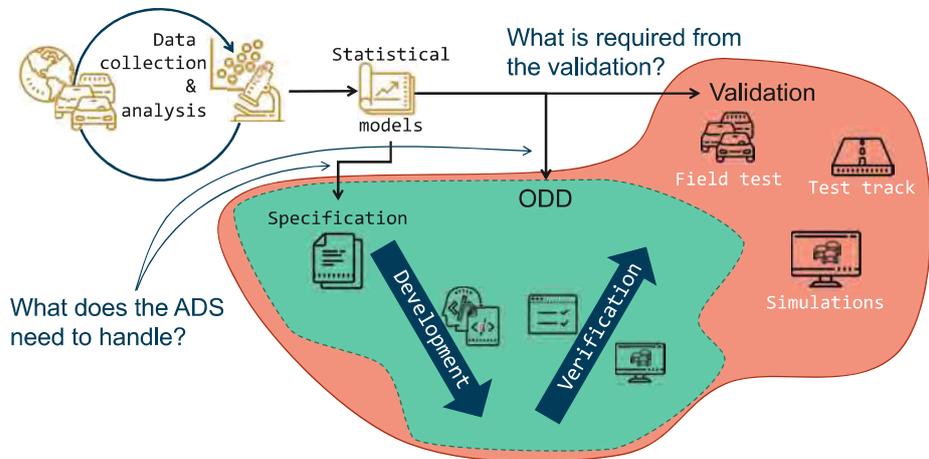
For releasing an Automated Driving System (ADS) into the public it is necessary to craft a credible and complete safety argumentation. Without a solid safety argumentation a perfectly working ADS might be kept off the market due to lack of safety proof. There are several caveats in crafting a safety argumentation. As the ADS is intended to operate in an uncertain environment, with other active traffic participants, one of the challenges is understanding what is to be expected from the ADS and what the ADS can expect from its environment. Having a model for the environmental challenges is crucial in defining clear requirements on the system as well as to enable verification and validation (V&V) efforts of the product. This project sets out answer the question: *How to know when the verification and validation of an ADS is complete?* The strategy to argue for completeness needs to be efficient to be industrially viable. Further, a way to leverage this strategy to support continuous safety argumentation will be investigated.



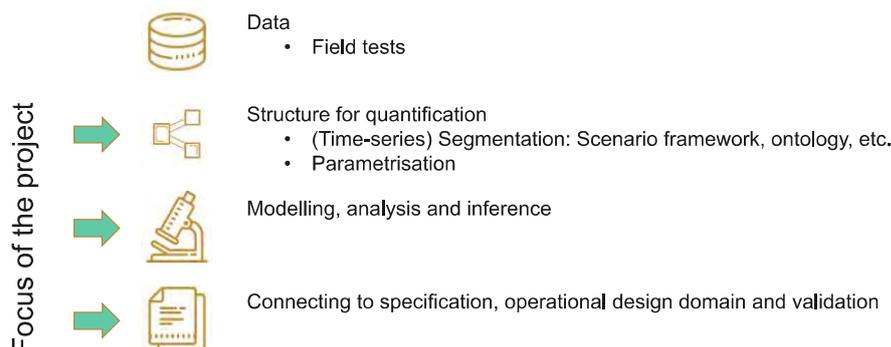
Why is it important to understand the environmental challenges of the ADS? If we rely on the laws of physics we will have a conservative ADS with high cost and low availability. Rule-based methods, on the other hand, are simple but might come with an elevated safety risk as the actual challenges might differ from what the rules postulate. Using data to estimate what is statistically probable will not only make the development more efficient (as compared to physical models), but it will also make it safer than rule-based methods.

The statistical models are constructed using collected and analysed data. The models can later be used to judge what is required during validation, be it through field test, test on test tracks or in simulations. Additionally, the models can be used to answer, in detail, what the ADS needs to handle. Thus feeding information into the specification of the system as well as defining its operational design domain (ODD).

To make this possible research needs to be done on how to construct the models, (a) what data should be used? (b) how can the models be connected to the specification or the ODD? and finally, (c) how can this help in achieving efficient safety assurance of the ADS?



What is required to construct the model?



Outcome and impact

- Providing an efficient strategy for safety assurance of the ADS
- Reducing unnecessary safety margins and minimising the residual risk before deployment
- Enabling timely, safe and continuous deployment of ADS features
- Supporting the reuse of argumentation and insights from previous product/feature releases when deploying new features