The SAUNA project — A Breakthrough for the Next Generation of AGVs for Autonomous Transportation in Sweden

Henrik Andreasson, Abdelbaki Bouguerra, Julien Bidot, Marcello Cirillo, Dimitar Dimitrov, Dimiter Driankov, Lars Karlsson, Achim J. Lilienthal, Federico Pecora, Jari Saarinen, Aleksander Sherikov, Todor Stoyanov

Center for Applied Autonomous Sensor Systems
Örebro University, Sweden
todor.stoyanov@oru.se

14.03.2014
Safe AUtonomous NAvigation (SAUNA) — Objectives

**Application:** mobile service vehicles for autonomous transportation in industrial environments
- transportation of goods by one or more vehicles
- time and resource constraints
- non-holonomic vehicles

**Focus of SAUNA:**
- collision avoidance
- flexible operation, accommodate run-time changes
- account for 3D environment
- general, reusable automation

**Realization:** in simulation and in the real world
- proof-of-concept demonstrators of selected functionalities
- deployment on real vehicles to be explored w/ industrial partners
Assumptions on the Environment

- Environment is 3D
- Presence of uncontrolled objects (e.g., humans, other vehicles, ...)
- Changes over time (e.g., gravel piles)
- Whereabouts of objects to transport known online
Assumptions on Vehicles

- Vehicles are non-holonomic $\Rightarrow$ no sideways motion
- “Car-like” vehicles

Vehicles have non-trivial form, e.g., articulated vehicles

Vehicles carry a payload, perform operations
The Fleet Automation Problem

1. The Fleet Automation Problem

2. Mapping, Registration and Localization in the SAUNA project
Current Practice in Fleet Automation

- **Current industrial practice**
  - lack of fleet-wide automation
  - no or cumbersome coordination

- **Fleet behavior hard to **specify** and hard to **verify**

- **Hard to **guarantee** overall mission requirements**
Current Practice in Fleet Automation

Mining (e.g., Atlas-Copco)

Construction (e.g., Volvo CE)

Logistics (e.g., Kollmorgen)

Overarching aim

Develop general methods for fleet automation that can be applied to different industrial domains
Perception in a Nutshell

- **Perception:** compute safe drivable areas, *given*
  - static obstacles (e.g., walls)
  - slow changes (e.g., piles)
  - external requirements (e.g., restricted areas)

- **Mapping and Localization:**
  - compute a map of the environment
  - and use it to localize vehicles
  - while accounting for dynamic entities
Task Allocation in a Nutshell

- **Task allocation**: compute where vehicles should go and when, **given**
  - drivable areas
  - tasks to be performed
  - temporal constraints (e.g., deadlines)
  - overall mission requirements (e.g., costs)
Motion Planning in a Nutshell

- **Motion planning**: compute vehicle paths, **given**
  - drivable areas
  - location goals
  - vehicle kinematic models
Coordination in a Nutshell

- **Coordination**: compute synchronizations that avoid collisions with other vehicles and deadlocks, **given**
  - vehicle paths
  - bounds on speed
Temporal Reasoning in a Nutshell

- **Temporal reasoning:**
  - compute vehicle speeds, given 
  - synchronizations
  - vehicle paths
  - temporal constraints (e.g., deadlines)
Control in a Nutshell

- **Control**: execute vehicle motions, **given**
  - reference paths
  - temporal profiles
  - drivable area
Collision Prediction in a Nutshell

- **Collision prediction**: react to unforeseen moving objects, given
  - observations of object trajectories
  - estimates of object speeds
  - reference trajectory of controlled vehicle
Is The Fleet Automation Problem Hard?

- Techniques exist for solving each sub-problem
- But obtaining mutually feasible solutions is difficult
  - e.g., allocate tasks, then find that vehicles cannot be coordinated
  - e.g., coordinate vehicles, then find that required trajectories are infeasible
  - e.g., vehicles cannot keep up with coordinated trajectories due to un-modeled dynamics

- Our approach: reduce to a Constraint Satisfaction Problem (CSP) whose solutions are feasible trajectories
Is The Fleet Automation Problem Hard?

- Techniques exist for solving each sub-problem
- But obtaining mutually feasible solutions is difficult
  - e.g., allocate tasks, then find that vehicles cannot be coordinated
  - e.g., coordinate vehicles, then find that required trajectories are infeasible
  - e.g., vehicles cannot keep up with coordinated trajectories due to un-modeled dynamics

- Our approach: reduce to a Constraint Satisfaction Problem (CSP) whose solutions are feasible trajectories
**Constraint-Based, Least Commitment Approach**

- **Idea:** impose *increasingly tight* spatial and temporal *constraints* on trajectories

- Constraints *exclude* trajectories that are *kinematically infeasible* and/or lead to *collisions/deadlocks*

- Choose specific trajectories *at execution time*, revise online *when needed*
Representing Trajectories as Constraints

[F. Pecora, M. Cirillo, D. Dimitrov, IROS 2012]

- **Trajectory:** $p(\sigma(t))$ ($\sigma$ = time history along path)

- **Trajectory envelope:** spatial constraints on $p$, temporal constraints on $\sigma$
Representing Trajectories as Constraints

[F. Pecora, M. Cirillo, D. Dimitrov, IROS 2012]

- **Trajectory**: $p(\sigma(t))$ ($\sigma = \text{time history along path}$)
- **Trajectory envelope**: spatial constraints on $p$, temporal constraints on $\sigma$
Representing Trajectories as Constraints

[F. Pecora, M. Cirillo, D. Dimitrov, IROS 2012]

- **Trajectory**: \( p(\sigma(t)) \) (\( \sigma \) = time history along path)
- **Trajectory envelope**: spatial constraints on \( p \), temporal constraints on \( \sigma \)

\[
\begin{align*}
\text{Spatial envelope:} & \quad \mathcal{J}_i \text{ (polygon } i \text{)} \\
& \text{Convex Polyhedral Constraints} \\
\text{Temporal envelope:} & \quad \mathcal{F}_i \\
& \text{Simple Temporal Problem (STP)} \\
& \quad \ell_i \leq e_i - s_i \leq u_i \\
& \quad s_i \leq e_i - s_{i+1} \leq u_{i+1} \\
& \quad \ell_{i+1} \leq e_{i+1} - s_{i+1} \leq u_{i+1} \\
& \quad \ell_{i,i+1} \leq e_i - s_{i+1} \leq u_{i,i+1} \\
\end{align*}
\]
The Fleet Automation Problem

SAUNA Architecture

[H. Andreasson, et al., Robotics and Automation Magazine (to appear)]

- All modules **read and post constraints** on trajectories
- Modules **operate continuously** to remove trajectories that do not satisfy constraints

---

**Perception**
- sensor readings
- spatial constraints

**Task Allocation**
- mission goals
- spatial & temporal constraints

**Motion Planning**
- spatial & temporal constraints

**Coordination**
- obstacle poses & bounding boxes
- temporal constraints

**Collision Prediction**
- spatial & temporal constraints

**Control**
- control actions
- temporal constraints

---

**Constraint-Based Representation (Trajectory Envelopes $\mathcal{E}$)**

- Temporal reasoning

---

T. Stoyanov et al. (MRO Lab, AASS)

SAUNA

14.03.2014 16 / 29
Outline

1. The Fleet Automation Problem

2. Mapping, Registration and Localization in the SAUNA project
The Normal Distributions Transform

- The Normal Distributions Transform (NDT) originally developed for 2D scan registration (Biber and Straßer, 2003)
- The NDT represents space, using a set of Gaussian probability density functions
- Space is partitioned in disjoint voxels (cells)
- A Gaussian pdf $\mathcal{N}_i$, parametrized by a Covariance matrix $\Sigma_i$ and mean $\mu_i$ used to represent space in each cell
- The NDT can be viewed as a grid-based method for estimating a Gaussian Mixture Model
The Normal Distributions Transform

- The Normal Distributions Transform (NDT) originally developed for 2D scan registration (Biber and Straßer, 2003)
- The NDT represents space, using a set of Gaussian probability density functions
- Space is partitioned in disjoint voxels (cells)

- A Gaussian pdf $\mathcal{N}_i$, parametrized by a Covariance matrix $\Sigma_i$ and mean $\mu_i$ used to represent space in each cell
- The NDT can be viewed as a grid-based method for estimating a Gaussian Mixture Model
The Normal Distributions Transform

- The Normal Distributions Transform (NDT) originally developed for 2D scan registration (Biber and Straßer, 2003)
- The NDT represents space, using a set of Gaussian probability density functions
- Space is partitioned in disjoint voxels (cells)

- A Gaussian pdf $\mathcal{N}_i$, parametrized by a Covariance matrix $\Sigma_i$ and mean $\mu_i$ used to represent space in each cell
- The NDT can be viewed as a grid-based method for estimating a Gaussian Mixture Model
Several problems arise if we want to use the NDT representation for more than scan-to-scan registration:

- **Scalability:** the formulation above requires point samples to estimate each Gaussian component $\mathcal{N}_i$
- **No explicit modeling of free space**
- **Agnostic to dynamics and change in the environment**

The NDT-OM framework solves these problems by adding an occupancy component to each cell, formulating a recursive update routine for NDT cells and tracking the consistency of each component with respect to new observations.
Several problems arise if we want to use the NDT representation for more than scan-to-scan registration:

- Scalability: the formulation above requires point samples to estimate each Gaussian component \( \mathcal{N}_i \)
- No explicit modeling of free space
- Agnostic to dynamics and change in the environment

The NDT-OM framework solves these problems by adding an occupancy component to each cell, formulating a recursive update routine for NDT cells and tracking the consistency of each component with respect to new observations.
NDT-OM: Overview
NDT-OM: Overview
NDT-OM: Overview
NDT-OM: Overview
NDT-OM: Overview

A Empty seen empty

B Occupied seen occupied

T. Stoyanov et al. (MRO Lab, AASS)
NDT-OM: Overview

A: Empty seen empty
B: Occupied seen occupied
C: Occupied seen empty (consistent)
NDT-OM: Overview

- A: Empty seen empty
- B: Occupied seen occupied
- C: Occupied seen empty (consistent)
- D: Distribution changed
NDT-OM: Overview

A. Empty seen empty
B. Occupied seen occupied
C. Occupied seen empty (consistent)
D. Distribution changed
E. Occupied seen empty (distribution vanished)
NDT-OM: Overview

A Empty seen empty
B Occupied seen occupied
C Occupied seen empty (consistent)
D Distribution changed
E Occupied seen empty (distribution vanished)
F Empty seen occupied
NDT-OM in dynamic environments

- We implemented and tested the NDT-OM framework on several data sets, including industrially relevant long-term operation
- Resulting in consistent long term operation at real-time update rates
NDT-OM in dynamic environments

- We implemented and tested the NDT-OM framework on several data sets, including industrially relevant long-term operation.
- Resulting in consistent long term operation at real-time update rates.

![Processing time graph](image-url)
Registration and Map Building — Overview

- Most navigation approaches require a map of the environment
- Mapping integrates multiple sensor views in a consistent model
- Very prominent research field, thousands of published articles
- *Registration* is a sub-problem in mapping:

  - Given two sets of points $\mathcal{P}_1$ and $\mathcal{P}_2$
  - Find the transformation $T = (R, t)$, which brings $\mathcal{P}_2$ in alignment with $\mathcal{P}_1$
Registration and Map Building — Overview

- Most navigation approaches require a map of the environment
- Mapping integrates multiple sensor views in a consistent model
- Very prominent research field, thousands of published articles
- *Registration* is a sub-problem in mapping:

- Given two sets of points \( \mathcal{P}_1 \) and \( \mathcal{P}_2 \)
- Find the transformation \( T = (R, t) \), which brings \( \mathcal{P}_2 \) in alignment with \( \mathcal{P}_1 \)
Registration and Map Building — Overview

- Most navigation approaches require a map of the environment
- Mapping integrates multiple sensor views in a consistent model
- Very prominent research field, thousands of published articles
- *Registration* is a sub-problem in mapping:

- Given two sets of points $\mathcal{P}_1$ and $\mathcal{P}_2$
- Find the transformation $T = (R, t)$, which brings $\mathcal{P}_2$ in alignment with $\mathcal{P}_1$
3D-NDT Distribution-to-Distribution Registration

[T. Stoyanov, M. Magnusson, A. J. Lilienthal, IJRR 2012]

- Compute the 3D-NDT both scans — $P_1$ and $P_2$
- Compute the likelihood of $M_{NDT}(P_2)$, given $M_{NDT}(P_1)$
- Find (local) maximum, using Newton’s method and analytical derivative expressions
A Simple 1D Example — 3D-NDT D2D

Figure: 3D-NDT Distribution to Distribution (D2D)
A Simple 1D Example — 3D-NDT D2D

Figure: 3D-NDT Distribution to Distribution (D2D)
A Simple 1D Example — 3D-NDT D2D

Figure: 3D-NDT Distribution to Distribution (D2D)
A Simple 1D Example — 3D-NDT D2D

Figure: 3D-NDT Distribution to Distribution (D2D)
ML mapping and tracking approach

We use NDT-OM for representation and NDT-D2D for registration

Register new measurements to a global NDT-OM (frame-to-model)

Achieve unlimited area coverage using submap tiling
NDT-OM Fuser Results

- Stable, near real-time tracking in industrially relevant scenarios.
NDT-OM and Localization

In industrial applications we are often more interested in localization than in SLAM.

NDT-OM can be readily used as a likelihood field: ideal for Monte Carlo localization.

Implementation using a particle filter to fuse odometry data and the D2D likelihood of points, given the NDT-OM map.
NDT-OM MCL Results

- We implemented the NDT MCL scheme and used it on board of one of our industrial test vehicles.
- Several different tests, measuring performance in dynamic environments and pose estimate smoothness.
- We used the pose estimate in the control loop and obtained results on par with the ground truth reflector-based localization.
NDT-OM MCL Results

- We implemented the NDT MCL scheme and used it on board of one of our industrial test vehicles.
- Several different tests, measuring performance in dynamic environments and pose estimate smoothness.
- We used the pose estimate in the control loop and obtained results on par with the ground truth reflector-based localization.
ROS packages and tutorials

- All of the algorithms discussed were implemented in the ROS framework and are available as open source packages at http://wiki.ros.org/perception_oru
Summary

The major scientific outcomes from SAUNA:

- An overall framework for vehicle fleet automation, based on CSP.
- Deployed and tested in small scale scenarios.
- To be integrated by our industrial partners in future products.
- A large volume of research in each of the major system components: Mapping/Localization, Perception, Motion Planning, Task Allocation, Coordination and Control.
- Overall, 5 journal and 12 conference publications, several more in preparation for publication.
Thank You!
The SAUNA project — A Breakthrough for the Next Generation of AGVs for Autonomous Transportation in Sweden

Henrik Andreasson, Abdelbaki Bouguerra, Julien Bidot, Marcello Cirillo, Dimitar Dimitrov, Dimiter Driankov, Lars Karlsson, Achim J. Lilienthal, Federico Pecora, Jari Saarinen, Aleksander Sherikov, Todor Stoyanov

Center for Applied Autonomous Sensor Systems
Örebro University, Sweden
todor.stoyanov@oru.se

14.03.2014