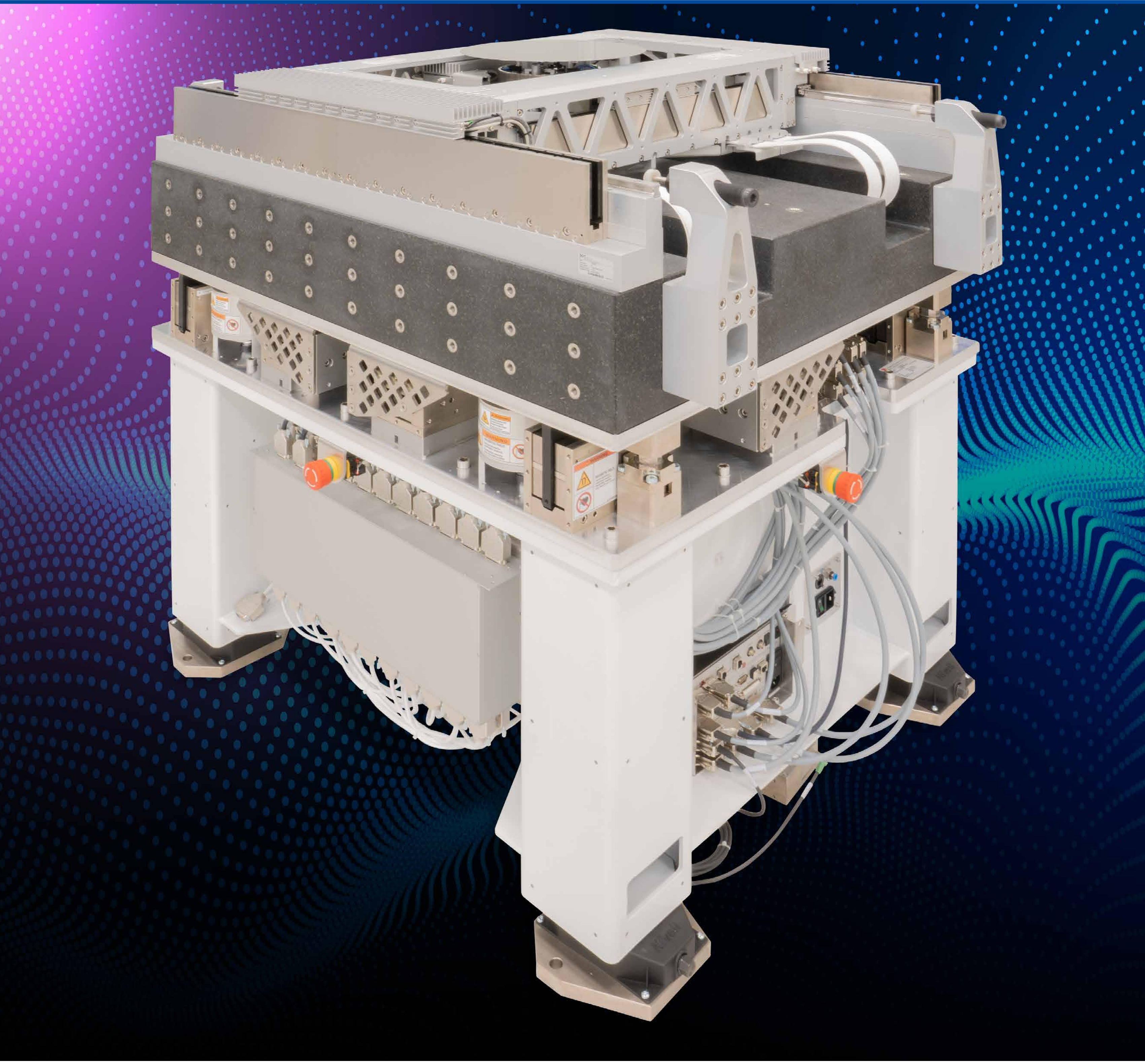
DSPE MIKRONIEK

2023 (VOL. 63) ISSUE 1

PROFESSIONAL JOURNAL ON PRECISION ENGINEERING



- **THEME: HIGH-TECH SYSTEMS**
- **MECHATRONIC MACHINE LEARNING**
- **FORMNEXT HIGHLIGHTS AND OTHER AM DEVELOPMENTS**
- **TESTING ROS2 MOTION CONTROLLERS FOR MOBILE ROBOT NAVIGATION**

MIKRONIEK IS A PUBLICATION OF THE DUTCH SOCIETY FOR PRECISION ENGINEERING – WWW.DSPE.NL

PUBLICATION INFORMATION

Objective
Professional journal on precision engineering and the official organ of DSPE, the Dutch Society for Precision Engineering. Mikroniek provides current information about scientific, technical and business developments in the fields of precision engineering, mechatronics and optics. The journal is read by researchers and professionals in charge of the development and realisation of advanced precision machinery.

DSPE
YOUR PRECISION PORTAL

Publisher
DSPE
Julie van Stiphout
High Tech Campus 1, 5656 AE Eindhoven
PO Box 80036, 5600 JW Eindhoven
info@dspe.nl, www.dspe.nl

Editor
Hans van Eerden, hans.vaneerden@dspe.nl

Advertising canvasser
Gerrit Kulsdom, Sales & Services
+31 (0)229 – 211 211, gerrit@salesandservices.nl

Design and realisation
Drukkerij Snep, Eindhoven
+31 (0)40 – 251 99 29, info@snep.nl

Subscription
Mikroniek is for DSPE members only.
DSPE membership is open to institutes, companies, self-employed professionals and private persons, and starts at € 80.00 (excl. VAT) per year.

Mikroniek appears six times a year.

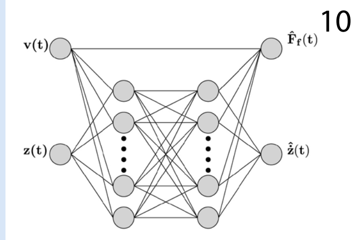
ISSN 0026-3699

The cover image (featuring the VEGA motion platform) is courtesy of PM. Read the article on page 23.

IN THIS ISSUE

THEME: HIGH-TECH SYSTEMS

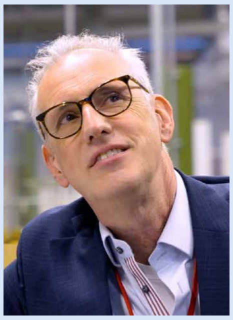
10



27

FEATURES

"IS IT LIVE OR IS IT MEMOREX?"



As in many places around the world, there's an ongoing discussion at Fontys about testing students' skills in relation to popular AI-chatbots such as ChatGPT, Chatsonic and YouChat. Chatbots have become increasingly popular with students to help them complete their assignments. Especially with those left to the last minute, they ensure certain rescue. The controversy revolves around issues such as the authenticity of assignments and rewarding students when they have not developed the ability to write a good essay themselves.

These conversations gave me déjà vu about the distress of teachers when I was a student myself. In the 1970s, electronic calculators were forbidden at primary school because using them jeopardised the development of mental arithmetic. Somewhat later, in the early 1980s, computers became available and as students we applied the Monte Carlo method for simulating tricky problems we needed to solve. Arguments concerning the inauthenticity of students' work and their lacking the capabilities to solve these problems statistically echoed down the hallways. Not much later, during a traineeship, I used the finite-element method to model delicate contact terminals for miniature electrical connectors, which again led to similar discussion: was it my work or the computer's?

In those debates about digital aids, the disagreement among professors on the subject was most interesting. Some of them had a hard time accepting such aids, while others considered them a positive addition to a modern toolbox. There were quite a few more ICT developments back then, including word processors, spreadsheets and CAD systems, although these tools were less intrusive in their contribution to knowledge.

All in all, there was enough for a good exchange of opinions from a didactic perspective. Being students, we withdrew from the discussion. For myself, I decided that with the aid of these modern tools I was able to deliver better assignments in respect to content, quality and time. Why would I, a modern engineer in the making, limit myself to old-school methods and not take advantage of these tools that were going to be commonplace in my future career?

So, now we are dealing with chatbots, again aids that are changing the playing field. A deeper inspection shows that they have been developed primarily to deal with language models and that is exactly what they do best, handle language. When I tried to play a simple game of tic-tac-toe with one of the bots, I won every game easily. At one point, the system did not even realise that I had already won. When I perceptively pointed that out, it politely excused itself for the mistake of not crediting me with my victory.

These systems do not appear intelligent necessarily, but they indeed are polite and well spoken. It turns out that AI-chatbots are computer software after all. However, after winning at chess, go and the American quiz Jeopardy, computers now have the added capability of handling text, including grammar. Again, it brings them a step closer to passing the Turing test.

Where it will end is a question that can only tickle fantasy. Technological developments follow each other quickly. I remember a TV commercial in my youth in which a Memorex cassette tape played the music of Ella Fitzgerald. The question posed was "Is it live or is it Memorex?", suggesting that the recording was so true to life that no one could tell the difference.

We all know what happened with Memorex; like many great inventions that relied on precision engineering, cassette tapes were replaced by CDs, and CDs eventually by online streaming. The same will happen to the current chatbots. They will be replaced by better chatbots, self-learning systems, intelligent image processors and later in time by intelligent robots, humanoids or even something bigger. Still, the basic question remains: why of all things would we as engineers, in the continuing process of our development, limit ourselves to old-school methods...

Erik Puik
Professor of SMART Manufacturing, Fontys University of Applied Sciences
erik.puik@fontys.nl, www.fontys.edu

PUTTING MOTION CONTROLLERS TO THE TEST

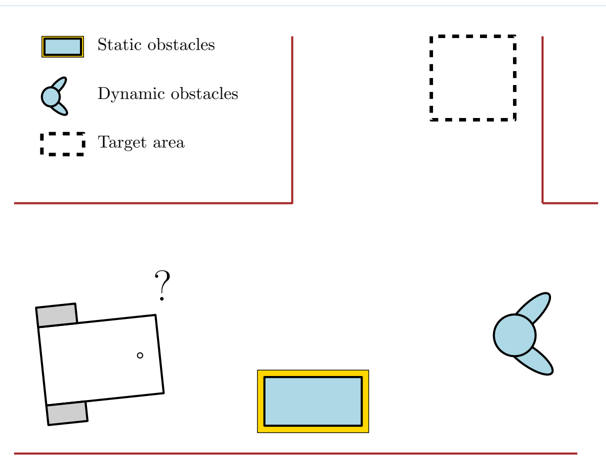
There is an increasing need for deploying Autonomous Mobile Robots (AMRs) in the care sector, the public domain (hospitality and surveillance) and the industry (inspection, maintenance, logistics and agriculture). One of the challenges when developing mobile robot navigation technology is selecting the right, application-specific motion controller. The different motion controllers available in the ROS2 robot operating system framework have been tested on two robots with a different footprint performing various tasks. The results have been translated into a concise selection guideline.

CÉSAR LÓPEZ, MUKUNDA BHARATHEESHA, BRAM ODROSSLIJ AND FRANK SPERLING

Introduction

AMR technology evolved to solve issues associated with AGVs (Autonomous Guided Vehicles), such as docking accuracy, localisation, navigation in unknown environments, and obstacle avoidance. One of the main enablers that has accelerated the growth of AMR technology is the open-source Robot Operating System, i.e. the ROS (latest version ROS2) middleware framework.

In this article, the focus is on mobile robot navigation technology (see Figure 1), which must ensure a safe, reliable and efficient operation in an environment with fixed structures (walls, doors, etc.), static yet movable obstacles and dynamic obstacles such as people. Motion planning algorithms deal with the problem of finding a sequence of velocity and steering commands that will result in the mobile robot successfully reaching the desired target, subject to certain constraints and performance criteria, such as maximum time, velocity and accuracy.



1

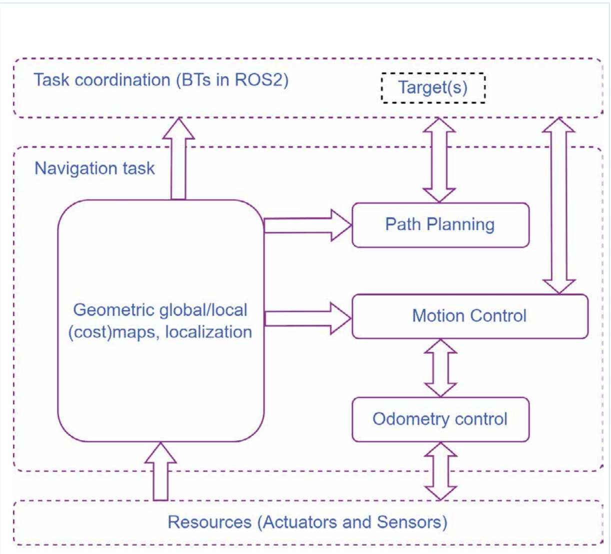
The mobile robot navigation problem: how to generate a set of velocity and steering commands to let the robot reach its target area?

ROS2 navigation architecture

ROS2 navigation uses a modular, configurable architecture that is gaining momentum as a standard for mobile robot navigation, comprising three main blocks: environment representation, global path planning, and local motion control; see Figure 2.

Environment representation

Global and local cost maps are used to represent the fixed environment, as well as the static and dynamic obstacles. Exact collisions cannot be detected at this stage since they depend on the robot's orientation. The fixed infrastructure is typically represented in the global cost map, which uses the map of the environment as input. The static and dynamic obstacles are represented in the local cost map, which uses inputs from sensors such as lidar, sonar and 3D cameras.



2

ROS2 robot navigation architecture.

AUTHORS' NOTE

César López (robotics designer), Mukunda Bharatheesha (robotics engineer) and Frank Sperling (director Technology) all work at Nobleo Technology in Eindhoven (NL), an engineering firm specialised in autonomous intelligent systems. Bram Odrosslij performed his M.Sc. graduation project as a student of Mechanical Engineering (Eindhoven University of Technology) at Nobleo Technology. This article presents a summary of their whitepaper "Mobile Robot Navigation in ROS2 – Motion Controllers Comparison", which can be found downloaded at www.nobleo-technology.nl/news/whitepaper-mobile-robot-navigation-ros2/. References can be found in the whitepaper.

frank.sperling@nobleo.nl
www.nobleo-technology.nl

Global path planning
Path planning is in charge of finding a collision-free path, i.e. a sequence of poses (positions and orientations) that connect a starting pose to a target pose. The path planner uses the global cost map together with the geometric footprint of the robot to assess whether it will be in collision with the fixed environment. In addition, path planners take into account the robot's kinematic constraints; for example, the most common driving mechanism in industrial robots is the differential drive, which cannot drive the robot sideways.

For path planning there exist multiple algorithms, from classical A* and Rapidly-exploring Random Trees to machine-learning-based path planning.

Local motion control
Motion control is in charge of generating the actual velocity commands, using the generated global path to guide the robot towards the target pose along a sequence of poses known as the local trajectory. Many motion controllers deal internally with unknown obstacles and generate velocities that drive the robot around them, resulting in local trajectories that can largely deviate from the original global path (Figure 3). An alternative is to request global path re-planning.

Execution
The high-level component task coordination is thus in charge of harmonising path planning and motion control to complete the navigation task. Finally, odometry control makes sure that the velocity and steering commands are properly executed in the robot.

ROS2 motion controllers

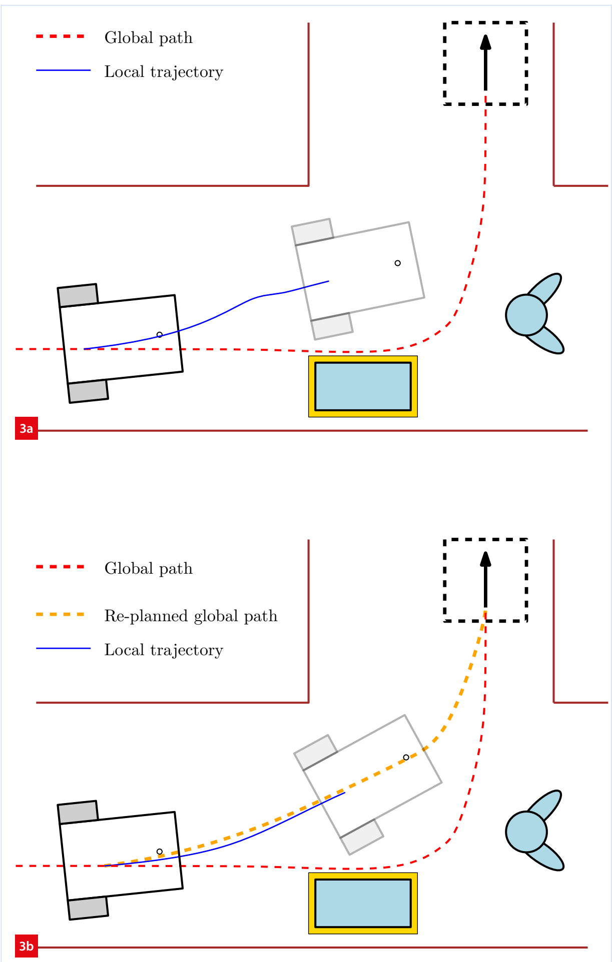
In ROS2 a number of motion controllers are available.

DWB: enhanced Dynamic Window Approach
Fundamentally, DWB is a modularised and enhanced version of the Dynamic Window Approach (DWA), featuring a configurable selection of (customisable) scoring functions, which can increase the efficiency of the controller and help to prevent navigation failures.

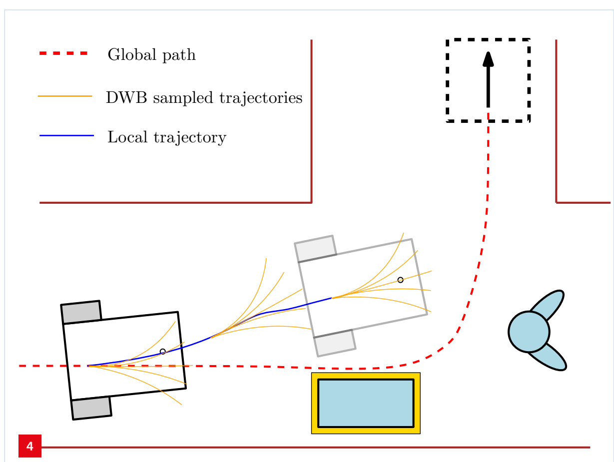
In its basic form, DWA/DWB (Figure 4) uses a trajectory generation and selection approach in an iterative process comprising four steps:

1. Discretely sampling the robot's control space.
2. Performing a forward simulation of each sampled control to predict its effect.
3. Scoring each resulting trajectory, using a metric that incorporates characteristics such as proximity to obstacles, proximity to the goal, proximity to the global path, and velocity.
4. Picking the highest-scoring trajectory and using the associated controls.

DWA has an inherent ability to deal with dynamic obstacles.



For previously unknown obstacles, two approaches can be followed.
(a) The motion controller planning a new trajectory.
(b) The path planner generating a new global path.



DWB strategy: discrete sampling of the robot control space followed by a selection based on several criteria.

ESTIMATING BALL-BEARING FRICTION

To date, Machine Learning (ML) has found only relatively limited implementation in high-end mechatronic systems. To test its capabilities in a real industrial application, the challenging case of friction estimation was investigated. A model was developed for predicting frictional properties of linear ball bearings, for very small displacements. The resulting ML model performed well during training and validation, but rather less so in stand-alone operation. ML is a promising tool for friction estimation, but clearly there is room for improvement in algorithm development.

JORN VEENENDAAL

Introduction

The field of Machine Learning (ML), a subset of Artificial Intelligence (AI), has seen impressive development and growth over the last decade. ML's astonishing rise to prominence within many disciplines is no doubt aided by its numerous impressive accomplishments, building on the ability of ML algorithms to improve themselves automatically through acquiring and processing experience [1]. Examples are DeepMind's AlphaGo algorithm that defeated the world champion in the game of go in 2016, OpenAI's ChatGPT's natural language processing capability, and DeepMind's AlphaFold that solved the long-standing protein folding problem in 2020.

Despite its many achievements, ML has seen only relatively limited, cautious implementation in the field of mechatronics. For this reason, MI-Partners, which specialises in the development of high-end mechatronic systems, decided to start a project to test the current capabilities of ML in a real industrial application. The research question prompted by this ambition was directed towards a practical use case: can an ML model reliably predict the friction of a linear ball bearing?

This article describes the development of an ML algorithm for this use case. First, the friction model is elaborated, followed by a discussion of the learning strategy and the algorithm selection, after which the detailed algorithm design is presented. To conclude, the algorithm is tested, with a direct comparison with the friction model concerned, the LuGre model.

Modelling friction

Currently, a Lund-Grenoble (LuGre) [2] friction model is a good method of modelling the friction behaviour of linear ball bearings. Although the model is accurate when well tuned, the reason for undertaking this investigation was to discover whether it is possible for an ML algorithm to approach the performance of the LuGre model predictions, and possibly overcome the LuGre shortcomings in parameter sensitivity and inability to model advanced friction properties, such as a change in viscosity or the creation of oil bumps.

The LuGre model, as described by Equations 1 to 3, builds upon the principles of the Dahl friction model [3]. It is governed predominantly by the relative velocity $v(t)$ and the hidden state variable $z(t)$. This $z(t)$ cannot be measured practically, but it can be interpreted as the averaged bristle deformation (see Figure 1), where a bristle describes a small connection between two sliding objects.

The bristle deformation, partially described by $g(v)$ in Equation 3, is based on constants such as the Coulomb and Stribeck force, F_c and F_{st}, respectively. Combining the scaled contribution of the bristle deflection and its derivative with respect to time, $\dot{z}$, with the viscous damping term $f(v) = \sigma_2 v$ defines the total predicted friction F_f and with it the benchmark for the ML friction model.

$$\dot{z} = v - \sigma_0 \frac{|v|}{g(v)} z \quad (1)$$

$$F_f = \sigma_0 z + \sigma_1 \dot{z} + f(v) \quad (2)$$

with

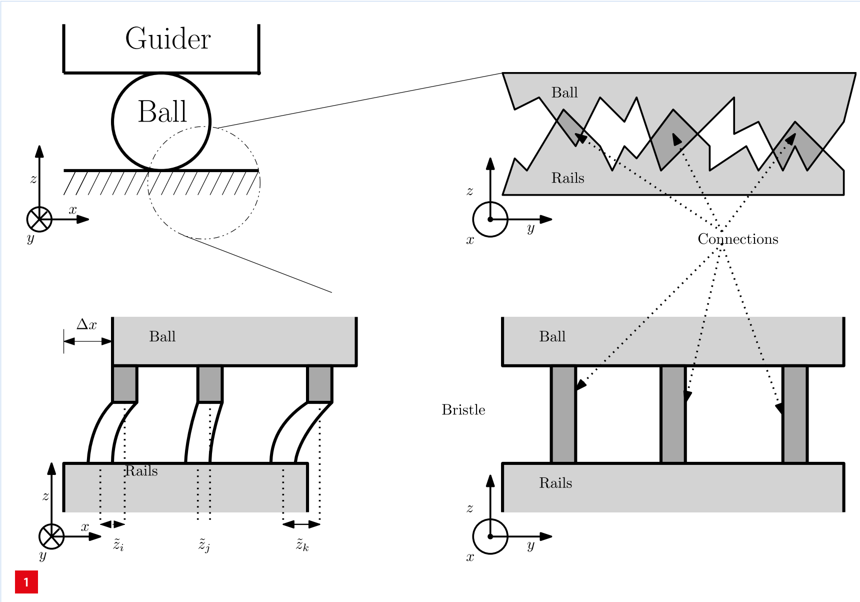
$$g(v) = F_c + (F_{st} - F_c) e^{-\left(\frac{v}{v_{st}}\right)^2} \quad (3)$$

Here σ_0, σ_1 are the material property coefficients for stiffness and damping, respectively. The constant v_{st} represents the characteristic velocity of the velocity-friction force.

AUTHOR'S NOTE

Jorn Veenendaal is a graduate student in Systems and Control at Eindhoven University of Technology, Eindhoven (NL). He studied Mechatronics Engineering at Fontys Engineering University of Applied Sciences in Eindhoven and performed his graduation work at MI-Partners in Veldhoven. For his thesis he received a nomination for the Wim van der Hoek Award 2021. He acknowledges the support and feedback of Wilco Pancras, software architect at MI-Partners.

jt.veenendaal@hotmail.com
www.mi-partners.nl



1

Graphical representation of bristle deformation [4].

Machine Learning strategy

ML algorithms are often characterised by their learning strategy. There are three main learning strategies and the differences between them derive from the way they allow an ML algorithm to extract patterns from a particular data set.

The first strategy concerns supervised learning, which encompasses algorithms that learn to improve their output predictions based on some input by comparing the output to a known true answer in a given training set [5] and using the resulting error to improve the ML model.
Another strategy involves unsupervised learning, its aim being to unravel the structure that underlies the given set of data [6]. Finally there is reinforcement learning, which distinguishes itself from the other strategies by its emphasis on an agent learning by direct interaction with its environment, without relying on exemplary supervision or complete models of the environment [7].

In the current use case, the nature of the generally well-understood phenomenon of friction did not favour the use of unsupervised strategies. That is because unsupervised learning specialises mainly in finding the main features that govern the observed behaviour in a data set. As centuries of research into the subject of friction have already provided most of the relevant information, it would be unwise (and inefficient) to neglect this knowledge. Furthermore, the required input and output data for the friction use case could be generated on demand via the LuGre model or physical experiments. This would create an abundance of representative training data sets.

Likewise, reinforcement learning was deemed a plausible but more complex and demanding solution compared to supervised learning. Altogether, supervised learning was selected as the most promising strategy for the friction use case, which paved the way for the next step in the project: algorithm selection.

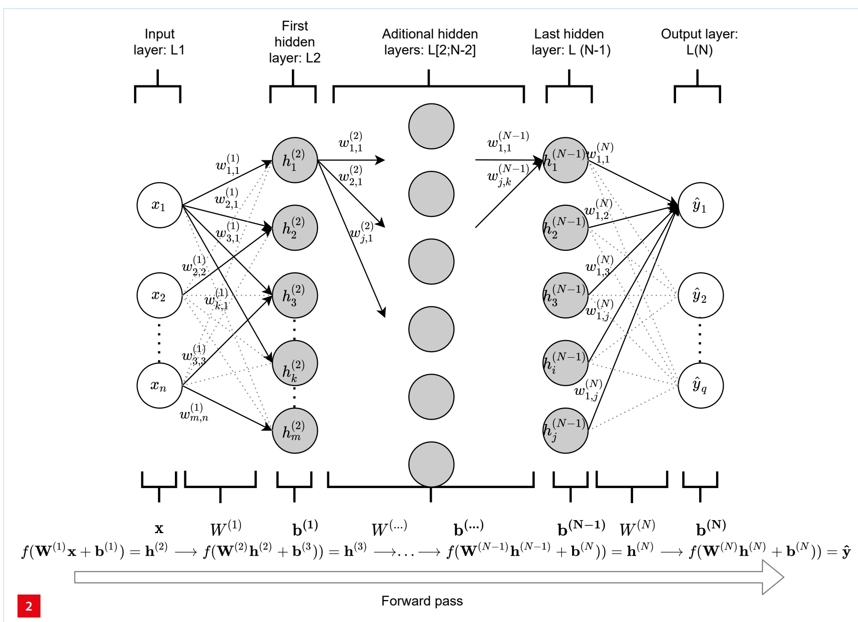
Algorithm selection

Within the strategy of supervised learning [8], three different algorithms were considered as serious candidates for this use case. The algorithms considered – K-Nearest Neighbour (KNN), Support Vector Regression (SVR) and Artificial Neural Networks (ANN) – are all well suited for the inherent regression problem type of this use case.

An ANN-based algorithm (for a schematic representation see Figure 2) was deemed best suited. ANNs are proven universal function approximators [9]. It is this property, combined with various examples that show they can solve differential equations [10] [11] and systems with hysteresis [12], that made ANNs a good candidate. Although the KNN and SVR were also considered, neither of these options was commonly associated with this type of problem, which made them a less promising solution compared to the ANN algorithm. With the algorithm selected, the next logical phase in the project was the detailed design.

Detailed design

Four sequential steps were taken for the design of the final ML algorithm: model architecture selection, data creation, model regularisation and hyperparameter tuning.
For the ML-model architecture, it was concluded that the ML model should have two inputs: the current velocity of the reference profile imposed on the system with friction,



2

Schematic representation of the main operating principle of an ANN: an interconnected group of perceptrons (artificial neurons) that are typically aggregated into layers. The mathematical representation on the bottom includes the nonlinear activation function f(·), which produces the output per layer.