Grant 1183822N

Project Name FWO DMP - Grant 1183822N

Project Identifier u0123309

Grant Title 1183822N

Principal Investigator / Researcher Brecht Evens

Project Data Contact +32476409846, brecht.evens@kuleuven.be

Description In recent years the industrial and robotics society have gained large interest in the decision-making and planning for autonomous systems, interacting directly with humans. Traditionally, this problem has been addressed in a sequential manner, first predicting the future human trajectories and subsequently planning to remain out of their way. This approach leads to overly conservative and defensive behavior since the mutual interactions between the system and its environment are neglected. In this project we will address this issue by performing prediction and planning simultaneously, using game-theoretic models to represent the strategic interactions between players. This approach poses two inherent challenges, namely finding solutions, i.e. Nash equilibria, of these dynamic models and dealing with the unknown human intentions in this framework. To do this we will develop a theoretical and methodological framework for interaction-aware, learning control of autonomous systems. The main contributions of this project will be (i) numerically sound, globally convergent algorithms for solving the emerging Nash equilibrium finding problems, (ii) online learning schemes for learning players' intentions from expert demonstrations and (iii) attractive demonstrations of the practical viability of our approach.

Institution KU Leuven

1. General Information

Name applicant

Brecht Evens

FWO Project Number & Title

FWO research project 1183822N: "Learning to interact safely with humans: A game-theoretic control framework for next-generation autonomous systems"

Affiliation

- KU Leuven

Department of Electrical Engineering (ESAT), STADIUS Center for Dynamical Systems, Signal Processing and Data Analytics

2. Data description

Will you generate/collect new data and/or make use of existing data?

- Generate new data

Describe in detail the origin, type and format of the data (per dataset) and its (estimated) volume. This may be easiest in a table (see example) or as a data flow and per WP or objective of the project. If you reuse existing data, specify the source of these data. Distinguish data types (the kind of content) from data formats (the technical format).

WP 1: Algorithms for solving GNEPs in static games

The main objective in this WP is the development of powerful, globally convergent algorithms for solving GNEPs in static games with shared constraints.

During the numerical validation of these algorithms, we will collect data regarding the convergence rate on toy example problems.

Type of data	Format	Volume	How created
Algorithm iterates	.dat	1-10MB	Collected algorithm iterates on toy example problems using Python.
Code	.py .mat	0-1MB	Python and MATLAB code for solving GNEPs with shared constraints.

WP2: Algorithms for solving GNEPs in dynamic games

The main objective in this WP is the development of powerful, globally convergent algorithms for solving GNEPs in dynamic games with shared constraints. We will exploit the particular structure emerging in these games to obtain efficient algorithms for their solution, which will enable performing interaction-aware motion planning in embedded applications.

During the numerical experiments, we will collect data on the performance of our algorithms in simulated motion planning scenarios.

Type of data	Format	Volume	How created
Algorithm and motion planning iterates, including - computed actions of each player - computation times (CPU and wall time) - cost values and constraint violations - states of each player, such as positions, velocities, accelerations, angles...	.dat	5-50MB	Collected algorithm iterates from simulated motion planning scenarios using Python.
Python pickle of the algorithm and motion planning iterates	.pkl	1-10MB	Collected algorithm iterates from simulated motion planning scenarios using Python and saved to a pickle for easy reloading into Python.
Statistical data (average, standard deviation, min and max value) of - computation times (CPU and wall time) - cost values and constraint violations	.csv	1-10MB	Analyzed algorithm iterates from simulated motion planning scenarios using Python.
Screenshots of the motion planning iterates	.png	5-50GB	Saved images of motion planning iterates using the Pyglet package in Python.
Videos of the motion planning iterates	.mp4	1-5GB	Generated videos from screenshots of motion planning iterates using ffmpeg.
Simulator code	.py	0-10MB	Python simulator for motion planning.

WP3: Efficient online algorithms for interaction-aware, learning control
The main objective in this WP is to introduce learning schemes allowing autonomous systems to learn the unknown behaviour of human players in an online fashion during interaction.

During the numerical experiments, we will collect data on the performance of our learning schemes in simulated motion planning scenarios, where one player attempts to learn the cost function and constraints of another player in an online fashion.

Type of data	Format	Volume	How created
Algorithm and motion planning iterates, including - computed actions of each player - computation times (CPU and wall time) for both the GNEP solver and the learning scheme - cost values and constraint violations of each player - the beliefs of each player in the cost function and constraint parameters of other players - states of each player, such as positions, velocities, accelerations, angles...	.dat	5-50MB	Collected algorithm iterates from simulated motion planning scenarios using Python.
Python pickle of the algorithm and motion planning iterates	.pkl	1-10MB	Collected algorithm iterates from simulated motion planning scenarios using Python and saved to a pickle for easy reloading into Python.
Statistical data (average, standard deviation, min and max value) of - computation times (CPU and wall time) for both the GNEP solver and the learning scheme - cost values and constraint violations of each player	.csv	1-10MB	Analyzed algorithm iterates from simulated motion planning scenarios using Python.
Screenshots of the motion planning iterates	.png	5-50GB	Saved images of motion planning iterates using the Pyglet package in Python.
Videos of the motion planning iterates	.mp4	1-5GB	Generated videos from screenshots of motion planning iterates using ffmpeg.
Additional simulator code	.py	0-5MB	Additional code for learning schemes in Python simulator for motion planning.

Additionally, the performance of our learning scheme might also be tested on existing datasets that are freely available on the internet such as the Highway Drone Dataset (https://www.highd-dataset.com/).

WP4: Validation in an experimental setting
The main objective in this workpackage is the experimental validation of the interaction-aware (learning) control developed in WP2 and WP3. To this end, our methods will be applied on an experimental setup for autonomous driving, where we will collect the following data:

Type of data	Format	Volume	How created
Algorithm and motion planning iterates, including - computed actions of each player - computation times (CPU and wall time) for the GNEP solver, the learning scheme and the image processing task - cost values and constraint violations of each player - the beliefs of each player in the cost function and constraint parameters of other players - state estimates of each player, such as positions, velocities, accelerations, angles...	.dat	5-50MB	Collected algorithm iterates from the experimental setup using Python.
Python pickle of the algorithm and motion planning iterates	.pkl	1-10MB	Collected algorithm iterates from simulated motion planning scenarios using Python and saved to a pickle for easy reloading into Python.
Statistical data (average, standard deviation, min and max value) of - computation times (CPU and wall time) for both the GNEP solver and the learning scheme - cost values and constraint violations of each player	.csv	1-10MB	Analyzed algorithm iterates from the experimental setup using Python.
Images of the motion planning iterates	.png	5-50GB	Saved images using the Python camera interface.
Videos of the motion planning iterates	.mp4	1-10GB	Saved videos using the Python camera interface.
Additional simulator code for interfacing with vehicles	.py	0-5MB	Additional code for interfacing with vehicles and performing state estimation based on the Python simulator framework.

3. Legal and ethical issues

Will you use personal data? If so, shortly describe the kind of personal data you will use. Add the reference to your file in KU Leuven's Register of Data Processing for Research and Public Service Purposes (PRET application). Be aware that registering the fact that you process personal data is a legal obligation.

- No

Are there any ethical issues concerning the creation and/or use of the data (e.g. experiments on humans or animals, dual use)? If so, add the reference to the formal approval by the relevant ethical review committee(s)

- No

Does your work possibly result in research data with potential for tech transfer and valorisation? Will IP restrictions be claimed for the data you created? If so, for what data and which restrictions will be asserted?

- No

Do existing 3rd party agreements restrict dissemination or exploitation of the data you (re)use? If so, to what data do they relate and what restrictions are in place?

- No

4. Documentation and metadata

What documentation will be provided to enable reuse of the data collected/generated in this project?

1. For the toy examples of WP1, the algorithm iterates will be collected per simulation test and stored into separate subfolders, including a .txt file with a clear description of what the data represent and

how they were generated. The name of the folder will correspond to the name of the corresponding toy example. All subfolders will be stored in a dedicated "experiments" folder, including a readme file.

2. For the numerical experiments of WP2 and WP3, the python code of the simulator environment will automatically create a dedicated subfolder for each new simulation, where the algorithm iterates, the analysis, the screenshots, the videos and the python pickle of the corresponding simulation are stored. Furthermore, each simulation can be investigated and optionally re-performed by reloading the pickle into Python. All subfolders will be stored in a dedicated "simulations" folder, including a readme file.

3. For the numerical experiments of WP4, the python code of the simulator environment will be extended, such that a dedicated subfolder for each new numerical experiment is created, where the algorithm iterates, the analysis, the camera images, the videos and the python pickle of the corresponding experiment are stored. Furthermore, each experiment can be investigated by reloading the pickle into Python. All subfolders will be stored in a dedicated "experiments" folder, including a readme file.

Will a metadata standard be used? If so, describe in detail which standard will be used. If no, state in detail which metadata will be created to make the data easy/easier to find and reuse.

- Yes

As we are mainly dealing with data from numerical simulations, we will use a dedicated metadata standard for this purpose such as the EngMeta standard.

5. Data storage and backup during the FWO project

Where will the data be stored?

1. A time-stamped version of the data will be stored on the STADIUS DATASET Server, which is our research unit central storage server. Copies can be made and kept on personal devices.
2. To allow for efficient collaboration with researchers from other research groups, the data will additionally be stored on GitHub (subject to the Github regulations).

How is backup of the data provided?

All data on the STADIUS DATASET Server are backed up daily and replicated to an off-site storage system housed in the ICTS data center.

Is there currently sufficient storage & backup capacity during the project? If yes, specify concisely. If no or insufficient storage or backup capacities are available then explain how this will be taken care of.

- Yes

The STADIUS DATASET Server has a total capacity of 14.88 TB. The capacity of the dataset server is monitored daily by the ESAT system admins.

What are the expected costs for data storage and back up during the project? How will these costs be covered?

All data storage and back up will be performed on the STADIUS DATASET Server in pre-existing storage facilities of the Department of Electrical Engineering (ESAT) without the need of purchasing new infrastructures.

Data security: how will you ensure that the data are securely stored and not accessed or modified by unauthorized persons?

The access of the data on the STADIUS DATASET Server is regulated by an access control list (ACL) that grants:

- read-write access to the project owner and the FWO fellow
- read-only access to specific users, such as other members of our research group

The ACL is managed by the project owner (Panagiotis Patrinos). Client computers can access the data using:

- SMB2 (or higher) from specific IP ranges
- NFSv4 from specific (IT managed) systems

6. Data preservation after the FWO project

Which data will be retained for the expected 5 year period after the end of the project? In case only a selection of the data can/will be preserved, clearly state the reasons for this (legal or contractual restrictions, physical preservation issues, ...).

All research data detailed previously will be retained for at least 5 years, conform the FWO data preservation policy.

Where will the data be archived (= stored for the longer term)?
The data will be stored on the university's central servers (with automatic back-up procedures) for at least 10 years, conform the KU Leuven RDM policy.

What are the expected costs for data preservation during the retention period of 5 years? How will the costs be covered?
All data hosted on the STADIUS DATASET Server will be stored for a long term in pre-existing storage facilities of the Department of Electrical Engineering (ESAT) without the need of purchasing new infrastructures. Similarly, the data hosted on GitHub will be stored for a long term.

7. Data sharing and reuse

Are there any factors restricting or preventing the sharing of (some of) the data (e.g. as defined in an agreement with a 3rd party, legal restrictions)?

- No

Which data will be made available after the end of the project?
All research articles submitted to journals and conference proceedings will be made publicly available. Furthermore, all relevant source code and datasets will be made publicly available in previously mentioned formats.

Where/how will the data be made available for reuse?

- In an Open Access repository

1. The research articles submitted to journals and conference proceedings will be made publicly available on ArXiv e-prints archive.
2. The source code will be released publicly on GitHub under the MIT license, along with the corresponding datasets.

When will the data be made available?

- Upon publication of the research results

Upon publication of new research results, the corresponding research article will be made publicly available on ArXiv and the corresponding source code and datasets will be released publicly on GitHub.

Who will be able to access the data and under what conditions?
1. The research articles submitted to journals and conference proceedings will be made publicly available on ArXiv e-prints archive under the perpetual, non-exclusive license. Therefore, it will be available to anyone for any purpose, provided that they give appropriate credit to the creators.
2. The source code will be released publicly on GitHub under the MIT license, along with the corresponding datasets. Hence, it will be available for everyone to use, change, and distribute the software, only requiring preservation of copyright and license notices.

What are the expected costs for data sharing? How will the costs be covered?
Since free services like GitHub and ArXiv are used to distribute the research data, there are no expected costs.

8. Responsibilities

Who will be responsible for data documentation & metadata?
The FWO Fellow will be responsible for the data documentation and the metadata.

Who will be responsible for data storage & back up during the project?
The FWO Fellow will be responsible for the data storage. The system administrators and data manager of the research division are responsible for the back up during and after the project.

Who will be responsible for ensuring data preservation and reuse ?
The FWO Fellow will be responsible for ensuring data preservation and reuse.

Who bears the end responsibility for updating & implementing this DMP?
The project owner (Panagiotis Patrinos) bears the end responsibility of updating & implementing this DMP.