A proposed software framework aimed at energy-efficient autonomous driving of electric vehicles

**Authors:** José Luis Torres, José Luis Blanco, Mauro Bellone, Francisco Rodríguez, Antonio Giménez and Giulio Reina

Bergamo, October 23th, 2014
Talk overview

1. Motivation
2. Multibody modeling of an electric car
3. Prototype description
4. Proposed software architecture
5. Experiments
6. Conclusions and future works
1. Motivation
2. Multibody modeling of an electric car
3. Prototype description
4. Proposed software architecture
5. Experiments
6. Conclusions and future works
Autonomous efficient electric vehicle

Platform for vehicle dynamics analysis
- Complex high-fidelity physical models validation
- Efficiency and consumption optimization
- Development of new controllers
- Autonomous driving algorithms implementation
- Software architecture
1. Motivation
2. Multibody modeling of an electric car
3. Prototype description
4. Proposed software architecture
5. Experiments
6. Conclusions and future works
Multibody model. CAD prototype
Multibody model. Kinematics

Hard-Points:
Natural & relative coordinates. Kinematic constraints
Multibody model. Dynamics

**Forces:**
- Gravity
- Spring-dampers
- Tires
- Braking
- Traction
Multibody model. Full vehicle

Summary:

- Nº de variables: $165 \times 3 = 495$
- Nº dof: 15 – guided var. = 14
- Nº of constraints equations: 154
- Size of Jacobian: $154 \times 195$
- Size of Mass matrix: $195 \times 195$
- Size of force vector: $195 \times 1$

$t_{\text{sim}} \gg t_{\text{real}}$

$t_{\text{sim}} \approx t_{\text{real}}$
1. Motivation
2. Multibody modeling of an electric car
3. Prototype description
4. Proposed software architecture
5. Experiments
6. Conclusions and future works
Adapted electric vehicle.

Mechanical characteristics

<table>
<thead>
<tr>
<th></th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Length \times width \times height</td>
<td>2680 \times 1525 \times 1780 mm</td>
</tr>
<tr>
<td>Track</td>
<td>1830 mm</td>
</tr>
<tr>
<td>Front/rear wheelbase</td>
<td>1285/1260 mm</td>
</tr>
<tr>
<td>Weight without/with batteries</td>
<td>472/700 kg</td>
</tr>
</tbody>
</table>

Electric characteristics

<table>
<thead>
<tr>
<th></th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>DC motor XQ 4.3</td>
<td>4.3 kW</td>
</tr>
<tr>
<td>Batteries (gel technology)</td>
<td>8 \times 6 V – 210 Ah</td>
</tr>
<tr>
<td>Autonomy</td>
<td>90 km</td>
</tr>
</tbody>
</table>
Prototype description
Instrumentation. DAQs & controllers
Instrumentation. Sensors.

10000 ppr

360 ppr
Instrumentation. Sensors.
Instrumentation. Sensors.
Instrumentation. Sensors.

18 Hz
Instrumentation. Sensors.

2 kHz
Instrumentation. Sensors.

80 kHz
Instrumentation. Sensors.

80 kHz
1. Motivation
2. Multibody modeling of an electric car
3. Prototype description
4. Proposed software architecture
5. Experiments
6. Conclusions and future works
The selected middle-ware must involve:

- Multi-platform
- Algebra and numerical methods
- Integration of libraries
- Modularity
- Distributed, robust and concurrent execution capable
- Lightweight
- Debugging
Proposals.

- Utilization of a structure of connected modules sharing complex data from sensors (namely images), simulated models and control algorithms.
- We evaluated these alternatives for software frameworks:
  - ROS
  - LabView
  - A MOOS-based framework (OpenMORA)
OpenMORA

Data transmission between modules through pub/sub pattern around milliseconds. Central module called MOOSDB.

Encapsulation of complex types of data. Interfaces for main sensors in robotics.

Modular and distributed software architecture based on MOOS and MRPT which integrates high-level layers (control algorithms) and low-level layers (sensors/actuators).

For more info visit http://openmora.github.io/
### Evaluation of alternatives.

<table>
<thead>
<tr>
<th>Feature</th>
<th>LabVIEW</th>
<th>ROS</th>
<th>OpenMORA</th>
</tr>
</thead>
<tbody>
<tr>
<td>Multi-platform</td>
<td>★★★★★</td>
<td>★★★★★</td>
<td>★★★★★</td>
</tr>
<tr>
<td>Modularity</td>
<td>★★★★★</td>
<td>★★★★★</td>
<td>★★★★★</td>
</tr>
<tr>
<td>Distributed</td>
<td>★★★★★</td>
<td>★★★★★</td>
<td>★★★★★</td>
</tr>
<tr>
<td>Algebra and numerical methods</td>
<td>★★★★★</td>
<td>★★★★★</td>
<td>★★★★★</td>
</tr>
<tr>
<td>Integration of libraries</td>
<td>★★★★★</td>
<td>★★★★★</td>
<td>★★★★★</td>
</tr>
<tr>
<td>Lightweight</td>
<td>★★★★★</td>
<td>★★★★★</td>
<td>★★★★★</td>
</tr>
<tr>
<td>Debugging</td>
<td>★★★★★</td>
<td>★★★★★</td>
<td>★★★★★</td>
</tr>
</tbody>
</table>
Example of a process

Estimation of slip angle:
- Filters
- Physical magnitudes
- Data acquisition
- Sensors
Module: eCarOdometry

Module info:

<table>
<thead>
<tr>
<th>Module name</th>
<th>Short description</th>
<th>Publishes</th>
<th>Subscribes</th>
</tr>
</thead>
<tbody>
<tr>
<td>eCarOdometry</td>
<td>Differential-drive odometry estimation from a pair of encoders in the rear wheels.</td>
<td>ODOMETRY, ODOMETRY_ANG_SPEED, ODOMETRY_INCR, ODOMETRY_SLN_SPEED, ODOMETRY_OBS</td>
<td>ENC1_CH0_AVRG_VEL, ENC1_CH0_COUNT, ENC1_CH1_AVRG_VEL, ENC1_CH1_COUNT</td>
</tr>
</tbody>
</table>

MOOS commands accepted by the module:
- (None)

Parameters accepted in the MOOS mission file:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>left_K</td>
<td>Ticks to meters constant for left wheel encoders.</td>
</tr>
<tr>
<td>odometry_verbose</td>
<td>Set to &quot;true&quot; to display live odometry to console.</td>
</tr>
<tr>
<td>right_K</td>
<td>Ticks to meters constant for right wheel encoders.</td>
</tr>
<tr>
<td>wheels_dist</td>
<td>Wheel-to-wheel distance (meters)</td>
</tr>
</tbody>
</table>

Detailed description:
Differential-drive odometry estimation from a pair of encoders in the rear wheels.
It is assumed that ENC1_CH0_COUNT is the LEFT wheel, ENC1_CH1_COUNT is the RIGHT encoder.

Module graph:
1. Motivation
2. Multibody modeling of an electric car
3. Prototype description
4. Proposed software architecture
5. Experiments
6. Conclusions and future works
Experiments
Experiment I

Encoders and laser scanner. Odometry calibration.
Odometry. Results.
Data grabbing.

**Experiment II**

All sensors.
Testing the architecture.
Data grabbing. Results.

(a) Ángulo de dirección

(b) Ángulo de guía

(c) Velocidad de guía

Trajectory GPS (m)
Model validation.

**Experiment III**

IMU, GPS.
Validation of the model through comparison with real measurements.
Double lane maneuver.
Open loop maneuver.

**Experiment IV**

IMU, GPS, Encoders and NI-DAQ. Repetitions of a maneuver
Open loop maneuver. Results.
1. Motivation
2. Multibody modeling of electric car
3. Prototype description
4. Proposed software architecture
5. Experiments
6. Conclusions and future works
Conclusions

- Evaluation of the accuracy of OpenMORA:
  - DataLogger.
  - Real-time processing of complex data.
  - Execution of controllers, estate observers and physical models.
- Promising results are obtained for the full automation of our electric vehicle prototype.
Thanks for your attention!