Model-based control techniques for automotive applications

Fabio Maran

PhD School on Information Engineering
XXV Series, ICT Section
Advisor: Prof. Alessandro Beghi

28th February 2013
Part I

Model Predictive Control for Motion Cueing
Interest in *dynamic* driving simulators is increasing, with research and application in different areas (rehab, prototyping, driving safety, racing).

**Ferrari simulator**

**Toyota simulator**
Interest in *dynamic* driving simulators is increasing, with research and application in different areas (rehab, prototyping, driving safety, racing).

A successful dynamic simulator platform has to reproduce at best the sensations that the user would have in the real vehicle: this is the task of the **Motion Cueing** (MC) Algorithm:

- deals with *inertial cues*
- strictly related to visual and audio hints

**MC has two purposes:**
- replicate driver’s perception
- keep the platform within its boundaries: **Washout Action**

**MC generates the trajectories** that the platform should track: it acts *before* the *position-control* layer (usually handled by PLC)
Motion Cueing Algorithm: Classic Approach

Standard (classic) approach: combination of passive filters, both high-pass and low-pass filters.

- **Linear accelerations**
  - High-pass filter
  - Double integration
  - High-pass filter
- **Angular accelerations**
  - High-pass filter
  - Double integration
  - Tilt coordination
  - High-pass filter
  - Double integration
  - Addition
  - High-pass filter

Translations
Motion Cueing Algorithm: Classic Approach

Standard (classic) approach: combination of passive filters, both high-pass and low-pass filters

High-pass filters on linear accelerations and rotations are applied to catch the fast dynamics, traduced into small and fast movements of the platform.
Motion Cueing Algorithm: Classic Approach

Standard **(classic)** approach: combination of passive filters, both high-pass and low-pass filters

Double integration is applied to the calculated signals to obtain the linear and angular positions from the accelerations.
Standard **(classic)** approach: combination of passive filters, both high-pass and low-pass filters

Low-pass filters on linear accelerations are used for **tilt coordination**: applying the right visual cues, the low frequencies, progressive accelerations are replicated by *tilting* the platform and exploiting the gravity acceleration.
Motion Cueing Algorithm: Classic Approach

Standard (classic) approach: combination of passive filters, both high-pass and low-pass filters

Washout filters (high-pass with different cut-off frequencies) are then applied to both the linear and rotational signals to assure that physical limits of the platform are not violated.
Motion Cueing Algorithm: Problems

- High-pass filtering (HPF) the accelerations gives origin to motion inversion: variations in accelerations around a positive value become variations in accelerations around the zero value, leading to motion sickness.
Motion Cueing Algorithm: Problems

- High-pass filtering (HPF) the accelerations gives origin to motion inversion: variations in accelerations around a positive value become variations in accelerations around the zero value, leading to motion sickness.

- Low-pass filtering (LPF) can introduce delays with respect to video/audio cues and high-frequency components of the motion (motion sickness).
Motion Cueing Algorithm: Problems

- High-pass filtering (HPF) the accelerations gives origin to motion inversion: variations in accelerations around a positive value become variations in accelerations around the zero value, leading to motion sickness

- Low-pass filtering (LPF) can introduce delays with respect to video/audio cues and high-frequency components of the motion (motion sickness)

- HPF for the Washout Action makes very hard to exploit all the available working area (no explicit constraints); it may also introduces other false cues
Motion Cueing Algorithm: Problems

- High-pass filtering (HPF) the accelerations gives origin to motion inversion: variations in accelerations around a positive value become variations in accelerations around the zero value, leading to motion sickness.
- Low-pass filtering (LPF) can introduce delays with respect to video/audio cues and high-frequency components of the motion (motion sickness).
- HPF for the Washout Action makes very hard to exploit all the available working area (no explicit constraints); it may also introduces other false cues.
- The tuning procedure is counterintuitive (manipulation of filter gains and cut-off frequencies ⇒ no physical meaning).
MPC for MC

New approach proposed: Model Predictive Control (MPC) techniques to control the platform motion

- Scaled translational acceleration and rotational velocities from simulation
- Vestibular model
- MPC based Motion Cueing
- Cost Function
- Optimization
- Platform Control System

**Features**

- **Model-based:** exploits platform (if known) and vestibular system models to calculate reference signals
- **Constrained, optimal problem:** explicit handling of performance and working area constraints
- **Prediction:** if available, it allows to improve the platform performance
Vestibular System

The Vestibular System is composed by two sub-systems

- **Semi-circular channels**: they are responsible for sensing *rotational velocities* applied to the body
- **Otoliths**: they sense *longitudinal accelerations*

\[
\hat{\omega}(s) = 5.73 \frac{80s^2}{(1 + 80s)(1 + 5.73s)}
\]

\[
\hat{a}(s) = 0.4 \frac{(1 + 10s)}{(1 + 5s)(1 + 0.016s)}
\]
The Vestibular System is composed by two sub-systems:

- **Semi-circular channels**: they are responsible for sensing *rotational velocities* applied to the body.
- **Otoliths**: they sense *longitudinal accelerations*.

Mathematically:

\[
\frac{\hat{\omega}(s)}{\omega(s)} = 5.73 \frac{80s^2}{(1 + 80s)(1 + 5.73s)}
\]

\[
\frac{\hat{a}(s)}{a(s)} = 0.4 \frac{(1 + 10s)}{(1 + 5s)(1 + 0.016s)}
\]
Vestibular System

The Vestibular System is composed by two sub-systems

- **Semi-circular channels**: they are responsible for sensing *rotational velocities* applied to the body
- **Otoliths**: they sense *longitudinal accelerations*

\[
\frac{\hat{\omega}(s)}{\omega(s)} = 5.73 \frac{80s^2}{(1 + 80s)(1 + 5.73s)}
\]

\[
\frac{\hat{a}(s)}{a(s)} = 0.4 \frac{(1 + 10s)}{(1 + 5s)(1 + 0.016s)}
\]
Vestibular system

- Under steady condition the otoliths are incapable of distinguishing translational acceleration from gravity force \( \Rightarrow \) rotational and/or visual sensorial information is needed
Under steady condition the otoliths are incapable of distinguishing translational acceleration from gravity force ⇒ rotational and/or visual sensorial information is needed.

This feature can be exploited when developing motion cueing algorithms: low frequency components of the acceleration can be reproduced by using tilt coordination.
Prediction

- Prediction is a crucial aspect of the MPC framework, with immediate advantages for MC applications
  - Better exploitation of the working area (tilt coordination, linear displacement)
  - More effective Washout Action
Prediction

- Prediction is a crucial aspect of the MPC framework, with immediate advantages for MC applications
  - Better exploitation of the working area (tilt coordination, linear displacement)
  - More effective Washout Action
- Real-time application of prediction is a challenging task
  - Human in the loop ⇒ hard to calculate a reliable trajectory
  - A high control frequency corresponds to large prediction horizon $N_P$ ⇒ higher resolution time
Prediction

- Prediction is a crucial aspect of the MPC framework, with immediate advantages for MC applications
  - Better exploitation of the working area (tilt coordination, linear displacement)
  - More effective Washout Action
- Real-time application of prediction is a challenging task
  - Human in the loop $\Rightarrow$ hard to calculate a reliable trajectory
  - A high control frequency corresponds to large prediction horizon $N_P \Rightarrow$ higher resolution time
- Classic MPC approach: constant reference in the prediction window
  - If system dynamics is “slow” enough w.r.t. the control frequency, tracking performance are not compromised
Prediction

- Prediction is a crucial aspect of the MPC framework, with immediate advantages for MC applications
  - Better exploitation of the working area (tilt coordination, linear displacement)
  - More effective Washout Action
- Real-time application of prediction is a challenging task
  - Human in the loop ⇒ hard to calculate a reliable trajectory
  - A high control frequency corresponds to large prediction horizon $N_P$ ⇒ higher resolution time
- Classic MPC approach: constant reference in the prediction window
  - If system dynamics is “slow” enough w.r.t. the control frequency, tracking performance are not compromised
- Supposing to have an available reference, a blocking strategy has been studied to improve the computational performance
  - To deal with a larger time window while keeping $N_P$ low enough, different sampling time steps are considered, increasing while receding from the present time instant
Tuning procedure: proposed approach

The tuning procedure has a key role in Motion Cueing systems.

Tuning is performed by variating the constraints, weights and prediction horizon $N_P$ values.
Tuning procedure: proposed approach

The tuning procedure has a key role in Motion Cueing systems

Tuning is performed by variating the *constraints*, *weights* and *prediction horizon* $N_P$ values

- *constraints* variations help the exploitation of the working area, preventing undesired behaviour (e.g. motion inversion)
Tuning procedure: proposed approach

The tuning procedure has a key role in Motion Cueing systems.

Tuning is performed by variating the constraints, weights and prediction horizon $N_P$ values:

- **constraints** variations help the exploitation of the working area, preventing undesired behaviour (e.g. motion inversion).
- **weights** variations change the dynamic behaviour of the platform (e.g., “penalizing” the position state variables, more conservative performance are achieved without affecting too much velocities and accelerations perceived).
Tuning procedure: proposed approach

The tuning procedure has a key role in Motion Cueing systems

Tuning is performed by variating the *constraints*, *weights* and *prediction horizon* $N_P$ values

- **constraints** variations help the exploitation of the working area, preventing undesired behaviour (e.g. motion inversion)
- **weights** variations change the dynamic behaviour of the platform (e.g., “penalizing” the position state variables, more conservative performance are achieved without affecting too much velocities and accelerations perceived)
- with **constant** prediction reference, the tracking performance is regulated by manipulating $N_P$, giving a more flexible algorithm, easily adaptable to different situations (overcoming the difficult in predicting the driver’s behaviour).
The whole procedure is intuitive: it deals with physical values rather than cut-off frequencies as the classical ones.
Tuning procedure: benefits

- The whole procedure is **intuitive**: it deals with physical values rather than cut-off frequencies as the classical ones.
- This approach improves the **feedback quality** from the driver: the suggestions can be traduced in parameters variations in a more natural way (specifying requirements on linear accelerations, angular positions etc.)
The whole procedure is intuitive: it deals with physical values rather than cut-off frequencies as the classical ones.

This approach improves the feedback quality from the driver: the suggestions can be traduced in parameters variations in a more natural way (specifying requirements on linear accelerations, angular positions etc.)

The implementation allows real-time variations to the parameters (using a Graphical User Interface).
Real-time application: VI-DriveSIM

New approach to dynamic simulator platforms
Real-time application: VI-DriveSIM

New approach to dynamic simulator platforms

- **Compact**: can fit a room (4m long, 5.5m wide, 3m high, considering maximum displacements)
- **Fast Dynamics with less power consumption**: reduced inertia and linear, electric actuators
- **Performance**: similar to most common racing simulators

<table>
<thead>
<tr>
<th>Range</th>
<th>Position</th>
<th>Velocity</th>
<th>Acceleration</th>
</tr>
</thead>
<tbody>
<tr>
<td>x</td>
<td>1m</td>
<td>1.3m/s</td>
<td>3.3m/s²</td>
</tr>
<tr>
<td>y</td>
<td>1m</td>
<td>1.3m/s</td>
<td>3.6m/s²</td>
</tr>
<tr>
<td>z</td>
<td>0.3m</td>
<td>0.9m/s</td>
<td>4.9m/s²</td>
</tr>
<tr>
<td>Roll</td>
<td>30deg</td>
<td>112deg/s</td>
<td>600deg/s²</td>
</tr>
<tr>
<td>Pitch</td>
<td>24deg</td>
<td>61deg/s</td>
<td>600deg/s²</td>
</tr>
<tr>
<td>Yaw</td>
<td>50deg</td>
<td>61deg/s</td>
<td>240deg/s²</td>
</tr>
</tbody>
</table>
Real-time application: implementation

- The application has strict real-time requirements
  - control frequency: 100 Hz
  - sport cars driven by professional drivers ⇒ fast dynamics involved
  - real-time tuning

Quadratic cost function: quadratic problem
- Independence along the degrees of freedom ⇒ model split in four sub-models
- Parallel resolution of four optimization problems
- Smaller size for each problem ⇒ faster computation

Online solver: qpOASES based on active-set with hot-start strategy
- The considered problem is well-suited for such approach (high frequencies ⇒ limited changed between two subsequent problems)
- A smart choice of weights and constraints (⇒ tuning) assures stability and fast computation

open-source, C++ implementation
Real-time application: implementation

- The application has strict **real-time** requirements
  - control frequency: 100 Hz
  - sport cars driven by professional drivers ⇒ fast dynamics involved
  - real-time tuning

- **Quadratic** cost function: quadratic problem

\[
J = \sum_{j=0}^{N_P} \delta(j) [\hat{y}(k+j|k) - r(k+j)]^2 + \sum_{j=1}^{N_C} \lambda(j) [u(k+j-1)]^2 + \gamma(j) [\Delta u(k+j-1)]^2
\]
Real-time application: implementation

- The application has strict real-time requirements
  - control frequency: 100 Hz
  - sport cars driven by professional drivers $\Rightarrow$ fast dynamics involved
  - real-time tuning

- Quadratic cost function: quadratic problem
- Independence along the degrees of freedom $\Rightarrow$ model split in four sub-models
  - parallel resolution of four optimization problems
  - smaller size for each problem $\Rightarrow$ faster computation
The application has strict real-time requirements
- control frequency: 100 Hz
- sport cars driven by professional drivers ⇒ fast dynamics involved
- real-time tuning

**Quadratic** cost function: quadratic problem

Independence along the degrees of freedom ⇒ model split in *four sub-models*
- parallel resolution of four optimization problems
- smaller size for each problem ⇒ faster computation

**Online solver:** *qpOASES*
- based on *active-set* with *hot-start* strategy
- the considered problem is well-suited for such approach (high frequencies ⇒ limited changed between two subsequent problems)
- a smart choice of weights and constraints (⇒ tuning) assures stability and fast computation
- open-source, C++ implementation
Experimental results are performed using a hot-hatch car (Volkswagen Golf R) on Calabogie track (Canada).

The algorithm is applied to the platform using a GUI developed with MatLab and the implemented and compiled in C++ programming language to improve real-time performances.

The platform is driven by a professional driver.

Since the platform DOFs are almost decoupled, only longitudinal acceleration and pitch velocity are shown.
A first FPGA implementation is under development with the collaboration of Imperial College London (Control and Power Group).

FPGA: Field Programmable Gate Array
- compromise between general purpose hardware and fully customized hardware
- consists of logic blocks that can be programmatically linked to obtain the desired functions
- on-board memory elements

FPGAs are well suited for MPC algorithms
- large amount of computation for a small amount of I/O
- provide the precise timing guarantees required for interfacing the controller to the physical system

Optimization method: interior point
- polynomial complexity
- takes advantage of the sparsity structure of the matrices
- pipelining: exploiting parallelism to keep the linear solver always active
Conclusions and future works

Conclusions

- The MPC algorithm simplify the working area limits management and the tuning procedure, avoiding undesired behaviours.
- The vestibular system model allows to obtain enhanced trajectory references with better exploitation of tilt coordination.
- The algorithm is already applied to the real platform, with real-time implementation.

Future works

- Introduce real-time prediction, taking advantage of repetitive laps (racing application).
- FPGA integration with the real platform, to test the performance improvement of an hardware-based implementation.
- Application of the algorithm to different devices.
Papers


- F. Maran, A. Beghi, M. Bruschetta: *An MPC approach to the design of motion cueing algorithms for driving simulators*. Convegno Annuale dei Docenti e Ricercatori Italiani in Automatica, interactive session, Pisa, September 2011.


- F. Maran, A. Beghi, M. Bruschetta. *A real time implementation of MPC based Motion Cueing strategy for driving simulators*. IEEE Conference on Decision and Control (CDC 2012), Maui, December 2012.
Part II

Control Techniques for an Hybrid Sport Motorcycle
Electrification of a 125cc motorcycle: Aprilia RS125

Improvement in torque performance ⇒

![Graph showing torque performance comparison between endothermic and hybrid systems]
Electrification of a 125cc motorcycle: Aprilia RS125

Main project goals

- Control strategies development
- Virtual environment for test and performance evaluation

Virtual Environment

- Motorbike and rider model: VI-Grade tools
- Battery model
- Hybrid engine model
Virtual Environment

**Simulink implementation**
Battery model: discharge

- Lithium-Polymer cell (by Kokam)
- No test-bench available for batteries
- Available informations: datasheet charge/discharge curves
- Polynomial fitting of the reference curves

What are the values of interest?

- Voltage $V$ as a function of State of Charge ($SOC$)
- Discharge characteristic $V(SOC)$ immediately available
Battery model: charge

Charge curve

- Two distinct charge stages: constant current $I$ ($SOC < 90\%$), constant $V$ ($SOC > 90\%$)
- Given $V(t)$ and $I(t)$ $\Rightarrow$ $\begin{align*}
P(t) \text{ (power)} \\ E(t) \text{ (energy)}
\end{align*}$ $\{ V(E) \Leftrightarrow V(SOC) \}$
Hybrid engine model: endothermic and electric components

Endothermic engine
- Torque map-based

Electric motor
- Charging/boosting operation modes
- Charging/boosting power estimation
- Different settable boost maps

Hypotheses
- Negligible dynamics of the electrical machine
- Current transient respected (ideal charge control)
- Model based estimation of the charging power status
Hybrid engine model: activation strategy

Activation signal: evaluation of the speed derivative

- Charge activation during deceleration or constant speed
- Boost activation during acceleration

Problem: high frequency variations in activation signal

Solution: temporized disabling of boost/charge when up-shift occurs
Hybrid engine model: activation strategy

Activation signal: evaluation of the speed derivative
- Charge activation during deceleration or constant speed
- Boost activation during acceleration

Gear shift control
- Speed decreasing during up-shift
- Problem: high frequency variations in activation signal
- Solution: temporized disabling of boost/charge when up-shift occurs
Results: simulation

- Virtual track with features similar to a urban path
Results: simulation

- Virtual track with features similar to a urban path
- Virtual driver performance regulated by using gear shift maps
- Problem: the virtual environment requires a fixed speed profile for the virtual driver. How to evaluate boost performance?
Results: simulation

- Virtual track with features similar to a urban path
- Virtual driver performance regulated by using gear shift maps
- Problem: the virtual environment requires a fixed speed profile for the virtual driver. How to evaluate boost performance?
- Solution: we compare the different throttle requests from the virtual driver for the endothermic and hybrid vehicles with the same speed reference
Results: control strategies comparison

- Stability of the motorcycle confirmed (smooth torque variation)
- Improvement in performance: throttle demand reduction
- Torque split allows a better battery exploitation
Results: control strategies comparison

- Stability of the motorcycle confirmed (smooth torque variation)
- Improvement in performance: throttle demand reduction
- Torque split allows a better battery exploitation
Conclusions and future works

Conclusions

- Polynomial battery model
- Driver-oriented analysis via a flexible virtual prototyping tool
- Effective battery management and performance improvement
- Optimal control strategy: regulation of contribution from the two engines
- Prototype implementation: on-track test

Future works

- Development of different boost-maps
- Auto-switching of boost-maps based on the driving style
- Refinement of the torque split strategy
- Model-based predictive control on the electric machine
Papers


Thank you for the attention!

Fabio Maran

PhD School on Information Engineering
XXV Series, ICT Section
Advisor: Prof. Alessandro Beghi

fabio.maran@dei.unipd.it