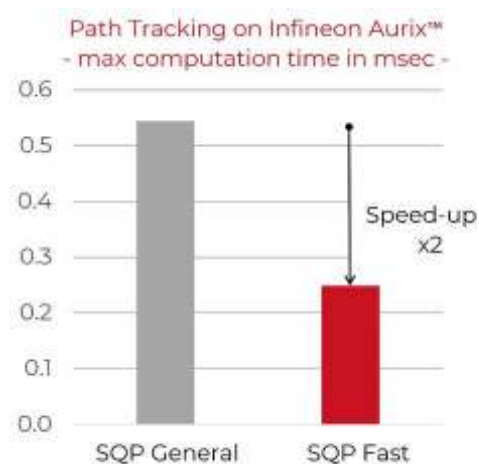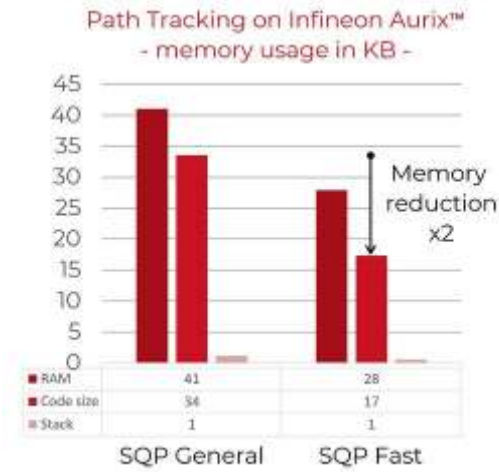# FORCESPRO 6.0 Release

---

FORCESPRO

New QP Fast and SQP Fast algorithms for automotive-grade embedded hardware

- **QP Fast** and **SQP Fast**: QP Fast and SQP Fast are new algorithmic options of FORCESPRO for deploying Linear and Nonlinear MPC on automotive-grade embedded hardware. The algorithms are tailor-made to be run in single-precision floating point accuracy. Best-in-class memory efficiency and record-low computation times enable MPC deployment on safety-rated hardware platforms such as Infineon Aurix™. Both algorithms are suited for reference tracking, closed-loop control tasks. One notable application example is vehicle path tracking.

- **Performance increase for Nonlinear MPC path tracking (DOCU / VIDEO TUTORIAL)**: For an exemplary vehicle path tracking application, the mean and max computation times are reduced by 52% and 54%, respectively, and the memory consumption by 50% with the new SQP Fast algorithm.
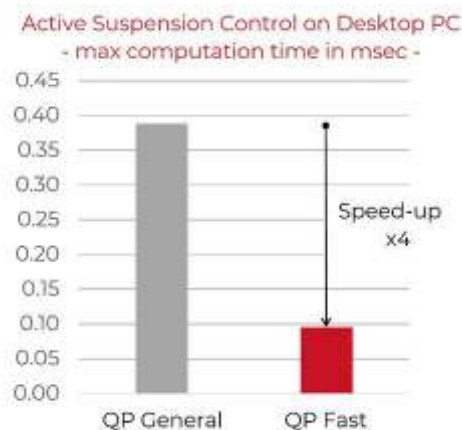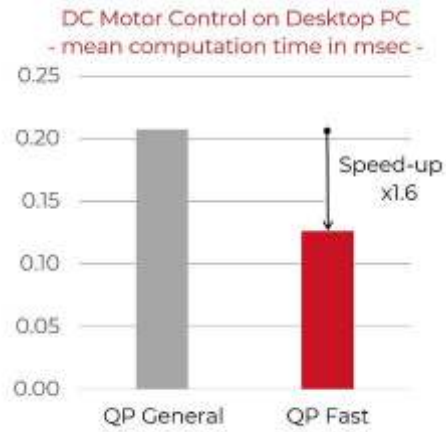
Path Tracking on Infineon Aurix™
- memory usage in KB -

	SQP General	SQP Fast
RAM	41	28
Code size	34	17
Stack	1	1

- **Performance increase for Linear MPC (DOCU / VIDEO TUTORIAL)**:

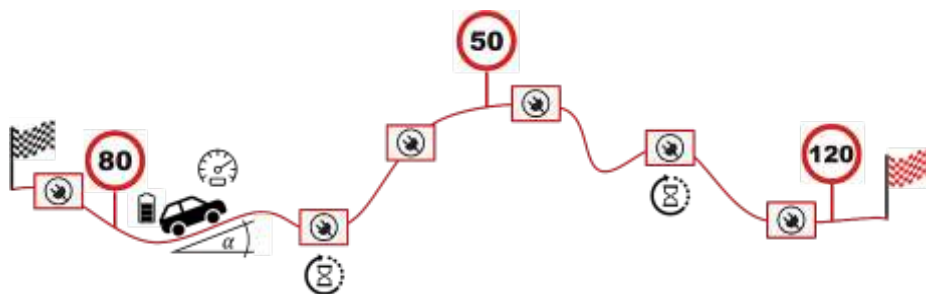 In case of an exemplary active suspension control application, the mean and max computation times are reduced by 92% and 75%, respectively, compared to our QP General algorithm.

  The new QP Fast algorithm can also be used within the MathWorks Model Predictive Control Toolbox™. The workflow shown for an exemplary DC motor control application, leads to a reduction in mean and max computation times by 40% and 20%, respectively. The closed-loop control performance remains unchanged in both examples.

Active Suspension Control on Desktop PC
- max computation time in msec -

DC Motor Control on Desktop PC
- mean computation time in msec -

- **Updated memory allocation mechanism**: Starting with version 6.0, the user has full control and transparency over memory allocation for all FORCESPRO (S)QP and NLP solvers on embedded devices. This release also extends thread safety (initially introduced in version 4.3) to all solve methods and the associated solver options. Furthermore, it allows the combination of external parallelism (running multiple solvers concurrently) and internal parallelism (running a solver on multiple cores).

- **New EV mission optimization example**: A new example demonstrates how to use the FORCESPRO NLP solver to determine the optimal charging and speed profile of an electric vehicle in order to minimize its total trip time. The use of traction and braking force, selection of charging stops, and respective charging time are optimized such that the given route is traversed in the shortest time possible, while simultaneously respecting speed limits as well as the vehicle's and battery's technical requirements. The computational efficiency of the FORCESPRO solver is showcased by solving the optimal charging and driving profile of a BMW i3 travelling from Munich to Cologne, where a 573 km long trip is calculated in 50 milliseconds.

You can find a list of all algorithmic improvements in the **Release Notes of FORCESPRO.**
Existing users can easily switch to the new version by using our **auto-update function.**
Alternatively, you can use the new server at: **https://forces-6-0-0.embotech.com**

You can find a list of settings for maintaining code options of old FORCESPRO versions, as well as the instructions on how to migrate to the new memory C interface, in the **legacy list**.

Please note:

- In order to accommodate major C API changes coming with the FORCESPRO 6.0 release, the users of FORCESPRO via C will have to adapt their C interface accordingly. The necessary changes for migration can be found in the **legacy list**. The new C interface is described in the **documentation (step 4)**.
- All releases prior to 4.4.0 will go offline as of September 2, 2022. Please contact support@embotech.com in case you want to continue working with one of these versions.

Connect with us

🌐 in 🐦 ▶ ✉

embotech ⁎