OpenSuperQ

An open QC platform as a catalyst for applications

Frank Wilhelm-Mauch, Shai Machnes, Nicolas Wittler, Kevin Pack, Federico Roy, Thorge Müller[1], David Headley[2]

PGI 12, Research Center Jülich

[1]also DlR [2]also Mercedes Benz Research

www.opensuperq.eu

The OpenSuperQ project has received funding from the European Union’s Horizon 2020 research and innovation programme under grant

OpenSuperQ

Contents

- OpenSuperQ: Presentation and approach
- Making progress in the NISQ era:
  - Rich gate sets
  - Co-design of hard+software
  - Better gates and a bigger challenge

OpenSuperQ

On OpenSuperQ

OpenSuperQ

Horizon 2020 Project OpenSuperQ

Overall vision: to build a hybrid high-performance quantum computer of up to 100 qubits and to sustainably make it available at a central site for external users.

10 Partners
5 Countries

Start Date
1 October 2018

Duration
36 Months

Budget
€ 10.33 Mio

OpenSuperQ

Project Partners

UNIVERSITÄT DES SAARLANDES

ETH zürich

JÜLICH Forschungszentrum

Zurich Instruments

LOW NOISE FACTORY www.lownoisefactory.com

CHALMERS UNIVERSITY OF TECHNOLOGY

eman ta zabal zazu Universidad del País Vasco Euskal Herriko Unibertsitatea UNIVERSITY OF THE BASQUE COUNTRY

VTT

BLUEFORS

EURICE EUROPEAN RESEARCH AND PROJECT OFFICE GMBH

4 Universities

2 R&T Organisations

4 Industry Partners

Partner Responsibilities

- Saarland University: coordination and management, benchmarking, firmware applications and theory
- ETH Zürich: chip fabrication, measurement, cryogenics and wiring
- Chalmers University: chip fabrication, control and modelling, applications
- University of the Basque Country: modelling, quantum algorithms and useful applications
- Forschungszentrum Jülich: modelling, high-level software and simulation, hosting
- VTT: readout and amplification, packaging, 3D integration
- Bluefors: cryogenics, cryo-wiring
- Zurich Instruments: hardware and software for readout and control
- Low Noise Factory: microwave technology
- Eurice: project management, exploitation and communication

OpenSuperQ

Full Hardware and Software Stack

Integrated unit?
Second part of
the talk!

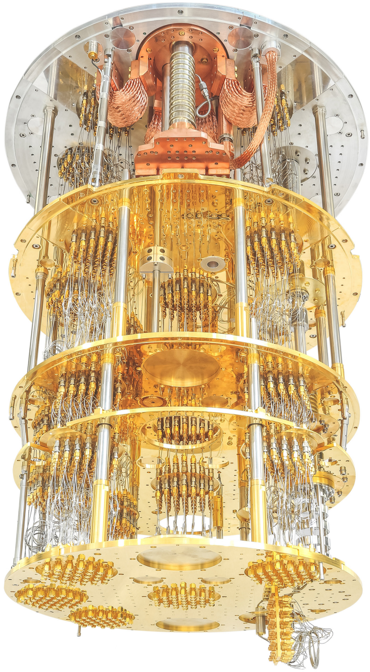
Programming Interface
Compiler
Gates
Real-time Control
Qubit

Room Temperature Control
Wiring and Cryostat
Control Chip
Quantum Chip Architecture
Qubit

OpenSuperQ

Classical Infrastructu

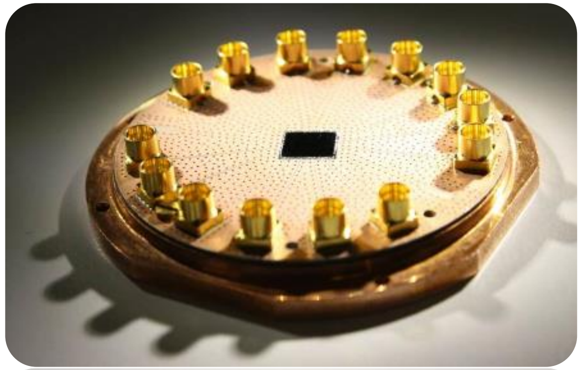
- Helium dilution cryostat
- Copper plates to maintain temperature at different stages
- Quantum computing control system
    - AWGs
    - Quantum analysers
    - Programmable system controller
- LabOne® instrument control software to connect to higher level in the quantum stack

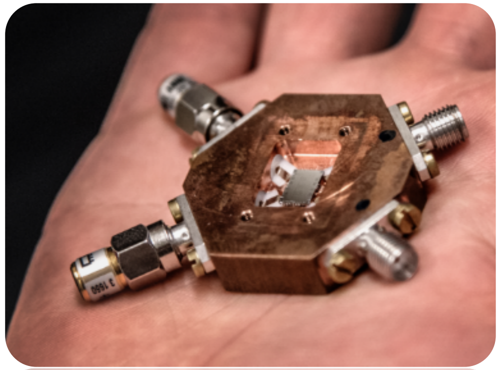


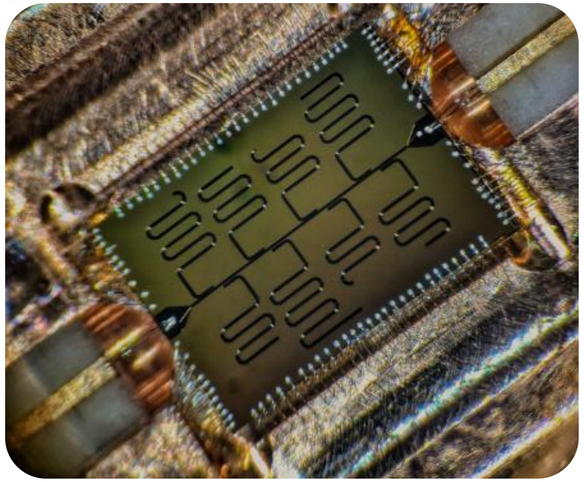
OpenSuperQ

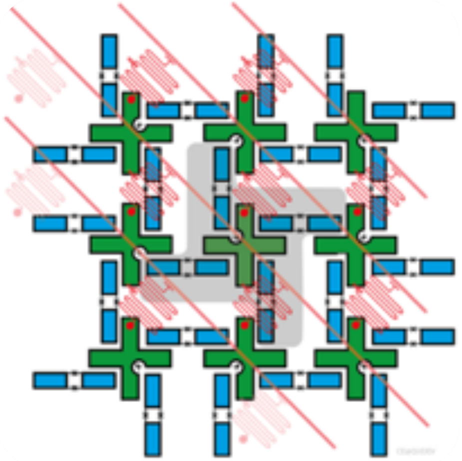
Package and Chip

- Package for microwave I/O
- Flip-chip 3D integration
- Array of coupled qubits





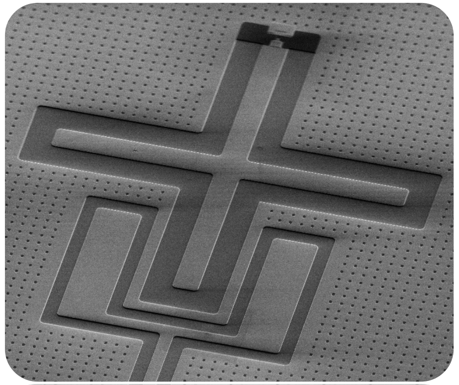


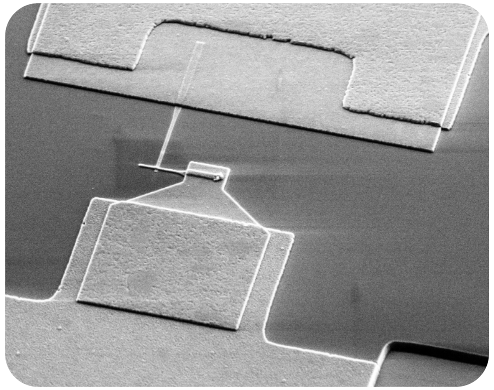


OpenSuperQ

Josephson Qubits

- Superconducting transmon qubits based on Josephson junctions

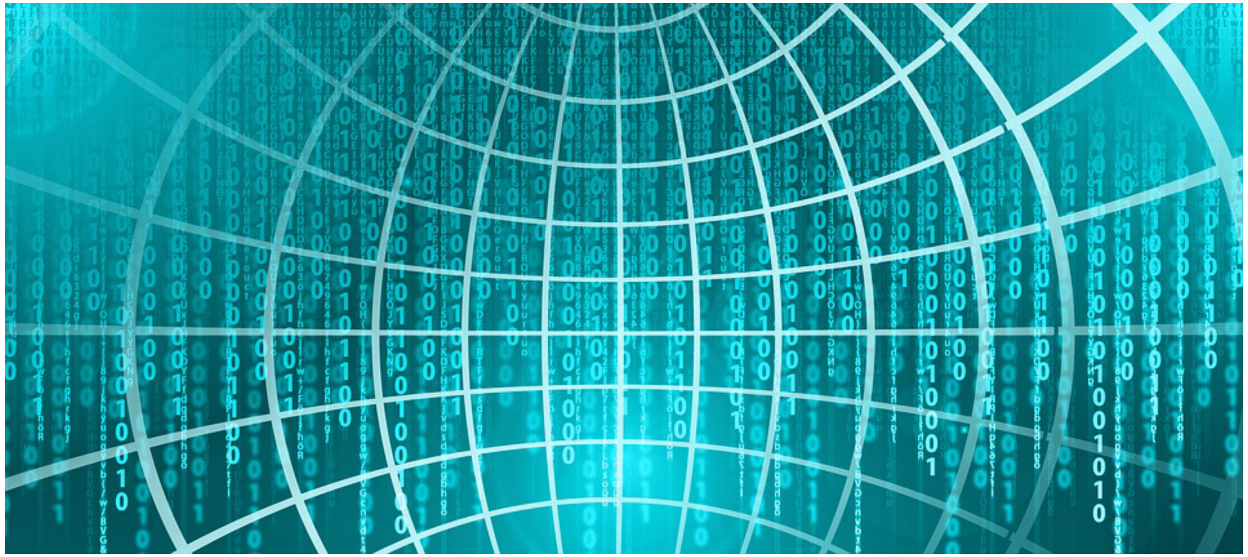




OpenSuperQ

Software

- Run in a central facility
- Programming interface for users
- Operable in a high-performance computing environment
- Tight integration with classical computers



OpenSuperQ

Programming and co-designing in the NISQ era

What can you achieve with deep access?

NISQ

Clive Sinclair

Simple, primitive, error-prone hardware: Coding needs to follow architecture, do not abstract too much

OpenSuperQ

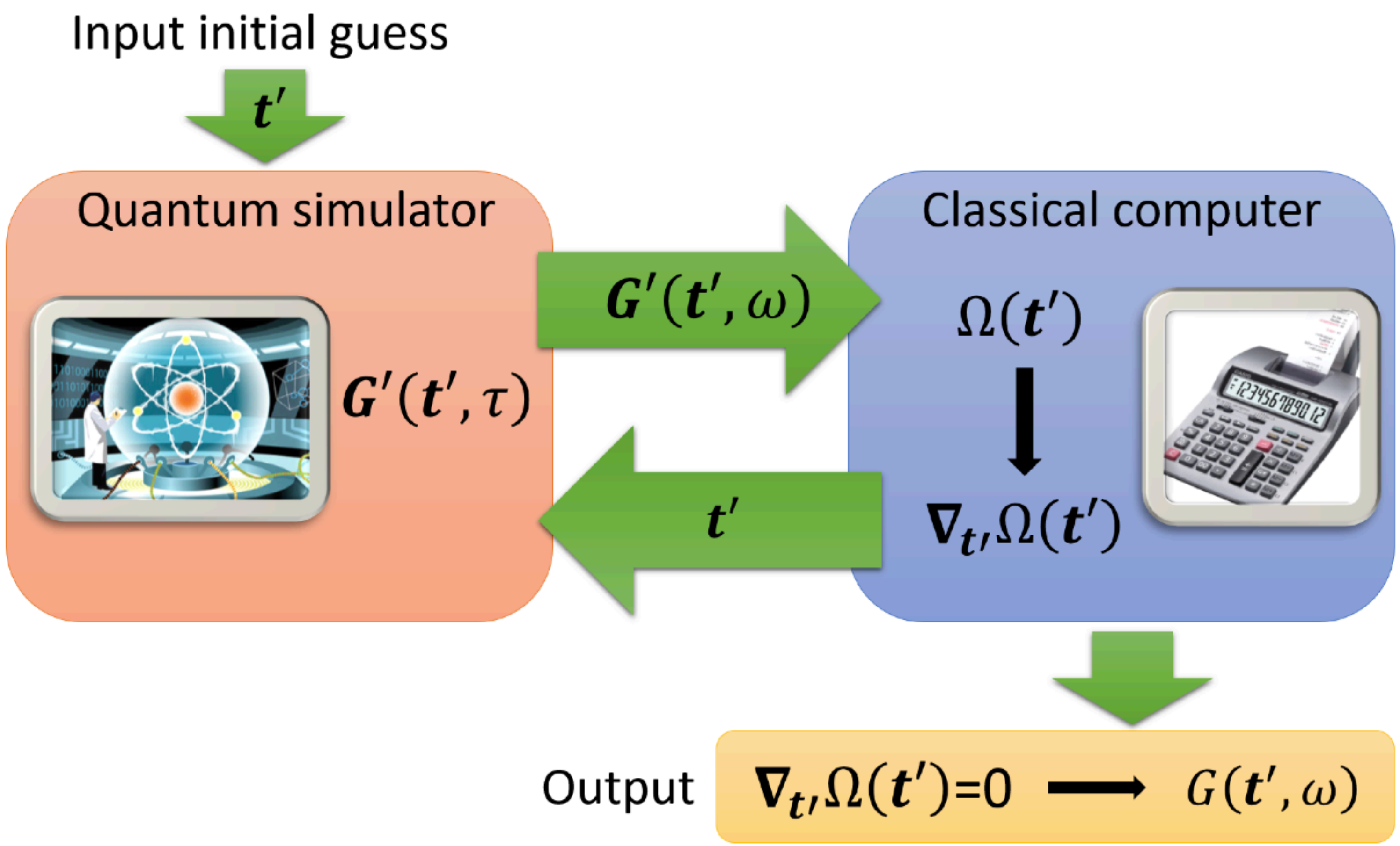
Reducing the size of the quantum operation

- let the (cheap) classical computer do what it is best at

- enhance its performance with the (expensive) quantum computer

OpenSuperQ

Modern variational algorithms



Dallaire-Demers and FKW, 2016

OpenSuperQ

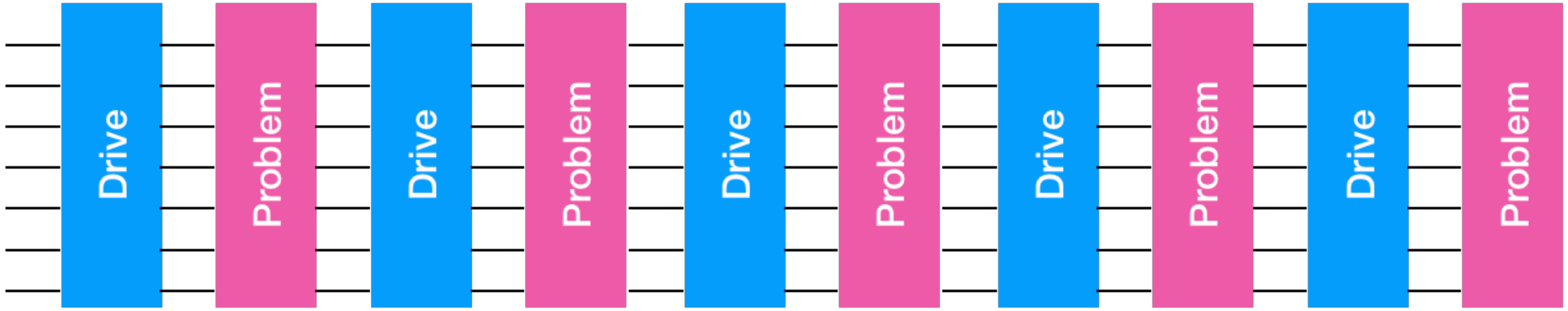
QAOA / Digitized AQC for combinatorical optimization

Problem Hamiltonian: Ising-type $H_p = \sum_i h_i Z_i + \sum_{i<j} J_{ij} Z_i Z_j + \dots$ $\exp\left(-i\beta_i H_p\right)$

Driver Hamiltonian: Tunneling $H_d = -\frac{\Delta}{2}\sum_i X_i$ $\exp\left(-i\gamma_i H_d\right)$

$$\hat{H} = \hat{H}_0 + \sum_i F_i(t)\hat{H}_i$$

$$\hat{H} = \sum_i \hat{H}_i(t) + \sum_{i<j} \hat{H}_{ij}(t)$$

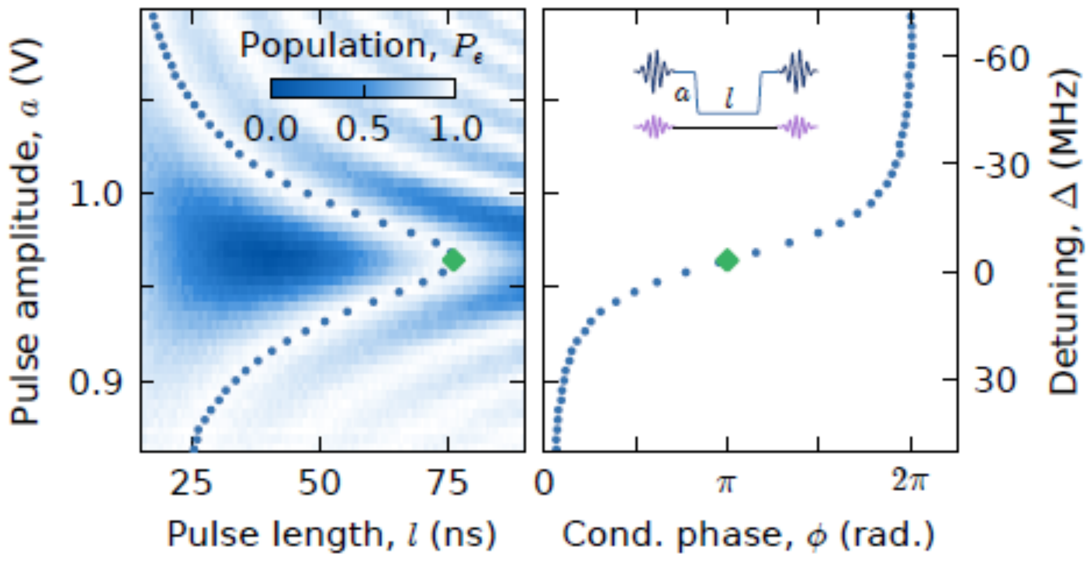


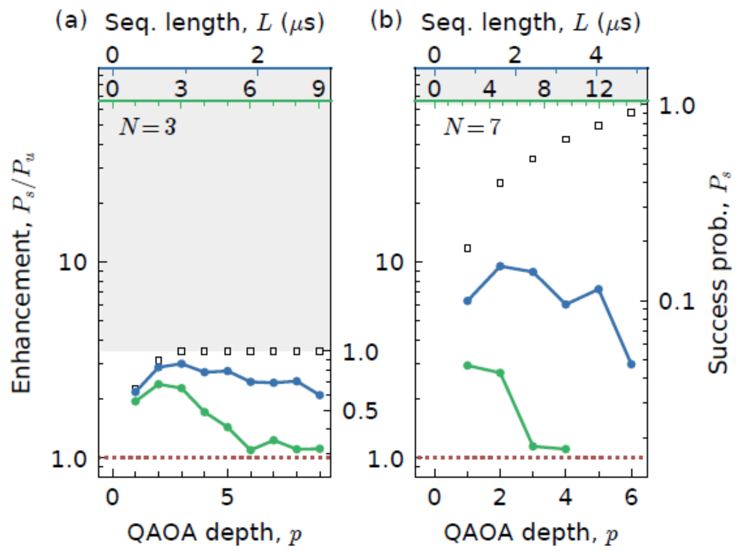
β_1 γ_1 β_2 γ_2 ... β_{n-1} γ_{n-1} β_n γ_n

Measure: $\left\langle \beta,\gamma \middle| H_p \middle| \beta,\gamma \right\rangle$ **Classically minimize** $\beta = \{\beta_i\}$ $\gamma = \{\gamma_i\}$

OpenSuperQ

Direct implementation with reduced compilation

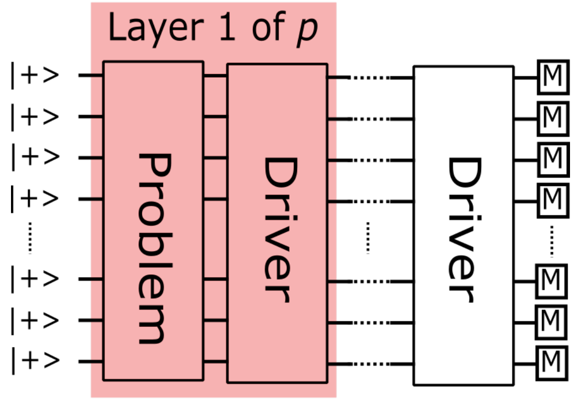




Lacroix et al., arXiv:2005.05275 (ETHZ)

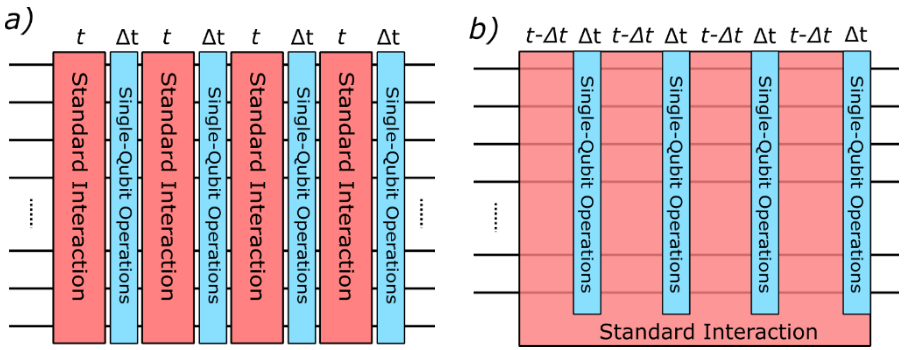
Digital Analogue QAOA

$$\left|\vec{\beta},\vec{\gamma}\right\rangle = \prod_{p'=0}^{p} e^{i\beta_{p'}H_{\mathrm{D}}} e^{i\gamma_{p'}H_{\mathrm{P}}} \left|+\right\rangle^{\otimes n}$$



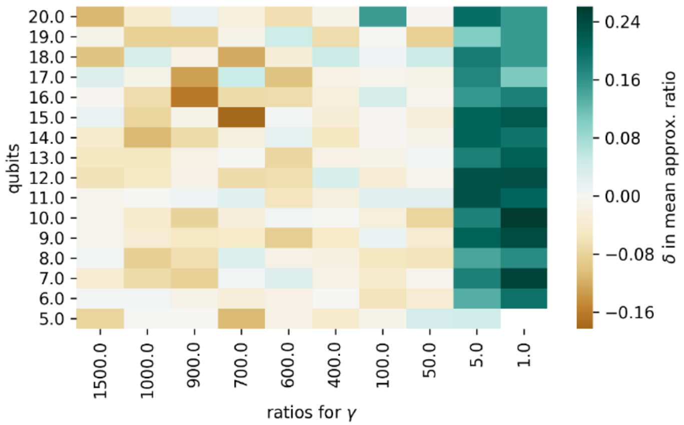
Digital Analogue Scheme

- **Start with fully connected graph**
- Use single-qubit operations to ‘steer’ this resource Hamiltonian
- If resource always on simultaneity of resource and single qubit ops causes error



DA-QAOA

- QAOA problem Hamiltonians suit DA scheme, easy to express
- QAOA can use variational freedom to ‘eat’ coherent DA error
- Faster single-qubit operations improve performance



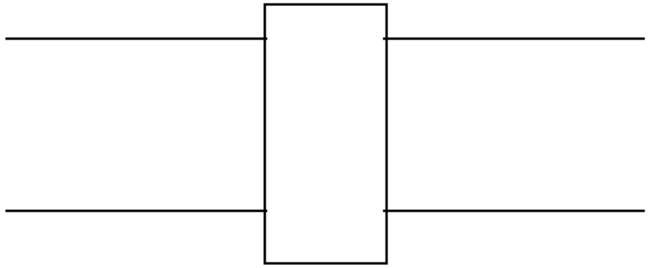
DAIMLER

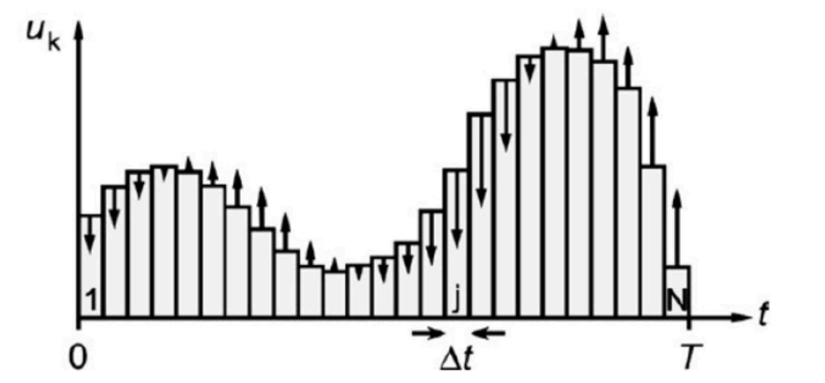
Hardware-co-design in progress

D. Headley, T. Müller, A. Martin, E. Solano, M. Sanz, FKW, 2020

DLR

Analog gate design





$$\hat{U}_{\text{gate}}(t_{i+1}, t_i): \quad i\partial_t \hat{U}(t, t_i) = \hat{H}(t)\hat{U}(t, t_i)$$

$$\hat{H} = \hat{H}_0 + \sum_i u_i(t)\hat{H}_i$$

Find controls implementing U
fast and reliably:
Analog control problem

Practical wishlist:

Fast (limited coherence!)
Simple (easy to calibrate)
Robust (tolerate fluctuations)

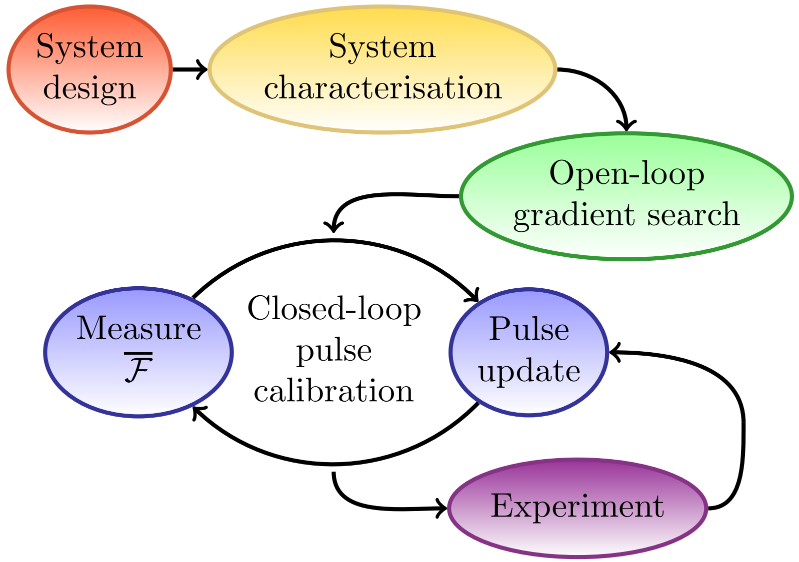
Find controls that maximize fidelity

S.J. Glaser et al., EPJ D 2015

OpenSuperQ

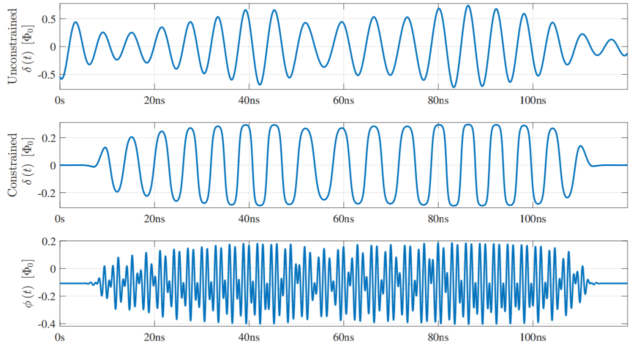
Ingredients

Closing the loop



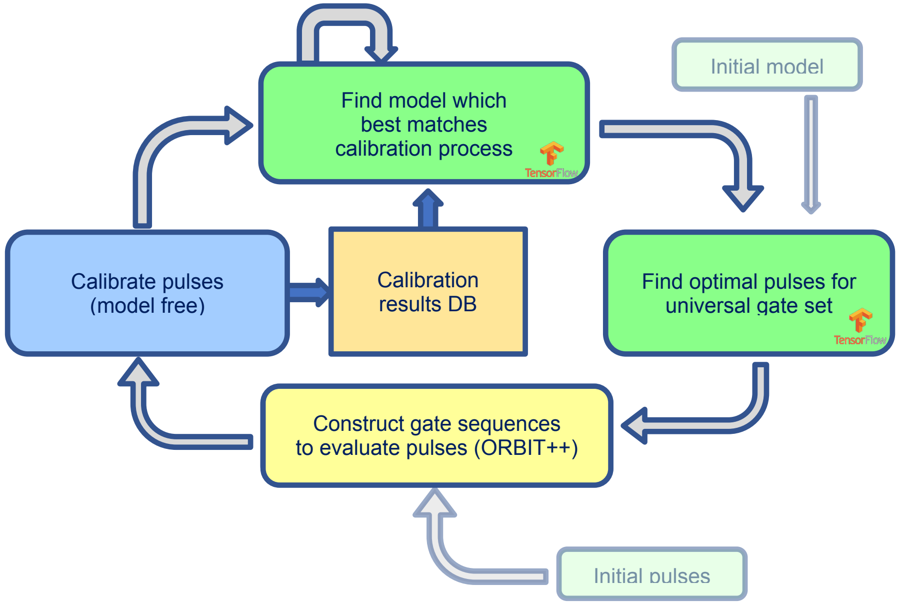
D.J. Egger and FKW, PRL 2014

Gradient search on simple Ansatz



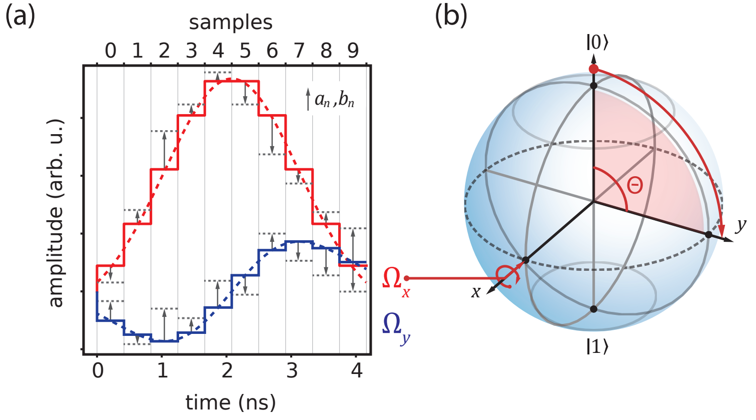
S. Machnes, E. Assemat, D. Tannor, FKW, 2018
S.Kirchhoff, T. Keßler, P.J. Liebermann, E. Assémat, S. Machnes, F. Motzoi, FKW, 2018

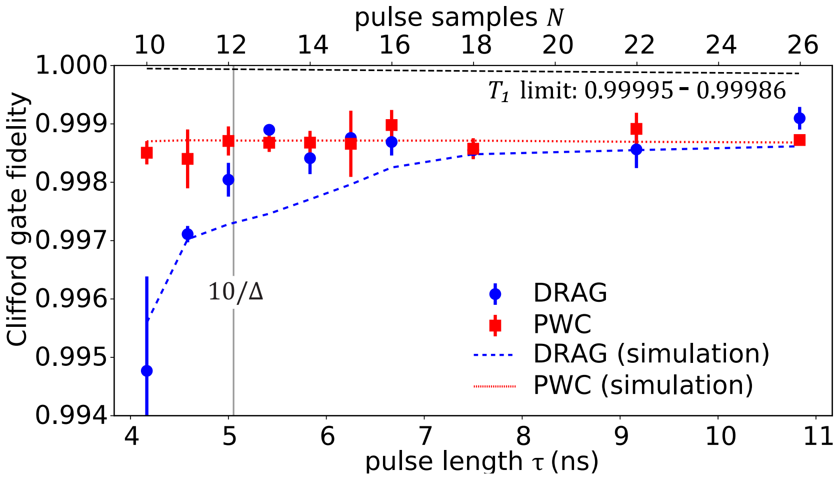
Model identification with AI
(C3 - Combined Control and Characterization)

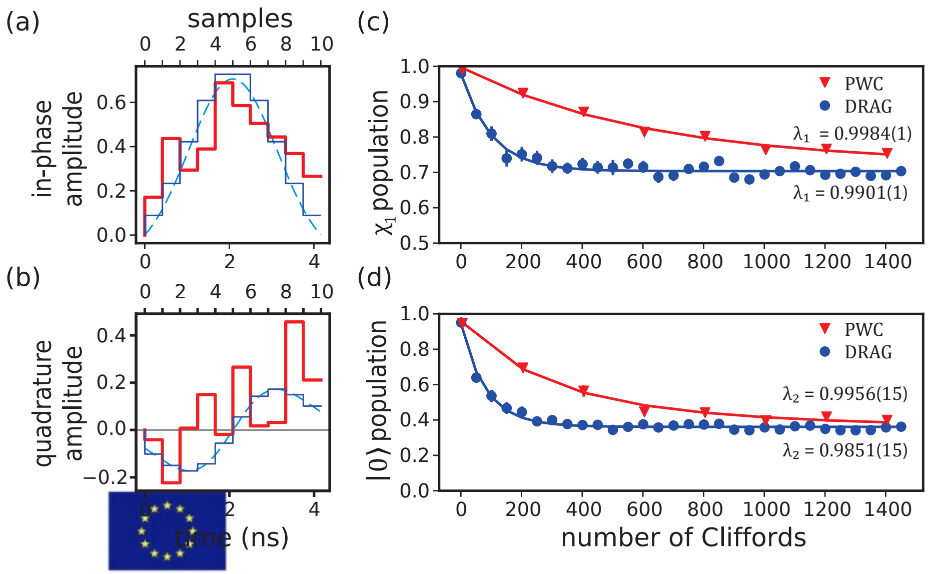


S. Machnes, N. Wittler, F. Roy, K. Pack, A.S. Roy, M. Werninghaus, D.J. Egger, S. Filipp, FKW
arXiv 2020

The breakthrough



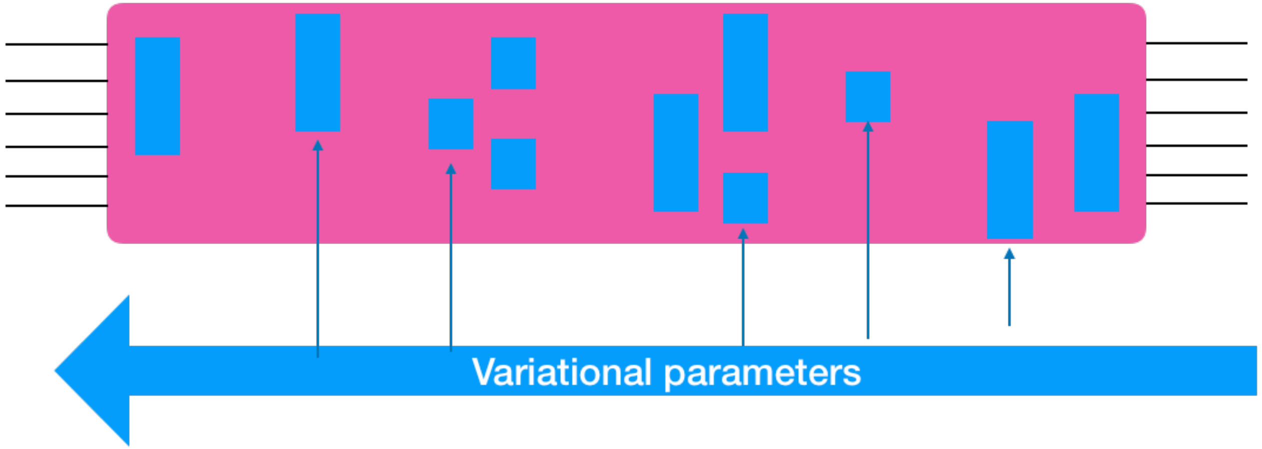




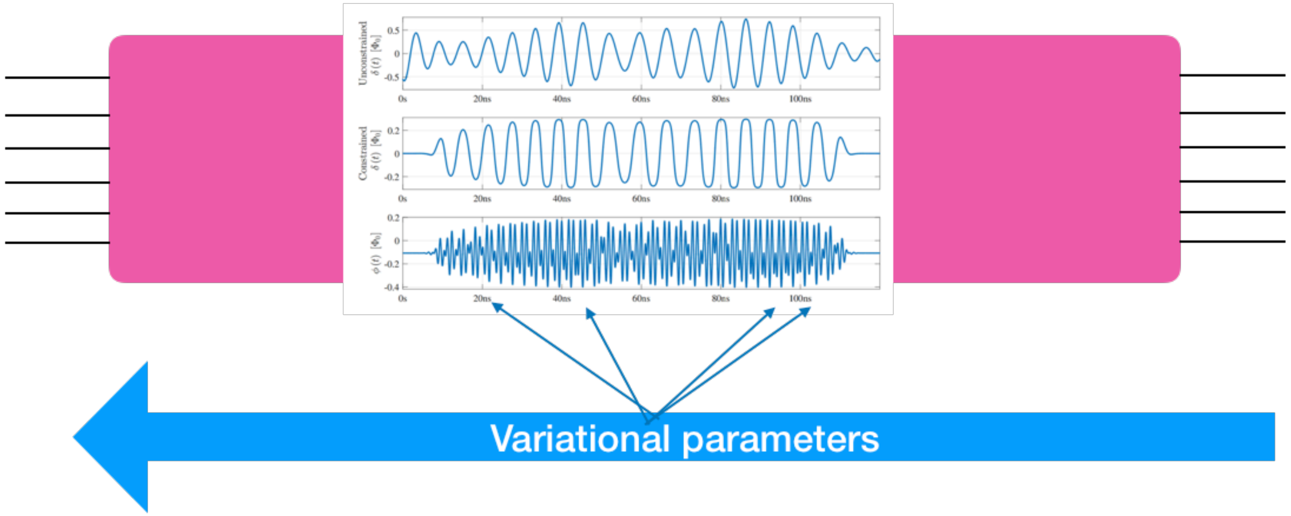
- Explore the quantum speed limit
- Speed up gates
- Discover an unexpected technical limitation

M. Werninghaus, D.J. Egger, F. Roy, S. Machnes, FKW, and S. Fillip 2020

Many ways to write an algorithm



Gate-based algorithm
Universal gate set
Tuneup of gates



Optimal control
Controllability
Analogue programming

OpenSuperQ

Statements for discussion

- Disruptive programming for quantum computers closely integrates software on and for quantum computers
- We have not found the best paradigm to program quantum computers yet - adiabatic, gate model and controls are just first guesses
- This cannot be done through a user interface to a walled garden, it needs deep access and collaborative research