

**United
Technologies**

Climate | Controls | Security

An Industrial Model Based Development Systems
Engineering Strategy

American Modelica Conference 2018, October 9-10 2018,
Cambridge, MA

Presenters:

Johan Åkesson, Modelon
Kristian Tuszynski, CCS
Clas Jacobson, CCS

AGENDA

- Team
- Key Points
- Drivers
- Tools chain
- Application examples
- Summary

TEAM

Modelon

Johan Åkesson, Johan Andreasson, Magnus Gäfvert, Magda Axelsson, Katrin Prolss, Iakov Nakhimovski, Fredrik Magnusson, Bryan Eisenhower

Carrier

Trevor Bailey, Degang Fu, Qingfan Zeng, Clas Jacobson, Kristian Tuszynski, Rui Huang, Chen Zhang, Lishan Wang, Dongzhi Guo

External

John Cassidy (UTC, retired), Larry Biegler (CMU), Karl Åström (LTH), Carl Laird (Sandia), Kevin Otto (RSS)

KEY POINTS

Modelica is a modeling language that (1) captures physics and is useful for modeling at the (2) system level and for modeling (3) heterogeneous systems:

- Need for system models of different scope, complexity and domains
- One modelling language

The use of Modelica is on simulation but goes beyond in “systems engineering” the (re)use of models for variability and robustness analysis, optimization and analysis of design freedom, and control design and analysis:

- Started with control design
- Goal: Unification of model development to Modelica

CCS is using Modelica for system level modeling and the Modelon tool chain to capture system level modeling and to deploy widely using library architectures, GUI and Python infrastructure:

- Support for a tools set that allows unification of models
- Steady-state/Dynamic/Cluster execution/Optimization/Variability

DRIVERS

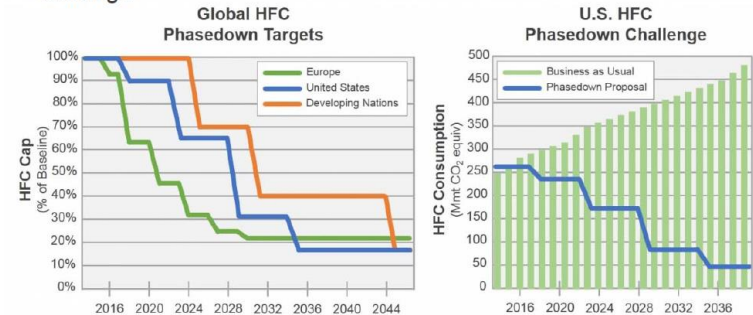
Systems Engineering Needs

Engineering effectiveness –
drive designs by models –
validate requirements and drive
efficient testing

Need to *deal with increased
system integration
complexity* – Components >
Chiller > Chiller plant > Building

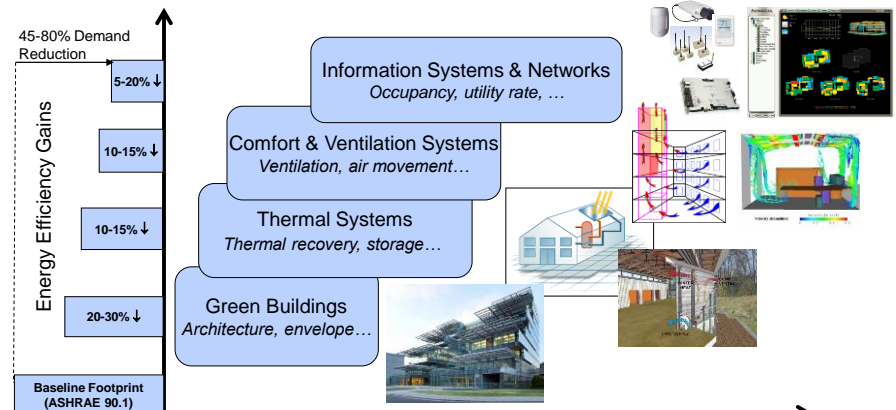
*Regulatory environment
demands design efficiencies*
(new technologies, refrigerant
changes) and energy efficiency
(whole building level)

- Action to phase down HFCs can avoid up to 0.5 °C of warming by 2100
- HVACR uses 50% of all energy in U.S. commercial and residential buildings



Source: U.S. Department of Energy
<http://energy.gov/eere/buildings/road-zero-does-next-generation-heating-and-cooling-rd-strategy>

Energy Savings in Commercial Buildings



- Reduce risk/enhance maturity
- Make solutions scalable & robust
- Drive commercial adoption

NEED TO EXPOSE DIFFERENT VIEWS...

Heterogeneous Modeling, Different Fidelities...

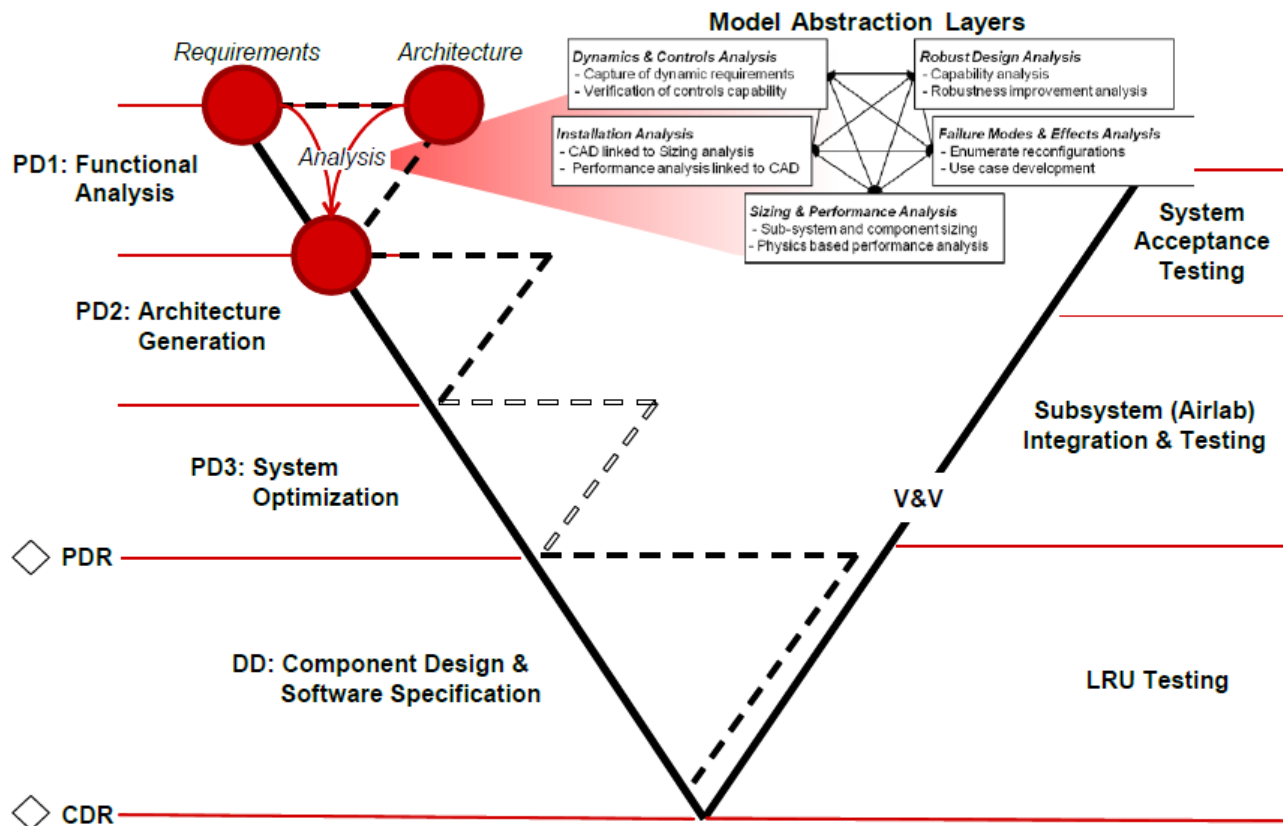
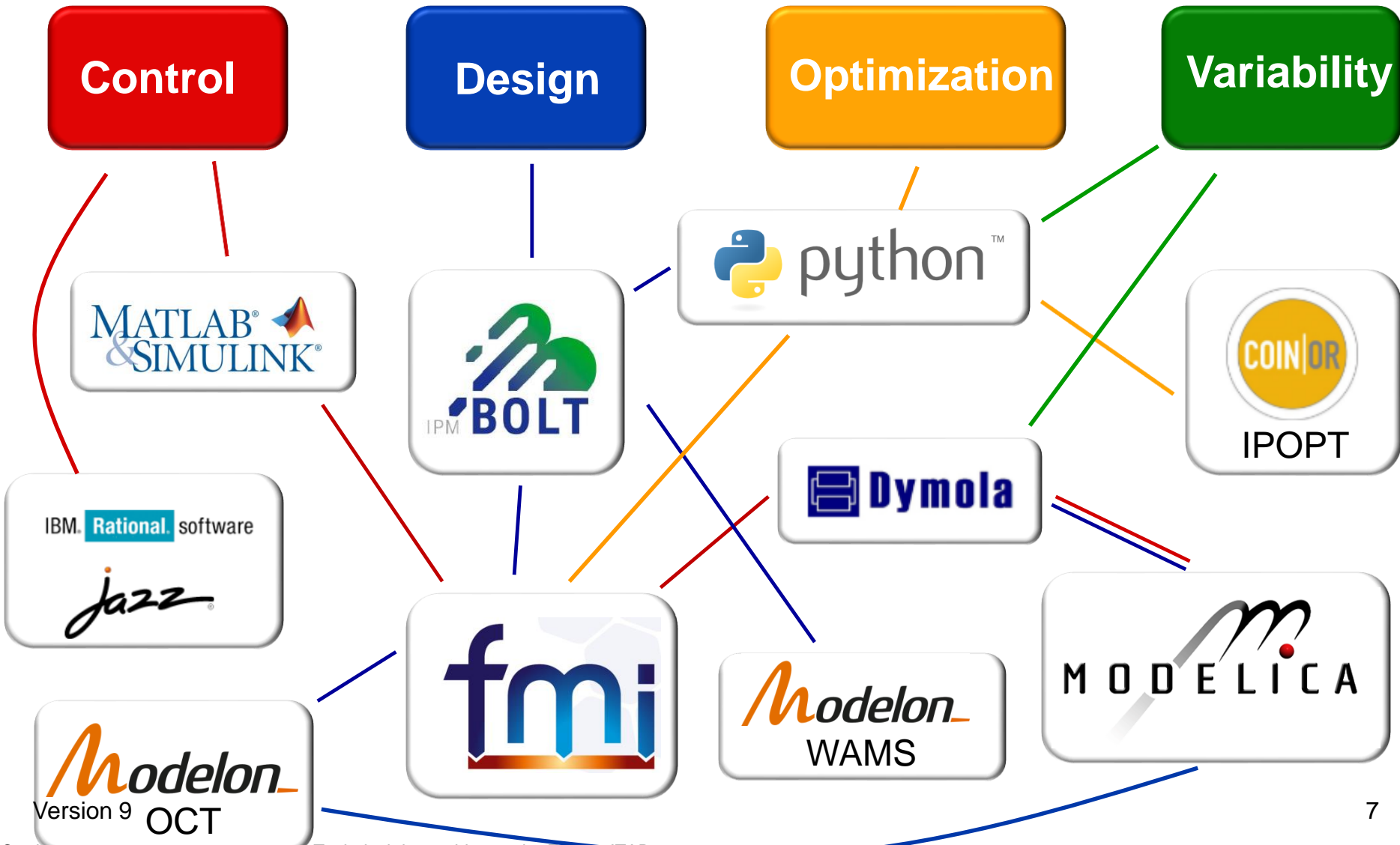


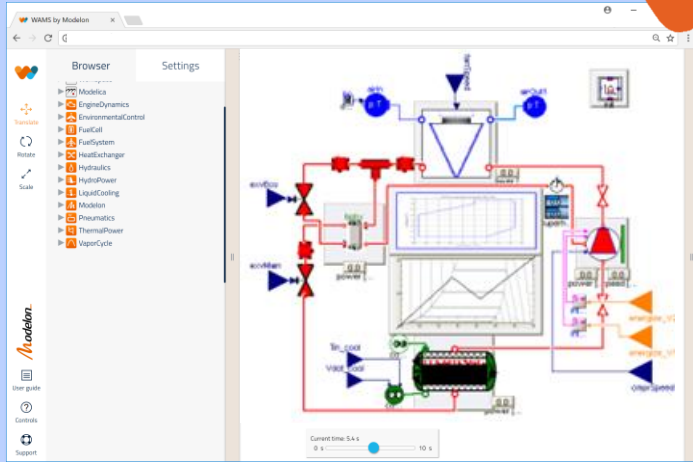
Figure 1: Proposed Model Based Development Process

TOOLCHAIN ECO-SYSTEM

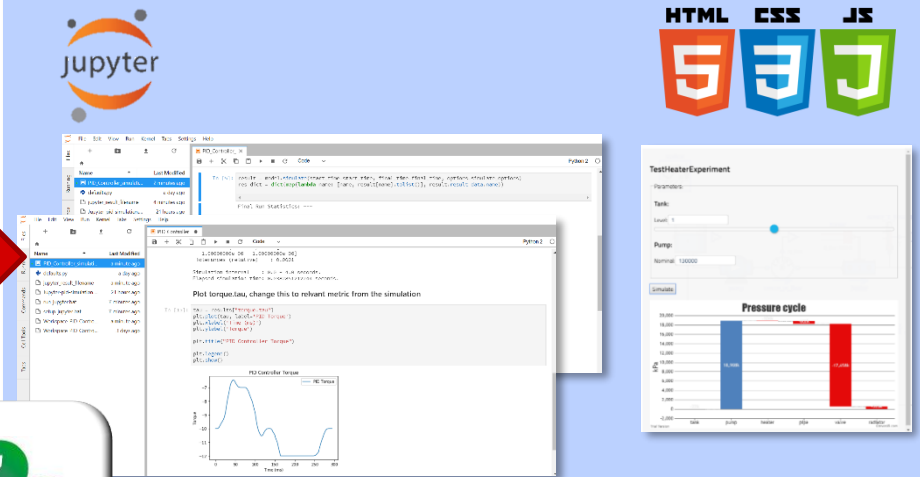


MODELON PLATFORM FOR MBD

WAMS



Customizations



Create

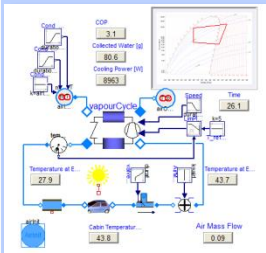
Deploy



OPTIMICA Compiler Toolkit

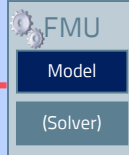
MODELICA

python

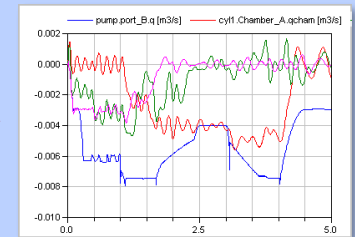


Compiler for Modelica and FMI

fmi
FUNCTIONAL MOCK-UP INTERFACE



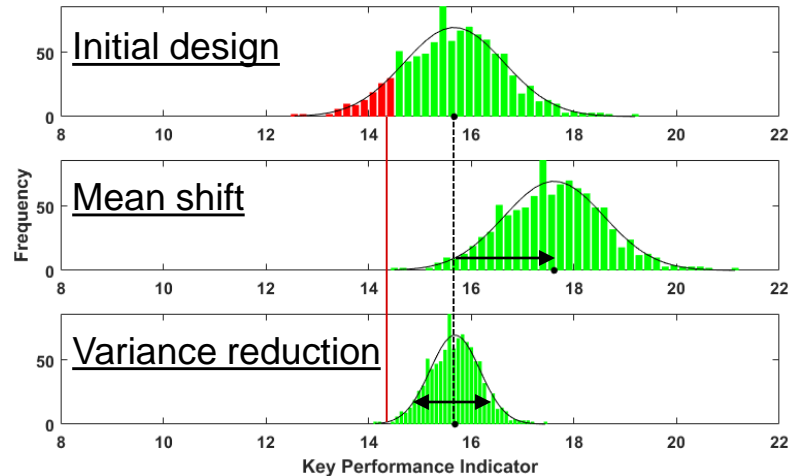
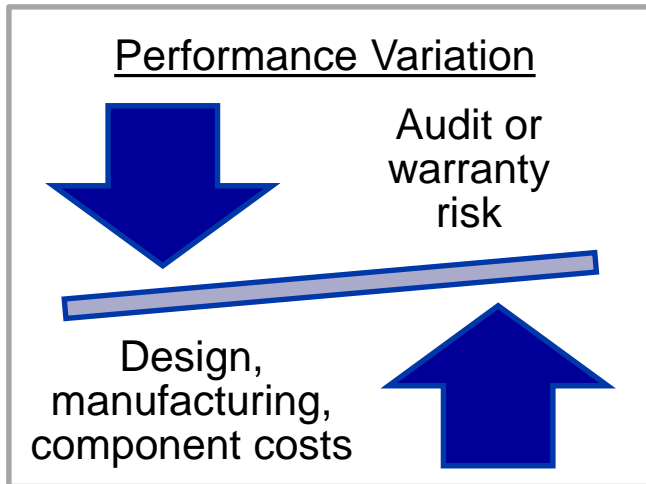
Numerical solvers for simulation and optimization



Compile

Compute

VARIABILITY – ENGINEERING METHODS

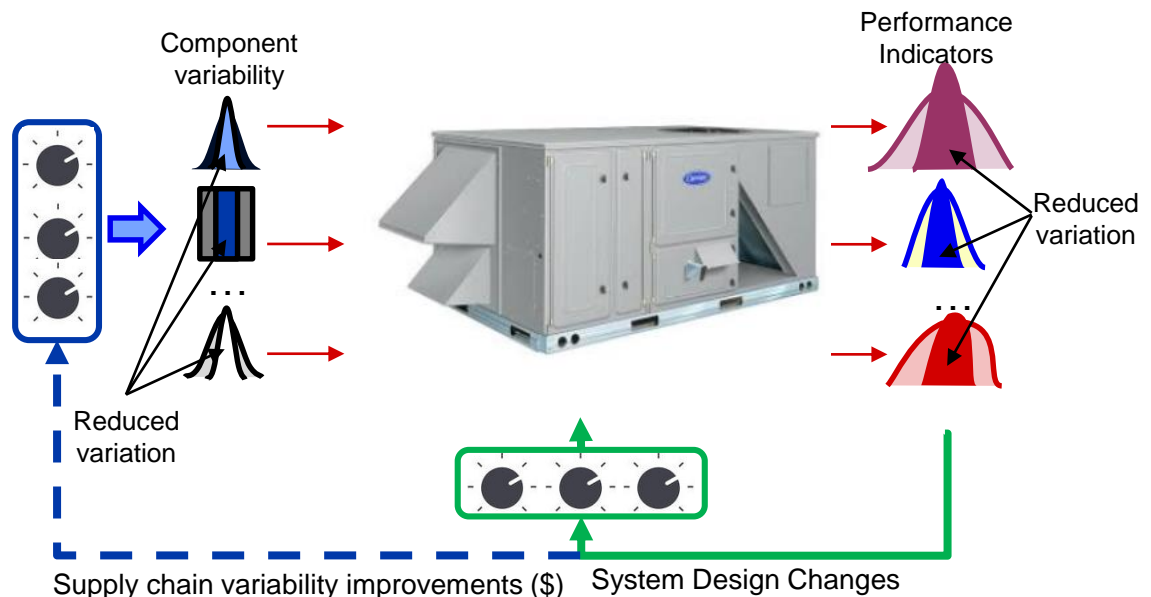


Analysis methods

- Model-based manufacturing analysis
- Test variation: Gage R&R, internal audits, extra testing

Design methods

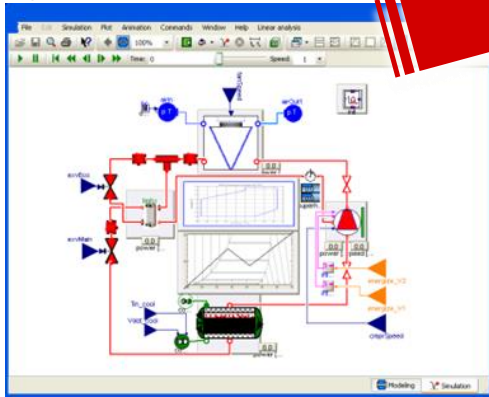
- Adjust the relationship between **mean** (designed), **deviation**, **tails** of KPI's
- Feasibility analysis
- Sensitivity/variability reduction



MODEL BASED CONTROL DESIGN

Software tools to support automatic code generation and testing

Dymola 
Dynamic model



FMI Toolbox 
Run model in Simulink



Matlab/Simulink/Stateflow
Create controller
Automatic code generation
Tool boxes: EmbeddedCoder



Generated code

Legacy code

```

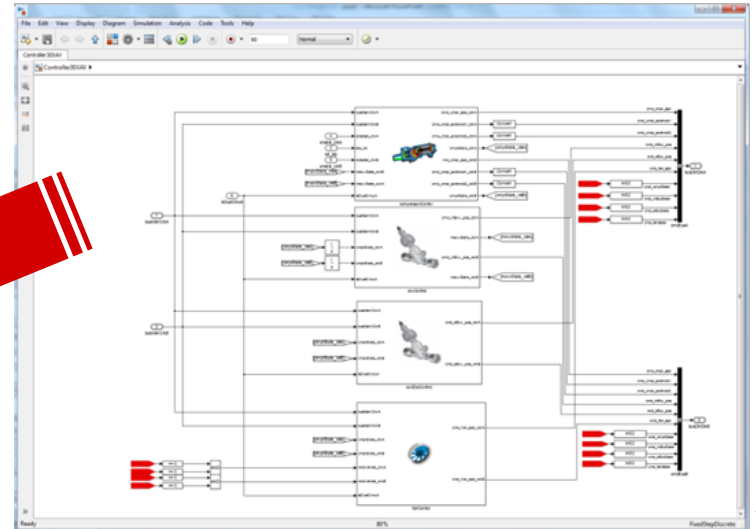
// Include needed header files
#include <OC_const.h> // OC Constants
#include <Common.h> // Basic types
#include <stdio.h> // Printf
#include <Data.h> // Vectors and matrices
#include <OC_nag9.h> // NAG function
    
```

```

// OC Constants
// Basic types
// Printf
// Vectors and matrices
// NAG function

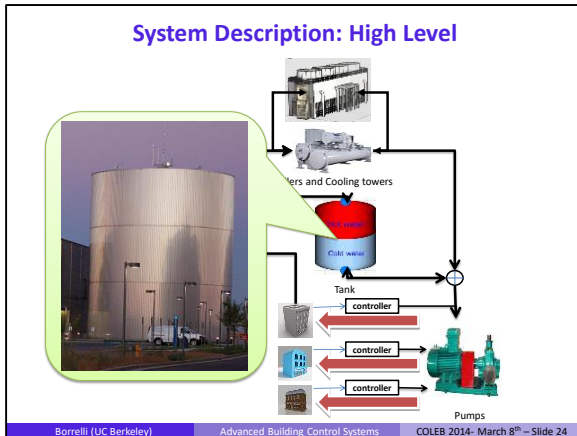
x code
parameters ***
s_region_mult_linear ***
vs
linux
ining data points
ining data points
ining weights
data
n data

attach OC Worksheet object
copy independent variables
    
```



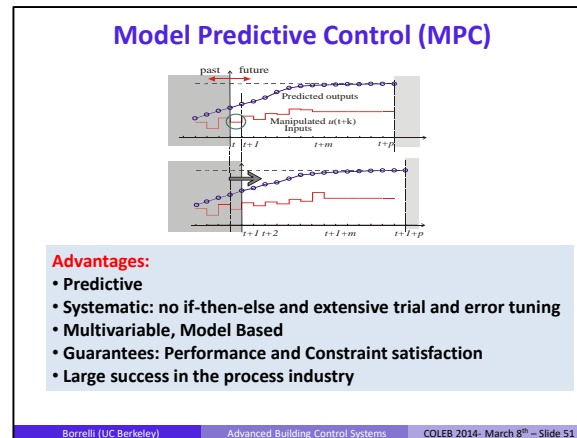
ENERGY EFFICIENCY @ SYSTEM LEVEL

MPC Used For Achievable Performance



System level considerations...heterogeneous system...

Examine limits of performance using MPC control design...



Implementation done in cloud environment...

System Complexity

The diagram shows a cloud icon on the left connected by a double-headed arrow to a floor plan of a building on the right. The BrightBox logo is in the top right corner.

- Each packaged unit needs 22 signals, each space 30 signals
- Building Alpha: 8 packaged units and 600 vav boxes: 18176 signals (sampling 5m)
- MPC: >300,000 vars. and >500,000 constraints (sampling 5 mins)
- Individual bacnet segments can be very slow. Link/Physical layers may include ARCNET, Ethernet, BACnet/IP, Point-To-Point over RS-232, Master-Slave/Token-Passing over RS-485, and LonTalk

KEY POINTS

Modelica is a modeling language that captures physics and is useful for modeling at the system level and for modeling heterogeneous systems (and both steady state & dynamics).

CCS is using Modelica for system level modeling and the Modelon tool chain to capture system level models and to deploy widely using library architectures, GUI and Python infrastructure.

The use of Modelica is on simulation but goes beyond in “systems engineering” the (re)use of models for variability and robustness analysis, optimization and analysis of design freedom, and control design and analysis.