WP1.2 MPC
Break-Out Sessions Summary

WP Leader: Lieve Helsen

Expert Meeting Paris
October 1-2, 2018

WP1.2 Plan for Break-Out Sessions

THE PLAN FOR BREAK OUT SESSIONS – DAY 1

BS 1-1	KPI		50 min	day 1 - session 1
	Discussion KPI matrix	Javier Arroyo	30 min	
		Draguna Vrabie		
	Modelica template for standardizing KPIs and inputs/outputs communication	David Blum	20 min	
BS 1-2	Emulators		60 min	day 1 - session 2
	Peer review of emulators	Filip Jorissen	40 min	
	checklists			
	tests			
	action plan emulator development	Lieve Helsen	20 min	

THE PLAN FOR BREAK OUT SESSIONS – DAY 2

BS 2-1	BOP-TEST		65 min	day 2 - session 1
	Introduction	David Blum	5 min	
	Example of User Needs	Draguna Vrabie	20 min	
	Current Status of Prototype	David Blum	20 min	
	Future Development	Kyle Benne	20 min	
BS 2-2	BOP-TEST		75 min	day 2 - session 2
	Discussion	David Blum	45 min	
	Documenting - user guide	David Blum	15 min	
	Action plan BOP-TEST	David Blum	15 min	
BS 2-3	MPC		75 min	day 2 - session 3
	Discussion MPC description template	Jan Drgona	45 min	
	Modelica Library for MPC: how to start up?	Filip Jorissen	30 min	
BS 2-4	Wrap-up and discussion		55 min	day 2 - session 4
	Prioritization of next steps, target timeline	Lieve Helsen	20 min	
	Action plan joint publications	Lieve Helsen	10 min	
	Keep open to discuss open issues	All	25 min	

WP1.2 – BS1-1

KPIs

9 attendees

KPI matrix – selection and quantification

- Control performance evaluation process
  - New controller
  - Control testing (simulation): select testing scenarios
  - Performance analysis (result analysis): define metrics, data processing
- Performance evaluation in existing studies: very fragmented, no unified approach → standardize approach
- Control performance metrics: large variety
- Matrix of metrics (KPI as a function of variables) → **core KPIs to compare controllers**
- Input and output requirements

KPI matrix – selection and quantification

- **Selection of core metrics**
  - Total building energy use in a given period
    (any energy vector we can model, scenarios defined determine which specific energy uses are calculated in addition, e.g. energy use for HVAC (heating, cooling, fans, pumps …), lighting)
  - CO_2 emission: emission factor needed → fixed for each emulator (boundary condition), this can be a profile
  - Flexibility: not yet decided (to be worked on), to be defined as a scenario that calculates the specific flexibility capability relative to a reference (given boundary conditions)

KPI matrix – selection and quantification

- **Selection of core metrics**
  - Installation metrics
    some are subjective, therefore qualitative measures
    - Engineering effort
      - Installation time
      - Installation knowledge level/training requirement
      - Installation cost
    - Richness of data
      - BAU or required extra excitements (+ specify which period(s) needed or give the data sets)
      - Required extra sensors or not (+ specify which, total cost can be calculated based on given costs per components – tables can be connected to the scenarios); emulator developer defines a set of sensors available

KPI matrix – selection and quantification

- **Selection of core metrics**
  - Total cost in a given period
    price profile of energy vectors is fixed for each emulator (boundary condition), specific archetypes of tariffs are defined (e.g. constant, moderately dynamic, highly dynamic) → each tariff is coupled to a specific scenario

  - Based on cost savings relative to a baseline controller, what is the cap-cost allowed to obtain a required pay-back period (e.g. 5 years)

KPI matrix – selection and quantification

- **Selection of core metrics**
  - <u>Thermal discomfort and air quality</u>
    total time when the comfort indicator (T, CO_2 …) is outside comfort range during a given period
    [Kh for T]
    What about humidity? Extremely important for radiant systems, less for air-based systems → emulators need to include humidity

KPI matrix – selection and quantification

- **Selection of core metrics**
  - Computational time
    - Controller prediction time (t_p)
    - Emulator simulation time (t_s) → peer review of emulator
    - Real building operation time (t_r)
    - Ratios of these

- **Other metrics are not in the core**
- **Not 1 global metric, but radar plot and box-plot for comparing the controllers**

WP1.2 – BS1-2

Emulators

12 attendees

Modelica template/guidelines for standardizing KPIs and inputs/outputs communication

- **Signal exchange blocks** for sensors and control actuation (to be added by emulator developer)
  - Control signal passes through or is overwritten
  - Put the sensors where you want in the model
- **Tags** for inclusion in KPI calculations
  - Annotation to sensor block? YES, annotation standards discussed in MPC template
  - Variable name and dictionary (external)?
  - Elevate a real output to high level model?
- **IBPSA Style** convention

Peer Review Emulators

- Documentation: qualitative description, assumptions/simplifications, inputs/outputs, control options (internal RBC if present), boundary conditions, benchmark specs
- Implementation = Docker consisting of
  - Documentation
  - Modelica model
  - Library dependencies
  - File dependencies
  - Translated FMU
  - BOP-TEST API

Peer Review Emulators

- Implementation:
  - emulator template (for core KPIs) or using the annotation approach?
    Scenarios: perfect prediction or given uncertainty on prediction.
    Weather file (external data) needs to be put somewhere (in BOP-TEST API), to be read by FMU.
    Standardization is needed, there is communication between FMU and BOP-TEST.

Peer Review Emulators

- Model requirements & Review by someone else than developer
  - Assert correct control inputs
  - Model is physical
  - Model is representative
  - Code documentation (IBPSA Style)
  - Clear Layout of hydraulics
  - Emulators are made in Modelica
- Unit testing & code storage
  - Library dependencies/versions → multiple repositories under IBPSA Project 1 Github? Use official releases, and commit number (if not official release)
  - Unit tests by developer

Action Plan Emulators

- Develop Emulators
- Document Emulators
- Review Emulators

} by next meeting (early Spring 2019)

- Make Emulators BOP-TEST ready

TIME SCHEDULE

Emulator	Developer	Developed	Documented	Reviewed by	BOP-TEST ready
BESTEST hydronic (1z)	Filip	In IDEAS	Not yet	Dave	
BESTEST air-based (1z)	Dave	To be started	Not yet	Filip	
8z residential hydronic heating	Valentin	ready	1st version	Krzysztof	
8z residential hydronic heating + air cooling	Valentin	ready	1st version	Krzysztof	
Air-based commercial (1z)	Dave	Almost ready	1st version	Jesus	
Hydronic commercial (1z)	Krzysztof	In progress	Not yet	Valentin	
5z air-based commercial	Dave	In BUILDINGS	1st version	Yanfea (Filip)	
mz hybrid office (simple)	Iago	Using IDEAS	Not yet	Valentin	
mz hybrid office (complex)	Filip	In progress	Not yet	PNNL?	
mz air-based prototype (complex)	Sen Huang	ready	1st version	Iago	

WP1.2 – BS2-1

BOP-TEST

13 attendees

BOP-TEST discussion

- **Workflow for use case**
  - MPC controlled prototype building emulator presented by Draguna (MPC in Julia, using Python to interface with Modelica in Dymola)
  - Data stored and used for calculate on of KPIs and check of calculation efficiency
- **Project Alfalfa** (made the bridge between NREL, LBNL, PNNL which led to the DoE funded project on BOP-TEST)
  - Collaboration between J2innovations (a Siemens company) and NREL
  - Open source standard Haystack (tagging conventions)
  - Alfalfa connects EnergyPlus -> Haystack -> FIN (docker-based)
  - Focus on organizing software, less on research

BOP-TEST discussion

- Make **list of use cases we would like to support in BOP-TEST**; BOP-TEST needs to allow both options:
  - Real-time building running (observe the effect on actual control of the building, and check computation time) = realistic testing environment
  - Comparing/benchmarking control algorithms → simulation will run only when it gets a controller action

- **Testing on own FMU** should be possible.

WP1.2 – BS2-2

BOP-TEST

12 attendees

BOP-TEST, development requirements and guide

Document is available

1. Add ‘goal of BOP-TEST’ and methods for benchmarking
2. User Case Narratives
   Specify scenario and time control use cases
3. Test Case Development
   Filip: align with slides of Paris meeting
   FMU should be generated without restrictions regarding licenses
   Different FMUs for different scenarios (e.g. weather, pricing) – discussion about emission factors (important for hybrid systems, should be consistent with pricing)
   What makes up the benchmark test is a next step, now it is about functionality
   KPI tagging: use string parameter instead of annotations?

BOP-TEST, development requirements and guide

Annotation standardized in MPC template

Filip to align peer review process with slides
Unit testing based on simulation, needs to include some baseline control

4. KPI specification and calculation
   Javier to align KPI section (core KPIs and calculation module) with slides and conclusions discussion
5. Run-time platform architecture
   Benne to complete
6. User guide
   Krzysztof to implement forecast

BOP-TEST, first prototype test

IT WORKS!

Feedback to be sent to Dave:

- Python 2 (official support is extended to 2020) versus Python 3 → suggestion to move to Python 3, the sooner the better

WP1.2 – BS2-3

MPC

13 attendees

MPC description template

- Control engineering notation (abstract domain formulation), can be non-linear → allows mathematical comparison (sparsity of matrix, size of problem …).
- Translation physical → abstract formulation in an automated way? Future work.
- MPC variables notation and translation between physical and abstract domain, including slack variables
- MPC parameters
- MPC formulation objectives - classification
- MPC formulation constraints
- Documentation IBPSA Project 1 repository: all can contribute
- Example case next meeting

MPC description template

Action Plan:

- Iago: complete table based on KPIs and put on IBPSA Project 1 repository
- All: check document and add/correct where needed
- Jan: an example case to be worked out

Modelica Library for MPC

- Common library that satisfies the requirements for MPC
- 3 attendees are using MPC based on Modelica equations (JModelica). DOML may be an alternative (Dave sends link around).
- Controller models are simpler
  - Less dynamics in HVAC system (remove dynamics to make them steady-state)
  - Simpler equations, e.g. regularizations more compact (fewer operations)
  - Model equations need to be twice continuously differentiable with respect to optimization variables to converge, e.g. switching between regimes (If … then … else …)
  - Our models need to be compliant with JModelica (no integer problems)

Modelica Library for MPC

- Controller models are simpler (ctn.)
  - External C-functions can not be used in Casadi, thus not compliant with JModelica
  - T as default state instead of internal energy
- JModelica allows to extract the equations → useful for control community however more interest in equations of controller model (not emulator model), however these are not always in (J)Modelica, and controller developer needs to take care.
- Optimization models contain optimization constructs (e.g. constraints) that will not be recognized by Dymola. Solution:
  - Dummy input → possible extension for the future

Modelica Library for MPC

- Parameter estimation needed
- Zone models? More difficult to agree on (easier for HVAC systems), possible extension for the future
- IBPSA library is the core, we start from these components
- Where to put this MPC library? Three possibilities exist for the controller model:
  - IBPSA model directly used
  - Extend IBPSA model
  - Based on IBPSA model however code is modified

  → Separate library that implements the IBPSA library (cfr IDEAS …)

Modelica Library for MPC

- Simulation-based optimization → use IBPSA library, however will be slower
- Type of constraints
  - Simple constraints: min/max
  - Time-based constraints: more complex → extension for the future
- Optimization or simulation-based unit tests? Both are usefull
  - Simulation: stretch the range
  - Optimization: test in optimization

Modelica Library for MPC

- Timeframe
  - Main drivers are Filip, Dave, Iago, using already Modelica for MPC
  - Overview of models we have already
  - Merge existing models already now and start building up experience and gaining insights → separate library
  - By next meeting

WP1.2 – BS2-4

Wrap-up

Next steps, priorities, target timeline

- **KPIs** (9 attendees):
  - Core KPIs selected. **Not 1 global metric,** but radar plot and box-plot for comparing the controllers
  - Javier presents core KPIs and their quantification in the development requirements and guide document
  - Unsolved: flexibility KPI → to be defined as a scenario that calculates the specific flexibility capability relative to a reference, action Javier and Krzysztof

Next steps, priorities, target timeline

- **Emulators** (12 attendees)**:**
  - Modelica template for standardizing KPIs and Input/output communication:
    - Signal exchange blocks for sensors and control actuation
    - Tags for inclusion in KPI calculations
    - IBPSA Style convention
  - Peer review Emulators:
    - Documentation
    - Implementation
    - Model requirements & Review by someone else than developer
    - Unit testing by developer (stress testing) & code storage
    - Filip presents peer review of emulators in the development requirements and guide document, creates checklist for review
    - Dave sets guidelines for unit tests (stress testing: model should pass in specific ranges)

Next steps, priorities, target timeline

- **Emulators** (12 attendees)**:**
  - Action plan Emulators
    - Develop Emulators
    - Document Emulators
    - Review Emulators

    } by next meeting (early Spring 2019)
    - Make Emulators BOP-TEST ready

Next steps, priorities, target timeline

Emulator	Developer	Developed	Documented	Reviewed by	BOP-TEST ready
BESTEST hydronic (1z)	Filip	In IDEAS	Not yet	Dave	
BESTEST air-based (1z)	Dave	To be started	Not yet	Filip	
8z residential hydronic heating	Valentin	ready	1st version	Krzysztof	
8z residential hydronic heating + air cooling	Valentin	ready	1st version	Krzysztof	
Air-based commercial (1z)	Dave	Almost ready	1st version	Jesus	
Hydronic commercial (1z)	Krzysztof	In progress	Not yet	Valentin	
5z air-based commercial	Dave	In BUILDINGS	1st version	Yanfea (Filip)	
mz hybrid office (simple)	Iago	Using IDEAS	Not yet	Valentin	
mz hybrid office (complex)	Filip	In progress	Not yet	PNNL?	
mz air-based prototype (complex)	Sen Huang	ready	1st version	Iago	

Next steps, priorities, target timeline

- **BOP-TEST** (13 and 12 attendees)**:**
  - Make list of use cases we would like to support in BOP-TEST
  - Development requirements and guide

    Document is available and needs to be completed

    1. Add ‘goal of BOP-TEST’ and methods for benchmarking
    2. User Case Narratives
       Specify scenario and time control use cases
    3. Test Case Development
       Filip: align components in test case with slides of Paris meeting
       FMU should be generated without restrictions regarding licenses
       What makes up the benchmark test is a next step, now it is about functionality

Next steps, priorities, target timeline

- **BOP-TEST** (13 and 12 attendees)**:**
  - Development requirements and guide

    KPI tagging: use string parameter
    Annotation of KPIs standardized in MPC template

    Filip to align peer review process with slides
    Dave: unit testing based on simulation (stress testing), needs to include some baseline control

  4. KPI specification and calculation
     Javier to align KPI section (core KPIs and calculation module) with slides and conclusions discussion
  5. Run-time platform architecture
     Kyle to complete
  6. User guide
     Krzysztof to implement forecast

Next steps, priorities, target timeline

- **BOP-TEST** (13 and 12 attendees)**:**
  - Python 2 (official support is extended to 2020) versus Python 3 → suggestion to move to Python 3, the sooner the better
  - User interface development: topic for later

Next steps, priorities, target timeline

- **MPC** (13 attendees)**:**
  - MPC description template
    - Translation physical → abstract formulation in an automated way? Future work.
    - Iago: complete table based on KPIs and put on IBPSA Project 1 repository
    - All: check document and add/correct where needed
    - Jan: an example case to be worked out
  - MPC Modelica Library
    - Main drivers are Filip, Dave, Iago, using already Modelica for MPC
    - Overview of models we have already
    - Merge existing models already now and start building up experience and gaining insights → separate library, by next meeting

Action plan joint publications and dissemination

- BOP-TEST framework @ BS 2019 Conference paper (submitted)
- Modelica 2019 Conference
- Review paper MPC for journal (in preparation)
- IBO Workshop 2020: Show BOP-TEST
- American/European Control Conference (ACC/ECC/CDC): present BOP-TEST to control community
- Data science conferences (December ICMLA 2019): present emulators and BOP-TEST framework to machine learning community
- Conference (BS, ASHRAE) and journal papers on assessment of MPC algorithms
- BOP-TEST Workshop connected to BS2021 in Bruges (Belgium)

Acknowledgements

This is part of the projects that have received funding from the European Union's Horizon 2020 research and innovation programme under grant agreements:

No 723649 hybridGEOTABS

No 731231 FHP

No 656889 GEOTeCH