Edmund Marth
edmund.marth@jku.at

Patrick Zorn
patrick.zorn@jku.at

Tolerance Analysis with SyMSpace

SyMSpace Days 2024
September 18-19, Linz

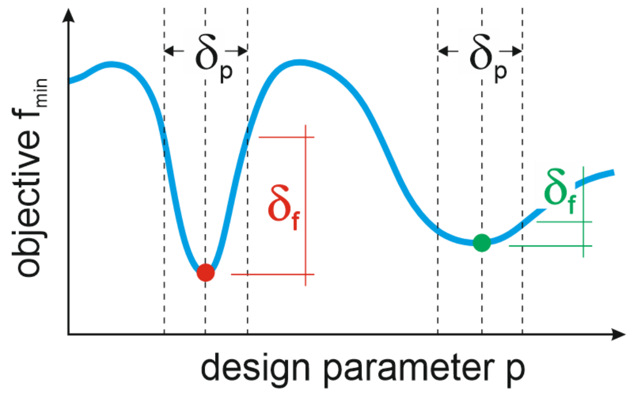
Institute for Electrical Drives and Power Electronics
LINZ CENTER OF MECHATRONICS

Motivation

Motivation

- With increasing capabilities of design tools (e.g. SyMSpace ;-)), highly optimized systems can be achieved
- Unfortunately, highly optimized systems are prone to be highly sensitive
- “Reality is seldom ideal”
- Effects like
  - Material uncertainties
  - Part- and assembly tolerances
  - Variations of external influences
  - …

  can make a (nominally) optimal system behave bad!

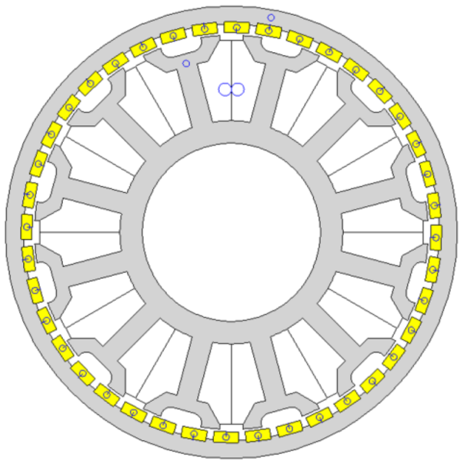


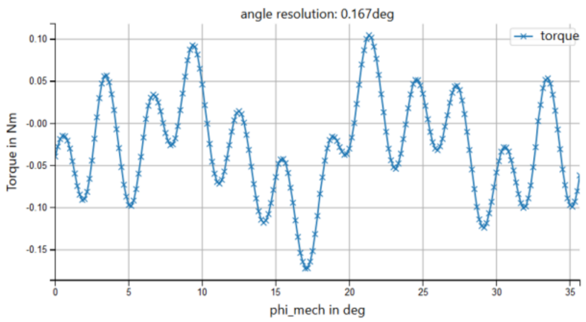
Low sensitivity against uncertainties is crucial for a robust design!

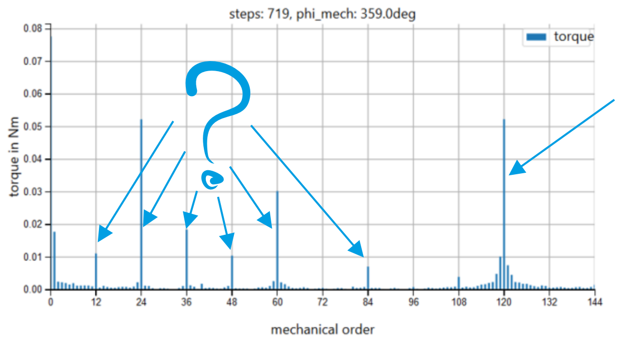
Motivation

10-fold higher cogging torque measured than simulated!

- Vernier machine
  - Tn = 2Nm
- Estimated cogging torque with SyMSpace standard motor project
  - Tr,cogg,pp ~ 0.03 Nm
- Measured cogging torque
  - Tr,cogg,pp ~ 0.28 Nm







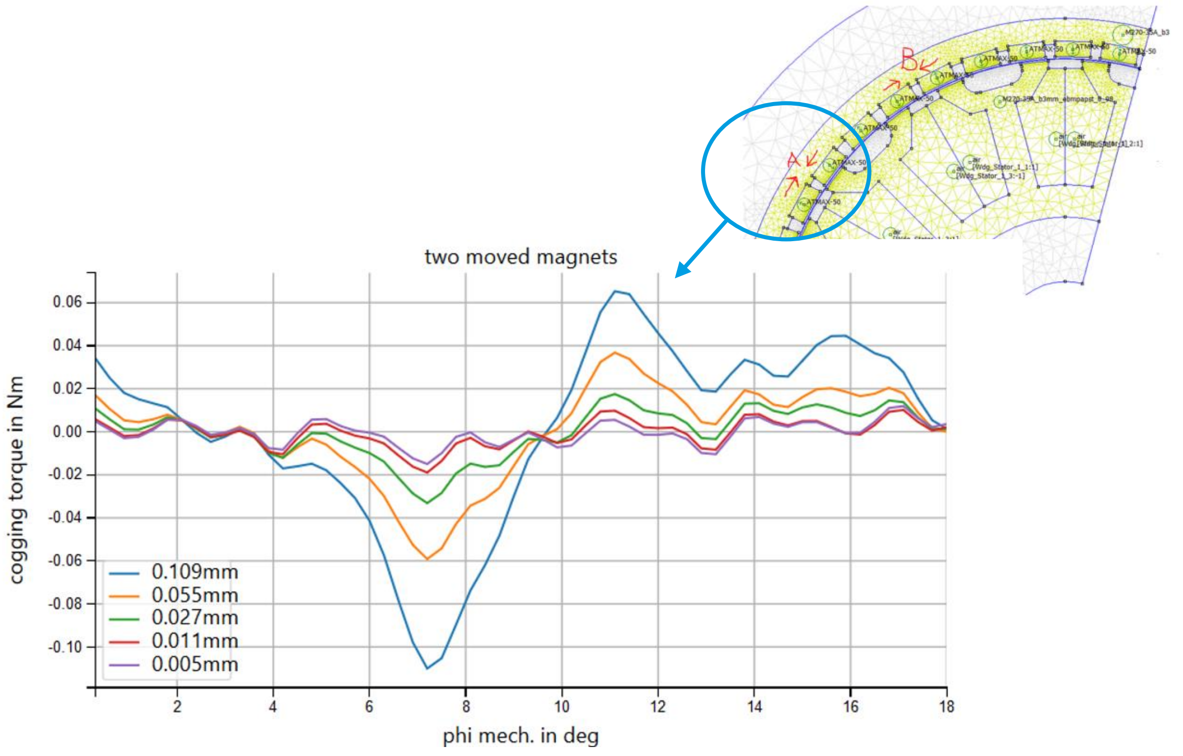
first harmonic order to be expected from theory!

Motivation

Where do the lower harmonics come from ?

- Testing …
  - various mesh settings
  - different calculation methods (e.g. sliding band boundary in the airgap)
  - eccentricity of rotor/stator
  - full/sector mode
  - material degradation
  - … showed minor (negligible) influence on cogging torque
- Displacing magnets showed significant influence!

Starting Point of SyMSpace Tolerance Framework



Agenda

- SyMSpace Framework for Tolerance Analysis
- Tolerable Geometry Models
- Sensitivity Analysis using **C**umulative-**D**istribution-**F**unctions (CDFs)
- Examples
  - Impacts of an asymmetric slot on the cogging torque of a PMSM
  - Driving range sensitivity analysis of a BEV
- Tolerances and Optimization (Outlook)

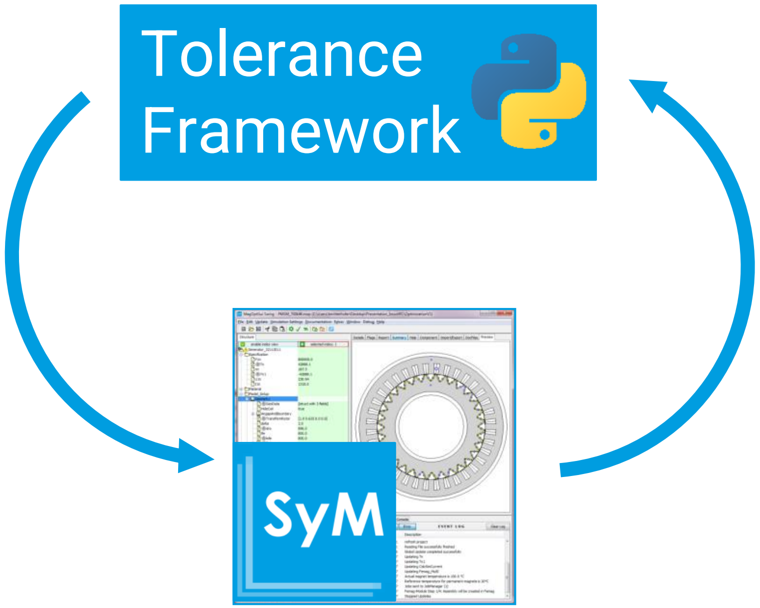
SyMSpace Tolerance Framework
Tolerance Analysis in SyMSpace

Implementation, Setup, How-To

Tolerance analysis in SyMSpace

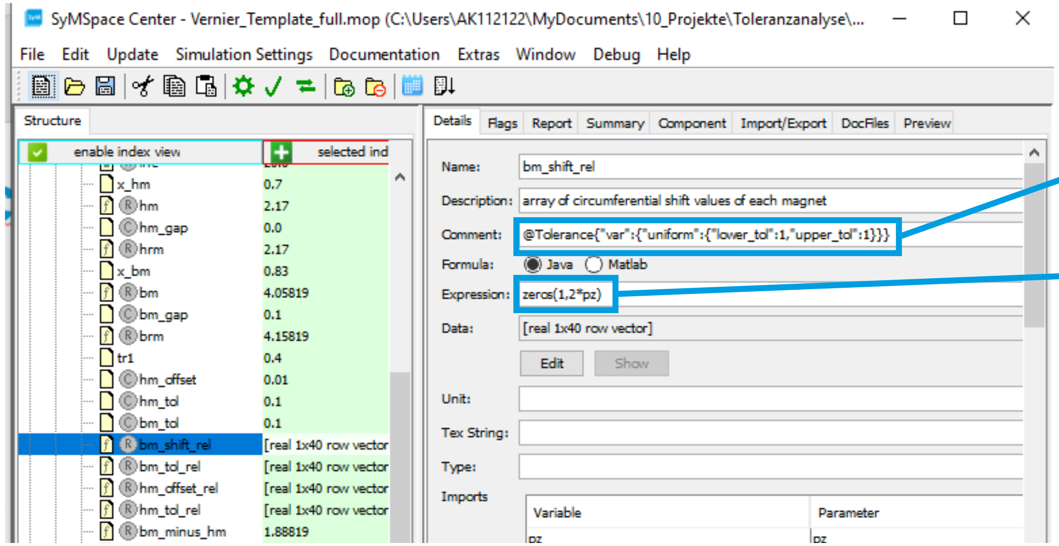
… under construction …

- Framework for tolerance analysis is still in beta phase
- Controlled via command line over SyMSpace Python Console
- No automated post-processing workflow yet (as, e.g., for standard motor projects)



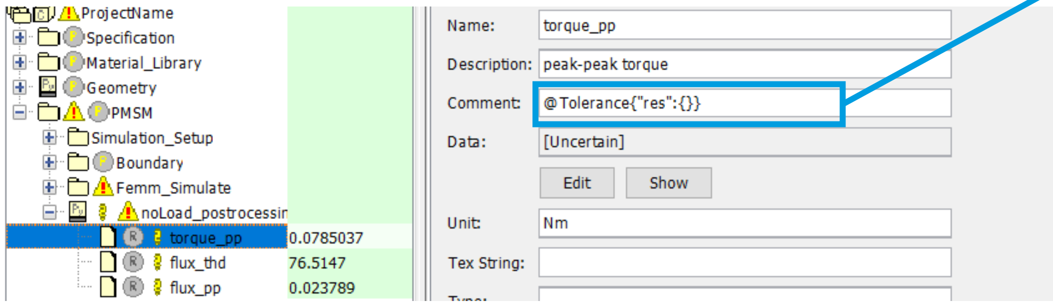
Tolerance analysis in SyMSpace

Defining tolerance affected variables and results



@Tolerance{"var":{"uniform":{"lower_tol":1,"upper_tol":1}}}

define nominal value (tolerance is added)

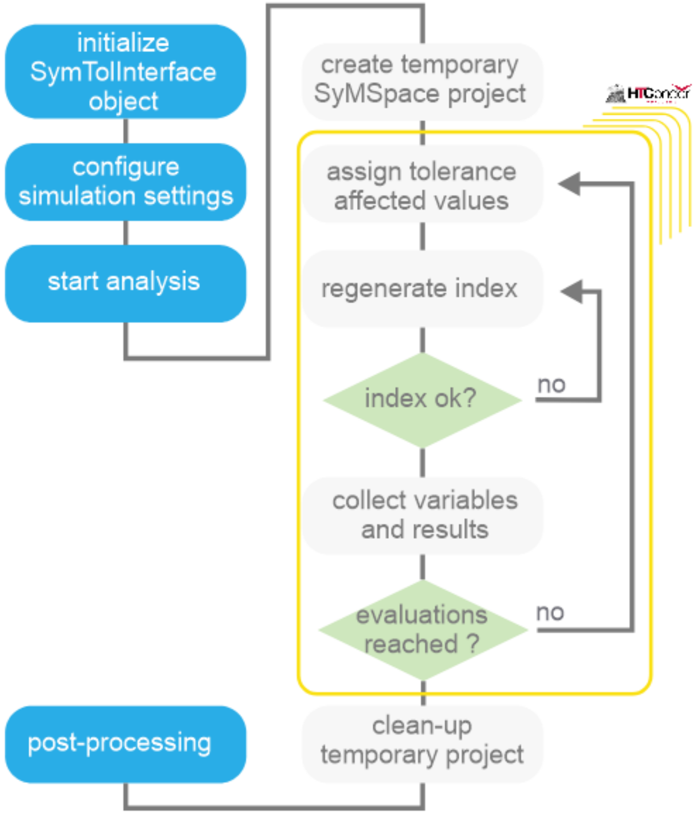


@Tolerance{“res":{}}

- define tolerance affected parameters using @Tolerance{} keyword in comments field
- @Tolerance{} information provided in json format

Tolerance analysis in SyMSpace
Workflow/ Analysis

- Prototype implementation: `SymTolInterface` python class
  - Sphinx documentation available, describing basic workflow step by step
- Create instance assigning SyMSpace project handle within SyMPython Console
  `sti = SymTolInterface(CurProject)`
- Configure simulation settings
- Start tolerance analysis with `sti.start()`
  - project tree is analyzed for @Tolerance keywords
  - original SyMSpace project is cloned
  - indices are processed similar to Optimizer
  - folder <SymProject>.tol.files (similar to < SymProject >.mop.files) is created to store full index data if desired
  - variables and results of analysis are pickled
- For postprocessing of scalar results: (e.g., Tcogg,pp)
  - functions to evaluate CDF and related confidence parameters and visualization functions are available



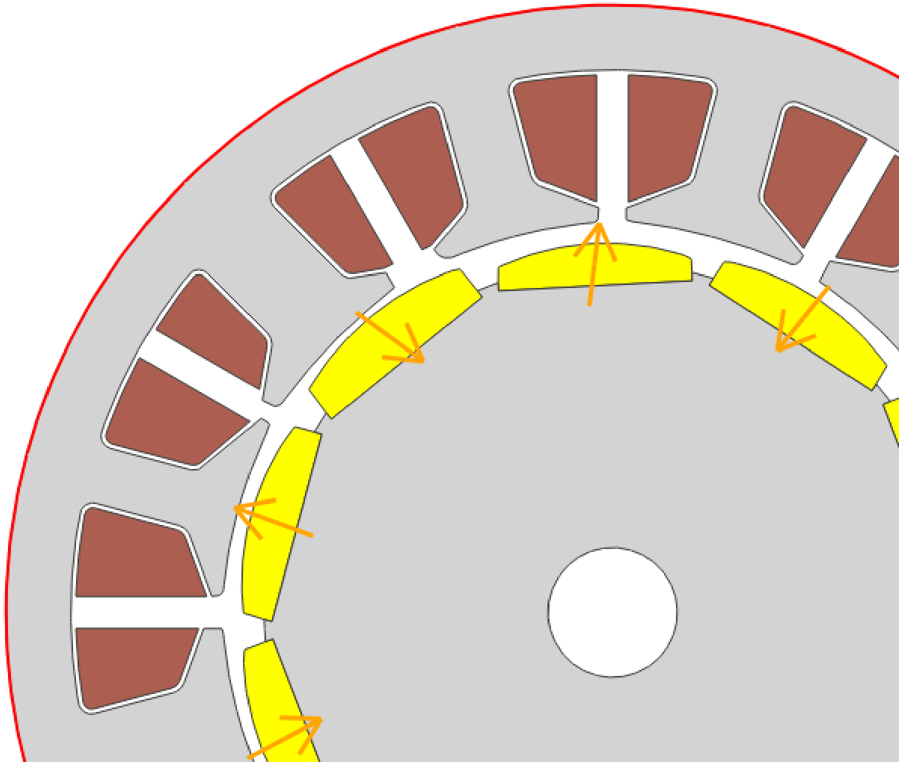
Tolerable Geometry Models

Classes, Implementations, Restrictions

Tolerable Geometry Models

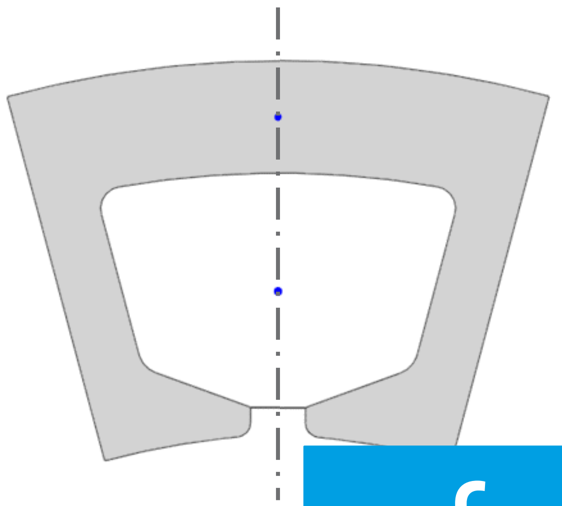
Implementation (alpha status)

- Functionality is added to the segment geometries (python script based)
- Functions implemented in a separate development branch of SyMSpace
  - as it is alpha status → need for verification and use cases
  - separate installer can be provided
- All script based stator and rotor topologies are capable to apply tolerances (CAD models don't work)
  - few adaptations necessary to the geometry definition

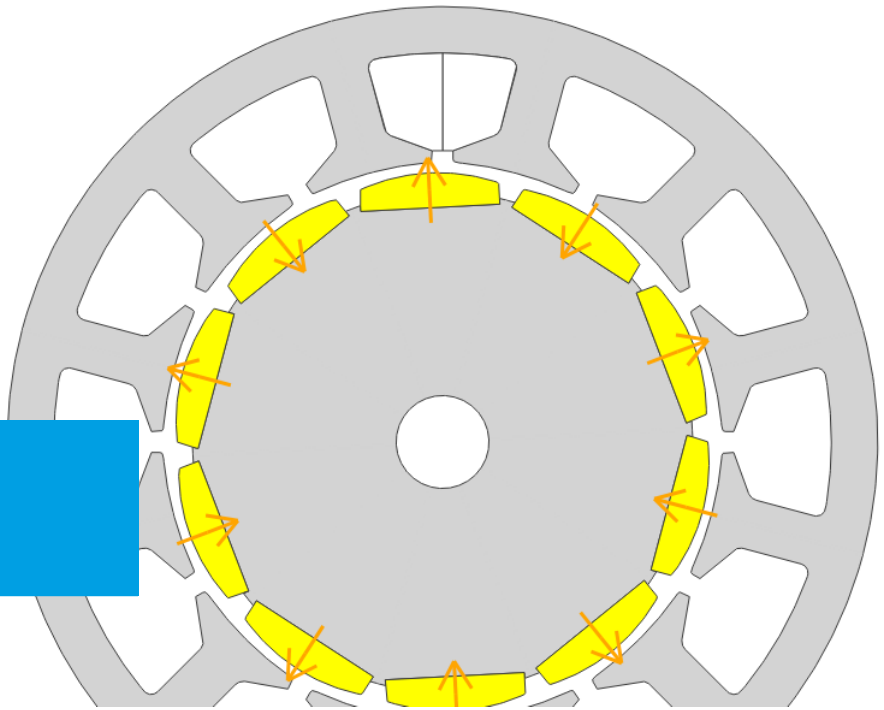


Tolerable Geometry Models

Standard Segment-Geometry Modeling



1. defining one half of the slot/pole
2. mirroring of the half to obtain one complete slot/pole configuration
3. Segment of the slot/pole is duplicated Ns resp. 2*pz times and transformed to obtain the entire cross-section

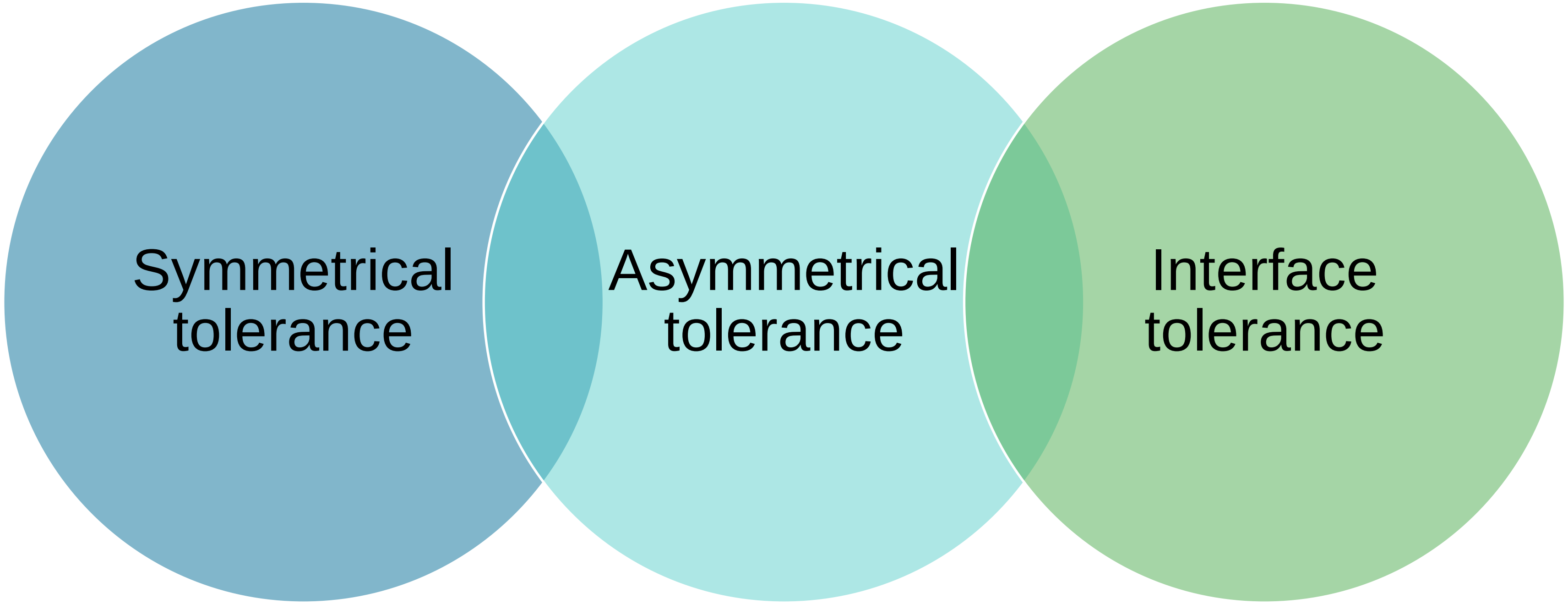


fully symmetrical

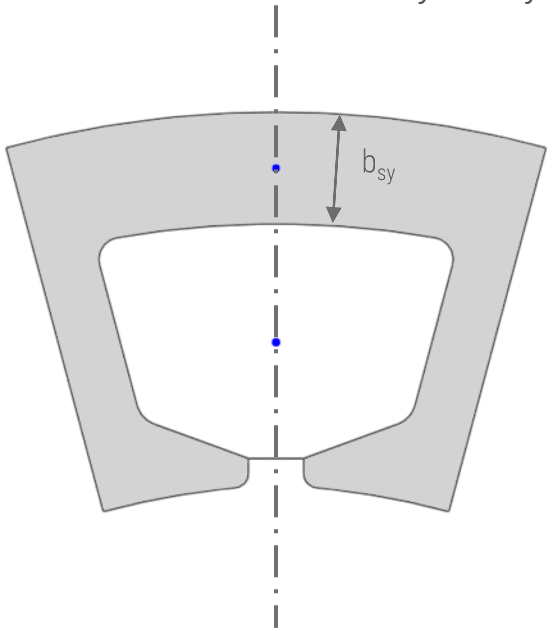
Tolerable Geometry Models

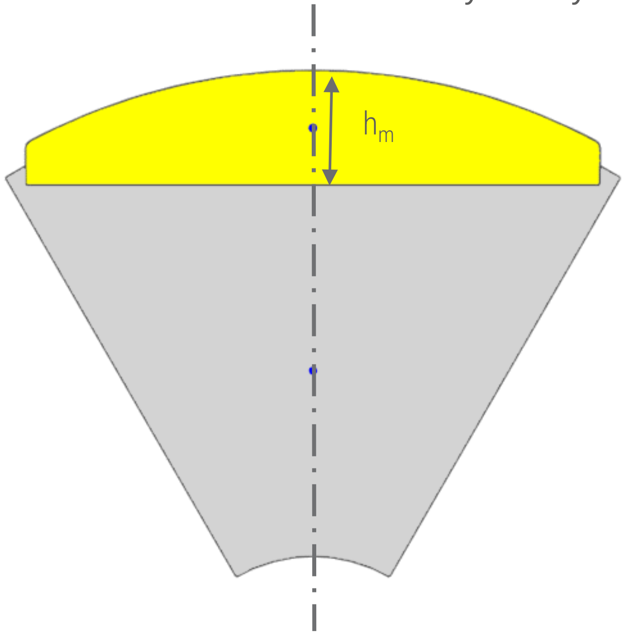
Segment-related Tolerance Classes

3 different tolerance classes are implemented



Tolerable Geometry Models
Symmetrical Tolerance





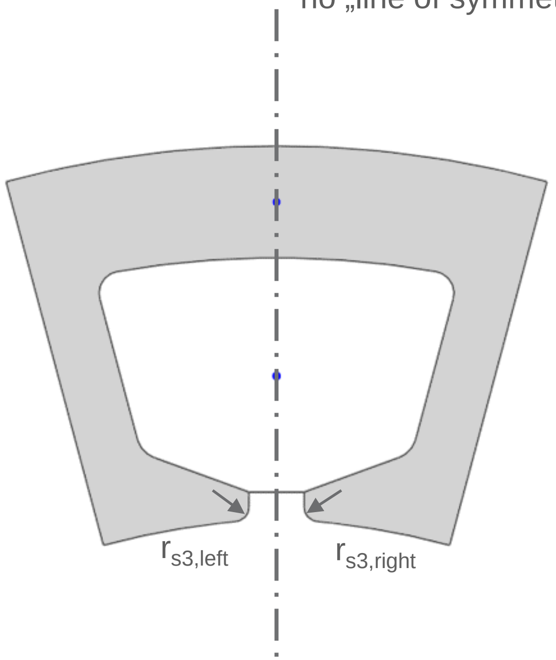
symmetrical tolerance values are provided by a [1 x Ns] resp. [1 x 2*pz] vector. e.g. for Ns = 12

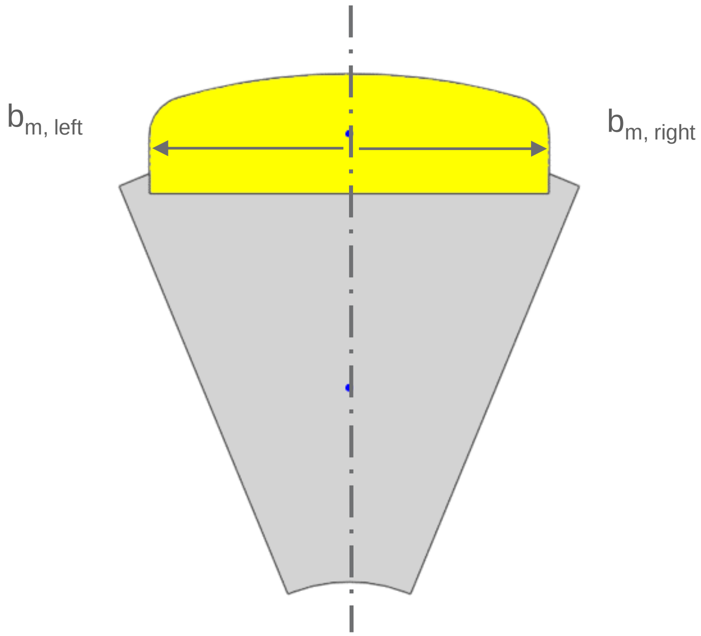
bsy__tol =

	0	1	2	3	4	5	6	7	8	9	10	11
0	-0.1	1.5	-0.8	0.2	0.0	0.5	-0.5	0.2	0.3	-0.3	0.4	-0.2

Tolerable Geometry Models

Asymmetrical Tolerance





asymmetrical tolerances are provided by a [2 x Ns] resp. [2 x 2*pz] vector. e.g. Ns = 12;

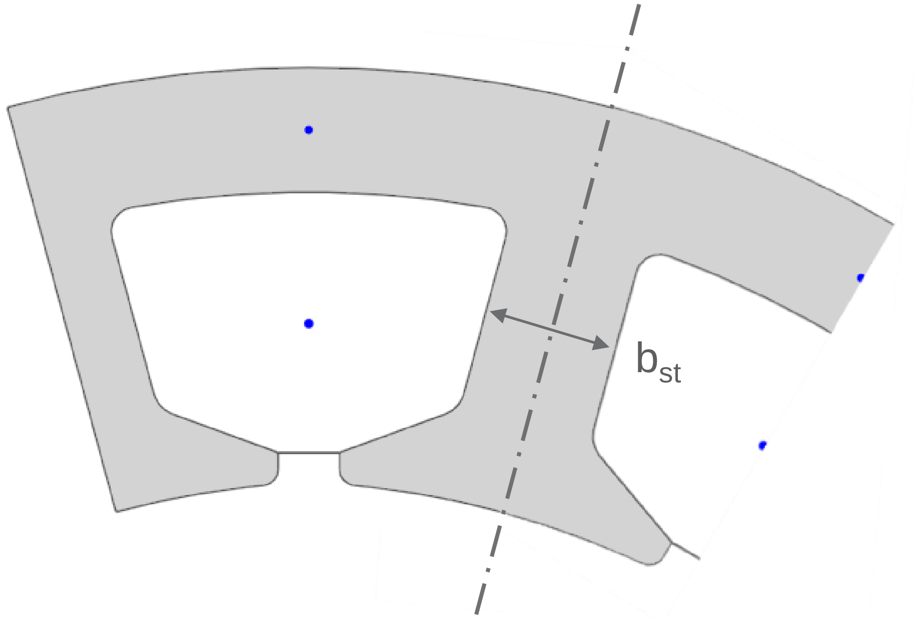
rs3__tol =

	0	1	2	3	4	5	6	7	8	9	10	11
0	-0.1	0.15	-0.05	0.2	0.0	0.18	-0.06	0.12	0.03	-0.13	0.14	-0.2
1	0.1	-0.15	0.05	-0.2	-0.0	-0.16	0.08	-0.18	-0.07	0.15	-0.11	0.2

Tolerable Geometry Models

Interface Tolerance

„Interface line of symmetry“



Interface Toleranzen werden mit einem [1 x Ns] bzw. [1 x 2*pz] Vektor angegeben. z.B.: Ns = 12;

bst__tol =

	0	1	2	3	4	5	6	7	8	9	10	11
0	-0.1	1.5	-0.8	0.2	0.0	0.5	-0.5	0.2	0.3	-0.3	0.4	-0.2

SyM

JKU Institute for Electrical Drives and Power Electronics

LCM LINZ CENTER OF MECHATRONICS

Tolerable Geometry Models

Definition of a Tolerance Class for a Parameter

- Appending tolerance class information to the variable definition
- The tolerance field for the related parameter is added automatically as import variable of the geometry

```
def geometryVars(self) -> list:
    """Geometry variables..."""
    geovar = [
        ('Ns', 'Number of slots', '', False, 'Numeric / [1 x 1]', 12, 'none'),
        ('dso', 'Stator outer diameter', 'mm', False, 'Numeric / [1 x 1]', 75.0, 'none'),
        ('dsi', 'Stator inner diameter', 'mm', False, 'Numeric / [1 x 1]', 48.0, 'none'),
        ('bst', 'Width stator tooth', 'mm', False, 'Numeric / [1 x 1]', 4.3, 'interface'),
        ('bsy', 'Stator yoke width', 'mm', False, 'Numeric / [1 x 1]', 4.0, 'none'),
        ('hss', 'Pole shoe height', 'mm', False, 'Numeric / [1 x 1]', 1.0, 'interface'),
        ('bss', 'Slot opening', 'mm', False, 'Numeric / [1 x 1]', 2.0, 'symmetric'),
        ('rs1', 'Radius slot 1', 'mm', False, 'Numeric / [1 x 1]', 0.8, 'none'),
        ('rs2', 'Radius slot 2', 'mm', False, 'Numeric / [1 x 1]', 1.0, 'none'),
        ('rs3', 'Radius slot 3', 'mm', False, 'Numeric / [1 x 1]', 0.5, 'asymmetric'),
        ('phis1', 'Pole shoe angle', 'deg', False, 'Numeric / [1 x 1]', 125, 'interface'),
        ('est', 'Stator tooth eccentricity', 'mm', False, 'Numeric / [1 x 1]', 5.0, 'interface')
    ]
    return geovar
```

Tolerable Geometry Models

Splitting the Geometry to Half Segments

- each half segment is parametrized differently in general because of asymmetrical or interface tolerances
- necessary to split up geometry script
- add code which handles left and right half creation to main geometry definition

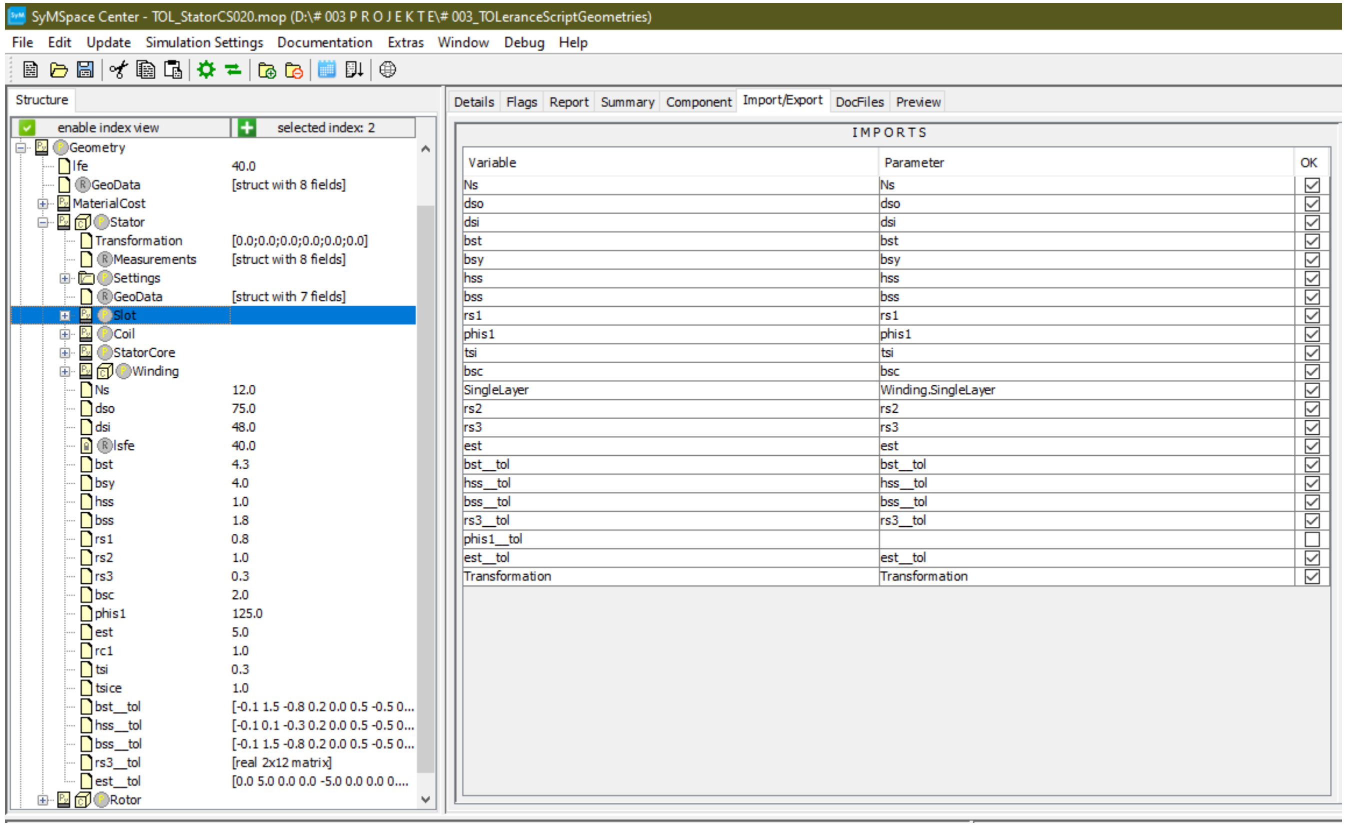
```
def geometry(self, param: dict, part: Geometry2D) -> None:
    """Geometry of stator segment..."""
    isTol = False
    for pv in param.values():
        if isinstance(pv, list) or isinstance(pv, np.ndarray):
            isTol = True
            break
    if not isTol:
        dso = param['dso']
        dsi = param['dsi']
        bsy = param['bsy']
        hss = param['hss']
        pnt = self.halfSegment(param=param, part=part)
        elem = part.mirror(Line( *args: [0, 0], [0, 1]), copy=True)
        # Line coil - closing coil area
        pnt_hss = pnt
        elem.append(Line( *args: pnt_hss, np.array([-pnt_hss[0], pnt_hss[1]])))
        ...t.addSegment(elem)

        ... copy.deepcopy(param)
        ...s = []
        ...amTolClass = self.getGeometryVarToleranceClass()
        for i in range(2):
            for k, v in paramTolClass.items():
                if v != 'none':
                    p[k] = param[k][i] if isinstance(param[k], list) else param[k]
            tmpPart = Geometry2D()
            pnts.append(self.halfSegment(param=p, part=tmpPart))
            if i == 1:
                tmpPart.mirror(axis=Line( *args: [0, 0], [0, 1]), copy=False)
            for j in range(tmpPart.numseg):
                part.addSegment(tmpPart.getSegment(j))
```

This part is intended to be redesigned for easier use in future!

Tolerable Geometry Models

Import fields for tolerance parameters

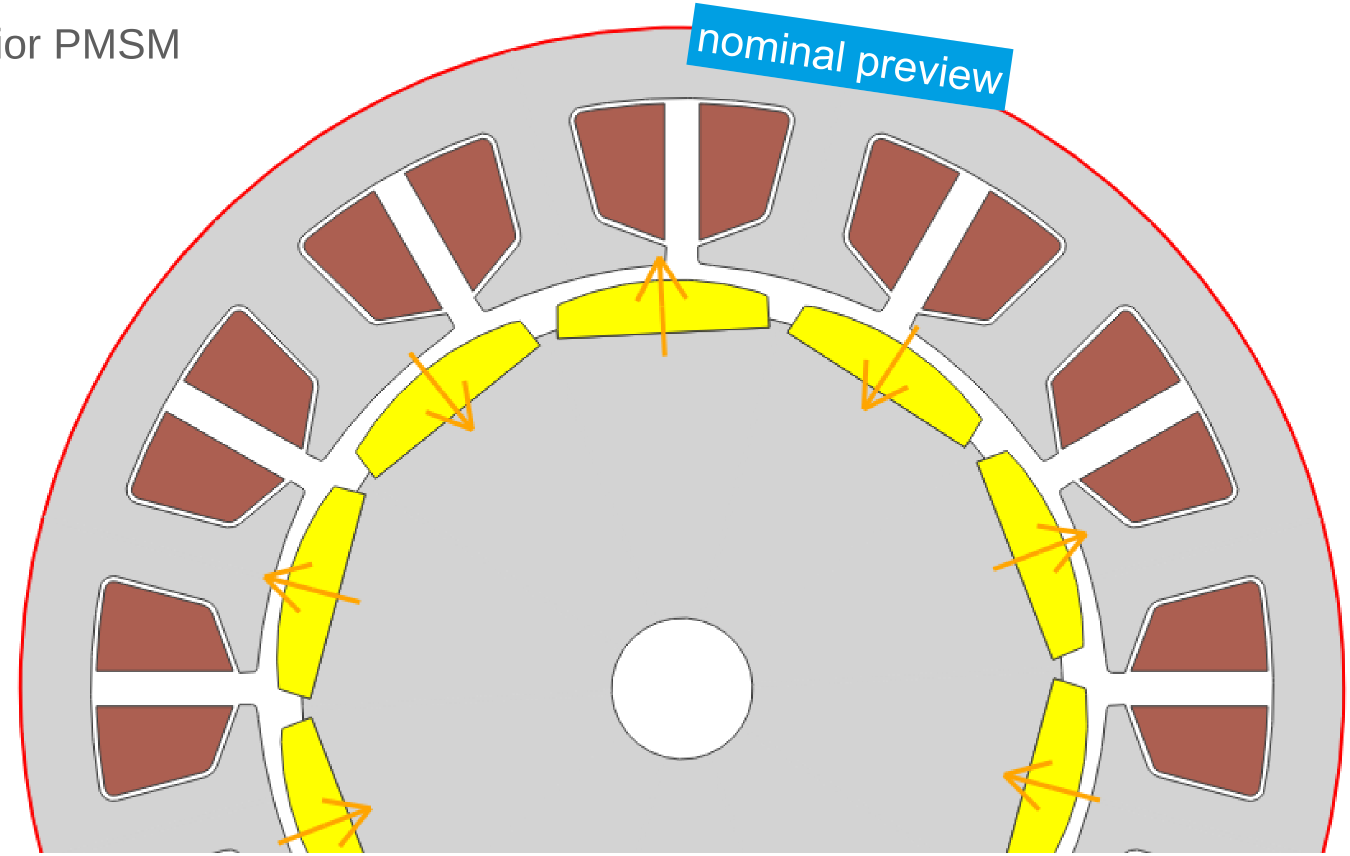
- tolerance fields are always optional
  - not all have to be assigned
  - can be left empty
- proper dimension is checked and error displayed if violated
- dimension requirement is also shown at mouse-over on import variable
- tolerance value is added to nominal value with sign



Tolerable Geometry Models

Preview of a Geometry with Tolerances

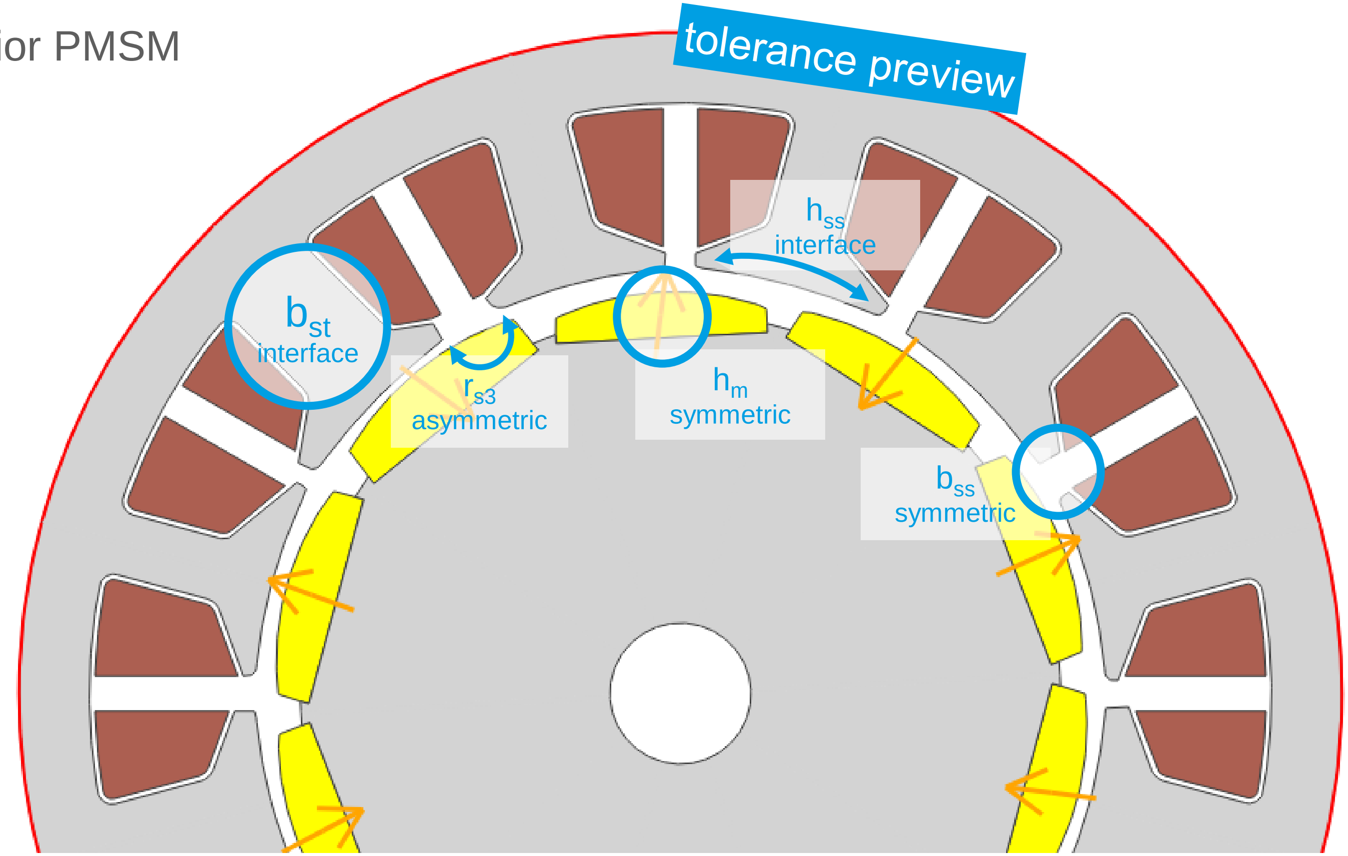
THERM2D preview of an interior PMSM without tolerances



Tolerable Geometry Models

Preview of a Geometry with Tolerances

THERM2D preview of an interior PMSM including tolerances



Tolerable Geometry Models

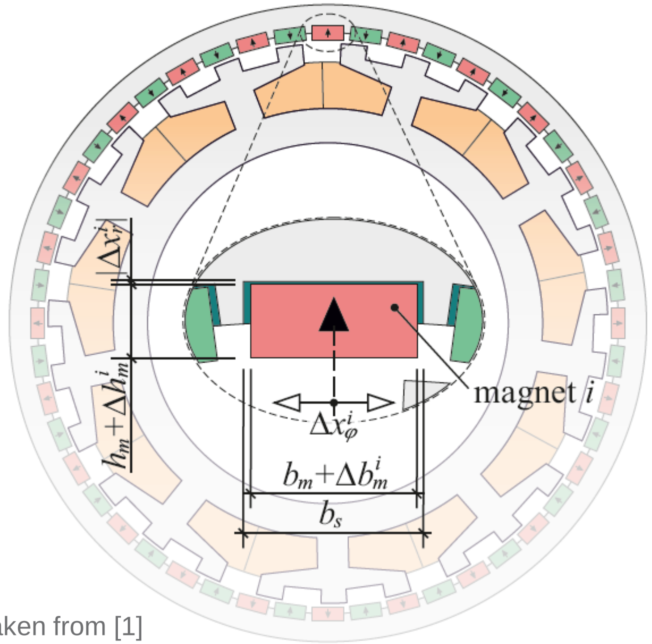
General Information

- Winding/Wire is placed within the nominal slot
  - as tolerances are small in general there should be no change to be expected for the coil layout
- Simulation settings need to be adapted
  - evaluation of the entire cross-section – no usage of symmetries possible
  - simulation time will be increased
- Intended to work seamlessly with the tolerance analysis python module – SymTolInterface

Evaluation of a Tolerance Analysis

Evaluation of a Tolerance Analysis

The Problem …



taken from [1]

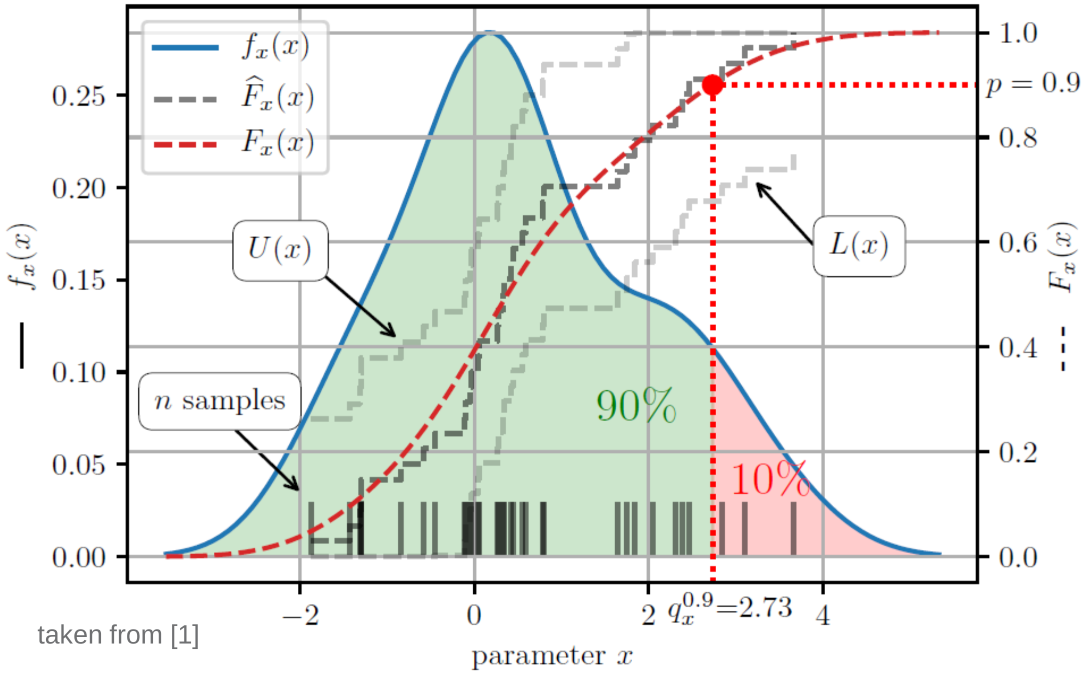
Example: Vernier motor

- 4 tolerance-affected parameters per magnet
  - width
  - height
  - circumferential position
  - radial position
- 48 magnets
- assume 3 discretization steps per tolerance-affected parameter
  - $(3^4)^{48} = 3^{192} > 4 \cdot 10^{91}$ possible variations
- assume $1\mu s$ to calculate one variation
  - time to calculate all variations $> 10^{78}$ years
  - symmetry conditions not taken into account ;-)

[1] Marth, E.; Bramerdorfer, G. On the Use of the Cumulative Distribution Function for Large-Scale Tolerance Analyses Applied to Electric Machine Design. Stats **2020**, 3, 412-426. https://doi.org/10.3390/stats3030026

Evaluation of a Tolerance Analysis

Approach using CDF (**C**umulative **D**istribution **F**unction)



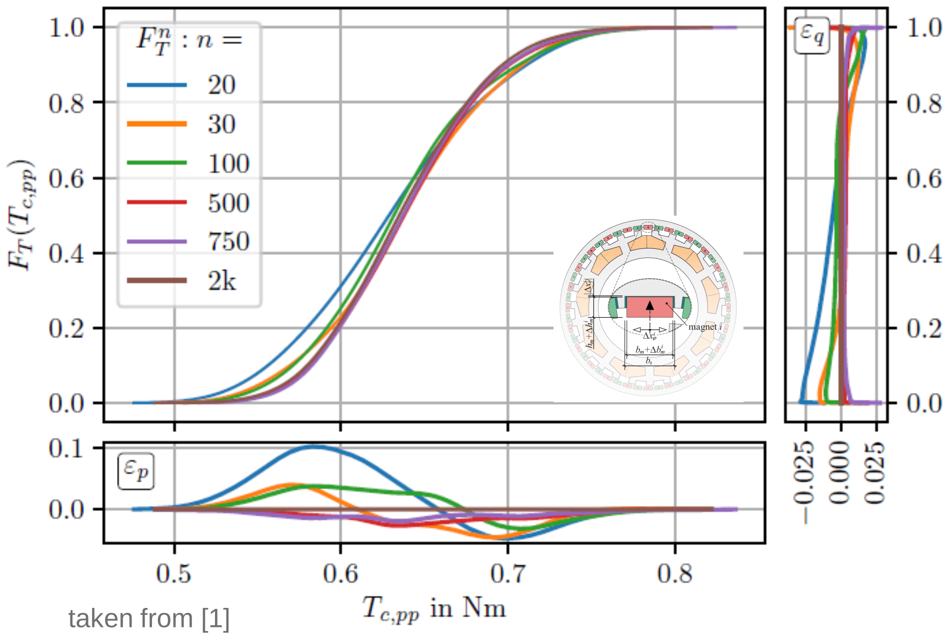
taken from [1]

- F_x cumulative distribution function
- f_x probability density function
- ${q_x}^p$ p-quantile value
- p probability
- L, U lower and upper confidence band

[1] Marth, E.; Bramerdorfer, G. On the Use of the Cumulative Distribution Function for Large-Scale Tolerance Analyses Applied to Electric Machine Design. Stats **2020**, 3, 412-426. https://doi.org/10.3390/stats3030026

Evaluation of a Tolerance Analysis

Approach using CDF (**C**umulative **D**istribution **F**unction)



taken from [1]

Example: Vernier motor

+ Convergence **after 1000 samples**!

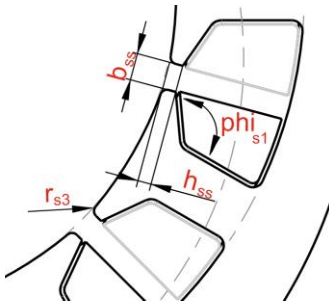
– Worst Case most likely not covered

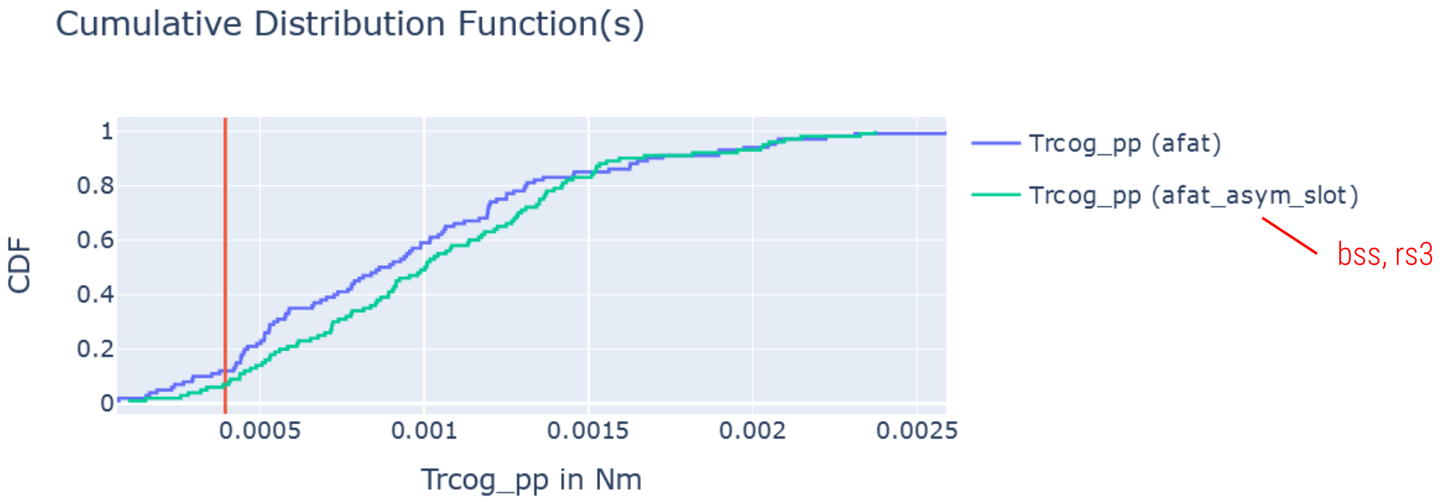
[1] Marth, E.; Bramerdorfer, G. On the Use of the Cumulative Distribution Function for Large-Scale Tolerance Analyses Applied to Electric Machine Design. Stats ***2020****, 3, 412-426. https://doi.org/10.3390/stats3030026*

Example

Asymmetrical Stator Slot
symmetric vs. asymmetric slot tolerances

Varying Parameters





Driving Range Sensitivity Analysis of a BEV

Based on Car and Motor Model of Tesla3

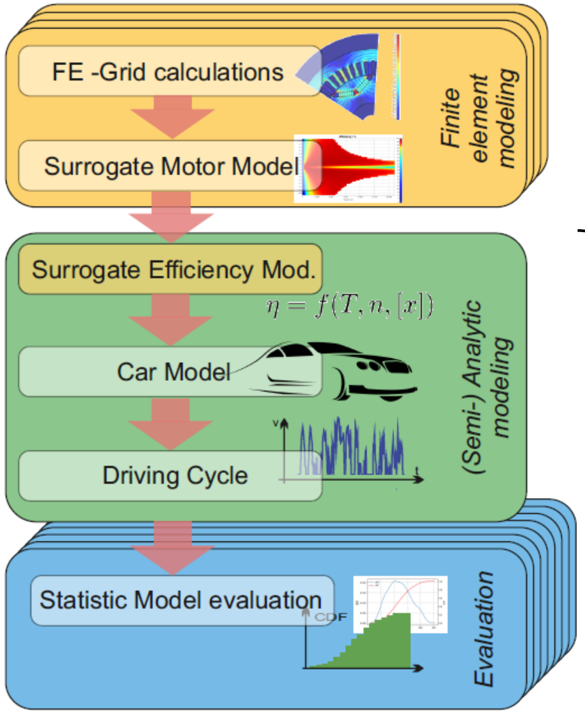
Question

What's the influence of various system parameters on the driving range of a BEV?

Varying Parameters

Group	Par.	[Unit]	Description
Motor Parameter	B_r	[T]	PM residual induction
Motor Parameter	d_{deg}	[mm]	width of impacted zone regarding the material degrad.
Motor Parameter	ϑ_{PM}	[°C]	permanent magnets temperature
Secondary Params	h	[m]	altitude above sea level
Secondary Params	Δ_m	[kg]	additional load
Secondary Params	η_{pe}	[-]	power electronics efficiency
Secondary Params	η_{pt}	[-]	powertrain efficiency

Modeling in SyMSpace



Scenario

AFAT	all factors at a time
G-I	technical parameters
G-II	environmental parameters

Sensitivity Analysis

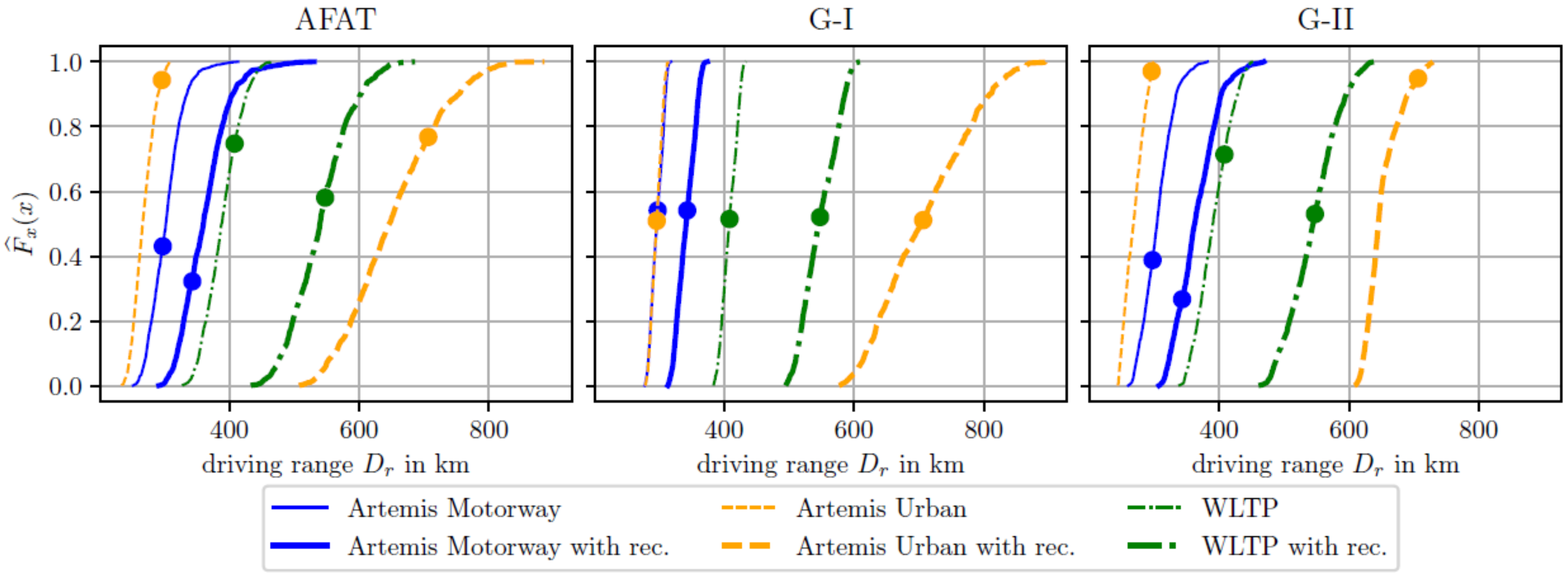
taken from [2]

[2] Bramerdorfer, G, Marth, E.; Computationally Efficient System-Level Evaluation of Battery Electric Vehicles. IEEE Workshop on Electrical Machines Design, Control and Diagnosis - WEMDCD 2021

Driving Range Sensitivity Analysis of a BEV

Based on Car and Motor Model of Tesla3

Results



Scenario

AFAT	all factors at a time
G-I	technical parameters
G-II	environmental parameters

taken from [2]

[2] Bramerdorfer, G, Marth, E.; Computationally Efficient System-Level Evaluation of Battery Electric Vehicles. IEEE Workshop on Electrical Machines Design, Control and Diagnosis - WEMDCD 2021

Robust optimization

Outlook

Robust Optimization

… incorporate sensitivity-objective in optimization

Even when using fast converging CDF evaluation, performing sensitivity analysis during e.g. genetic optimization (1000 additional samples per individual) takes too much time

Alternative methods, mainly using additional surrogate models, are under development

… meanwhile …

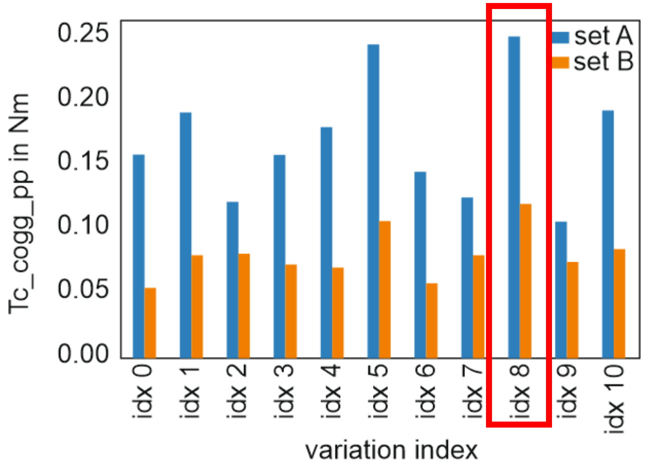
Robust Optimization

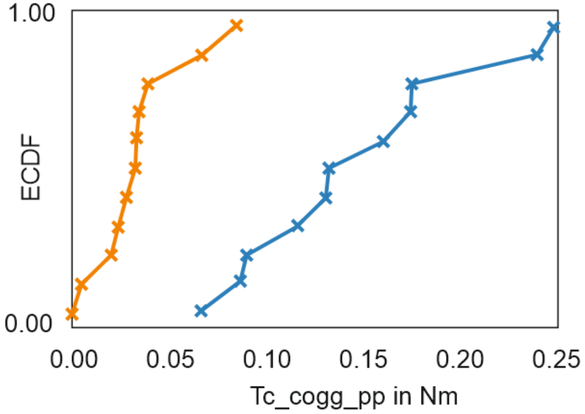
Simple Approach (Example: Magnetic Gear Optimization)

1. Evaluation of a set of tolerance-affected samples (set A) with randomly assigned tolerance values
2. Choose one of the bad samples and use this tolerance configuration for an optimization
3. Select Pareto optimal sample from optimization with “fixed” tolerance settings
4. Recalculate all sets from step 1. with selection of step 3. (set B)
5. compare CDFs

Approach might be used to at least take tolerance effects into account.

Of course, with this approach it is unlikely to find the most robust design!





Closing the presentation

Final summary

- Beta Version of Tolerance-Analysis-Workflow for SyMSpace available
- All segment geometries capable of tolerance analysis (with small changes to definition)
- Extensive documentation included for SymTolInterface python package
  - Ready to use within SyMSpace application (so far not working from WEB GUI)
  - Caution: proper evaluation of subharmonic signal content has to be checked!
- Projects welcome for further development and refinement of existing workflow

Thank you for your attention!

Science becomes
reality

SyM

JKU Institute for Electrical Drives and Power Electronics

LCM LINZ CENTER OF MECHATRONICS