A practical approach to reducing development risk and time through Model-driven Development

Introduction: Who am I

- Jasper Keuning; Lead Electronics Engineer @ 3T since 2012
- Modeling and Simulation
  - Mathworks tooling (Matlab, Simulink, Simscape)
  - Signal & Power Integrity (Ansys)
  - Thermal Analysis (Ansys)
- High speed digital boards
- Medical devices
- Standards and Regulations

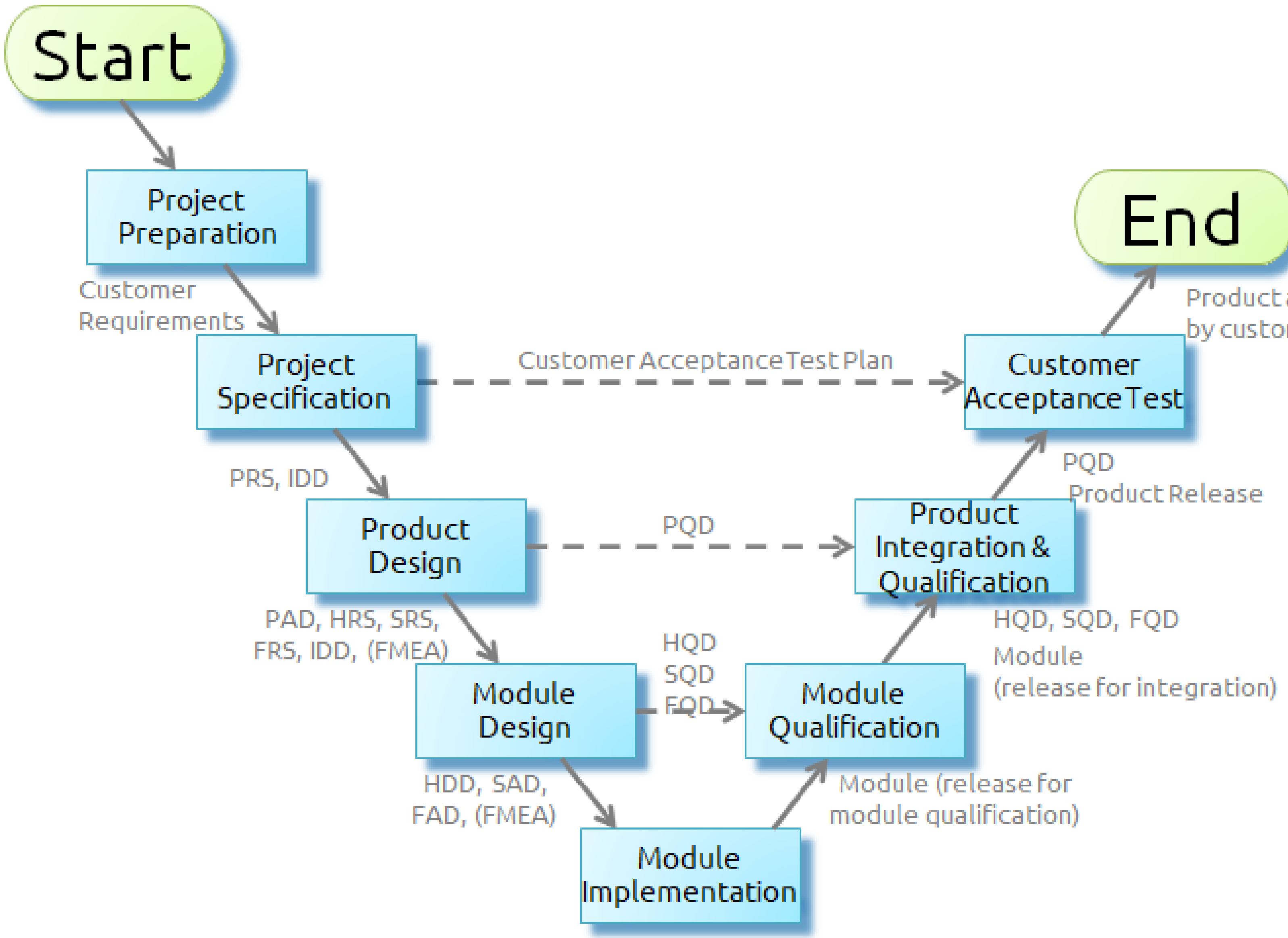
Agenda

- What is Model-driven Development (MDD)
- MDD in the design process
- Project: Skytron Freedom

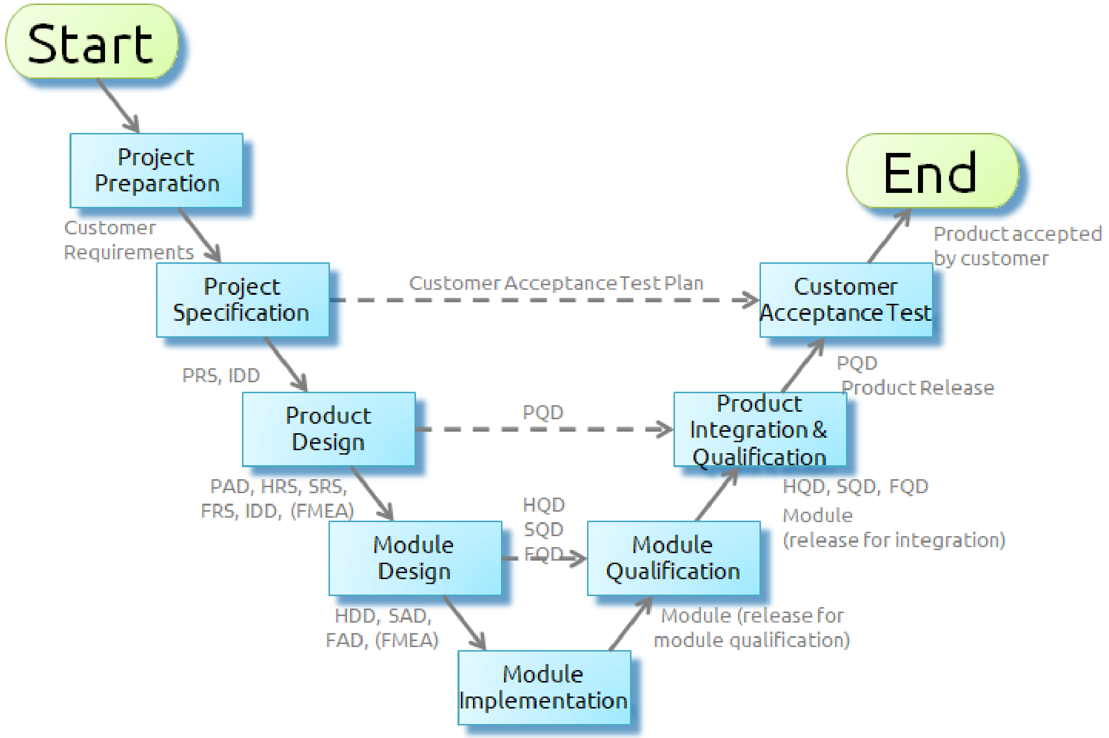
What is Model-driven development (MDD)

- Using models and simulation to:
  - Create
  - Evaluate
  - Optimize
- It is not a goal, just a means to enhance the design process
- Applicable for many phases of the development cycle

MDD in the design process



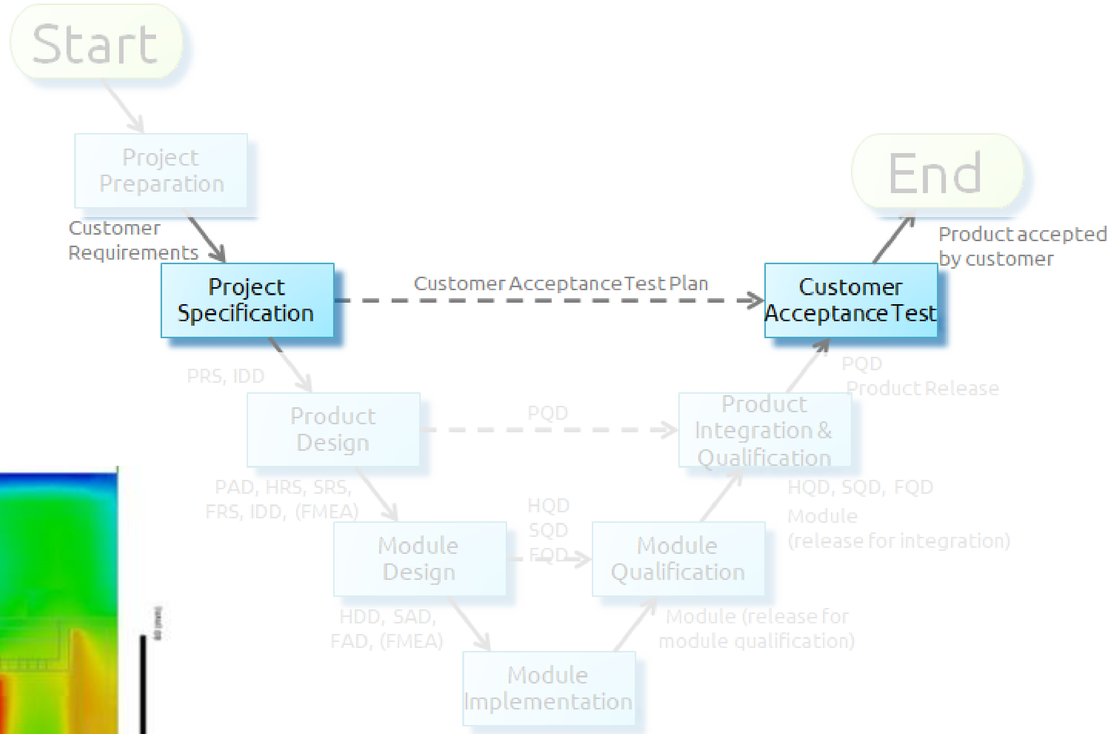
MDD in the design process

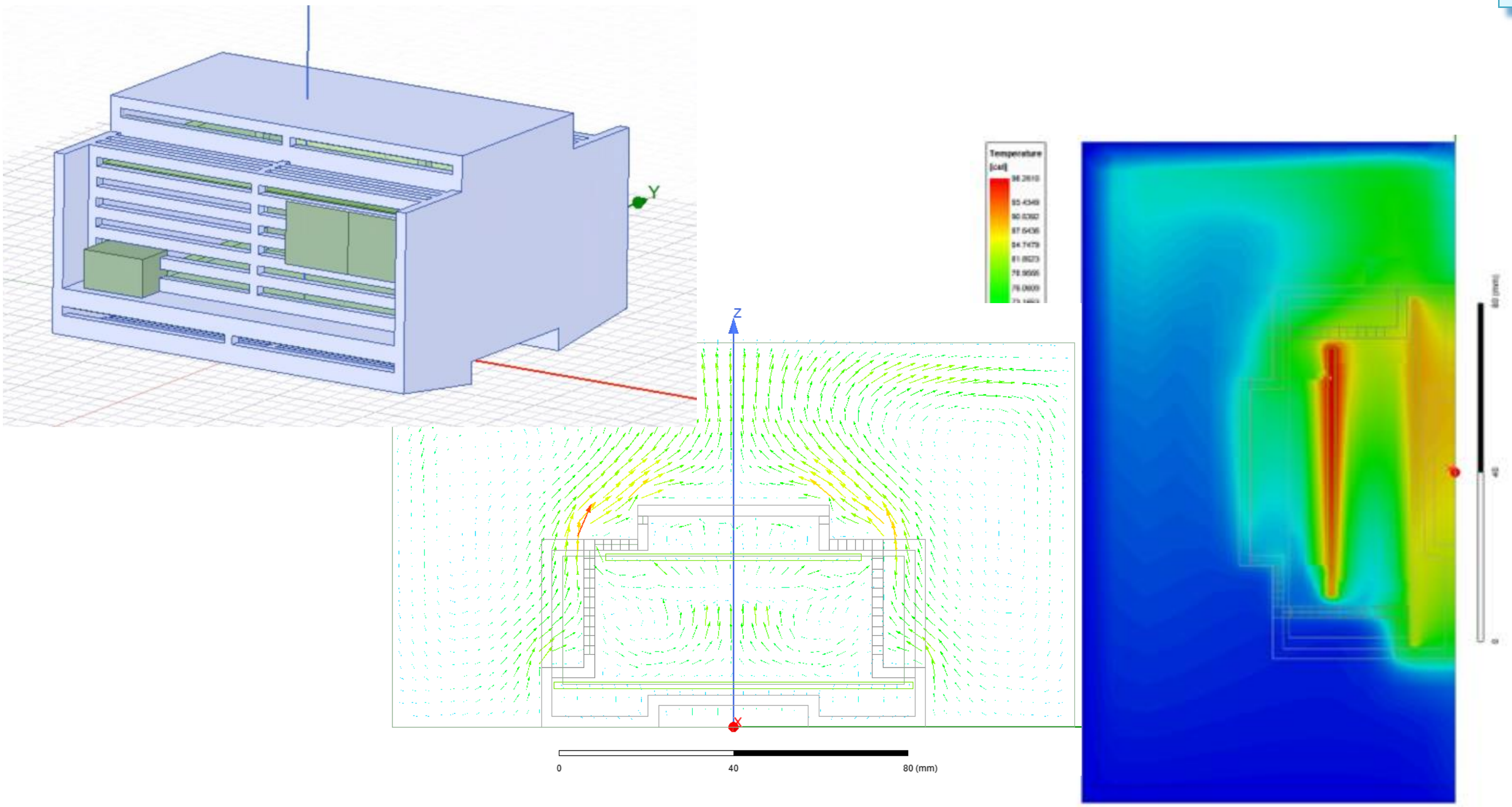


D&E EVENT
Het ontwerpen van innovatieve elektronica
Woensdag 19 april 2023
1931 Congrescentrum 's-Hertogenbosch

MDD in the design process

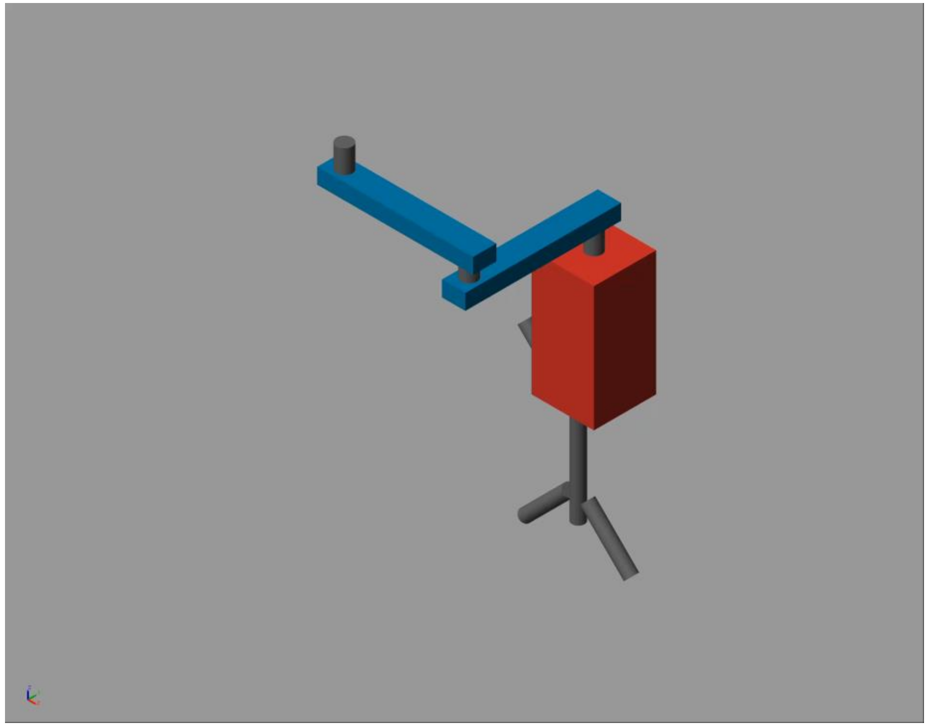
- Specification Phase:
  - Feasibility





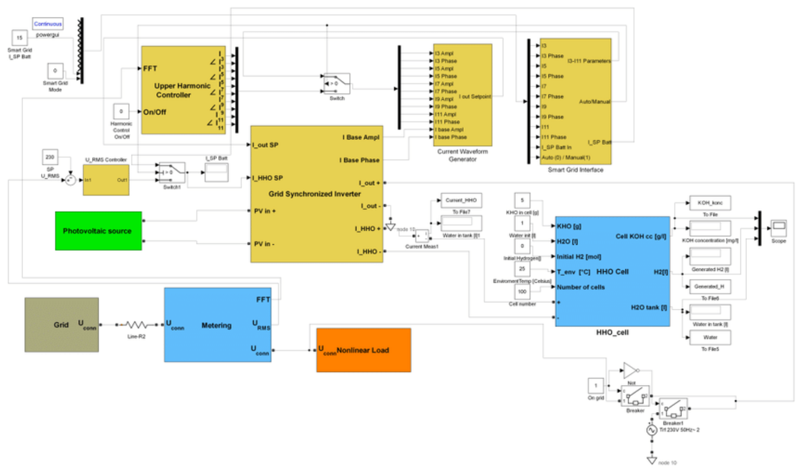
MDD in the design process

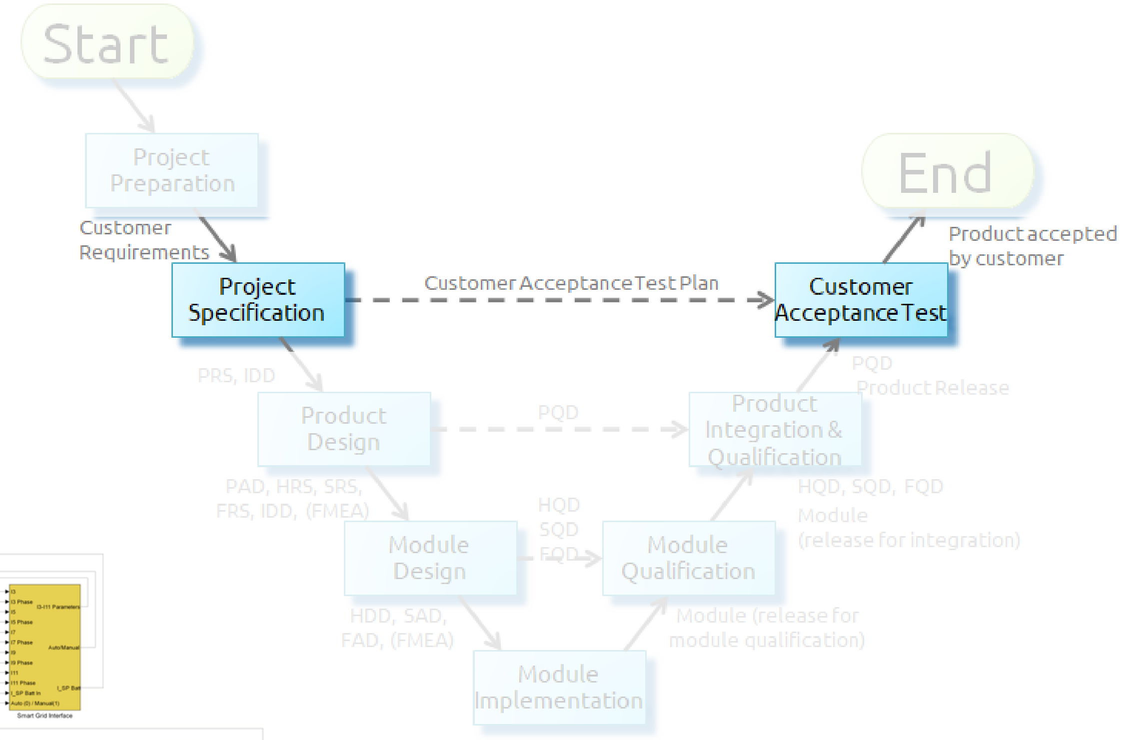
- Specification Phase:
  - Feasibility
  - Identification of critical behavior



MDD in the design process

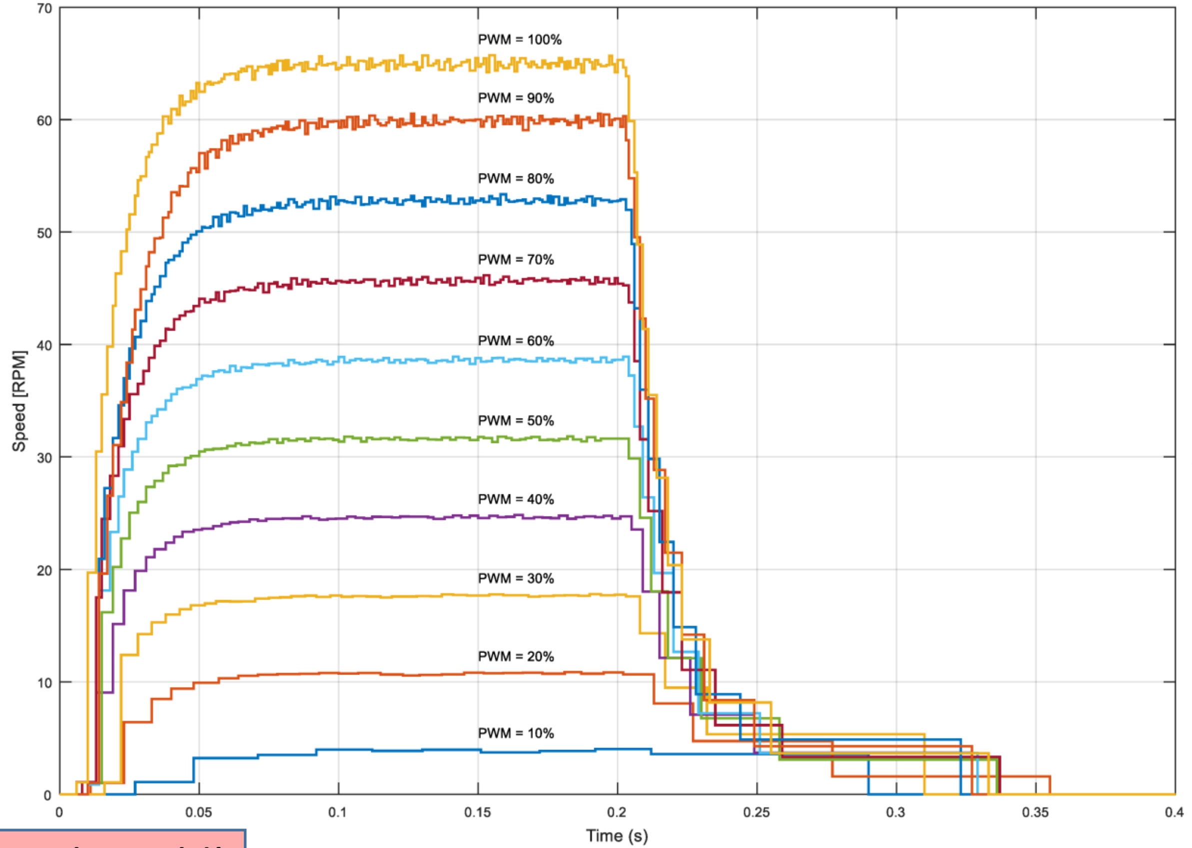
- Specification Phase:
  - Feasibility
  - Identification of critical behavior
  - Model as Requirement

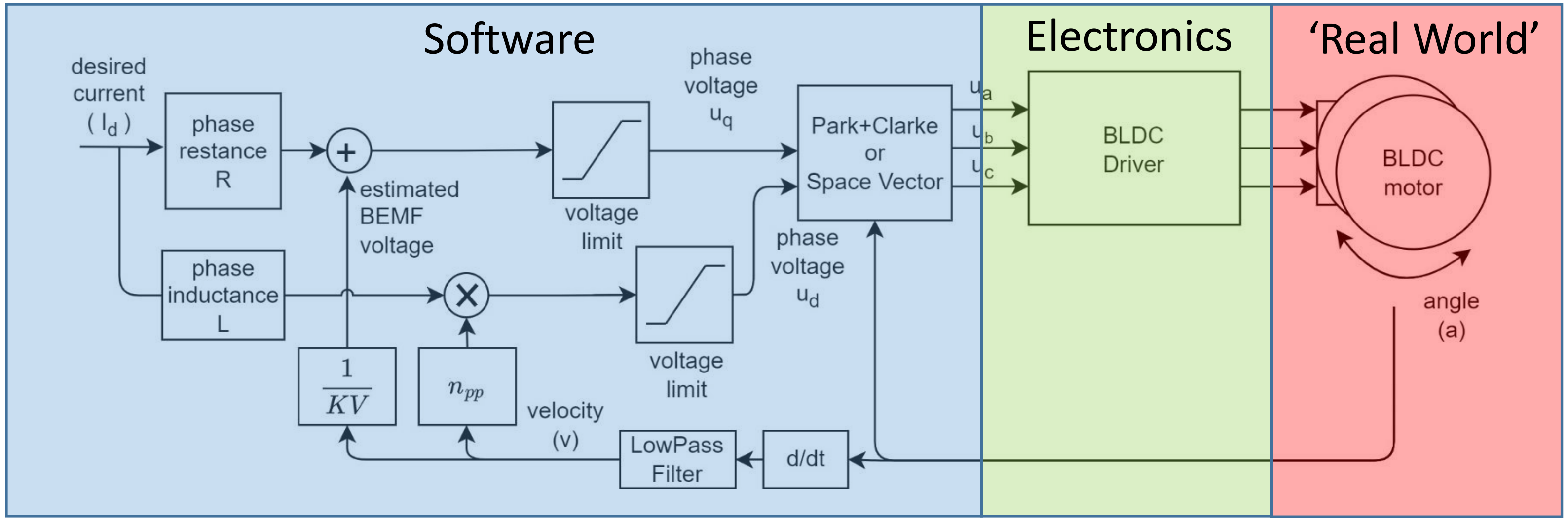




MDD in the design process

- Product level design:
  - Multidomain modeling and simulation
  - System level evaluation/optimization

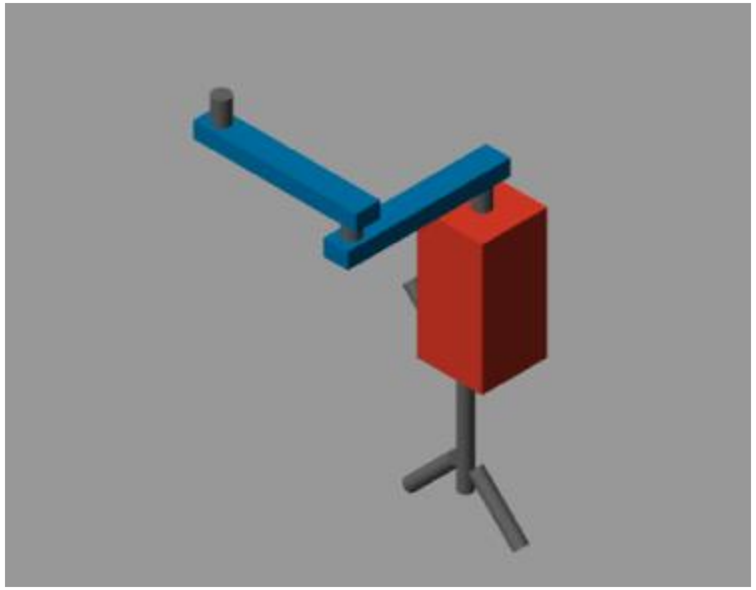


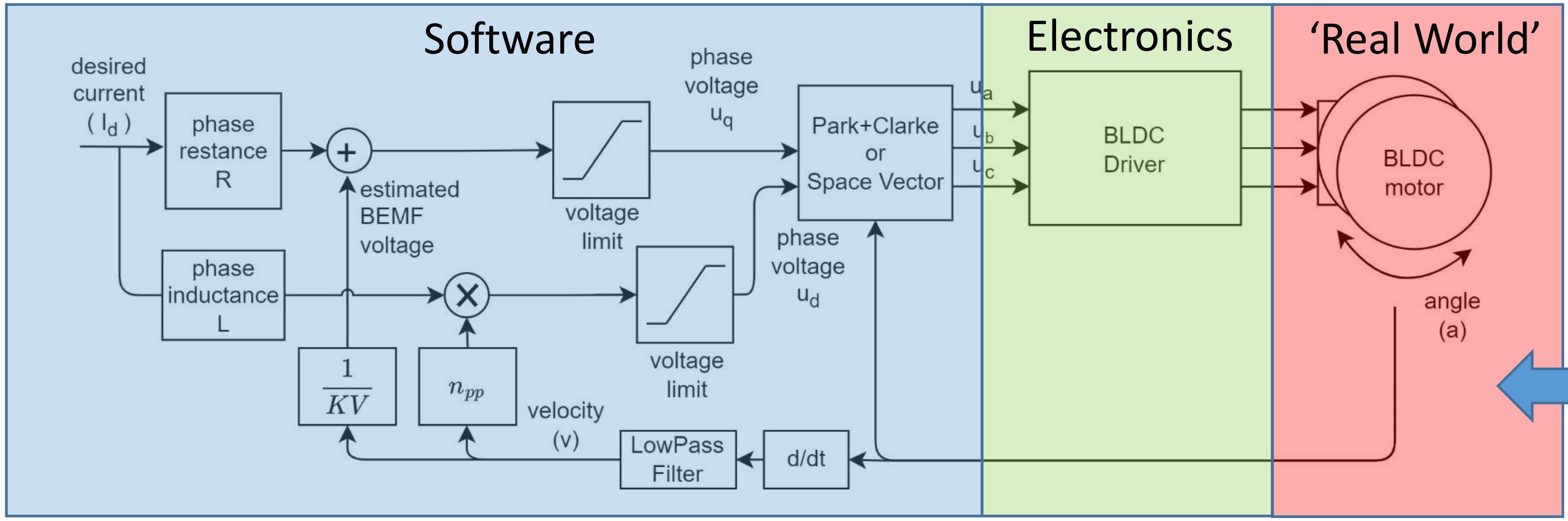


FAD, (FMEA)
module qualification)
Module Implementation

MDD in the design process

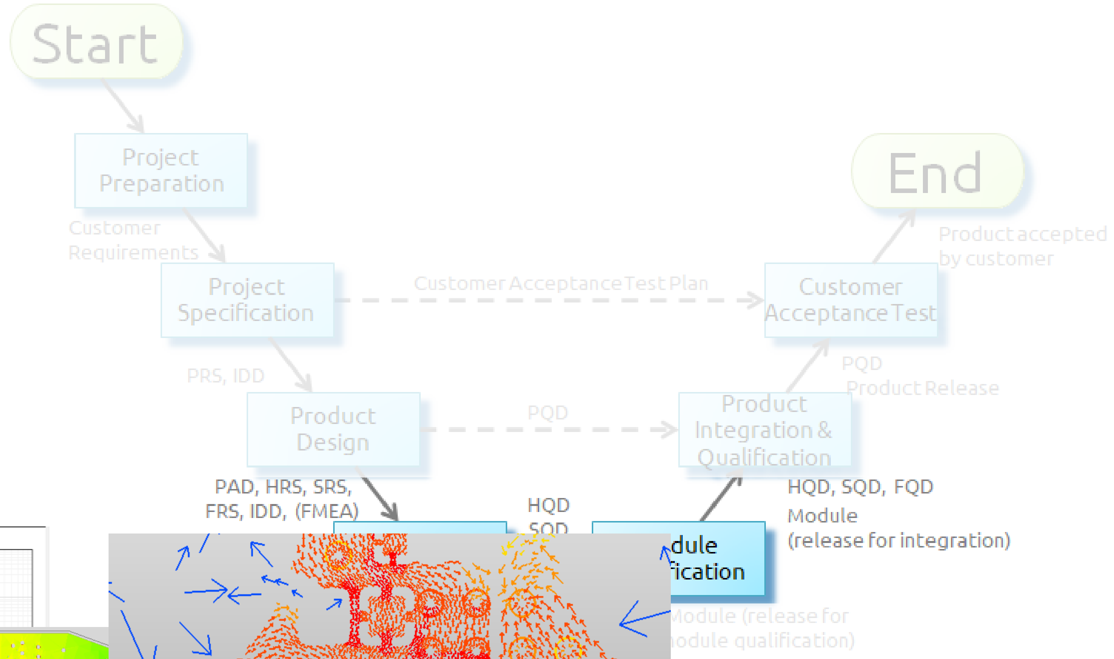
- Product level design:
  - Multidomain modeling and simulation
  - System level evaluation/optimization
  - System level verification and integration
  - Multidisciplinary design language

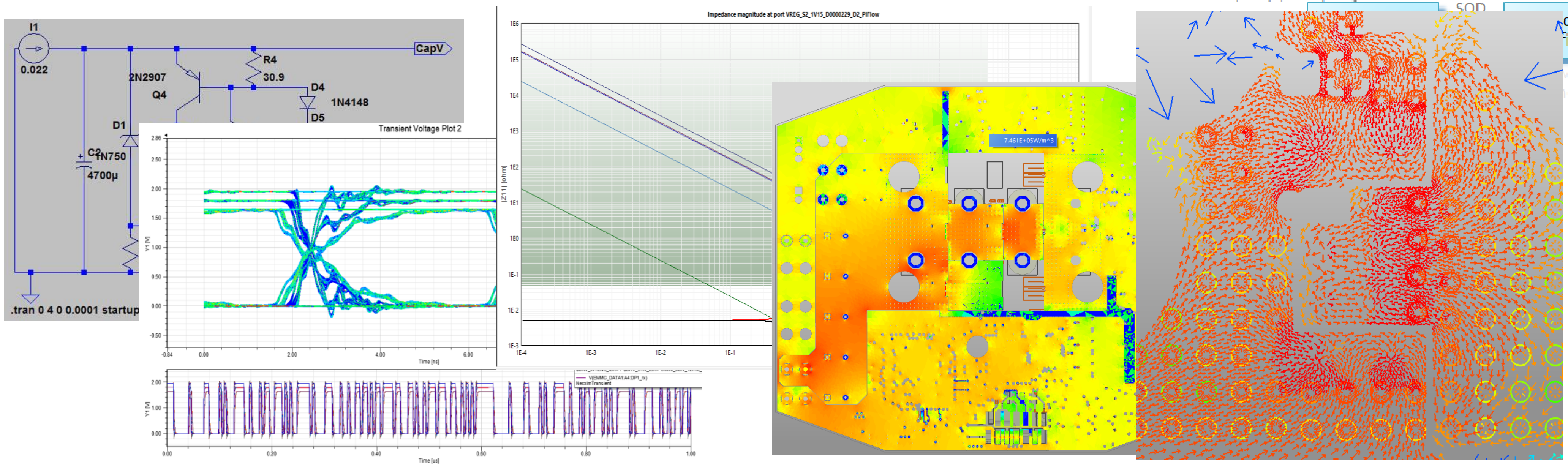




MDD in the design process

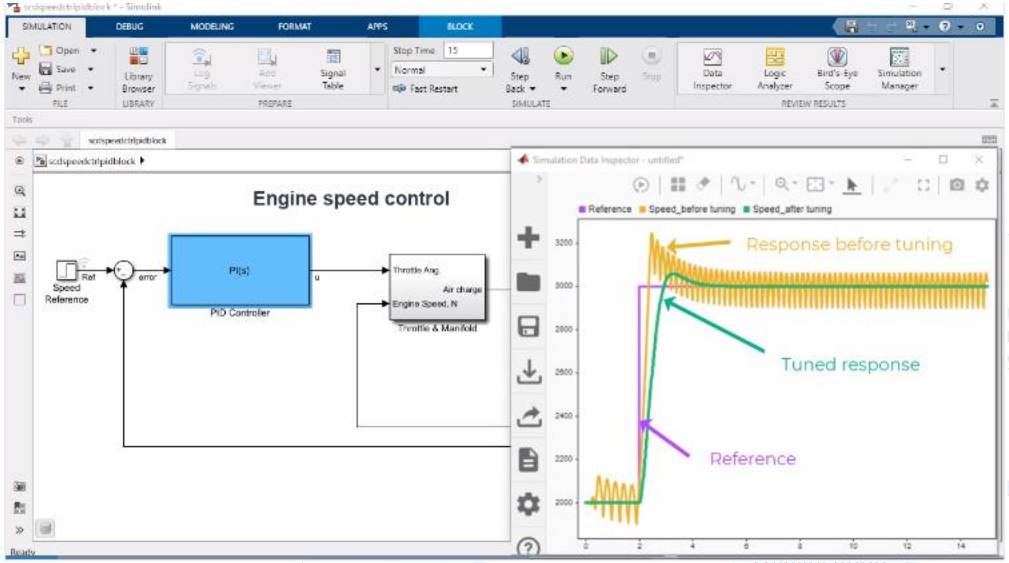
- Module design Electronics:
  - SPICE
  - SI & PI
  - Thermal
  - …

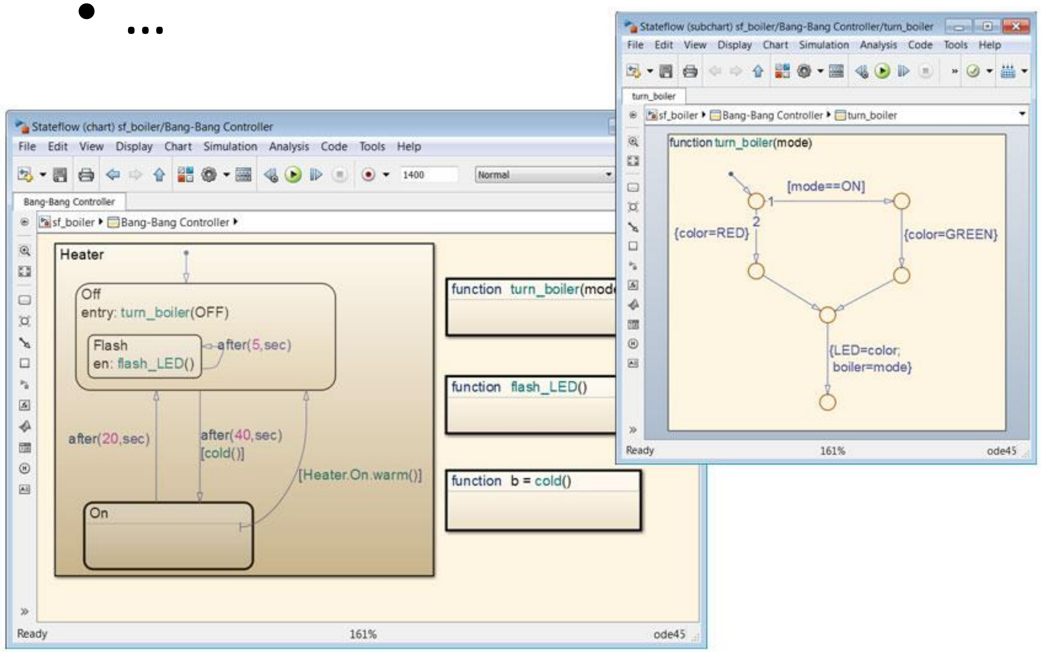


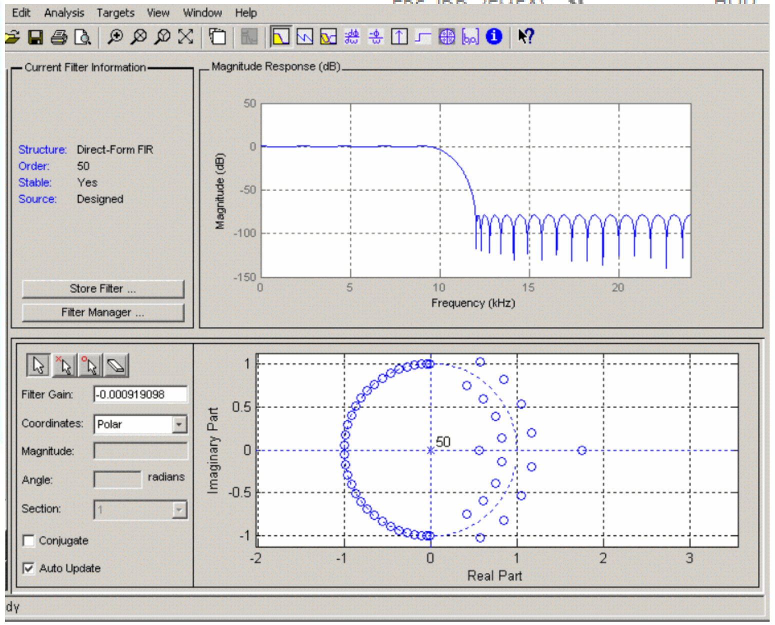


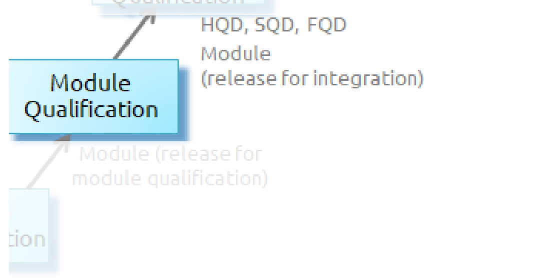
MDD in the design process

- Module design Software:
  - State diagrams
  - Filters
  - Control Algorithms
  - …



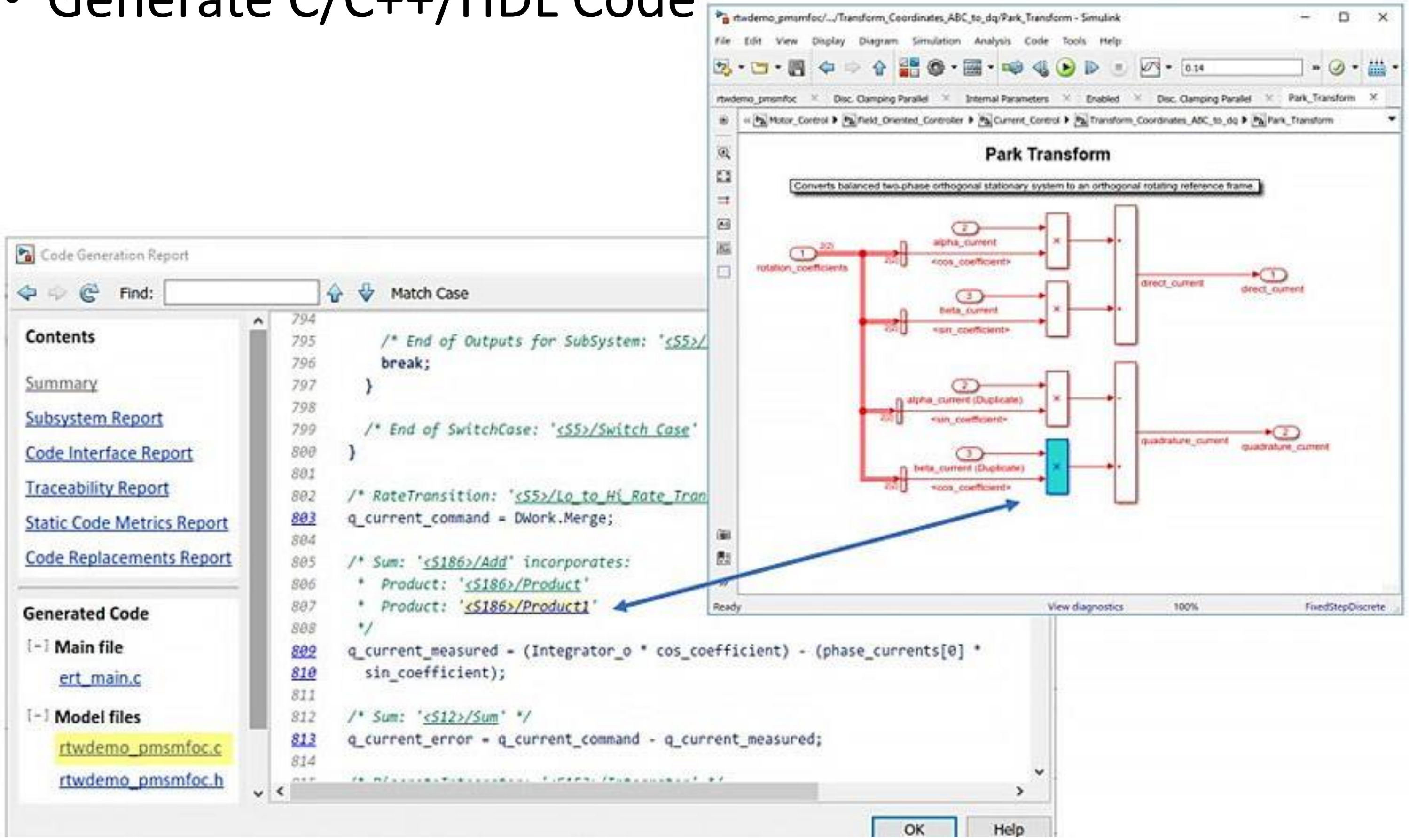


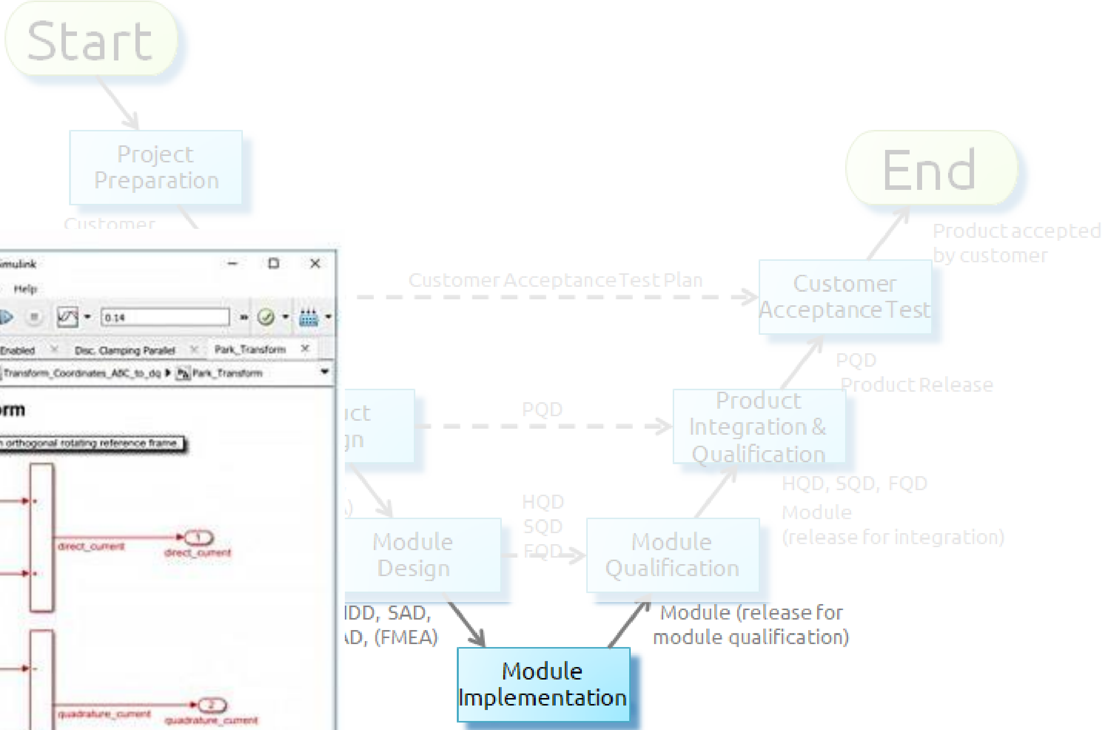




MDD in the design process

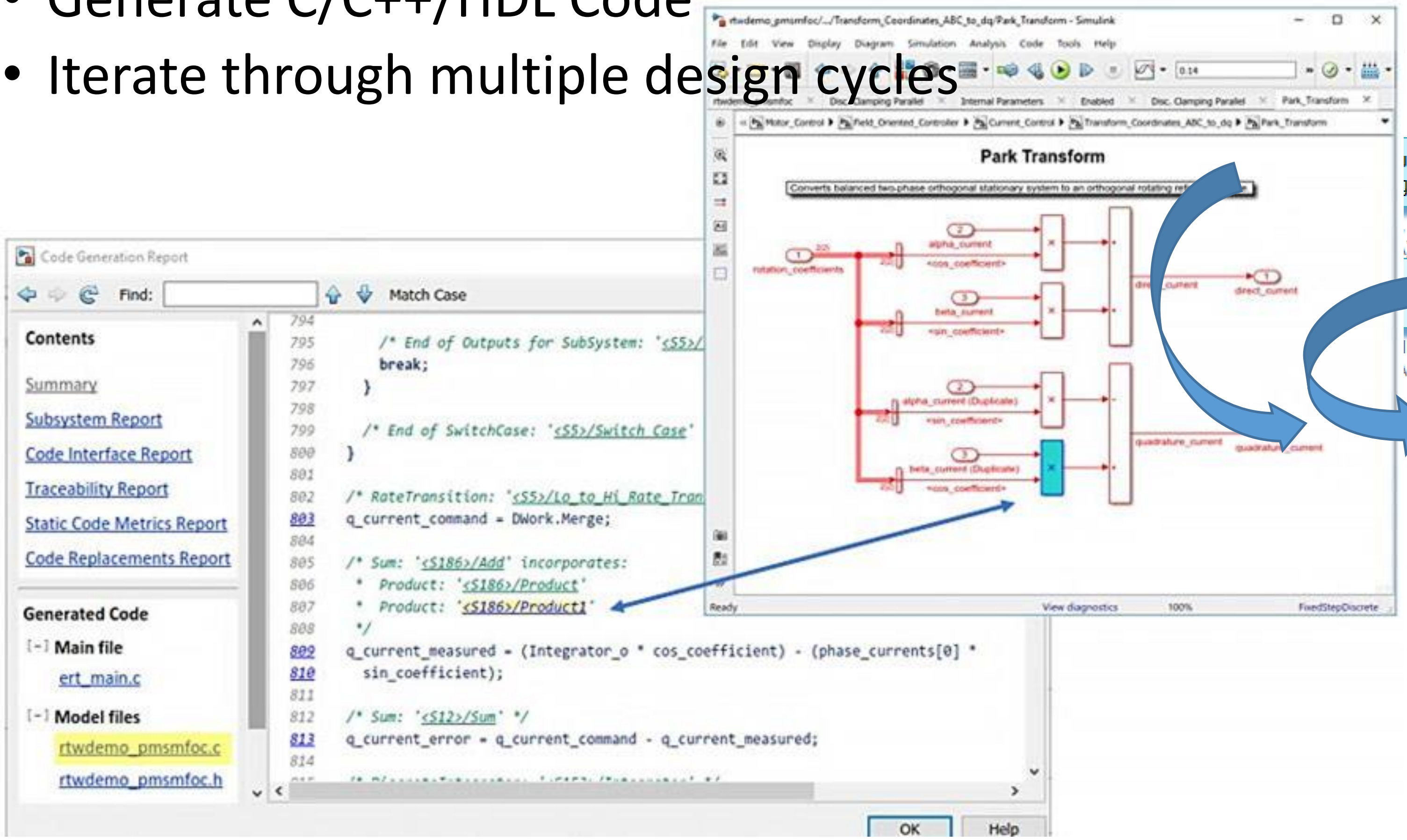
- (Software) Implementation
  - Generate C/C++/HDL Code

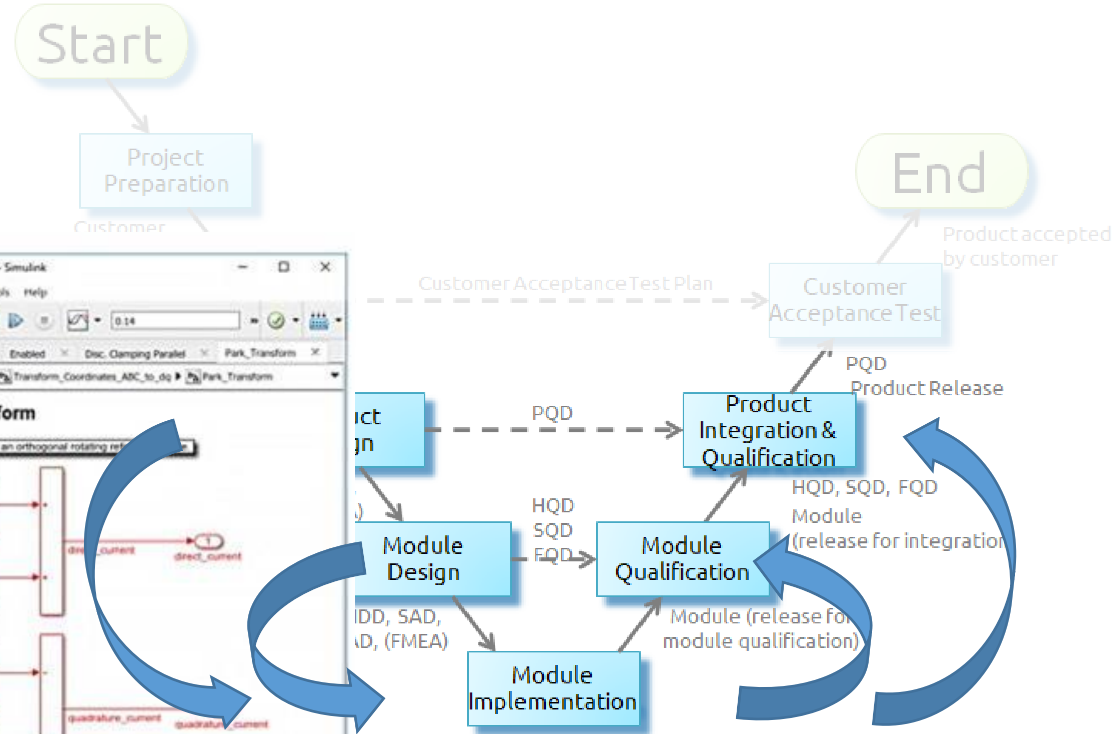




MDD in the design process

- (Software) Implementation
  - Generate C/C++/HDL Code
  - Iterate through multiple design cycles





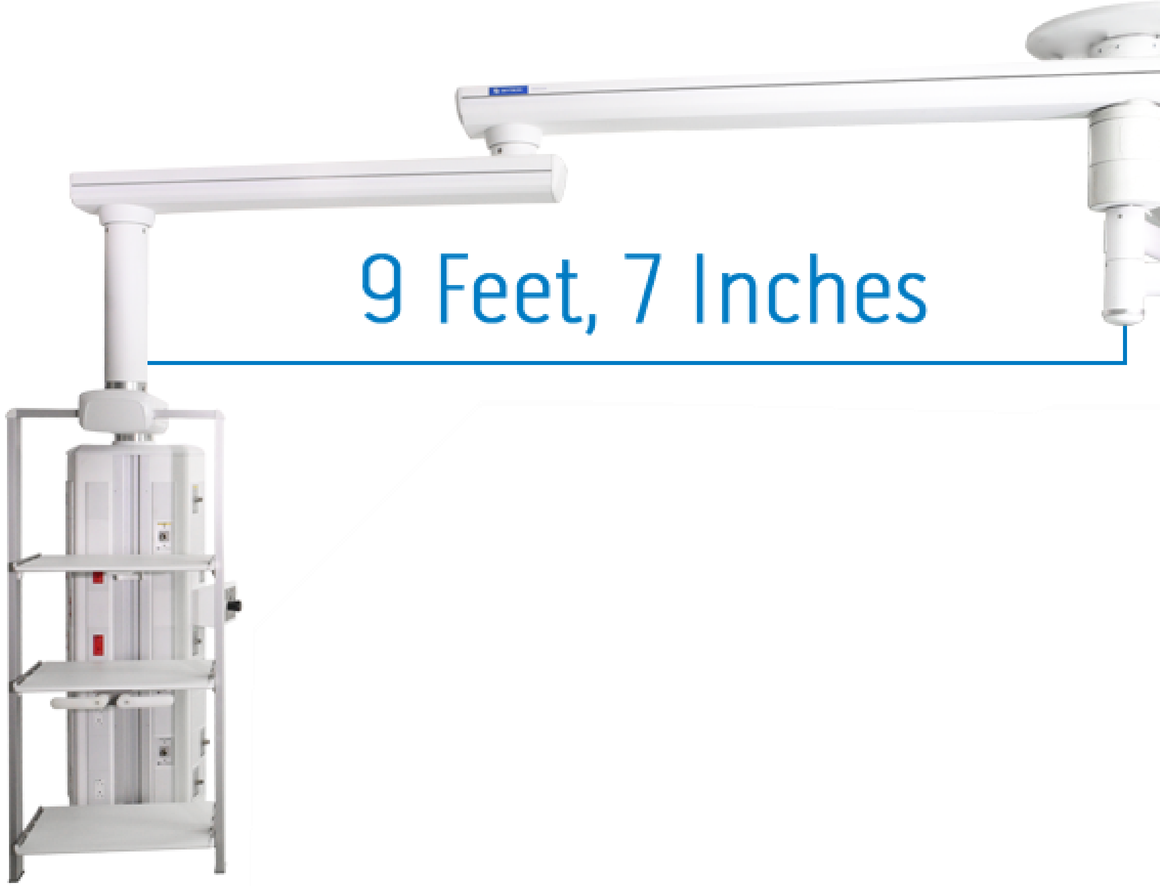
Project: Skytron Freedom

Project: Skytron Freedom

The challenge:

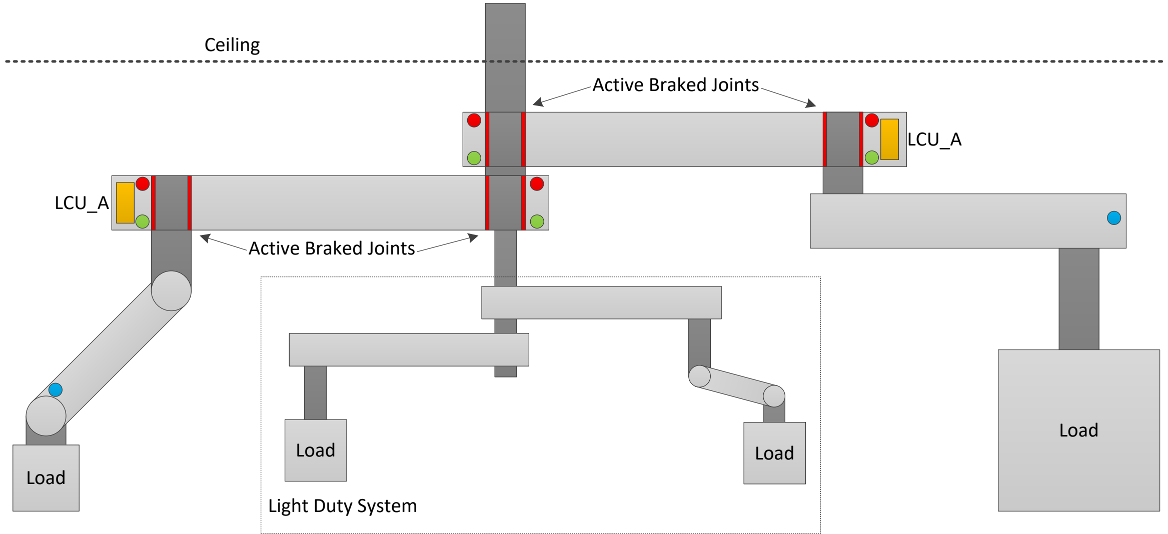
- Easy positioning and movement of arm
  - Drift due to mechanical bending
  - Interacting with devices on arm cause movement
  - Accidental Impacts may have large consequences
- Intentional friction in joints
  - Takes quite some effort to move arm around
  - Wear on friction mechanism

Project: Skytron Freedom



The Solution:

- Smart Braking System
  - Easy to move when needed
  - Stable when stationary
  - No special user interaction needed
  - Sensor system to detect user intention



Project: Skytron Freedom

Responsibilities:

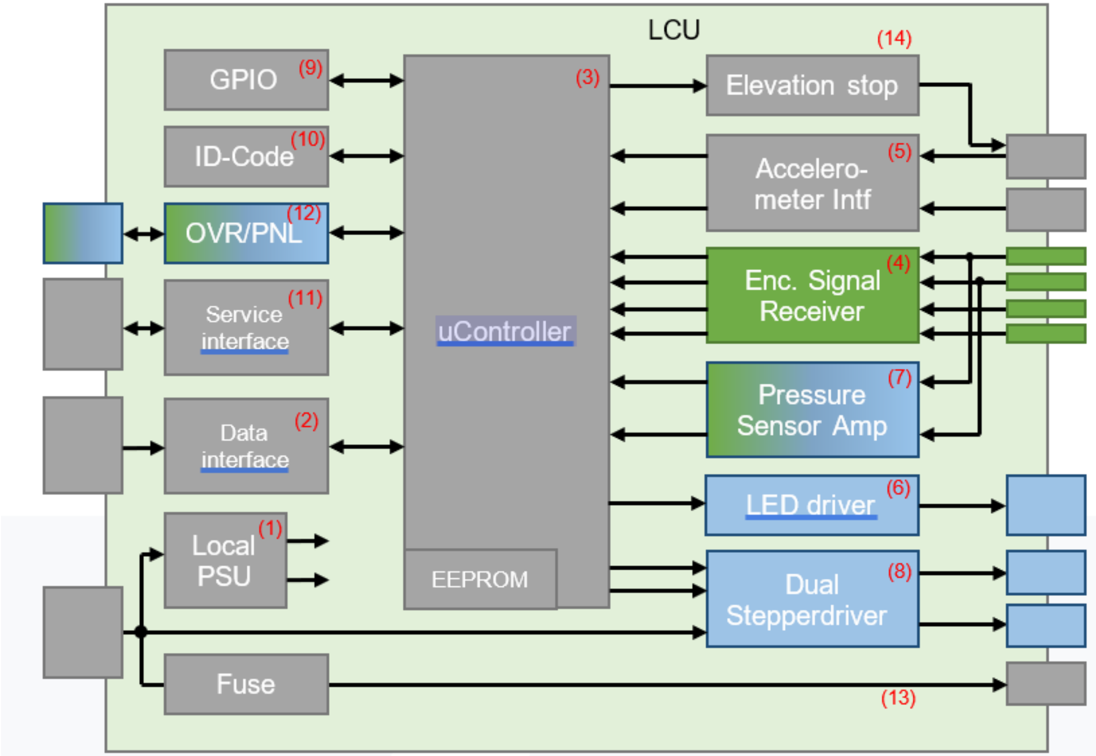
- Customer will be responsible for:
  - mechanical design of system
  - user experience
- 3T to develop Electronics and embedded Software

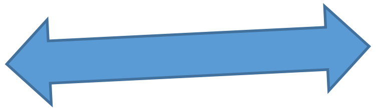
Goal of project: An excellent user experience

Project: Skytron Freedom

Specification:

- Electronics

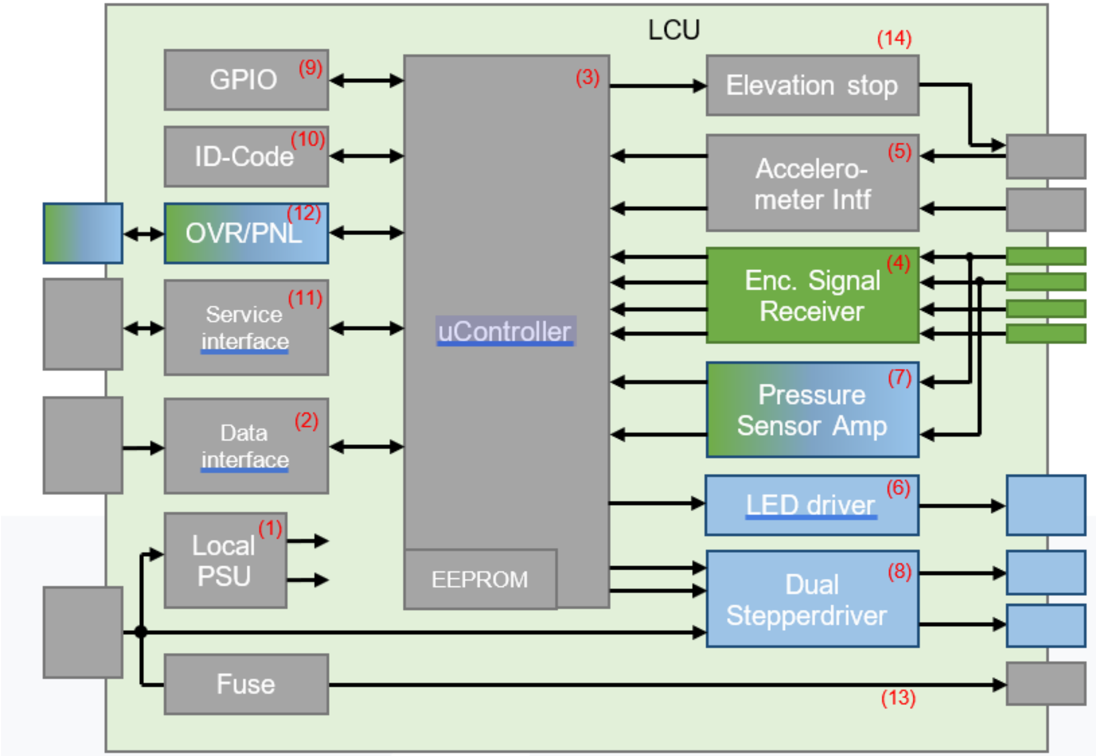




Project: Skytron Freedom

Specification:

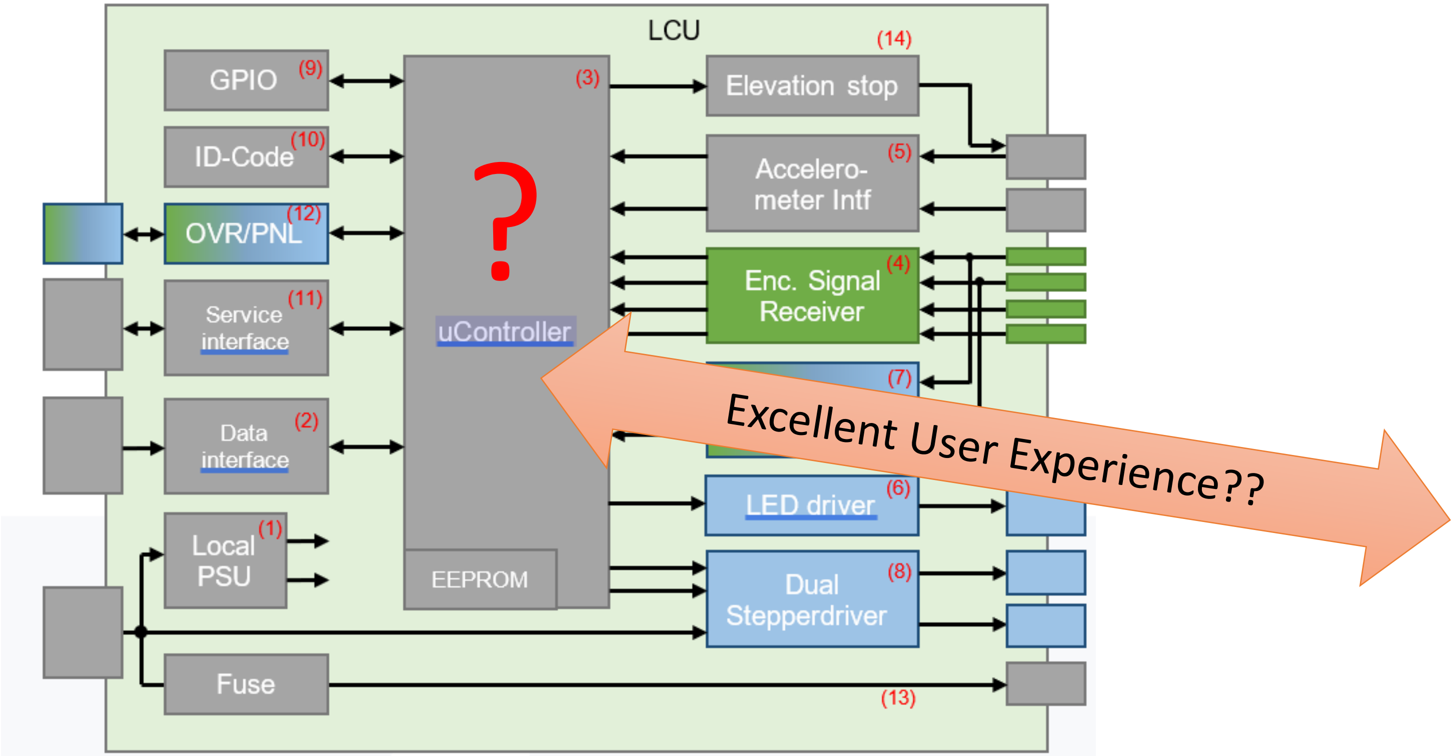
- Electronics



Project: Skytron Freedom

Specification:

- Software specification



Project: Skytron Freedom

Software specifications (?):

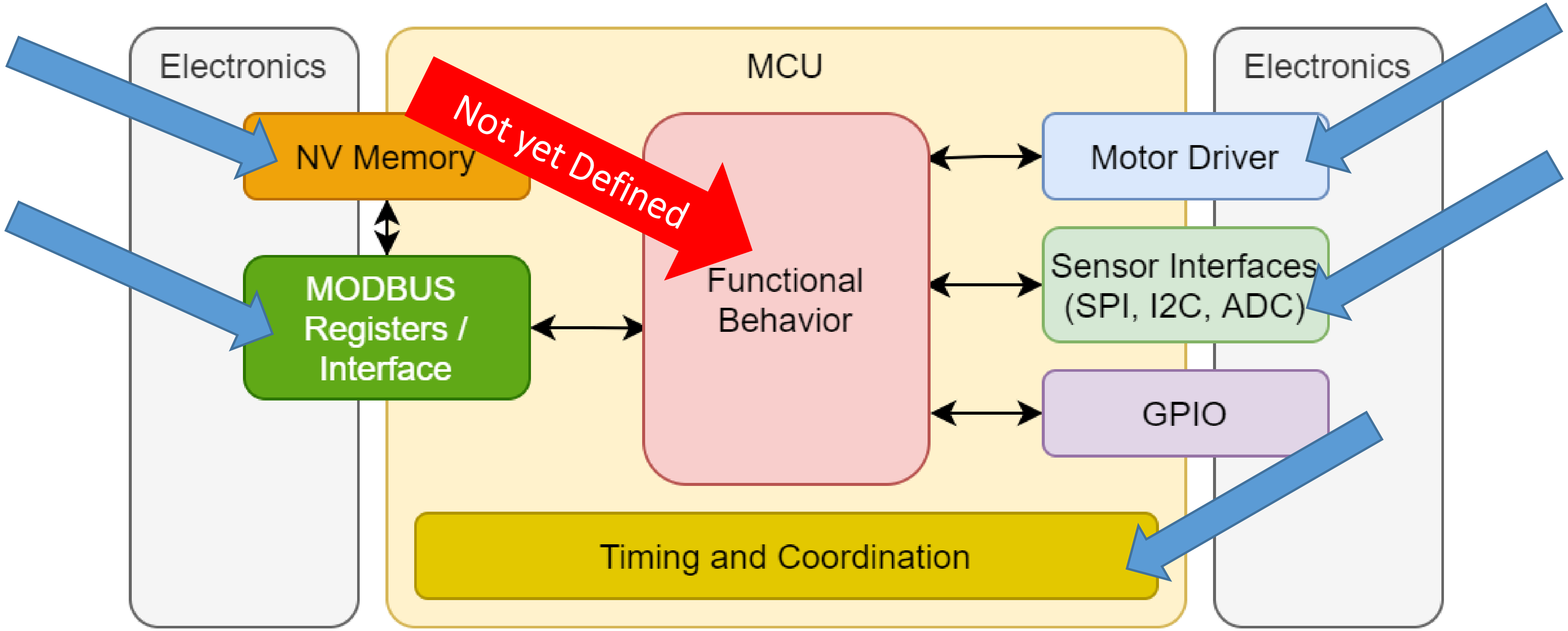
- Ensure all hardware interfaces are accessible and controllable
- The desired user experience of the end-product
- Mechanical design of brakes and arm not available at project start

In other words, a lot still undetermined

Project: Skytron Freedom

Development Strategy: Reduce development time and risk

- Use of existing design experience for electronics and software



Project: Skytron Freedom

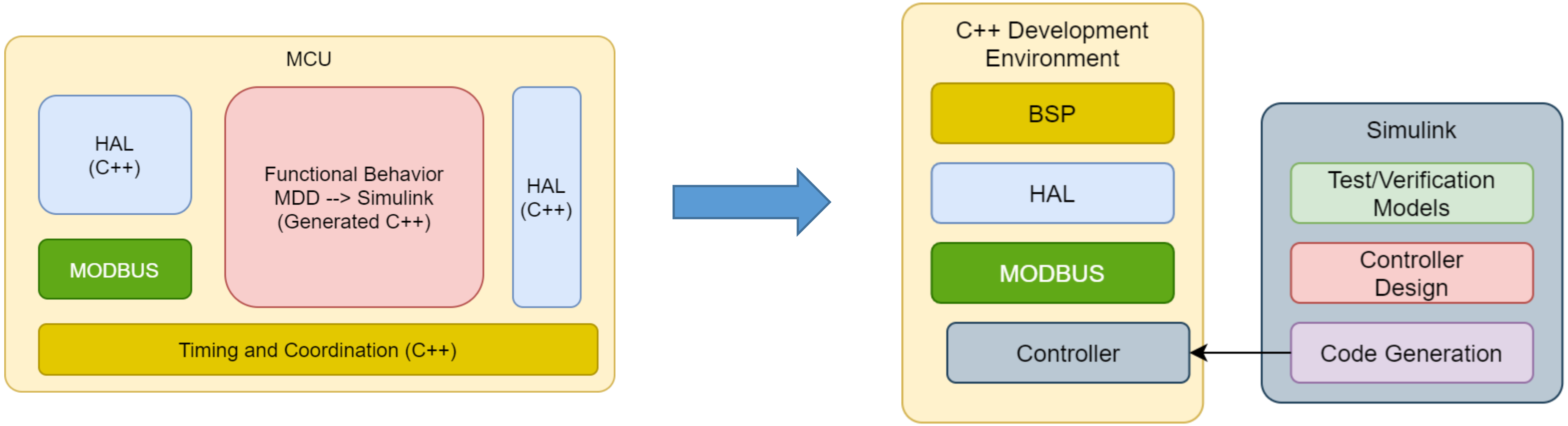
Development Strategy: Reduce development time and risk

- Use MDD for Functional Behavior design and implementation
    - Common multidisciplinary language for communication
    - Evaluate against model and allow rapid iteration cycles
    - Ability to look deep inside model to gain insight and solve issues
    - Design engineer does not need to be a software engineer

Project: Skytron Freedom

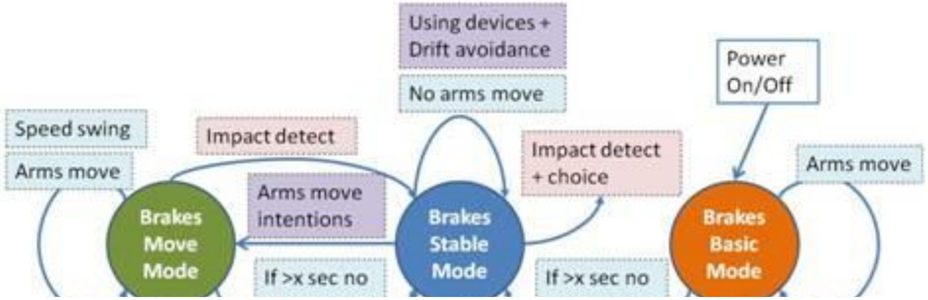
Development Strategy: Reduce development time and risk

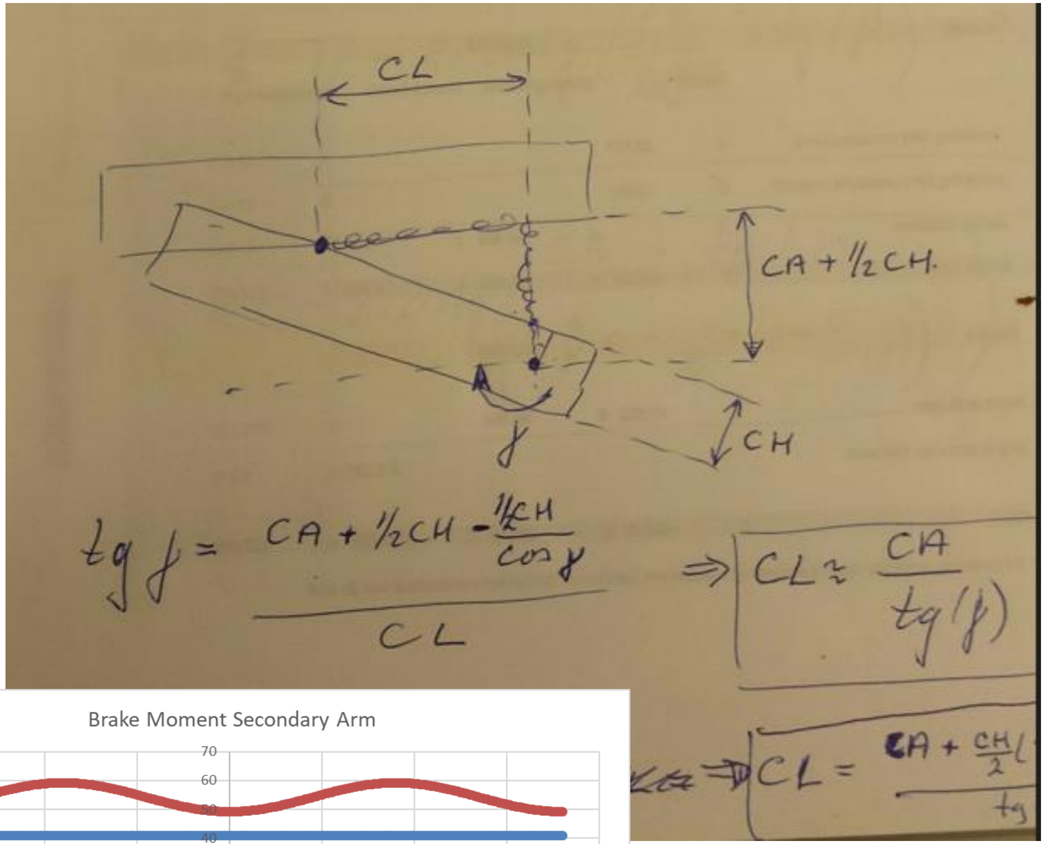
- Functional Behavior will be generated C++ function to be included in C++ project

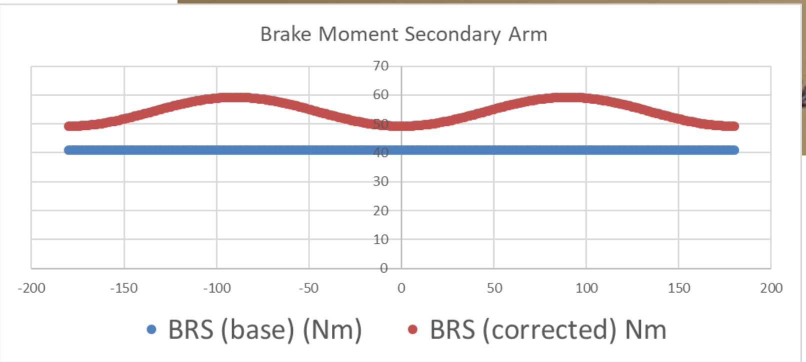


Project: Skytron Freedom

Customer's Language







Aanpassen Brake Moment:

BRP-mimp: Brake = ALS(IMDM=true; *BRP-brake-max; Brake (last)*)

BRP-stab: Brake = ALS(STAB=true; BRP-stab; *Brake (last)*)

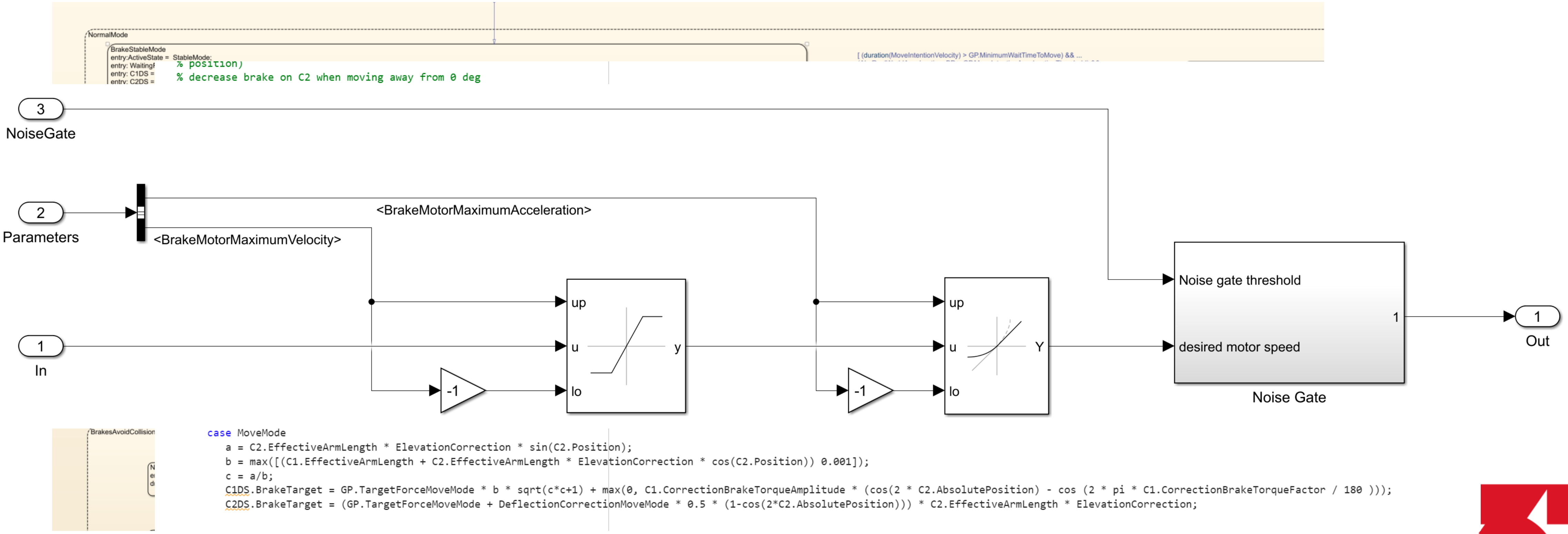
Aanpassen Brake Moment:

BRS-mimp: Brake = ALS(IMDM=true; *BRS-brake-max; Brake (last)*)

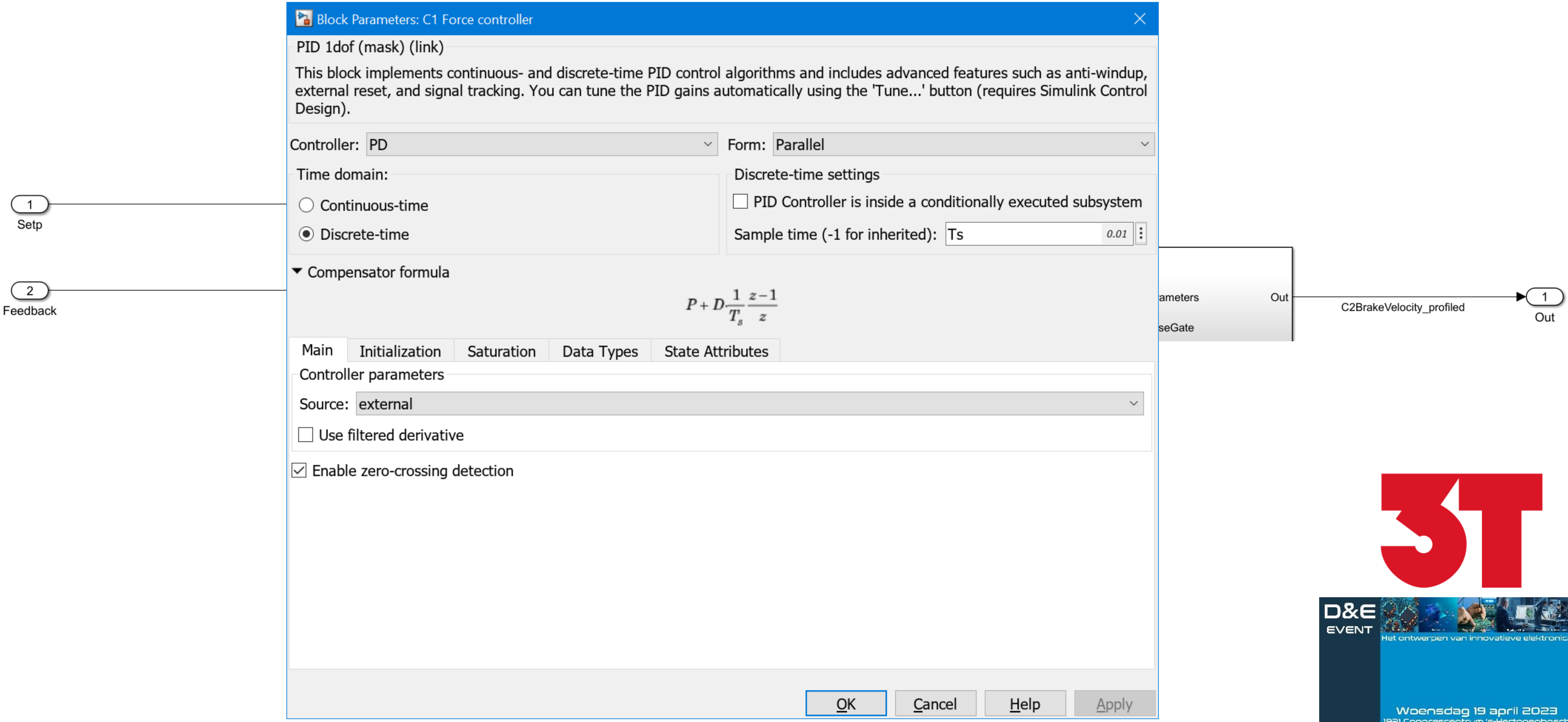
BRS-stab: Brake = ALS(STAB=true; BRS-stab; *Brake (last)*)

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
1	(mg)	(mg)	(mg)	(rad)	(rad)	(rad)	(rad/msec)	(rad/msec)	(rad/msec)		MOVE	MOVE	ROTATE	ROTATE		STABLE	D
2	Ax	Ay	Az	Ha	Hb	Hc	Ha	Hb	Hc		intent	active	pendant	carrier		pendant	H
47	3.267	-1.930	0.000	-1.300	-1.981	-1.539	3.728	22.371	22.371							JA	
48	3.118	-1.782	0.000	-1.300	-1.981	-1.545	3.728	26.099	22.371							JA	
49	2.970	-1.633	-0.015	-1.300	-1.969	-1.545	0.000	33.556	26.099				JA				
50	3.118	-1.782	0.000	-1.300	-1.969	-1.551	0.000	37.284	29.827				JA				

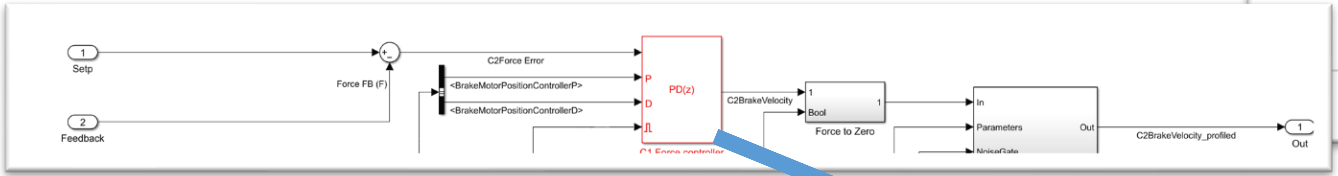
Project: Skytron Freedom

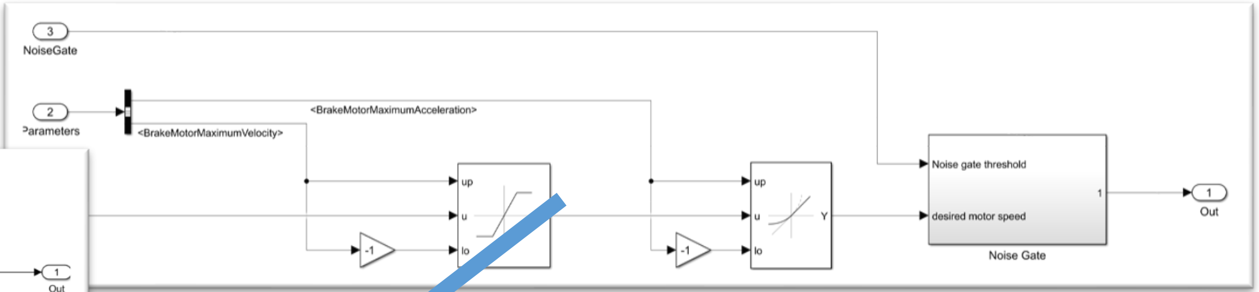


Project: Skytron Freedom



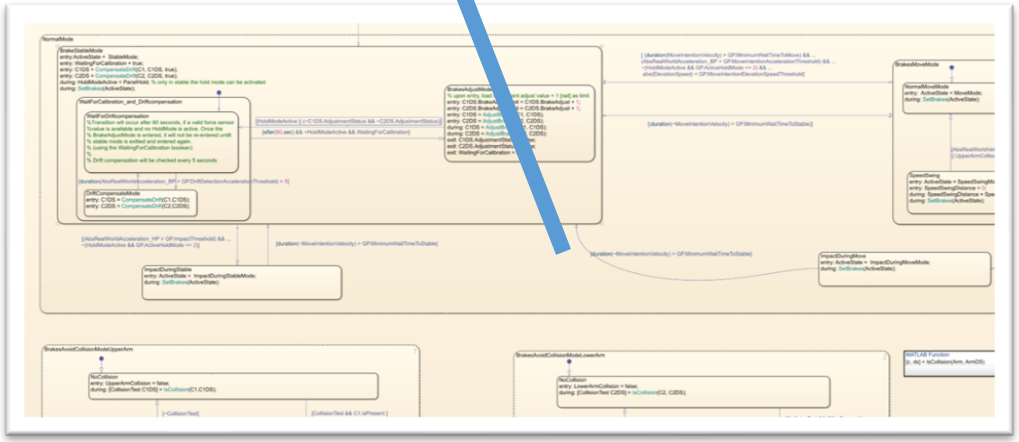
Project: Skytron Freedom





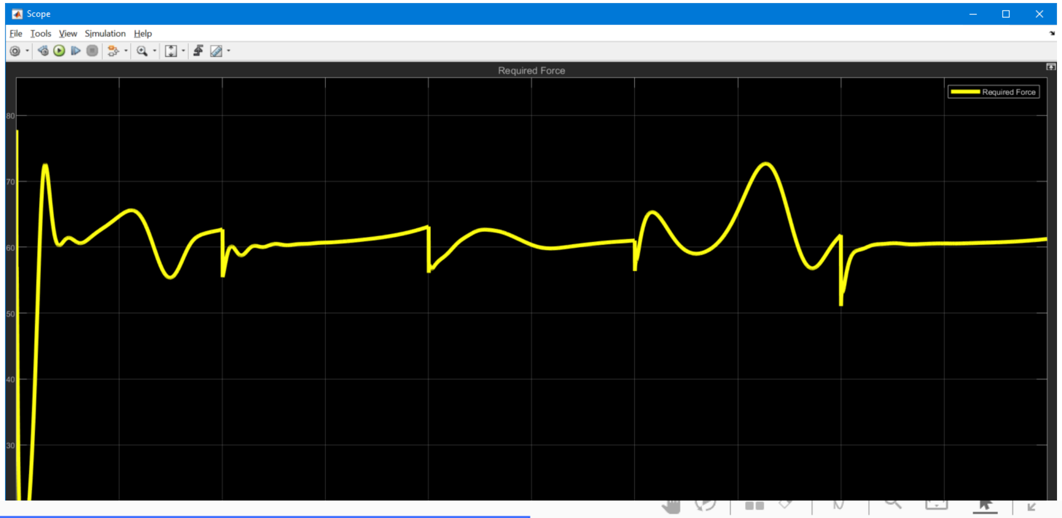
The Design of the Functional Behavior

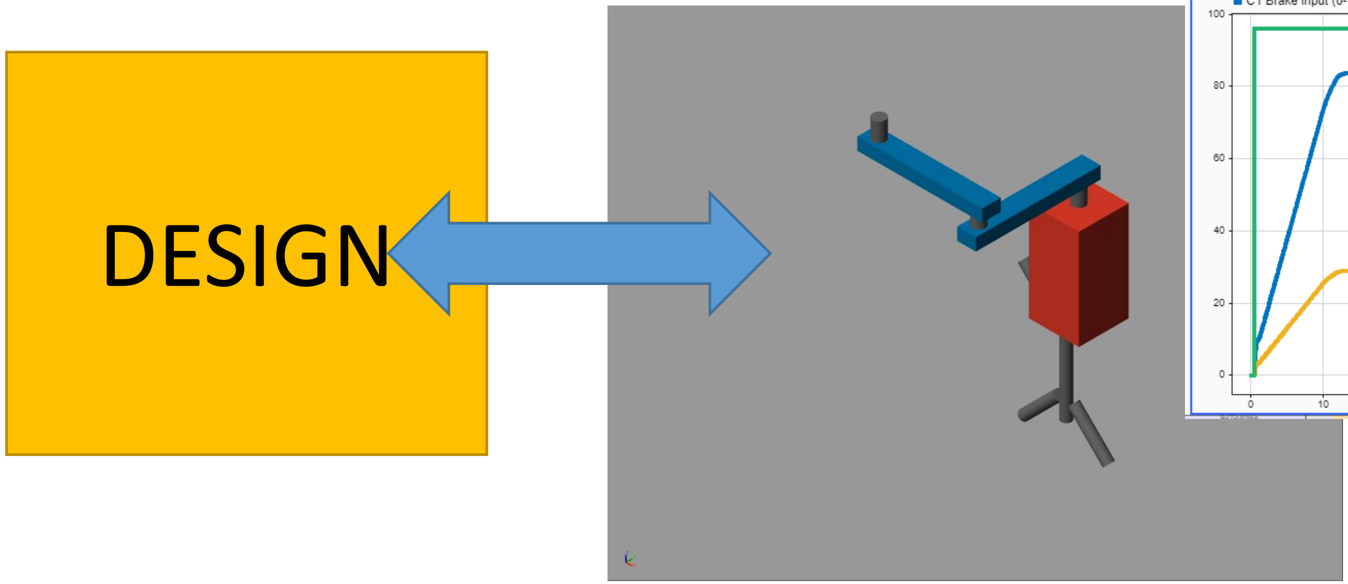
```
% decrease brake on C2 when moving away from 0 deg
if C2.Velocity * C2.AbsolutePosition < 0
    DeflectionCorrectionMoveMode = 135/250 * LoadParams.Weight;
else
    DeflectionCorrectionMoveMode = -1 * 135/250 * LoadParams.Weight * GP.TargetForceMoveMode / 200;
end


switch(inp)
    case {InitMode, InitBrakesMode, BasicMode}
        a = C2.EffectiveArmLength * ElevationCorrection;
        b = C1.EffectiveArmLength;
        c = a/b;
        C1DS.BrakeTarget = 1.2 * GP.TargetForceStableMode * C1DS.TargetForceCompensation * b * sqrt(c*c+1);
        C2DS.BrakeTarget = 1.2 * GP.TargetForceStableMode * C2DS.TargetForceCompensation * C2.EffectiveArmLength * ElevationCorrection ;

    case StableMode
        a = C2.EffectiveArmLength * ElevationCorrection * sin(C2.Position);
        b = max([(C1.EffectiveArmLength + C2.EffectiveArmLength * ElevationCorrection * cos(C2.Position)) 0.001]);
        c = a/b;
        C1DS.BrakeTarget = GP.TargetForceStableMode * C1DS.TargetForceCompensation * b * sqrt(c*c+1) ;

     case MoveMode
        a = C2.EffectiveArmLength * ElevationCorrection * sin(C2.Position);
        b = max([(C1.EffectiveArmLength + C2.EffectiveArmLength * ElevationCorrection * cos(C2.Position)) 0.001]);
        c = a/b;
        C1DS.BrakeTarget = GP.TargetForceMoveMode * b * sqrt(c*c+1) + max(0, C1.CorrectionBrakeTorqueAmplitude * (cos(2 * C2.AbsolutePosition) - cos (2 * pi * C1.CorrectionBrakeTorqueFactor / 180 )));
        C2DS.BrakeTarget = (GP.TargetForceMoveMode + DeflectionCorrectionMoveMode * 0.5 * (1-cos(2*C2.AbsolutePosition))) * C2.EffectiveArmLength * ElevationCorrection;
```

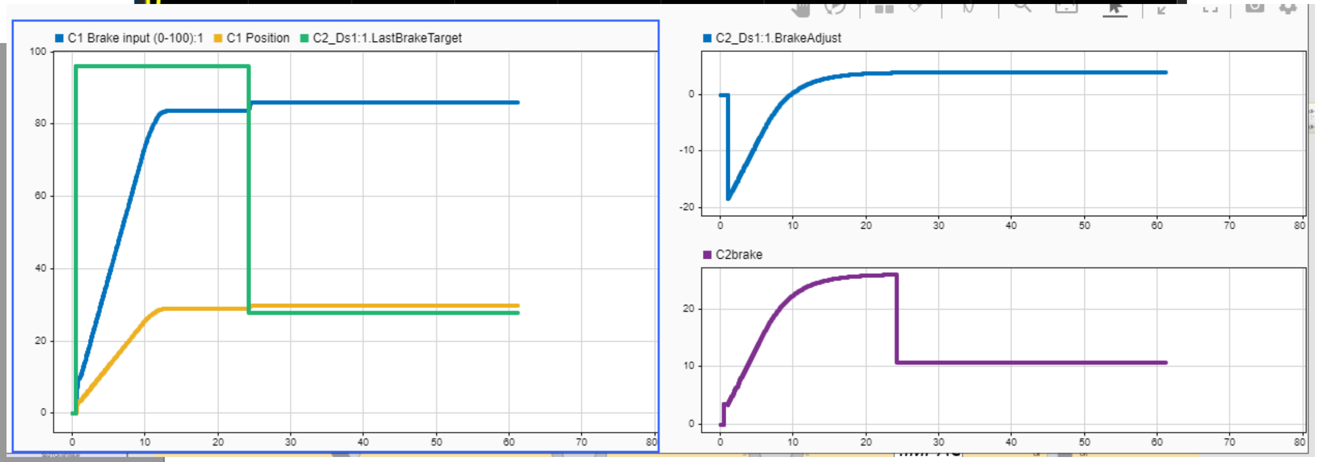


Project: Skytron Freedom

- Simulate against model of ‘real world’

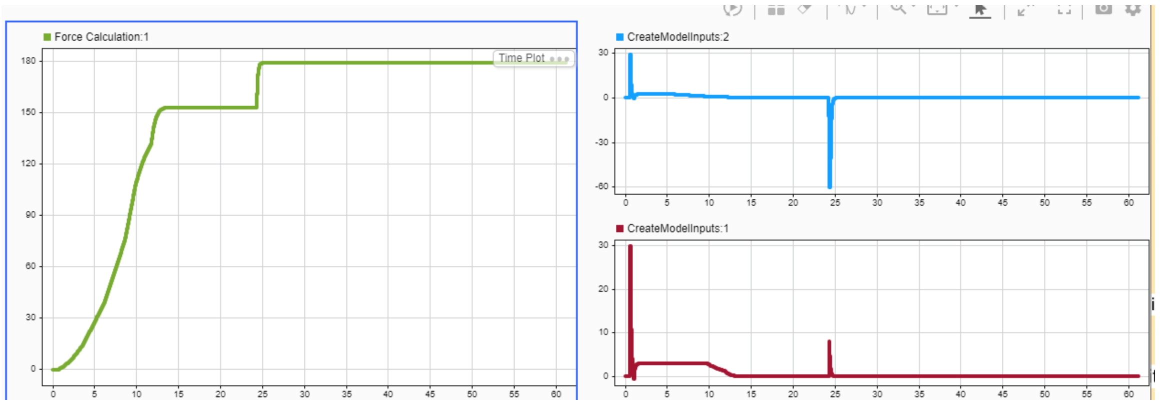


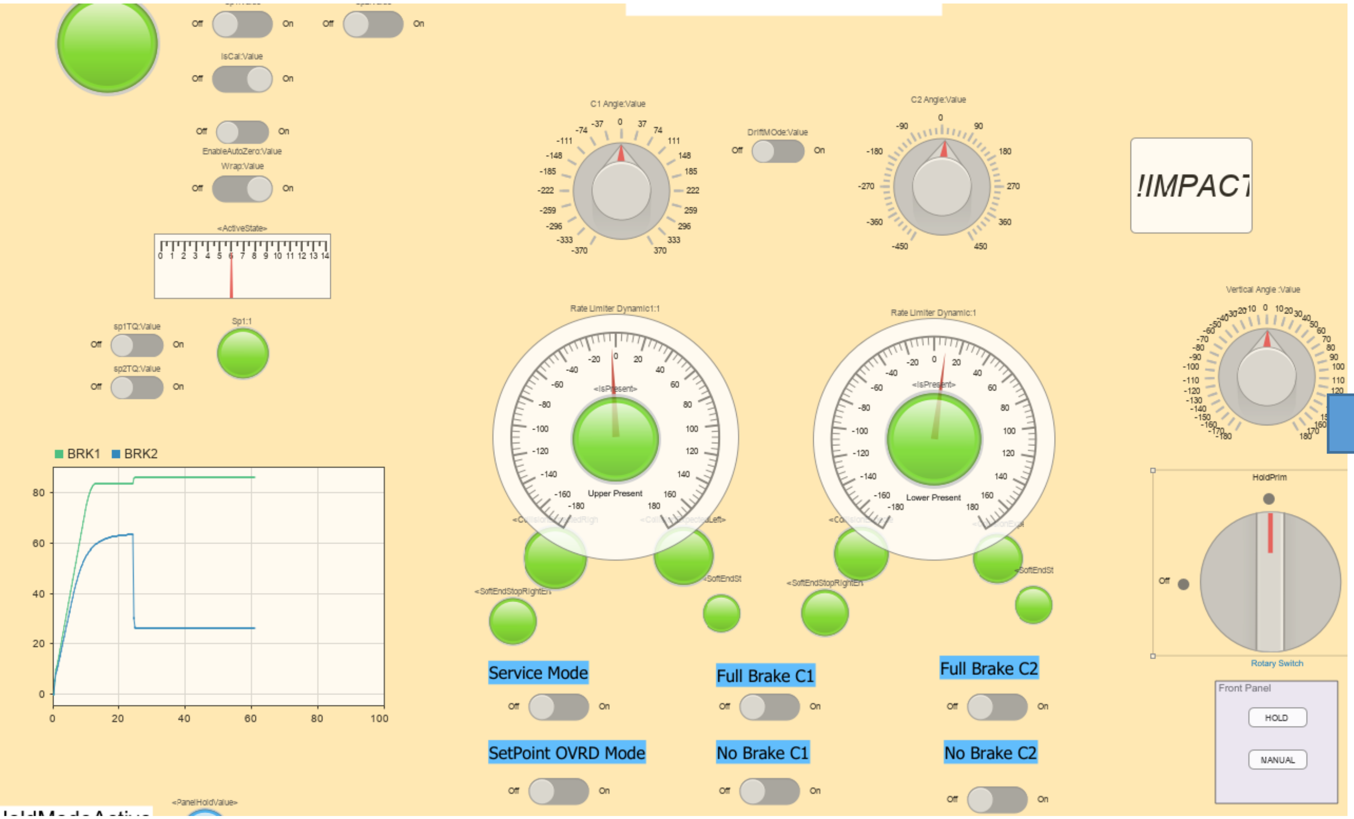


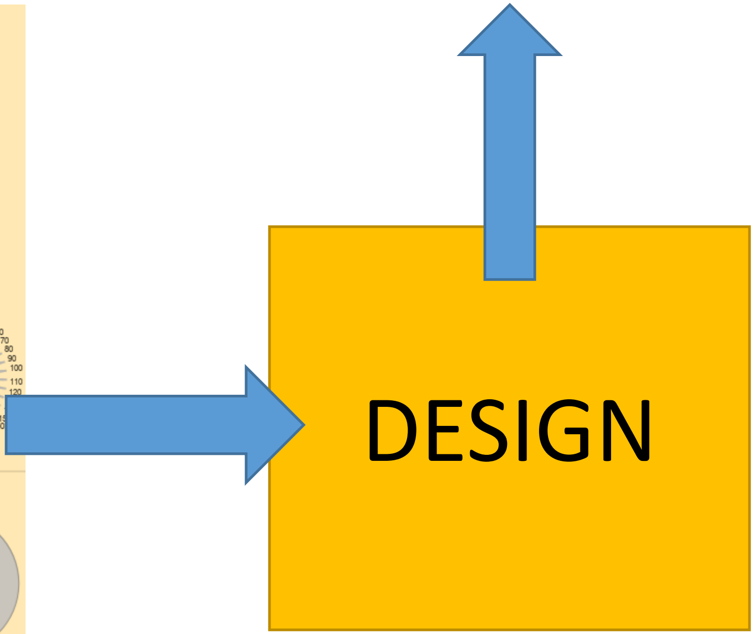


Project: Skytron Freedom

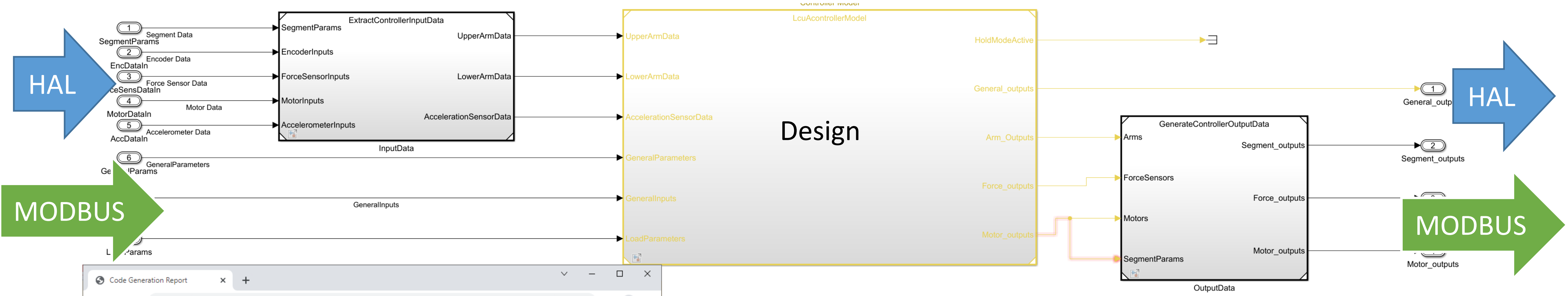
- Simulate against specific scenario's

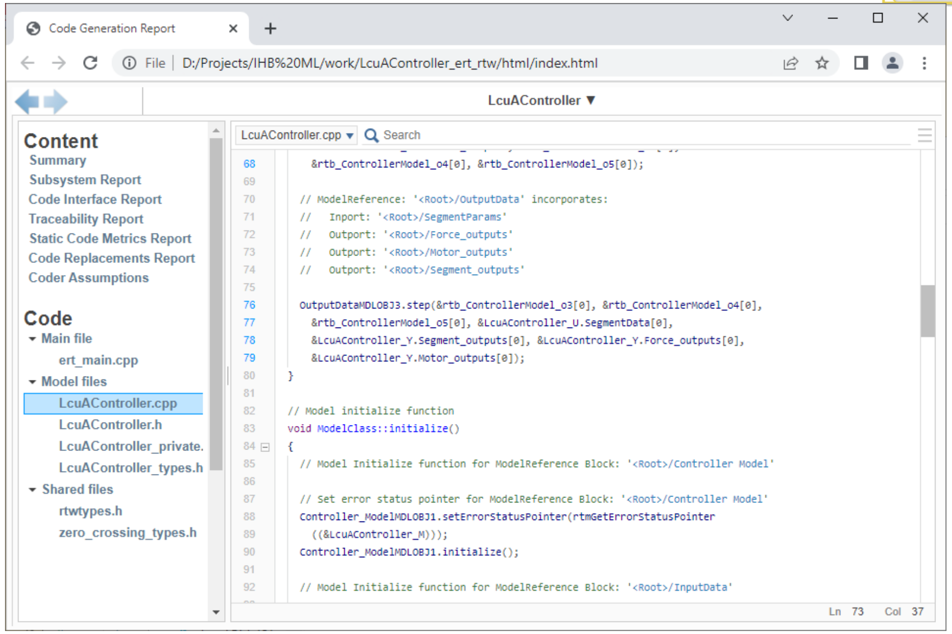


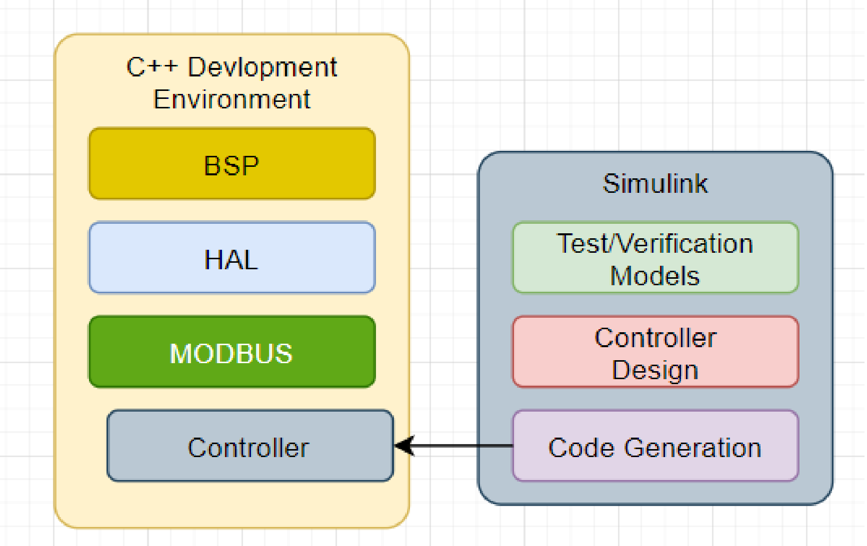




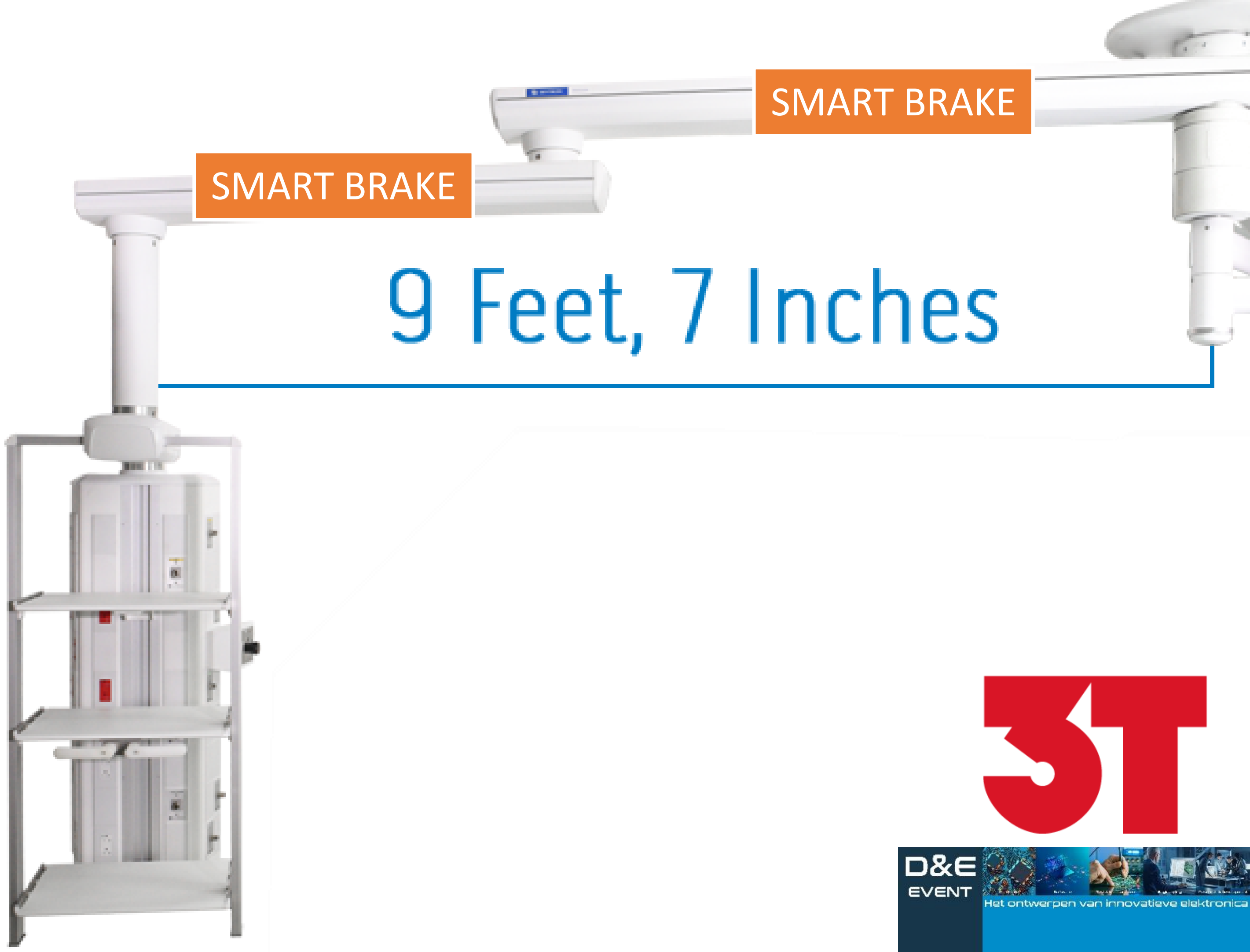
Project: Skytron Freedom







Project: Skytron Freedom



Project: Skytron Freedom

Integration:

- Comparison between model and mechanical system
  - Optimize system and model
- Rapid design iterations to achieve the *Excellent user Experience*

Project: Skytron Freedom

Reduced time and risk;

- Design and design-changes represented in common language
- The design has already gone through several evaluation and optimization cycles
- (Major) Changes in high level structure can be made fairly easy without too much pain
- Understanding and solving integration issues without needing access to system
- SW development not longer limited to SW designers

How to reach us

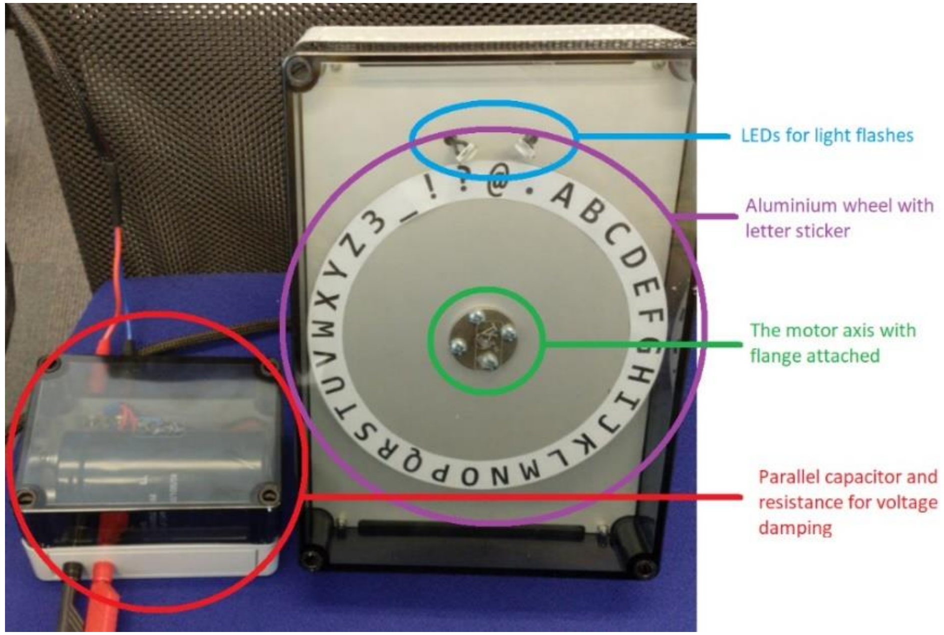
Marcel van Wilpe

Business Developer

marcel.vanwilpe@kendrion.com

+31 538516626

www.3T.nl



We’re at stand number 2

With an FPGA based MDD demonstration!