

PEARL Workshop 2007

06.12.2007

Echtzeit- und Regelungstechnische Aspekte bei der automatischen Transformation von Matlab/Simulink in SPS-basierten Steuerungscode

M. Sc. Gülden Bayrak,

Dipl.-Ing. Andreas Wannagat, Prof. Dr.-Ing. Vogel-Heuser

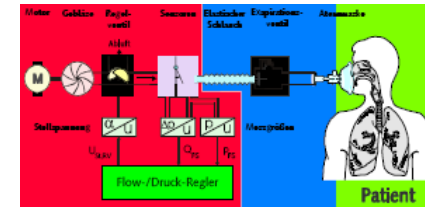
- Motivation
- Stand der Technik
- Codegenerator
 - Schnittstellen
 - Prinzip der Codegenerierung
- Evaluation
- Zusammenfassung und Ausblick

- **Motivation**
- Stand der Technik
- Codegenerator
 - Schnittstellen
 - Prinzip der Codegenerierung
- Evaluation
- Zusammenfassung und Ausblick

Automobiltechnik



Medizintechnik



Windenergieanlagen



Spritzgusstechnik



Luft- und Raumfahrt



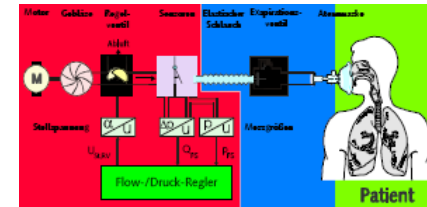
Anwendungsgebiete:

Messtechnik, Steuerungs- und
Regelungstechnik, Bildverarbeitung, Signalverarbeitung und
Kommunikation

Automobiltechnik



Medizintechnik



Windenergieanlagen



Spritzgusstechnik



Abfüllanlagen



**Automatisierungstechnik
IEC 61131-3**

Luft- und Raumfahrt



- Unterschiedliche Werkzeuge je nach Projektphase und Personengruppe
- In der Entwicklung regelungstechnischer Modelle werden spezialisierte Werkzeuge wie Matlab/Simulink eingesetzt
- Manueller Übergang zwischen den Projektphasen
 - Mehraufwand
 - Gefahr von Übertragungsfehlern

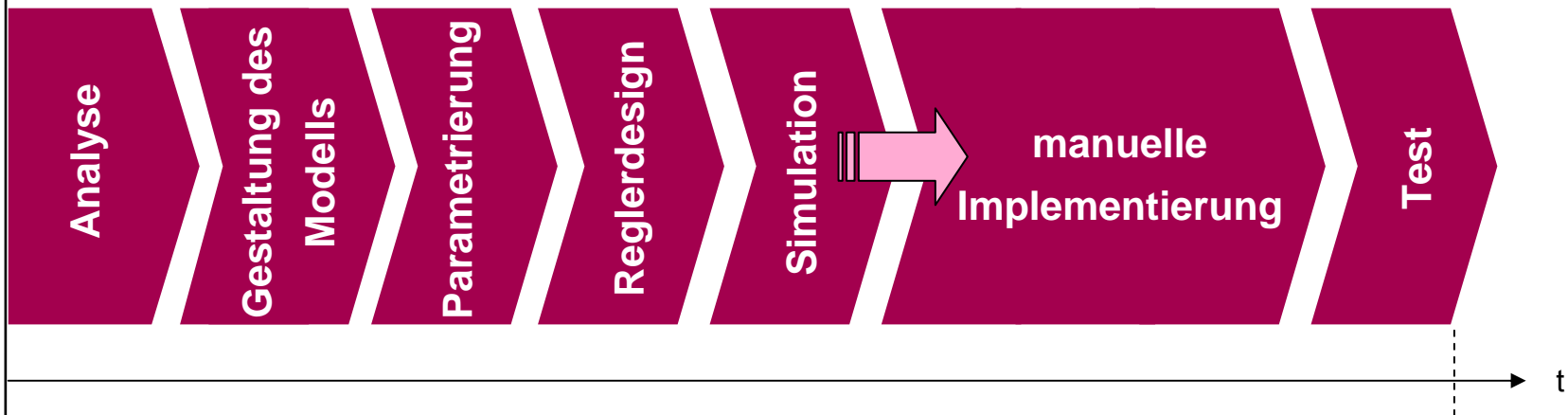
Phasen einer Reglerentwicklung



Standard-Lösung
Vorgehensweise ohne Codegenerierung

- Unterschiedliche Werkzeuge je nach Projektphase und Personengruppe
- In der Entwicklung regelungstechnischer Modelle werden spezialisierte Werkzeuge wie Matlab/Simulink eingesetzt
- Manueller Übergang zwischen den Projektphasen
 - Mehraufwand
 - Gefahr von Übertragungsfehlern

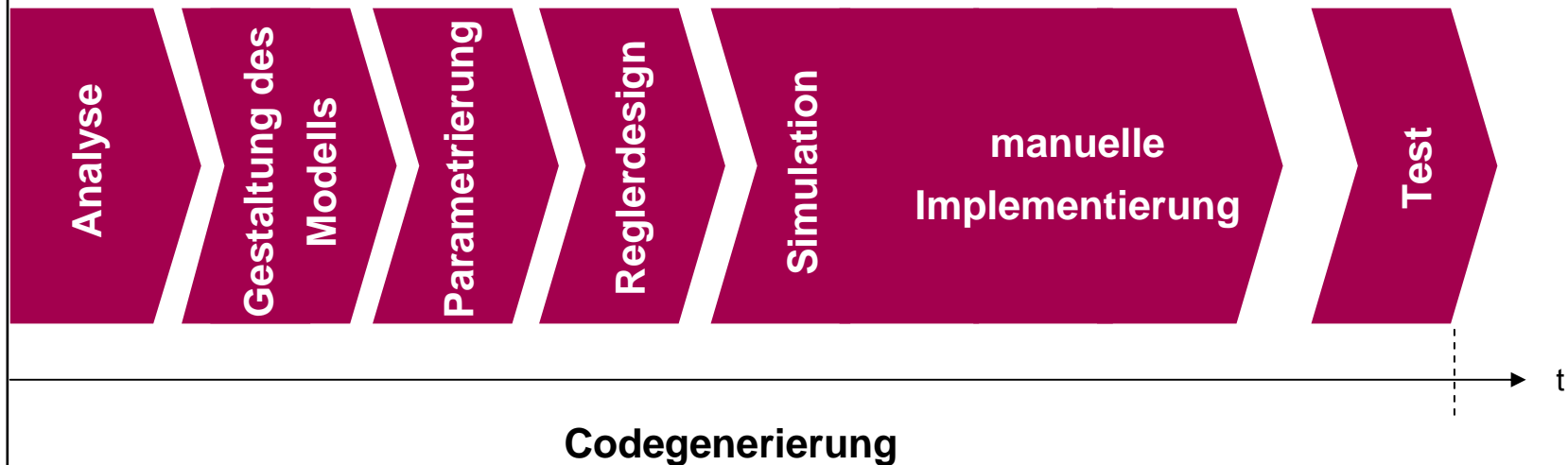
Phasen einer Reglerentwicklung



Standard-Lösung
Vorgehensweise ohne Codegenerierung

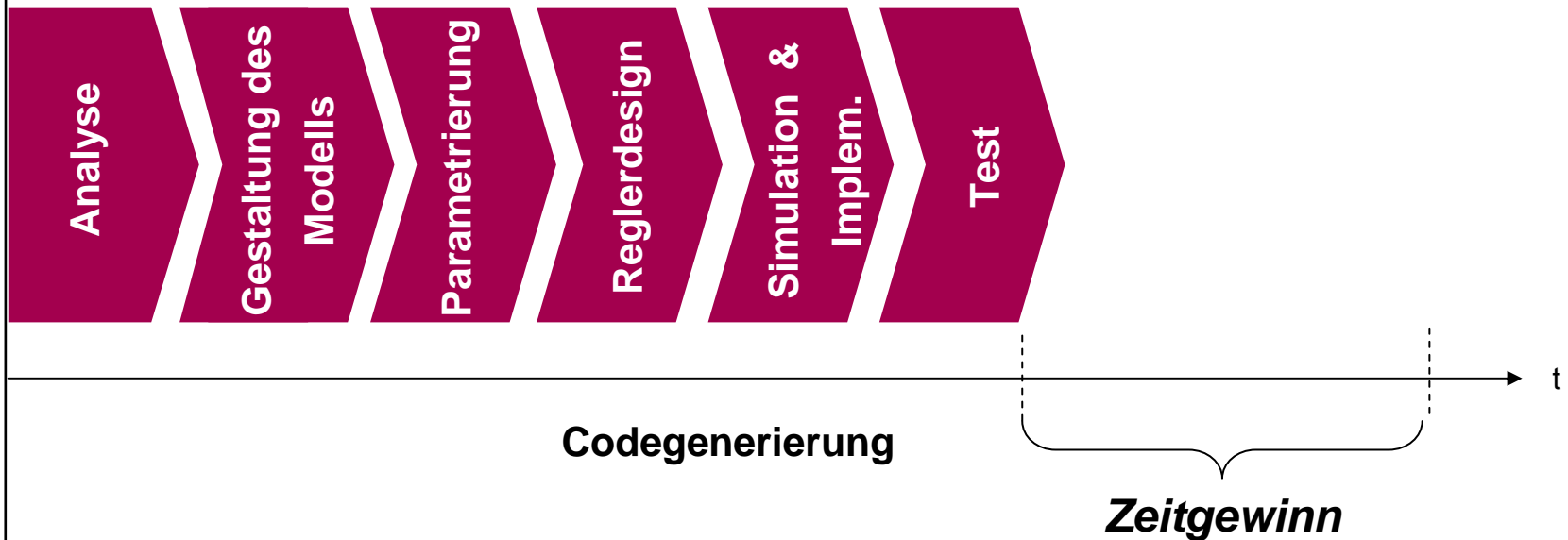
- Unterschiedliche Werkzeuge je nach Projektphase und Personengruppe
- In der Entwicklung regelungstechnischer Modelle werden spezialisierte Werkzeuge wie Matlab/Simulink eingesetzt
- Manueller Übergang zwischen den Projektphasen
 - Mehraufwand
 - Gefahr von Übertragungsfehlern

Phasen einer Reglerentwicklung




- Generierung von Steuerungscode
 - die gewohnte Entwicklungsumgebung mit ihren Vorteilen bleibt erhalten
 - der manuelle fehleranfällige Transfer in die Zielsprache fällt weg

Phasen einer Reglerentwicklung

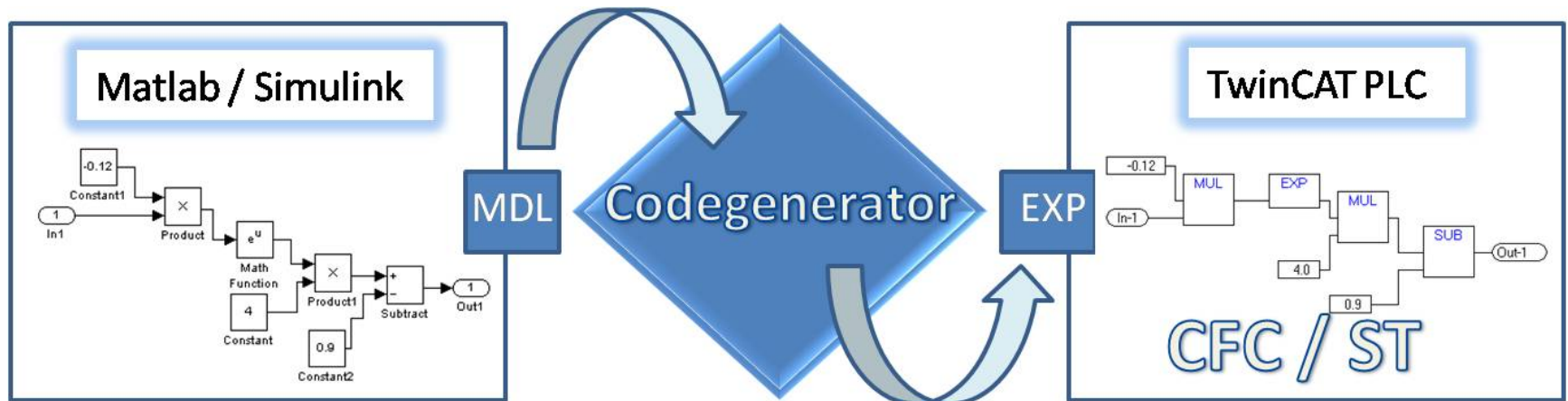


- Motivation
- **Stand der Technik**
- Codegenerator
 - Schnittstellen
 - Prinzip der Codegenerierung
- Evaluation
- Zusammenfassung und Ausblick

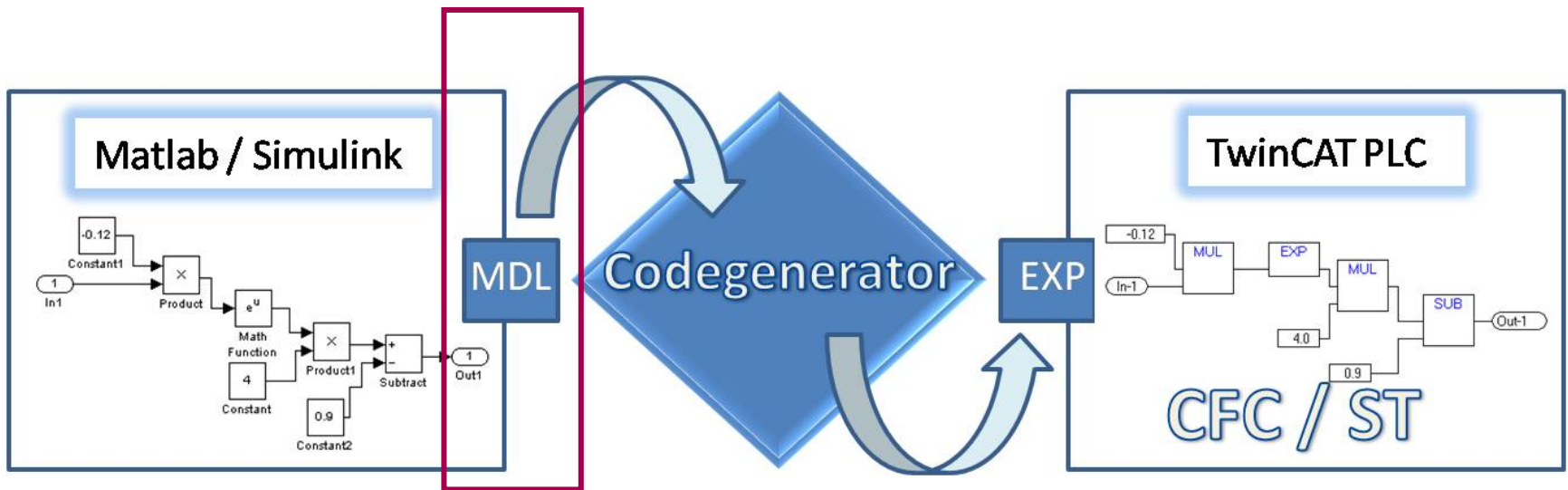
	Vorteil	Nachteil
dSpace	<ul style="list-style-type: none"> • Modellierung in Matlab/Simulink 	<ul style="list-style-type: none"> • C-Code nur für Mikrokontroller • Zusätzliche Kosten (RTW) • Rückverfolgung schwierig
Bachmann	<ul style="list-style-type: none"> • Modellierung in Matlab/Simulink 	<ul style="list-style-type: none"> • C- Code nur für ihre eigene M1-Steuerung • Zusätzliche Kosten (RTW) • Rückverfolgung schwierig
HybridTech	<ul style="list-style-type: none"> • Modellierung in Matlab/Simulink • Übersetzung in die DIN EN IEC 61131-3 Strukturierter Text 	<ul style="list-style-type: none"> • Übersetzung nur in die Sprache des strukturierten Texts • Rückverfolgung schwierig

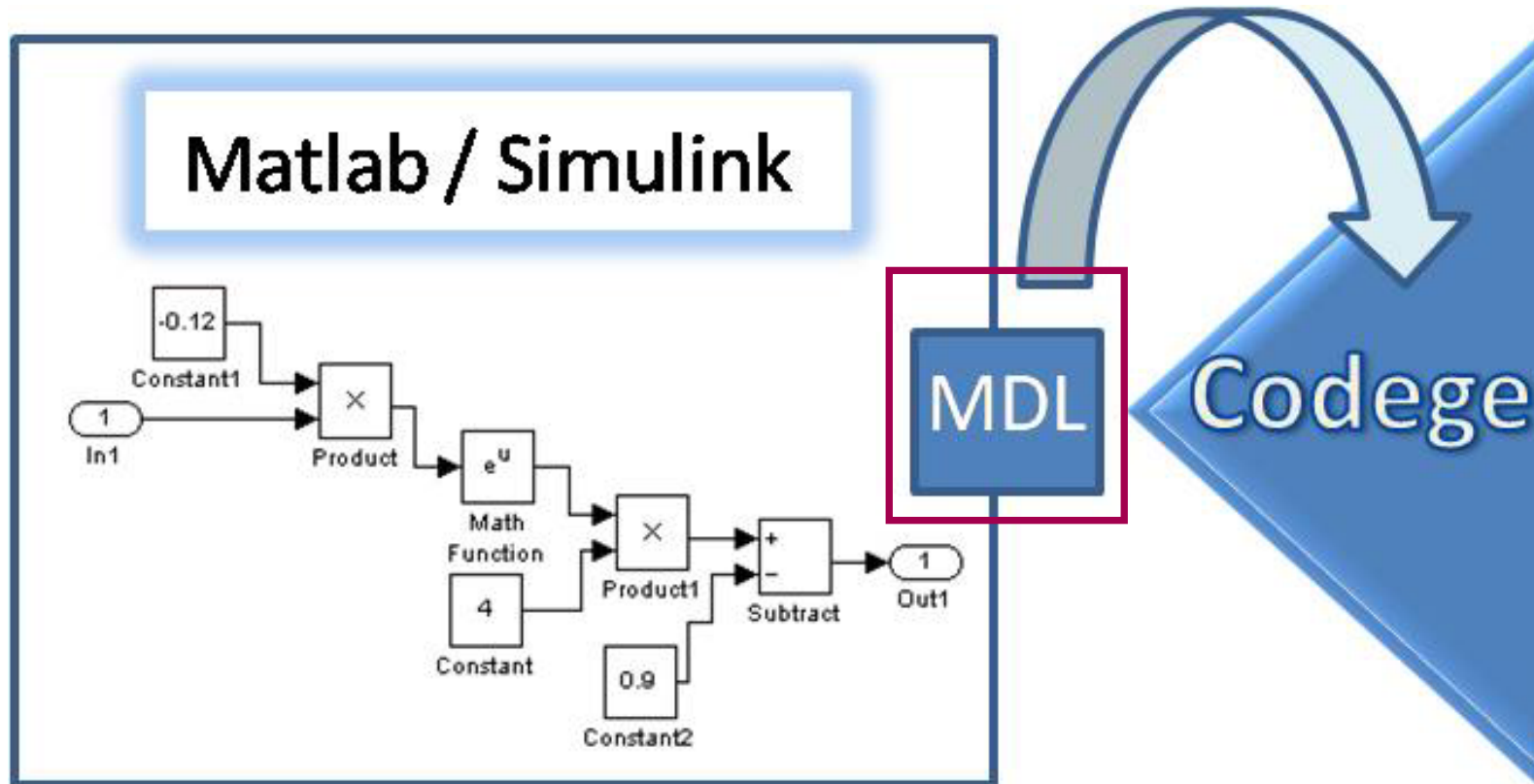
	Vorteil	Nachteil
dSpace	<ul style="list-style-type: none"> • Modellierung in Matlab/Simulink 	<ul style="list-style-type: none"> • C-Code nur für Mikrokontroller • Zusätzliche Kosten (RTW) • Rückverfolgung schwierig
Bachmann	<ul style="list-style-type: none"> • Modellierung in Matlab/Simulink 	<ul style="list-style-type: none"> • C-Code nur für ihre eigene M1-Steuerung • Zusätzliche Kosten (RTW) • Rückverfolgung schwierig
HybridTech	<ul style="list-style-type: none"> • Modellierung in Matlab/Simulink • Übersetzung in die DIN EN IEC 61131-3 Strukturierter Text 	<ul style="list-style-type: none"> • Übersetzung nur in die Sprache des strukturierten Textes • Rückverfolgung schwierig
Code-generator 	<ul style="list-style-type: none"> • Modellierung in Matlab/Simulink • Übersetzung in die grafische Programmiersprache (FBD/CFC) →Vergleichbarkeit der Modelle →Vereinfachung der Rückverfolgung bzw. der Fehlersuche 	

- Motivation
- Stand der Technik
- **Codegenerator**
 - Schnittstellen
 - Prinzip der Codegenerierung
- Evaluation
- Zusammenfassung und Ausblick

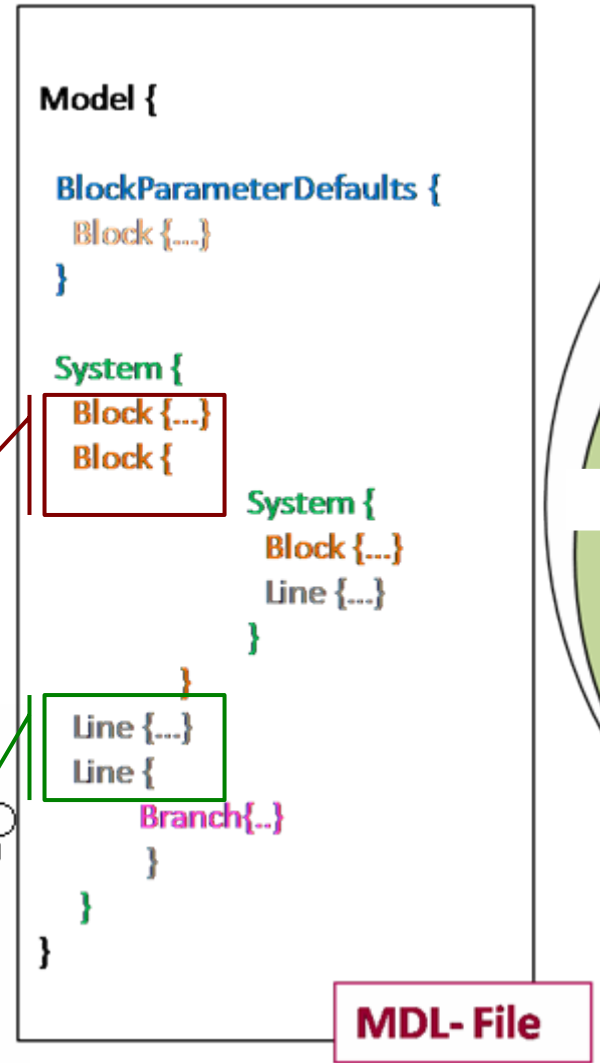
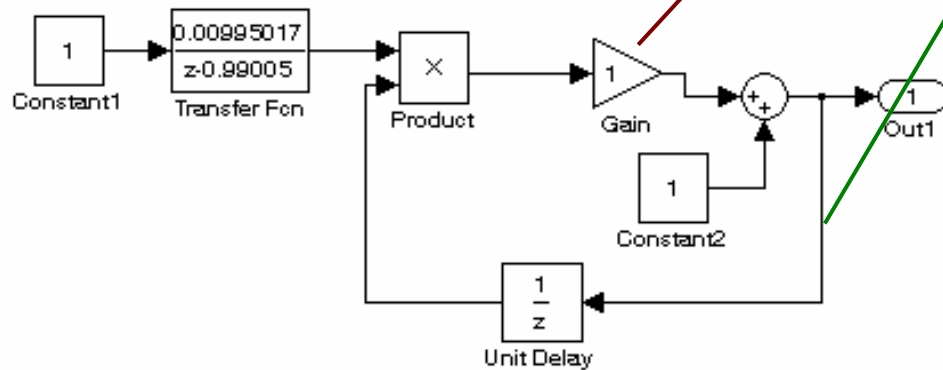
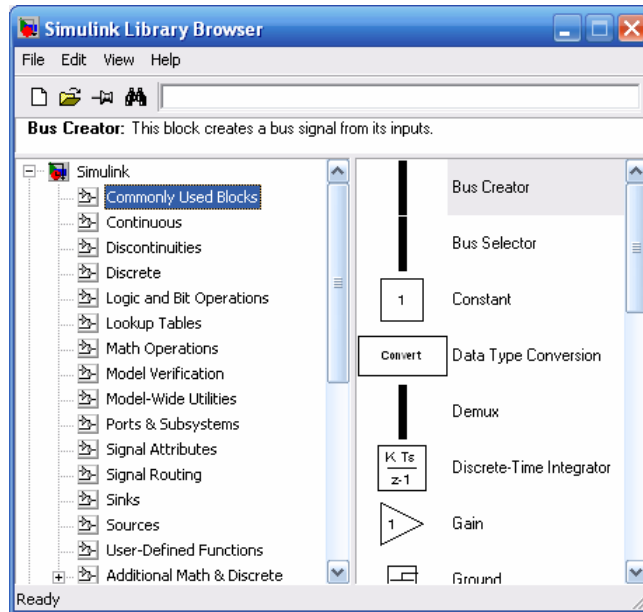


- Motivation
- Stand der Technik
- **Codegenerator**
 - **Schnittstellen**
 - Prinzip der Codegenerierung
- Evaluation
- Zusammenfassung und Ausblick

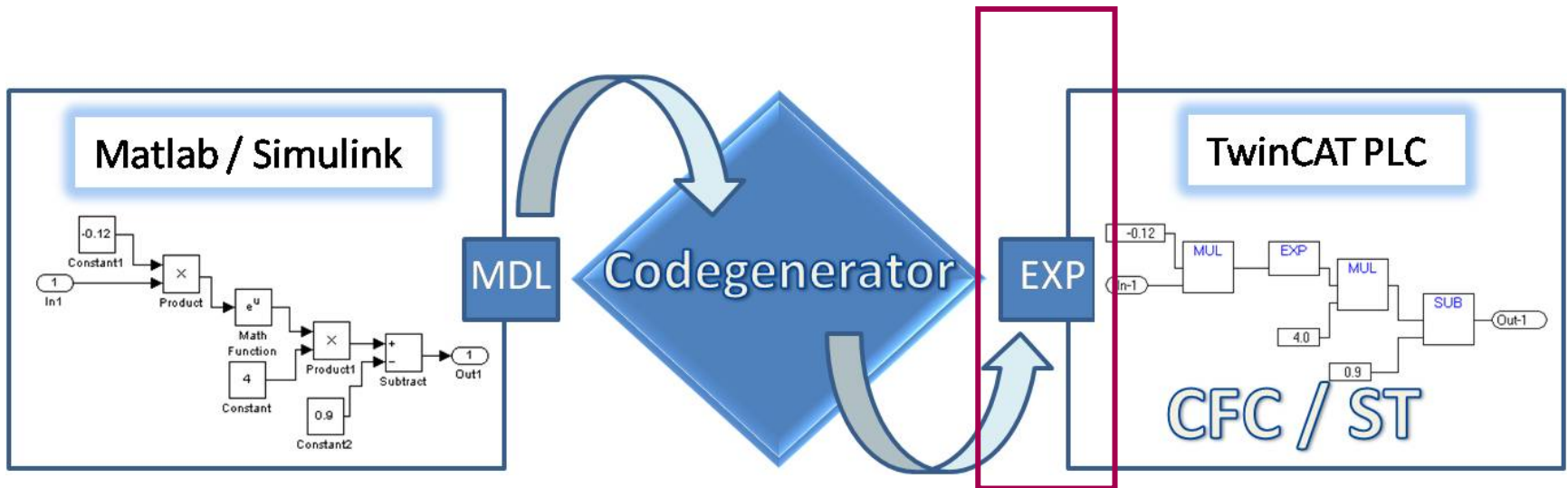




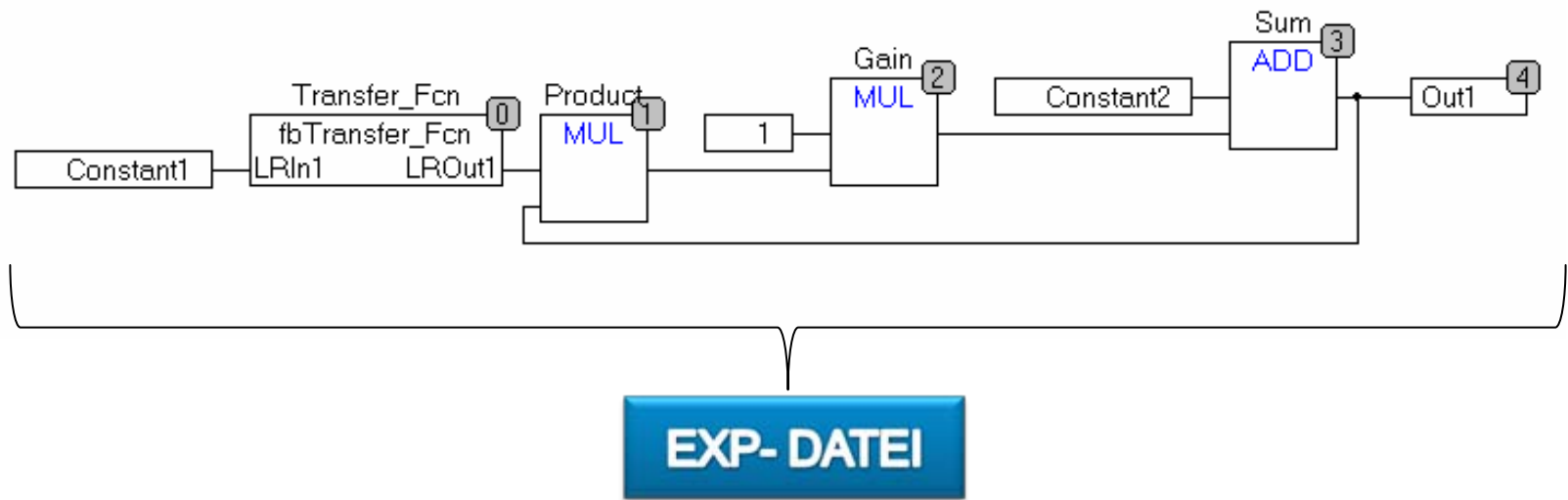
Speicherformat von Simulink



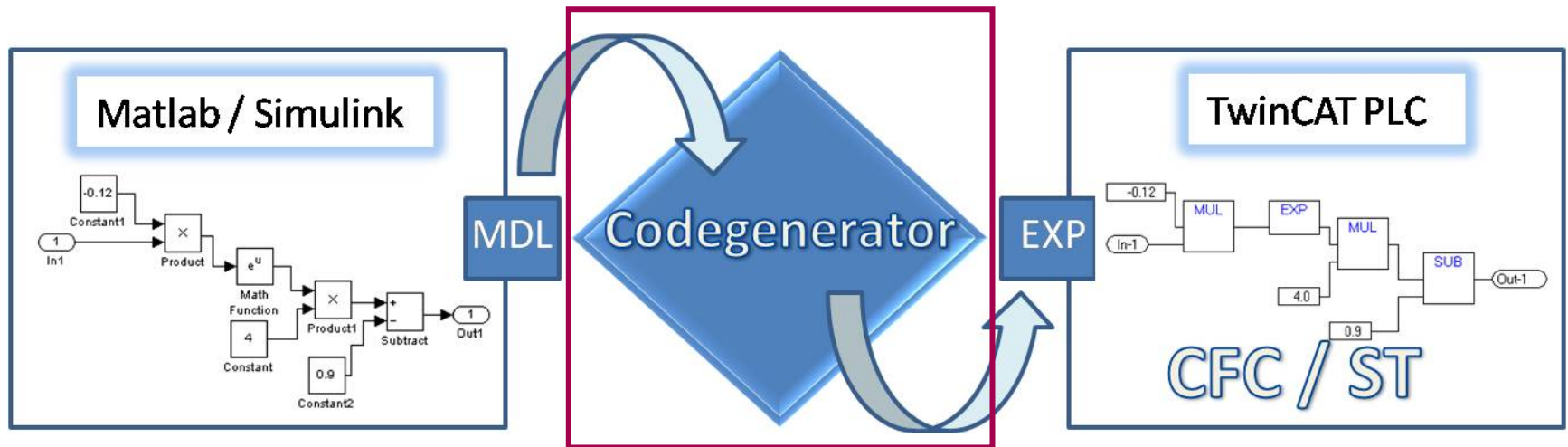
- Motivation
- Stand der Technik
- **Codegenerator**
 - **Schnittstellen**
 - Prinzip der Codegenerierung
- Evaluation
- Zusammenfassung und Ausblick

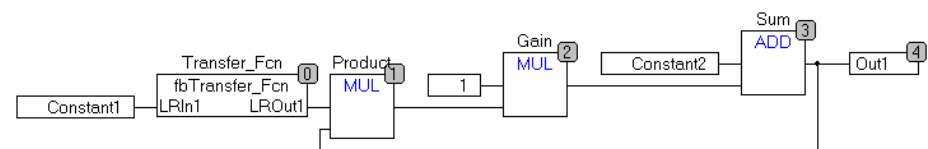
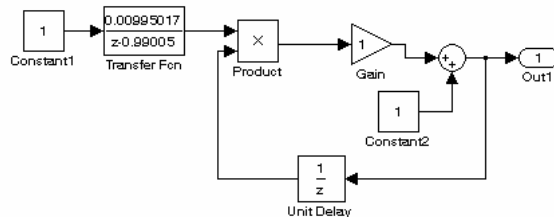
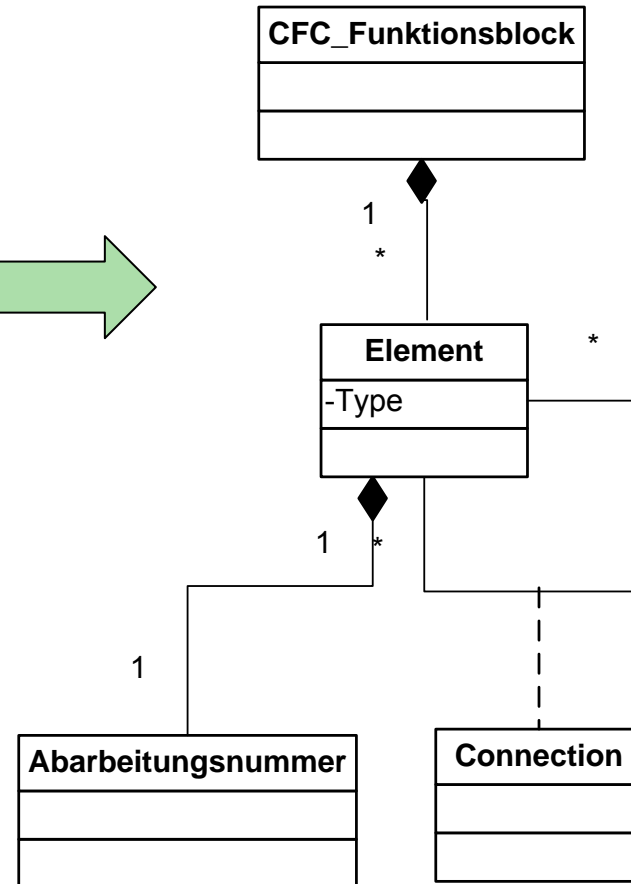
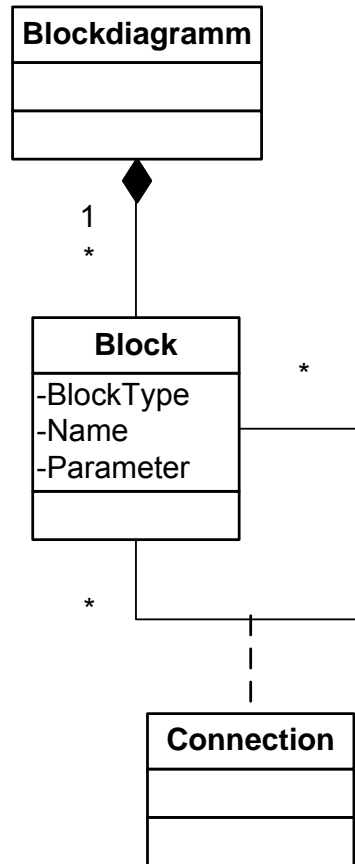


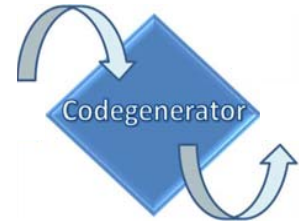
17 Folien , 13.12.2007 12:00 © ES



- Motivation
- Stand der Technik
- **Codegenerator**
 - Schnittstellen
 - **Prinzip der Codegenerierung**
- Evaluation
- Zusammenfassung und Ausblick

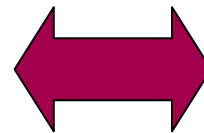






Simulink Bibliothek	
BlockType	Parameter
TransferFcn	<i>numerator</i>
Saturation	<i>min, max</i>
PID	<i>P, I, D</i>
⋮	⋮

TwinCAT Controller Bibliothek	
ControllerType	Parameter
FB_CTRL_TRANS	<i>numerator</i>
FB_CTRL_LIMITER	<i>min,max</i>
FB_CTRL_PID	<i>Kp, Tn, Tv</i>
⋮	⋮



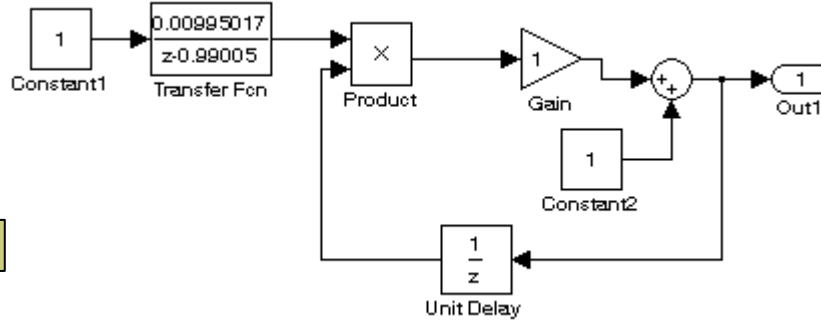
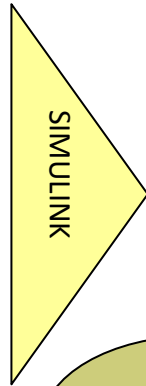
Zuordnungstabelle

Values
➔

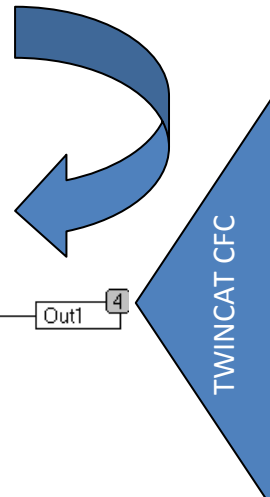
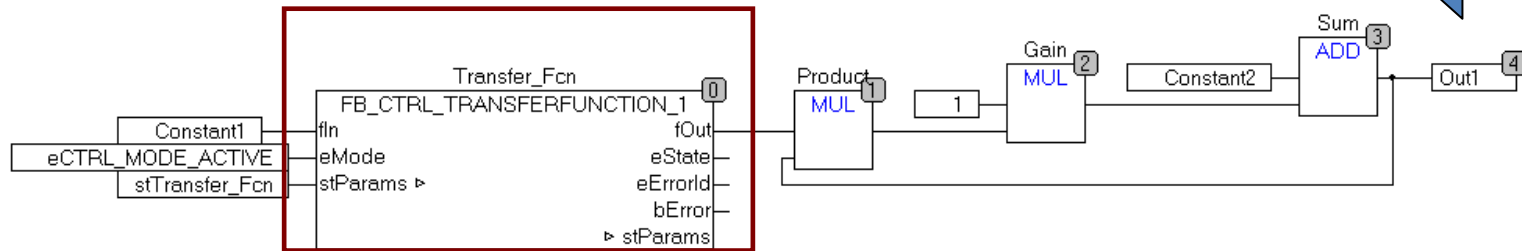
BlockType	ControllerType
TransferFcn	FB_CTRL_TRANS
Saturation	FB_CTRL_LIMITER
PID	FB_CTRL_PID
⋮	⋮

➔ Values

1. Möglichkeit: direkte Abbildung auf die Bausteine der CLC Bibliothek



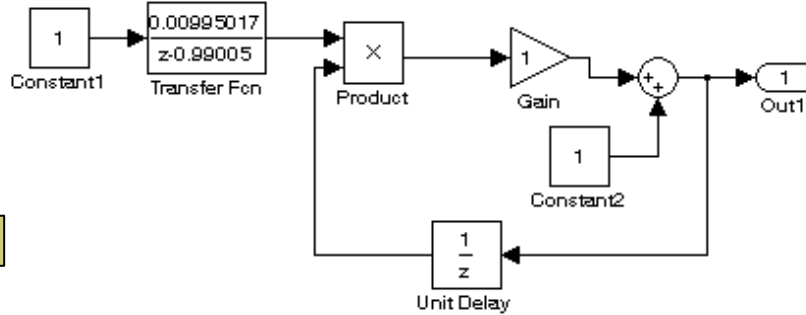
Simulink-Blöcke	Parameterübergabe / Umformungen	Baustein aus der CLC Bibliothek
		<pre> FB_CTRL_TRANSFERFUNCTION_1 -fIn fOut -fManValue eState -eMode eErrorId -stParams ▶ bError </pre>



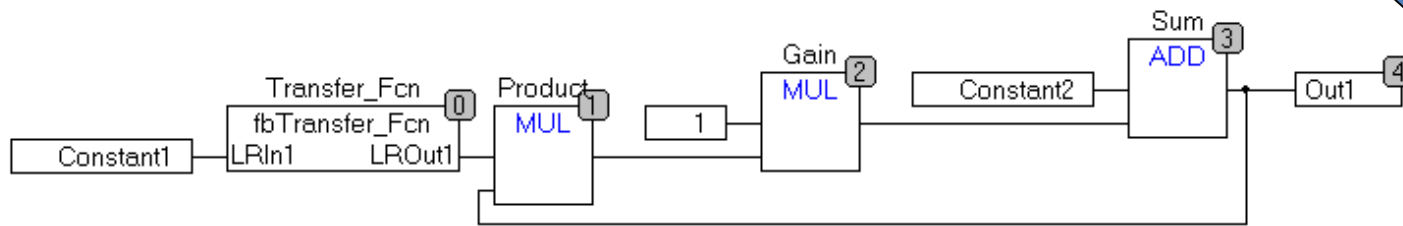
Strukturähnliche Abbildung

2. Möglichkeit: Anpassung der Struktur

SIMULINK



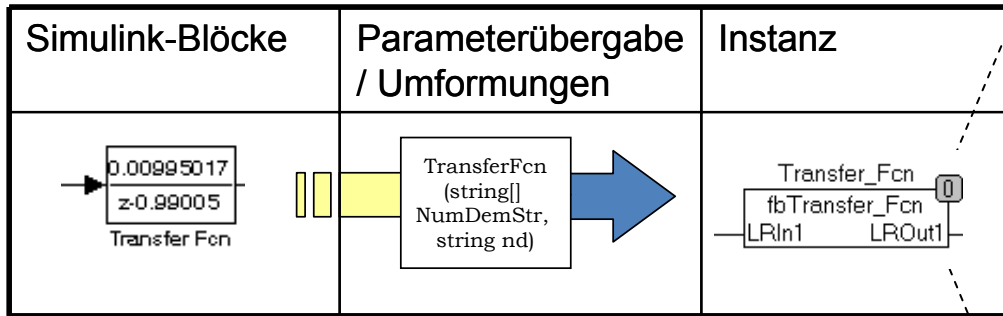
Simulink-Blöcke	Parameterübergabe / Umformungen	Instanz



TWINCAT CFC

2. Möglichkeit: Anpassung der Struktur

Instanz des Controllerbausteines
FB_CTRL_TRANSFER



Parameterwerte werden im
Funktionsbaustein initialisiert

```

FUNCTION_BLOCK fbTransfer_Fcn
VAR_INPUT
    LRIn1 : LREAL;
END_VAR
VAR_OUTPUT
    LROut1 : LREAL;
END_VAR
VAR_IN_OUT
END_VAR
VAR
    bFirstInit : BOOL := TRUE;
    fbTransfer_Fcn : FB_CTRL_TRANSFER;
    ar_fNumeratorArray : ARRAY[0..1] OF FL
END_VAR

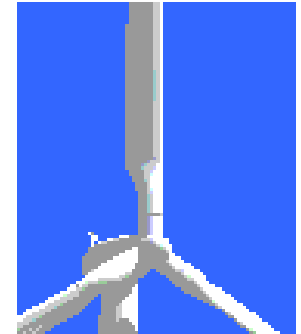
IF bFirstInit THEN
    ar_fNumeratorArray[0] := 0.00995017;
    ar_fNumeratorArray[1] := 0;
    ar_DenominatorArray[0] := -0.99005;
    ar_DenominatorArray[1] := 1;
    stTransfer_Fcn.nOrderOfTheTransferfunc := 1;
    stTransfer_Fcn.tCtrlCycleTime := T#50ms;
    stTransfer_Fcn.tTaskCycleTime := T#50ns;
    bFirstInit := FALSE;
END_IF

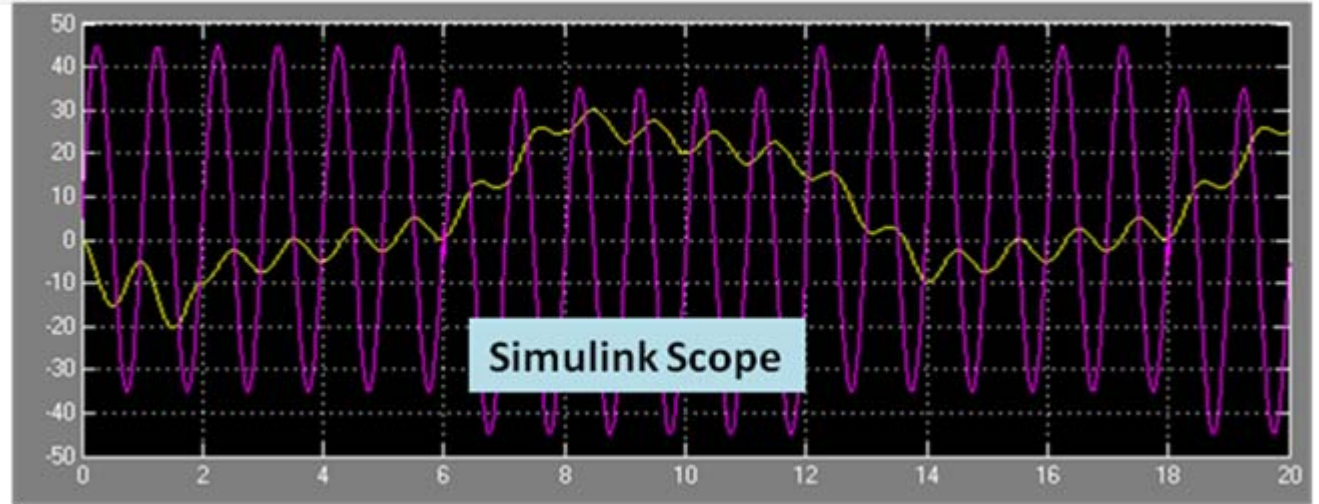
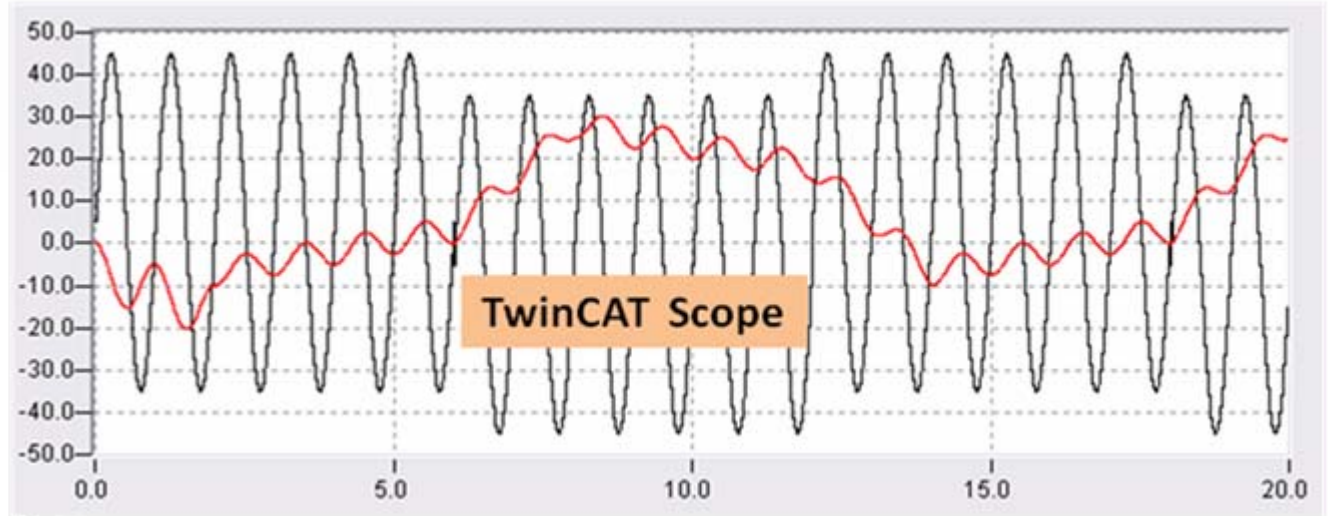
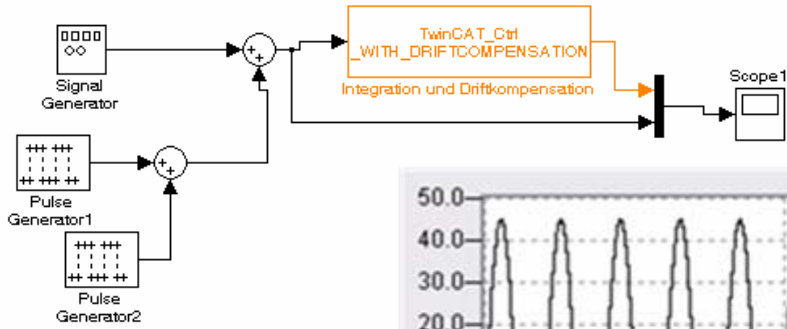
stTransfer_Fcn.pNumeratorArray_ADR := ADDR(fbTransfer_Fcn.ar_fNumeratorArray);
stTransfer_Fcn.nNumeratorArray_SIZEOF := SIZEOF(fbTransfer_Fcn.ar_fNumeratorArray);
stTransfer_Fcn.pDenominatorArray_ADR := ADDR(fbTransfer_Fcn.ar_DenominatorArray);
stTransfer_Fcn.nDenominatorArray_SIZEOF := SIZEOF(fbTransfer_Fcn.ar_DenominatorArray);
stTransfer_Fcn.pTransferfunction1Data_ADR := ADDR(fbTransfer_Fcn.stTransfer_Fcn);
stTransfer_Fcn.nTransferfunction1Data_SIZEOF := SIZEOF(fbTransfer_Fcn.stTransfer_Fcn);

fbTransfer_Fcn(
    fln:= LRIn1,
    eMode:= eCTRL_MODE_ACTIVE,
    stParams:= stTransfer_Fcn,
    fOut=> LROut1);
    
```


- Motivation
- Stand der Technik
- Codegenerator
 - Schnittstellen
 - Prinzip der Codegenerierung
- **Evaluation**
- Zusammenfassung und Ausblick

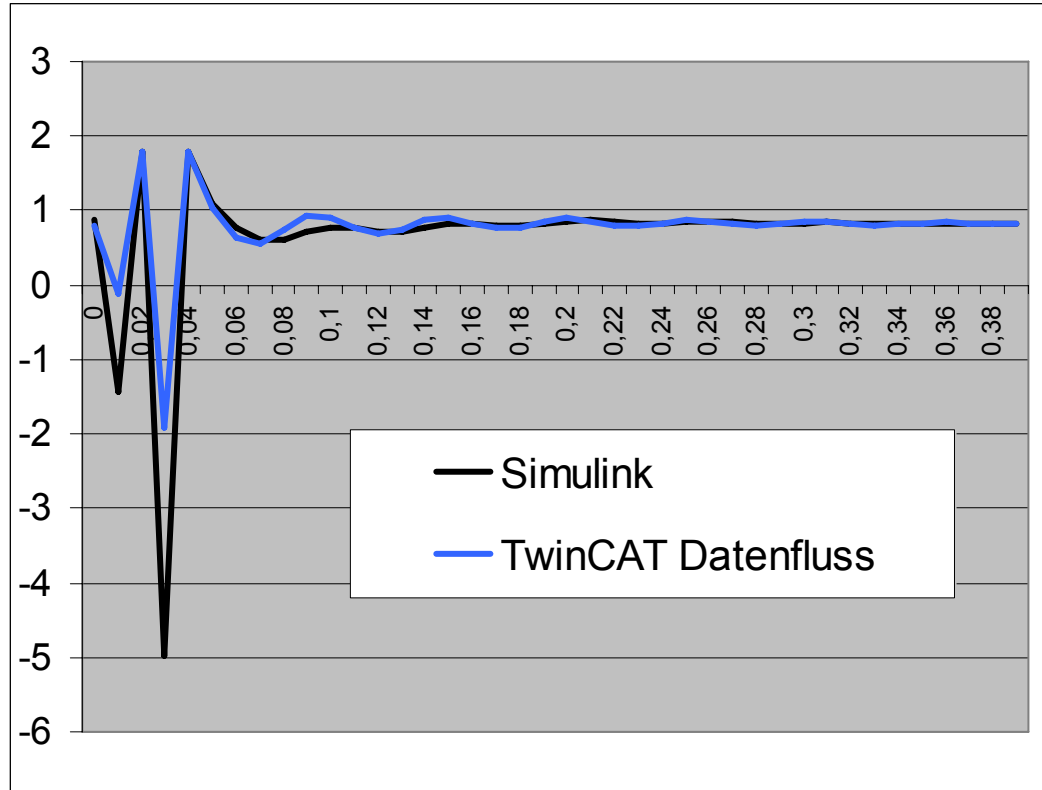
- Die Blattwinkelregelung hat als Eingangsgrößen:
 - die Drehzahl am Generator,
 - die Windgeschwindigkeit,
 - die Blattwinkel der drei Rotorblätter und
 - die Gondelbeschleunigung in Richtung der Rotorachse.
- Ausgangsgrößen:
 - Blattwinkel aller drei Rotorblätter
- Anzahl der Blöcke: 51
- Anzahl der Verbindungen: 62
- Anzahl der Subsysteme: 3

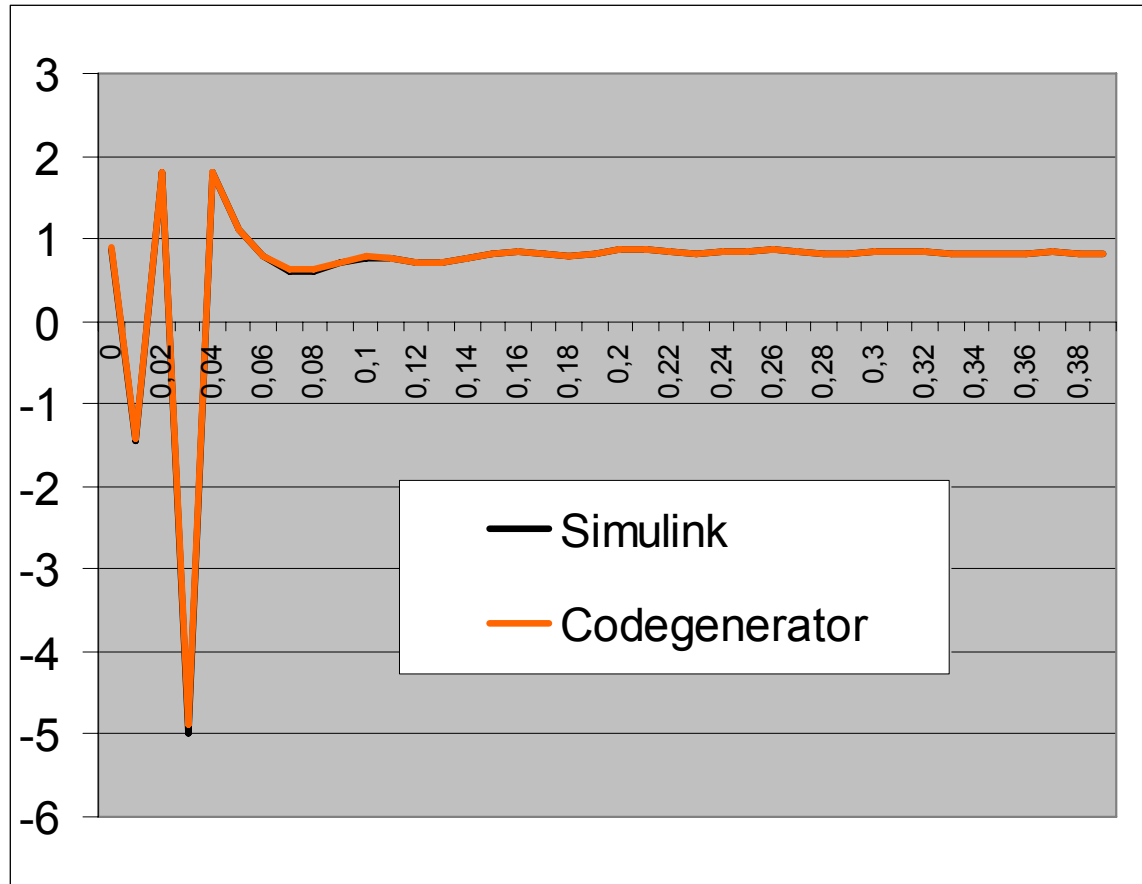




Zeit t in sec

Zeitvergleich zwischen manueller Übersetzung (ohne Berücksichtigung der SPS-Eigenschaften) und Simulink





Zeit t in sec

— Simulink
— Codegenerator

- Motivation
- Stand der Technik
- Codegenerator
 - Schnittstellen
 - Prinzip der Codegenerierung
- Evaluation
- **Zusammenfassung und Ausblick**

- Strukturähnliche Übersetzung
- Übersetzung in Sekunden
 - Übersetzung des Beispielmotells (Blattwinkelregelung) \approx 1 Sekunde
- Keine zusätzlichen Kosten (durch die Anschaffung von RTW)
- Korrekte Funktionsweise wurde evaluiert (Blattwinkelregler)
- Berücksichtigung der SPS Eigenschaften
 - Zeitverhalten nahezu identisch wie in Simulink

- Realisierung der Rücktransformation, um die verschiedenen Modelle auch nach Änderungen synchronisieren zu können
- Transformation des Zustandsdiagramms (aus Simulink) in die Ablaufsprache der IEC 61131-3
- Online-Verbindung zwischen Simulink und der Steuerungssoftware, um die Fehlersuche im Sinne eines „Debuggings“ zu ermöglichen

Vielen Dank für Ihre Aufmerksamkeit!

