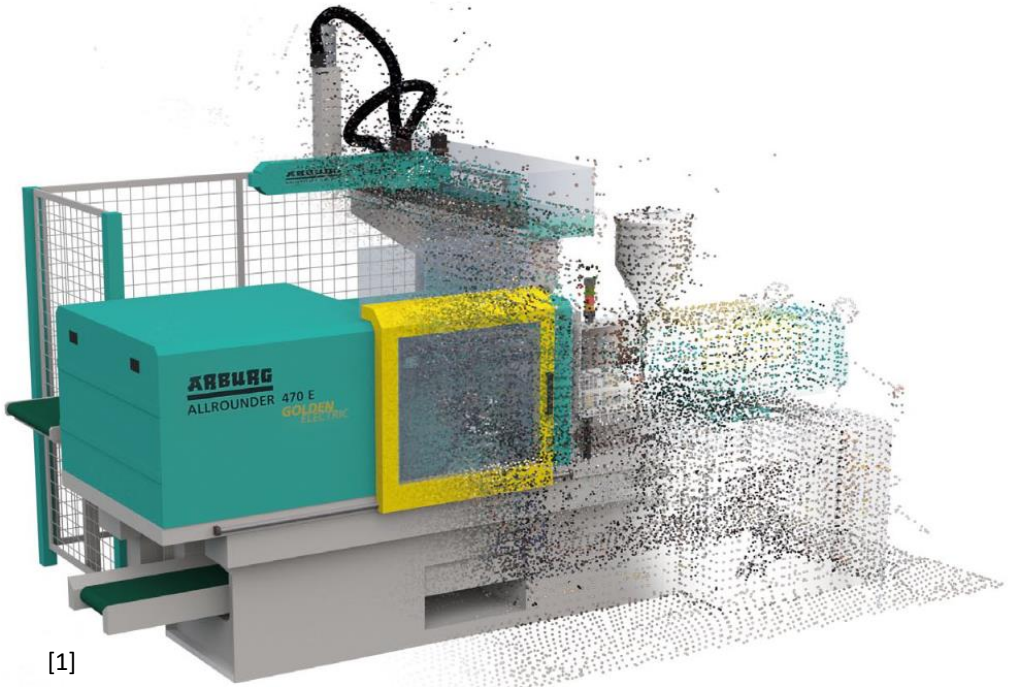
EUROPÄISCHE UNION:
Investition in Ihre Zukunft
Europäischer Fonds für regionale Entwicklung

Datengetriebe Modellbildung - Teil 2
Seminar
Digital Twin of Injection Molding (DIM)

08.06.2022



[1]

Digital Twin of Injection Molding

EUROPÄISCHE UNION:
Investition in Ihre Zukunft
Europäischer Fonds für regionale Entwicklung

Forschungskonsortium

- Institut für Werkstofftechnik / FG Kunststofftechnik, Prof. Dr.-Ing. H.-P. Heim
- FG Mess- und Regelungstechnik, Prof. Dr.-Ing. A. Kroll

Alexander Rehmer

+

Marco Klute

+

Stefan Rosenbach

+

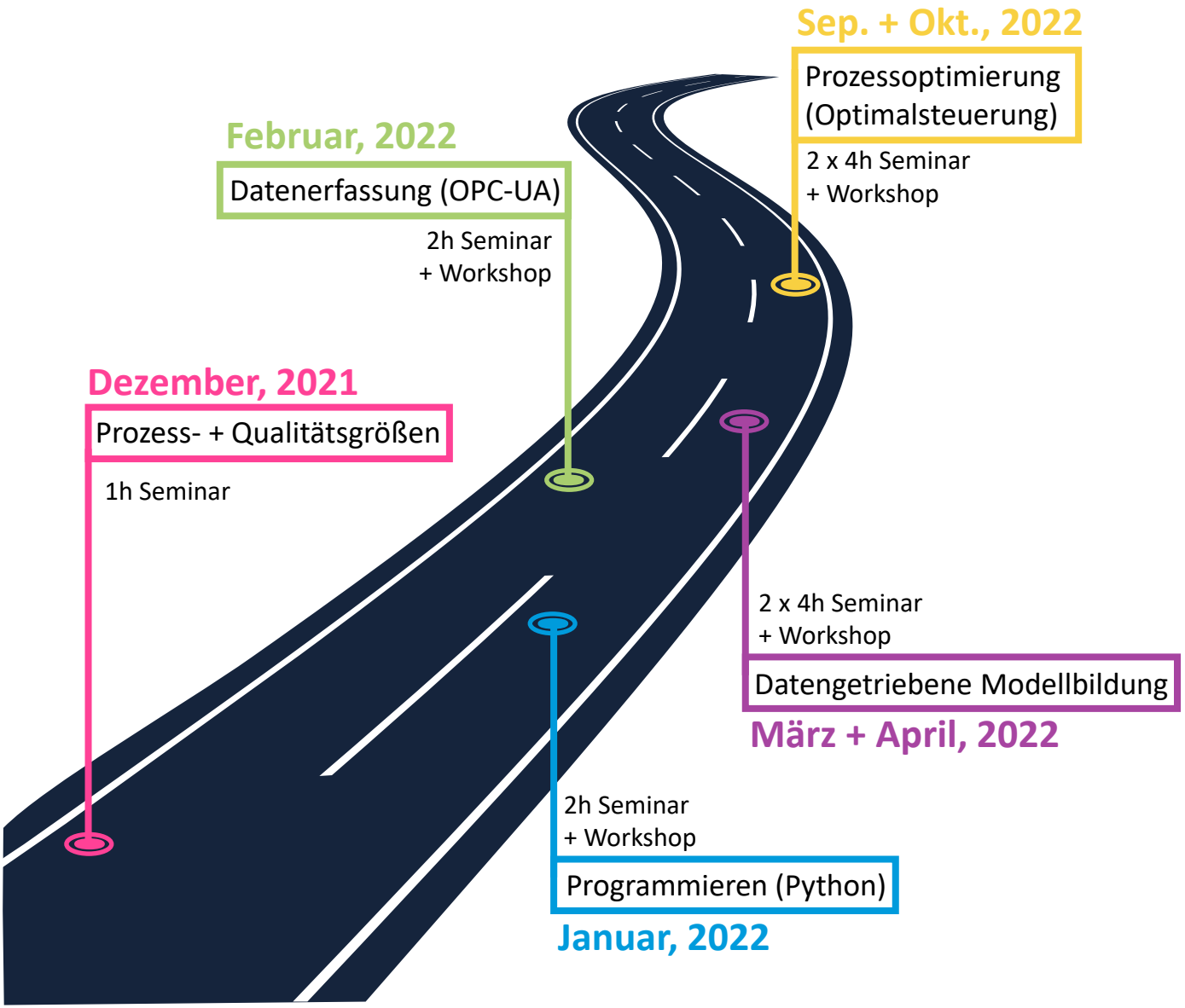
Studentische Hilfskräfte

Kontakt:

- dim@uni-kassel.de
- www.uni-kassel.de/go/DIM

Überblick über die Workshopreihe

EUROPÄISCHE UNION:
Investition in Ihre Zukunft
Europäischer Fonds für regionale Entwicklung



- Erfassung von Prozess- und Qualitätsgrößen
- **Programmieren mit Python**
  - Seminar: Vermittlung grundlegender und fortgeschrittener Aspekte der objektorientierten Programmierung mit Python
  - Workshop: Selbstständige Bearbeitung von Programmieraufgaben
  - Ziele:
    - Installieren und Ausführen von Python
    - Programmieren und Lesen von Python Code
- Datenerfassung mit OPC-UA
- Datengetriebene Modellbildung
- Prozessoptimierung mittels numerischer Optimalsteuerung

Agenda

EUROPÄISCHE UNION:
Investition in Ihre Zukunft
Europäischer Fonds für regionale Entwicklung

- **Rückblick Modellbildung Teil 1**
- **DIM - Toolbox**
- **Datengetriebene Modellbildung des Spritzgießprozesses**
- **Fallstudien**

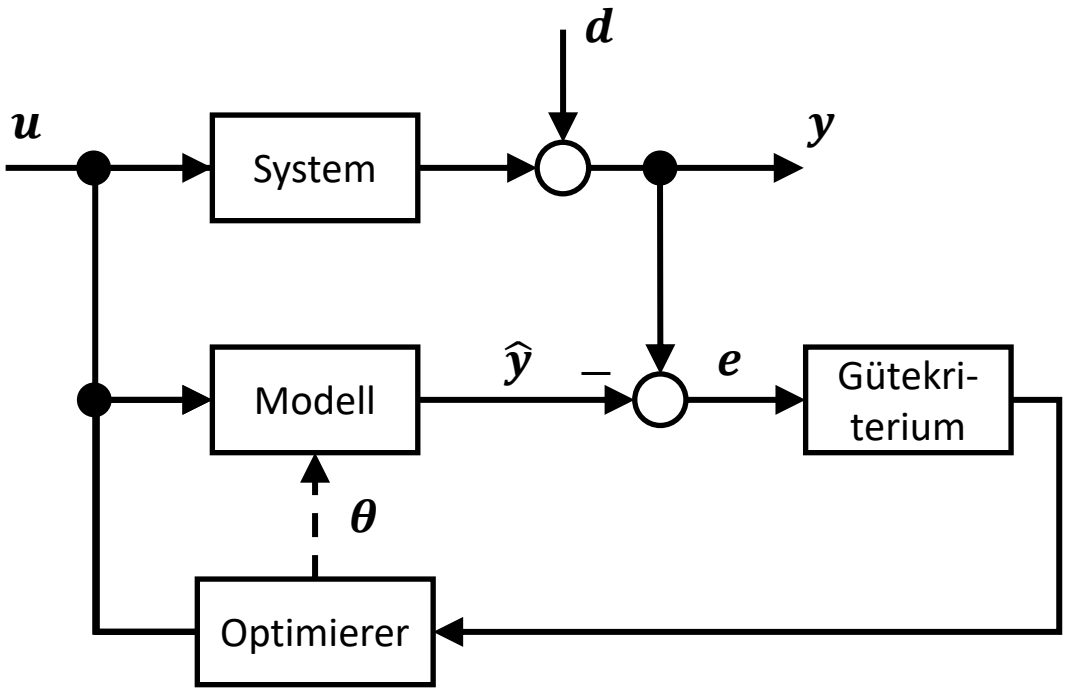
Agenda

EUROPÄISCHE UNION:
Investition in Ihre Zukunft
Europäischer Fonds für regionale Entwicklung

- **Rückblick Modellbildung Teil 1**
  - Ablauf der datengetriebenen Modellbildung
  - Designentscheidungen
  - Datengetriebene Modellbildung mit CasADi
- DIM - Toolbox
- Datengetriebene Modellbildung des Spritzgießprozesses
- Fallstudien

Ablauf der datengetriebenen Modellbildung

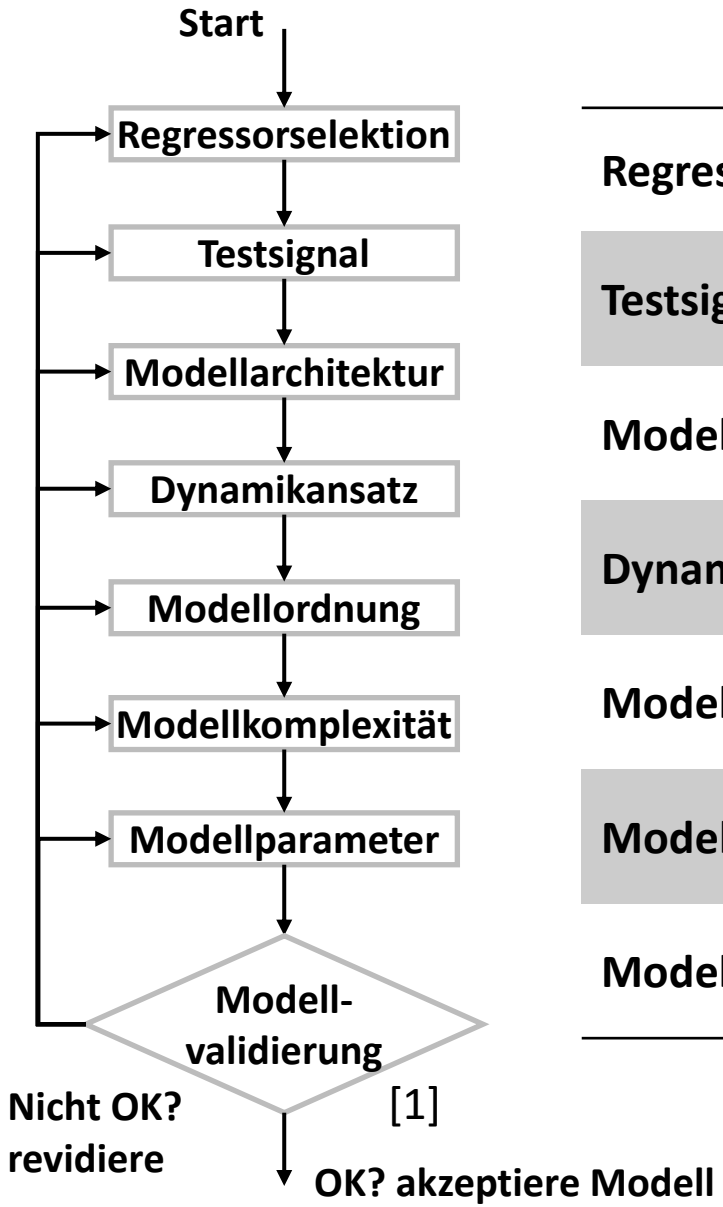
EUROPÄISCHE UNION:
Investition in Ihre Zukunft
Europäischer Fonds für regionale Entwicklung



- System und Modell werden mit geeigneten Testsignalen angeregt
- Der Modellansatz kann aus theoretischen Überlegungen (physikalisch) folgen, muss aber nicht (Black-Box)
- Aus den Ausgangssignalen (System $\boldsymbol{y}$ und Modell $\hat{\boldsymbol{y}}$) wird die Differenz $\boldsymbol{e}$ gebildet und anhand eines Gütekriteriums (Kostenfunktion) bewertet
- Das Ergebnis der Bewertung wird von einem Optimierungsverfahren verwendet, um die Modellparameter anzupassen
- Das Optimierungsverfahren wird beendet, sobald eine gewünschte Güte erreicht oder ein anderes Abbruchkriterium erfüllt ist

Designentscheidungen

EUROPÄISCHE UNION:
Investition in Ihre Zukunft
Europäischer Fonds für regionale Entwicklung



[1]

Regressorselektion	Welche Eingänge sollen als Modelleingänge verwendet werden?
Testsignal	Wie muss das Testsignal gewählt werden, damit das relevante Systemverhalten in den Daten enthalten ist?
Modellarchitektur	Welche Funktion soll zur Approximation des Systems angesetzt werden? Polynom, Multilayer Perceptron, Takagi Sugeno Fuzzy, ...
Dynamikansatz	Soll das System als statisches System oder als dynamisches System mit externer/interner Dynamik approximiert werden?
Modellordnung	Falls ein dynamisches Modell gebildet werden soll, handelt es sich um ein System 1. Ordnung, 2. Ordnung, etc.
Modellkomplexität	Wie flexibel muss die Modellarchitektur gewählt werden, um das deterministische Systemverhalten abbilden zu können?
Modellparameter	Welches Optimierungsverfahren soll eingesetzt werden, um die Modellparameter zu schätzen?

1. Definieren der Modellgleichung als CasADi-Funktion, die Modellparameter müssen als symbolische Variablen definiert werden:

```
theta = cs.MX.sym('theta',2,1)
u = cs.MX.sym('u',2,1)
y = theta[0]*u[0] + theta[1]*u[1]
f_model = cs.Function('f_model',[u,theta],[y],['u','theta'],['y'])
```

2. Auswerten der CasADi-Funktion auf den aufgenommenen Daten in der entsprechenden Konfiguration und Berechnung der "Kosten" basierend auf dem Prädiktionsfehler $\hat{y}^i - y^i$

```
L = 0
for k in range(0,999):
    u_k = arx_data.loc[k][['u_k-1','y_k-1']].values.reshape((2,1))
    y_k = arx_data.loc[k]['y_k']
    y_hat = f_model(u=u_k, theta=theta)['y']
    L = L + 0.5*(y_hat - y_k)**2
```

3. Übergeben der Kosten an einen Optimierer, der die optimalen Modellparameter ermittelt

```
S = cs.qpsol('S', 'qpoases', {'x':theta, 'f':L})
r=S()
theta_opt = r['x']
```

Quelle: Tutorials/Dynamische Systemidentifikation mit CasADi.ipynb

Agenda

EUROPÄISCHE UNION:
Investition in Ihre Zukunft
Europäischer Fonds für regionale Entwicklung

- Rückblick Modellbildung Teil 1
- **DIM - Toolbox**
  - Aufbau der Toolbox
  - Modul model_structures
  - Modul injection_molding
  - Modul param_optim
- Datengetriebene Modellbildung des Spritzgießprozesses
- Fallstudien

Aufbau Toolbox

EUROPÄISCHE UNION:
Investition in Ihre Zukunft
Europäischer Fonds für regionale Entwicklung

```
DigitalTwinInjectionMoulding/
|
├── models/
|   ├── injection_molding.py
|   └── model_structures.py
|
├── optim/
|   ├── common.py
|   └── control_optim.py
|   └── param_optim.py
|   └── hyperparam_optim.py
|
├── tests/
|   ├── MultiStageOptimization.py
|   ├── ...
```

models

- model_structures enthält als Klassen implementierte Modellarchitekturen, wie MLP (statisch und dynamisch), GRU und LSTM.
- injection_molding enthält nur zwei Klassen: ProcessModel und QualityModel. Diese Klassen dienen der Verknüpfung mehrerer Modelle aus model_structures zu einem Prozess- bzw. Qualitätsmodell

models

- control_optim enthält alle Methoden zur optimalsteuerungsbasierten Prozessoptimierung
- param_optim enthält alle Methoden zur Parameteroptimierung
- hyperparam_optim enthält eine partikelschwarmbasierte Methode zur Hyperparameteroptimierung

tests

- Methoden und Skripte für den persönlichen Gebrauch des Entwicklers

model_structures

EUROPÄISCHE UNION:
Investition in Ihre Zukunft
Europäischer Fonds für regionale Entwicklung

```
590     def OneStepPrediction(self,u0,params=None):
591         """
592         OneStepPrediction() evaluates the model equation defined in                and
593         self.Function()                                                            ialized
594
595         self.Function() takes initial state x0, input u0 and all model
596         parameters as input. The model parameters can either be optimization
597         variables themselves (as in system identification) or the take specific
598         values (when the estimated model is used for control)
599
600         Parameters
601         ----------
602         u0 : array-like with dimension [self.dim_u, 1]
603             input
604         params : dictionary, optional
605             params is None: This is the case during model based control,
606             self.Function() is evaluated with the numerical
607             values of the model parameters saved in self.Parameters
608             params is dictionary of opti.variables: During system identification
609             the model parameters are optimization variables themselves, so a
610             dictionary of opti.variables is passed to self.Function()
611
612         Returns
613         -------
614         y : array-like with dimension [self.dim_x, 1]
615             output of the Feedforward Neural Network
616
617         """
618         if params==None:
619             params = self.Parameters
620
621         params_new = []
622
623         for name in  self.Function.name_in():
624             try:                                                                   ne]
625                 params_new.append(params[name])                  # Parameters
626             except:
627                 continue
628
629         y = self.Function(u0,*params_new)                                          ut_names)
630
631         return y
586         self.ParameterInitialization()
```

- __init__() ist die Konstrukturmethode
- Bei Erzeugung einer Instanz der Klasse müssen bestimmte Parameter übergeben werden, um die Modellstruktur (Anzahl Eingänge & Ausgänge, Anzahl Neuronen in verdeckter Schicht) festzulegen.
- __init__() ruft dann automatisch die Initialize() Methode auf, in welcher die Modellgleichungen als CasADi-Funktion generiert werden.
- OneStepPrediction() wertet die Modellgleichungen (CasADi-Funktion) für gegebene Werte der Eingangsgröße aus
- Simulation() ruft OneStepPrediction() in einer Schleife für eine gegebene Trajektorie an Eingangsgrößen auf
- Die letzteren beiden Methoden
  - sind für alle Modellstrukturen identisch (Copy&Paste für eigene Modellstrukturen)
  - müssen vom Nutzer für gewöhnlich nicht verwendet werden. Die Modellauswertung erfolgt mit den Funktionen parallel_mode(), static_mode() und series_parallel_mode

injection_molding

EUROPÄISCHE UNION:
Investition in Ihre Zukunft
Europäischer Fonds für regionale Entwicklung

```
class ProcessModel():
    '''
    Container for the model which estimates the quality of the part given
    trajectories of the process variables
    '''
    def __init__(self,subsystems,name):

    def Initialize(self):

    def Simulation(self,x0,u,params=None,switching_instances=None,**kwargs):

    def ParameterInitialization(self):

    def SetParameters(self,params):


class QualityModel():
    '''
    Container for the model which estimates the quality of the part given
    trajectories of the process variables
    '''
    def __init__(self,subsystems,name):

    def Simulation(self,c0,u,params=None,switching_instances=None):

    def ParameterInitialization(self):

    def SetParameters(self,params):
```

- ProcessModel() und QualityModel() ermöglichen ein dynamisches Prozess- bzw. Qualitätsmodell zu bilden, welches aus unterschiedlichen Submodellen besteht
- So kann das schaltende Verhalten des Spritzgießprozesses nachgebildet werden

Wichtigste Funktionen des Moduls param_optim für den Nutzer

```
inj_model = GRU(...,name='inj')

press_model = GRU(...,name='press')

cool_model = GRU(...,name='cool')

quality_model = QualityModel(subsystems=[inj_model,press_model,cool_model],
                             name='q_model')
```

param_optim

EUROPÄISCHE UNION:
Investition in Ihre Zukunft
Europäischer Fonds für regionale Entwicklung

```python
def ModelTraining(model,data_train,data_val,initializations=10, BFR=False,
                  p_opts=None, s_opts=None,mode='parallel'):
                    ...

def ParallelModelTraining(model,data_train,data_val,initializations=10,
                          BFR=False, p_opts=None, s_opts=None,mode='parallel',
                          n_pool=5):
                    ...

def parallel_mode(model,data,params=None):
                    ...

def static_mode(model,data,params=None):
                    ...

def series_parallel_mode(model,data,params=None):
                    ...
```

- **ModelTraining()** optimiert die Parameter eines Modells (model_structure oder injection_molding).
- Zur Parameterschätzung werden die Daten in data_train verwendet. Zur Modellvalidierung die Daten in data_val.
- Der Nutzer bestimmt, wie viele zufällige Initialisierungen mit anschließender Optimierung durchgeführt werden sollen.
- p_opts und s_opts sind Argumente, die an CasADi bzw. den Optimierer IPOPT übergeben werden können.

Wichtigste Funktionen des Moduls param_optim für den Nutzer

- Außerdem muss festgelegt werden, in welcher Konfiguration das Modell trainiert werden soll:
  - 'parallel': Modell wird auf den Simulationsfehler optimiert
  - 'series': Modell wird als Einschrittprädiktor trainiert
  - 'static': Statische Modellauswertung
- **ParallelModelTraining()** nutzt mehrere Kerne gleichzeitig, um mehrere Modell auf einmal zu trainieren. Die Anzahl an Kernen wird durch n_pool festgelegt
- **parallel_mode()**, **series_parallel_mode()** und **static_mode()** werten ein Modell in der jeweiligen Konfiguration auf gegebenen Daten aus

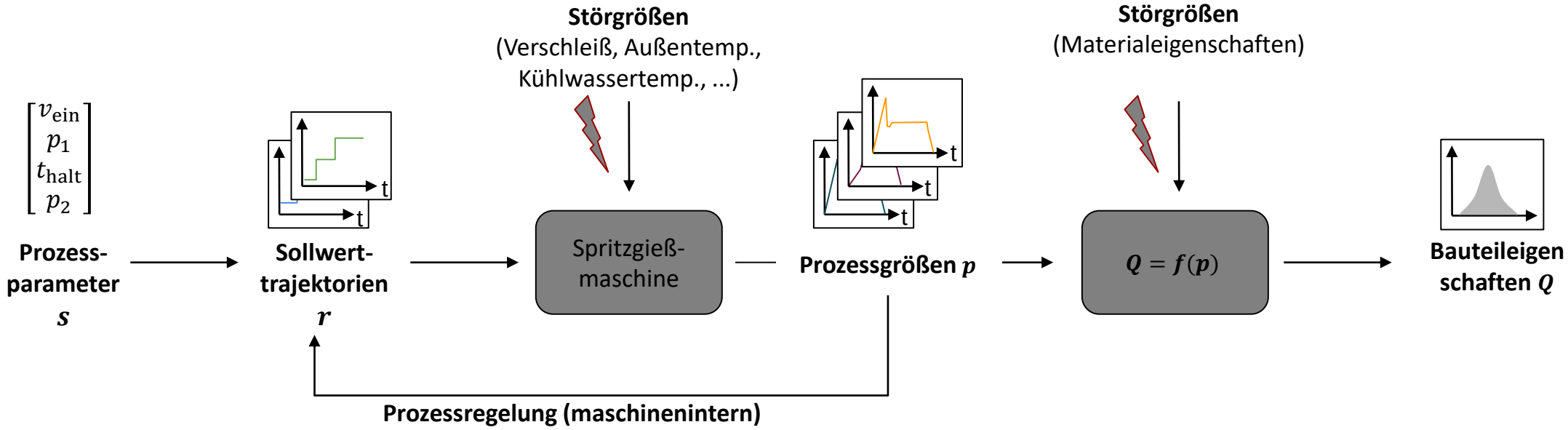
Agenda

EUROPÄISCHE UNION:
Investition in Ihre Zukunft
Europäischer Fonds für regionale Entwicklung

- Rückblick Modellbildung Teil 1
- DIM - Toolbox
- **Datengetriebene Modellbildung des Spritzgießprozesses**
  - Überblick Modellierungsansätze
  - Statische Setpoint Modellierung
  - Statische Feature Modellierung
  - Dynamische Modellierung
- Fallstudien

Modellierungsansätze

EUROPÄISCHE UNION:
Investition in Ihre Zukunft
Europäischer Fonds für regionale Entwicklung

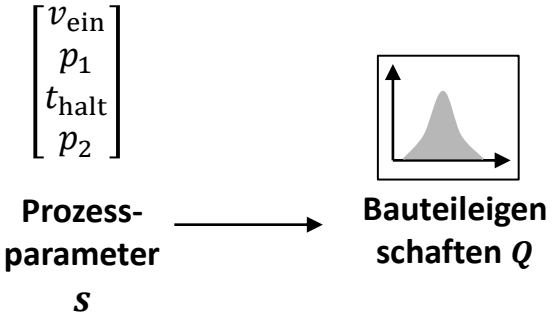


Modellierungsansätze

1. Statische Setpoint-Modelle: Direkte Abbildung von Prozessparametern (Setpoints) $\boldsymbol{s}$ auf die Qualitätsmerkmale $\boldsymbol{Q}$
2. Statische Feature-Modelle: Extraktion von Features $\boldsymbol{f}$ aus Prozessgrößen $\boldsymbol{p}$ basierend auf Expertenwissen. Dann Abbildung der Prozessparameter $\boldsymbol{s}$ und Features $\boldsymbol{f}$ auf die Qualitätsmerkmale $\boldsymbol{Q}$
3. Dynamische Modelle: Dynamische Abbildung der Sollwerttrajektorien $\boldsymbol{r}$ auf die Prozessgrößentrajektorien $\boldsymbol{p}$. Anschließende dynamische Abbildung der Prozessgrößentrajektorien $\boldsymbol{p}$ auf die Qualitätsmerkmale $\boldsymbol{Q}$.

Statische Setpoint-Modelle

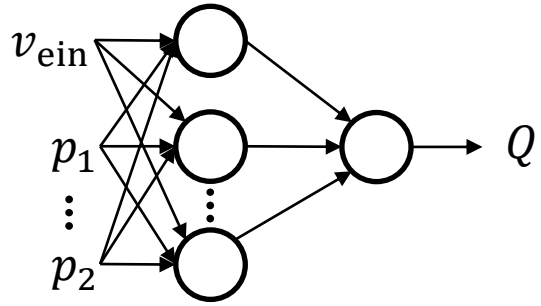
EUROPÄISCHE UNION:
Investition in Ihre Zukunft
Europäischer Fonds für regionale Entwicklung



- Realisierung bspw. mit polynomialem Modell oder Neuronalem Netz (NN)

$$Q = a_0 + a_1 \cdot v_{ein} + b_1 \cdot p_1 + \cdots + d_1 \cdot p_2 + a_2 \cdot v_{ein}^2 + b_2 \cdot p_1^2 + \cdots + d_2 \cdot p_2^2$$

Polynom 2. Grades ohne Interaktionsterme



NN mit beliebig vielen Neuronen in der verdeckten Schicht

Vorteile

- Am einfachsten zu implementierender Modellierungsansatz
- Falls Modell linear in den Parametern (LiP) ist, ist auch Optimierungsproblem LiP, d.h. optimale Modellparameter können in einem Schritt ermittelt werden
- Nur ein Modell für Modellierung des gesamten Spritzgießprozesses erforderlich

Nachteile

- Modellansatz mit den stärksten Vereinfachungen: Tatsächlicher Verlauf der Prozessgrößen wird bei Modellierung der Qualität nicht berücksichtigt
- Modell ist daher nur im stationären Betrieb gültig
- D.h. im transienten Betrieb (z.b. nach Änderung der Prozessparameter) kann die Bauteilqualität nicht zuverlässig prädiziert werden
- Wirkung von Störgrößen bleibt unbemerkt

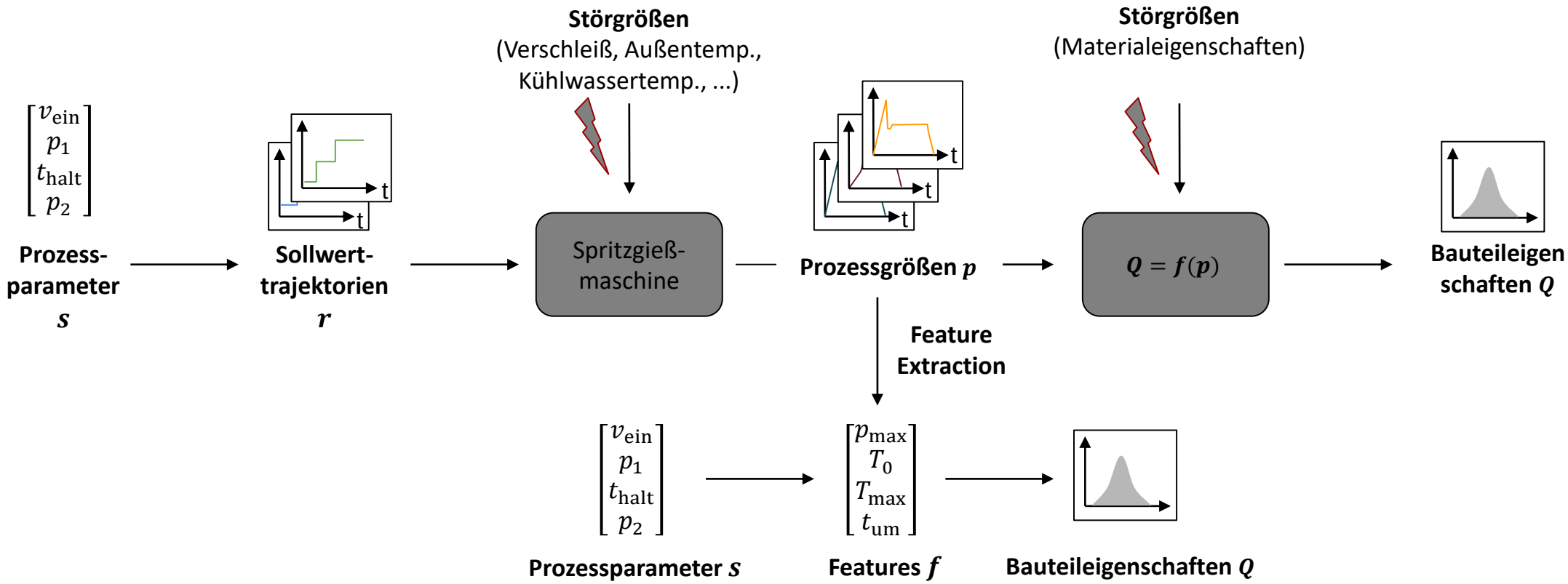
Designentscheidungen

EUROPÄISCHE UNION:
Investition in Ihre Zukunft
Europäischer Fonds für regionale Entwicklung

Designentscheidung	Bemerkung
Regressorselektion	• Als Regressoren kommen alle Maschinenparameter in Frage, welche im Betrieb vom Anlagenführer variiert werden.
Testsignal	• Nicht ganz zutreffend, da ein statisches Modell gebildet wird. • Für die Erhebung von Trainingsdaten sollte idealerweise ein Versuchsplan durchgeführt werden, welcher den relevanten Betriebsbereich abdeckt. • Nur Beobachtungen nach Abklingen der Einschwingvorgänge sollten zum Modelltraining verwendet werden.
Modellarchitektur	• Die Güte linearer Modelle war nicht zufriedenstellend • Polynome führten zu Modellen mit höchster Güte • Multilayer Perceptron (MLP) führten zu zufriedenstellenden Modellen
Dynamikansatz	-
Modellordnung	-
Modellkomplexität	• Polynome fünfter Ordnung und MLP mit mindestens vier Neuronen in der verdeckten Schicht führten zu Modellen mit guter Modellgüte

Statische Feature-Modelle

EUROPÄISCHE UNION:
Investition in Ihre Zukunft
Europäischer Fonds für regionale Entwicklung

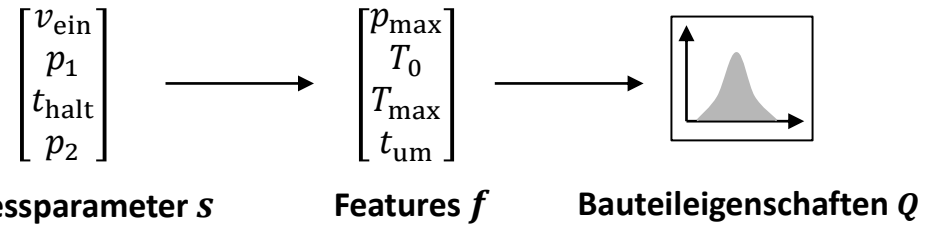


- Umgehung der Notwendigkeit der Bildung dynamischer Modelle, bei gleichzeitiger Ausnutzung dynamischer Information basierend auf Expertenwissen

Statische Feature-Modelle

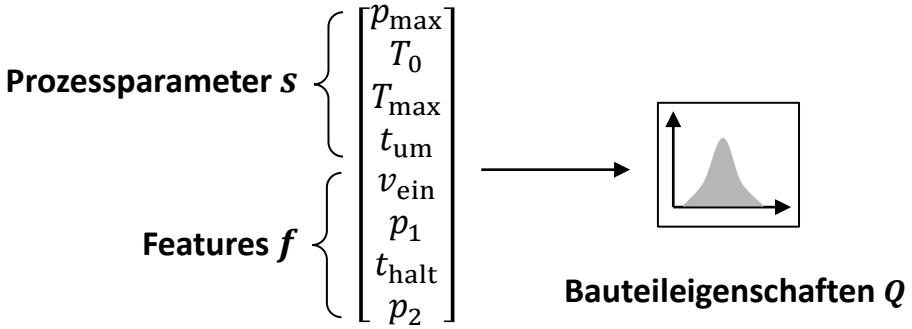
EUROPÄISCHE UNION:
Investition in Ihre Zukunft
Europäischer Fonds für regionale Entwicklung

Variante 1



- Für die Abbildungen $s \rightarrow \hat{f}$ und $f \rightarrow \hat{Q}$ konnten jeweils sehr gute geschätzt werden
- Der akkumulierte Fehler bei der Abbildung $s \rightarrow \hat{f} \rightarrow \hat{Q}$ war dann aber in derselben Größenordnung, wie bei der direkten statischen Modellbildung $s \rightarrow \hat{Q}$

Variante 2

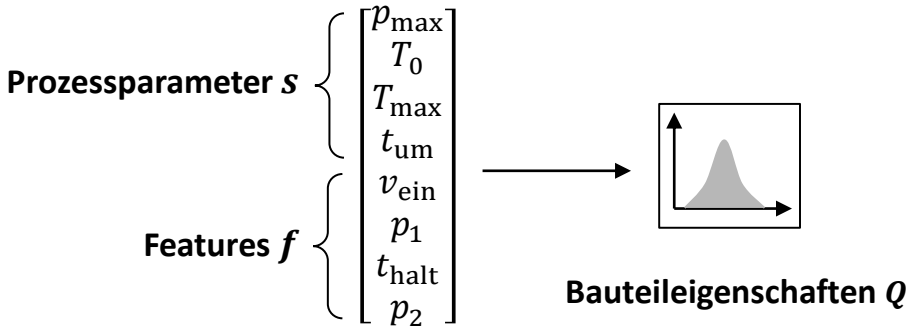


- Aus diesem Grund wurden Setpoints s und Features f zu einem gemeinsamen Vektor $\begin{bmatrix} s \\ f \end{bmatrix}$ konkateniert
- Der konkatenierte Vektor bildet die Eingangsgröße für ein statisches Modell, welches direkt auf die Bauteileigenschaften Q abbildet
- Die hierdurch erreichbaren Modellgüten übertrafen sowohl den Modellierungsansatz $s \rightarrow \hat{Q}$ als auch $s \rightarrow \hat{f} \rightarrow \hat{Q}$

- Grund: Im Gegensatz zu dem statischen Setpoint-Modell $s \rightarrow \hat{Q}$ werden durch die Verwendung von Features bspw. unterschiedliche Anfangsbedingungen berücksichtigt. Im Gegensatz zu $s \rightarrow \hat{f} \rightarrow \hat{Q}$ kommt zu keiner Fehlerakkumulation.

Statische Feature-Modelle

EUROPÄISCHE UNION:
Investition in Ihre Zukunft
Europäischer Fonds für regionale Entwicklung



- Prozessmodell: $s \rightarrow f$ (statisch)
- Qualitätsmodell: $f \rightarrow Q$ (statisch)
- Realisierung der Modelle bspw. mit polynomialen Modellen oder Neuronalen Netzen (NN)

Vorteile

- Relativ einfach zu implementieren
- Falls Modelle (LiP) sind, ist auch Optimierungsproblem LiP
- Guter Kompromiss zwischen Ausnutzung vorhandener (dynamischer) Informationen und Komplexität der Modellbildung
- Modelle bestenfalls auch im transienten Betrieb gültig

Nachteile

- Expertise zur Auswahl aussagekräftiger Features erforderlich
- Keine Ausnutzung der "gesamten" dynamischen Information (d.h. der Prozessgrößentrajektorien)

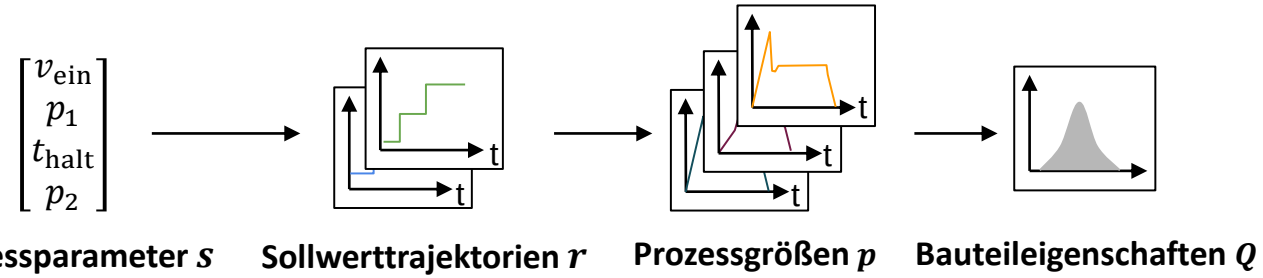
Designentscheidungen

EUROPÄISCHE UNION:
Investition in Ihre Zukunft
Europäischer Fonds für regionale Entwicklung

Designentscheidung	Bemerkung
Regressorselektion	• Prozessmodell: Kandidaten sind Maschinenparameter und auf Expertenwissen basierende Features
Testsignal	• Idealerweise Durchführung eines Versuchsplans, welcher den relevanten Betriebsbereich abdeckt. • Features sind abhängige Größen, welche nicht beliebig manipuliert werden können. Daher keine Möglichkeit eine gleichmäßige Abdeckung des "Feature-Raums" zu gewährleisten.
Modellarchitektur	• Polynom neigt zu oszillierender Interpolation, da Feature-Raum ungleichmäßig abgedeckt ist --> MLP bessere Variante
Dynamikansatz	-
Modellordnung	-
Modellkomplexität	• Polynome dritter Ordnung und MLP mit mindestens drei Neuronen in der verdeckten Schicht führten bereits zu Modellen mit guter Modellgüte

Dynamische Modelle

EUROPÄISCHE UNION:
Investition in Ihre Zukunft
Europäischer Fonds für regionale Entwicklung



Prozessparameter s **Sollwerttrajektorien r** **Prozessgrößen p** **Bauteileigenschaften Q**

- Umrechnung $\boldsymbol{s} \rightarrow \boldsymbol{r}$ bekannt (statisch)
- Prozessmodell: $\boldsymbol{r} \rightarrow \boldsymbol{p}$ (dynamisch)
- Qualitätsmodell: $\boldsymbol{p} \rightarrow \boldsymbol{Q}$ (dynamisch)

Vorteile

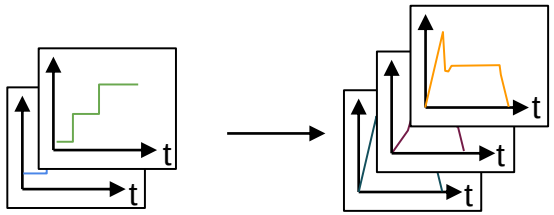
- Keine Feature-Selektion erforderlich
- Ausnutzung der gesamten dynamischen Information durch Verwendung der Prozessgrößentrajektorien
- Prozessmodell und Qualitätsmodell im transienten Betrieb gültig

Nachteile

- Sehr aufwändige Implementierung
- Sehr hohe Komplexität der Modellbildung
- Optimierungsprobleme sind immer nichtlinear, selbst wenn Modelle LiP sind

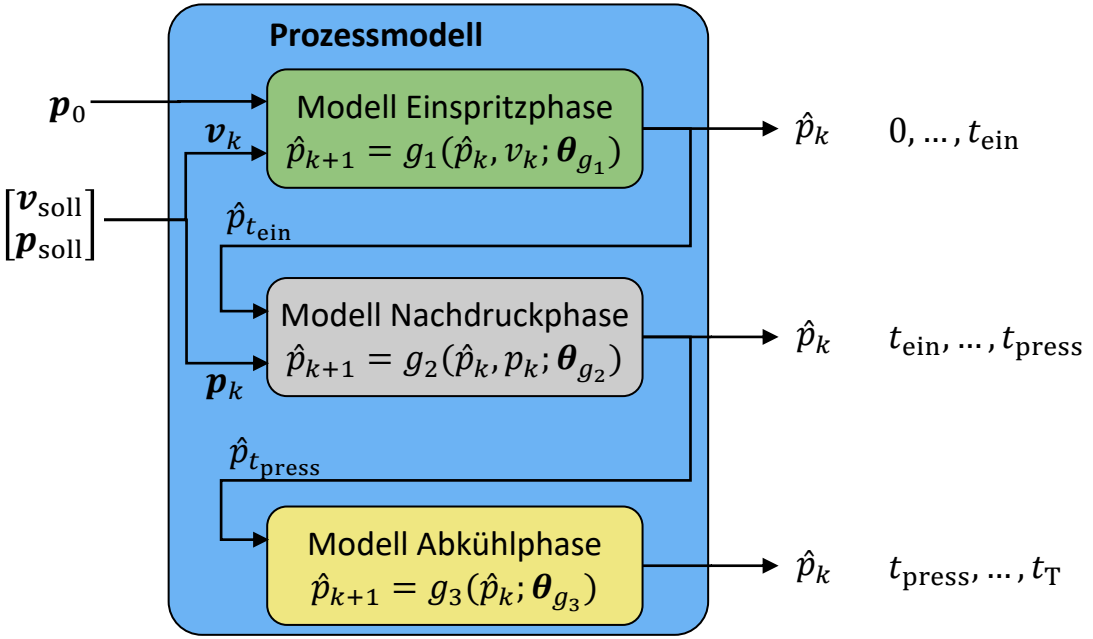
Dynamische Modellbildung - Prozessmodell

EUROPÄISCHE UNION:
Investition in Ihre Zukunft
Europäischer Fonds für regionale Entwicklung



Sollwerttrajektorien r **Prozessgrößen p**

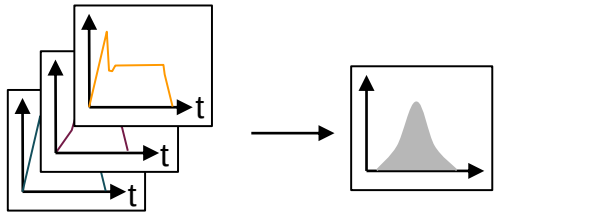
- Der geregelte Spritzgießprozess ist ein schaltendes System:
  1. Einspritzphase ist geschwindigkeitsgeregelt
  2. Nachdruckphase ist druckgeregelt
  3. Abkühlphase ist ein rein autoregressiver Prozess
- Dieses Verhalten muss im Modellansatz berücksichtigt werden



- Das Prozessmodell besteht aus drei unterschiedlichen Modellen, jeweils für die verschiedenen Phasen des Spritzgießprozesses
- Die Modelle sind an den jeweiligen Umschaltpunkten miteinander verknüpft

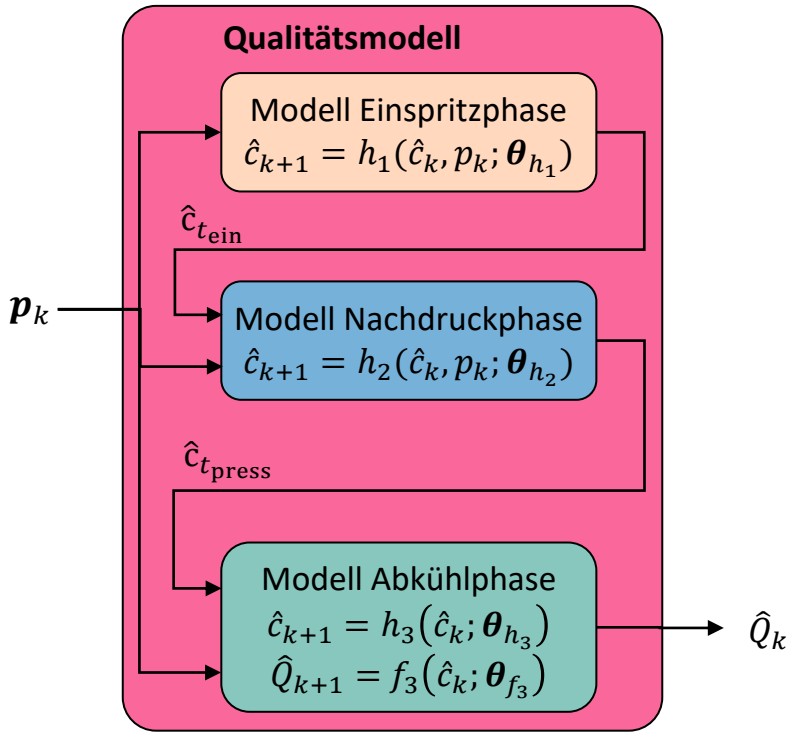
Dynamische Modellbildung - Qualitätsmodell

EUROPÄISCHE UNION:
Investition in Ihre Zukunft
Europäischer Fonds für regionale Entwicklung



Prozessgrößen p **Bauteileigenschaften Q**

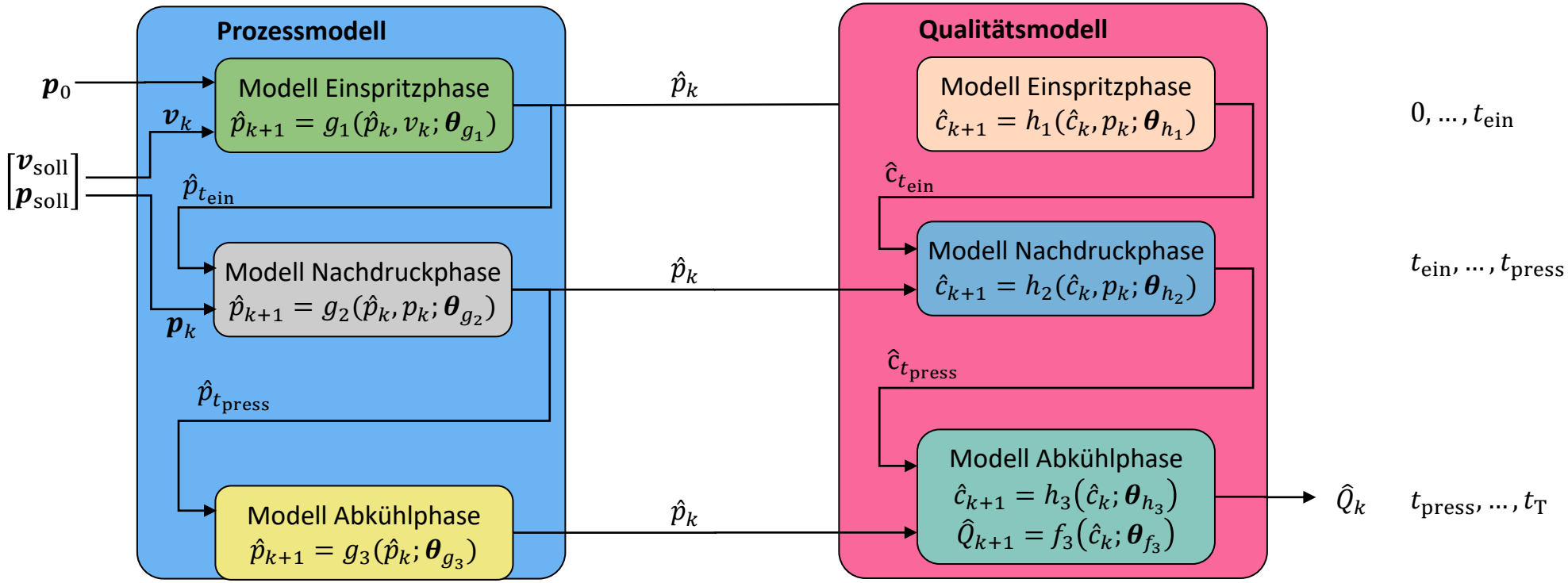
- Das schaltende Verhalten des Spritzgießprozesses, bedeutet nicht, dass auch die Abbildung von Prozessgrößentrajektorien $\boldsymbol{p}$ auf Bauteileigenschaften $\boldsymbol{Q}$ durch ein schaltendes Modell beschrieben werden muss.
- Das Bauteil unterliegt jedoch unter anderem einem Wechsel des Aggregatzustandes, der ungefähr zum Beginn der Nachdruckphase abgeschlossen sein sollte.
  - Es ist daher unwahrscheinlich, dass der Prozess der Ausbildung der Qualitätseigenschaften durch ein einziges Modell mit fixen Parametern beschrieben werden kann.
  - Diese Vermutung wird auch durch Versuche empirisch gestützt
  - Zwei Optionen:
    - Modellparameter ändern sich kontinuierlich. Realitätsnäher, führt zu sehr komplexem Optimierungsproblem
    - Modellparameter ändern sich zu diskreten Zeitpunkten, wie Prozessmodell. Einfacher und hinreichend realitätsnah.



Dynamische Modellbildung - komplett

EUROPÄISCHE UNION:
Investition in Ihre Zukunft
Europäischer Fonds für regionale Entwicklung

- Prozess- und Qualitätsmodell werden in Reihe geschaltet, um die Bauteileigenschaften $\boldsymbol{Q}$ aus den eingestellten Setpoints $\boldsymbol{s} = \begin{bmatrix} \boldsymbol{v}_{\text{soll}} \\ \boldsymbol{p}_{\text{soll}} \end{bmatrix}$ zu prädizieren



Designentscheidungen

EUROPÄISCHE UNION:
Investition in Ihre Zukunft
Europäischer Fonds für regionale Entwicklung

Designentscheidung	Bemerkung
Regressorselektion	• Prozessmodell: Sollwerttrajektorien der Regler • Qualitätsmodell: Kandidaten sind alle erfassten Prozessgößen
Testsignal	• Prozessmodell: Idealerweise Durchführung eines Versuchsplans, welcher den relevanten Betriebsbereich abdeckt. • Qualitätsmodell: Die resultierenden Prozessgrößentrajektorien sind abhängige Größen, die nicht gezielt manipuliert werden können.
Modellarchitektur	• Prozessmodell: Bisher MLP versucht, jedoch noch keine zufriedenstellenden Ergebnisse • Qualitätsmodell: Modell mit interner Dynamik sollte verwendet werden, GRU führt zu besseren Ergebnisse als LSTM
Dynamikansatz	• Prozessmodell: Sowohl Modell mit externer als auch interner Dynamik anwendbar (?) • Qualitätsmodell: Modell mit interner Dynamik empfohlen
Modellordnung	• Prozessmodell: Empfehlung wird ausgesprochen sobald gute Modelle geschätzt werden konnten • Qualitätsmodell: Bereits ein GRU mit einem internen Zustand führt zu guten Ergebnissen
Modellkomplexität	• Prozessmodell: Empfehlung wird ausgesprochen sobald gute Modelle geschätzt werden konnten • Qualitätsmodell: Bereits ein GRU mit einem internen Zustand und zehn verdeckten Neuronen in der Ausgabe führt zu guten Ergebnissen

Agenda

EUROPÄISCHE UNION:
Investition in Ihre Zukunft
Europäischer Fonds für regionale Entwicklung

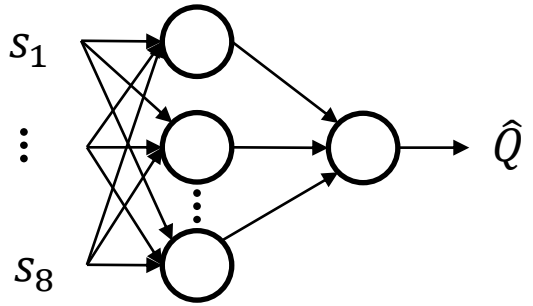
- Rückblick Modellbildung Teil 1
- DIM - Toolbox
- Datengetriebene Modellbildung des Spritzgießprozesses
- **Fallstudien**
  - Statisches Setpoint Qualitätsmodell mit MLP
  - Statisches Feature Qualitätsmodell mit MLP
  - Statisches Prozess- und Qualitätsmodell mit MLP
  - Dynamisches Qualitätsmodell mit GRU

Statisches Qualitätsmodell mit MLP

EUROPÄISCHE UNION:
Investition in Ihre Zukunft
Europäischer Fonds für regionale Entwicklung

Ziel: Bildung eines statischen Modells, welches Maschinenparameter $\boldsymbol{s}$ auf Qualitätsdaten $\boldsymbol{Q}$ abbildet.

Modellansatz: Multilayer Perceptron, implementiert durch die Klasse Static_MLP



Eingangsgrößen $\boldsymbol{u}$: [Düsentemp., Werkzeugtemp., Einsprizgeschw., Umschaltpunkt, Nachdruckhöhe, Nachdruckzeit, Staudruck, Kühlzeit]

Ausgangsgrößen $\boldsymbol{Q}$: [Durchmesser_innen]

Datei: *QualityModel_static_MLP_Toolbox.ipynb*

Feature-Qualitätsmodell mit MLP

EUROPÄISCHE UNION:
Investition in Ihre Zukunft
Europäischer Fonds für regionale Entwicklung

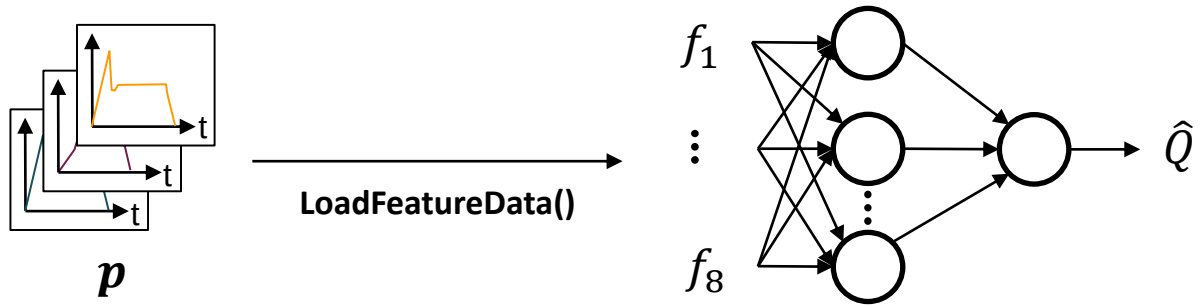
Ziel: Bildung eines statischen Modells, welches aus Prozessgrößen $\boldsymbol{p}$ extrahierte Features $\boldsymbol{f}$ auf Qualitätsdaten $\boldsymbol{Q}$ abbildet.

Modellansatz: Multilayer Perceptron, implementiert durch die Klasse Static_MLP

Eingangsgrößen $\boldsymbol{u}$: [max. Werkzeuginnendruck., Zeitpunkt des Auftretens des max. Werkzeuginnendrucks, Zeitpunkt des Auftretens der max. Werkzeuginnentemp.]

Ausgangsgrößen $\boldsymbol{Q}$: [Durchmesser_innen]

Datei: *QualityModel_feature_MLP_Toolbox.ipynb*



Prozess- und Feature-Qualitätsmodell mit MLP

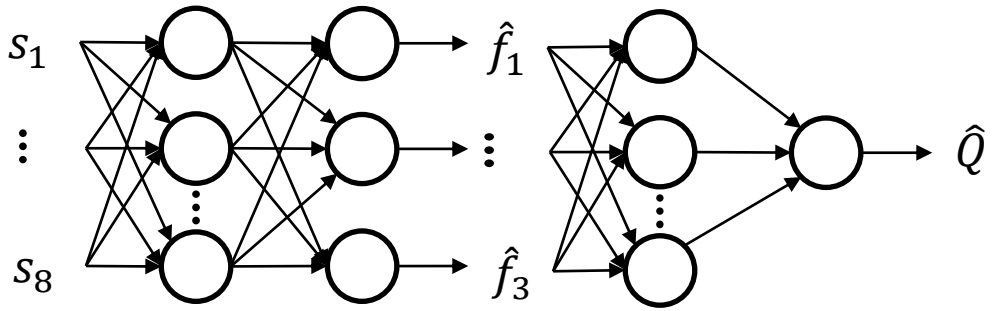
EUROPÄISCHE UNION:
Investition in Ihre Zukunft
Europäischer Fonds für regionale Entwicklung

Ziel: Das Feature-Qualitätsmodell kann die Bauteilqualität genauer vorhersagen, als das statische Qualitätsmodell. Das Ziel ist jedoch, zu ermitteln, wie die Maschine eingestellt werden muss, um eine bestimmte Bauteilqualität zu erzielen. Daher ist es erforderlich den Zusammenhang zwischen eingestellten Maschinenparametern $\boldsymbol{s}$ und resultierenden Features $\boldsymbol{f}$ zu modellieren.

Modellansatz: Multilayer Perceptron, implementiert durch die Klasse Static_MLP

Eingangsgrößen $\boldsymbol{u}$: [max. Werkzeuginnendruck., Zeitpunkt des Auftretens des max. Werkzeuginnendrucks, Zeitpunkt des Auftretens der max. Werkzeuginnentemp.]

Ausgangsgrößen $\boldsymbol{Q}$: [Durchmesser_innen]



Dateien: *ProcessModel_feature_MLP_Toolbox.ipynb*

MLP_Feature_Model_Toolbox.ipynb

Kominiertes statisches Setpoint-Feature Modell

EUROPÄISCHE UNION:
Investition in Ihre Zukunft
Europäischer Fonds für regionale Entwicklung

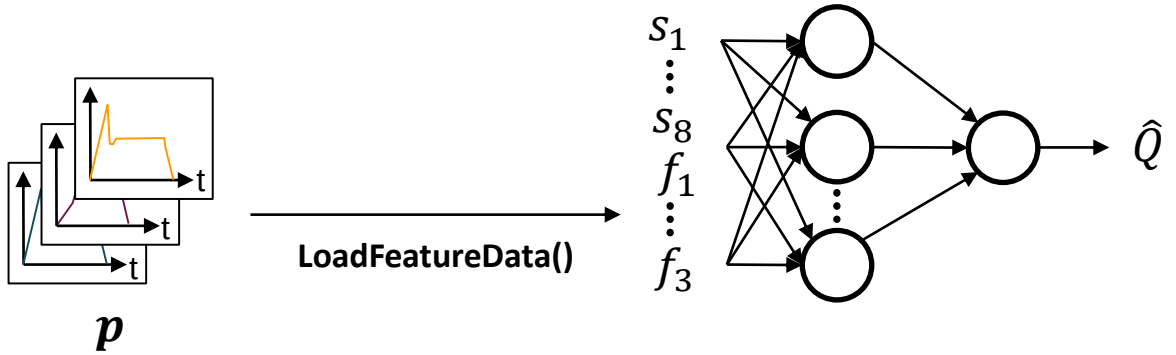
Ziel: Die verkettete Abbildung $\boldsymbol{s} \rightarrow \hat{\boldsymbol{f}} \rightarrow \hat{\boldsymbol{Q}}$ führt zu einer Fehlerakkumulation. Um dies zu vermeiden, werden Setpoints $\boldsymbol{s}$ und Features $\boldsymbol{f}$ zu einem Vektor konkateniert und direkt auf die Bauteilqualität abgegbildet.

Modellansatz: Multilayer Perceptron, implementiert durch die Klasse Static_MLP

Eingangsgrößen $\boldsymbol{u}$: [max. Werkzeuginnendruck., Zeitpunkt des Auftretens des max. Werkzeuginnendrucks, Zeitpunkt des Auftretens der max. Werkzeuginnentemp.]

Ausgangsgrößen $\boldsymbol{Q}$: [Durchmesser_innen]

Datei: *CompleteModel_feature_MLPs_Toolbox.ipynb*



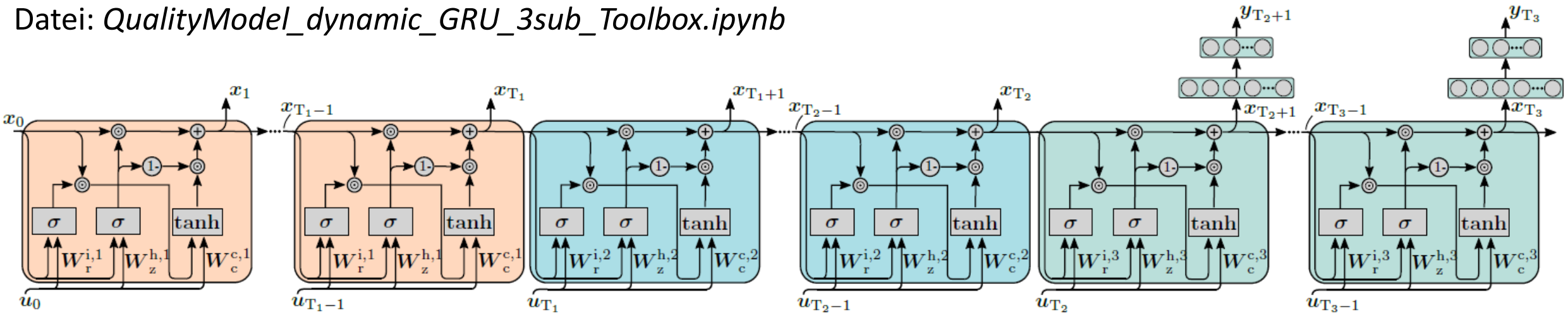
Ziel: Um die komplette in den Trajektorien enthaltende Information zur Prädiktion der Bauteilqualität zu nutzen, sollen diese durch ein dynamisches Modell direct auf die Bauteileigenschaften abgebildet werden.

Modellansatz: 3 Gated Recurrent Units, implementiert durch die Klasse GRU. Verknüpfung der drei Submodellefür die drei Phasen zu einem Qualitätsmodell durch die Klasse QualityModel

Eingangsgrößen: [Werkzeuginnendruck, Werkzeuginnentemperatur]

Ausgangsgrößen: [Durchmesser_innen]

Datei: *QualityModel_dynamic_GRU_3sub_Toolbox.ipynb*



Dynamisches Prozessmodell

EUROPÄISCHE UNION:
Investition in Ihre Zukunft
Europäischer Fonds für regionale Entwicklung

Ziel: Die Prädiktion der Bauteileigenschaften mit dem dynamischen Qualitätsmodell $\boldsymbol{p} \rightarrow \boldsymbol{Q}$ führt zu den besten Ergebnissen. Allerdings soll letztendlich eine modellbasierte Prozessoptimierung bezüglich der einzustellenden Maschinenparameter $\boldsymbol{s}$ stattfinden. Es gilt also ein dynamisches Prozessmodell zu bilden, welches die Sollgrößentrajektorien $\boldsymbol{r}$ auf die Prozessgrößentrajektorien $\boldsymbol{p}$ abbildet . Der Zusammenhang $\boldsymbol{s} \rightarrow \boldsymbol{r}$ ist bekannt und bedarf keiner Modellbildung.

Modellansatz: dynamisch nichtlinear

Eingangsgrößen $\boldsymbol{r}$: [Soll-Einspritzgeschwindigkeit, Soll-Hydraulikdruck]

Ausgangsgrößen $\boldsymbol{p}$: [Schneckenposition, Schneckengeschwindigkeit, Hydraulikdruck, Werkzeuginnendruck, Werkzeuginnentemperatur]

Datei: -