

Bachelorarbeit:
Indizierdruckbasierte Regelung
eines Verbrennungsmotors
basierend auf virtuellen Sensoren

Von:

Kleinpötzl Florian

Malfent Daniel

Betreuer:

Dipl.-Ing. Harald Waschl

Motivation

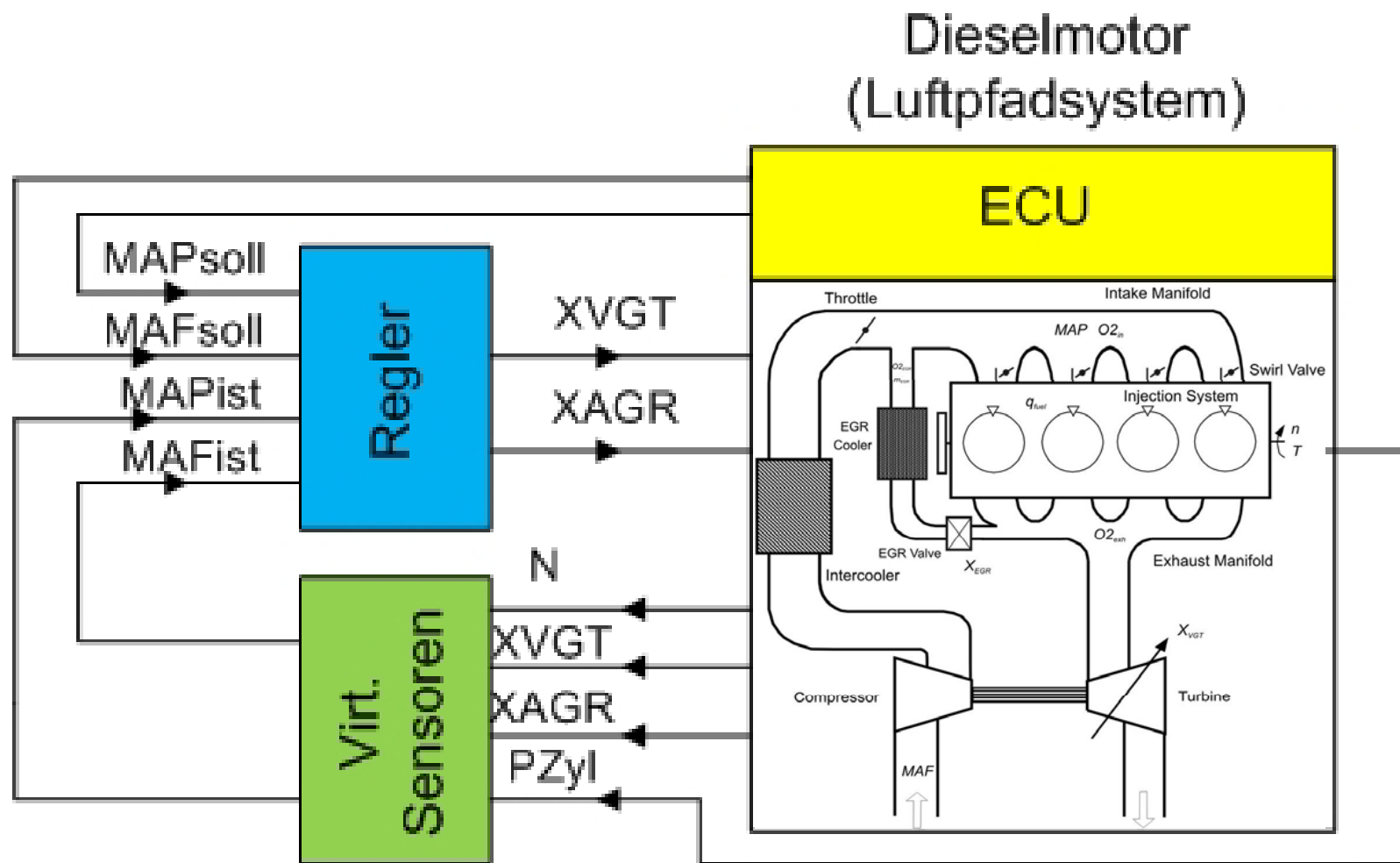
- für die Regelung eines Motors sind eine Vielzahl von Sensoren notwendig
- Sensoren einsparen durch Verwendung virtueller Sensoren
- Reduktion der Kosten
- Kenntnisse für die Bildung der Sensoren aus der Zylinderdruckkurve

Aufgabenstellung

- Entwerfen virtueller Sensoren für den Dieselmotor z.B.:
 - Luftmasse (MAF)
 - Ladedruck (MAP)
- Regelung des Luftsystems mit den virtuellen Sensoren

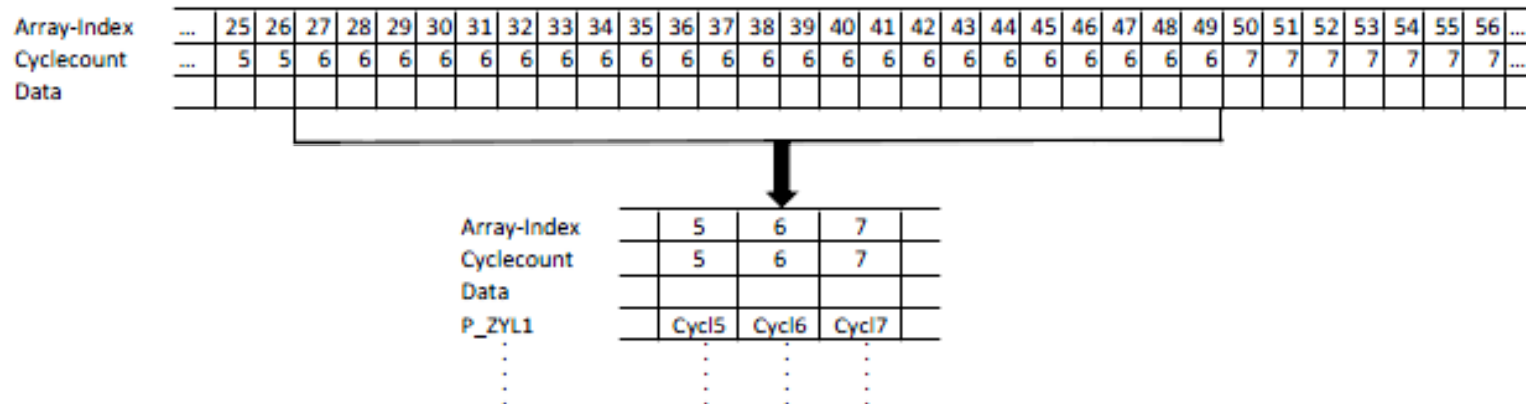


Überblick



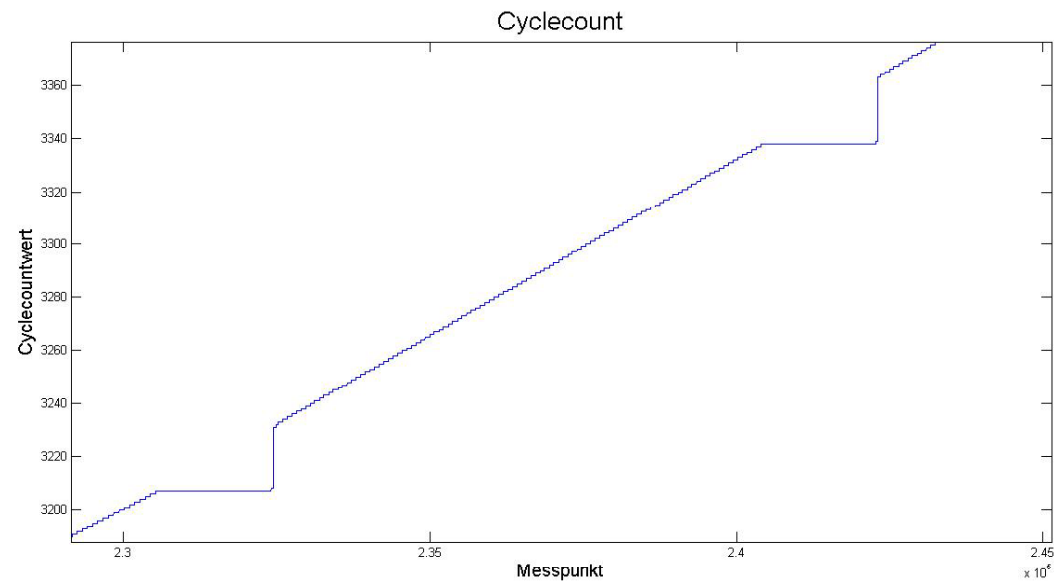
Identifikation/Messdatenaufnahme

- Aufnehmen einer DOE – Messung für die Identifikation
- Vorgehensweise für MAF und MAP identisch
- Synchronisation zwischen Zylinderdruckdaten vom Indicom und den dSpace Messdaten



Identifikation/Messdatenaufnahme

- aufgetretene Probleme:
 - Synchronisieren

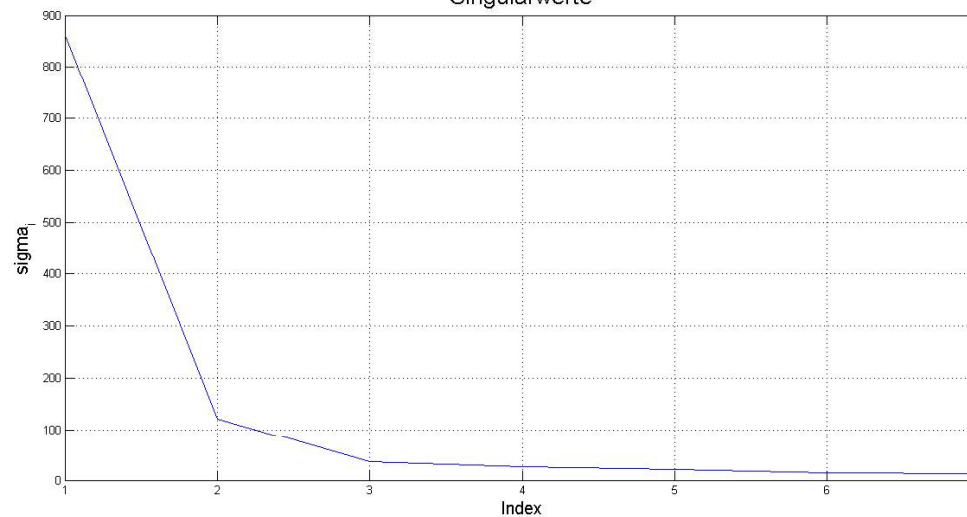


Identifikation/Messdatenaufnahme

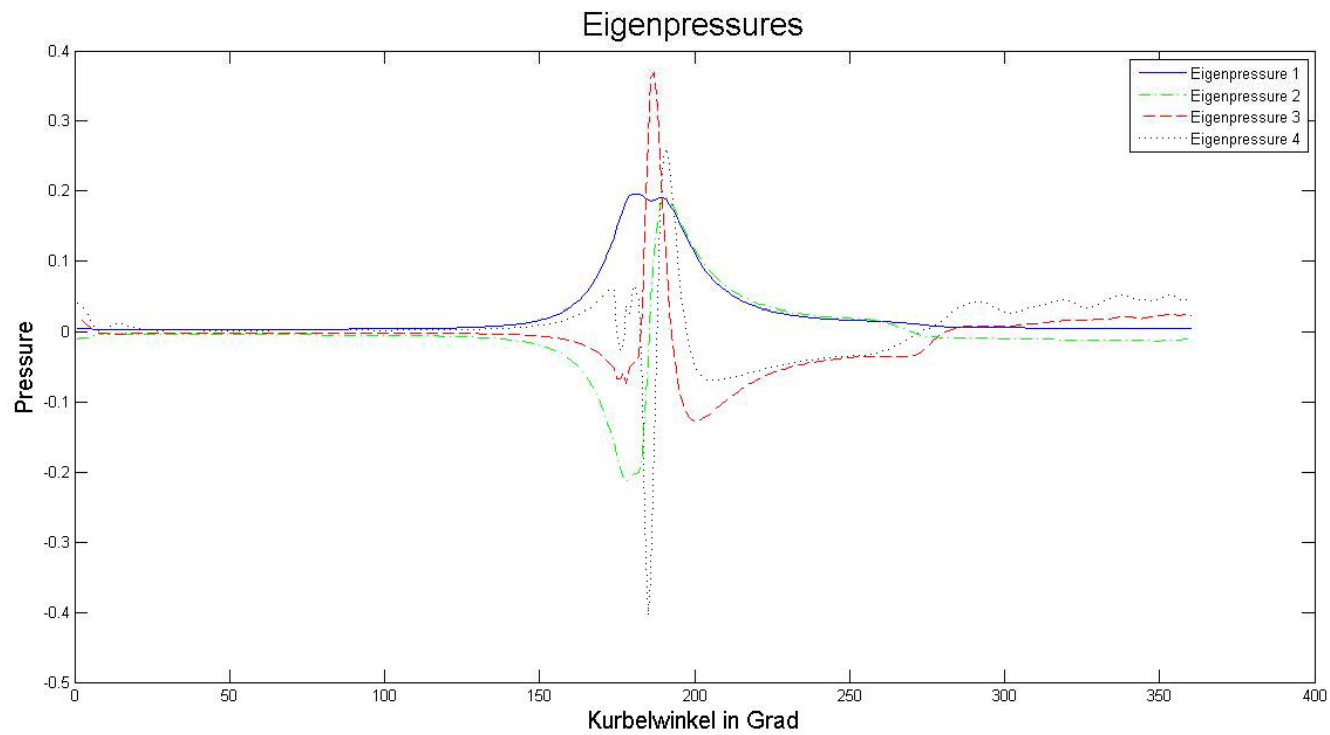
- Anwenden der Principal Component Analysis (PCA) Methode
 - $[U,S,V] = \text{svd}(P_{\text{zyl}})$
 - $PV = US$

$$PV = \begin{bmatrix} p_{\text{rec}1,1} & p_{\text{rec}2,1} & \cdots & p_{\text{rec}n,1} \\ p_{\text{rec}1,2} & p_{\text{rec}2,2} & \cdots & p_{\text{rec}n,2} \\ \vdots & \vdots & \ddots & \vdots \\ p_{\text{rec}1,p} & p_{\text{rec}2,n} & \cdots & p_{\text{rec}n,p} \end{bmatrix}$$

Singularwerte



Identifikation/Messdatenaufnahme

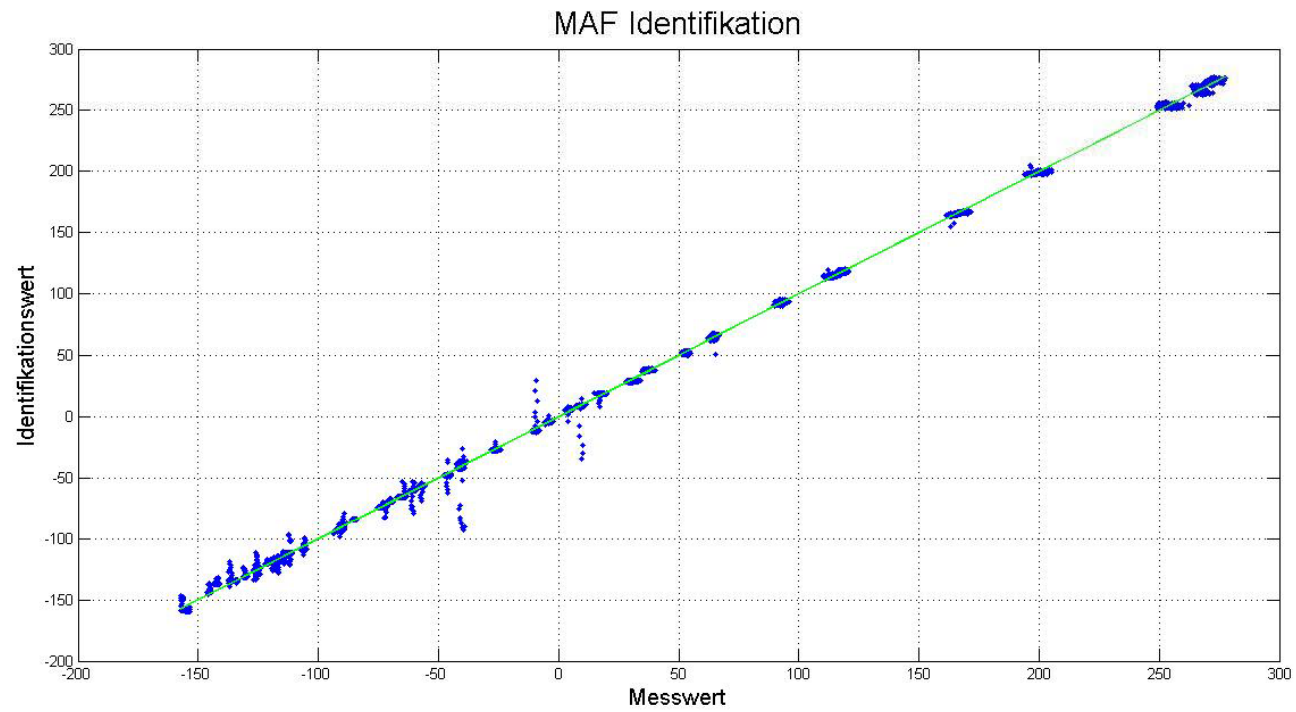


Identifikation/Messdatenaufnahme

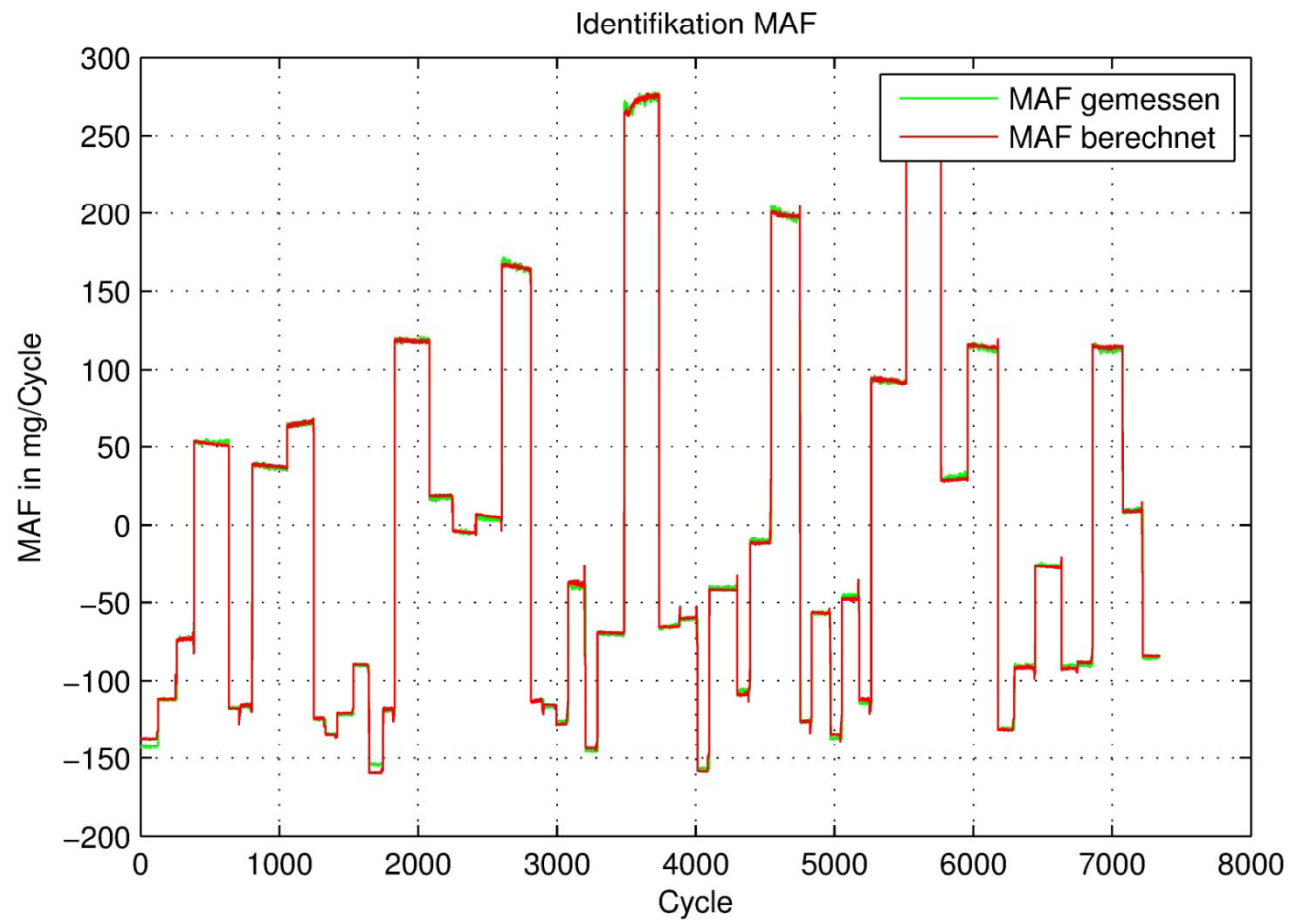
- Identifikation mittels Least Squares Methode
- vollständiger quadratischer Ansatz für Modell

$$\begin{aligned} Y_{MAF} = & +\theta_1 p_{rec1} + \theta_2 p_{rec1} p_{rec1} + \theta_3 p_{rec1} p_{rec2} + \theta_4 p_{rec1} p_{rec3} + \theta_5 p_{rec1} p_{rec4} + \theta_6 p_{rec1} X_{VGT} \\ & + \theta_7 p_{rec1} X_{EGR} + \theta_8 p_{rec1} N + \theta_9 p_{rec2} + \theta_{10} p_{rec2} p_{rec2} + \theta_{11} p_{rec2} p_{rec3} + \theta_{12} p_{rec2} p_{rec4} \\ & + \theta_{13} p_{rec2} X_{VGT} + \theta_{14} p_{rec2} X_{EGR} + \theta_{15} p_{rec2} N + \theta_{16} p_{rec3} + \theta_{17} p_{rec3} p_{rec3} \\ & + \theta_{18} p_{rec3} p_{rec4} + \theta_{19} p_{rec3} X_{VGT} + \theta_{20} p_{rec3} X_{EGR} + \theta_{21} p_{rec3} N + \theta_{22} p_{rec4} \\ & + \theta_{23} p_{rec4} p_{rec4} + \theta_{24} p_{rec4} X_{VGT} + \theta_{25} p_{rec4} X_{EGR} + \theta_{26} p_{rec4} N + \theta_{27} X_{VGT} \\ & + \theta_{28} X_{VGT} X_{VGT} + \theta_{29} X_{VGT} X_{EGR} + \theta_{30} X_{VGT} N + \theta_{31} X_{EGR} + \theta_{32} X_{EGR} X_{EGR} + \theta_{33} X_{EGR} N \\ & + \theta_{34} N + \theta_{35} N N = \Phi_{MAF}^T \Theta_{MAF} \end{aligned}$$

MAF Sensor

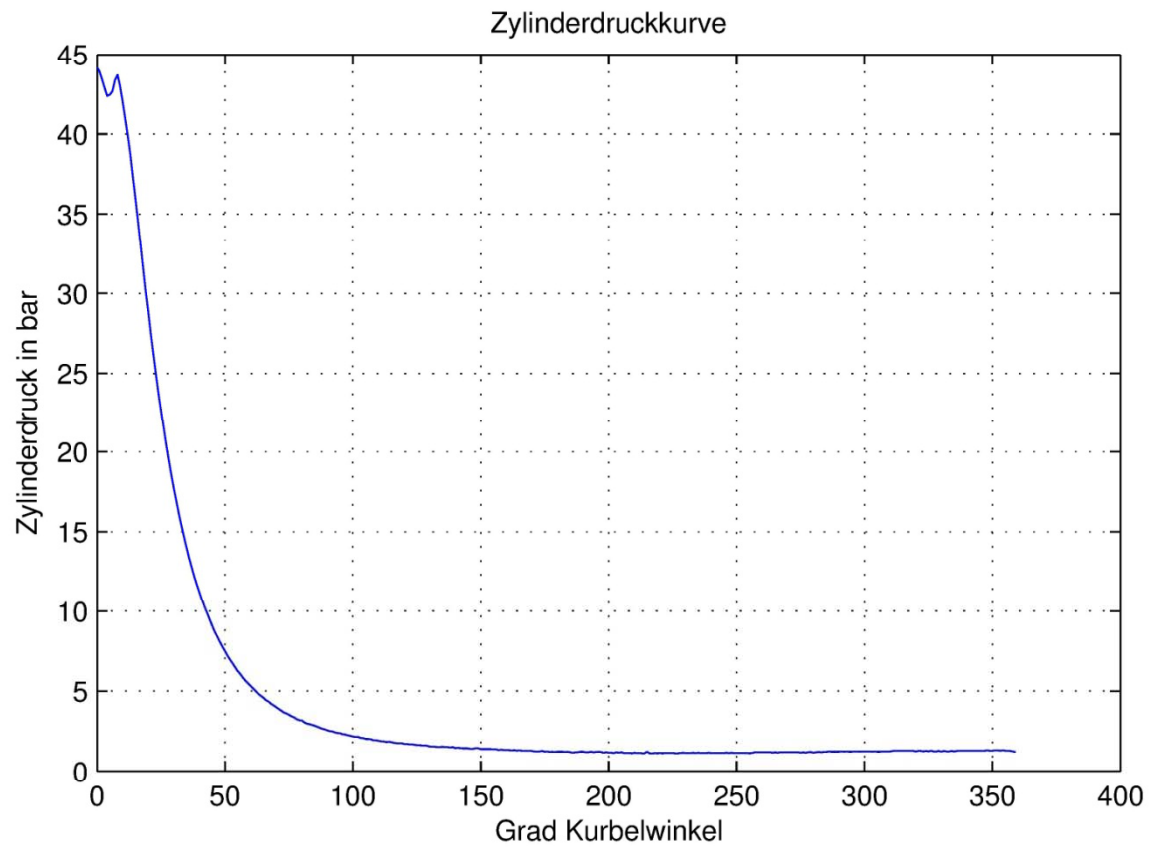


MAF Sensor

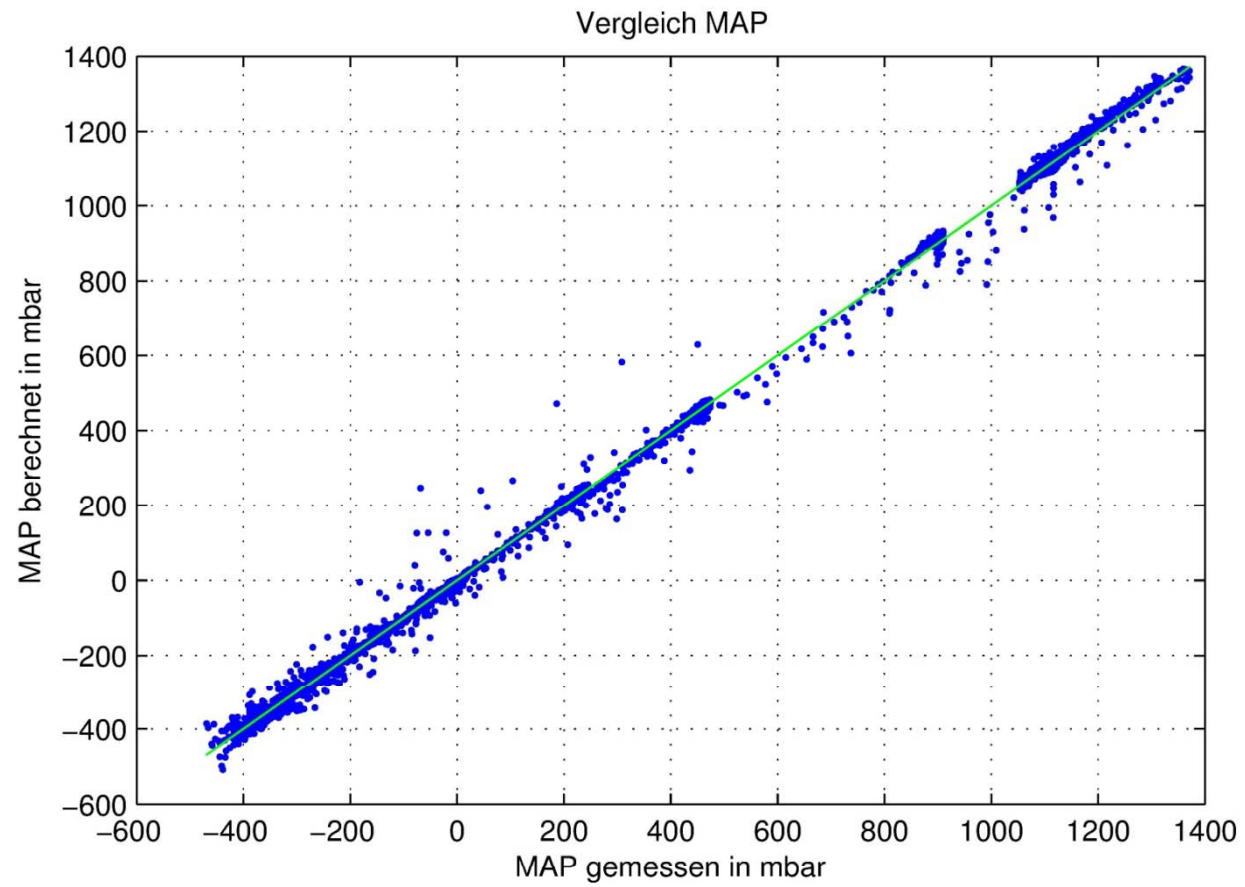


MAP Sensor

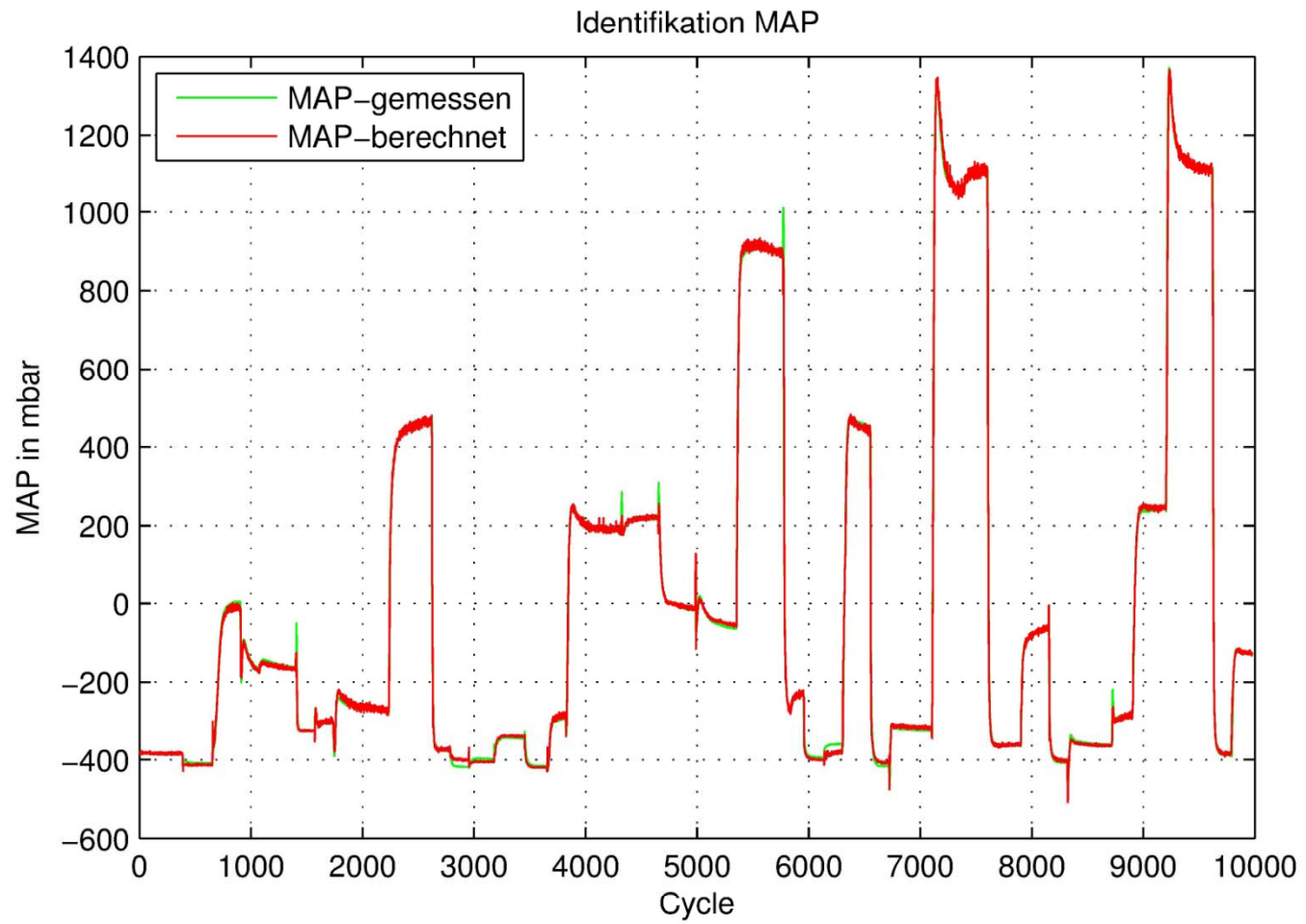
- anderer Bereich der Zylinderdruckkurve



MAP Sensor



MAP Sensor



Implementierung der Onlineberechnung

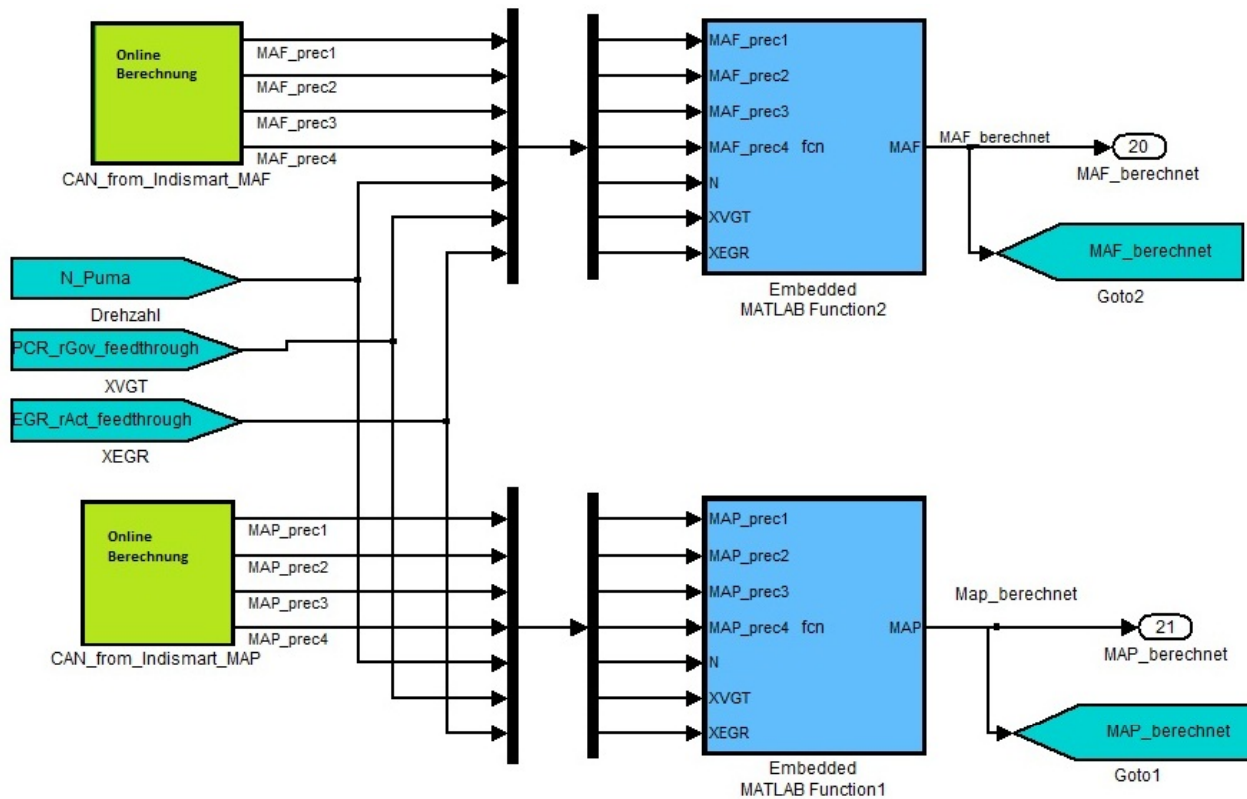
- Features werden aus der aktuellen Zylinderdruckkurve berechnet

```
1 Berechnung Feature 1 MAP
2
3 V1 = {0.005556, 0.005378, 0.005108, 0.004786, ..., 0.004485}
4
5 P1 = 0
6
7 for z=1 to 360
8
9 Pzyl_mean = {1.897130, 1.829926, 1.744634, ..., 1.904429}
10
11 P1 = P1+V1.y[z]*(CA'P_ZYL2.y[z*4-3]-Pzyl_mean.y[z])
12
13 next z
14
15 MafPrec1=P1
```

```
1 Berechnung Feature 1 MAP
2
3 V1 = {0.150633, 0.150375, 0.150175, 0.150139, 0.150457, ..., 0.005647}
4
5 P1 = 0
6
7 for z=1 to 360
8
9 Pzyl_mean = {1.897130, 1.829926, 1.744634, 1.642483, ..., 1.939496}
10
11 P1 = P1 + V1.y[z]*(CA'P_ZYL2.y[z*2+720]-Pzyl_mean.y[z])
12
13 next z
14
15 MapPrec1 = P1
```

Implementierung der Onlineberechnung

- berechnete Features werden ins dSpace geschickt

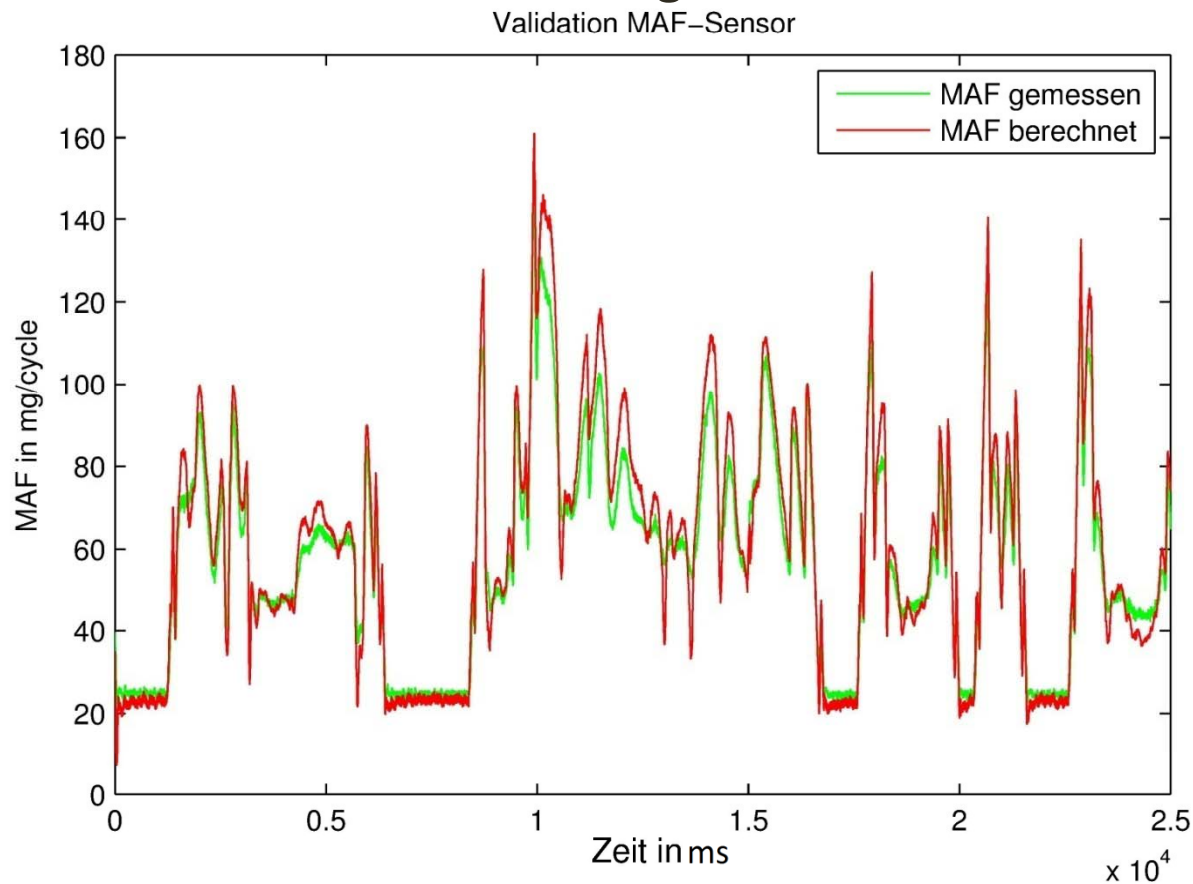


Implementierung der Onlineberechnung

```
1 function MAP = fcn (MAP_prec1, MAP_prec2, MAP_prec3, MAP_prec4, N, XVGT, XEGR)
2
3 P1 = MAP_prec1;
4 P2 = MAP_prec2;
5 P3 = MAP_prec3;
6 P4 = MAP_prec4;
7 MAP_mean = 1422.65481567273;
8
9 Theta = [-478.1634;0.9681682;2.461883;-3.807275;-0.9052705;0.1808627; ...;0.005277437];
10
11 M=[P1 P2 P3 P4 N XVGT XEGR];
12
13 k=1;
14 Q=eye(1,28);
15
16 for i=1:size(M,2)
17     for j=i:size(M,2)
18         Q(:,k)=M(:,i).*M(:,j);
19         k=k+1;
20     end
21 end
22
23 Phi = [1 M Q];
24
25 MAP = Phi*Theta + MAP_mean;
```

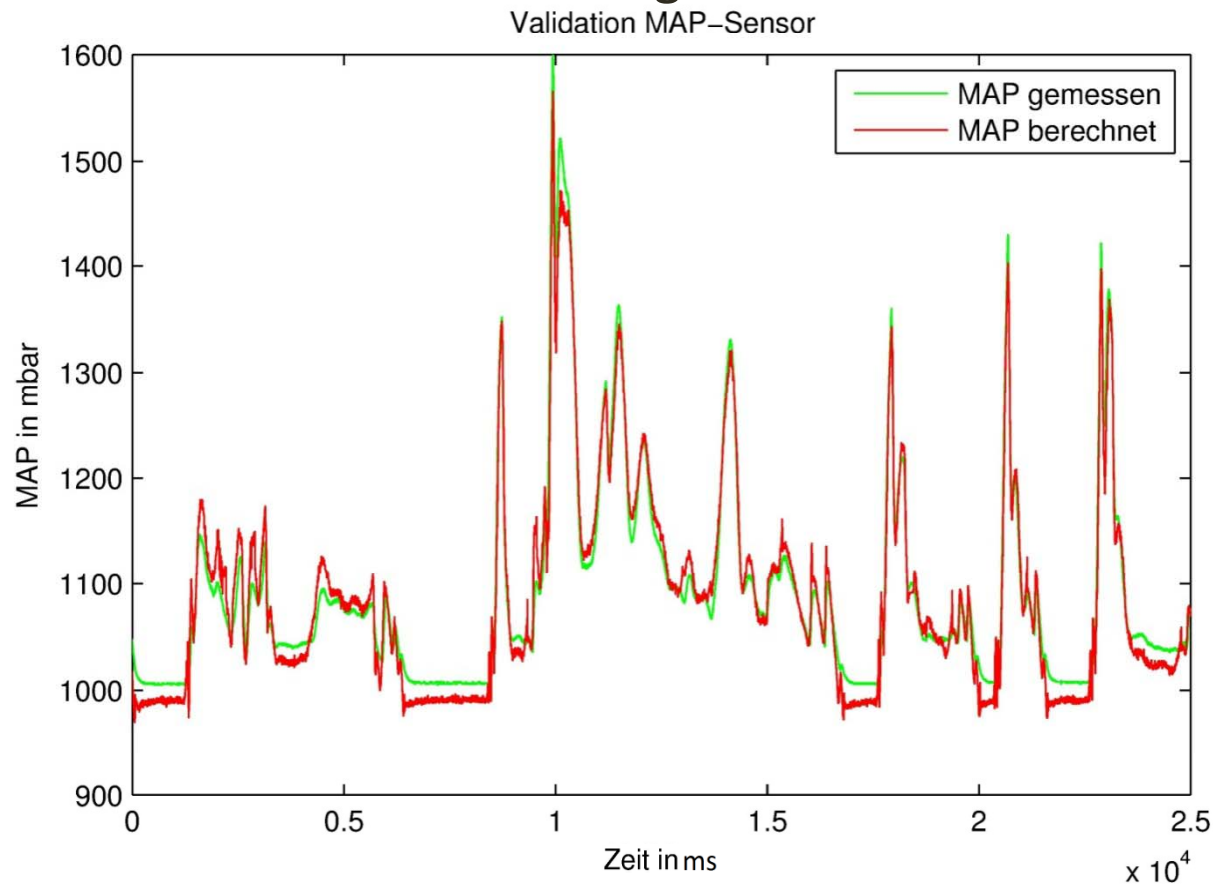
Implementierung der Onlineberechnung

- maximale Abweichung 26%
- durchschnittliche Abweichung 9%



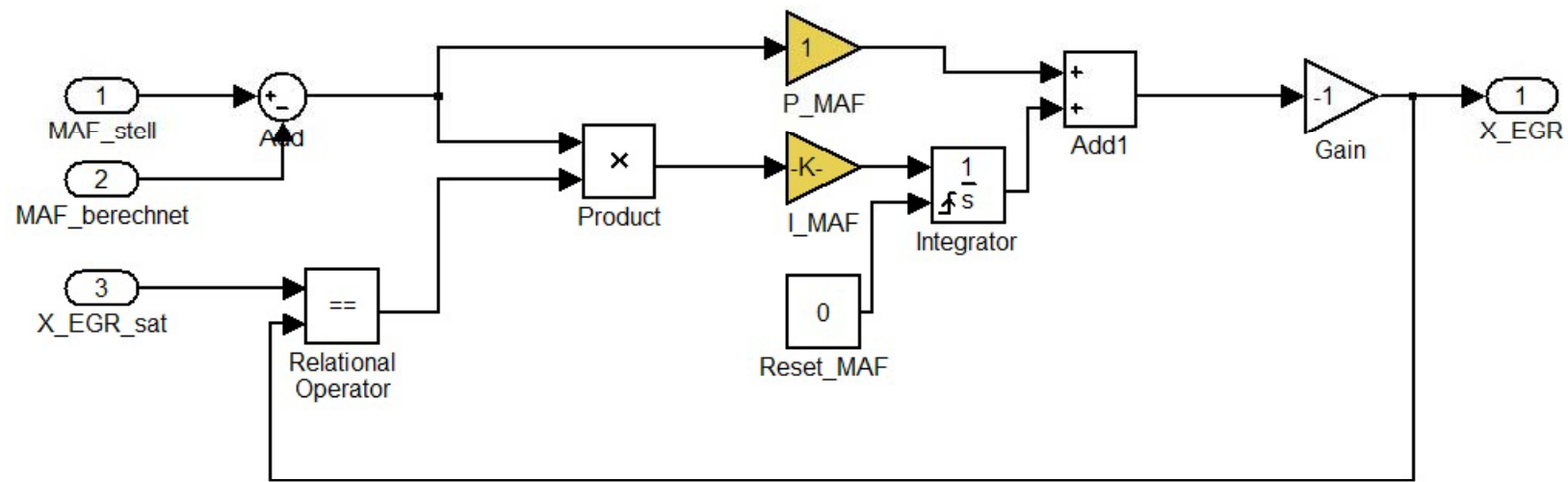
Implementierung der Onlineberechnung

- maximale Abweichung 8%
- durchschnittliche Abweichung 1.5%

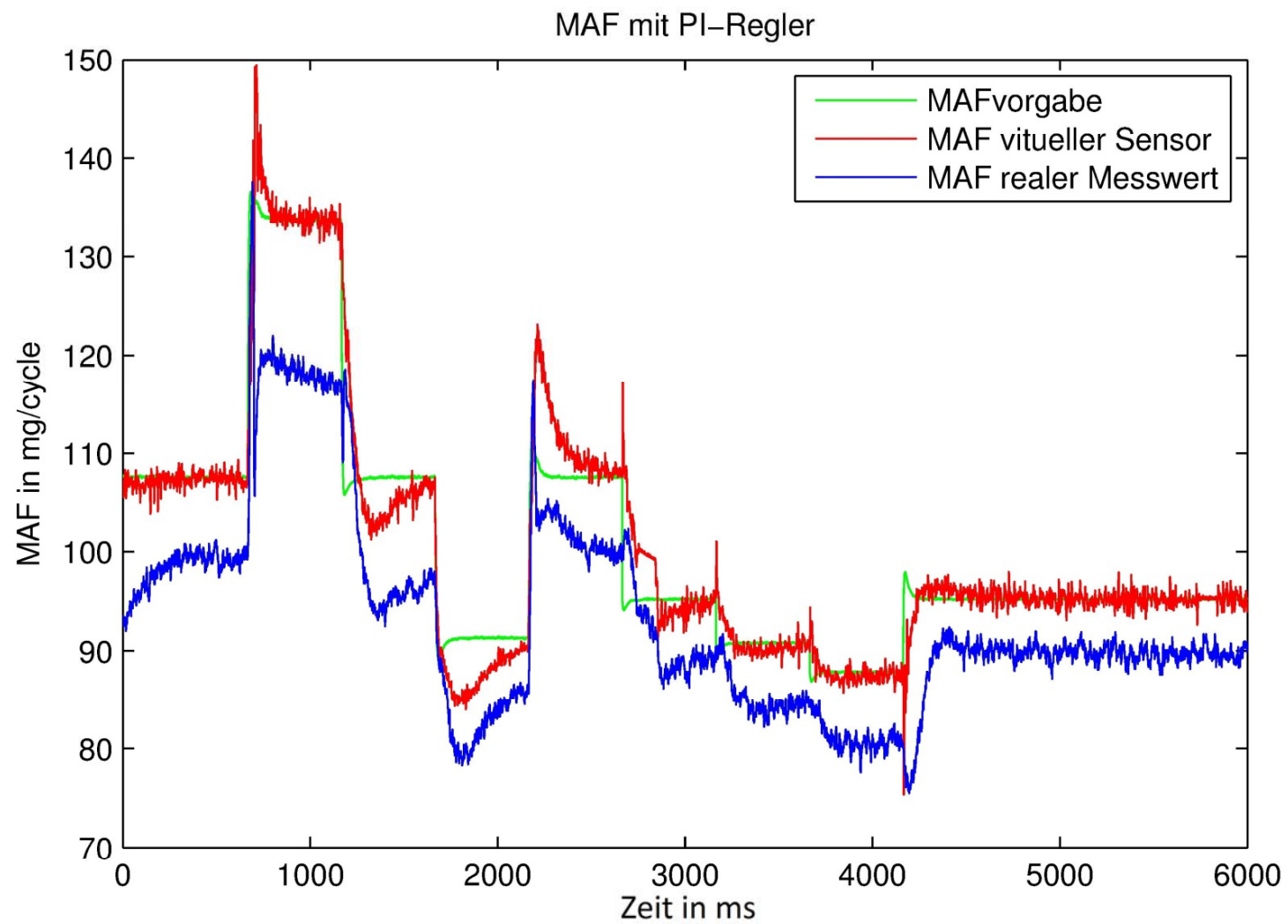


PI Regler

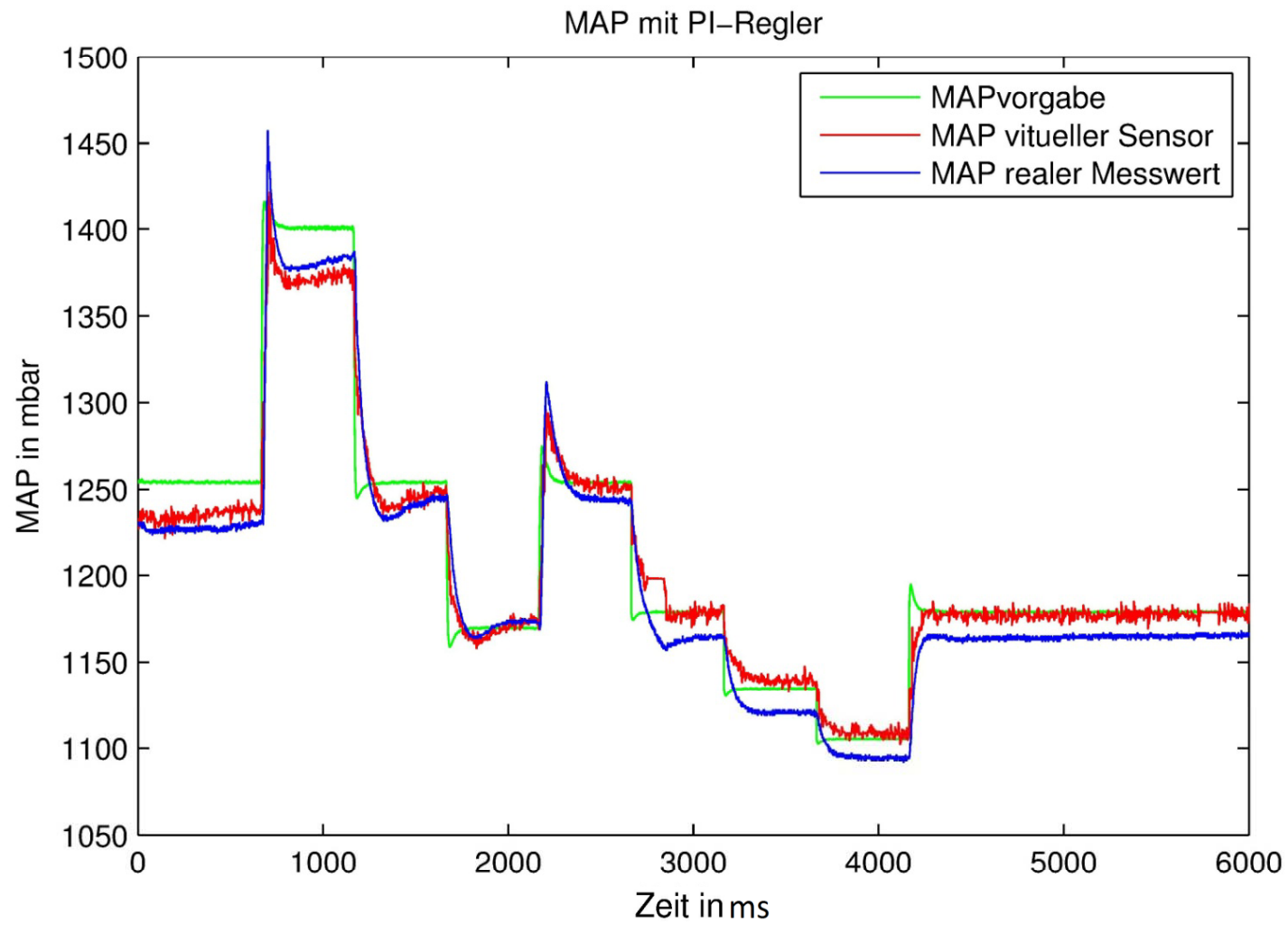
- mit Sprungantwort von MAF und MAP nach Ziegler Nichols ausgelegt
- Feintuning des Reglers direkt am Prüfstand
- Reset für Integrator und Anti-wind-up Schaltung



PI Regler

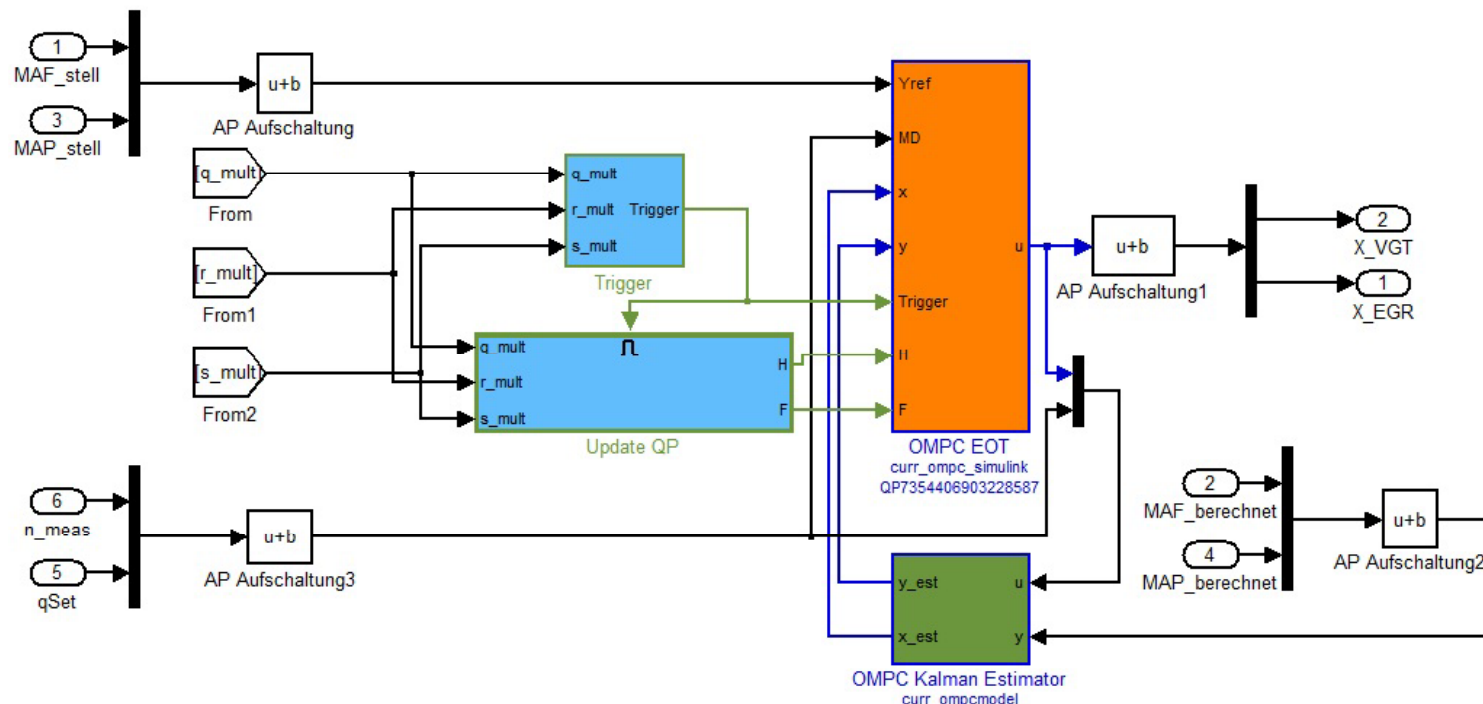


PI Regler



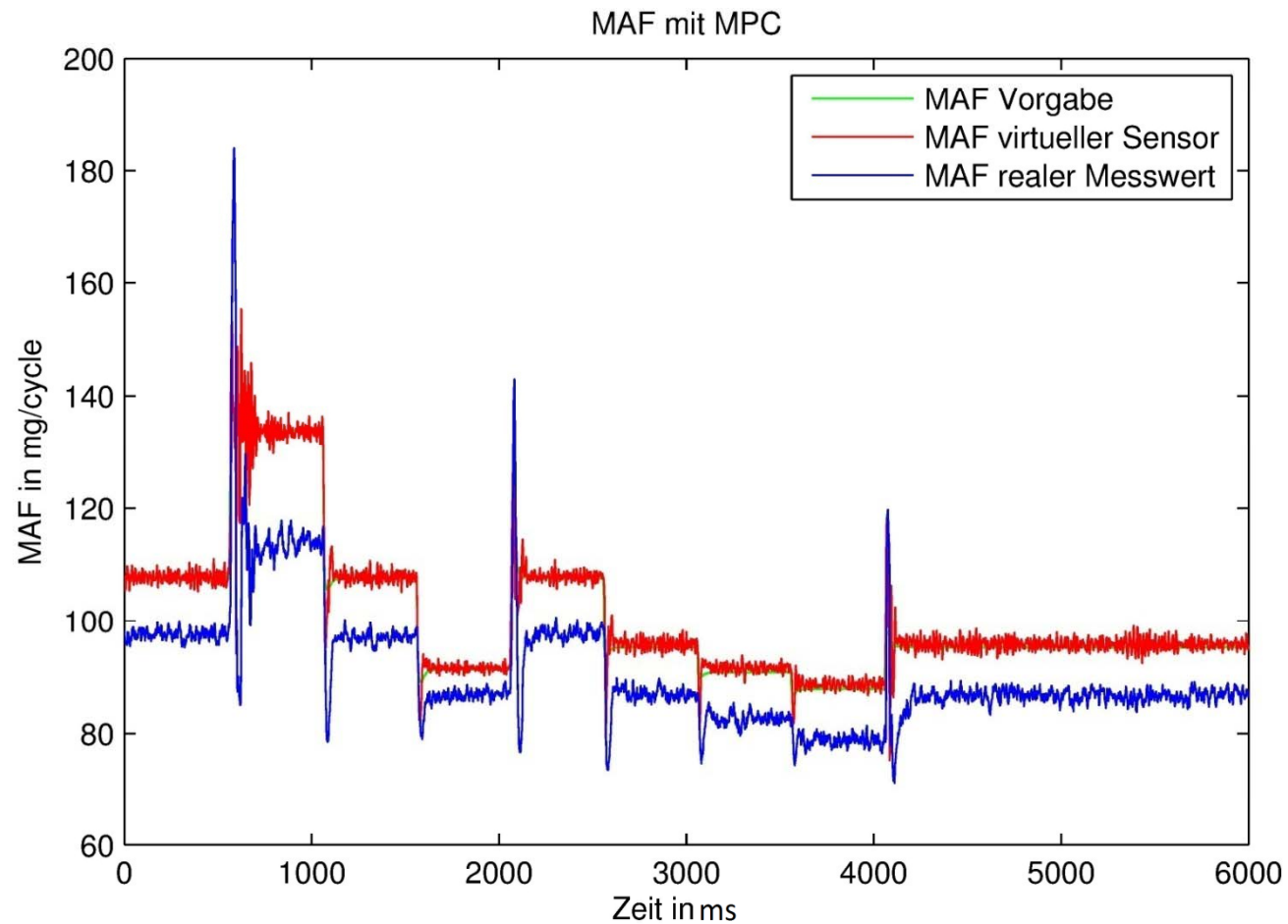
Modellprädiktive Regelung (MPC)

- Modellidentifikation mit Matlab Toolbox *ident*
- Auslegung in Simulation mit linearem Modell
- Implementierung am Prüfstand mit qpOASES



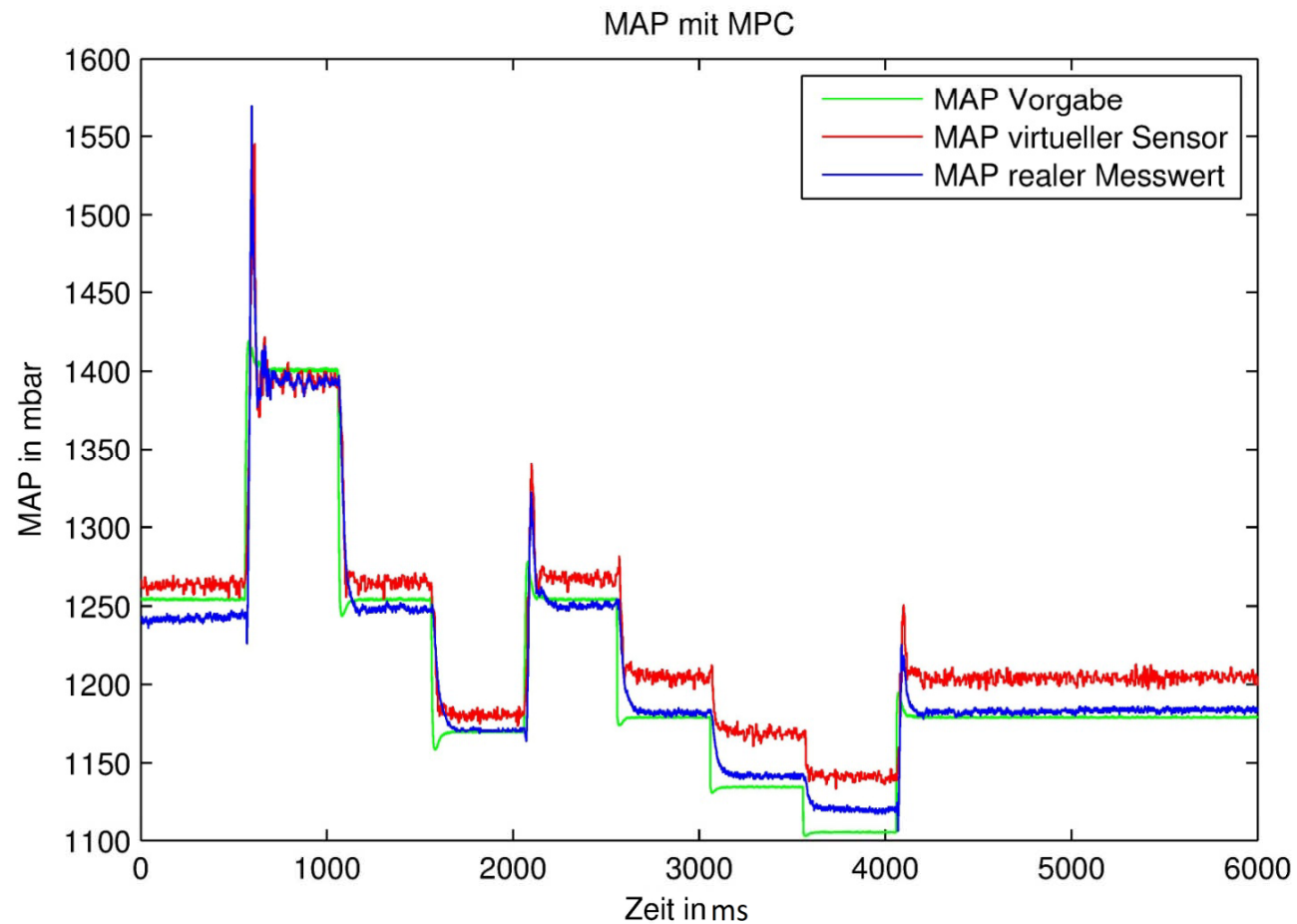
Modellprädiktive Regelung (MPC)

- Testzyklus um den Arbeitspunkt der Identifikation



Modellprädiktive Regelung (MPC)

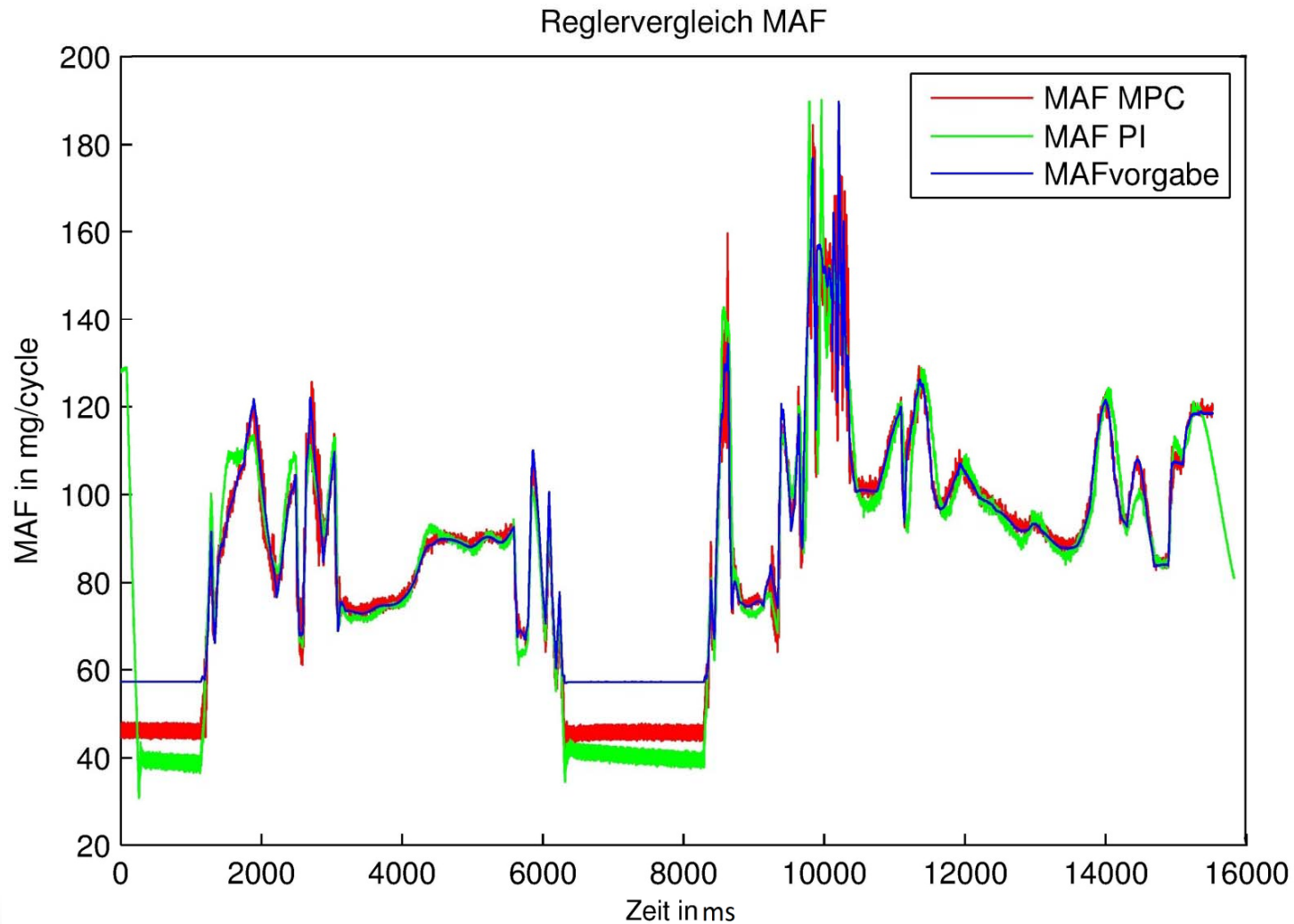
- Testzyklus um den Arbeitspunkt der Identifikation



Vergleich der Regler

○ realer Testzyklus

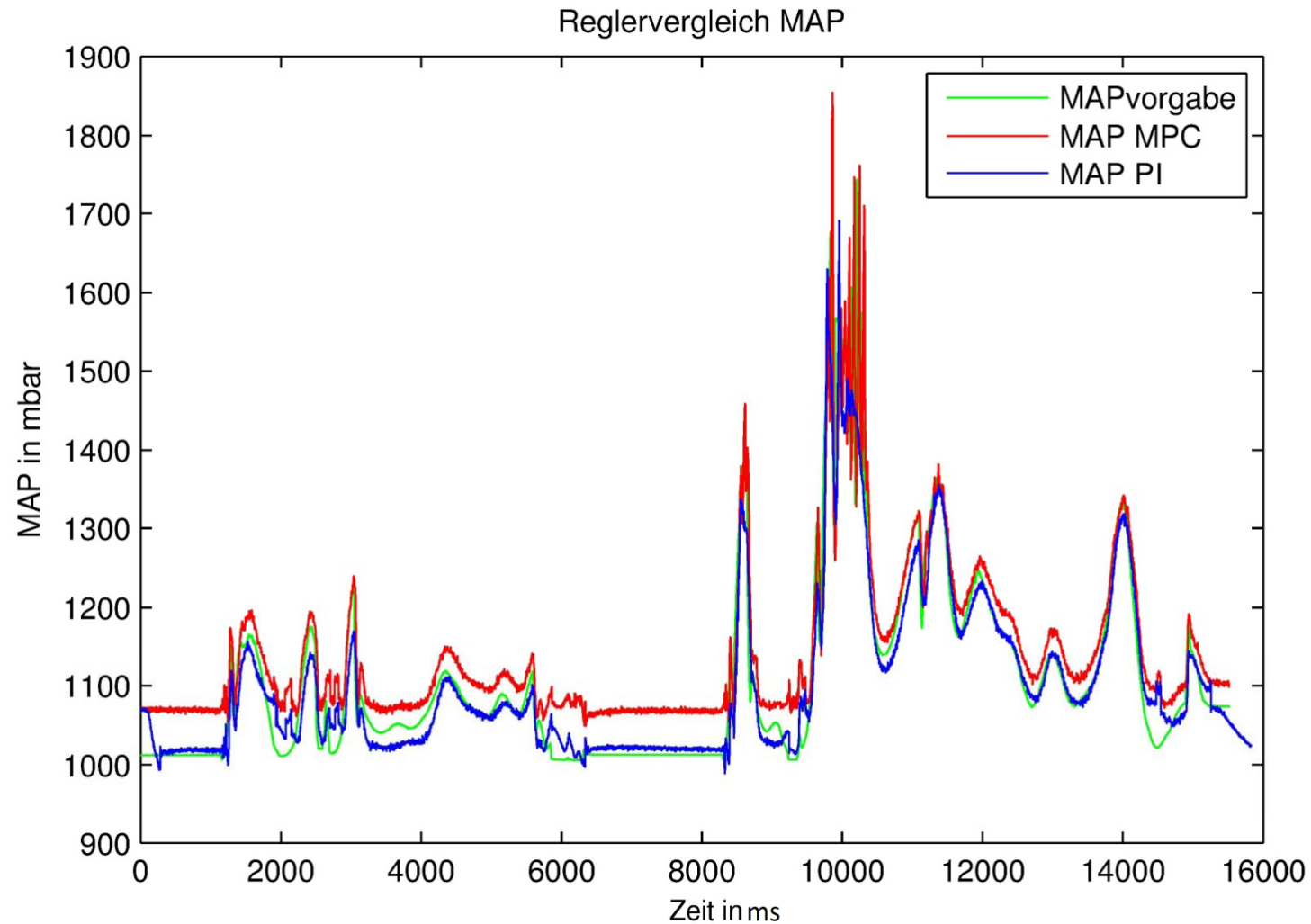
	Abweichung	
	maximal	durschsc.
MAF MPC	46%	6%
MAF PI	33%	9%



Vergleich der Regler

○ realer Testzyklus

	Abweichung	
	maximal	durschsc.
MAP MPC	28%	4%
MAP PI	13%	2%



Danke für Ihre
Aufmerksamkeit