Generierung des Eingangssingals für Barrier Bucket RF Systeme and der GSI

TECHNISCHE UNIVERSITÄT DARMSTADT

Jonas Christ, Artem Moskalew, Maximilian Nolte
Jens Harzheim, M.Sc.

Projektseminar Beschleunigertechnik

Outline

1. Einführung
   - Problemstellung
   - Zielsetzung
   - Gegeben
2. Erreichtes
   - Gerätekommunikation
   - Code
3. Evaluierung
   - Gerätekommunikation
   - Code
4. Ausblick

Problemstellung

Zielsetzung

Gegeben

MLBS.py

Gegeben

FFT.py

MLBS.py

Gegeben

FFT.py

MLBS.py

getH.py

Gegeben

FFT.py

MLBS.py

getH.py

computeUin.py

Gegeben

FFT.py

MLBS.py

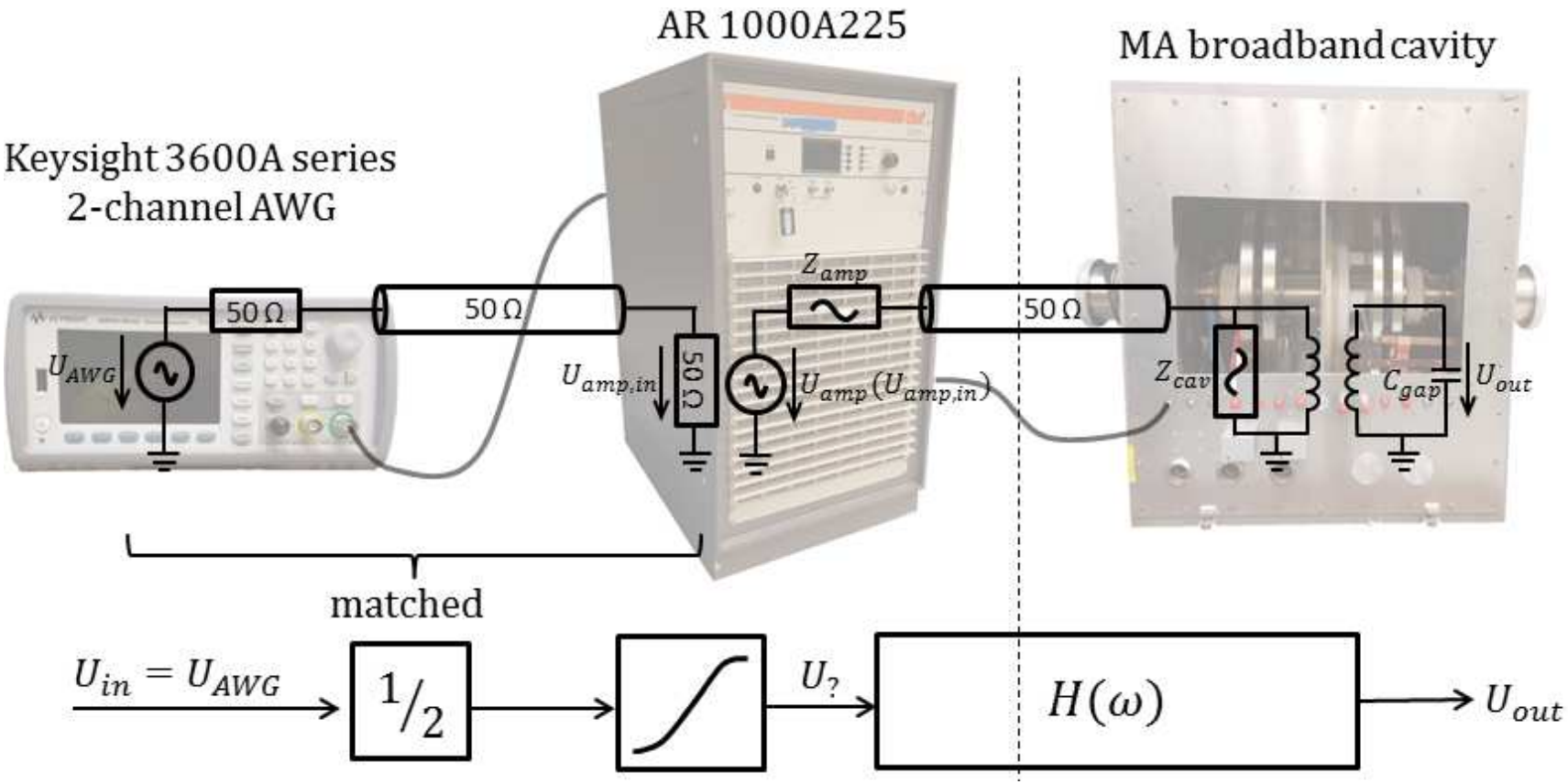
getH.py

computeUin.py

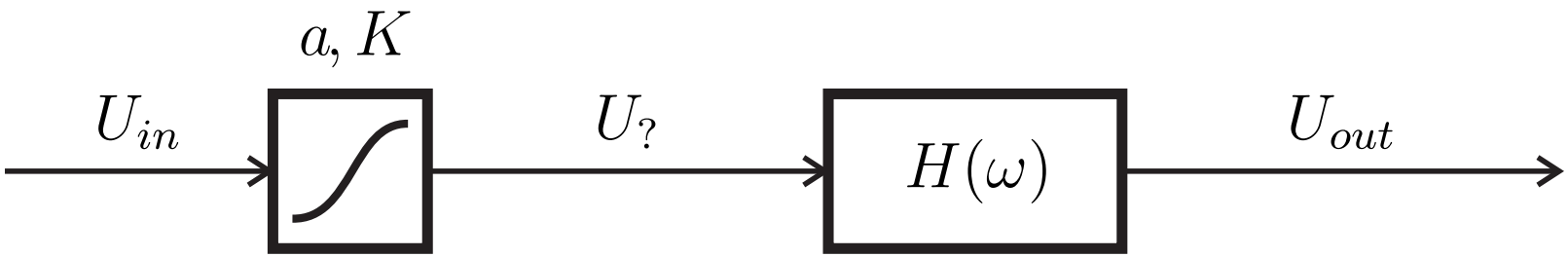
runme_compute.py

Erreichtes: das VISA-Handbuch

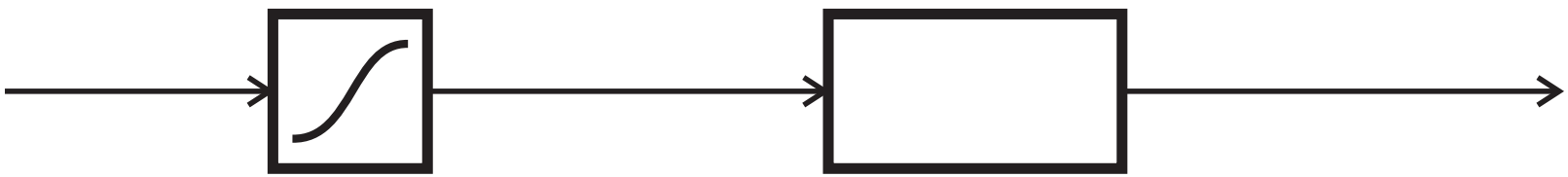
Erreichtes: Code



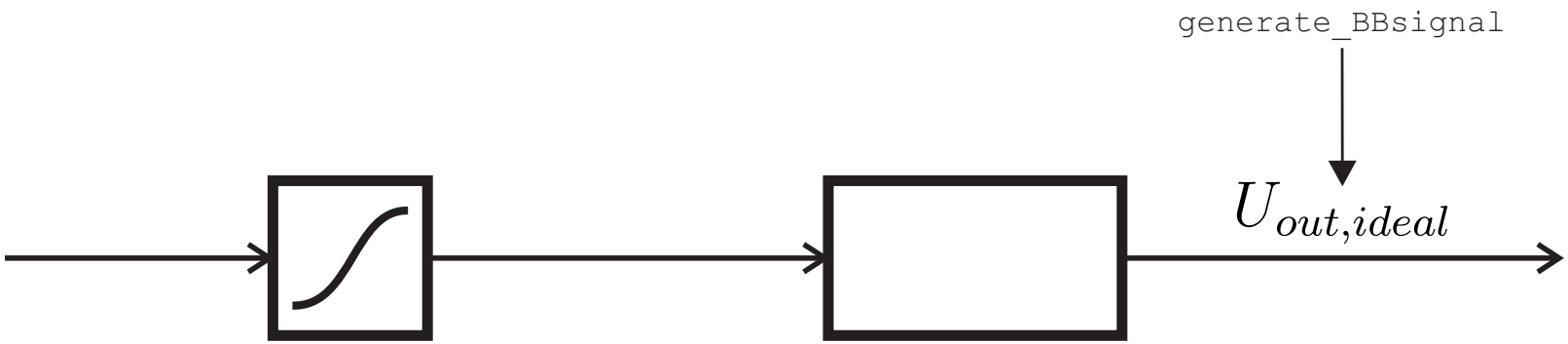
Erreichtes: Code



Erreichtes: Code

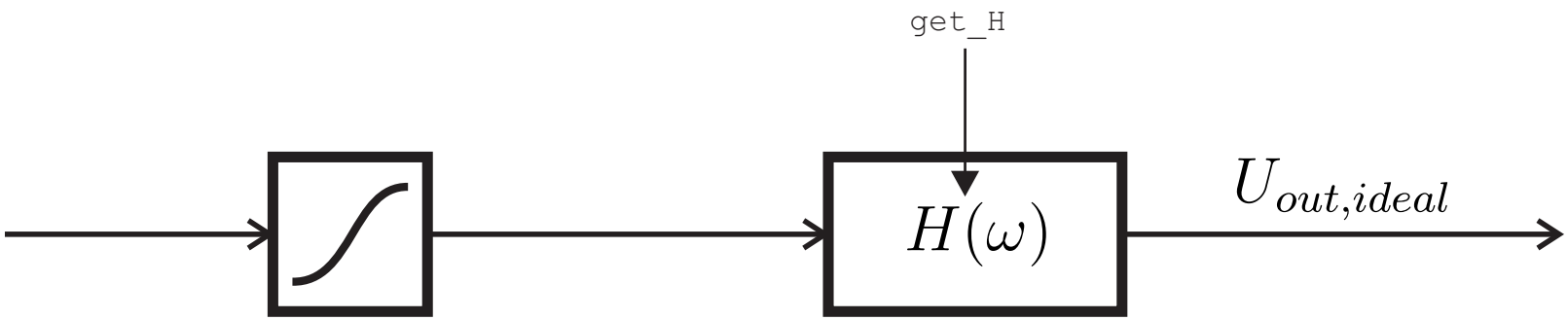


Erreichtes: Code



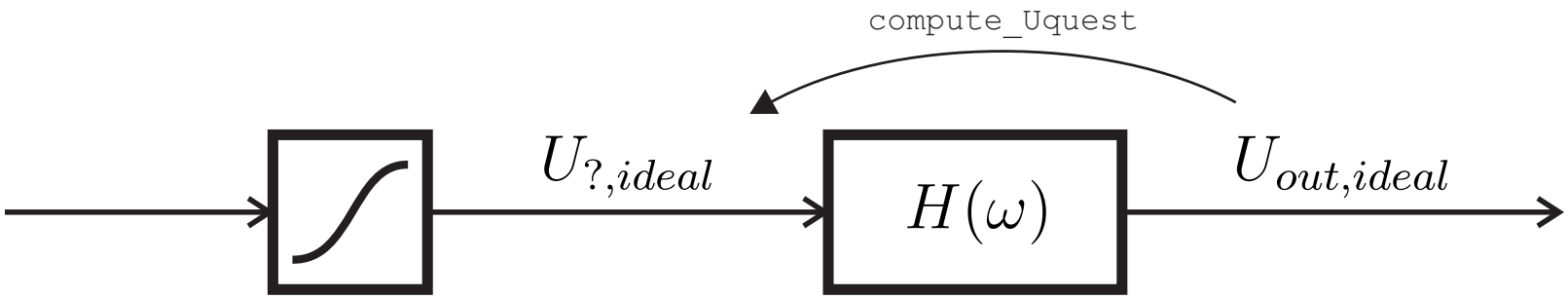

```
1 Uout_ideal = generate_BBsignal ( fq_rep , fq_bb , vpp )
```

Erreichtes: Code



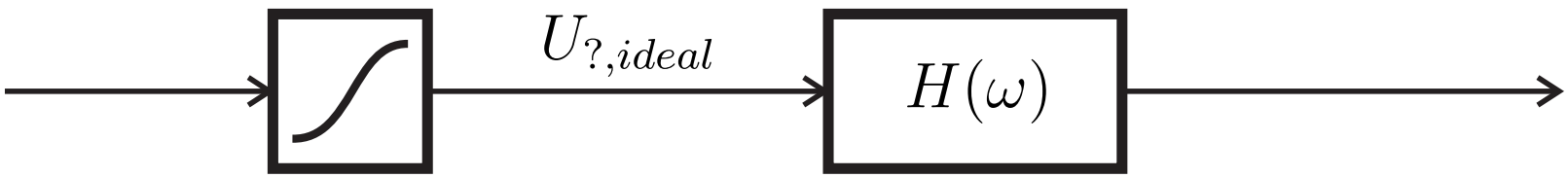

```
1 Uout_ideal = generate_BBsignal ( fq_rep, fq_bb, vpp )
2 H = measure_H ( )
```

Erreichtes: Code



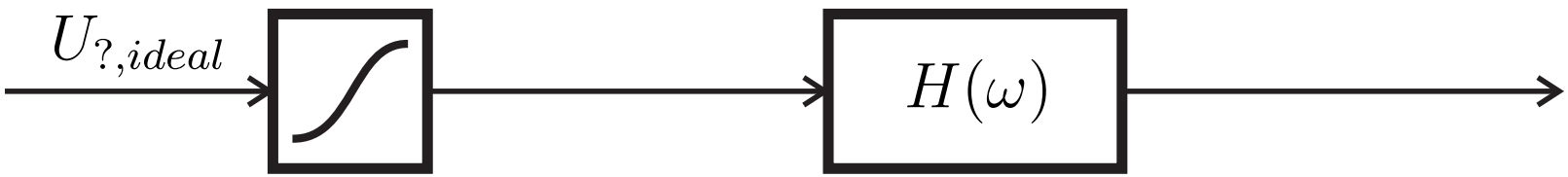

```
1 Uout_ideal = generate_BBsignal ( fq_rep, fq_bb, vpp )
2 H = measure_H ( )
3 Uquest_ideal = compute_Uquest ( Uout_ideal, H )
```

Erreichtes: Code



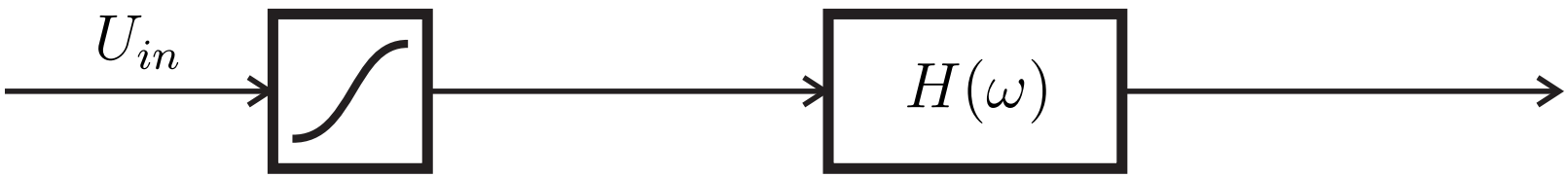

```
1 Uout_ideal = generate_BBsignal ( fq_rep, fq_bb, vpp )
2 H = measure_H ( )
3 Uquest_ideal = compute_Uquest ( Uout_ideal, H )
```

Erreichtes: Code



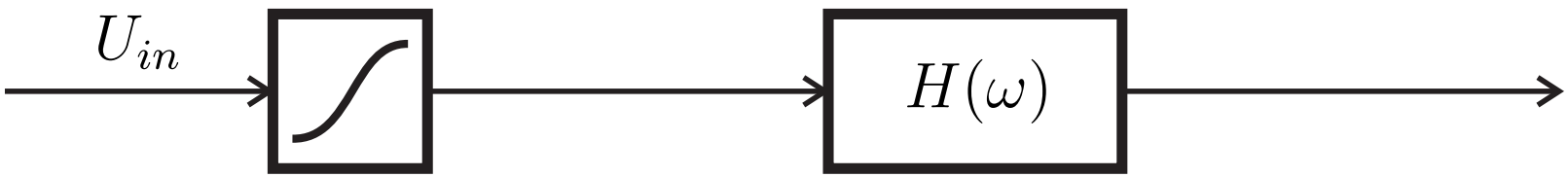

```
1 Uout_ideal = generate_BBsignal ( fq_rep, fq_bb, vpp )
2 H = measure_H ( )
3 Uquest_ideal = compute_Uquest ( Uout_ideal, H )
```

Erreichtes: Code



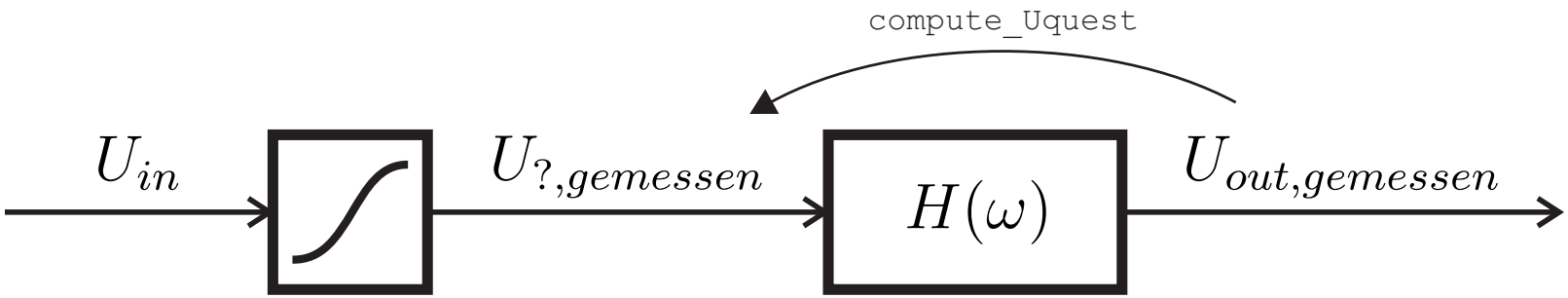

```
1 Uout_ideal = generate_BBsignal ( fq_rep, fq_bb, vpp )
2 H = measure_H ( )
3 Uquest_ideal = compute_Uquest ( Uout_ideal, H )
4 Uin = Uquest_ideal
```

Erreichtes: Code



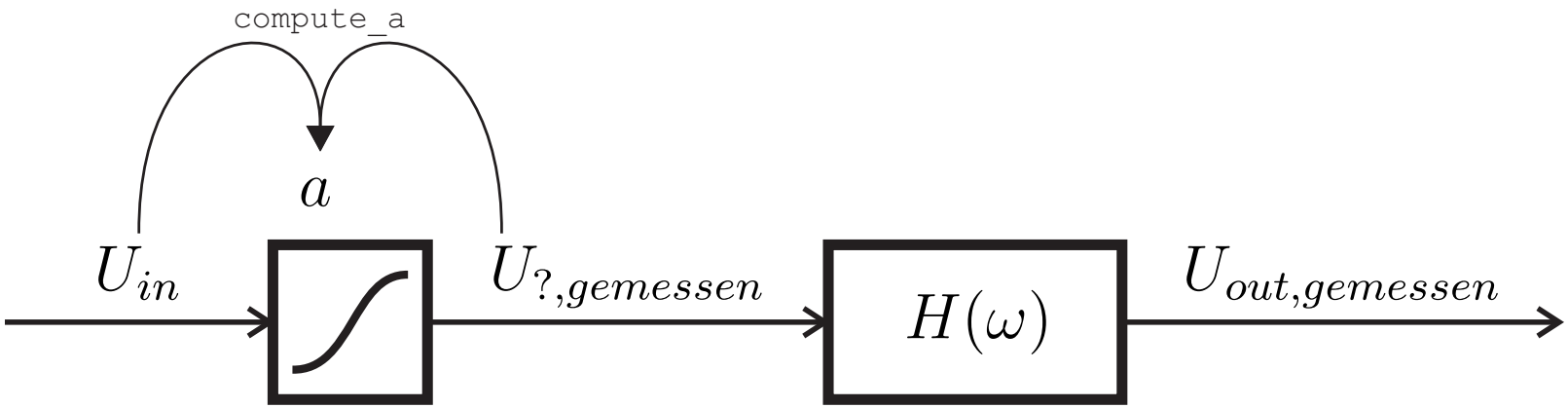

```
1 Uout_ideal = generate_BBsignal ( fq_rep, fq_bb, vpp )
2 H = measure_H ( )
3 Uquest_ideal = compute_Uquest ( Uout_ideal, H )
4 Uin = Uquest_ideal
5 Uout_measured = measure_Uout ( Uin )
```

Erreichtes: Code



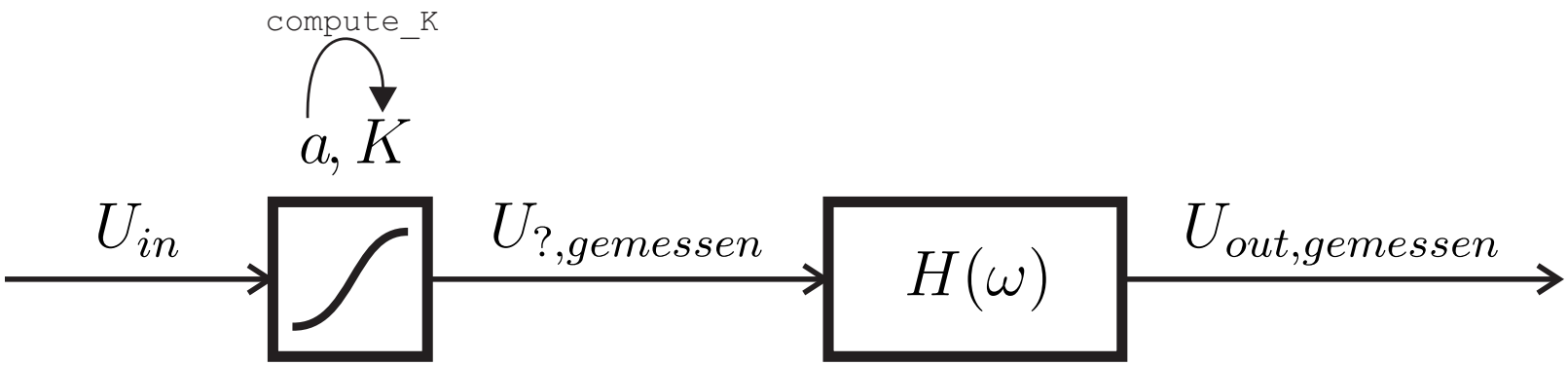

```
1 Uout_ideal = generate_BBsignal ( fq_rep, fq_bb, vpp )
2 H = measure_H ( )
3 Uquest_ideal = compute_Uquest ( Uout_ideal, H )
4 Uin = Uquest_ideal
5 Uout_measured = measure_Uout ( Uin )
6 Uquest_measured = compute_Uquest ( Uout_measured, H )
```

Erreichtes: Code



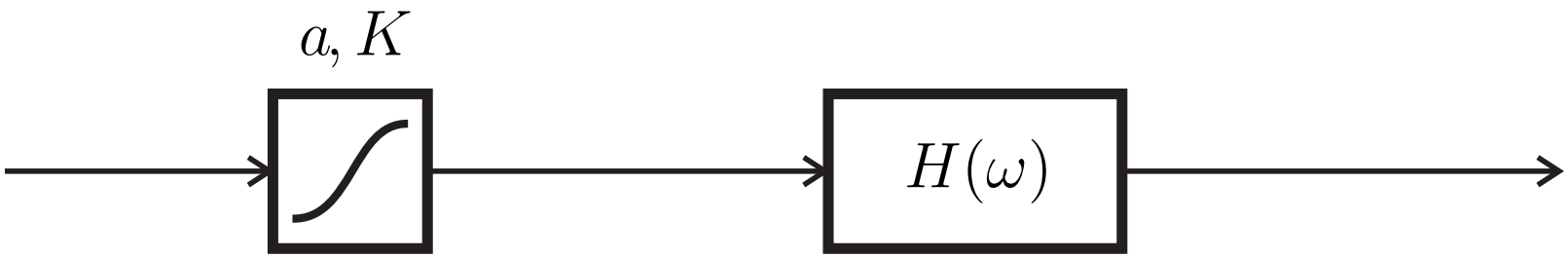

```
1 Uout_ideal = generate_BBsignal ( fq_rep, fq_bb, vpp )
2 H = measure_H ( )
3 Uquest_ideal = compute_Uquest ( Uout_ideal, H )
4 Uin = Uquest_ideal
5 Uout_measured = measure_Uout ( Uin )
6 Uquest_measured = compute_Uquest ( Uout_measured, H )
7 a = compute_a ( Uin, Uquest_measured, N )
```

Erreichtes: Code



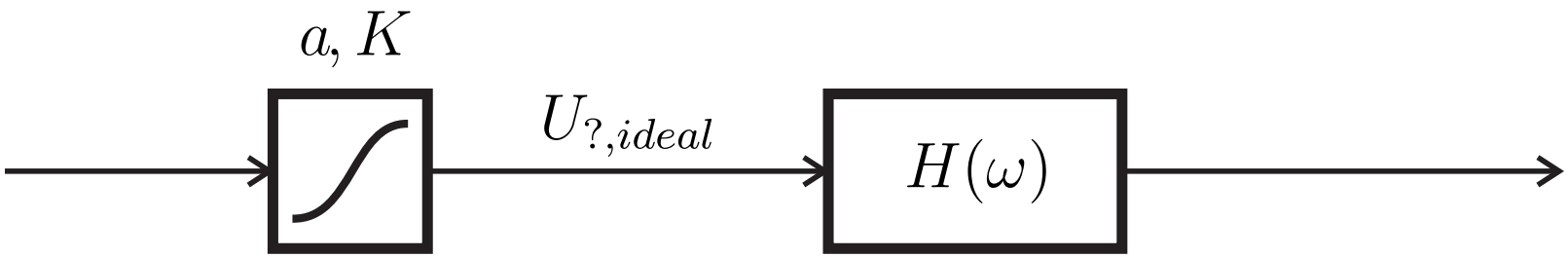

```
1 Uout_ideal = generate_BBsignal ( fq_rep, fq_bb, vpp )
2 H = measure_H ( )
3 Uquest_ideal = compute_Uquest ( Uout_ideal, H )
4 Uin = Uquest_ideal
5 Uout_measured = measure_Uout ( Uin )
6 Uquest_measured = compute_Uquest ( Uout_measured, H )
7 a = compute_a ( Uin, Uquest_measured, N )
8 K = compute_K ( a )
```

Erreichtes: Code



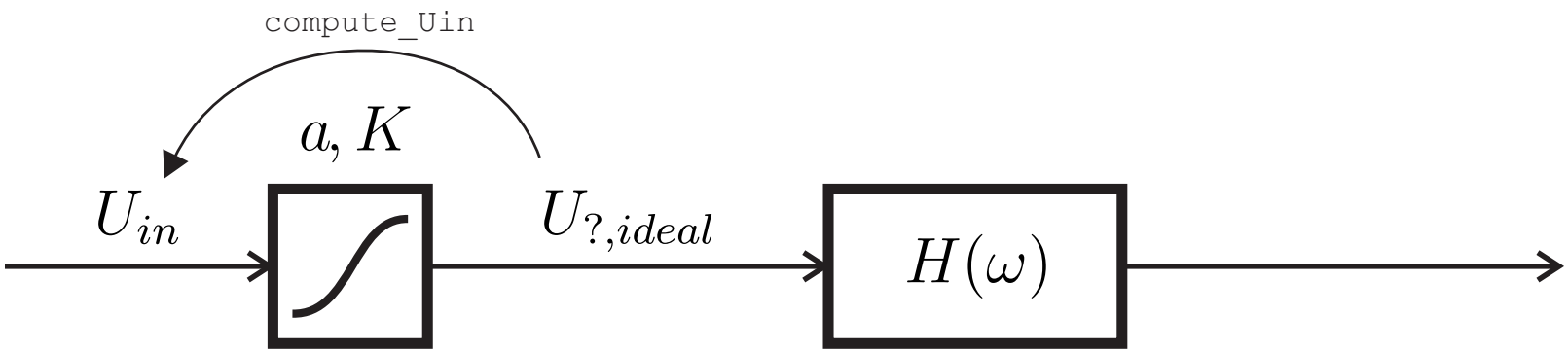

```
Uout_ideal = generate_BBsignal ( fq_rep, fq_bb, vpp )
H = measure_H ( )
Uquest_ideal = compute_Uquest ( Uout_ideal, H )
Uin = Uquest_ideal
Uout_measured = measure_Uout ( Uin )
Uquest_measured = compute_Uquest ( Uout_measured, H )
a = compute_a ( Uin, Uquest_measured, N )
K = compute_K ( a )
```

Erreichtes: Code



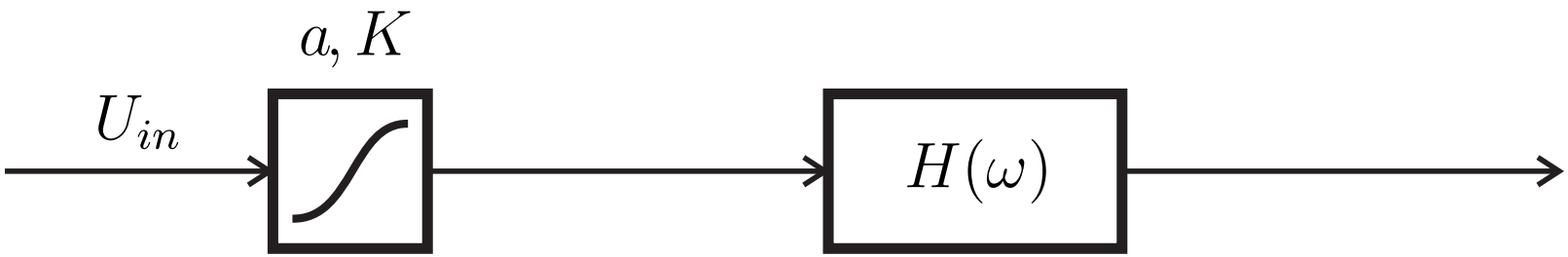

```
1 Uout_ideal = generate_BBsignal ( fq_rep, fq_bb, vpp )
2 H = measure_H ( )
3 Uquest_ideal = compute_Uquest ( Uout_ideal, H )
4 Uin = Uquest_ideal
5 Uout_measured = measure_Uout ( Uin )
6 Uquest_measured = compute_Uquest ( Uout_measured, H )
7 a = compute_a ( Uin, Uquest_measured, N )
8 K = compute_K ( a )
```

Erreichtes: Code



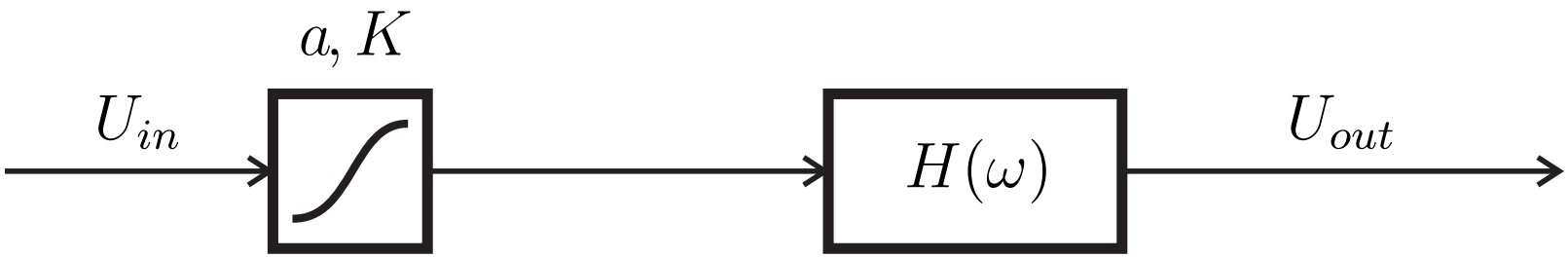

```
1 Uout_ideal = generate_BBsignal ( fq_rep, fq_bb, vpp )
2 H = measure_H ( )
3 Uquest_ideal = compute_Uquest ( Uout_ideal, H )
4 Uin = Uquest_ideal
5 Uout_measured = measure_Uout ( Uin )
6 Uquest_measured = compute_Uquest ( Uout_measured, H )
7 a = compute_a ( Uin, Uquest_measured, N )
8 K = compute_K ( a )
9 Uin = compute_Uin ( Uquest_ideal, K, Vpp )
```

Erreichtes: Code




```
1 Uout_ideal = generate_BBsignal ( fq_rep, fq_bb, vpp )
2 H = measure_H ( )
3 Uquest_ideal = compute_Uquest ( Uout_ideal, H )
4 Uin = Uquest_ideal
5 Uout_measured = measure_Uout ( Uin )
6 Uquest_measured = compute_Uquest ( Uout_measured, H )
7 a = compute_a ( Uin, Uquest_measured, N )
8 K = compute_K ( a )
9 Uin = compute_Uin ( Uquest_ideal, K, Vpp )
```

Erreichtes: Code




```
1  Uout_ideal = generate_BBsignal ( fq_rep, fq_bb, vpp )
2  H = measure_H ( )
3  Uquest_ideal = compute_Uquest ( Uout_ideal, H )
4  Uin = Uquest_ideal
5  Uout_measured = measure_Uout ( Uin )
6  Uquest_measured = compute_Uquest ( Uout_measured, H )
7  a = compute_a ( Uin, Uquest_measured, N )
8  K = compute_K ( a )
9  Uin = compute_Uin ( Uquest_ideal, K, Vpp )
10 Uout = measure_Uout ( Uin )
```

Evaluierung: Gerätekommunikation

Evaluierung: Python-Code für die nichtlinere Verzerrung

Ausblick