

# 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



Institut für Theorie  
Elektromagnetischer Felder  
Computational Electromagnetics  
Research Group at GSCE

# 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

---

```
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 )  
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

---

```
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 )
```

---

## 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 )
10 Uout_measured = measure_Uout ( Uin )
```



---

# Evaluierung: Gerätekommunikation

---

---

# Evaluierung: Python-Code für die nichtlinere Verzerrung

---

---

# Ausblick

---