



University of Stuttgart
Institute for System Dynamics

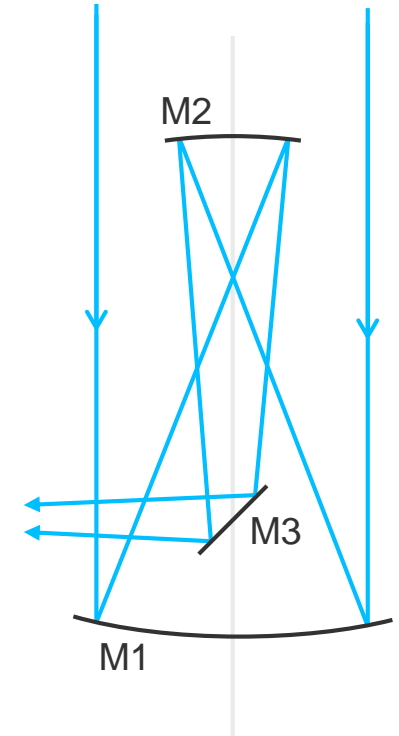
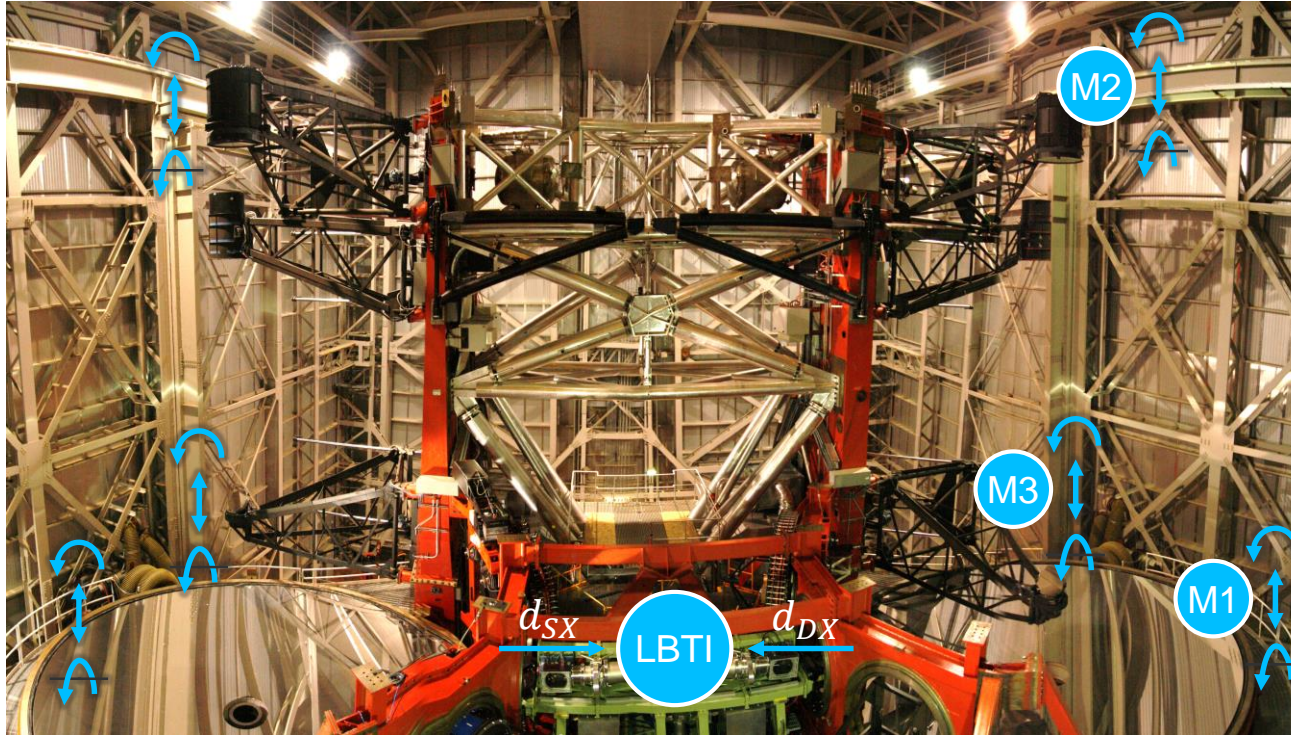
OVMS-plus

Disturbance compensation at the LBT



Problem description

Optical setup at the Large Binocular Telescope (LBT)

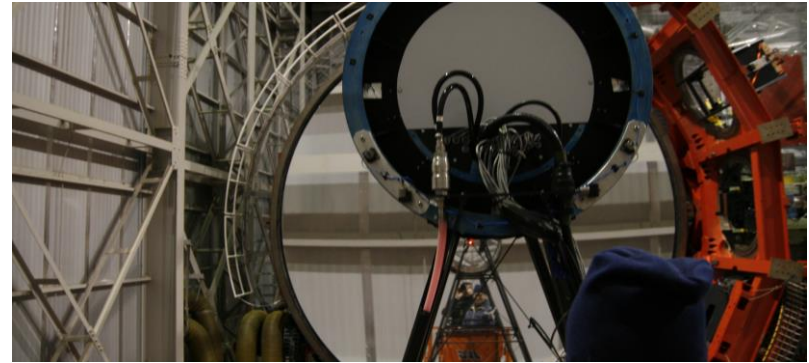


$$\text{Goal: } d_{SX} - d_{DX} = 0$$

Problem description

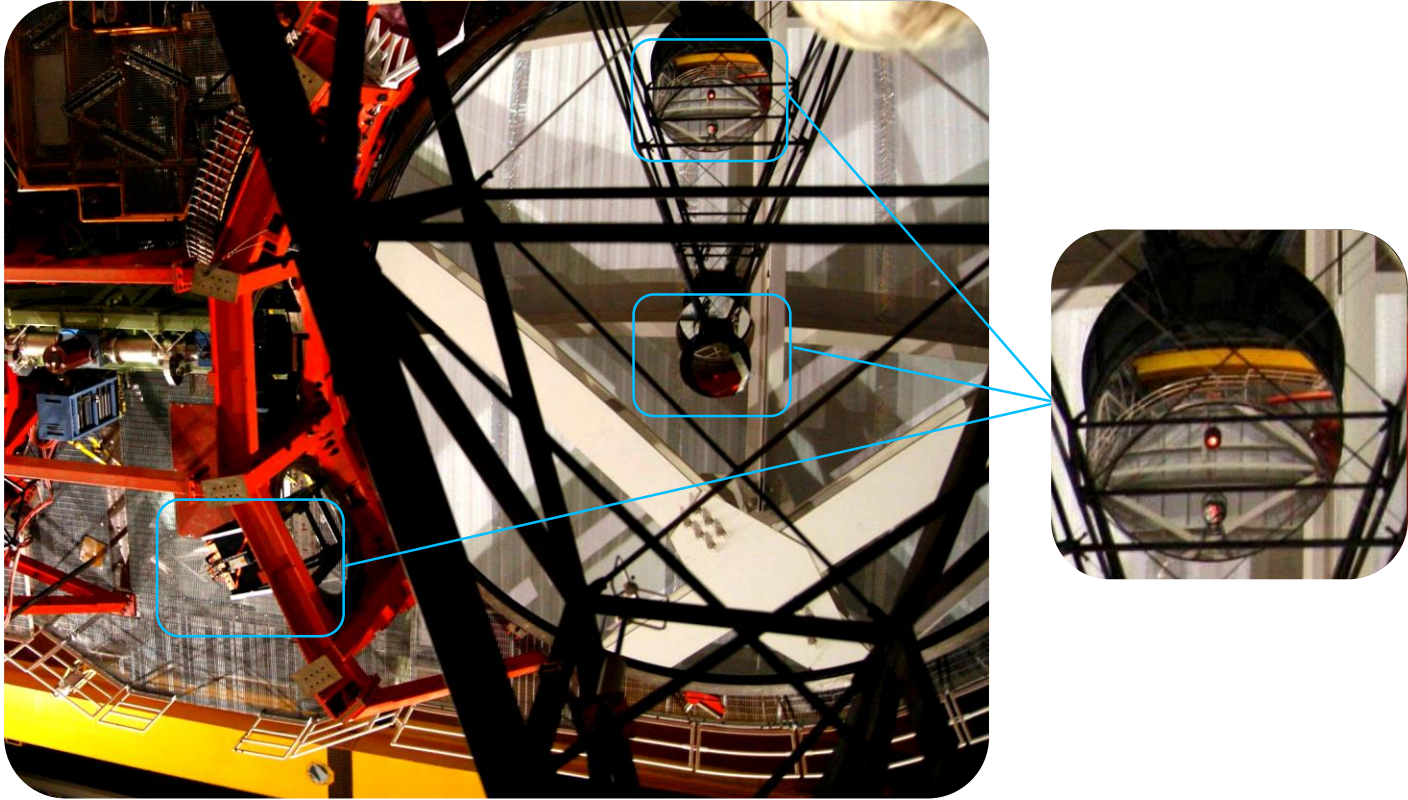
Initial OVMS setup

- Measurement of vibrations with accelerometers
- Task:
 - Reconstruction of mirror motion
 - Distribution to LBT instruments
- Assumptions:
 - Mirrors are rigid bodies
 - No dynamic coupling between low order aberrations



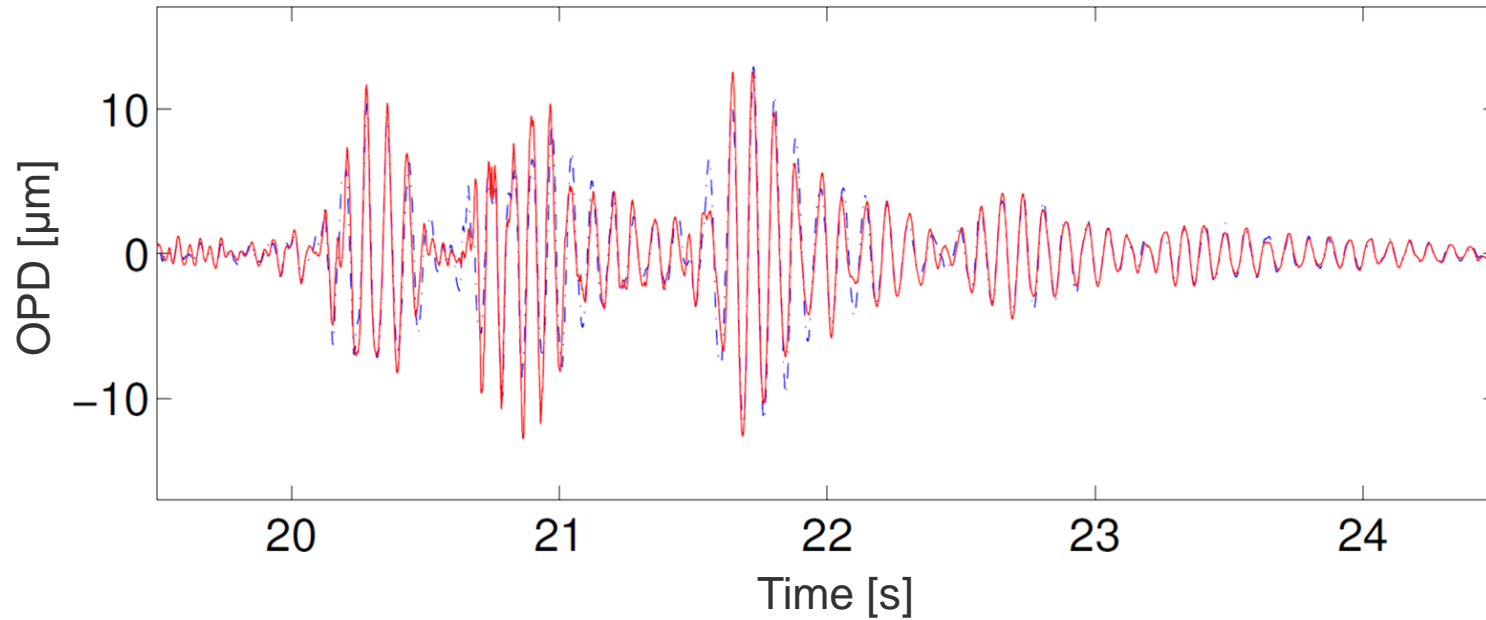
Measurement Campaign

Is it feasible?



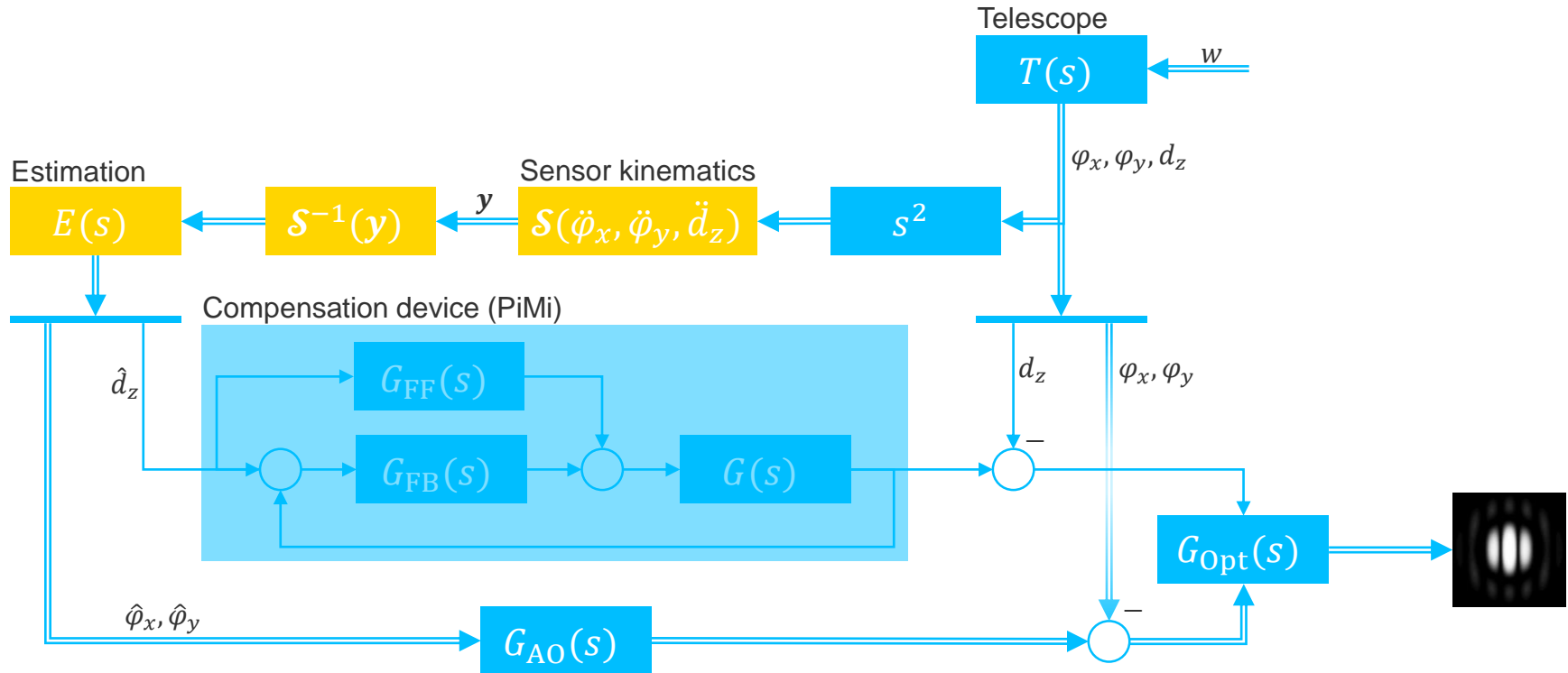
Measurement Campaign

It is feasible!



Problem description

Disturbance compensation overview



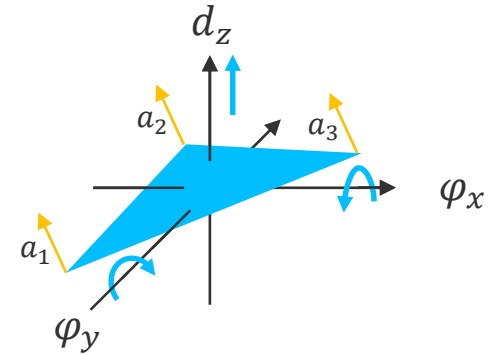
Sensor kinematics

Sensor kinematics

Forward kinematics

- Calculate sensor values from rigid body motion
 - Mirror is a moving reference frame, transformation to fixed frame using $T_{K_0}^{K_1}(t)$
- Linearization around zero yields:

$$\begin{aligned}
 \mathbf{a}_{k,m}(t) &= \mathbf{u}_{k,m}^\top \begin{pmatrix} 0 & -\ddot{\phi}_{z,m} & \ddot{\phi}_{y,m} & \ddot{d}_{x,m} \\ \ddot{\phi}_{z,m} & 0 & -\ddot{\phi}_{x,m} & \ddot{d}_{y,m} \\ -\ddot{\phi}_{y,m} & \ddot{\phi}_{x,m} & 0 & \ddot{d}_{z,m} \\ 0 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} x_{k,m} \\ y_{k,m} \\ z_{k,m} \\ 1 \end{pmatrix} \\
 &= \mathbf{u}_{k,m}^\top \begin{pmatrix} 0 & z_{k,m} & -y_{k,m} & 1 & 0 & 0 \\ -z_{k,m} & 0 & x_{k,m} & 0 & 1 & 0 \\ y_{k,m} & -x_{k,m} & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \ddot{\boldsymbol{\eta}}_m \Rightarrow \mathbf{a}_m(t) = \mathcal{S}(\ddot{\boldsymbol{\eta}}_m, \boldsymbol{\theta}_m)
 \end{aligned}$$





Sensor kinematics

Inverse kinematics

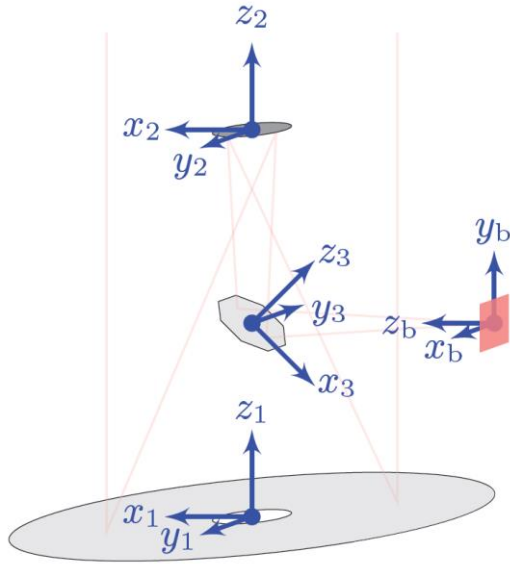
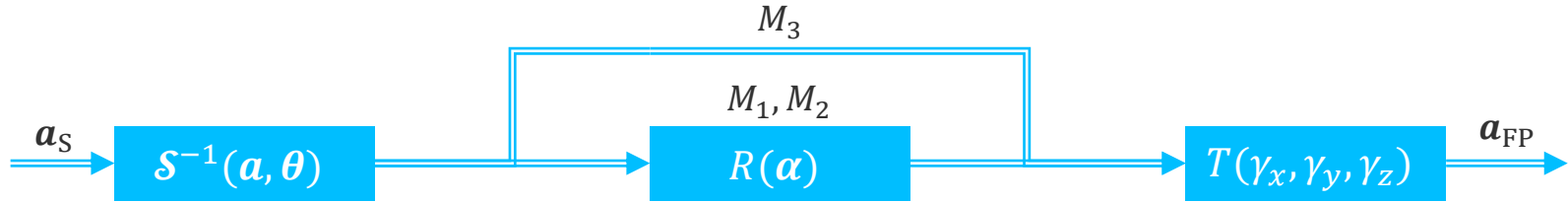
- Calculate rigid body motion from sensor values
 - Inversion of $\mathcal{S}(\ddot{\boldsymbol{\eta}}_m, \boldsymbol{\theta}_m)$
 - Overdetermined for more than sensors than degrees of freedom
 - Use of pseudo inverse yields:

$$\ddot{\boldsymbol{\eta}}_m(t) = \mathcal{S}^{-1}(\boldsymbol{a}_m, \boldsymbol{\theta}_m)$$

- LBT:
 - 3 out-of-plane sensors \rightarrow 3 out-of-plane degrees of freedom 
 - 2 in-plane sensors \rightarrow 3 in-plane degrees of freedom 

Sensor kinematics

Concluding block diagram



	α
LUCI	25.9°
LBTI	0°
LINC	18.5°

	LUCI/LBTI	LINC
$\gamma_{1,x}$	1175	1198
$\gamma_{1,y}$	-1175	-1198
$\gamma_{1,z}$	-2	-2
$\gamma_{2,x}$	130	133
$\gamma_{2,y}$	-130	-133
$\gamma_{2,z}$	2	2
$\gamma_{3,x}$	37	38
$\gamma_{3,y}$	-52	-54
$\gamma_{3,z}$	$-\sqrt{2}$	$-\sqrt{2}$

Position estimation

Position estimation

Broad band filtering

- Filter with broad pass band aiming at a perfect double integration from 10 – 60 Hz
- State space equations:

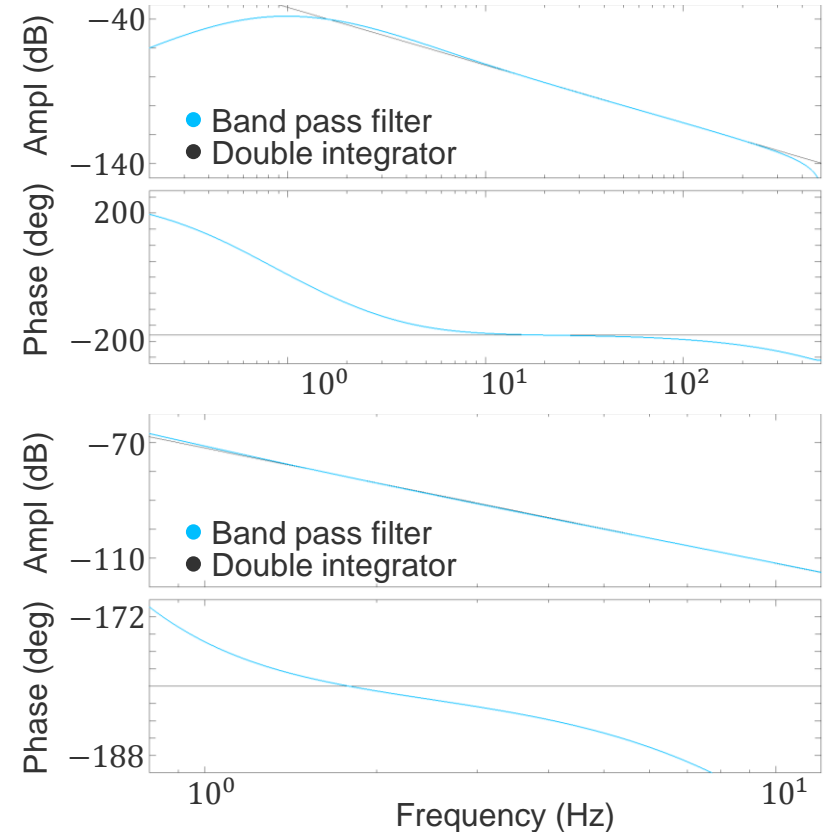
$$\dot{\xi}(t) = A\xi(t) + By(t)$$

$$x(t) = C\xi(t)$$

$$A = \begin{pmatrix} -p_2 & 0 & 0 & 0 & 0 & 0 & 0 \\ z_2 - p_2 & -p_1 & 0 & 0 & 0 & 0 & 0 \\ z_2 - p_2 & z_1 - p_1 & -\omega_H & 0 & 0 & 0 & 0 \\ z_2 - p_2 & z_1 - p_1 & -\omega_H & \omega_L & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & -\omega_H & 0 & 0 \\ 0 & 0 & 0 & 1 & -\omega_H & -\omega_L & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & -\omega_H \end{pmatrix}$$

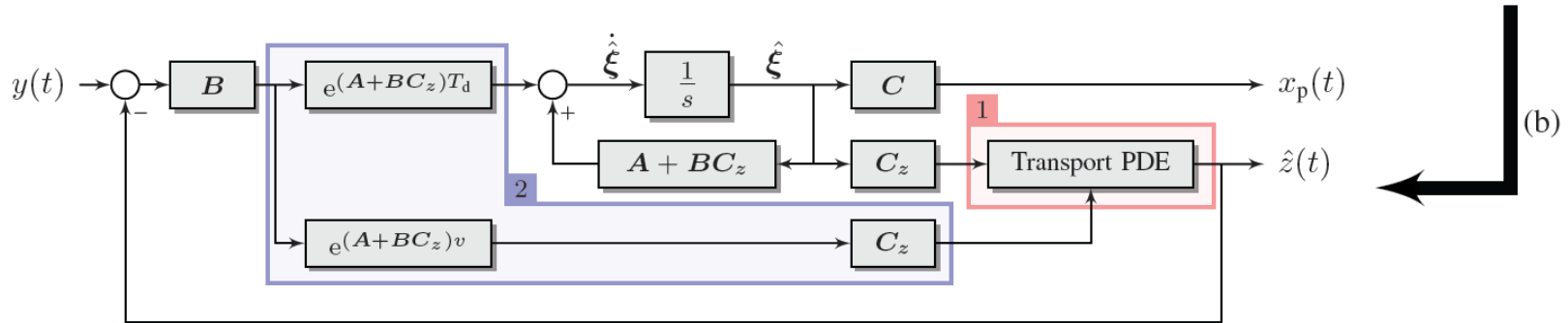
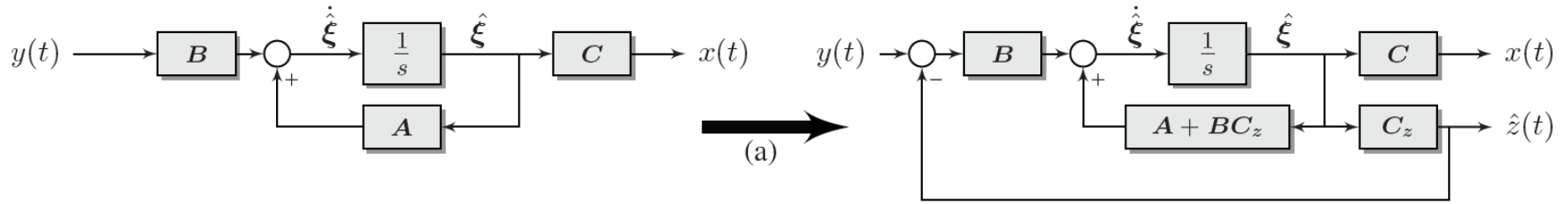
$$B = (1 \ 1 \ 1 \ 1 \ 0 \ 0 \ 0)^T$$

$$C = (0 \ 0 \ 0 \ 0 \ 0 \ 1 \ -\omega_H)$$



Position estimation

Delay compensation



(a) Transformation to Luenberger observer

(b) Extension for delay compensation

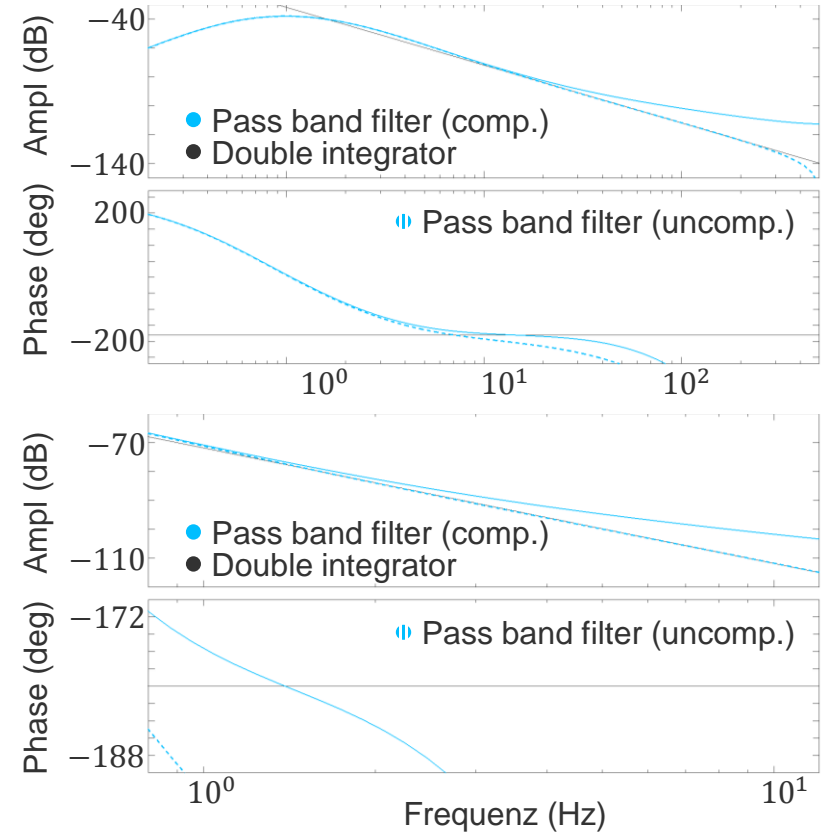
Position estimation

Delay compensation

- Transfer function:

$$X_p(s) = \frac{C(sI - A - BC_z)^{-1}e^{(A+BC_z)T_d}B}{1 + C_z(sI - A - BC_z)^{-1}B}Y_d(s)$$

- Example bode plot for $T_d = 5\text{ms}$



Implementation

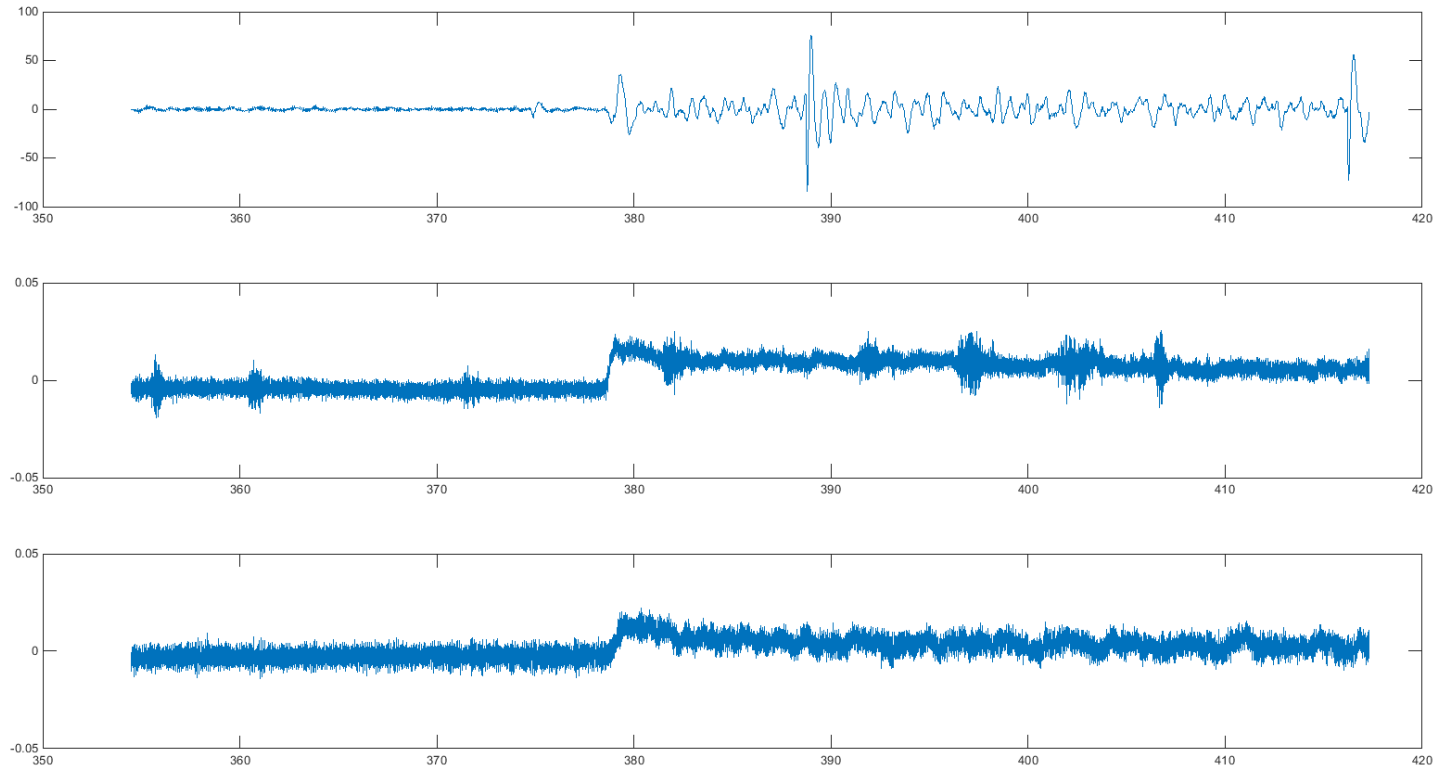
Implementation

Constraints

- Many users with different affinity to the overall system
 - LBTO operational and technical staff
 - Instrument scientists and astronomers
- Change of the observing instrument
 - Modification of frame of reference
- Detect failure of components and minimize loss of functionality
- Delay depends partly on the observing instrument
- Sensor parameters might change

Implementation

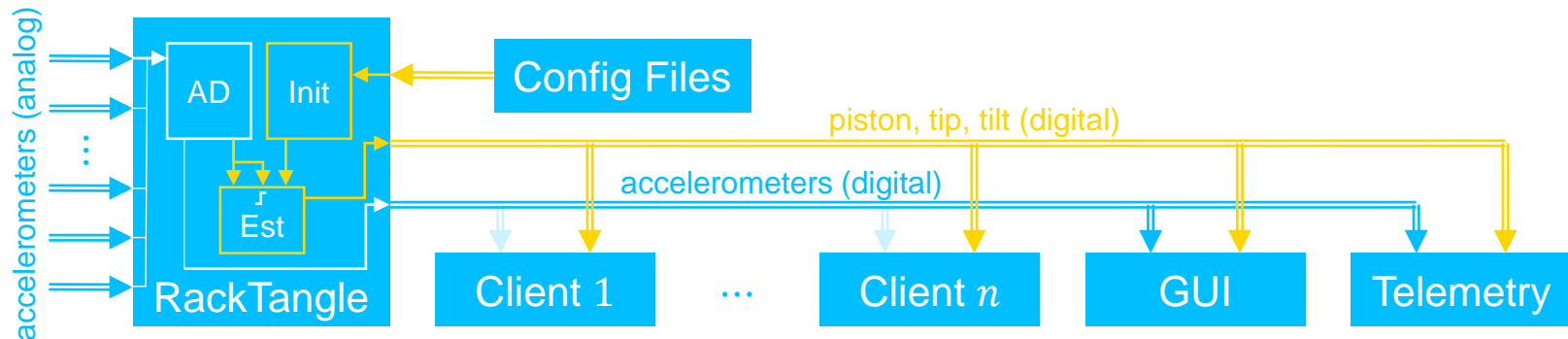
Process



Implementation

Idea of OVMS-plus

- OVMS
 - Distribution of sensor values via UDP-Multicast
 - Calculations are left to the instrument
- OVMS-plus
 - Calculations are **centralized on the RackTangle**
 - Distribution of sensor values **and piston, tip, tilt** via UDP-Multicast



Implementation

Software architecture

Sensor positions

```
1 # Accelerometer Locations:
2 #####
3 M1_SX
4 # Locations:
5 loc:
6 +4.2064, 0.0131
7 1.7659, -3.0178
8 2.4234, 3.4392
9 # Channels:
10 chas:
11 12
12 13
13 14
14 # calibration data:
15 cal:
16 1.030
17 1.030
18 1.044
19 #####
20 M2_SX
21 # Locations:
22 loc:
23 0.0, 0.4250
24 0.3681, -0.2125
25 -0.3681, -0.2125
26 # Channels:
27 chas:
```

```
1 #####
2 configIMU:
3 # M3-rotation:
4 alpha_RX:0.0
5 alpha_RY:0.0
6 # focal plane sensitivity at 1° tilt around x
7 (px) and y (py), and opd-sensitivity factor (opi):
8 #
9 M1_SX
10 px:1175.0
11 py:-1175.0
12 opi:-2.0
13 #
14 M2_SX
15 px:130.0
16 py:-130.0
17 opi:2.0
18 #
19 M3_SX
20 px:137.0
21 py:-52.0
22 opi:-1.4142135623730950488
23 #
24 M1_DX
25 px:-1175.0
26 py:1175.0
27 opi:-2.0
```

Instrument configuration

UEI RackTangle

Initialization

```
1 # Initialization routine
2 # This routine initializes the system and sets up the data structures
3 # for the estimation process.
4 #
5 # Parameters:
6 #   None
7 #
8 # Returns:
9 #   None
10 #
11 # Notes:
12 #   This routine is called at the start of the program.
13 #   It initializes the system and sets up the data structures
14 #   for the estimation process.
15 #
16 #   The routine is called at the start of the program.
17 #   It initializes the system and sets up the data structures
18 #   for the estimation process.
19 #
20 #   The routine is called at the start of the program.
21 #   It initializes the system and sets up the data structures
22 #   for the estimation process.
```

Estimator structure

Estimation

```
1 # Estimation routine
2 # This routine performs the estimation process.
3 # It takes the sensor data and the system model
4 # and returns the estimated state.
5 #
6 # Parameters:
7 #   sensor_data: The sensor data.
8 #   system_model: The system model.
9 #
10 # Returns:
11 #   estimated_state: The estimated state.
12 #
13 # Notes:
14 #   This routine is called at the start of the program.
15 #   It takes the sensor data and the system model
16 #   and returns the estimated state.
```

Sensor values

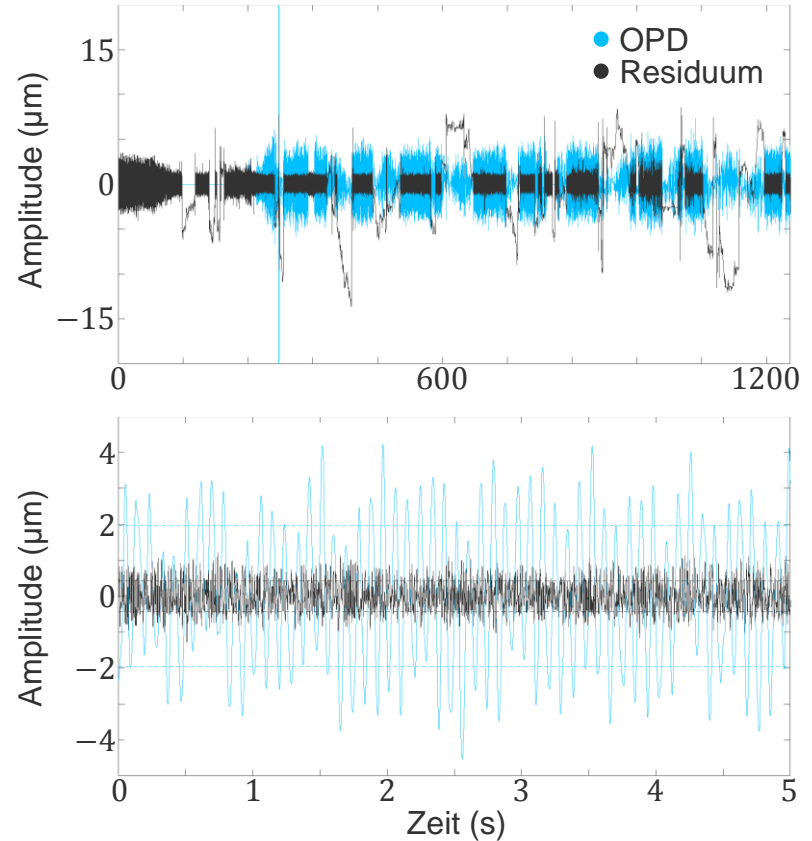
OPD Estimation

Results

Results

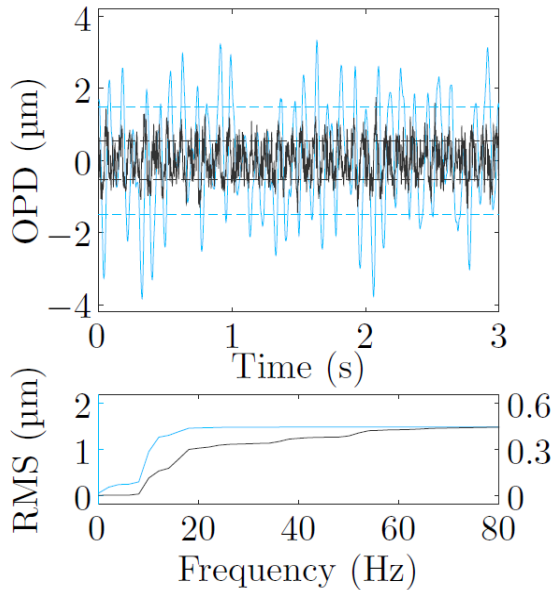
Measurements using LBTI (1/2)

- OPD reduction from $1.9545\mu\text{m}$ (RMS) to $0.4213\mu\text{m}$ (RMS) (-78.5%)
- Problems with malfunctional sensor channel (fixed now)
- LBTI Phasecam feedback gain reduced by a factor of 3
 - Permits longer integration times

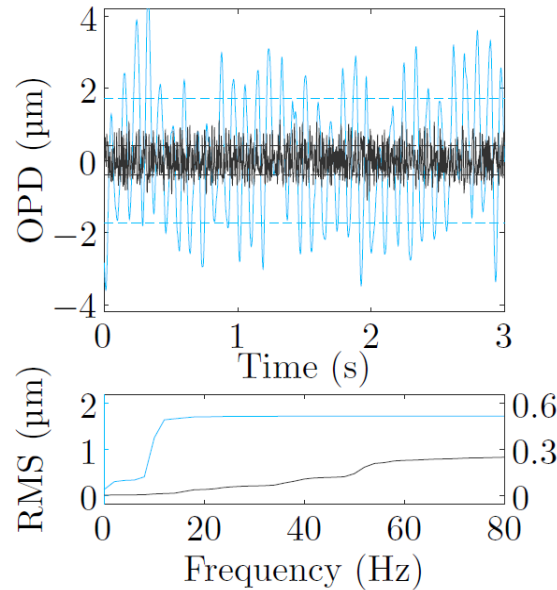


Results

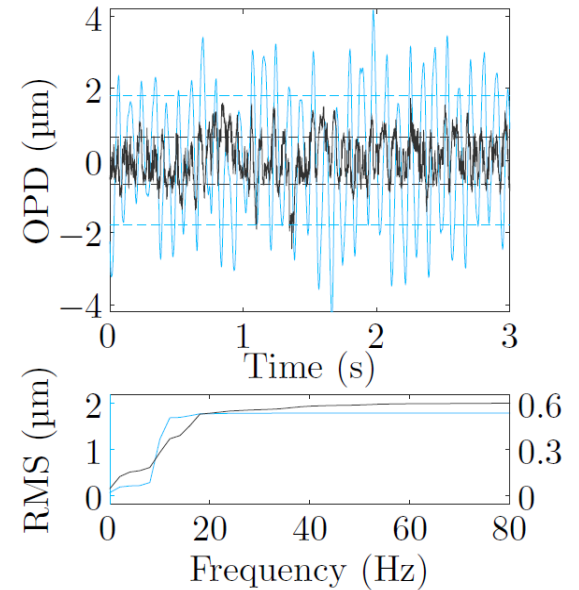
Measurements using LBTI (2/2)



(a) OVMS⁺ OFF, $K_i = 300$



(b) OVMS⁺ ON, $K_i = 300$



(c) OVMS⁺ ON, $K_i = 30$

Conclusion

Conclusion

- Estimation of mirror motion possible using a broad band filter
- Delay compensation up to 15ms using backstepping for PDEs
- Successful implementation at the LBT (called OVMS-plus)
 - Flexible change of parameters using config-files
 - Variable output-modes allow for flexible use
 - Delay can be individually set at service start-up
- Routine use by LBTO engineers and staff and LBTI instrument scientists and astronomers
- Proved very valuable for LBTI (and potentially for LINC-NIRVANA)



University of Stuttgart
Germany

Thank you!



Dipl.-Ing. Michael Böhm

e-mail boehm@isys.uni-stuttgart.de

phone +49 (0) 711 685-66291

fax +49 (0) 711 685-56291

University of Stuttgart
Institute for System Dynamics
Waldburgstraße 17/19, 70563 Stuttgart