

Use of Kalman filters in time and frequency analysis

John Davis
1st May 2011

Overview of Talk

- Simple example of Kalman filter
- Discuss components and operation of filter
- Models of clock and time transfer noise and deterministic properties
- Examples of use of Kalman filter in time and frequency

Predictors, Filters and Smoothing Algorithms

- Predictor: *predicts* parameter values ahead of current measurements
- Filter: *estimates* parameter values using current and previous measurements
- Smoothing Algorithm: *estimates* parameter values using future, current and previous measurements

What is a Kalman Filter ?

- A Kalman filter is an optimal recursive data processing algorithm
- The Kalman filter incorporates all information that can be provided to it. It processes all available measurements, regardless of their precision, to estimate the current value of the variables of interest
- Computationally efficient due to its recursive structure
- Assumes that variables being estimated are time dependent

Simple Example

Simple Example (Including Estimates)

Estimation Errors and Uncertainty

Operation of Kalman Filter

- Provides an estimate of the current parameters using current measurements and previous parameter estimates
- Should provide a close to optimal estimate if the models used in the filter match the physical situation
- World is full of badly designed Kalman filters

What are the applications in time and frequency?

- Analysis of time transfer / clock measurements where measurement noise is significant
- Clock and time scale predictors
- Clock ensemble algorithms
- Estimating noise processes
- Clock and UTC steering algorithms
- GNSS applications

Starting point for iteration 11 of filter

Five steps in the operation of a Kalman filter

- (1) State Vector propagation
- (2) Parameter Covariance Matrix propagation
- (3) Compute Kalman Gain
- (4) State Vector update
- (5) Parameter Covariance Matrix update

Components of the Kalman Filter (1)

State vector x

- Contains required parameters. These parameters usually contain deterministic and stochastic components.
- Normally include: time offset, normalised frequency offset, and linear frequency drift between two clocks or timescales.
- State vector components may also include: Markov processes where we have memory within noise processes and also components of periodic instabilities

Components of the Kalman Filter (2)

State Propagation Matrix $\Phi(\tau)$

- *Matrix used to extrapolate the deterministic characteristics of the state vector from time t to $t+\tau$*

Steps in the operation of the Kalman Filter

Step 1: State vector propagation from iteration n-1 to n

$$\underline{\hat{x}}^-(t_n) = \underline{\underline{\Phi}}(\tau) \underline{\hat{x}}^+(t_{n-1})$$

$\underline{\hat{x}}^-(t_n)$ Estimate of the state vector at iteration n not including measurement n.

$\underline{\underline{\Phi}}(\tau)$ State propagation matrix calculated for time spacing τ

$\underline{\hat{x}}^+(t_{n-1})$ Estimate of the state vector at iteration n-1 including measurement n-1.

Step 1: State Vector Extrapolation

Components of the Kalman Filter (3)

Parameter Covariance Matrix P

- Contains estimates of the uncertainty and correlation between uncertainties of state vector components. Based on information supplied to the Kalman filter. Not obtained from measurements.

Process Covariance Matrix $Q(\tau)$

- *Matrix describing instabilities of components of the state vector, e.g. clock noise, time transfer noise moving from time t to $t+\tau$*

Noise parameters σ

- *Parameters used to determine the elements of the process covariance matrix, describe individual noise processes.*

Steps in the operation of the Kalman Filter

Step 2: Parameter covariance matrix propagation from iteration n-1 to n

$$\underline{\underline{P}}^-(t_n) = \underline{\underline{\Phi}}(\tau) \underline{\underline{P}}^+(t_{n-1}) \underline{\underline{\Phi}}^T(\tau) + \underline{\underline{Q}}(t_n)$$

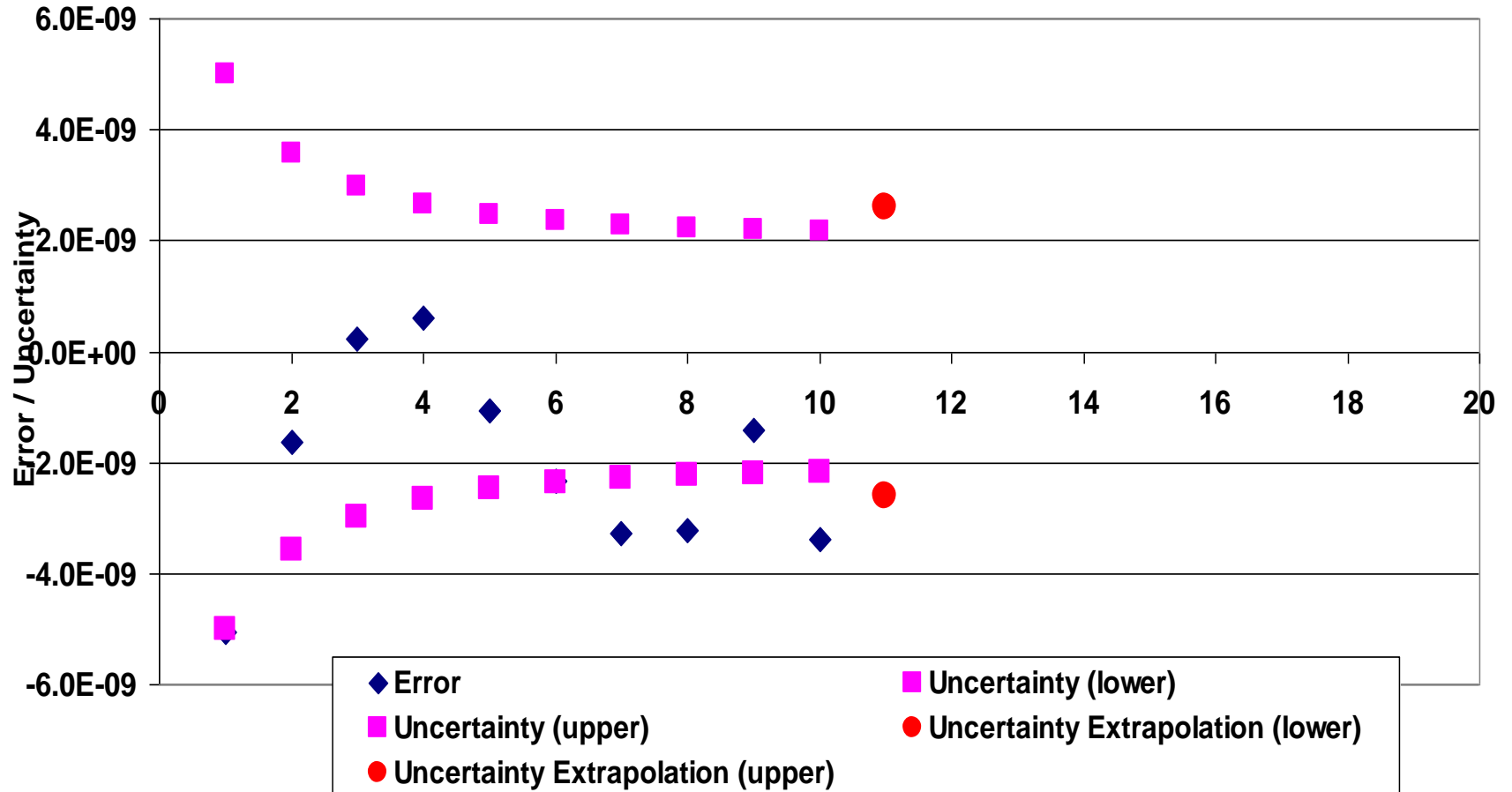
Parameter covariance matrix, at iteration n-1 including measurement n-1

$\underline{\underline{\Phi}}(\tau)$ $\underline{\underline{\Phi}}^T(\tau)$ State propagation matrix and transpose calculated for time spacing τ

$\underline{\underline{P}}^-(t_n)$ Parameter covariance matrix, at iteration n not including measurement n

$\underline{\underline{Q}}(t_n)$ Process covariance matrix

Uncertainty Extrapolation



Components of the Kalman Filter (4)

Design Matrix H

- Matrix that relates the measurement vector and the state vector using $y = H x$

Measurement Covariance Matrix R

- *Describes measurement noise, white but individual measurements may be correlated. May be removed.*

Kalman Gain K

- *Determines the weighting of current measurements and estimates from previous iteration. Computed by filter.*

Steps in the operation of the Kalman Filter

Step 3 :Kalman gain computation

$$\underline{\underline{K}}(t_n) = \underline{\underline{P}}^-(t_n) \underline{\underline{H}}^T \left[\underline{\underline{H}} \underline{\underline{P}}^-(t_n) \underline{\underline{H}}^T + \underline{\underline{R}}(t_n) \right]^{-1}$$

$\underline{\underline{P}}^-(t_n)$ Parameter covariance matrix, at iteration n not including measurement n

$\underline{\underline{H}}$ $\underline{\underline{H}}^T$ Design Matrix and Transpose

$\underline{\underline{K}}$ Kalman Gain

$\underline{\underline{R}}(t_n)$ Measurement covariance matrix

Components of the Kalman Filter (5)

Measurement Vector y

- Vector containing measurements input during a single iteration of the filter

Identity Matrix I

Steps in the operation of the Kalman Filter

Step 4 : State vector update

$$\hat{\underline{x}}^+(t_n) = \hat{\underline{x}}^-(t_n) + \underline{\underline{K}}(t_n) \left[\underline{y}(t_n) - \underline{\underline{H}} \hat{\underline{x}}^-(t_n) \right]$$

$\hat{\underline{x}}^+(t_n)$ $\hat{\underline{x}}^-(t_n)$ Estimate of the state vector at iteration n
including and not including measurement n respectively.

Kalman Gain

$\underline{\underline{y}}(t_n)$ Measurement Vector

$\underline{\underline{H}}$ Design Matrix

Step 4 :State Vector Update

Steps in the operation of the Kalman Filter

Step 5 :Parameter Covariance Matrix update

$$\underline{\underline{P}}(t_n^+) = \left[\underline{\underline{I}} - \underline{\underline{K}}(t_n) \underline{\underline{H}} \right] \underline{\underline{P}}(t_n^-)$$

$\underline{\underline{K}}(t_n)$ Kalman Gain

$\underline{\underline{H}}$ Design Matrix

$\underline{\underline{P}}(t_n^+)$ $\underline{\underline{P}}(t_n^-)$ Parameter Covariance Matrix, at iteration n, including and not including measurement n respectively

$\underline{\underline{I}}$ Identity Matrix

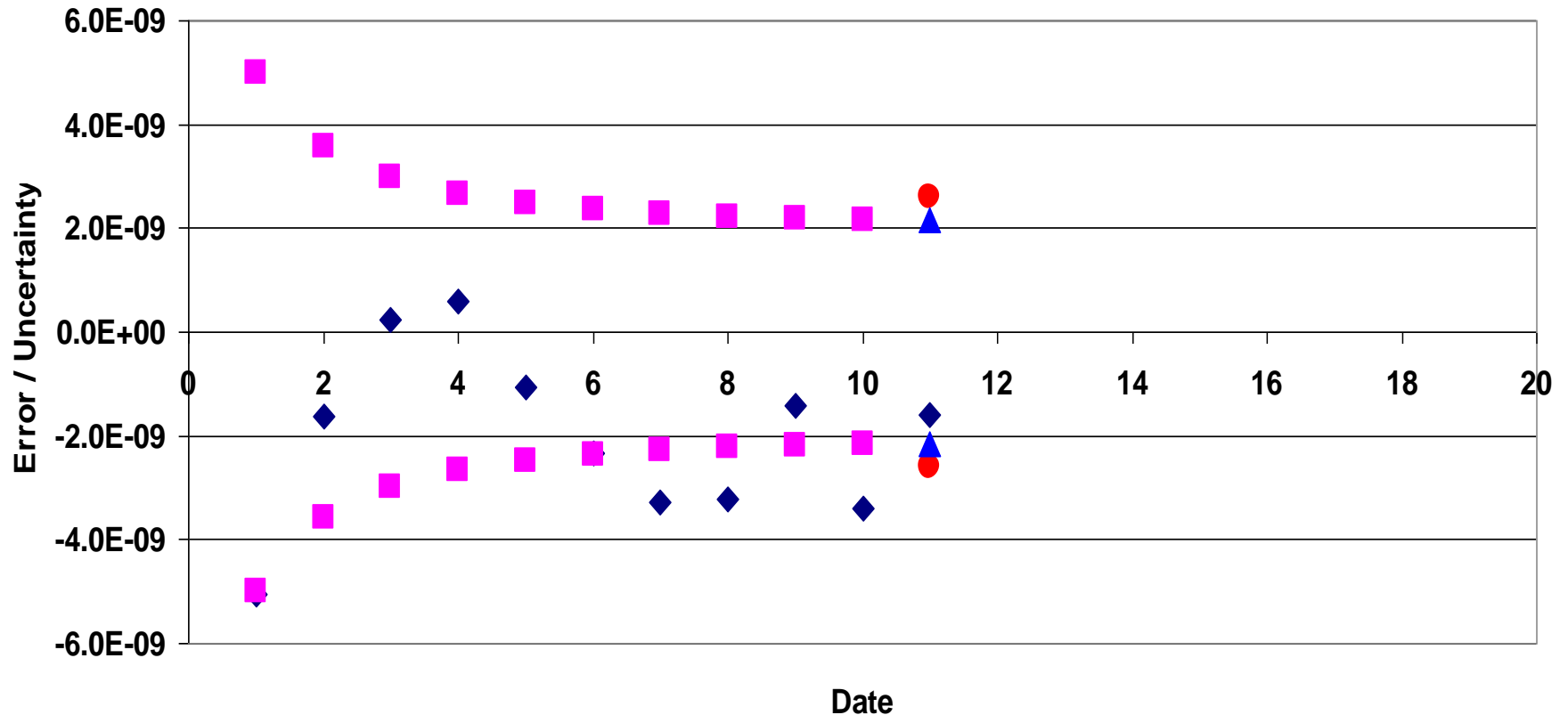
Steps in the operation of the Kalman Filter

If Kalman gain is deliberately set sub-optimal, then use

$$\underline{\underline{P}}(t_n^+) = [\underline{\underline{I}} - \underline{\underline{K}}(t_n)\underline{\underline{H}}]\underline{\underline{P}}(t_n^-)[\underline{\underline{I}} - \underline{\underline{K}}(t_n)\underline{\underline{H}}]^T$$

$\underline{\underline{K}}$ Kalman Gain
 $\underline{\underline{H}}$ Design Matrix
 $\underline{\underline{P}}$ Parameter Covariance Matrix, at
 $\underline{\underline{I}}$ iteration n , including and not including measurement
 n respectively
 $\underline{\underline{I}}$ Identity Matrix

Step 5: Uncertainty Update



◆ Error ■ Uncertainty (lower) ■ Uncertainty (upper)
 ● Uncertainty Extrapolation (lower) ● Uncertainty Extrapolation (upper) ▲ Uncertainty Update (lower)
 ▲ Uncertainty update (upper)

Constructing a Kalman Filter: Lots to think about!

- Choice of State Vector components
- Choice of Noise Parameters
- Choice of measurements
- Description of measurement noise
- Choice of Design Matrix
- Choice of State Propagation Matrix
- Initial set up conditions
- Dealing with missing data
- Testing out the filter
- Computational methods

Initialising the Kalman Filter

- Set the initial parameters to physically realistic values
- If you have no information set diagonal values of Parameter Covariance Matrix high, filter will then give a high weight to the first measurement.
- Set Parameters with known variances e.g. Markov components of Parameter Covariance Matrices to their steady state values.
- Set Markov State Vector components and periodic elements to zero.
- Elements of \mathbf{H} , $\Phi(\tau)$, $\mathbf{Q}(\tau)$, \mathbf{R} are pre-determined

Simulating Noise and Deterministic Processes

- Use the same construction of state vector components \mathbf{x} , state propagation matrix $\Phi(\tau)$, parameter covariance matrix \mathbf{P} , process covariance matrix $\mathbf{Q}(\tau)$, design matrix \mathbf{H} and measurement covariance matrix \mathbf{R} .
- Simulate all required data sets
- Simulate “true” values of state vector for error analysis
- May determine ADEV, HDEV, MDEV and TDEV statistics from parameter covariance matrix

Simulating Noise and Deterministic Processes

$$\underline{x}(t_n) = \underline{\Phi}(\tau) \underline{x}(t_{n-1}) + \underline{L}_Q(t_n) \underline{e}(t_n)$$

Where $\underline{Q} = \underline{L}_Q(t_n) \underline{L}_Q^T(t_n)$

\underline{Q} = Process Covariance Matrix

$\underline{x}(t_n)$ $\underline{x}(t_{n-1})$ = State Vector at iteration n and n-1 respectively

$\underline{\Phi}(\tau)$ = State Propagation Matrix calculated for time spacing τ

$\underline{e}(t_n)$ = Vector of normal distribution noise of unity magnitude

Simulating Noise and Deterministic Processes

$$\underline{\underline{P}}(t_n) = \underline{\underline{\Phi}}(\tau) \underline{\underline{P}}(t_{n-1}) \underline{\underline{\Phi}}^T(\tau) + \underline{\underline{Q}}(t_n)$$

$\underline{\underline{P}}(t_n)$ $\underline{\underline{P}}(t_{n-1})$ Process Covariance Matrix at iterations n and n-1 respectively

$\underline{\underline{\Phi}}(\tau)$ $\underline{\underline{\Phi}}^T(\tau)$ State Propagation Matrix and transpose computed at time spacing τ .

$\underline{\underline{Q}}(t_n)$ Process Covariance Matrix calculated at time spacing τ

Simulating Noise and Deterministic Processes

$$\underline{y}(t_n) = \underline{\underline{H}} \underline{x}(t_n) + \underline{\underline{L}}_R(t_n) \underline{e}(t_n)$$

where $\underline{\underline{R}} = \underline{\underline{L}}_R(t_n) \underline{\underline{L}}_R^T(t_n)$

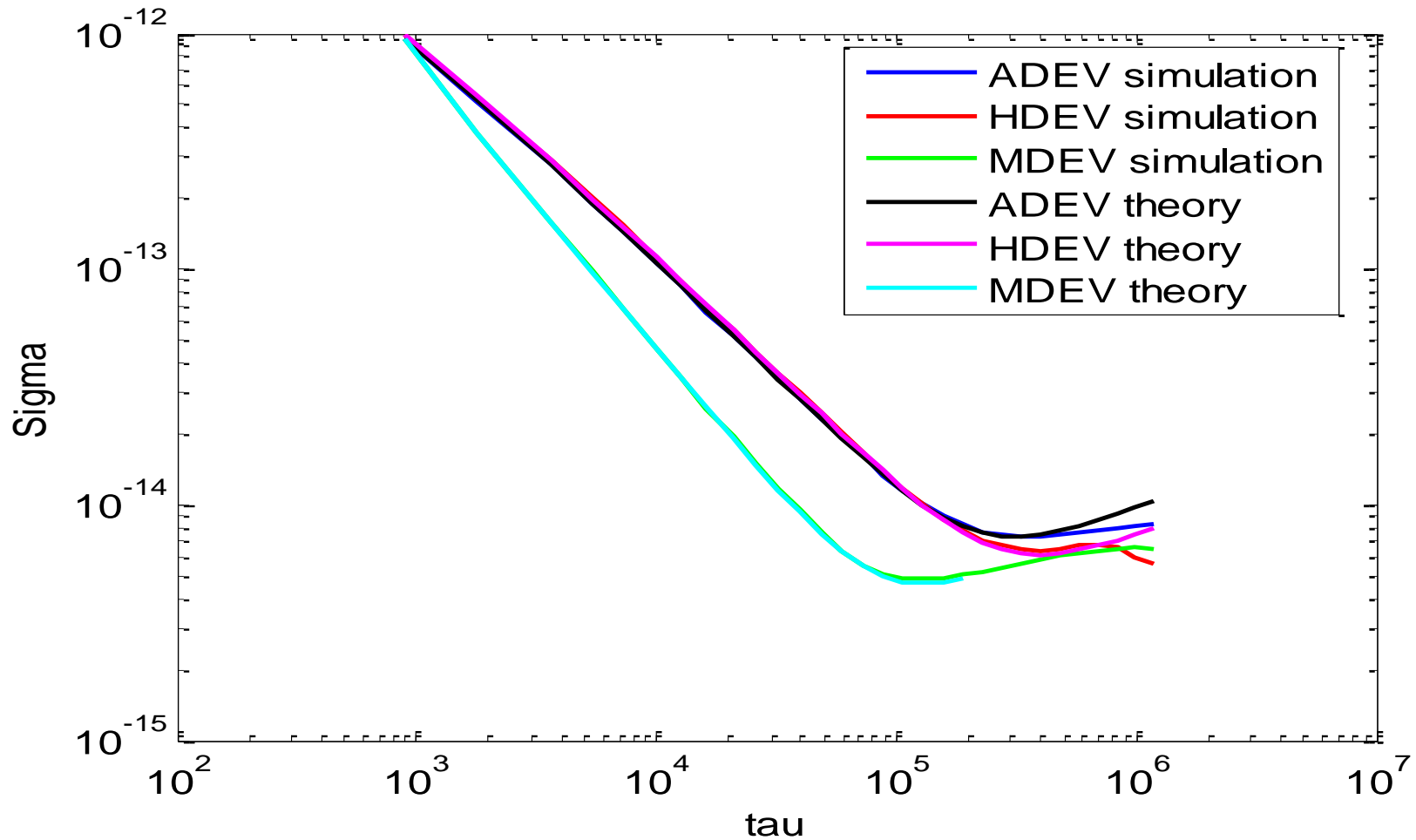
$\underline{y}(t_n)$ Measurement Vector

$\underline{\underline{H}}$ Design Matrix

$\underline{\underline{R}}$ Measurement Covariance Matrix

$\underline{e}(t_n)$ Vector of normal distribution noise of unity magnitude

ADEV, HDEV, and MDEV determined from Parameter Covariance Matrix and simulation



Simulating Noise and Deterministic Processes

- Test Kalman filter with a data set with same stochastic and deterministic properties as is expected by the Kalman filter.
- Very near optimal performance of Kalman filter under these conditions.

Deterministic Properties

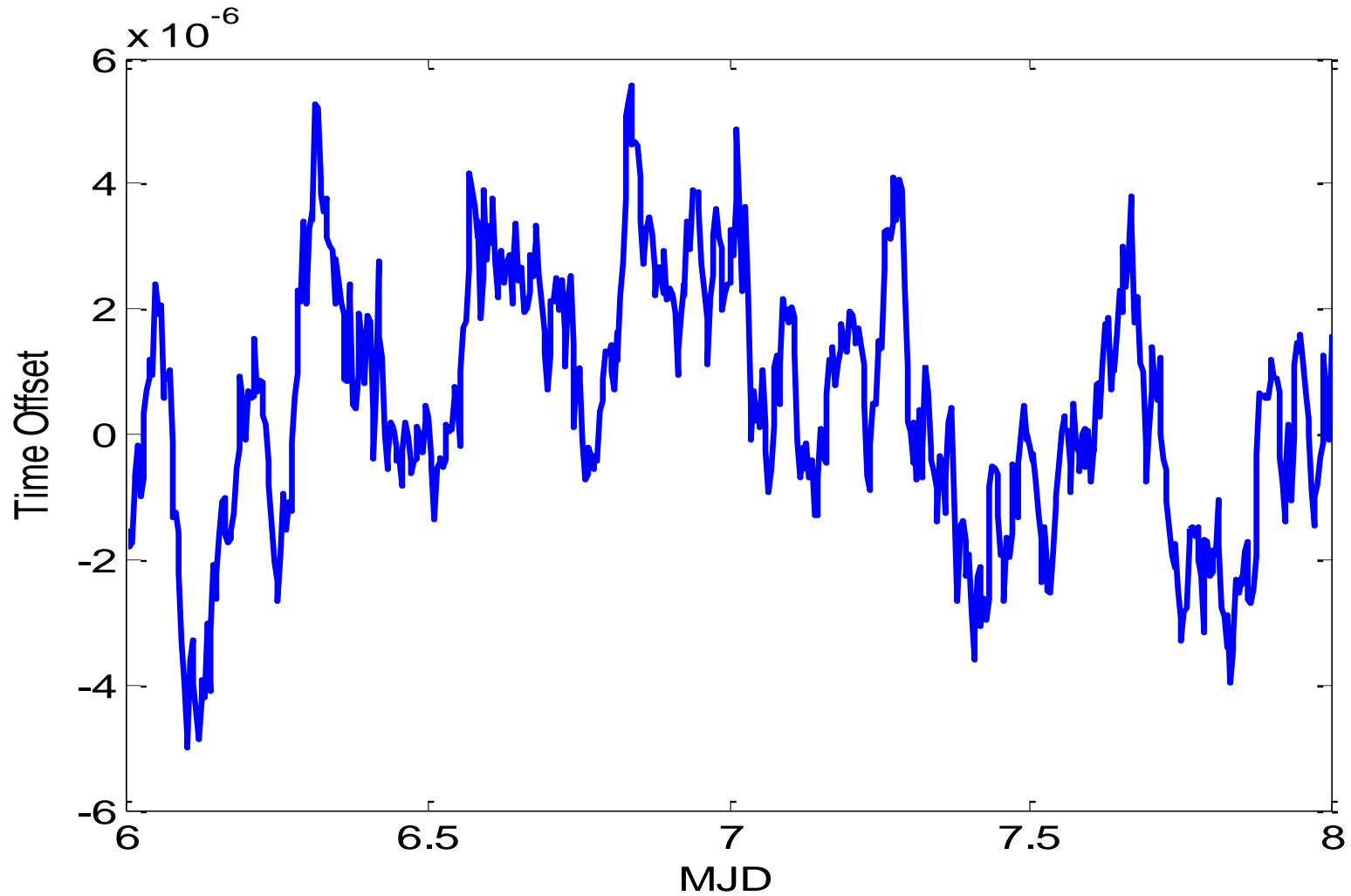
- Linear frequency drift
- Use three state vector components, time offset, normalised frequency offset and linear frequency drift

$$\Phi = \begin{pmatrix} 1 & \tau & \tau^2/2 & \cdots \\ 0 & 1 & \tau & \\ 0 & 0 & 1 & \\ \vdots & & & \ddots \end{pmatrix}$$

Markov Noise Process

- White Noise $x_n = e$
- Random Walk Noise $x_n = x_{n-1} + e$
- Markov Noise $x_n = kx_{n-1} + e \quad 0 < k < 1$
- Flicker noise, combination of Markov processes with continuous range of relaxation parameters k
- Markov noise processes and flicker noise have memory
- Kalman filters tend to work very well with Markov noise processes
- May only construct an approximation to flicker noise in a Kalman filter

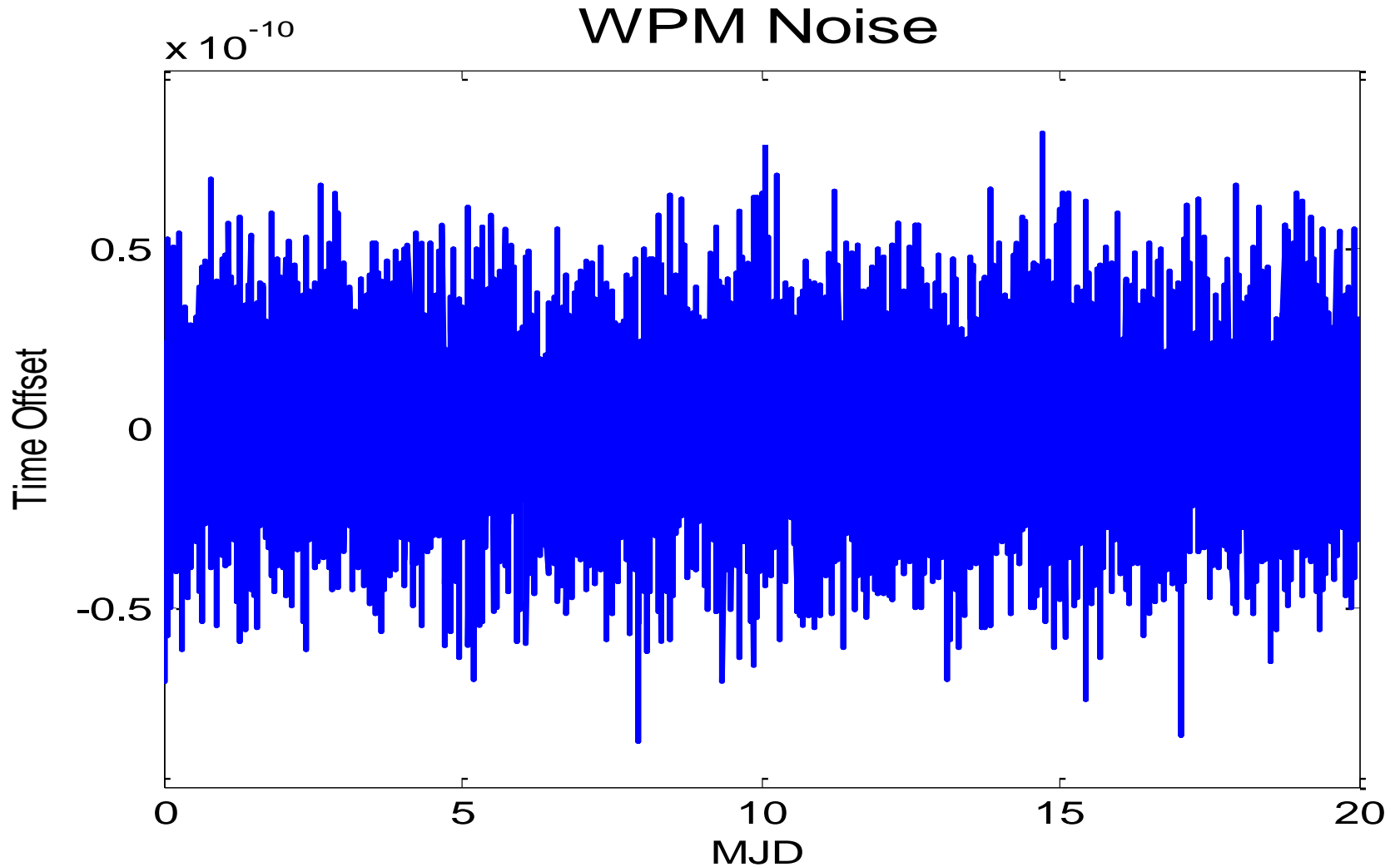
Markov Noise Process



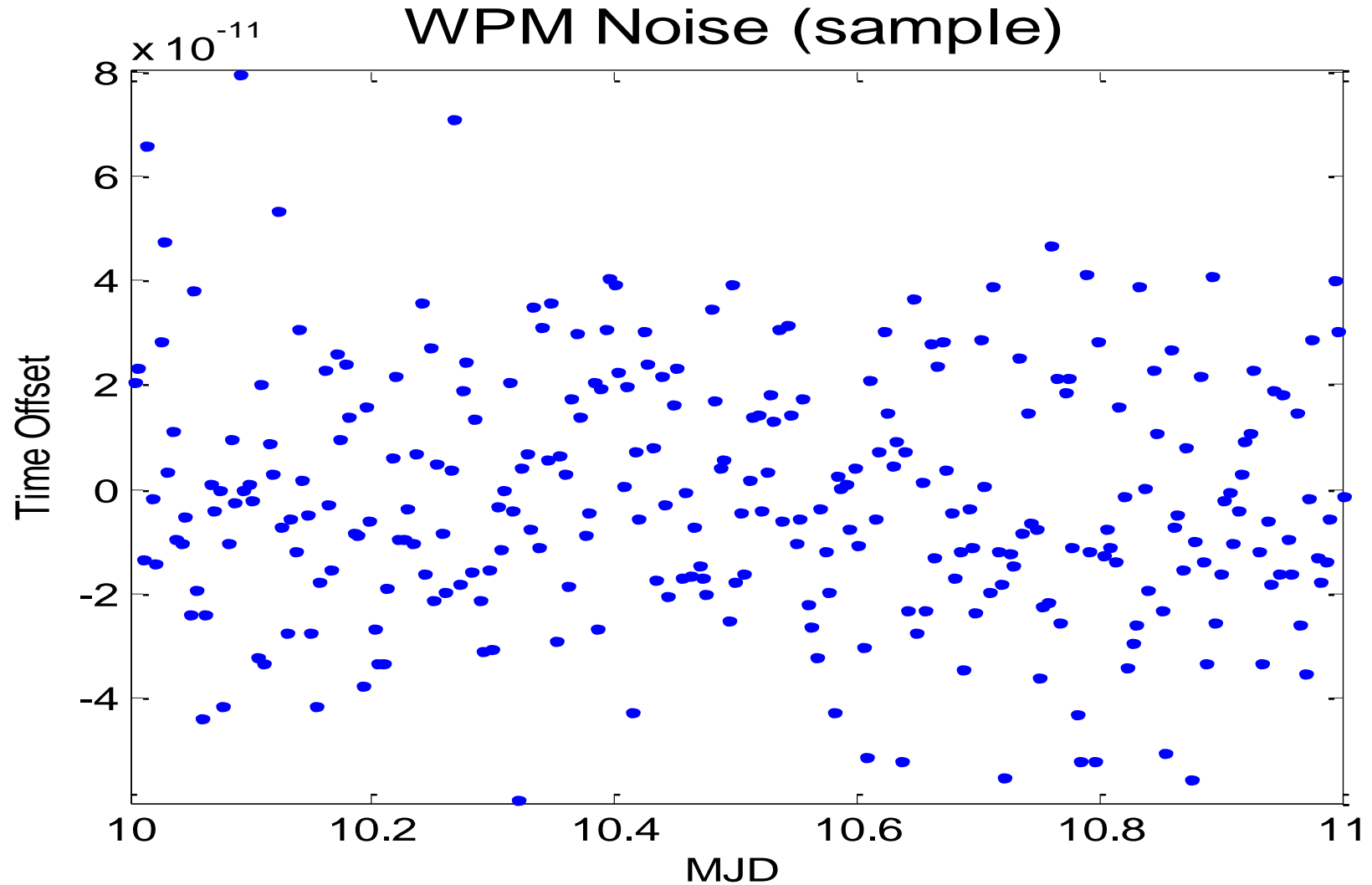
Describing Measurement Noise

- Kalman filter assumes measurement noise introduced via matrix **R** is white
- Time transfer noise e.g TWSTFT and GNSS noise is often far from white
- Add extra components to state vector to describe non white noise, particularly useful if noise has “memory” e.g. flicker phase noise
- NPL has not had any problems in not using a measurement noise matrix **R**

White Phase Modulation

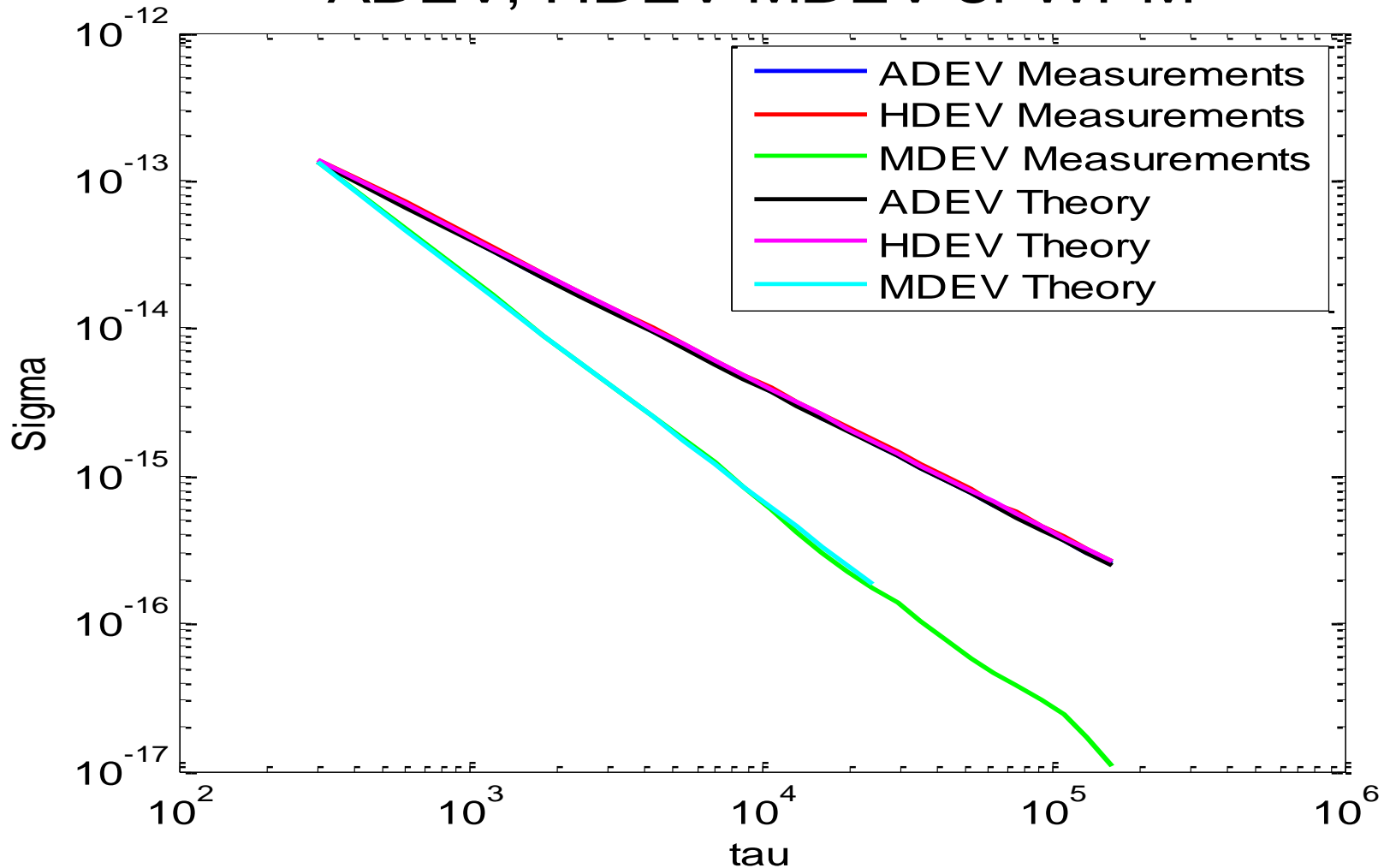


White Phase Modulation

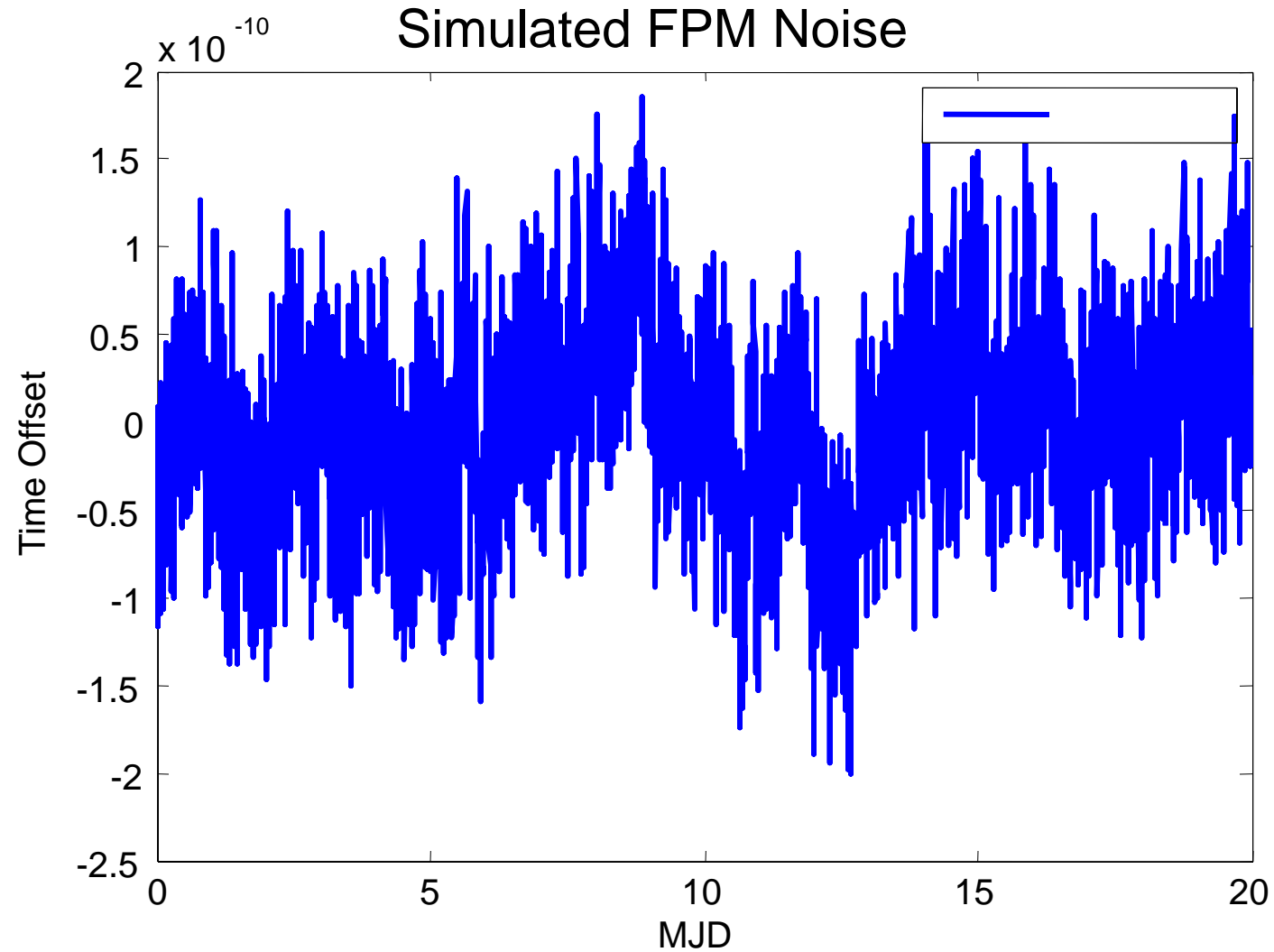


White Phase Modulation

ADEV, HDEV MDEV of WPM

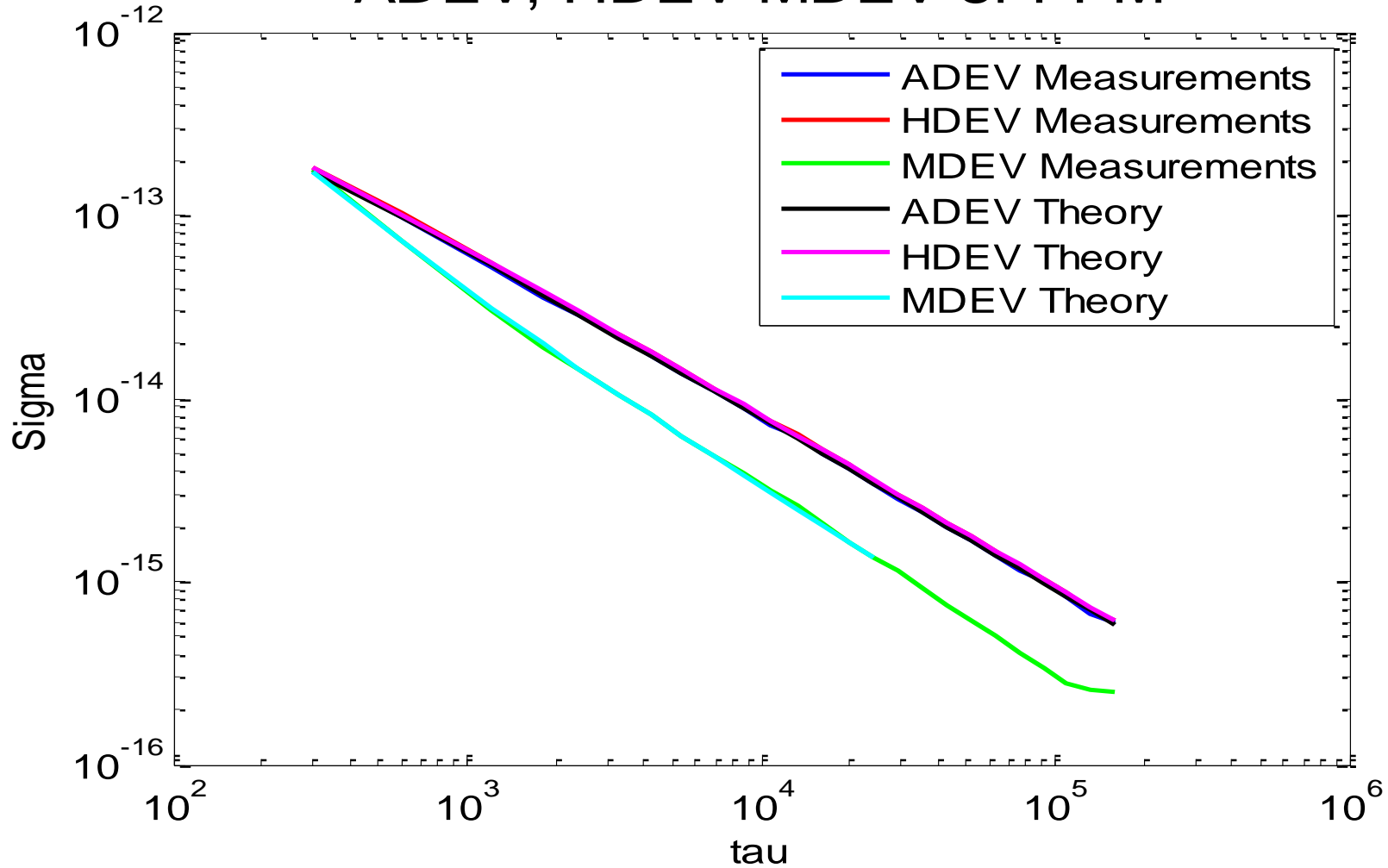


Flicker Phase Modulation

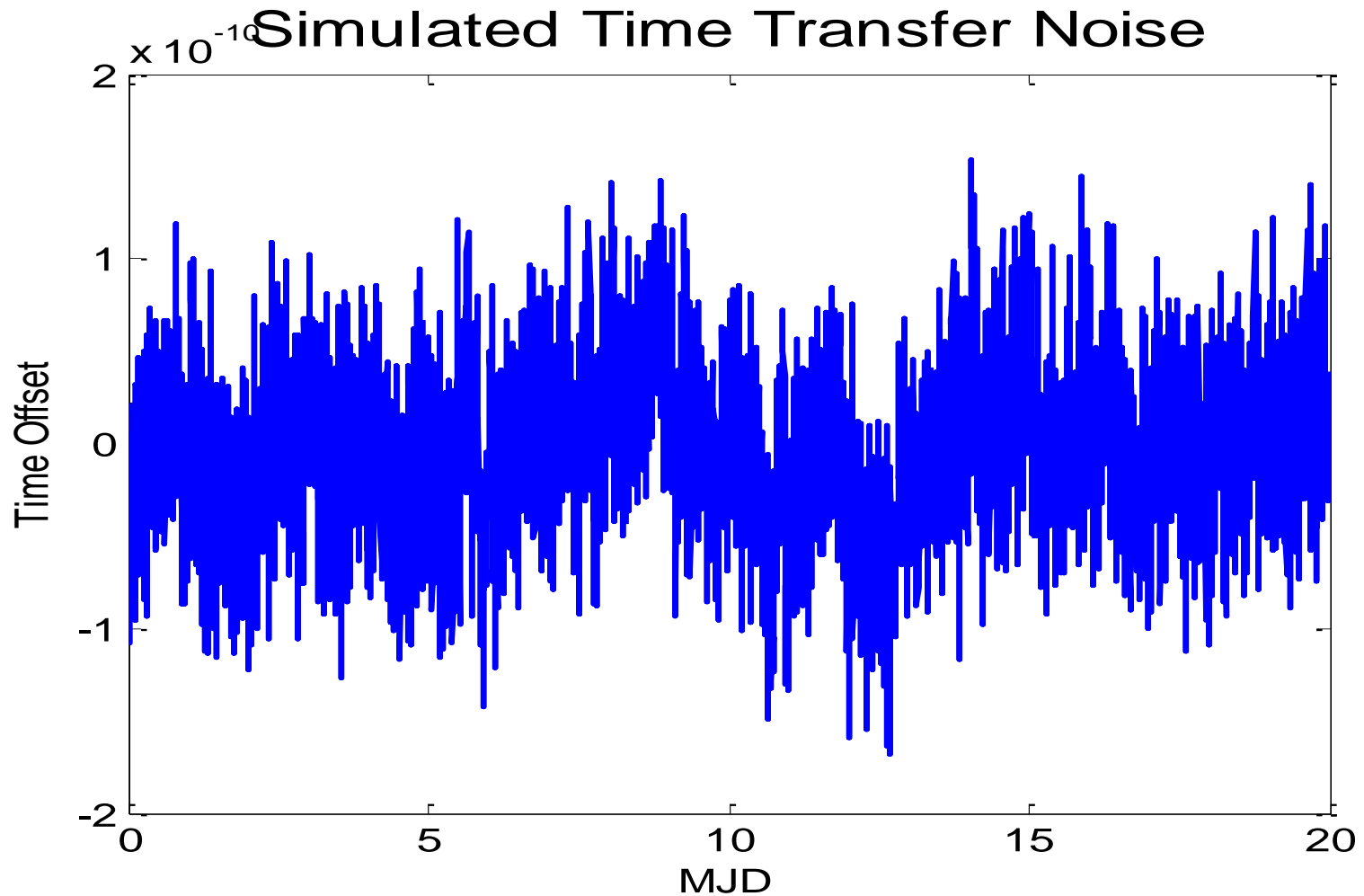


Flicker Phase Modulation

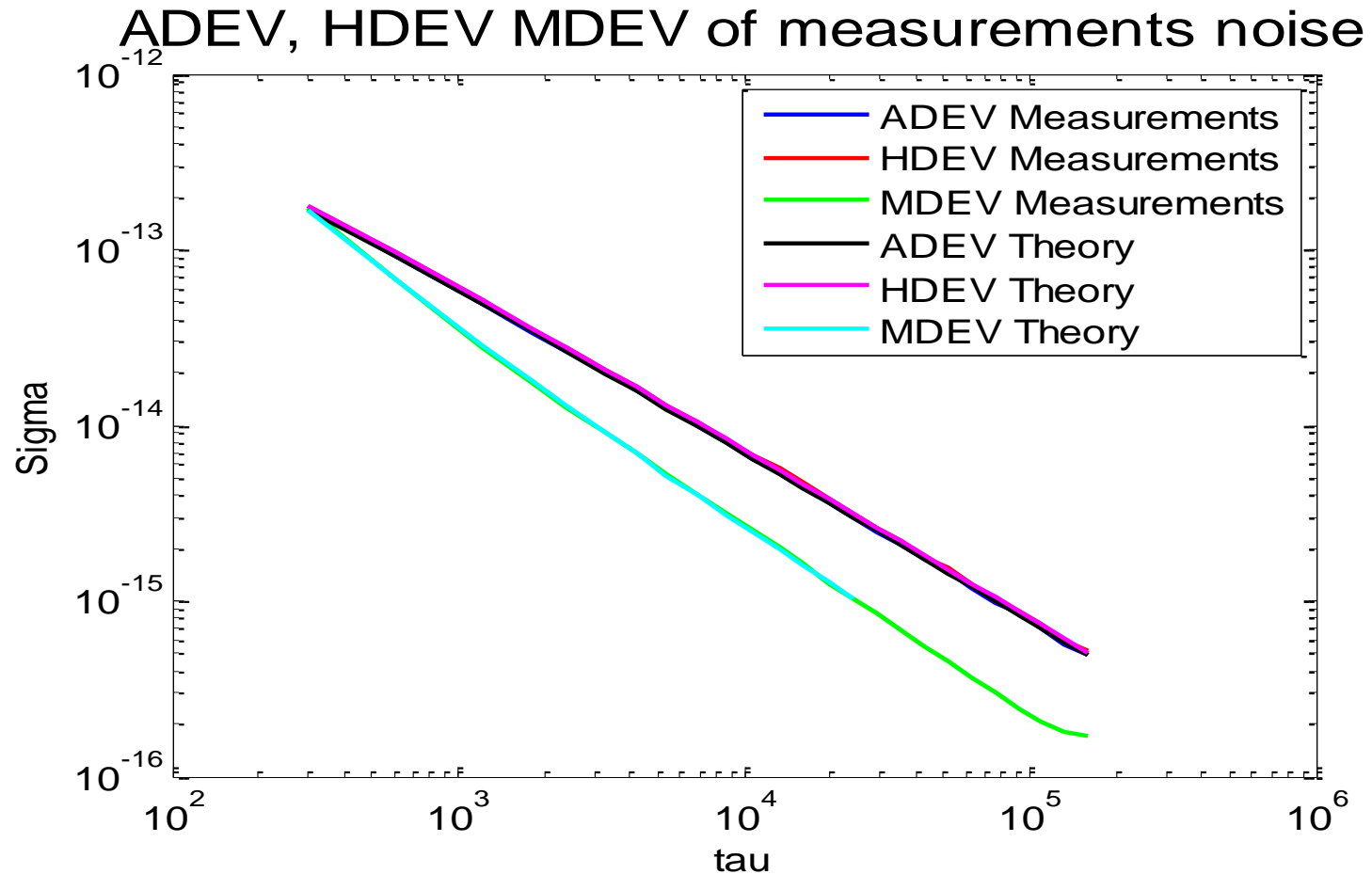
ADEV, HDEV MDEV of FFM



Measurement noise used in Kalman filter



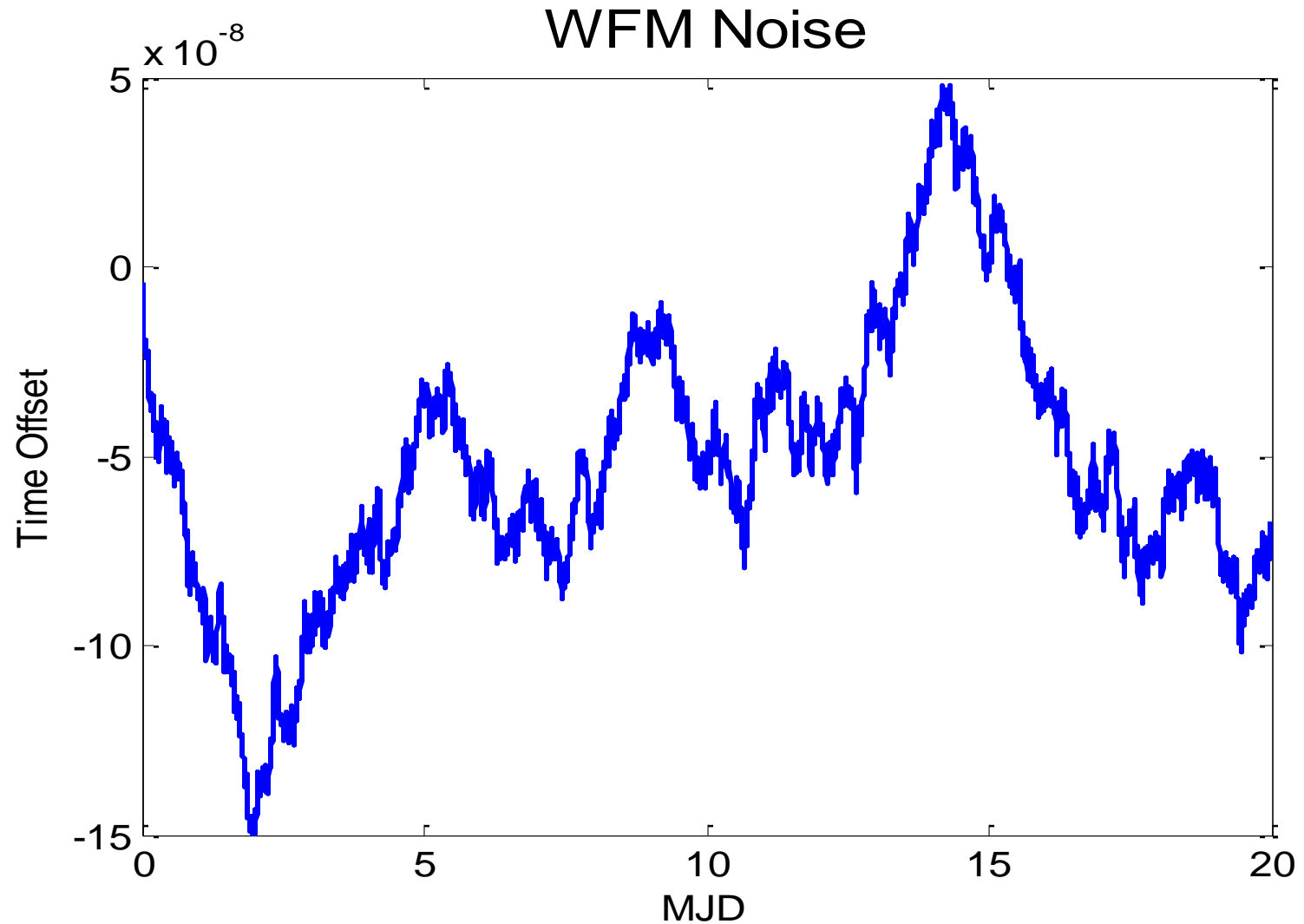
Measurement noise ADEV, MDEV and HDEV



Clock Noise

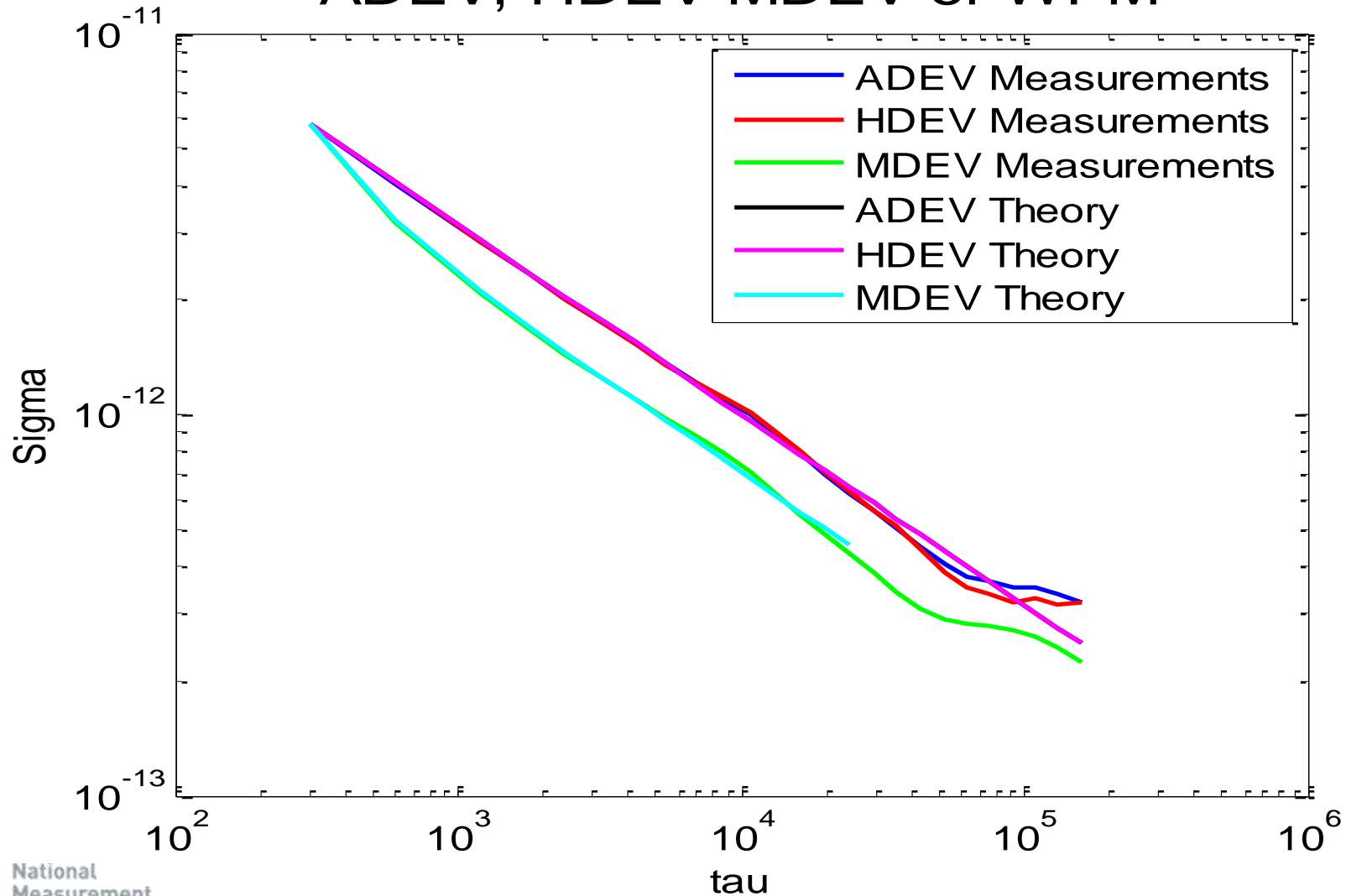
- White Frequency Modulation and Random Walk Frequency Modulation easy to include as single noise parameters.
- Flicker Frequency Modulation may be modelled as linear combination of Integrated Markov Noise Parameters

White Frequency Modulation

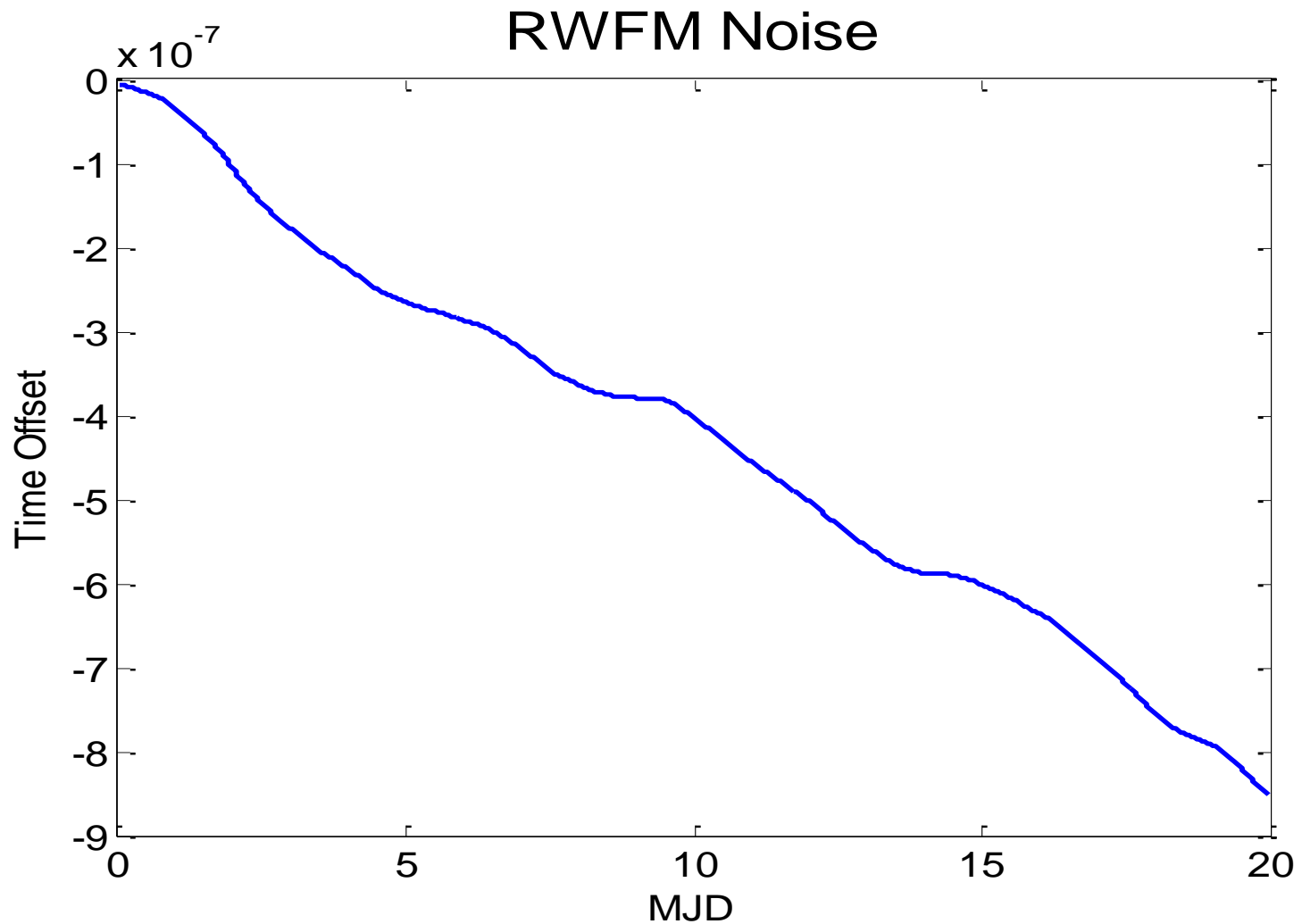


White Frequency Modulation

ADEV, HDEV MDEV of WFM

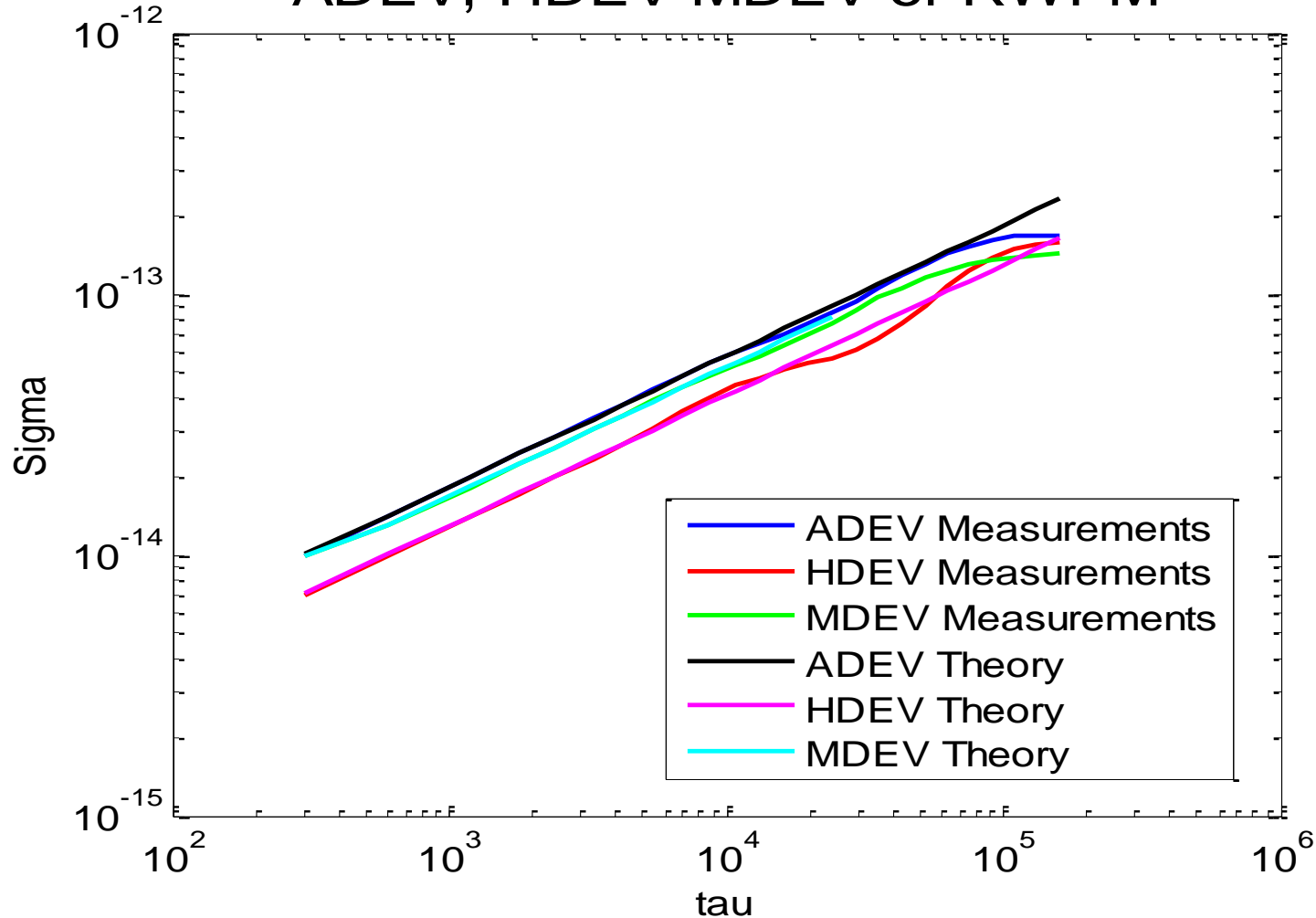


Random Walk Frequency Modulation



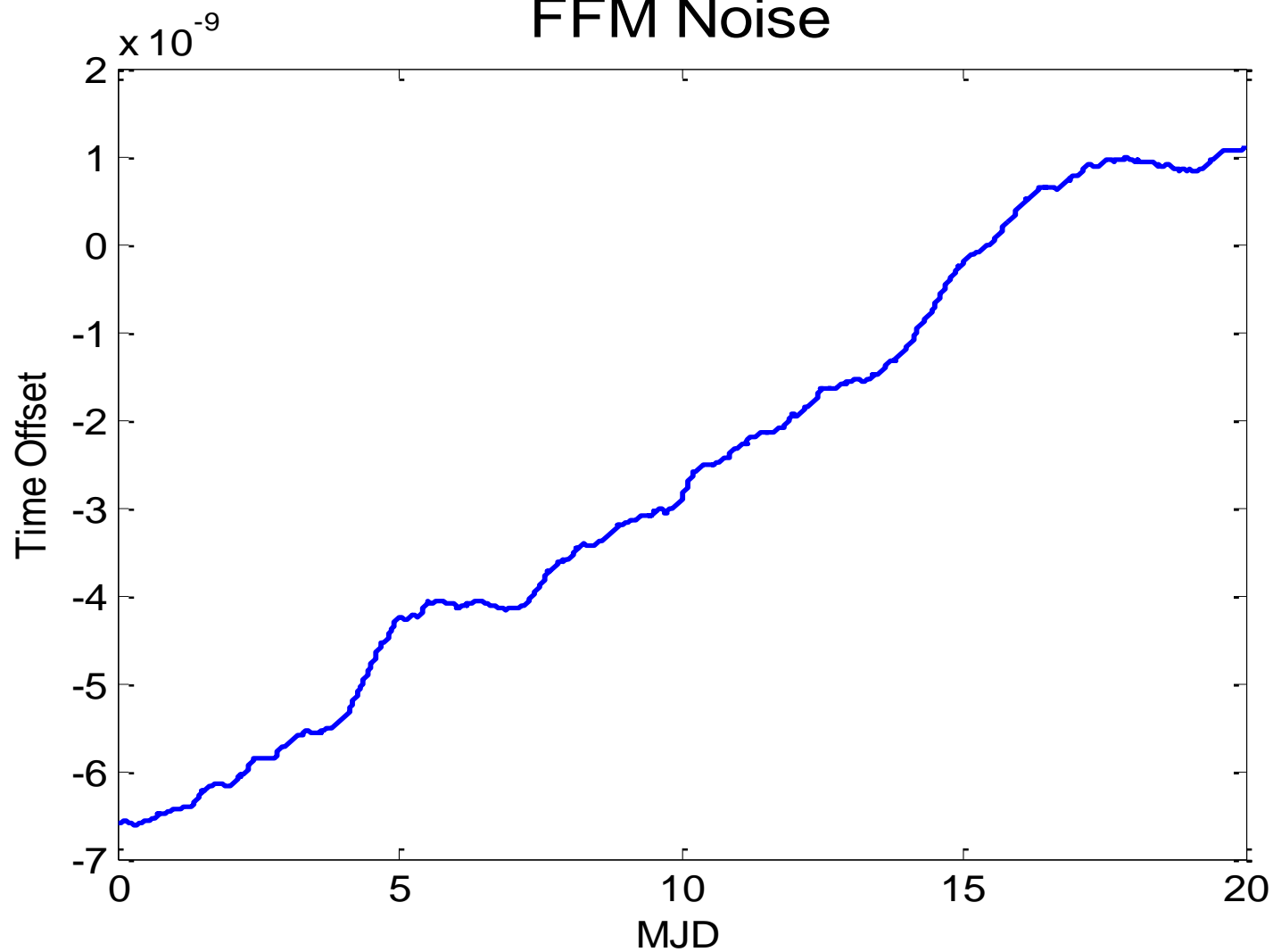
Random Walk Frequency Modulation

ADEV, HDEV MDEV of RWFM



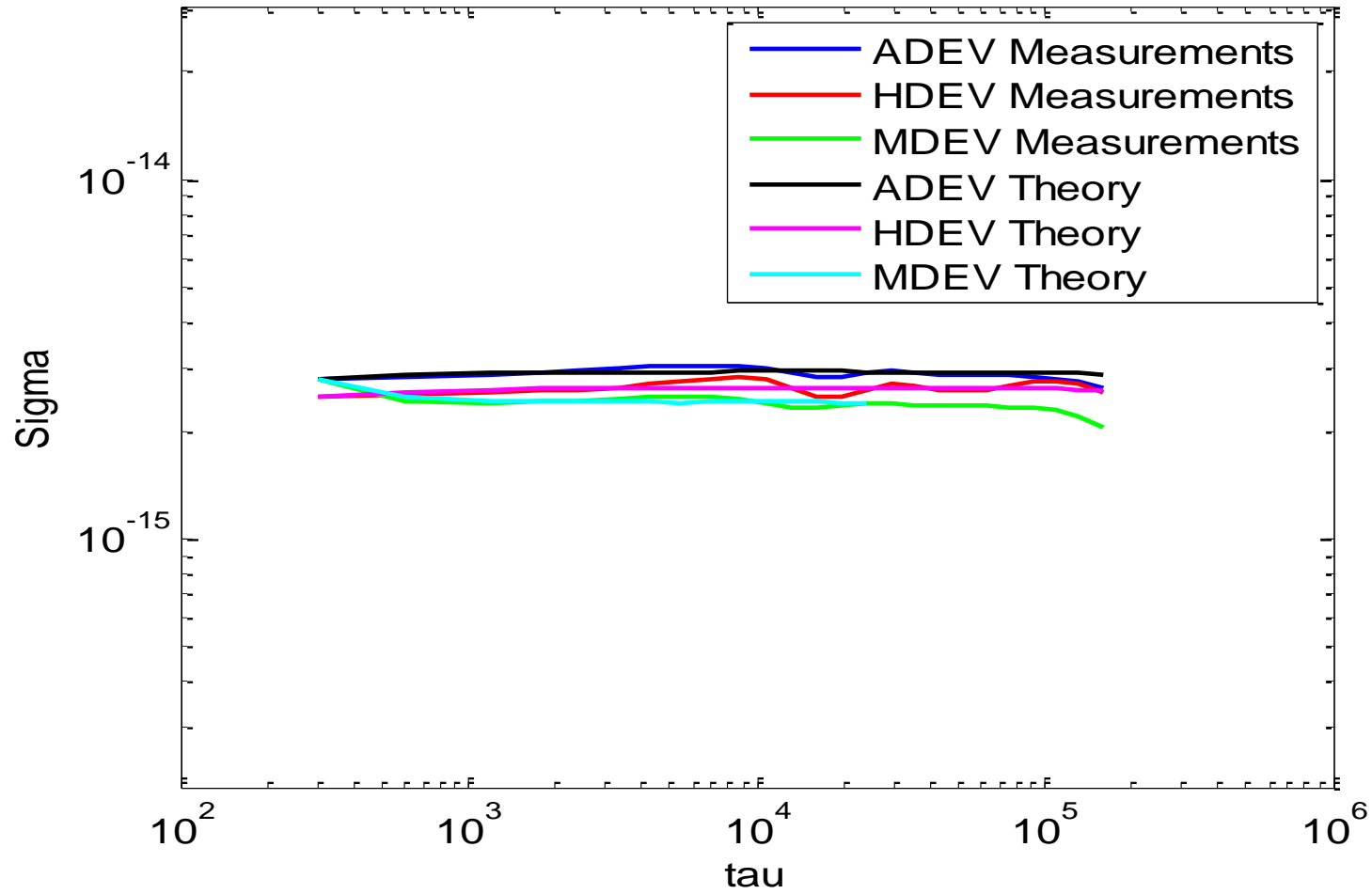
FFM Noise

FFM Noise



Flicker Frequency Modulation

ADEV, HDEV MDEV of FFM noise

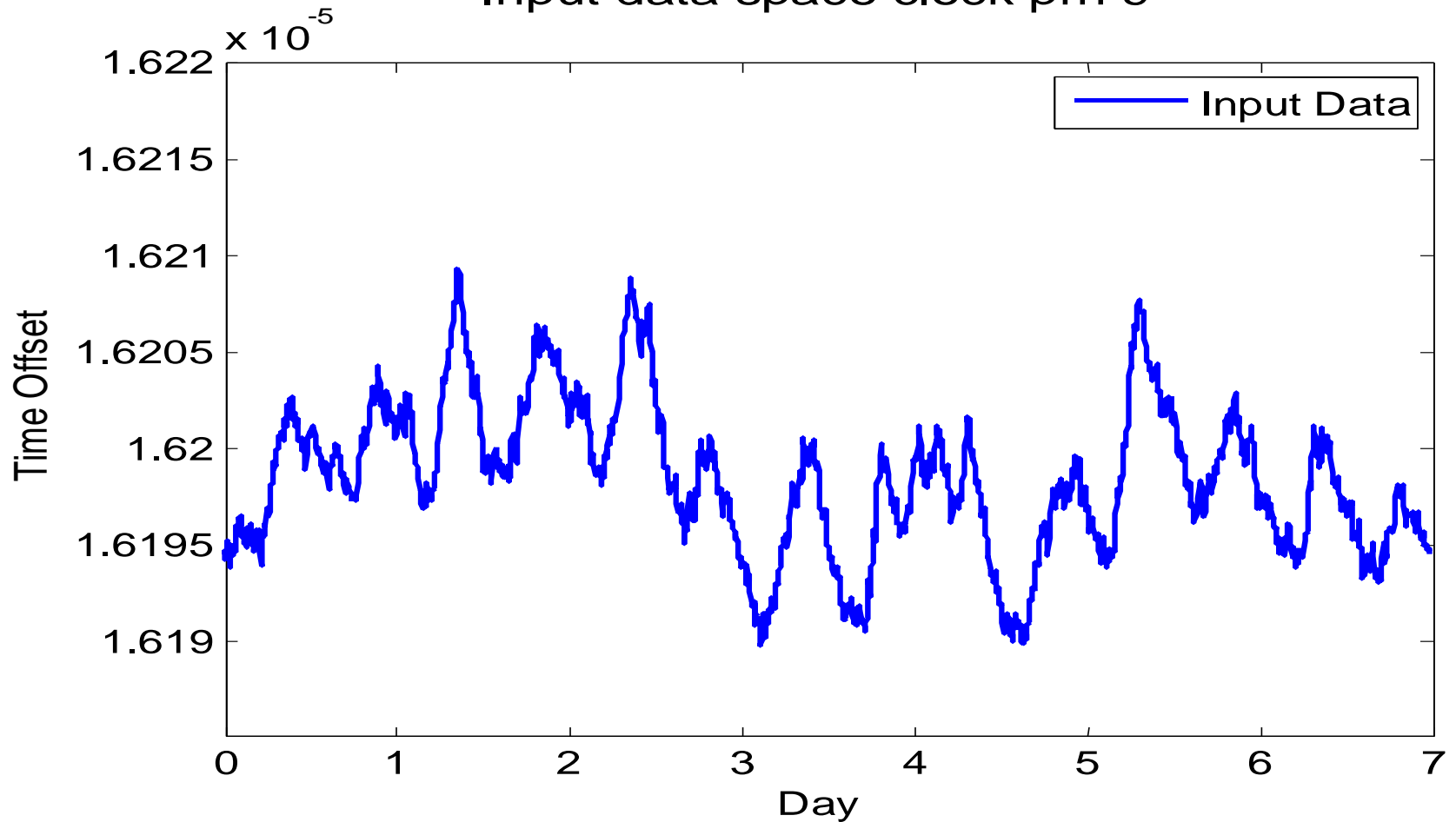


Periodic Instabilities

- Often occur in GNSS Applications
- Orbit mis-modelling
- Diurnal effects on Time Transfer hardware
- Periodicity decays over time
- Model as combination of Markov noise process and periodic parameter
- Two extra state vector component to represent periodic variations

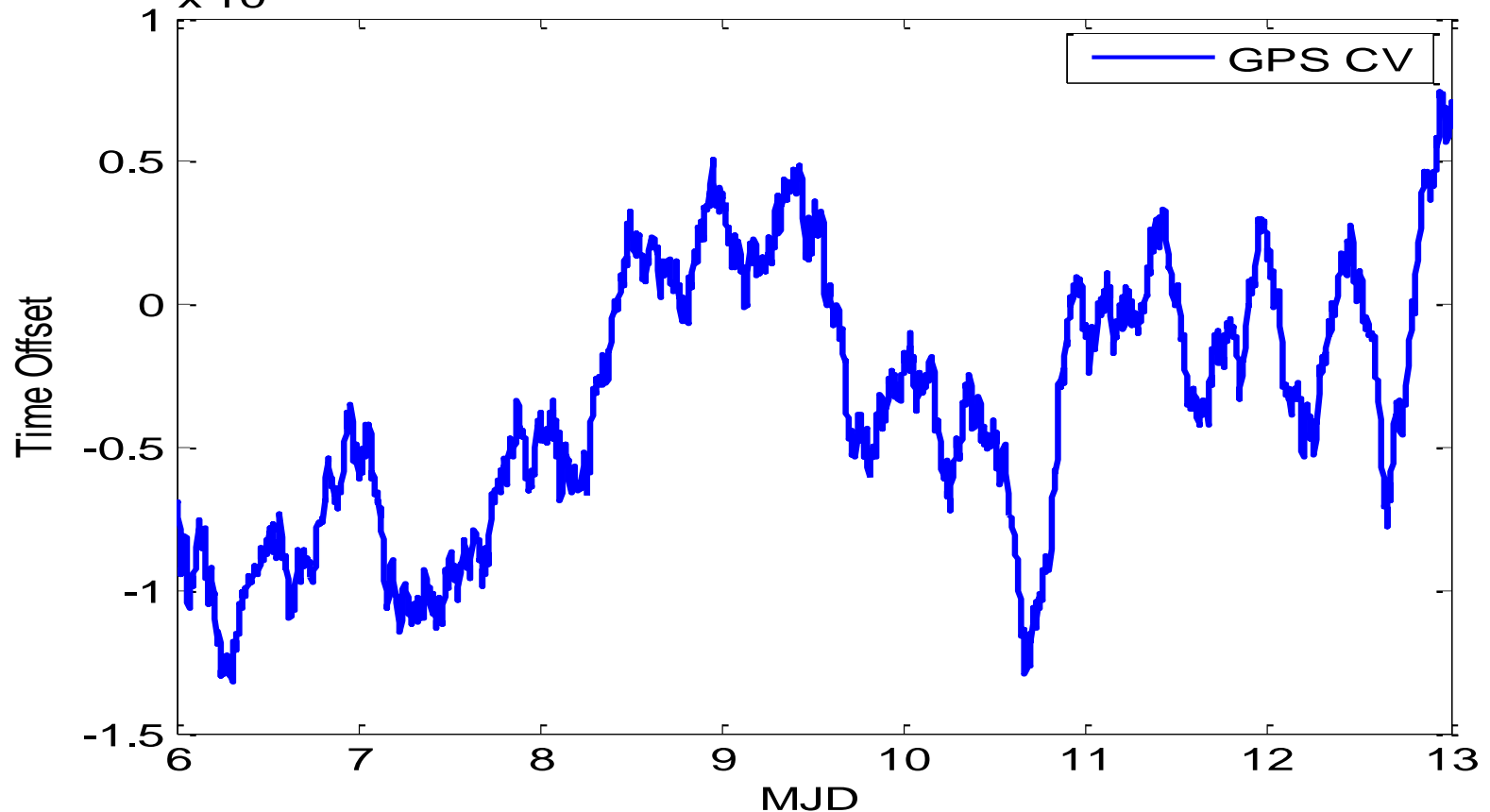
GPS Time – Individual Clock (Real Data)

Input data space clock prn 9

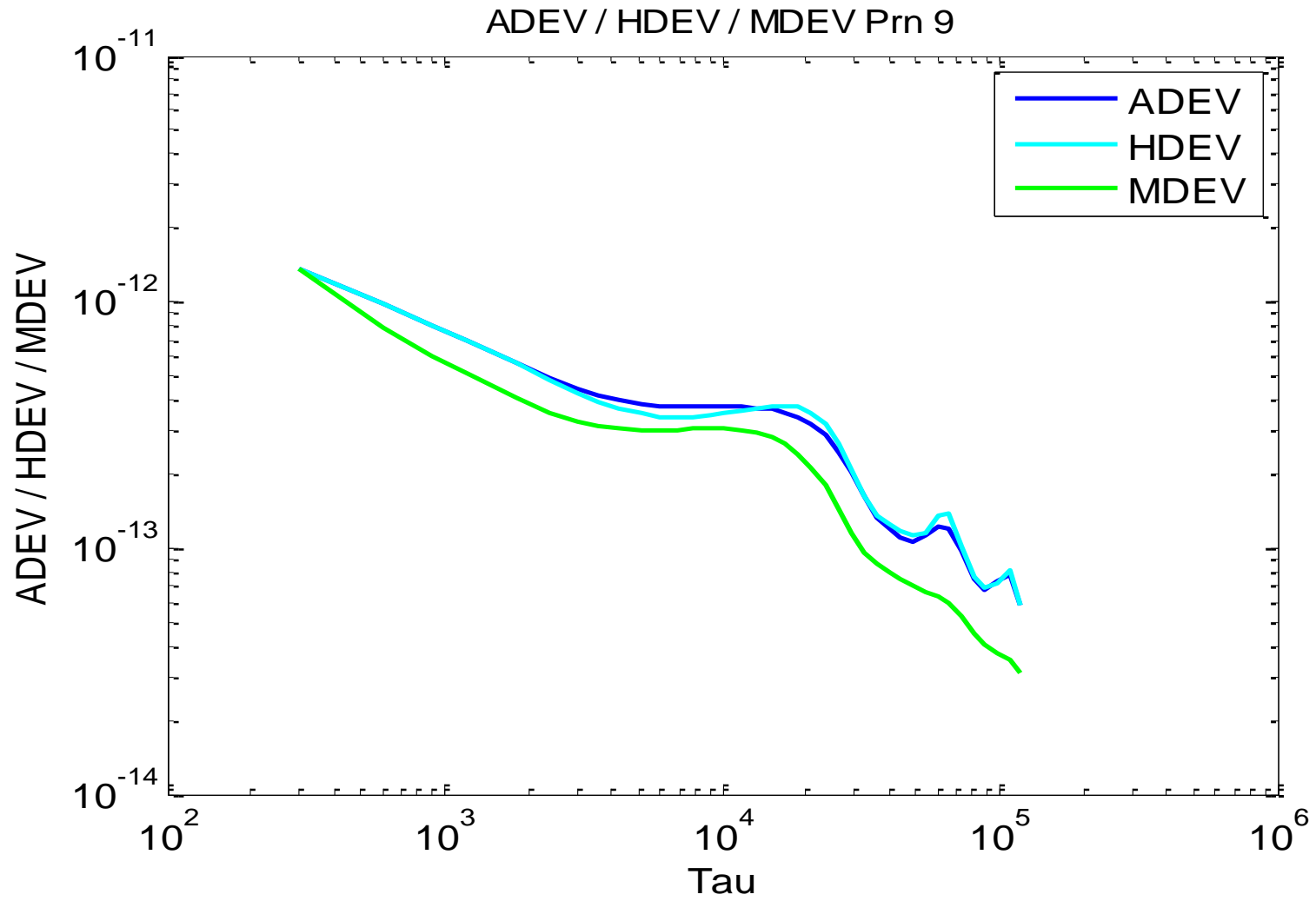


GPS Time – Individual Clock (Simulated Data)

Simulated data with similar characteristics

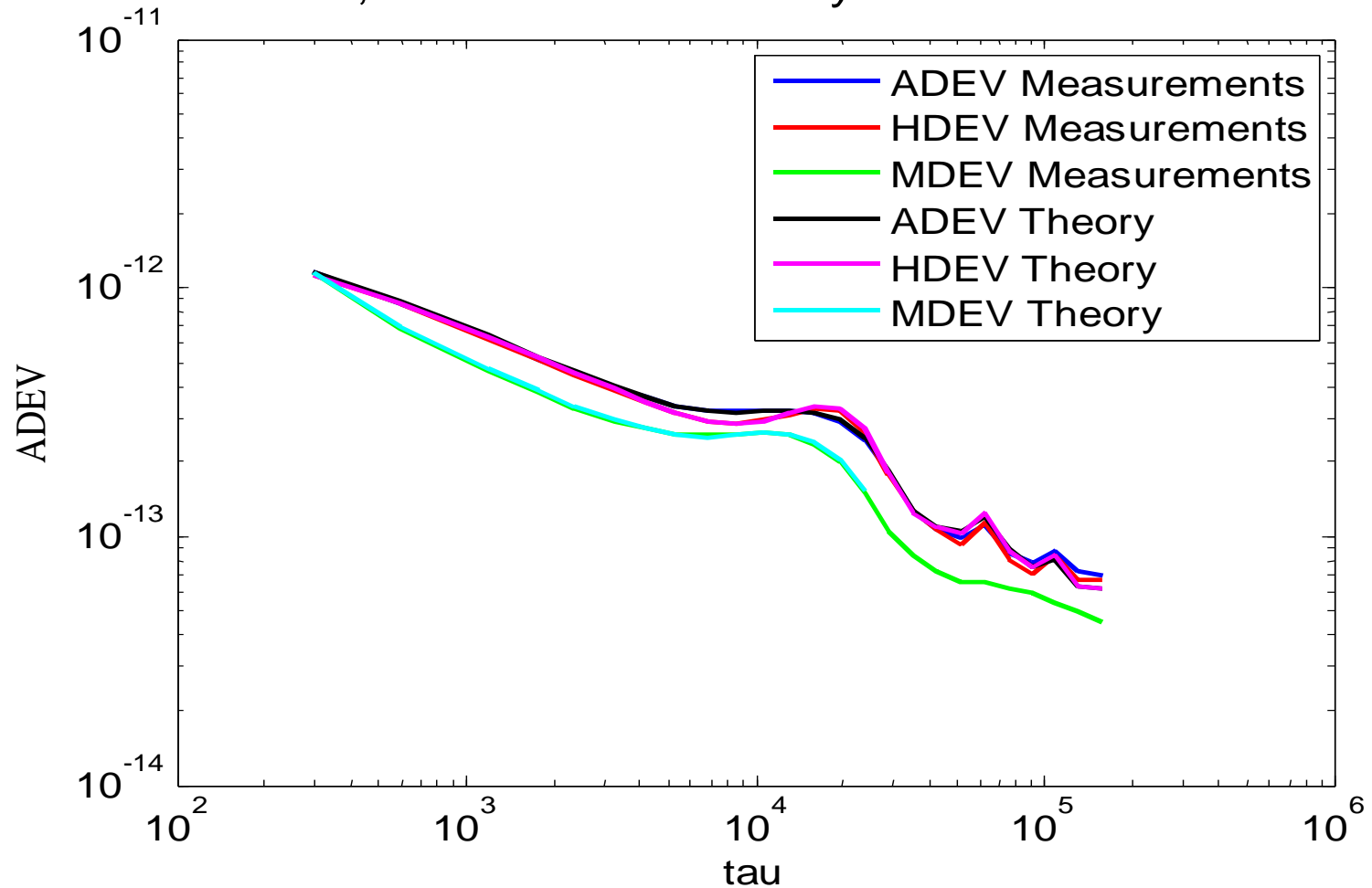


ADEV, HDEV, MDEV Real Data



ADEV, HDEV, MDEV Simulated Data

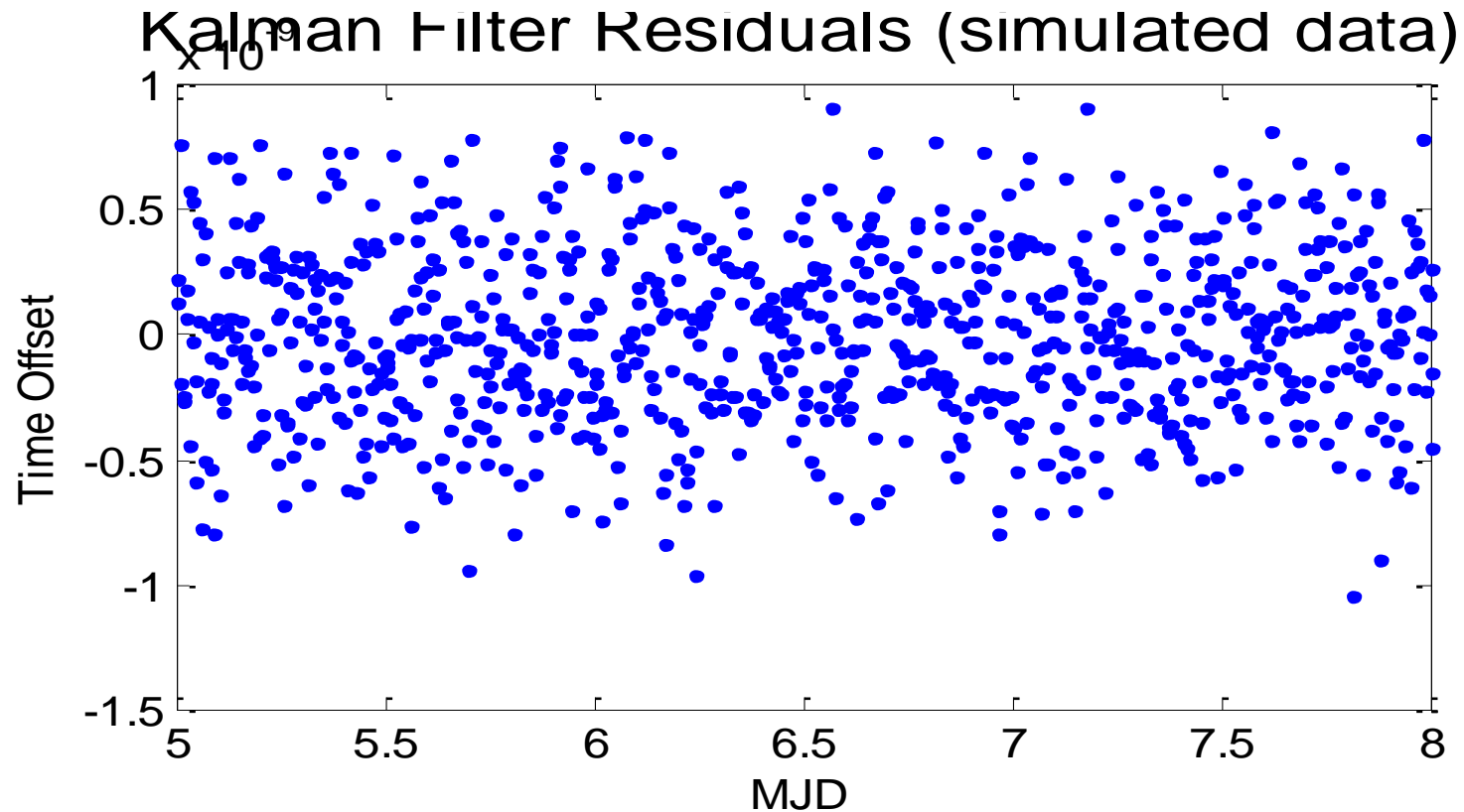
ADEV, HDEV MDEV Theory and Measurement



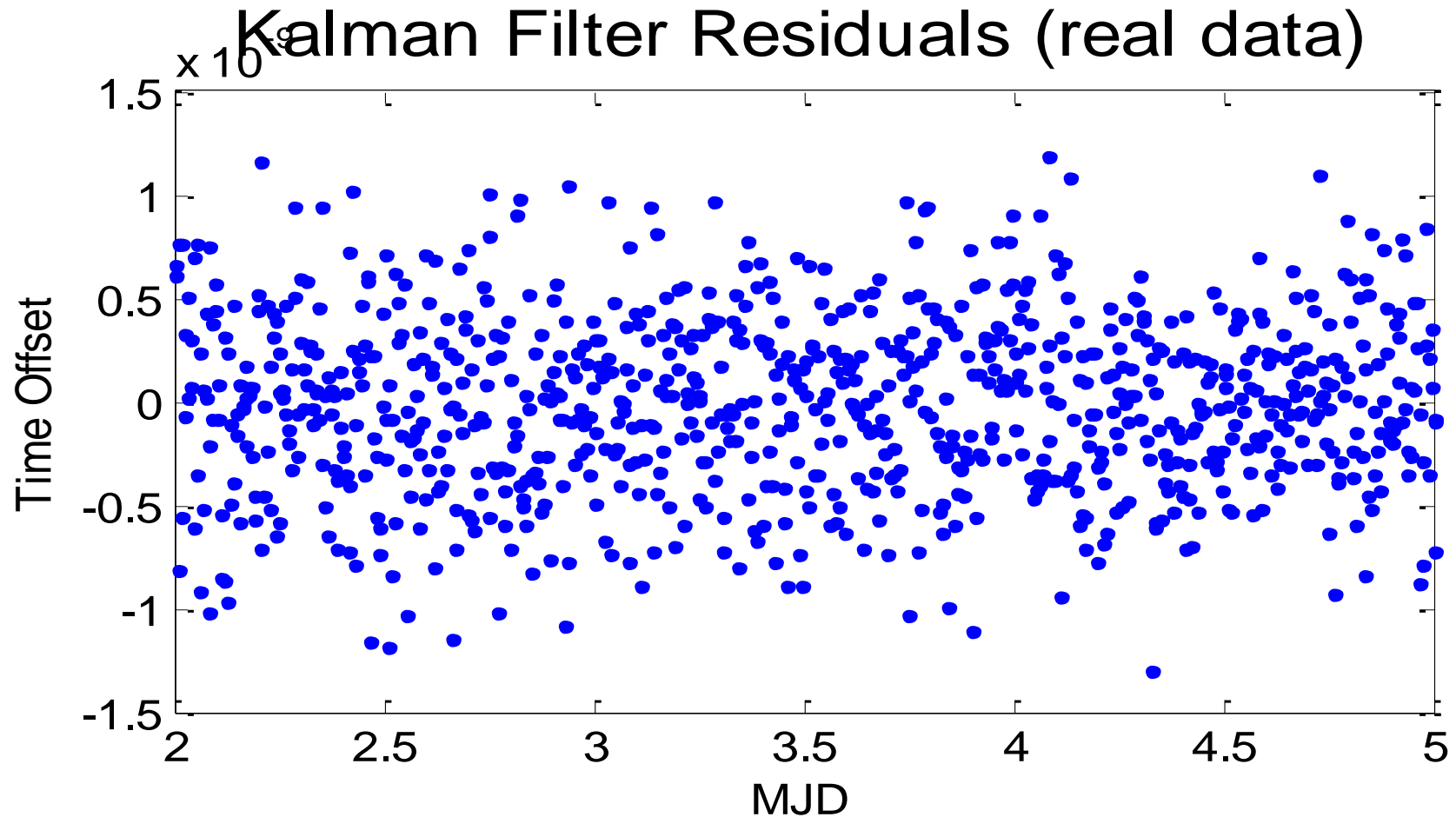
Kalman Filter Residuals

- Residuals $(y - H\hat{x}^-)$
- Residual Variance (Theory) $V = (HP^-H^T)$
- In an optimally constructed Kalman Filter the Residuals should be white!
- Residual Variance obtained from Parameter Covariance matrix should match that obtained from statistics applied to residuals

Kalman Filter Residuals (Simulated Data)



Kalman Filter Residuals (Real Data)



Computational Issues

- Too many state vector components result in a significant computational overhead
- Observability of state vector components
- Direct computation vs factorisation
- Direct use of Kalman filter equations usually numerically stable and easy to follow.
- Issue of RRFM scaling if a linear frequency drift state vector component is used

Clock Predictor Application

- May provide a close to optimal prediction in an environment that contains complex noise and deterministic characteristics
- If noise processes are simple e.g. WFM and no linear frequency drift then very simple predictors will work very well.

Clock Predictor Equations (Prediction)

$$\underline{\hat{x}}(t_n^-) = \underline{\underline{\Phi}}(\tau) \underline{\hat{x}}(t_{n-1}^+)$$

- Repeated application of above equation.
- Apply above equations with $\underline{\underline{\Phi}}(\tau)$ calculated at the required prediction length $\underline{\underline{\tau}}$

Clock Predictor Equations (PED)

$$\underline{\underline{P}}(t_n^-) = \underline{\underline{\Phi}}(\tau) P(t_{n-1}^+) \underline{\underline{\Phi}}^T(\tau) + \underline{\underline{Q}}(t_n)$$

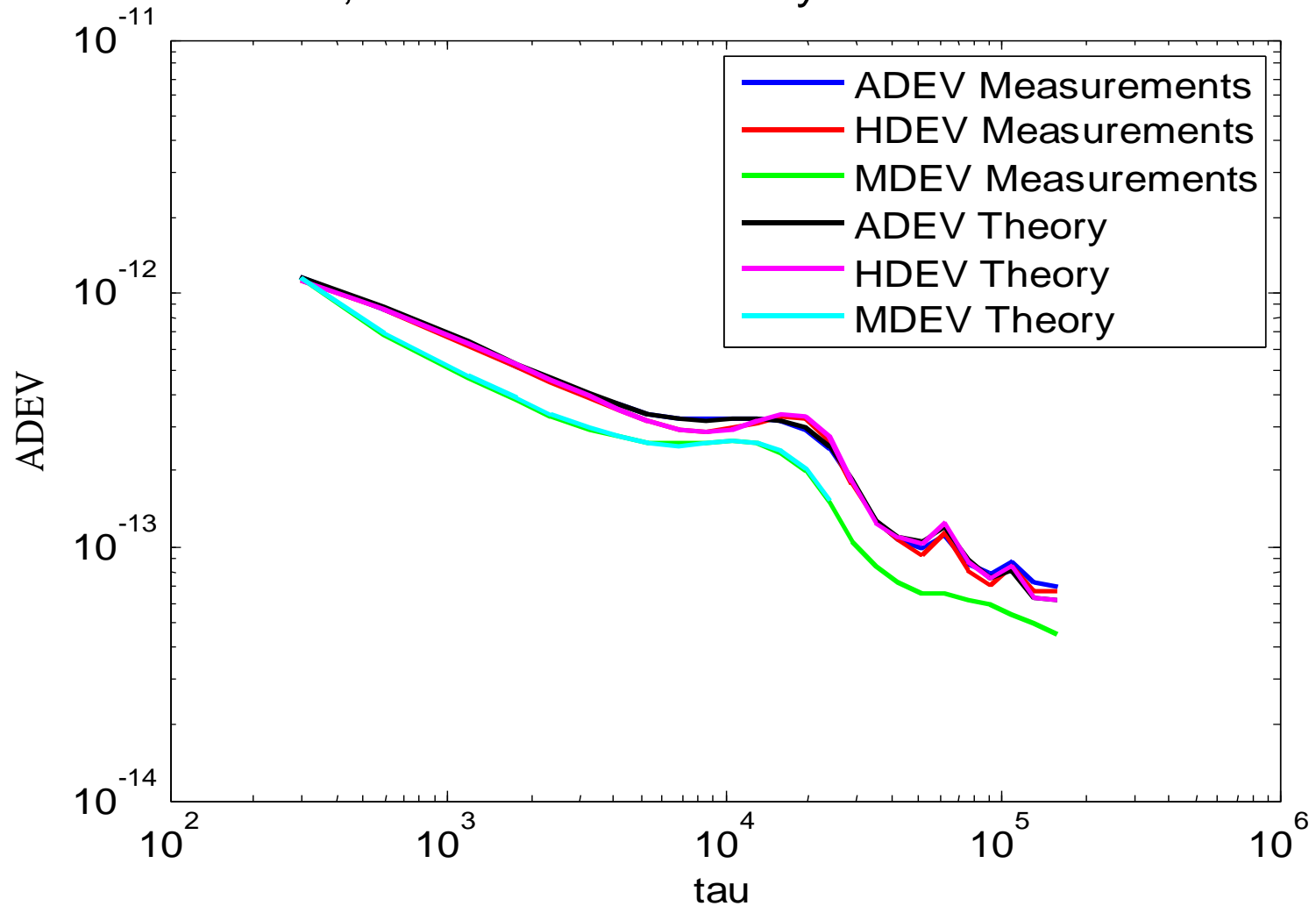
- Repeated application of above equation.
- Apply above equations with $\underline{\underline{\Phi}}(\tau)$ calculated at the required prediction length τ

Example: Predicting clock offset in presence of periodic instabilities

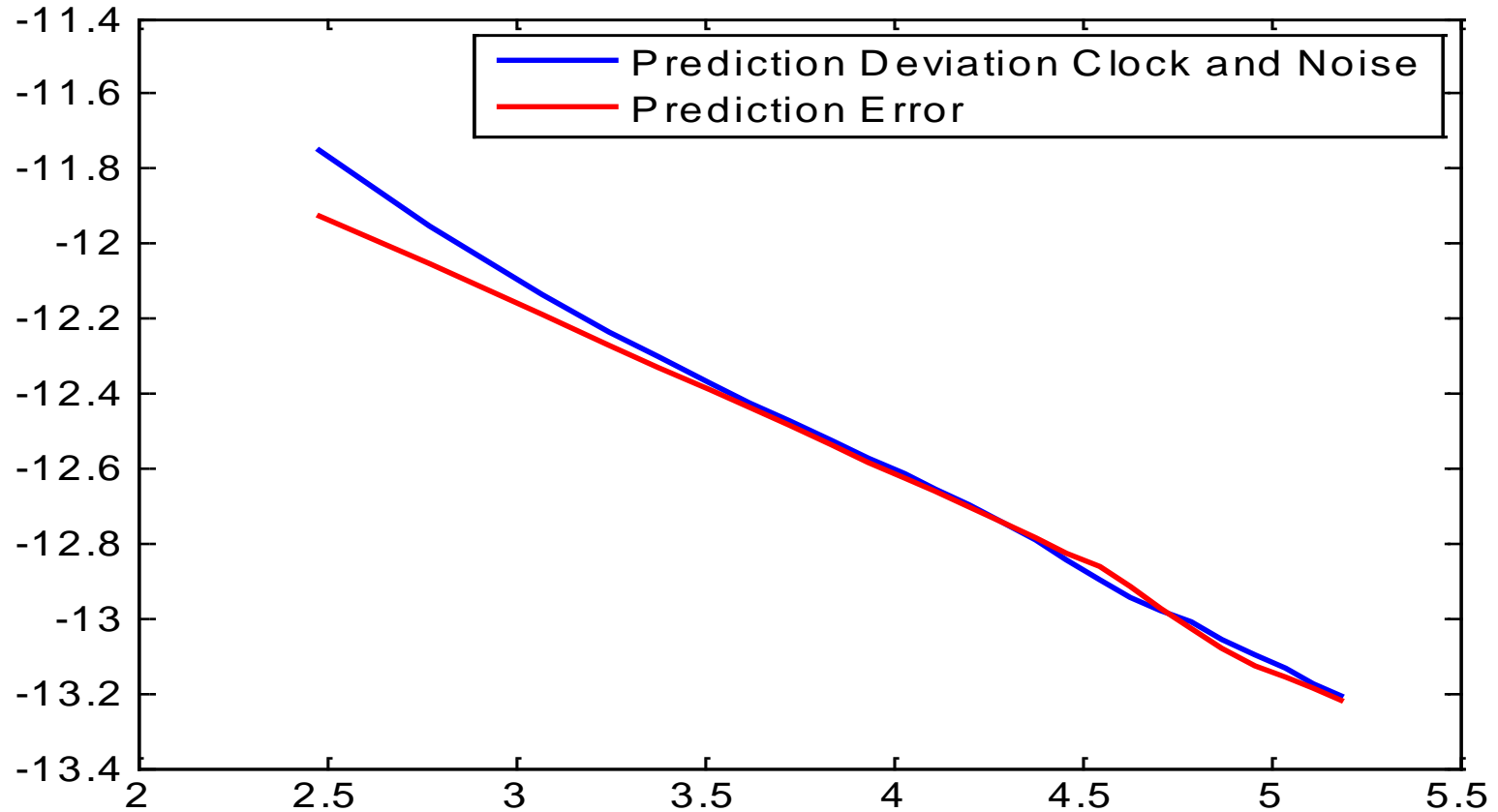
- $ADEV = PED/Tau$
- Above equation exact in case of WFM and RWFM noise processes, and a reasonable approximation in other situations

ADEV, HDEV, MDEV Simulated Data

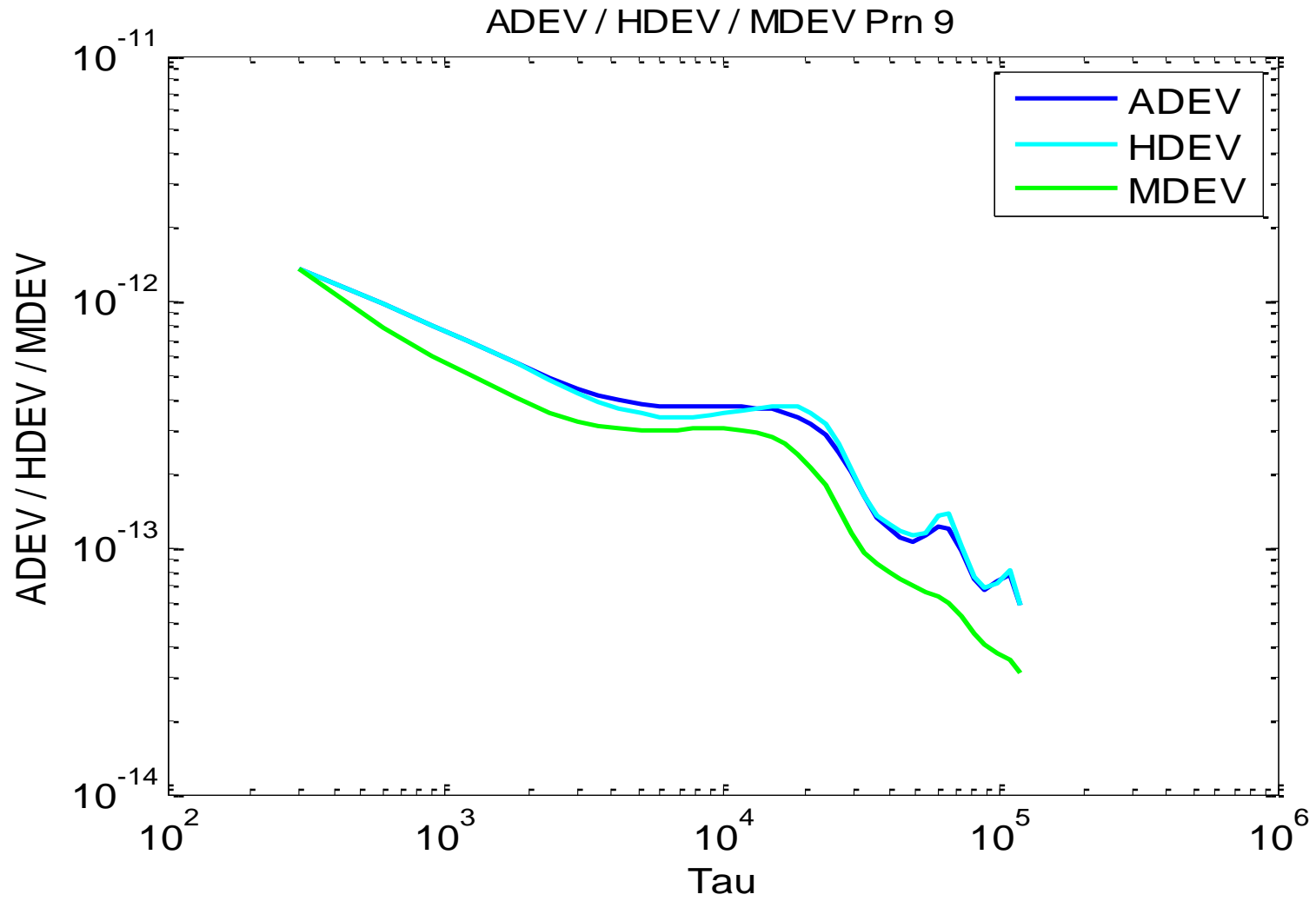
ADEV, HDEV MDEV Theory and Measurement



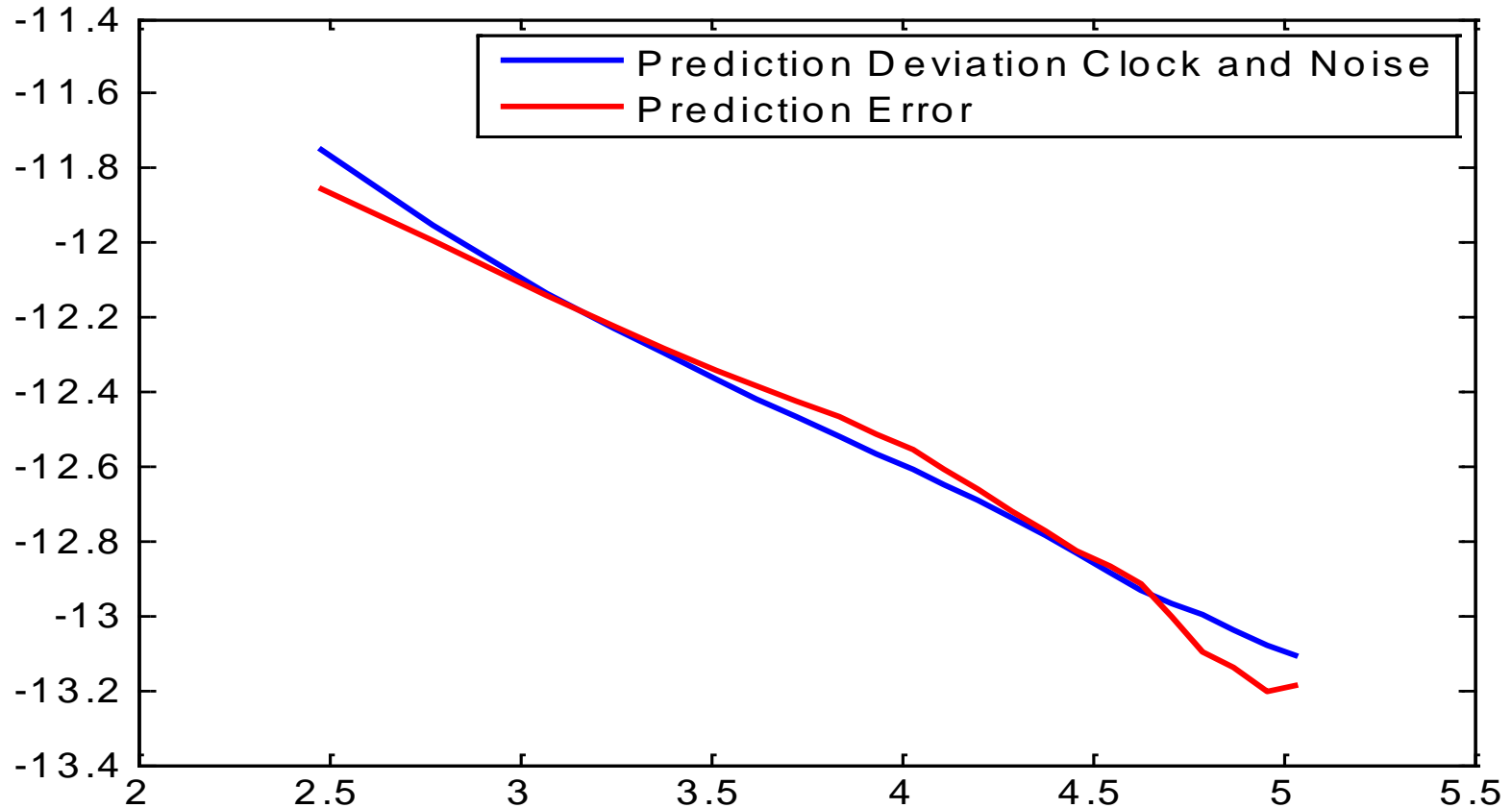
(PED /Tau) Prediction of Space Clocks (simulated data)



ADEV, HDEV, MDEV Real Data



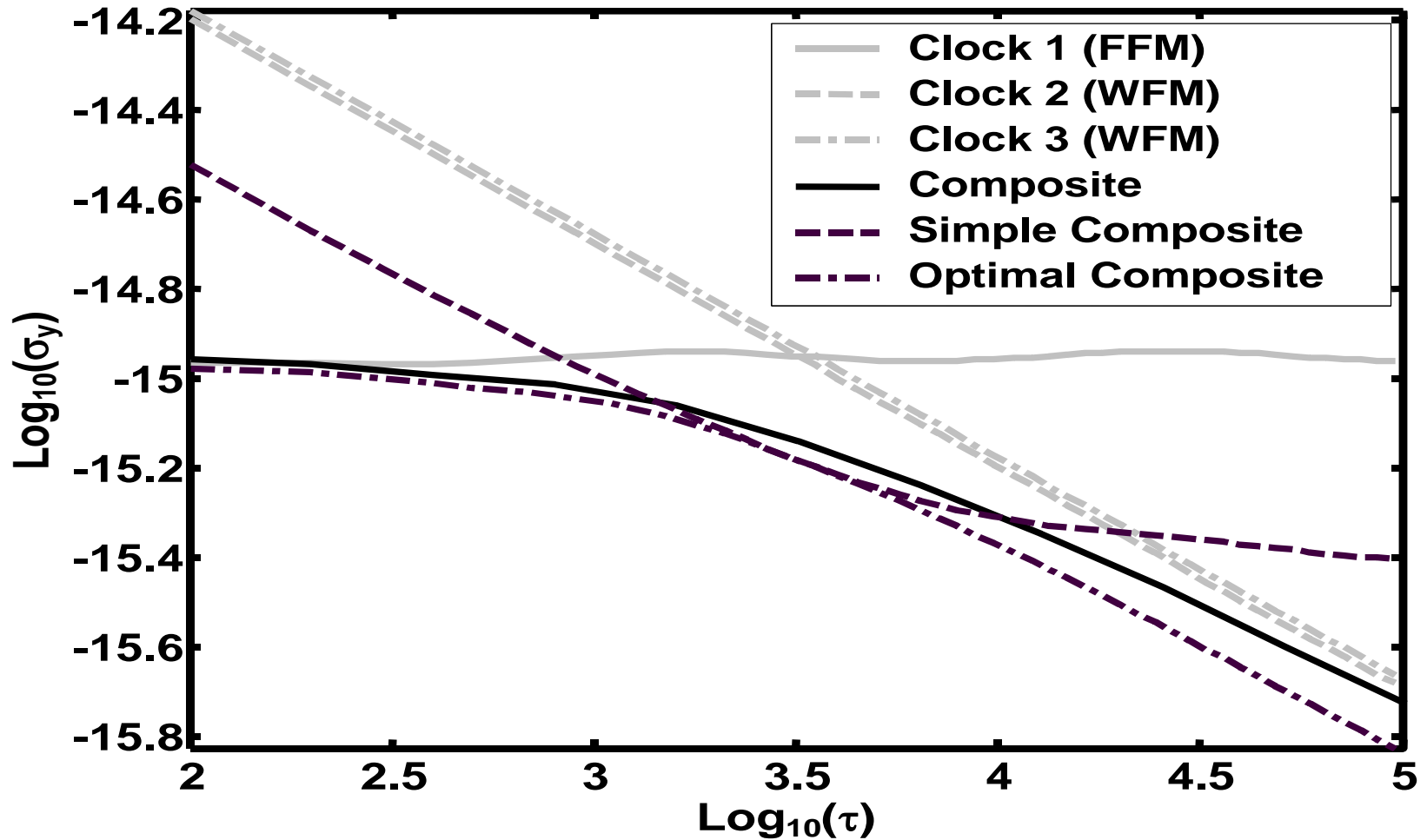
(PED /Tau) Prediction of Space Clocks (real data)



Kalman Filter Ensemble Algorithm

- Most famous time and frequency application of the Kalman filter
- Initially assume measurements are noiseless.
- Measurements $(C_j - C_1)$, difference between pairs of individual clocks
- Estimates $(T - C_i)$, where T is a “perfect” clock.
- Parameters being estimated cannot be constructed from the measurements:
- Badly designed Kalman filter.
- Problem solved using covariance x reduction

Stability of Composite Timescales



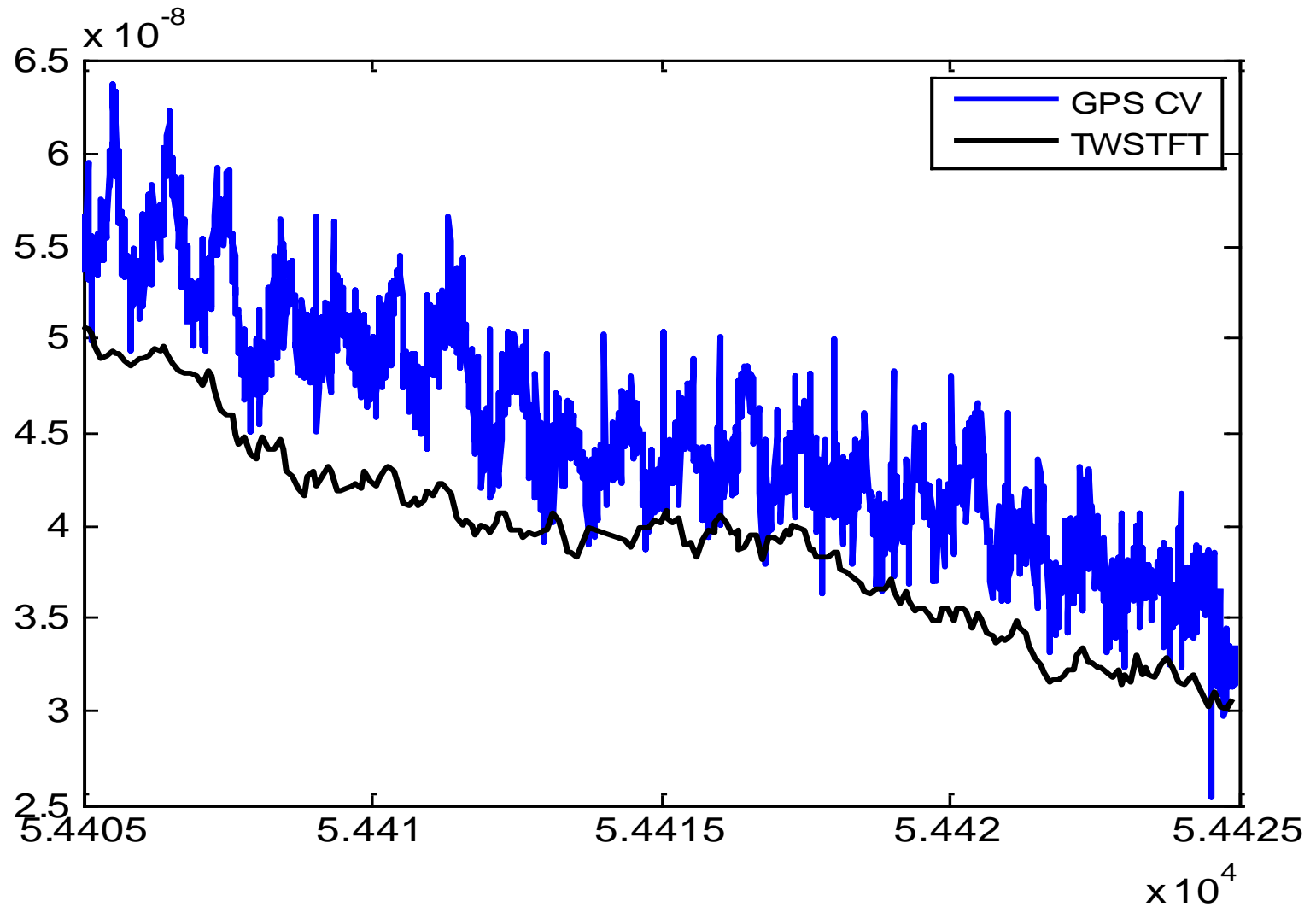
Advantage of Kalman Filter based Ensemble Algorithm

- Potential of close to optimal performance over a range of averaging times
- Classical Ensemble Algorithms provide a very close to optimal performance at a single averaging time

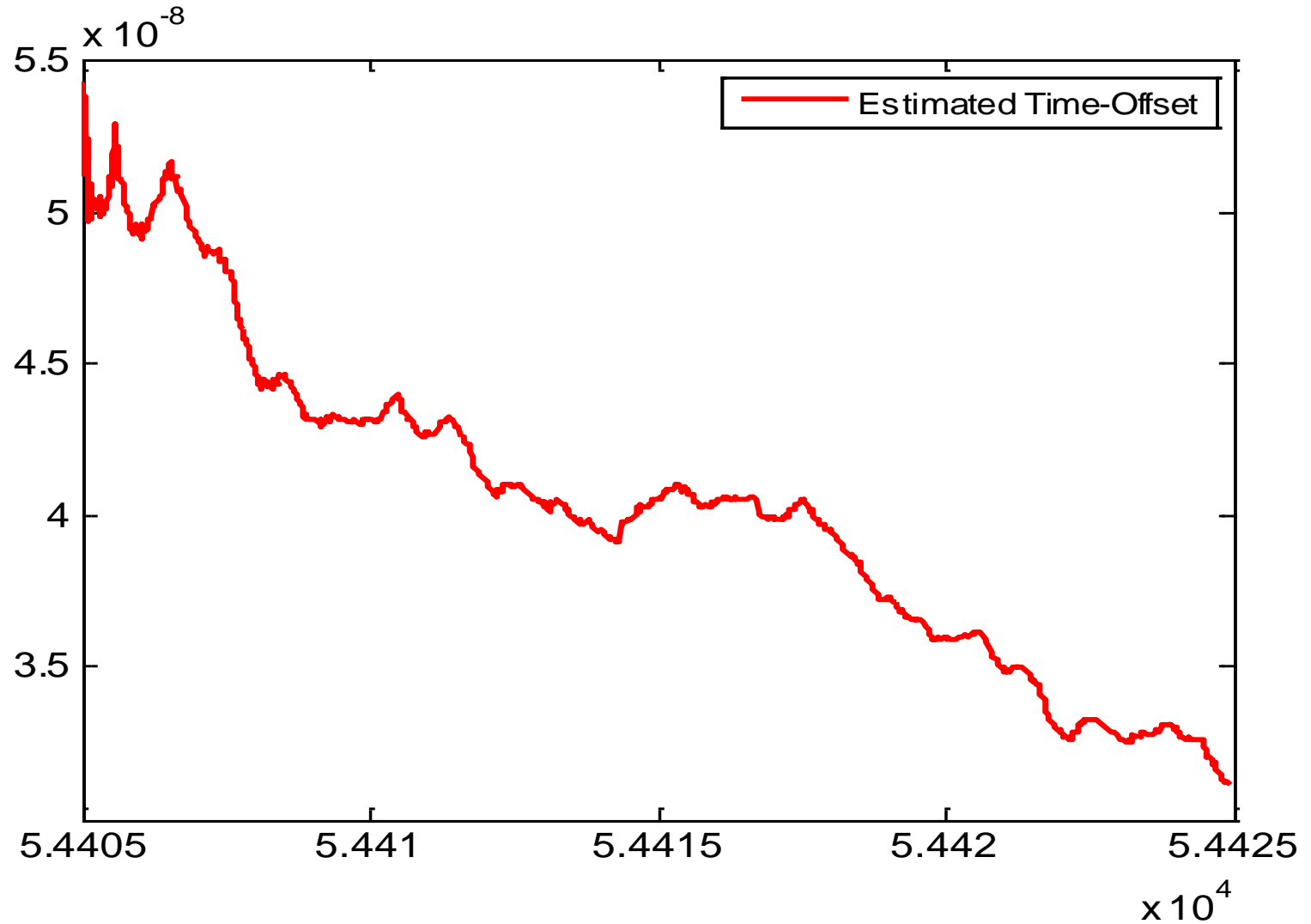
Combining TWSTFT and GPS Common-View measurements

- Time transfer noise modelled using Markov Noise processes
- Calibration biases modelled using long relaxation time Markov

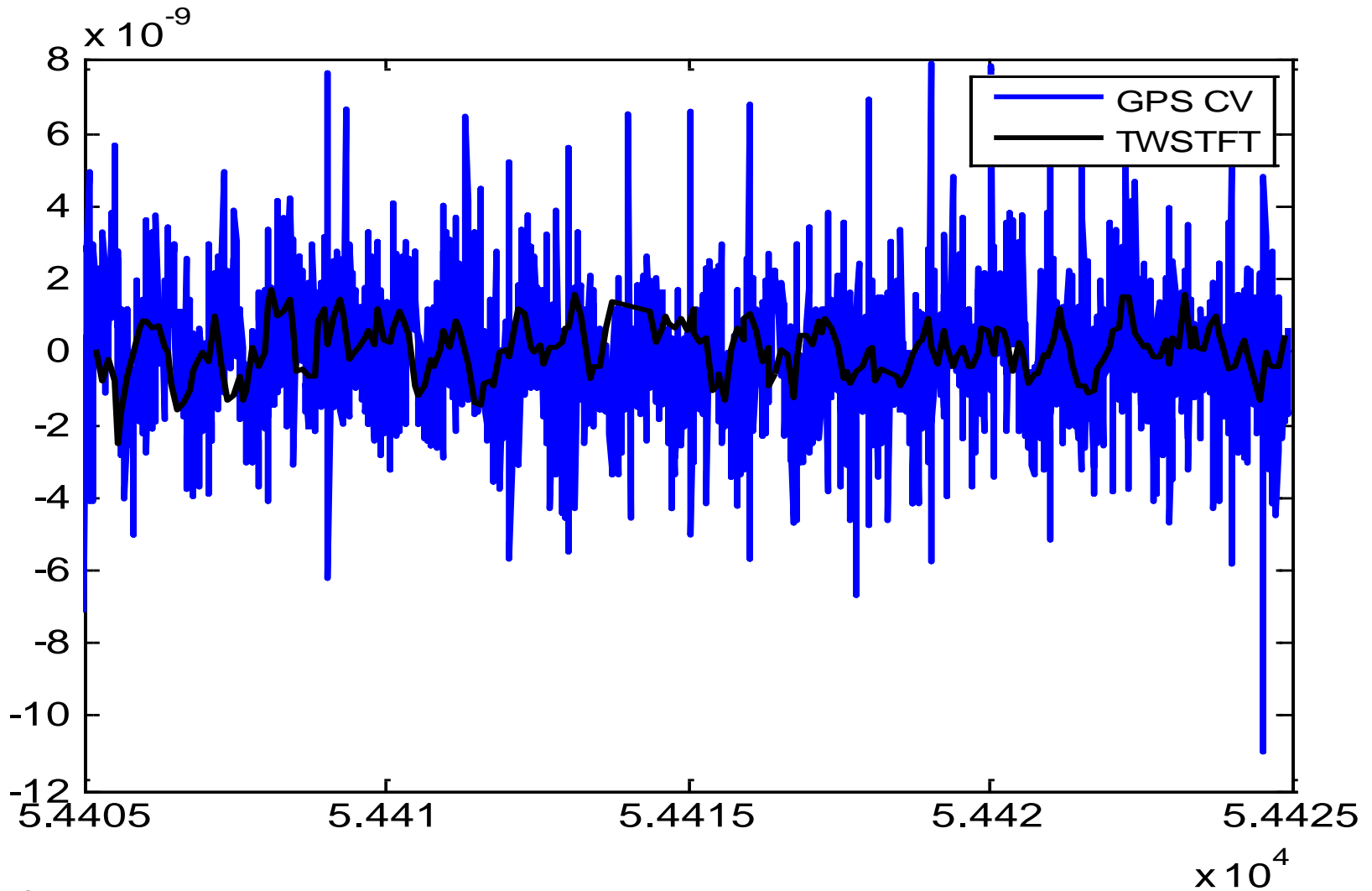
Real Time Transfer Data Example



Kalman Filter Estimates



Kalman Filter Residuals



Summary

- Operation of the Kalman filter
- Stochastic and deterministic models for time and frequency applications
- Real examples of use Kalman Filter

National Measurement System



The National Measurement System delivers world-class measurement science & technology through these organisations



The National Measurement System is the UK's national infrastructure of measurement Laboratories, which deliver world-class measurement science and technology through four National Measurement Institutes (NMIs): LGC, NPL the National Physical Laboratory, TUV NEL The former National Engineering Laboratory, and the National Measurement Office (NMO).