

Department of Mathematics

25<sup>th</sup> October 2023

Modeling and Estimation of a Continuous  
Flexible Structure using the Theory of Functional  
Connections

Dr. Riccardo Bevilacqua  
Aerospace Engineering Professor  
bevilacr@erau.edu  
www.riccardobevilacqua.com





**M. Sc. In Aerospace Engineering** *Sapienza University, Rome, Italy, 2002.*



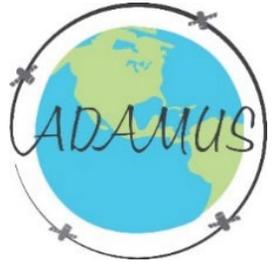
**Project Engineer in Mission Analysis** *Grupo Mecanica del Vuelo, Madrid, Spain, 2003.*



**Ph. D. in Applied Mathematics, Aerospace Engineering focus,** *Sapienza University, 2007.*



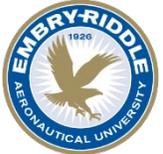
**Post-Doctoral appointment** *NPS, Monterey, CA (National Research Council), 2007-2010*



**Assistant Professor, RPI,** 2010-2014



**Associate Professor, UF,** 2014-2021



**Professor, ERAU,** 2021-current

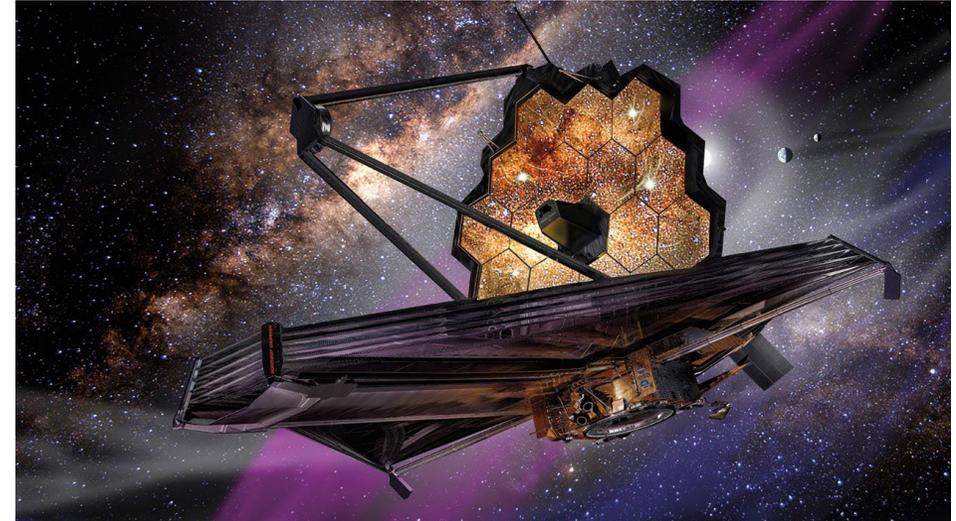




# Some Historical Notes



**Explorer 1 (1958)**



**JWST (2021)**



Oct 04 1957  
(Sputnik 1)

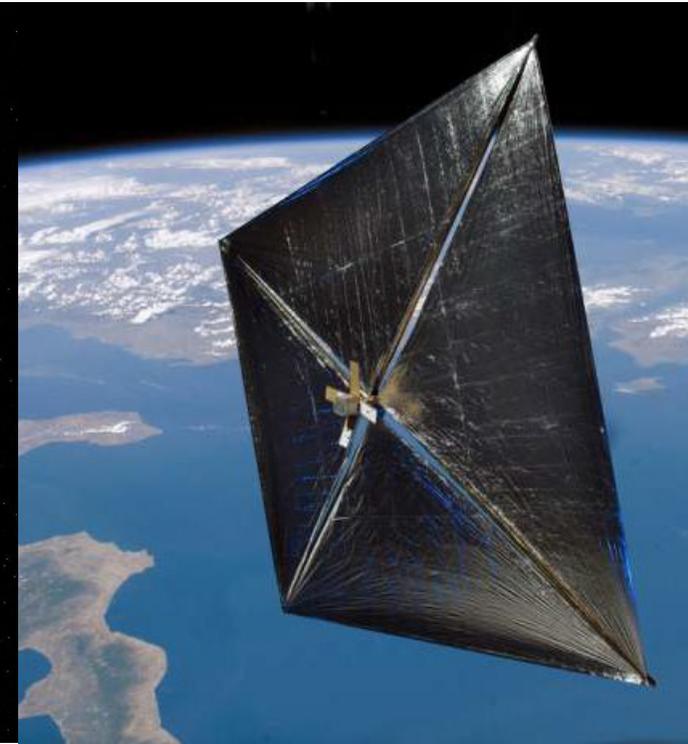
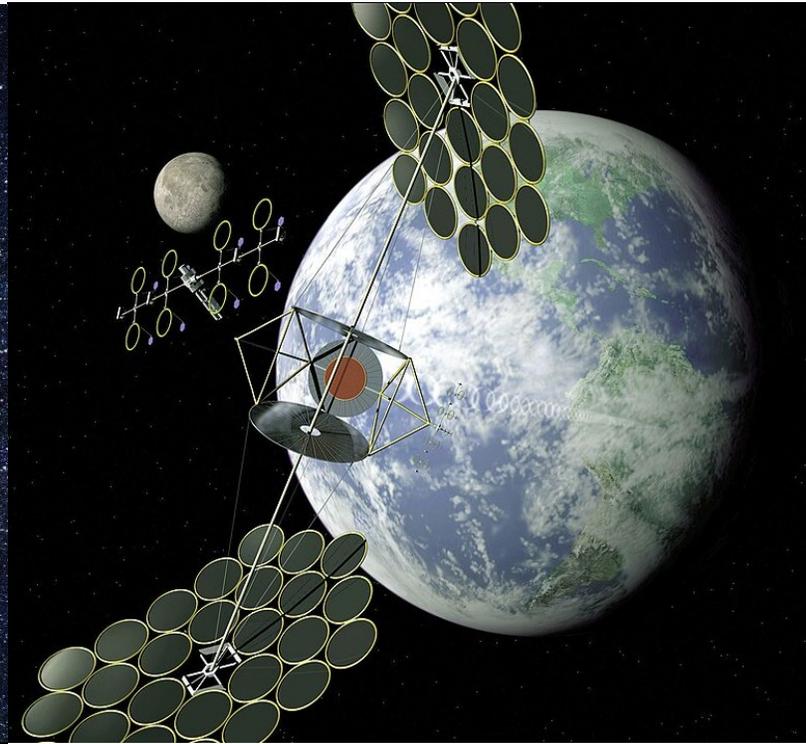
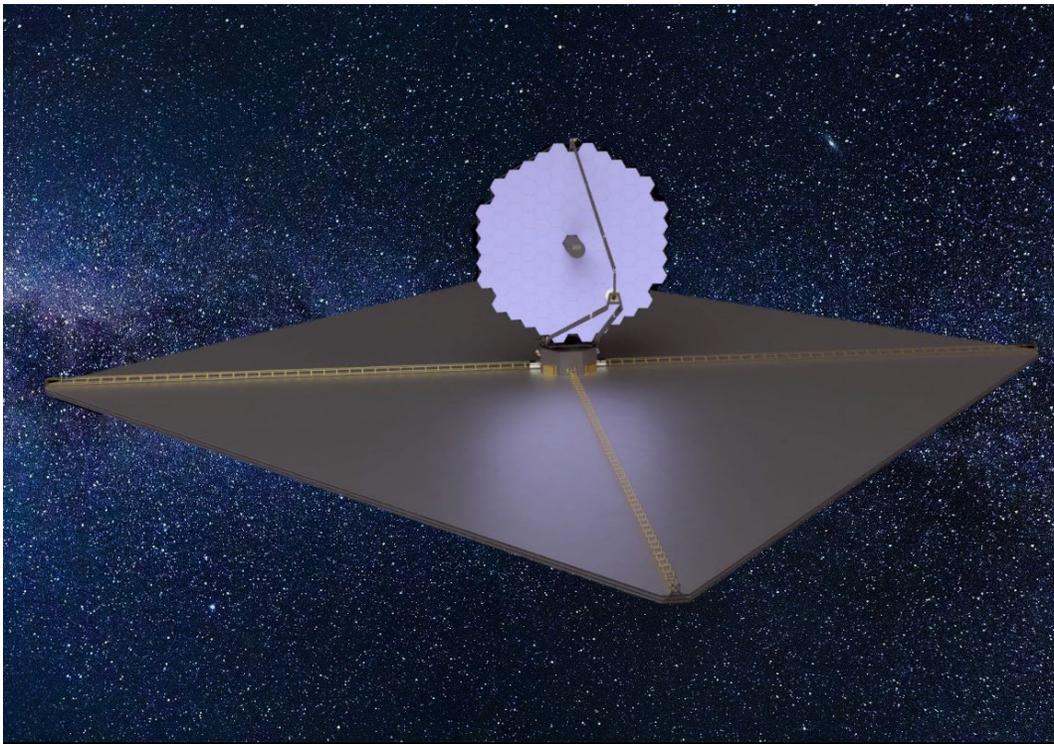
Feb 01 1958

Dec 25 2021

Today



# Flexible Spacecrafts Today



## Why?

- Lightweight structures
- Large deployable elements
- High accuracy requirements



# Flexible dynamics modeling

## MODELING METHODS

### INFINITE-DIMENSIONAL

- Euler-Bernoulli Beam
- Timoshenko Beam
- Kirchhoff Plate

E  
X  
A  
M  
P  
L  
E  
S

### FINITE-DIMENSIONAL

- Finite Element Method
- Modal Analysis (truncated)
- Assumed Modes
- Lumped Parameters



## Flexible dynamics modeling

# INFINITE-DIMENSIONAL METHODS

- Lead to sets of Partial Differential Equations
- Based on continuum mechanics
- Bring to models characterized by various levels of complexity

## PROs

- Account for the true continuous nature of the real-world object
- Closest as possible to the reality

## CONs

- Implementing into a control scheme may represent a challenge



# Flexible dynamics modeling

## FINITE-DIMENSIONAL METHODS

- Lead to sets of Ordinary Differential Equations
- Approximate the continuous structure using a finite number of variables
- Bring to models characterized by various levels of complexity

### PROs

- More convenient to integrate into classic controls
- Mathematically simpler
- Very effective if properly handled

### CONs

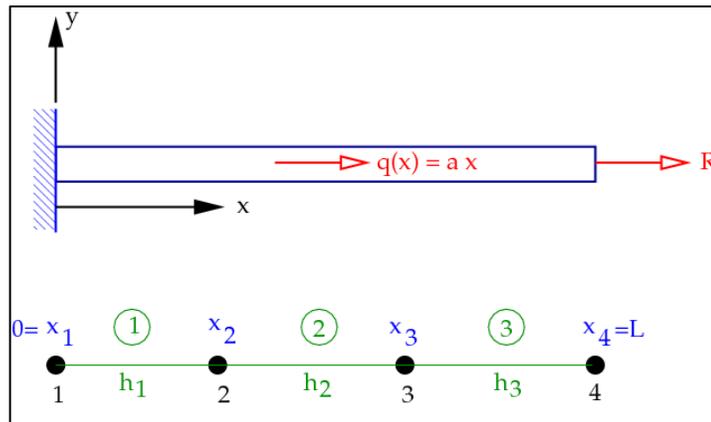
- Approximations of the reality
- May lead to computationally complex models



# Flexible dynamics modeling

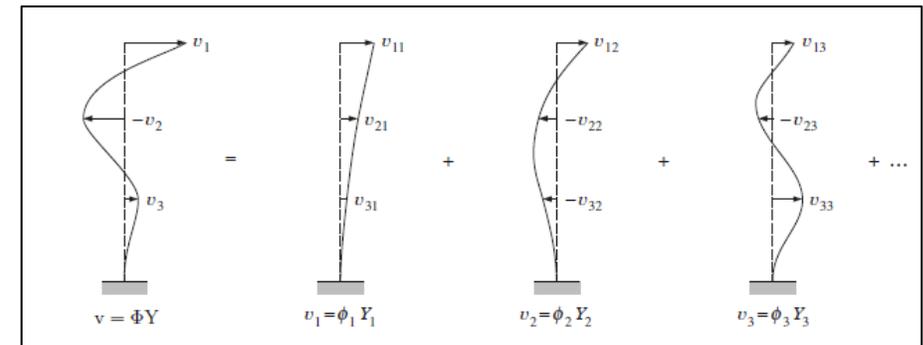
## FINITE-DIMENSIONAL METHODS

### Finite Elements Method



- Capable of effectively modeling structures of any type and complexity
- Models can be built quite intuitively using appropriate software
- Widely diffused (many software products available)
- Computational complexity to obtain accurate results may be high

### Assumed Modes Method



- Inspired by the analytic modal analysis
- High accuracy achievable with less computational complexity
- Choosing the synthetic modes may be challenging in some cases



# Flexible dynamics modeling

## THE PROPOSED METHOD

### Finite Elements Method

- Intuitively build the model
- High computational complexity

### Assumed Modes Method

- Lower computational complexity
- Choosing the synthetic modes can be nontrivial

Is a choice to be made?

Maybe not...



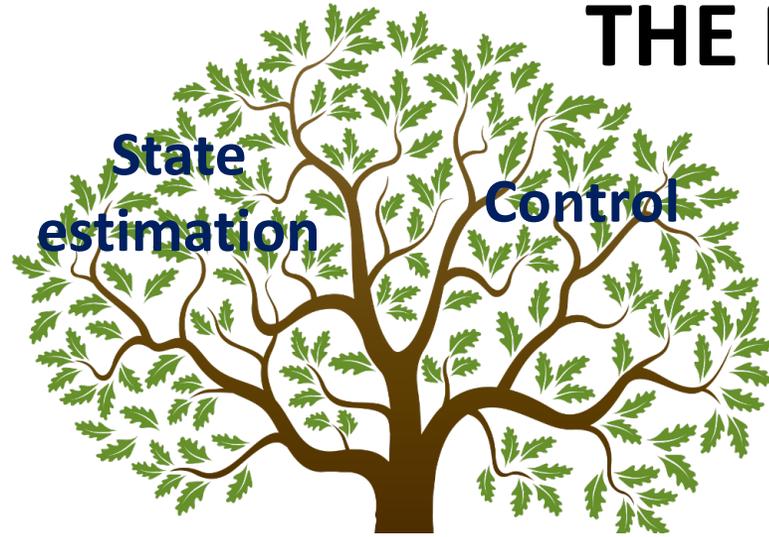
- Objective.** Develop dynamical models that are:
1. More compact and computationally efficient compared to FEM
  2. Easier to extend to any structure than assumed modes

## TFC-BASED DYNAMIC MODELING



# Flexible dynamics modeling

## THE PROPOSED METHOD



## Theory of Functional Connections



The resulting dynamic model can be used in several practical implementations, thanks to its compactness and simplicity.

The consistency with the physical (and mathematical) constraints is achieved exploiting the Theory of Functional Connections (TFC), which provides the main mathematical structure to the method.

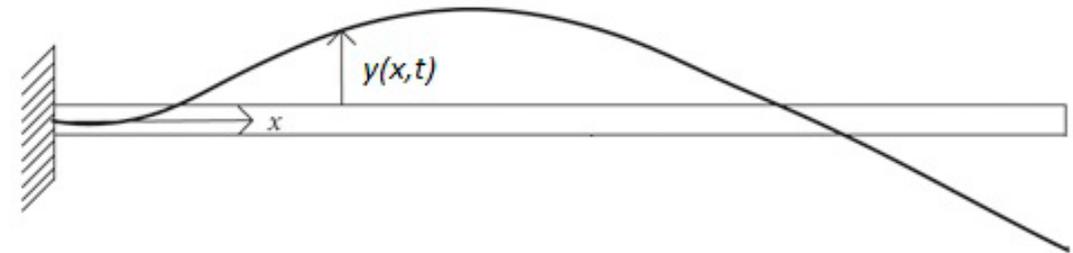
The proposed method takes an inspiration from the assumed modes idea of having the instantaneous shape of the structure approximated by the weighted sum of opportunely chosen functions.



# Flexible dynamics modeling

## THE ORIGIN: ASSUMED MODES METHOD

$$y(x, t) = \sum_{i=1}^{N_m} \eta_i(t) \Phi_i(x)$$



$\eta_i(t)$ : time dependent coefficients

$\Phi_i(x)$ : synthetic modes (equivalent to the modal shapes, but can be selected arbitrarily)

The requirements are:

1. the  $\Phi_i(x)$  satisfy the boundary conditions.
2. the  $\Phi_i(x)$  meet the continuity and differentiability conditions required by the physical model (e.g., for the Euler-Bernoulli beam  $\Phi_i(x)$  must be twice differentiable).

The result is that finding suitable  $\Phi_i$  is a nontrivial problem even in rather simple problems



## THE MATHEMATICAL TOOL: THEORY OF FUNCTIONAL CONNECTIONS

### Constrained expression

$$y(x, g(x)) = g(x) + \sum_{i=1}^{n_c} \rho_i(x, g(x)) \Phi_i(x, s(x))$$

Varying  $g(x)$

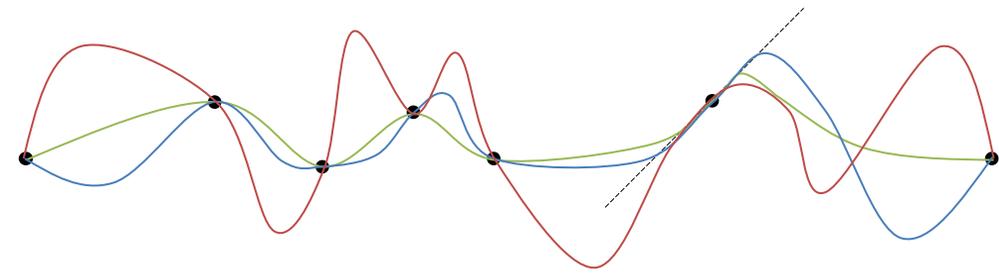


$$y(x, g(x)) = g(x) + \sum_{i=1}^{n_c} \eta_i(x, g(x)) s_i(x)$$

Free function

Coefficient functionals

Support functions



### Anatomy of a constraint

$$\mathfrak{C}_i(y(x_i, g(x_i))) = \kappa_i \quad (i = 1, \dots, n_c)$$

Constraint operator

Constraint constant term



# Theory of Functional Connections (TFC)

## THE FREE FUNCTION ( $g(x)$ )

$$g(x): D \mapsto g(x)$$

$$C \subseteq D$$

$$C = \{x_1, \dots, x_{N_c}\} \quad \text{constraint points}$$

- The free function must be defined at all the constraint points
- A representation for the entire space of the functions satisfying the requirement above must be found

The following solutions proved to give good results with the Theory of Functional Connections in a variety of problems:

### ORTHOGONAL POLYNOMIALS EXPANSIONS

$$g(x) = \sum_{j=1}^{n_b} \xi_j h_j(z(x)) = \bar{h}^T(z(x)) \bar{\xi}$$

$\bar{h}(z)$  is a vector of orthogonal polynomials

$\bar{\xi}$  is the vector of coefficients

$z(x)$  is a function mapping from  $D$  to the domain of the polynomials (depends on the type of polynomials)

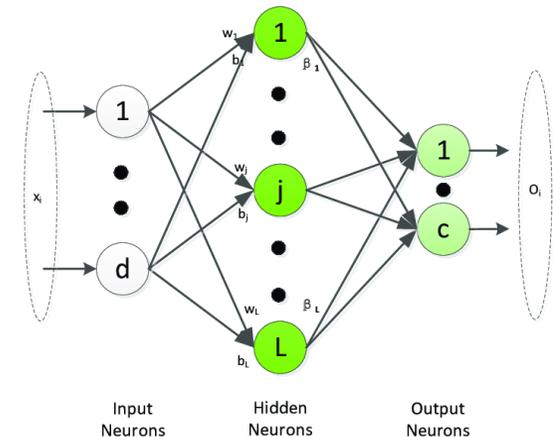
### NEURAL NETWORKS

Extreme Learning Machine (ELM)

+  
TFC

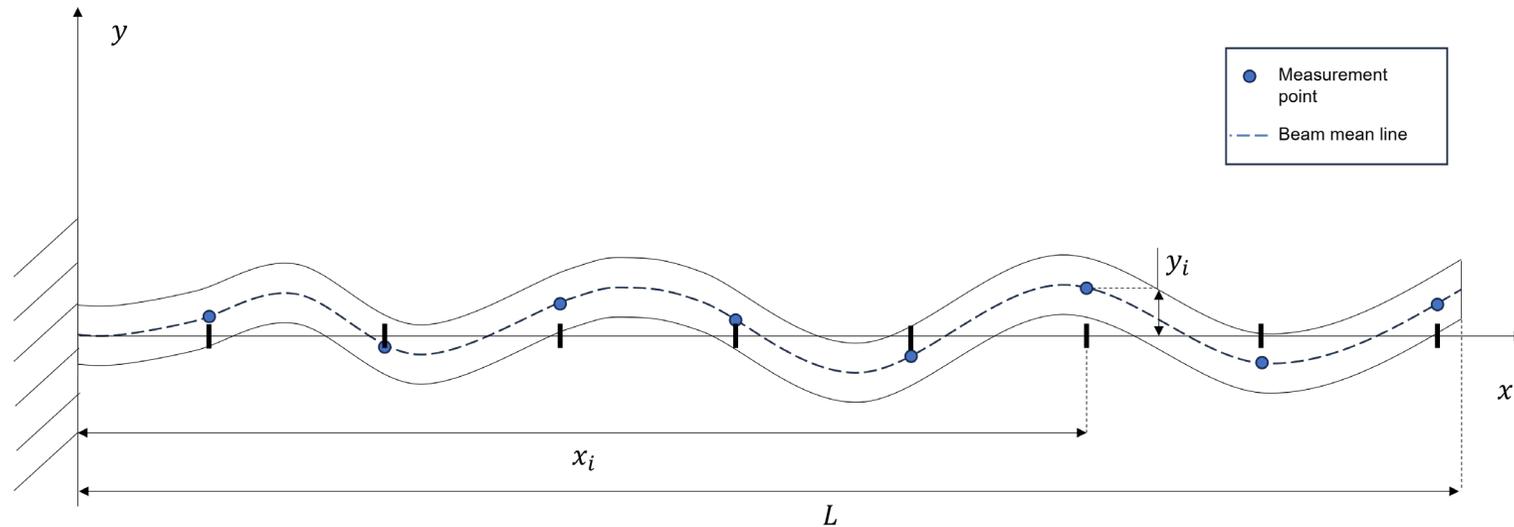


X-TFC





# The continuous model



## Euler-Bernoulli beam dynamic equation

$$\rho A \frac{\partial^2}{\partial t^2} y(x, g(x, t)) + \frac{\partial^2}{\partial x^2} \left( EI \frac{\partial^2}{\partial x^2} y(x, g(x, t)) \right) = q(x, t)$$

$\rho =$  mass density

$A =$  cross section area

$E =$  Young's modulus

$I =$  cross section area moment of inertia

$q(x, t) =$  external input

## Boundary conditions (cantilever beam)

$$\begin{cases} y(0, g(0)) = 0 \\ \frac{d}{dx} y(0, g(0)) = 0 \\ \frac{d^2}{dx^2} y(L, g(L)) = 0 \\ \frac{d^3}{dx^3} y(L, g(L)) = 0 \end{cases} \quad x \in [0, L]$$

No bending moment and no shear forces at L



# The TFC approximation

## THE CONSTRAINED EXPRESSION

$$y(x, g(x)) = g(x) + \sum_{i=1}^{n_c} \eta_i(x, g(x)) s_i(x)$$

Imposing the boundary conditions



$$\begin{bmatrix} \kappa_1 - \mathfrak{C}_1(g(x_1)) \\ \vdots \\ \kappa_{n_c} - \mathfrak{C}_{n_c}(g(x_{n_c})) \end{bmatrix} = \begin{bmatrix} \mathfrak{C}_1(s_1(x_1)) & \cdots & \mathfrak{C}_1(s_{n_c}(x_1)) \\ \vdots & \ddots & \vdots \\ \mathfrak{C}_{n_c}(s_1(x_{n_c})) & \cdots & \mathfrak{C}_{n_c}(s_{n_c}(x_{n_c})) \end{bmatrix} \begin{bmatrix} \eta_1 \\ \vdots \\ \eta_{n_c} \end{bmatrix}$$

In this case, we have 4 constraints ( $n_c = 4$ ), corresponding to the 4 boundary conditions for the cantilever beam



$$\left\{ \begin{array}{l} \kappa_1 = \kappa_2 = \kappa_3 = \kappa_4 = 0 \\ \mathfrak{C}_1 = \frac{d^0}{dx^0} \quad \mathfrak{C}_2 = \frac{d}{dx} \quad \mathfrak{C}_3 = \frac{d^2}{dx^2} \quad \mathfrak{C}_4 = \frac{d^3}{dx^3} \\ x_1 = 0 \quad x_2 = 0 \quad x_3 = L \quad x_4 = L \end{array} \right.$$



The last step is to choose a set of support functions  $s_i(x)$  ( $i = 1, \dots, 4$ ) such that the resulting matrix is nonsingular. A good choice can be the following:

$$s_1(x) = 1 \quad s_2(x) = x \quad s_3(x) = x^2 \quad s_4(x) = x^3$$



The Constrained Expression

$$y(x, g(x)) = g(x) - g(0) - g_x(0)x + \left( \frac{L}{2} g_{xxx}(L) - \frac{1}{2} g_{xx}(L) \right) x^2 - \frac{1}{6} g_{xxx}(L) x^3$$



# The TFC approximation

## THE FREE FUNCTION REPRESENTATION

- The free function must be defined at the constraint points
- A representation must be found capable of representing the entire function space
- Several representations that can be used for the free function exist

### Chebyshev polynomials (first type)

$$g(x) = \sum_{j=1}^{n_b} \xi_j h_j(z(x))$$

$z: [0, L] \mapsto [-1, 1]$

$h_j$  :  $j^{th}$  Chebyshev polynomial  
 $\xi_j$  : coefficient. In the following developments it will be considered as some function of time

- Using orthogonal polynomials in general to represent the free function is a convenient choice to obtain the dynamic model representation we are looking for. This because of the simple linear expression depending on the  $\xi_i$  coefficients.
- In particular, the Chebyshev polynomials expansion has the advantage of a  $z(x)$  function mapping from the physical domain to the polynomials domain that is linear



# The TFC approximation

## THE CONSTRAINED EXPRESSION IN TERMS OF CHEBYSHEV POLYNOMIALS

The Free Function expansion

$$g(x) = \sum_{j=1}^{n_b} \xi_j h_j(z(x)) = \bar{h}(x)^T \bar{\xi}$$

The Constrained Expression

$$y(x, g(x)) = g(x) - g(0) - g_x(0)x + \left( \frac{L}{2} g_{xxx}(L) - \frac{1}{2} g_{xx}(L) \right) x^2 - \frac{1}{6} g_{xxx}(L) x^3$$



Combining the two expressions



$$\bar{h}^T(z(x)) \bar{\xi}(t) - \bar{h}^T(z(0)) \bar{\xi}(t) - \bar{h}_x^T(z(0)) \bar{\xi}(t) x + \left( \frac{L}{2} \bar{h}_{xxx}^T(z(L)) \bar{\xi}(t) - \frac{1}{2} \bar{h}_{xx}^T(z(L)) \bar{\xi}(t) \right) x^2 - \frac{1}{6} \bar{h}_{xxx}^T(z(L)) \bar{\xi}(t) x^3$$

$$y(x, t)$$



# The TFC approximation

## THE CONSTRAINED EXPRESSION IN TERMS OF CHEBYSHEV POLYNOMIALS

$$\bar{h}^T(z(x))\bar{\xi}(t) - \bar{h}^T(z(0))\bar{\xi}(t) - \bar{h}_x^T(z(0))\bar{\xi}(t)x + \left( \frac{L}{2} \bar{h}_{xxx}^T(z(L))\bar{\xi}(t) - \frac{1}{2} \bar{h}_{xx}^T(z(L))\bar{\xi}(t) \right) x^2 - \frac{1}{6} \bar{h}_{xxx}^T(z(L))\bar{\xi}(t)x^3$$

- **Compact operative formulation**
  - All the Chebyshev polynomials are included in the  $\bar{h}$  vector
  - The displacement at any time instant and location along the beam axis can be evaluated substituting the corresponding  $(x, t)$  pair
- **Automatically provides a separation between the space  $(\bar{h}(x))$  and time  $(\bar{\xi}(t))$  dependence**
- **This vectorial formulation will be exploited to obtain a compact final representation with little effort**



# TFC dynamic model

At this point, we need a way to link the TFC mathematical representation to the dynamic nature of the real-world object. A strategy to reach the goal is to use the Lagrangian mechanics.

## LAGRANGIAN MECHANICS (A summary)

$$\mathcal{L} = T - U$$

**LAGRANGIAN EXPRESSION**

$$\frac{d}{dt} \left( \frac{\partial \mathcal{L}}{\partial \dot{\bar{q}}} \right) - \frac{\partial \mathcal{L}}{\partial \bar{q}} = \bar{Q}$$

**EULER-LAGRANGE EQUATION**

$\bar{q}(t)$ : generalized coordinates vector

$\mathcal{L}(\bar{q}, \dot{\bar{q}})$ : Lagrangian

$T(\bar{q}, \dot{\bar{q}})$ : kinetic energy

$U(\bar{q}, \dot{\bar{q}})$ : potential energy

$\bar{Q}$  is the vector of the generalized non-conservative, non-constraint forces:

$$\bar{Q} = \left( \frac{\partial \bar{r}}{\partial \bar{q}} \right)^T \bar{p}$$

Deriving the TFC constrained expression in term of Chebyshev polynomials, we found that:

- The free function can vary with time:

$$g(x, t) = g(x, \bar{\xi}(t))$$

- Consequently, the displacement of a point:

$$y_i(t) = y(x_i, t) = y(x_i, g(x_i, t))$$



We can use the  $\xi_i(t)$  coefficients of the polynomial expansion as the generalized coordinates of the system



# TFC dynamic model

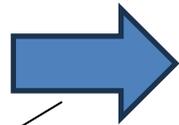
Now, we can build the Lagrangian expression in terms of the chosen set of generalized coordinates ( $\bar{\xi}$ )

**LAGRANGIAN:**  $\mathcal{L} = T - U$

For a Euler-Bernoulli beam, we have:

$$T = \frac{1}{2} \int_0^L \rho(x) A(x) \left( \frac{\partial y}{\partial t} \right)^2 dx$$

$$U = \frac{1}{2} \int_0^L E(x) I(x) \left( \frac{\partial^2 y}{\partial x^2} \right)^2 dx$$



$$T = \frac{1}{2} \dot{\bar{\xi}}^T(t) M \dot{\bar{\xi}}(t)$$

$$U = \frac{1}{2} \bar{\xi}^T(t) K \bar{\xi}(t)$$

Assuming a system of forces acting on  $n_m$  points along the beam, the generalized forces in matrix form will be:

Exploiting:

1. The matrix representation of the constrained expression
2. The time and space dependency separation provided by the polynomial expansion of  $g(x)$

$M$  and  $K$  are symmetric matrices containing the information about the mechanical and geometric parameters of the structure ( $L, A, \rho, E, I$ )

$$\bar{Q} = H^T \bar{p}(t)$$

$$H = \frac{\partial \bar{y}}{\partial \bar{q}}$$



# TFC dynamic model

Finally, we can apply the Euler-Lagrange equation to obtain the dynamic equation of the structure based on the TFC

**LAGRANGIAN:**

$$\mathcal{L} = T - U = \frac{1}{2} \left( \dot{\bar{\xi}}^T(t) M \dot{\bar{\xi}}(t) - \bar{\xi}^T(t) K \bar{\xi}(t) \right)$$



$$\frac{d}{dt} \left( \frac{\partial \mathcal{L}}{\partial \dot{\bar{\xi}}} \right) - \frac{\partial \mathcal{L}}{\partial \bar{\xi}} = \bar{Q}$$



$$M \ddot{\bar{\xi}} + K \bar{\xi} = H^T \bar{p}(t)$$

For the derivatives, we take advantage of the symmetry of  $M$  and  $K$ :

$$\frac{\partial T}{\partial \dot{\bar{\xi}}} = \frac{1}{2} (M + M^T) \dot{\bar{\xi}} = M \dot{\bar{\xi}}$$

$$\frac{\partial U}{\partial \bar{\xi}} = \frac{1}{2} (K + K^T) \bar{\xi} = K \bar{\xi}$$



## TFC dynamic model

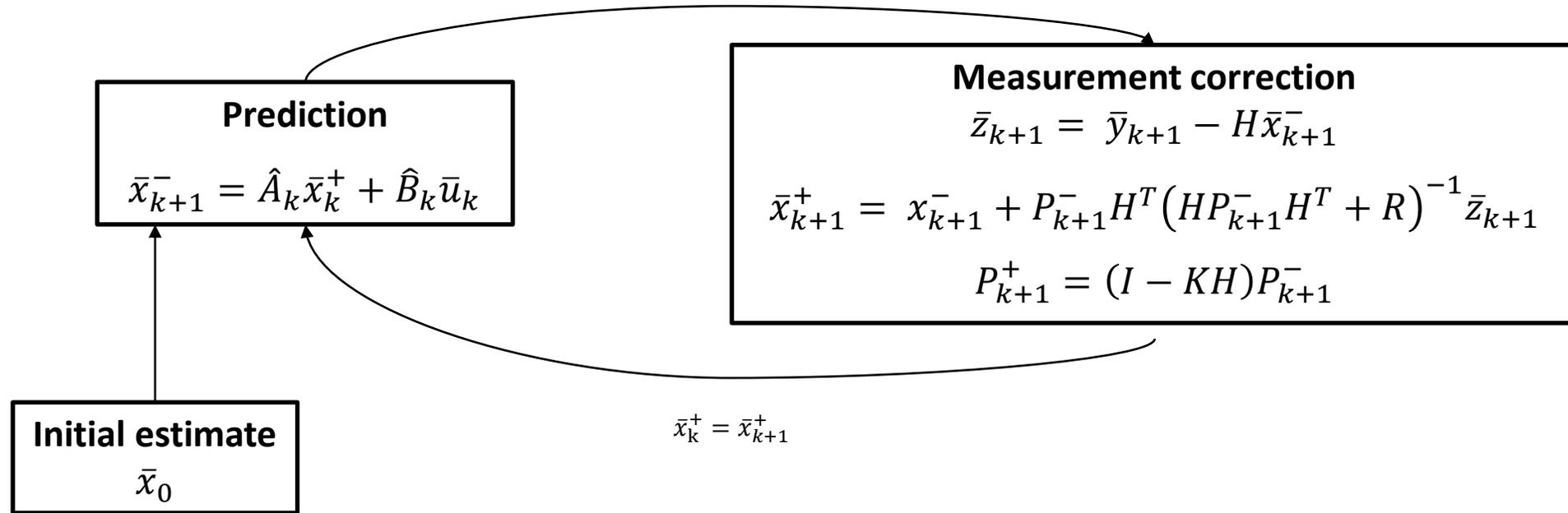
$$M\ddot{\bar{\xi}} + K\bar{\xi} = H^T \bar{p}(t)$$

- The final formulation is in the classic form of a Multiple Degrees of Freedom (MDOF) system with pseudo mass and stiffness matrices and a forcing term that is obtained from the true forces through the  $H$  matrix
- $M$  and  $K$  are two symmetric matrices that need to be computed once and for all, based on the geometric and mechanical properties of the structure
- An opportune choice of the Chebyshev polynomials (removing first 3 orders) used to expand the free function ensures that the  $M$  matrix is full rank (which property will be useful later)



# Kalman filter implementation

The next step is to implement the dynamic model obtained so far into a Kalman filter to produce accurate estimates of the structure dynamics in the presence of noisy measurements. Since the obtained model is linear, the traditional linear Kalman filter can be employed. A summary is reported in the diagram below:





# Kalman filter variables

$x$	state variable	$n \times 1$ column vector	Output
$P$	state covariance matrix	$n \times n$ matrix	Output
$z$	measurement	$m \times 1$ column vector	Input
$A$	state transition matrix	$n \times n$ matrix	System Model
$H$	state-to-measurement matrix	$m \times n$ matrix	System Model
$R$	measurement covariance matrix	$m \times m$ matrix	Input
$Q$	process noise covariance matrix	$n \times n$ matrix	System Model
$K$	Kalman Gain	$n \times m$	Internal



# Kalman filter implementation

The steps to reach the goal are the following:

1. Obtain a state space representation of the TFC-based dynamics
2. Transform from continuous to discrete-time
3. Define a suitable measurement model
4. Get the Kalman filter implementation

## 1. State Space representation

– TFC-based model:

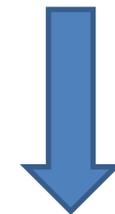
$$M \ddot{\xi}(t) + K \bar{\xi}(t) = \bar{q}(t)$$

– Define an augmented state vector:

$$\bar{\xi} = \begin{bmatrix} \xi \\ \dot{\xi} \end{bmatrix}$$



$$\tilde{M} \dot{\bar{\xi}} + \tilde{K} \bar{\xi} = \bar{Q}$$



$\tilde{M}$  is full rank

$$A = -(\tilde{M}^T \tilde{M})^{-1} \tilde{M}^T \tilde{K}$$

$$B = (\tilde{M}^T \tilde{M})^{-1} \tilde{M}^T$$

$$\dot{\bar{\xi}} = A \bar{\xi} + B \bar{Q}$$



# Kalman filter implementation

## 2. From continuous to discrete-time

$$\dot{\bar{\mathbf{E}}} = A \bar{\mathbf{E}} + B \bar{Q}$$



Assuming that the input can be considered as constant during the time step  $\Delta t$

$$\bar{\mathbf{E}}_{k+1} = \hat{A} \bar{\mathbf{E}}_k + \hat{B} \bar{Q}_k$$

$$\hat{A} = e^{A\Delta t}$$

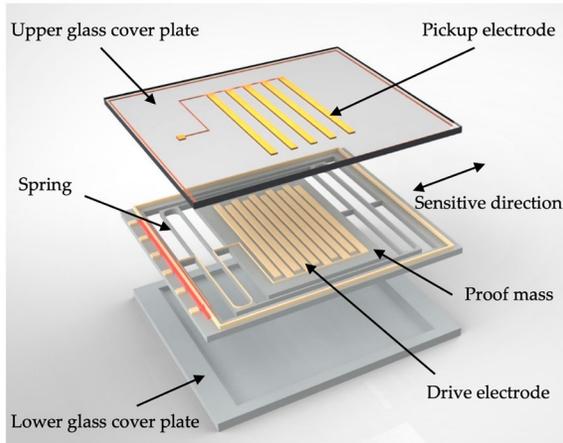
$$\hat{B} = \int_0^{\Delta t} e^{A\tau} d\tau B$$



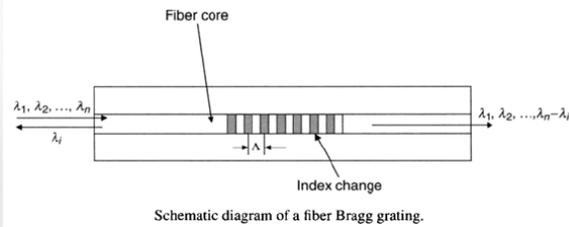
# Kalman filter implementation

## SENSORS

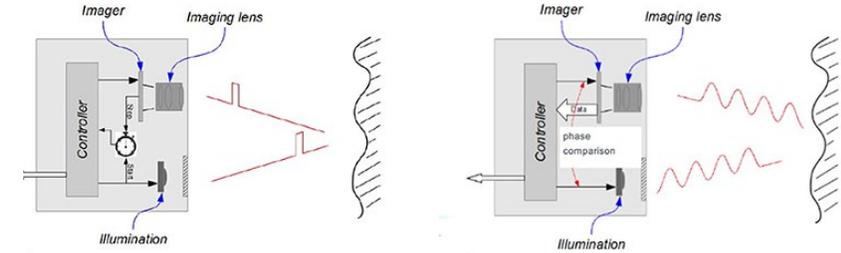
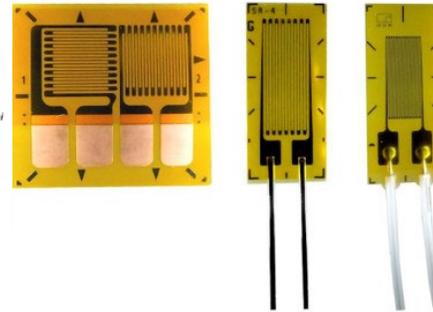
Many types of sensors:



Accelerometers



Strain gauges



Non-contact optical sensors (LASER, TOF cameras, ...)

Different measurables:

- Displacement
- Displacement velocity
- Strain
- Acceleration



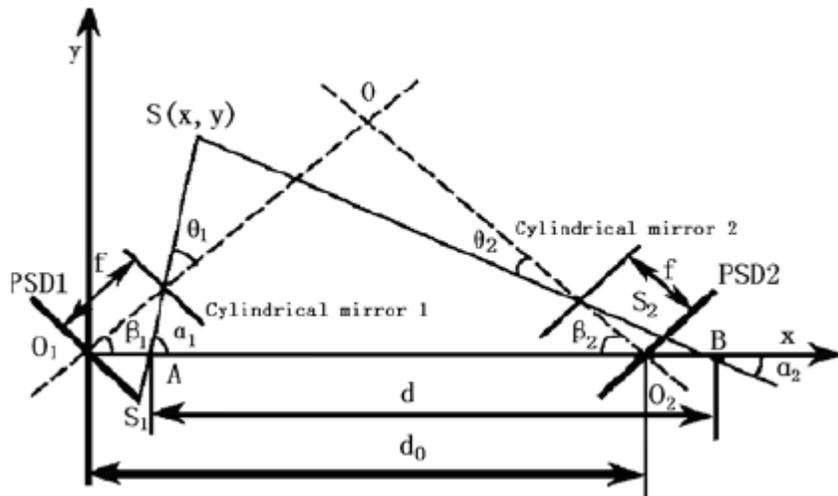
# Kalman filter implementation

## SENSORS

### Position Sensitive Detectors (PSD)

- Highly accurate in measuring the position of the light beam incident point
- Can be used for non-contact displacement measurement
- Do not require image processing

### PSD-based measurement geometry



$$\theta_1 = \arctg \frac{T_1}{f}$$

$$\theta_2 = \arctg \frac{T_2}{f}$$

$$\alpha_1 = \beta_1 + \theta_1$$

$$\alpha_2 = \beta_2 - \theta_2$$

$$d = d_0 + f \left( \frac{\sin \theta_2}{\sin \alpha_2} - \frac{\sin \theta_1}{\sin \alpha_1} \right)$$

$$x = f \frac{\sin \theta_1}{\sin \alpha_1} + \frac{d \sin \alpha_2 \cos \alpha_1}{\sin(\alpha_1 + \alpha_2)}$$

$$y = \frac{d \sin \alpha_2 \sin \alpha_1}{\sin(\alpha_1 + \alpha_2)}$$



# Kalman filter implementation

## 3. Measurement Model

- States: orthogonal polynomials expansion coefficients  $\xi$  and their first time derivatives  $\dot{\xi}$
- Measurable quantities: beam displacements and displacement velocities

What is the relationship between these two sets?

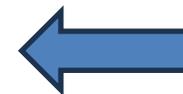


The **Constrained Expression**



The constrained expression can be used to build the Kalman filter measurement model

$$\begin{bmatrix} \bar{y} \\ \dot{\bar{y}} \end{bmatrix} = \tilde{H} \bar{\xi}$$



The matrix  $\tilde{H}$  is built using the function bases and their derivative evaluated at the measurement points



# Kalman filter implementation

## 4. The complete Kalman Filter

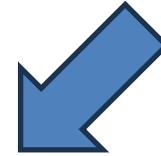
PROCESS MODEL

$$\bar{\mathbf{E}}_{k+1} = \hat{\mathbf{A}}\bar{\mathbf{E}}_k + \hat{\mathbf{B}}\bar{\mathbf{Q}}_k$$



MEASUREMENT MODEL

$$\begin{bmatrix} \bar{y} \\ \dot{\bar{y}} \end{bmatrix} = \tilde{\mathbf{H}} \bar{\mathbf{E}}$$



$$\bar{\mathbf{E}}_{k+1}^- = \hat{\mathbf{A}}\bar{\mathbf{E}}_k + \hat{\mathbf{B}}\bar{\mathbf{Q}}_k$$

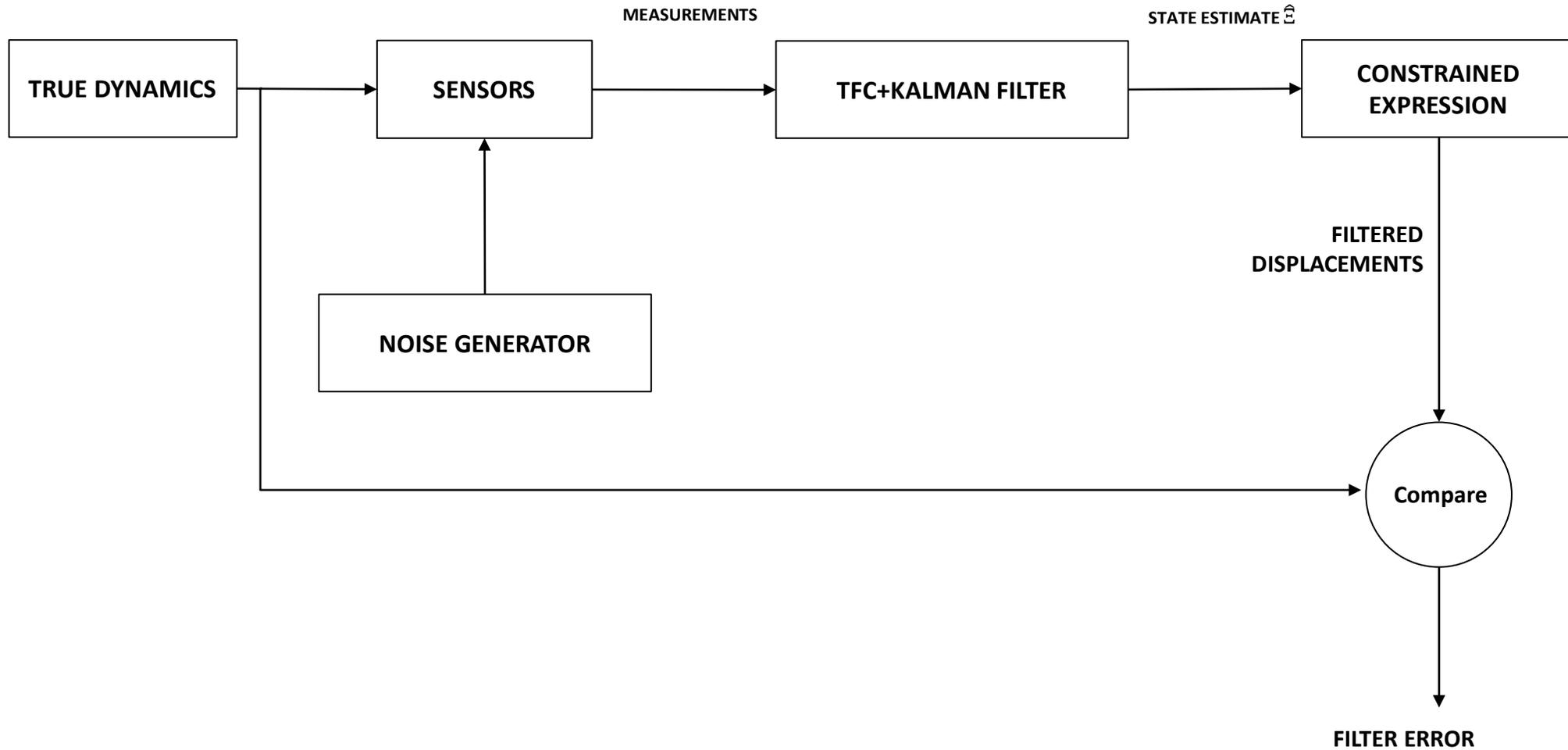
$$\bar{\mathbf{z}}_{k+1} = \begin{bmatrix} \bar{y} \\ \dot{\bar{y}} \end{bmatrix} - \tilde{\mathbf{H}}\bar{\mathbf{E}}_{k+1}^-$$

$$\bar{\mathbf{E}}_{k+1}^+ = \bar{\mathbf{E}}_{k+1}^- + P_{k+1}^- \tilde{\mathbf{H}}^T (\tilde{\mathbf{H}} P_{k+1}^- \tilde{\mathbf{H}}^T + R)^{-1} \bar{\mathbf{z}}_{k+1}$$



# Simulation and Results

## SIMULATION WORKFLOW



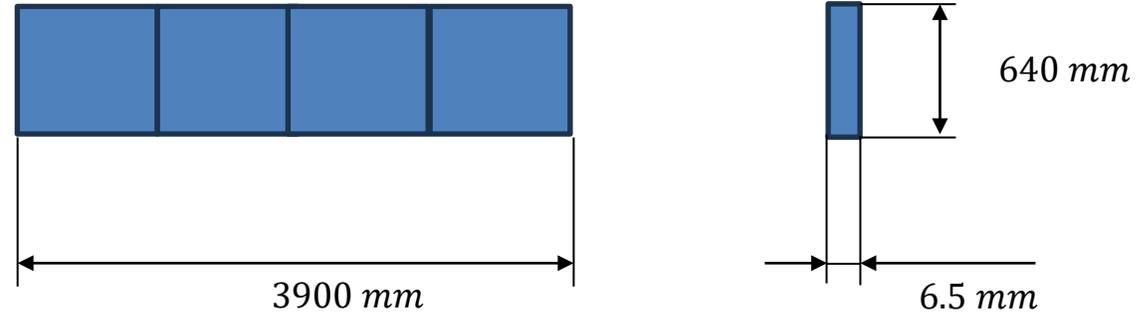


# Simulation and Results

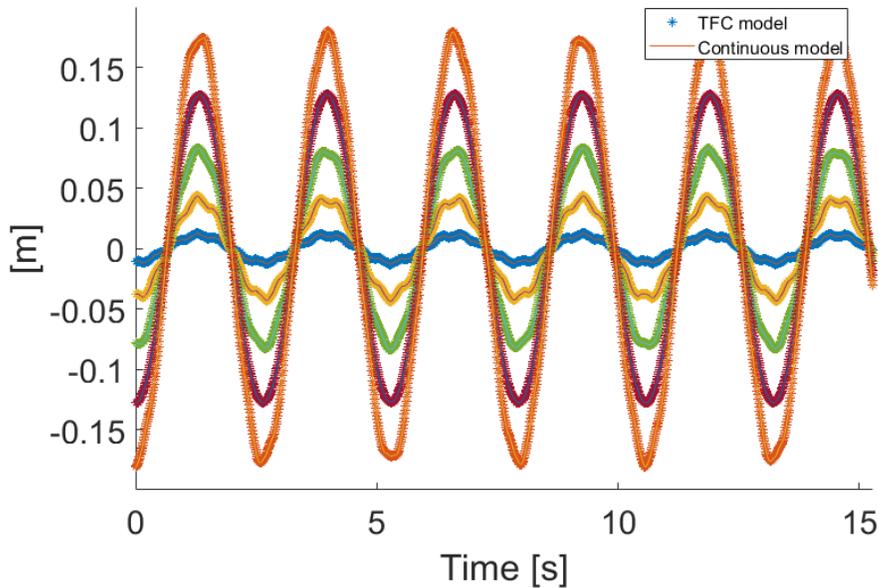
## Simulated beam characteristics

$$\begin{cases} E = 22.4 \text{ GPa} \\ \rho = 740.4 \text{ kg/m}^3 \end{cases}$$

$$\begin{cases} L = 3.90 \text{ m} \\ A_s = 4.16 \times 10^{-3} \text{ m}^2 \\ I = 1.44 \times 10^{-9} \text{ m}^4 \end{cases}$$



## Displacement time histories



TFC vs continuous model comparison

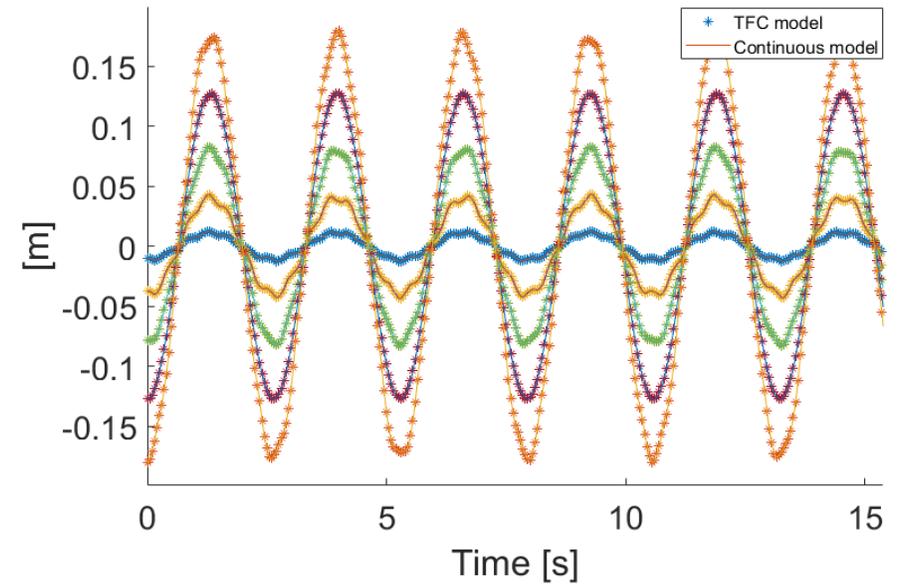
$\Delta t = 0.001 \text{ s}$



$\Delta t = 0.01 \text{ s}$



## Displacement time histories





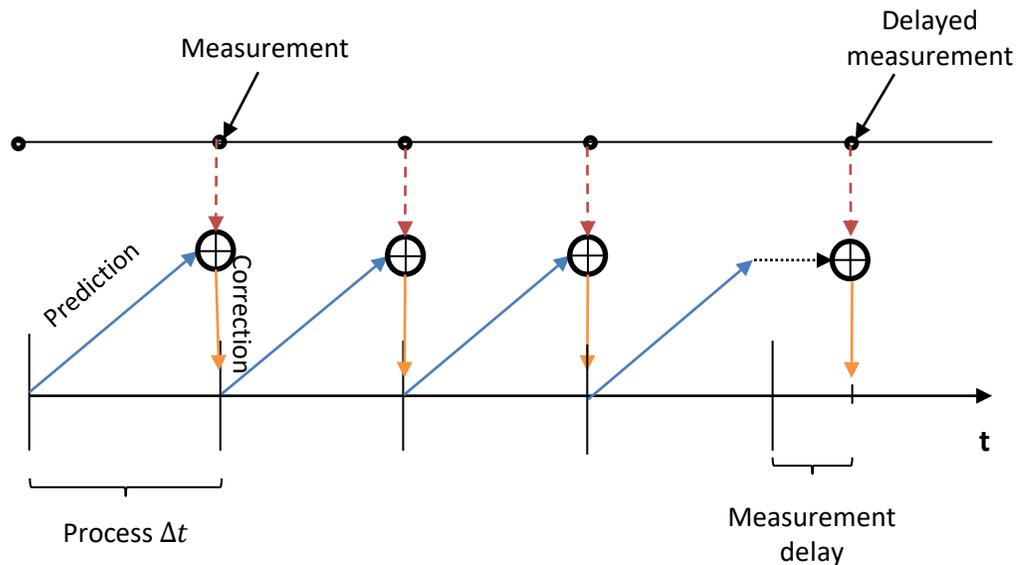
# Simulation and Results

## THE MEASUREMENT DELAY PROBLEM

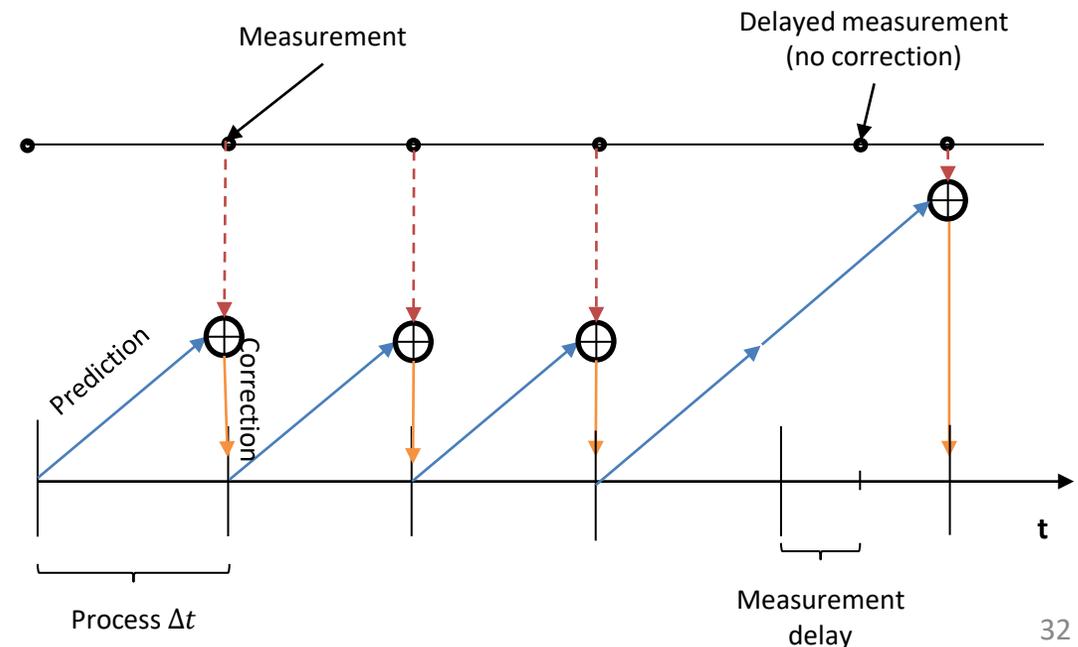
- What happens if a measurement is delayed?
- Is the filter robust against this perturbation?
- A simple mitigation strategy is tested

1. The measurement time is compared with the prediction time
2. If the delay is above a threshold, it is ignored, and the filter relies on the prediction alone for the estimate at that time

### No mitigation



### With mitigation





# Simulation and Results

## Tests carried out

- No delayed measurements
  - Free response
  - Forced response
- Delayed measurements
  - Without mitigation
  - With mitigation

In the simulator the generation of delayed measurements is governed by the following 2 parameters:

- 1. Delay probability:** the probability a delay in the measurement occurs at each time instant (set to 0.05)
- 2. Delay  $\mu$  and  $\sigma$ :** mean and standard deviation of the probability distribution (Gaussian) from which the delay is drawn (computed based on the filter time step)

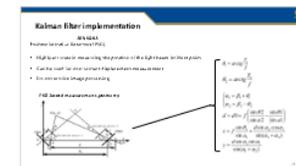
## Measurement noise (typical PSD)

$$\sigma = 0.150 \text{ mm}$$

Number of sensors = 10



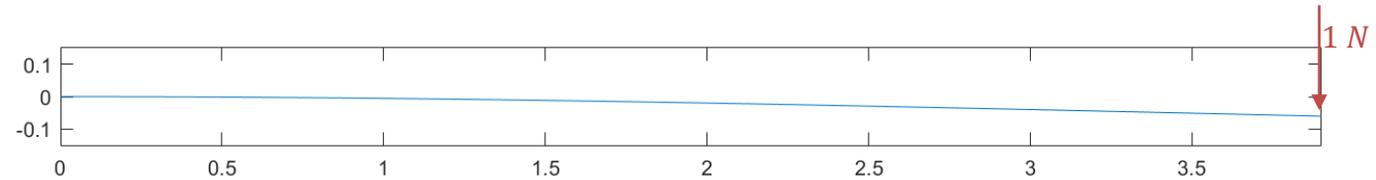
The covariance is propagated through a linearized version of the measurement model shown before to get the displacement uncertainties





# Simulation and Results

## Free response (no delay)

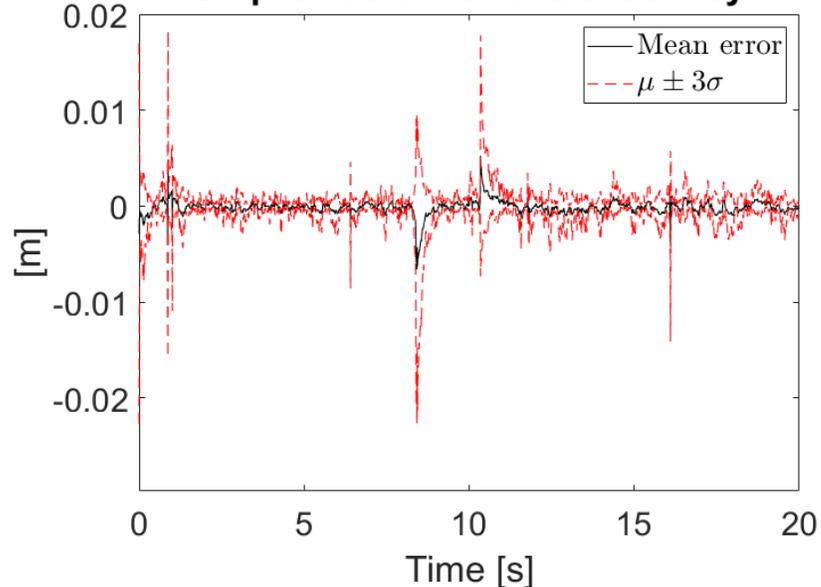


### Initial conditions

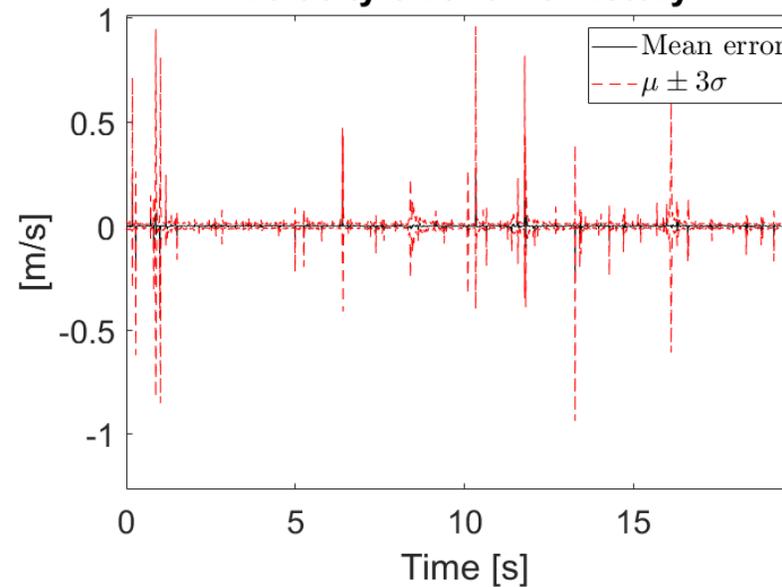
- Static deformed shape under a tip load of  $P = 1 \text{ N}$
- Zero initial velocity

$$y(x, 0) = \frac{Px^2}{6EI} (3L - x)$$
$$\dot{y}(x, 0) = 0$$

Displacement error time history



Velocity error time history

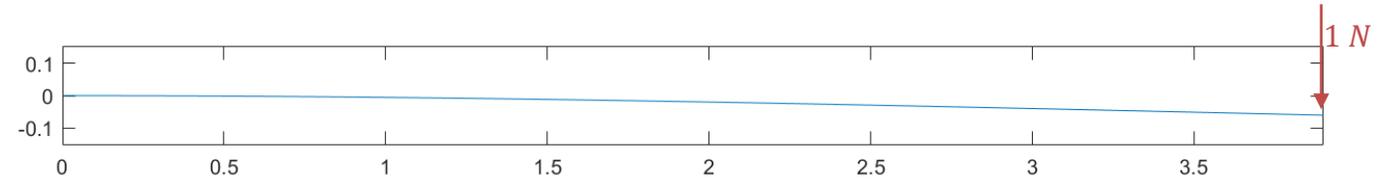


**Time step**  
 $\Delta t = 0.01 \text{ s}$



# Simulation and Results

## Free response (no delay)

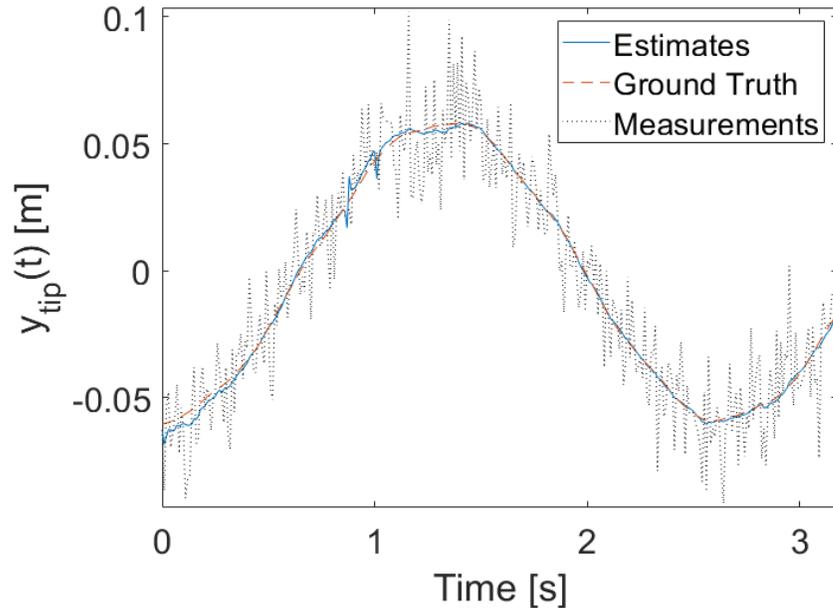


### Initial conditions

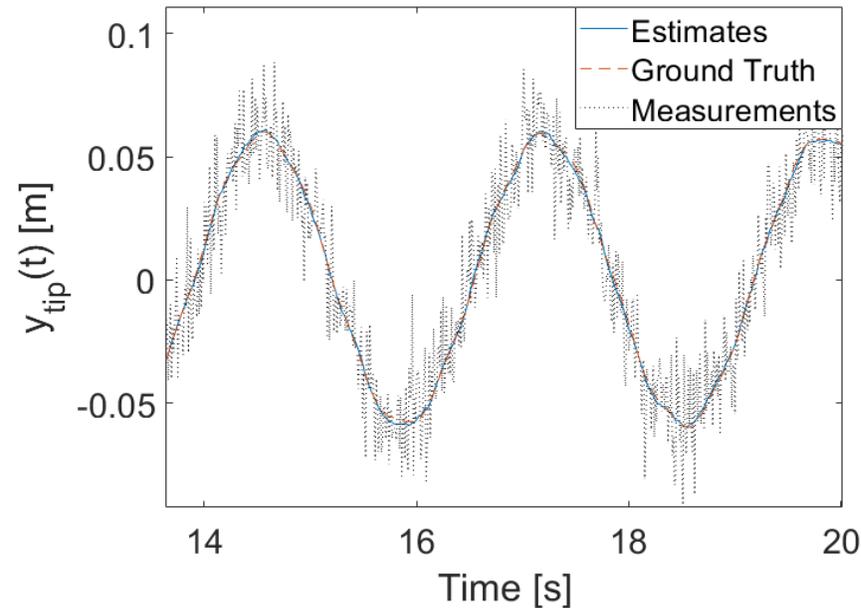
- Static deformed shape under a tip load of  $P = 1\text{ N}$
- Zero initial velocity

$$y(x, 0) = \frac{Px^2}{6EI} (3L - x)$$
$$\dot{y}(x, 0) = 0$$

Free tip dynamics



Free tip dynamics



**Time step**  
 $\Delta t = 0.01\text{ s}$



# Simulation and Results

## Forced response (no delay)

### Initial conditions

- Zero displacement
- Zero initial velocity

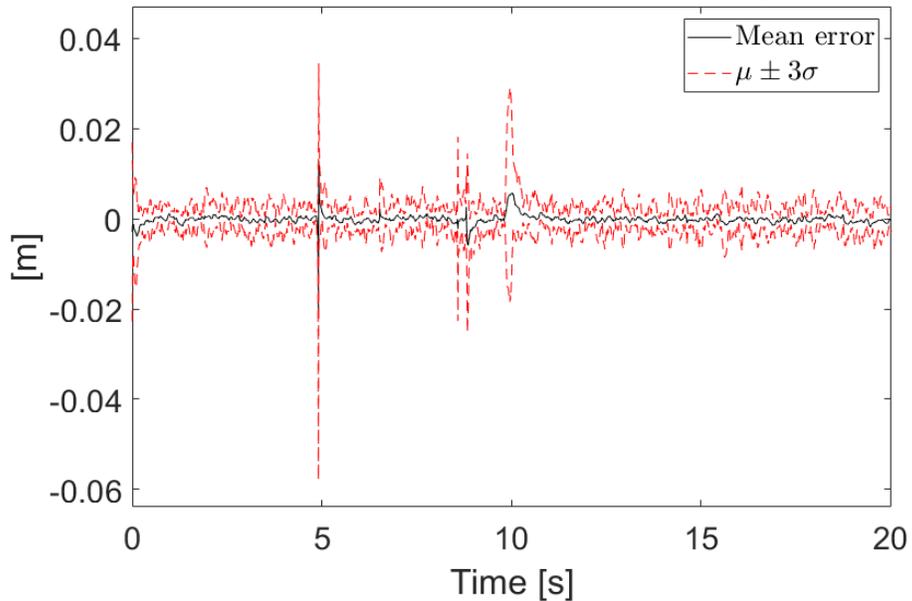
### Forcing input

Cosine tip load

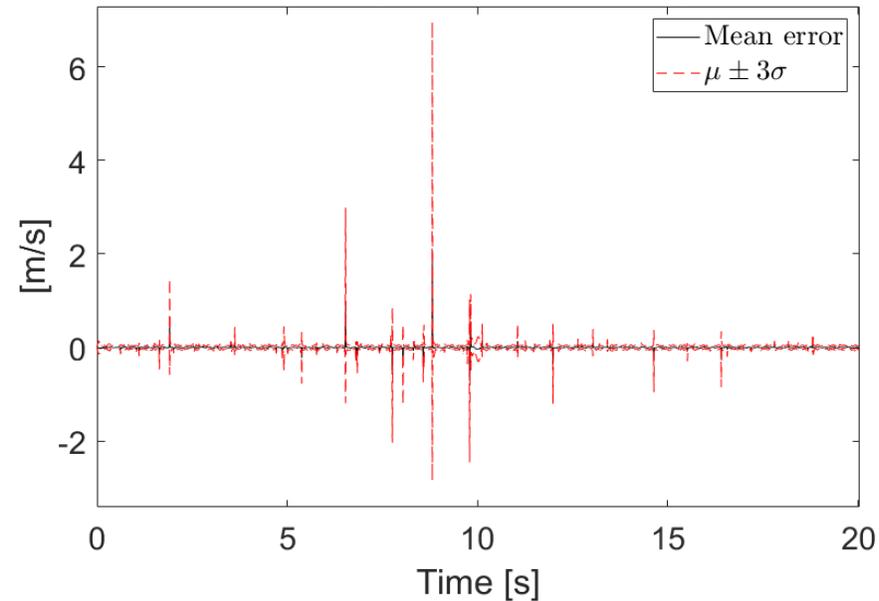
$$q(x, t) = \begin{cases} F_0 \cos(2\pi ft) & x = L \\ 0 & x \neq L \end{cases}$$

$F_0 = 4 \text{ N}, f = 5 \text{ Hz}$

Displacement error time history



Velocity error time history



**Time step**  
 $\Delta t = 0.01 \text{ s}$



# Simulation and Results

## Forced response (no delay)

### Initial conditions

- Zero displacement
- Zero initial velocity

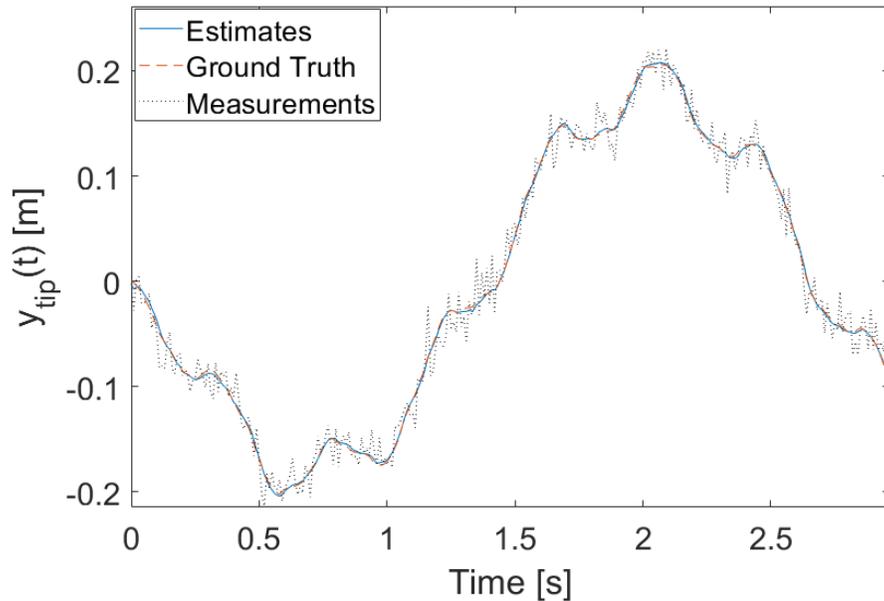
### Forcing input

Cosine tip load

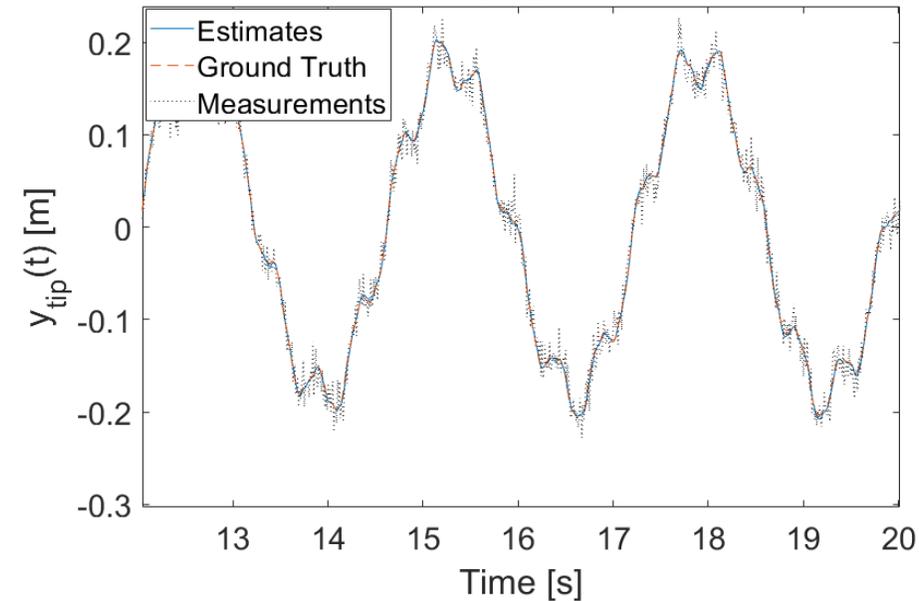
$$q(x, t) = \begin{cases} F_0 \cos(2\pi ft) & x = L \\ 0 & x \neq L \end{cases}$$

$F_0 = 4 \text{ N}, f = 5 \text{ Hz}$

Free tip dynamics



Free tip dynamics



**Time step**  
 $\Delta t = 0.01 \text{ s}$

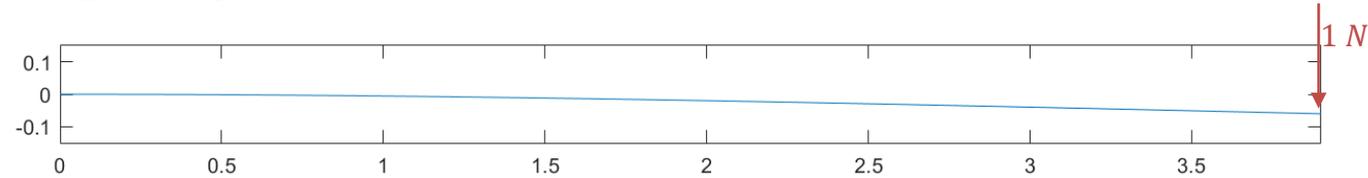


# Simulation and Results

## Free response with delayed measurements (no mitigation)

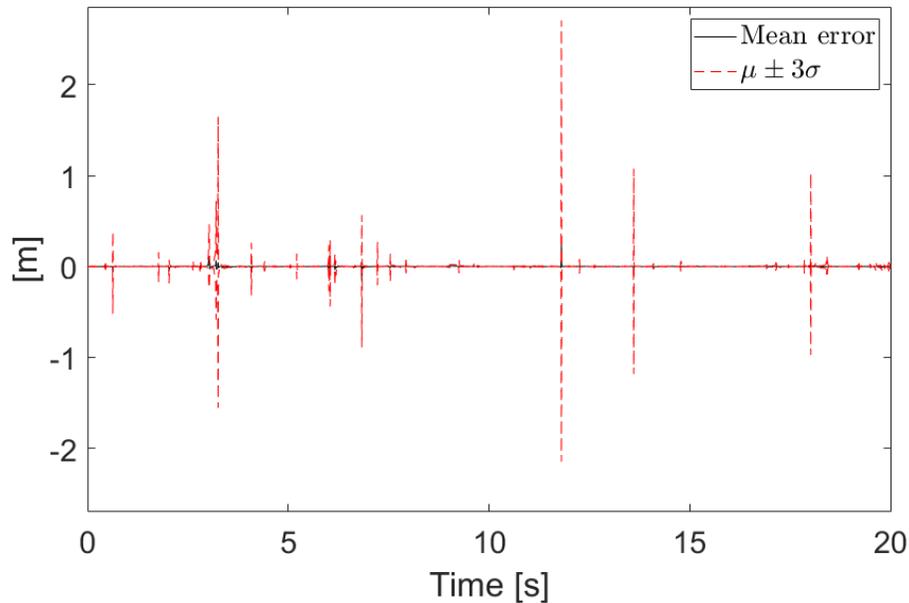
### Initial conditions

- Static deformed shape under a tip load of  $P = 1 \text{ N}$
- Zero initial velocity

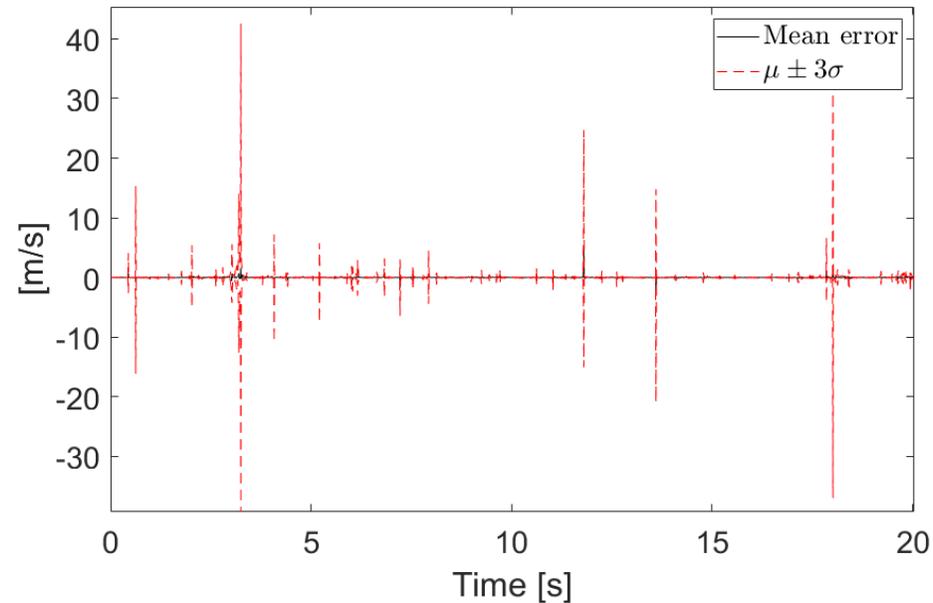


$$y(x, 0) = \frac{Px^2}{6EI} (3L - x)$$
$$\dot{y}(x, 0) = 0$$

Displacement error time history



Velocity error time history

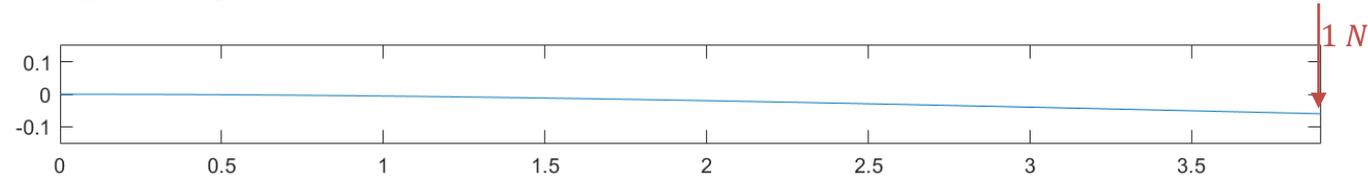


**Time step**  
 $\Delta t = 0.01 \text{ s}$



# Simulation and Results

## Free response with delayed measurements (no mitigation)

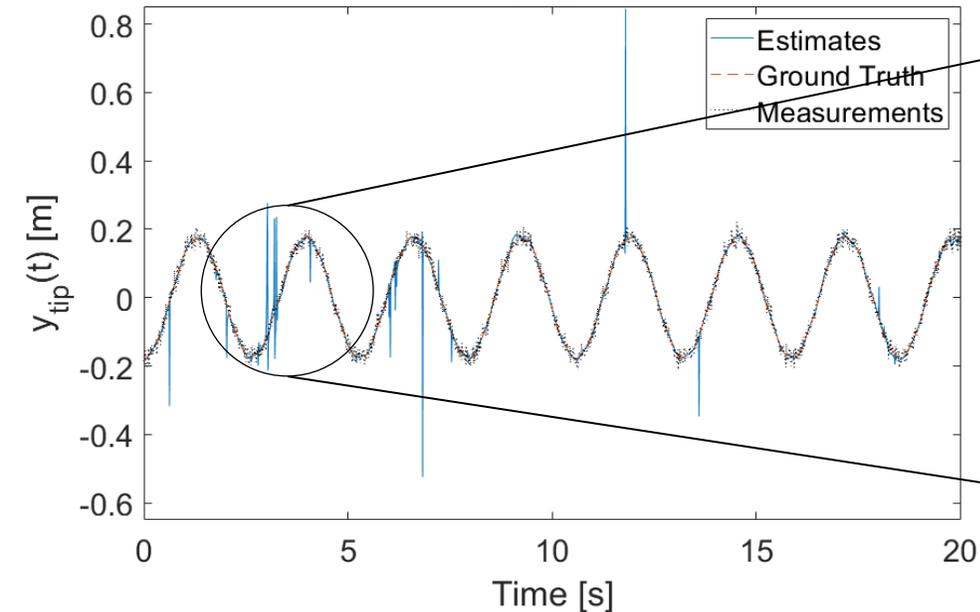


### Initial conditions

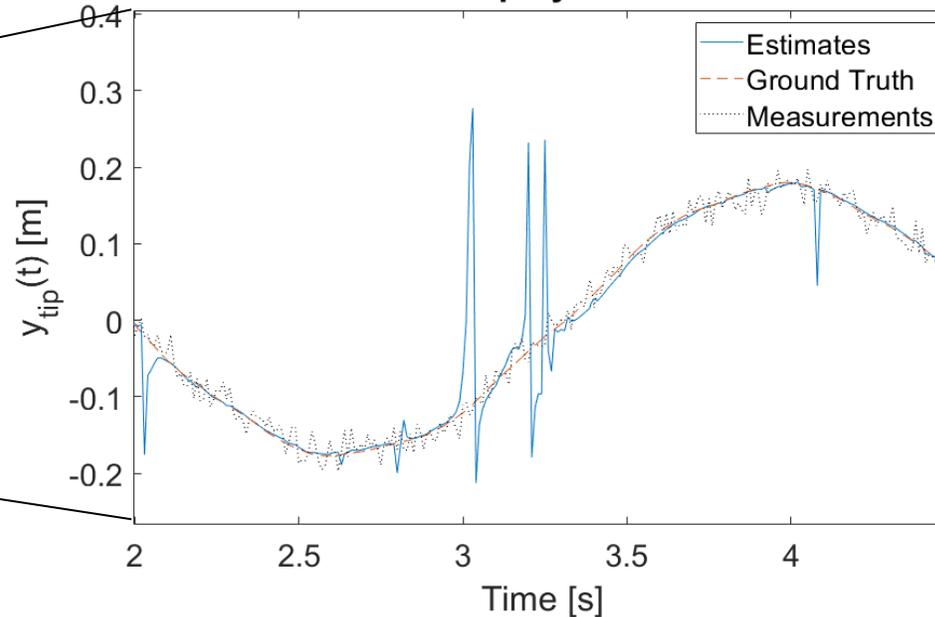
- Static deformed shape under a tip load of  $P = 1\text{ N}$
- Zero initial velocity

$$y(x, 0) = \frac{Px^2}{6EI} (3L - x)$$
$$\dot{y}(x, 0) = 0$$

Free tip dynamics



Free tip dynamics



Time step  
 $\Delta t = 0.01\text{ s}$



# Simulation and Results

## Free response with delayed measurements (with mitigation)

### Initial conditions

- Zero displacement
- Zero initial velocity

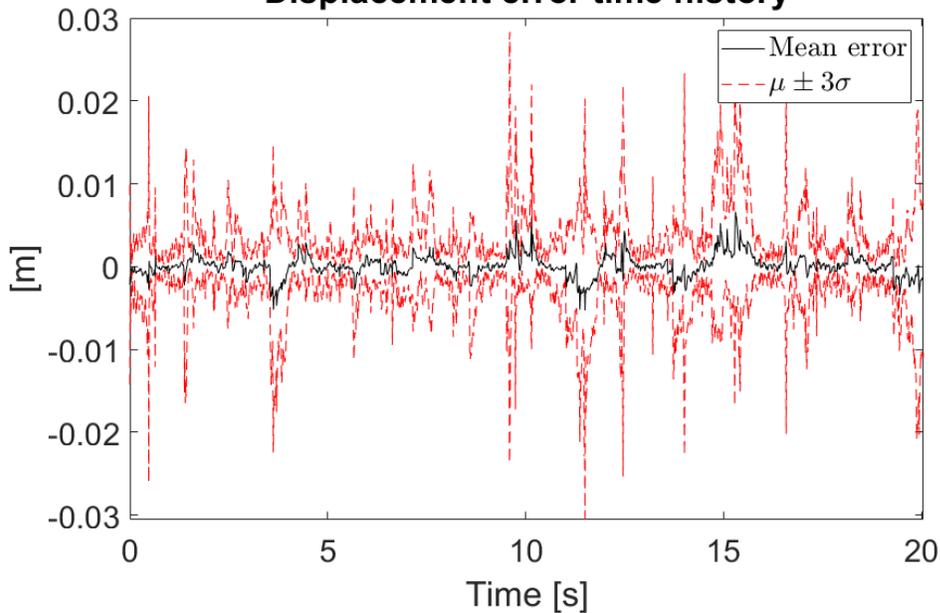
### Forcing input

Cosine tip load

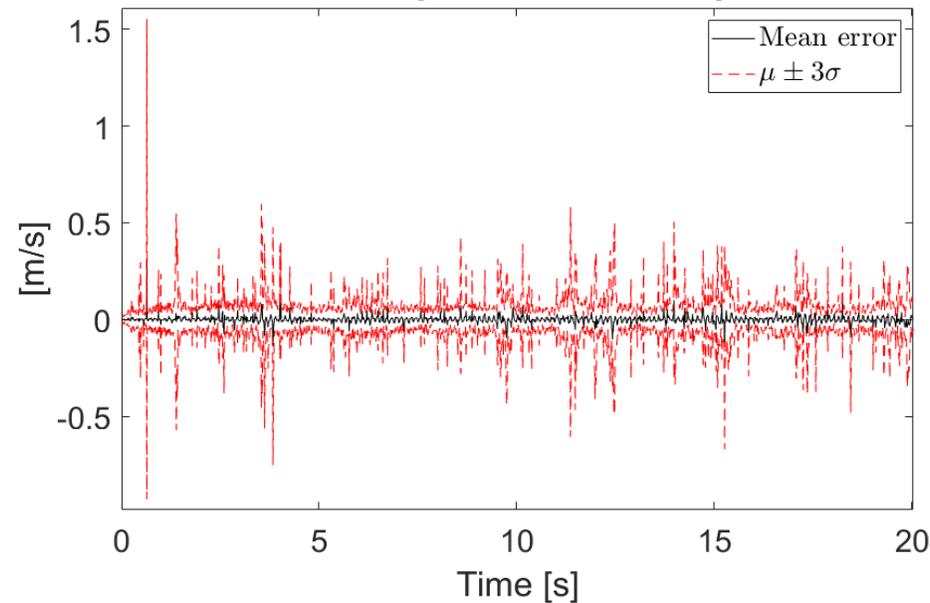
$$q(x, t) = \begin{cases} F_0 \cos(2\pi ft) & x = L \\ 0 & x \neq L \end{cases}$$

$F_0 = 4 \text{ N}, f = 5 \text{ Hz}$

Displacement error time history



Velocity error time history



**Time step**  
 $\Delta t = 0.01 \text{ s}$



# Simulation and Results

## Free response with delayed measurements (with mitigation)

### Initial conditions

- Zero displacement
- Zero initial velocity

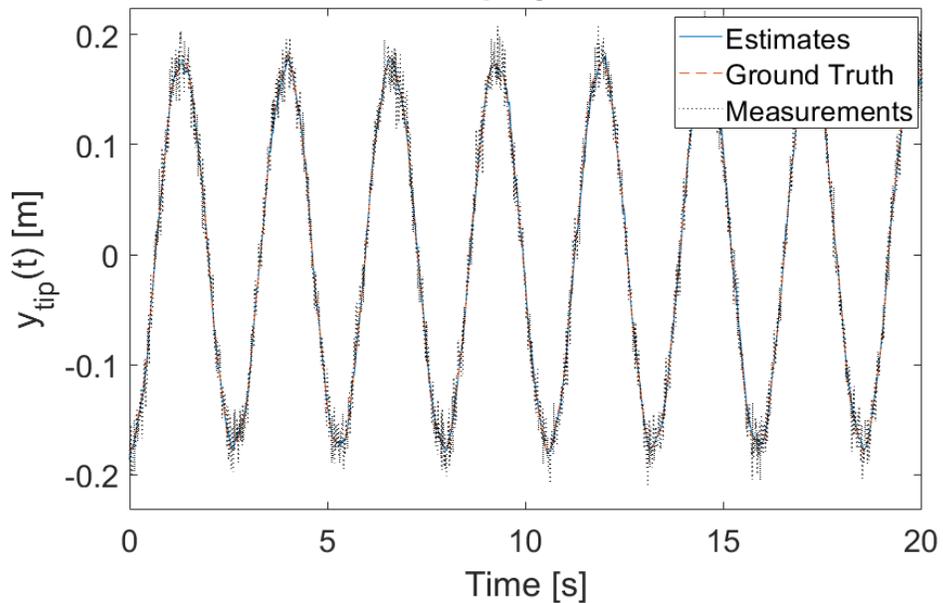
### Forcing input

Cosine tip load

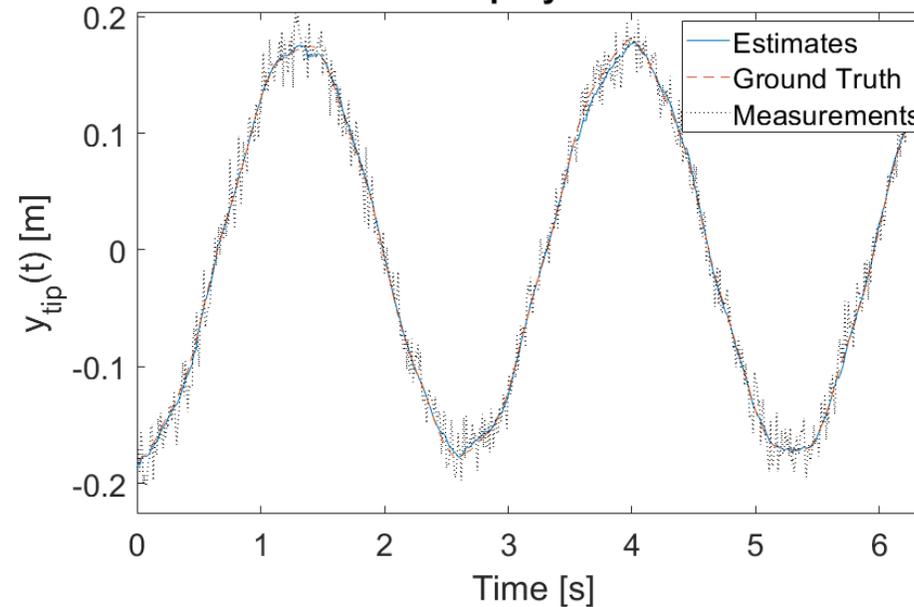
$$q(x, t) = \begin{cases} F_0 \cos(2\pi f t) & x = L \\ 0 & x \neq L \end{cases}$$

$F_0 = 4 \text{ N}, f = 5 \text{ Hz}$

Free tip dynamics



Free tip dynamics



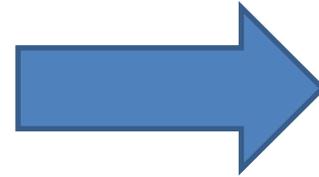
**Time step**  
 $\Delta t = 0.01 \text{ s}$



## Conclusions

## Theory of Functional Connections

**Euler-Bernoulli beam  
dynamics PDE  
(infinite-dimensional model)**



**Compact finite-dimensional  
structure dynamics model**

### Advantages

- Limited number of state variables (reduced computational complexity)
- Generality of the approach

### For the future

- New implementations of the models
- Extension to more complex structures
- Laboratory experiments

QUESTIONS?

THANK YOU!