An introduction to experimental continuation

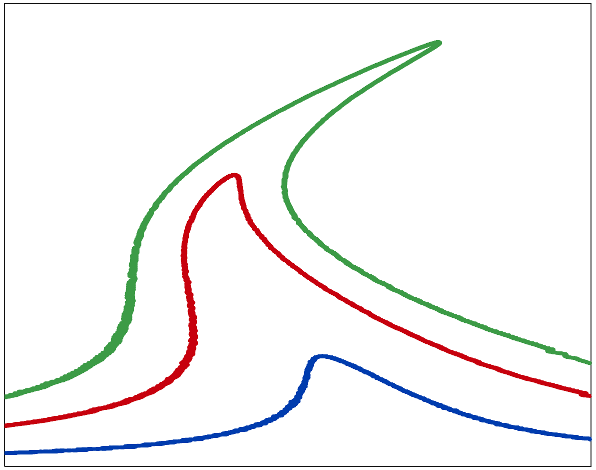
Ghislain Raze

Space Structures and Systems Lab

Aerospace and Mechanical Eng. Dept.

University of Liège, Belgium

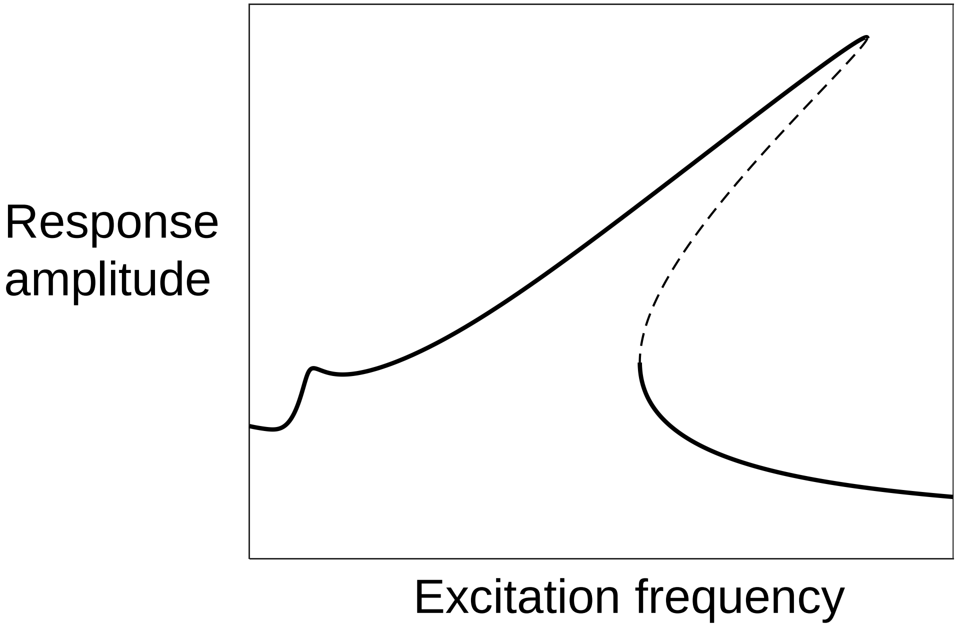
g.raze@uliege.be



ADVANCED STRUCTURAL DYNAMICS – 17 Jun 2025

Challenges in nonlinear vibration testing

Experimental NFRs can feature a **complicated shape** and **unstable branches**.



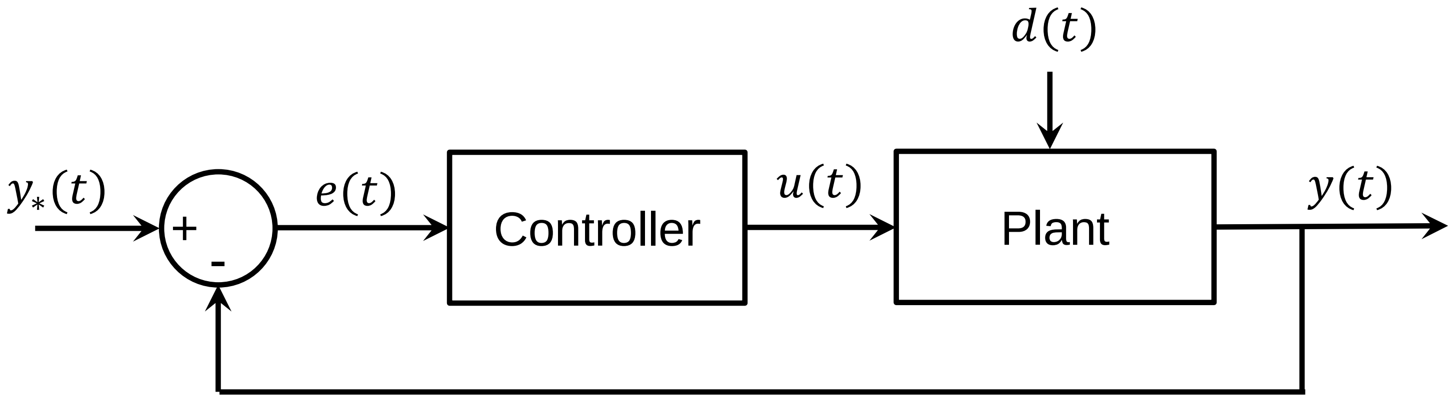
To address these issues, we need

- something to **navigate** the frequency response
- something to **stabilize** its unstable parts

How can I stabilize my experiment?

Non-invasive control

Feedback control – crash course



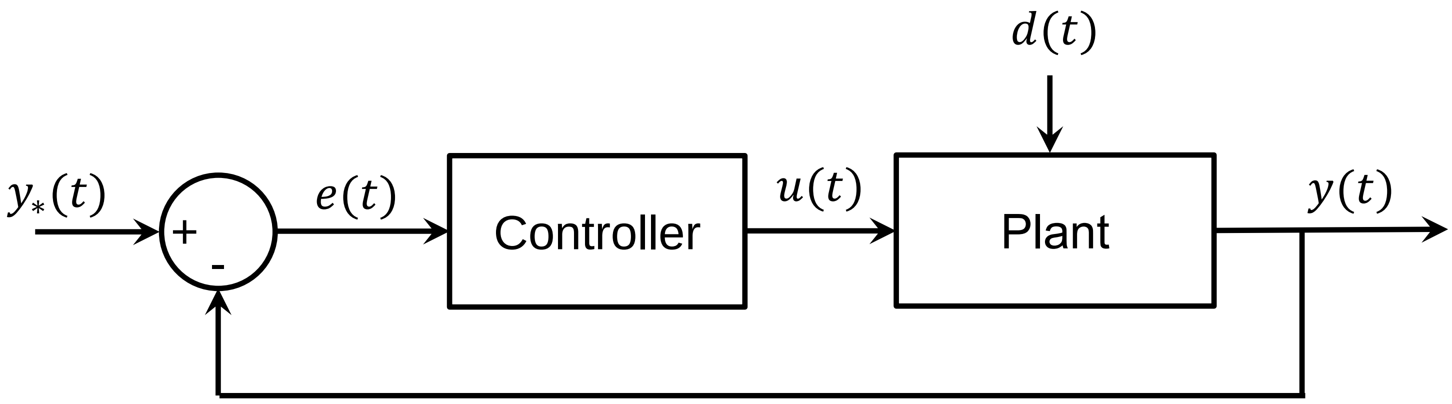
Let us imagine that we wish to set the output of a system (called the plant) $y(t)$ to $y_*(t)$. We can adjust the input $u(t)$, but the disturbance $d(t)$ is unknown.

We can add a controller that will drive the system as a function of the error,

$$e(t) = y_*(t) - y(t)$$

and will steer the output $y(t)$ toward the reference $y_*(t)$.

Feedback control for a linear plant



If the open-loop plant is LTI, its output can be expressed in the Laplace domain

$$Y(s) = H_{YU}(s)U(s) + H_{YD}(s)D(s)$$

The tracking error is

$$E(s) = Y_*(s) - Y(s)$$

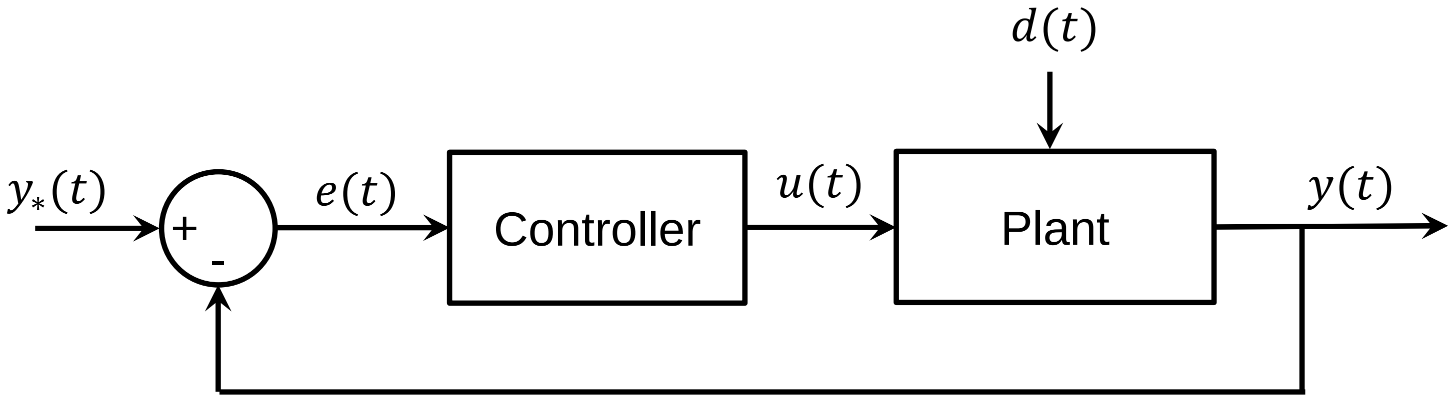
and if the controller is also LTI the input is

$$U(s) = C(s)E(s)$$

Combining these equations yields

$$Y(s) = \frac{H_{YU}(s)C(s)}{1 + H_{YU}(s)C(s)} Y_*(s) + \frac{H_{YD}(s)}{1 + H_{YU}(s)C(s)} D(s)$$

Feedback control for reference tracking



The closed-loop dynamics are described by

$$Y(s) = \frac{H_{YU}(s)C(s)}{1 + H_{YU}(s)C(s)} Y_*(s) + \frac{H_{YD}(s)}{1 + H_{YU}(s)C(s)} D(s)$$

We then have

$$\lim_{|C(s)|\to\infty} Y(s) = Y_*(s)$$

In other words, $y(t) \to y_*(t)$ if the transfer function of the controller is large enough, regardless of the disturbance $d(t)$!

Practical considerations for control

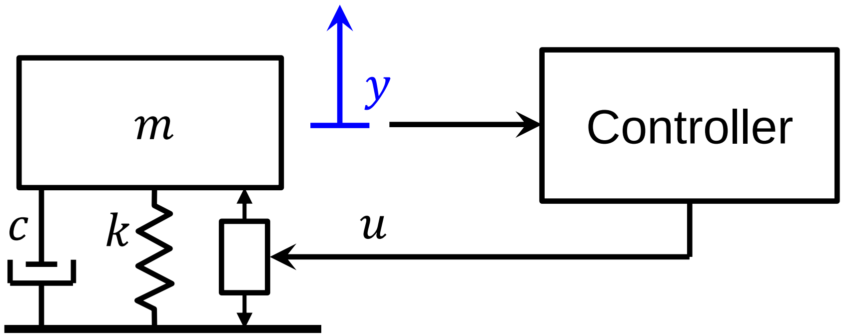
In practice, a finite gain is chosen because (among other reasons)

- sensor noise will be fully transmitted
- transients will make the controller very aggressive
- closed-loop stability must be guaranteed

Closed-loop stability can be assessed or even predicted from open-loop characteristics when the system is linear. If anything is nonlinear, things get a lot more complicated. We will not look at this problem in this lecture.

Let us now have a look at one of the most classical controller, the proportional-integral-derivative controller.

Test case – harmonic oscillator control



$$H_{YU}(s) = \frac{1}{ms^2 + cs + k}$$

$$C(s) = k_\mathrm{p} + sk_\mathrm{d} + \frac{k_\mathrm{i}}{s}$$

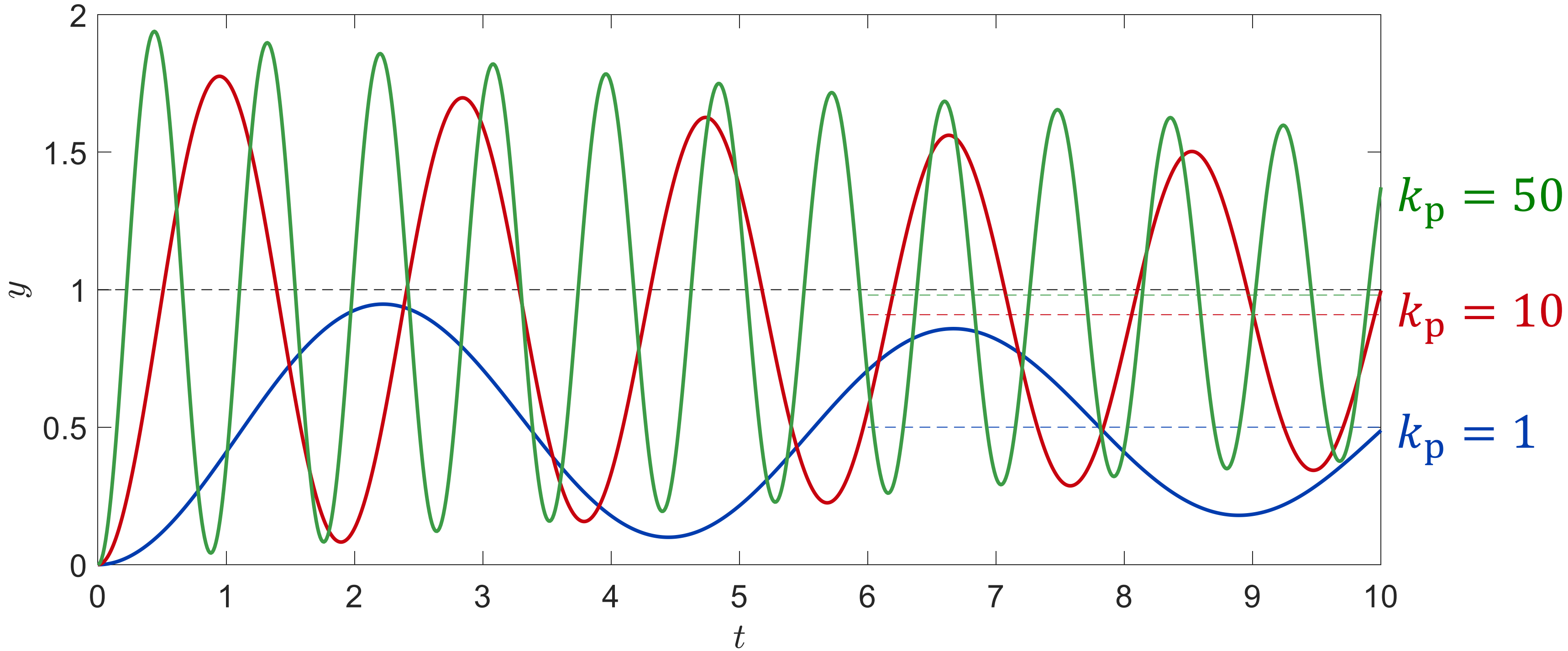
$$y_* = 1, m = 1, c = 0.1, k = 1$$

Experiment with PID control tuning for yourself!

https://ghislain_raze.pyscriptapps.com/pid-tuning/

PID control – proportional gain

$$k_\mathrm{d} = 0, k_\mathrm{i} = 0$$

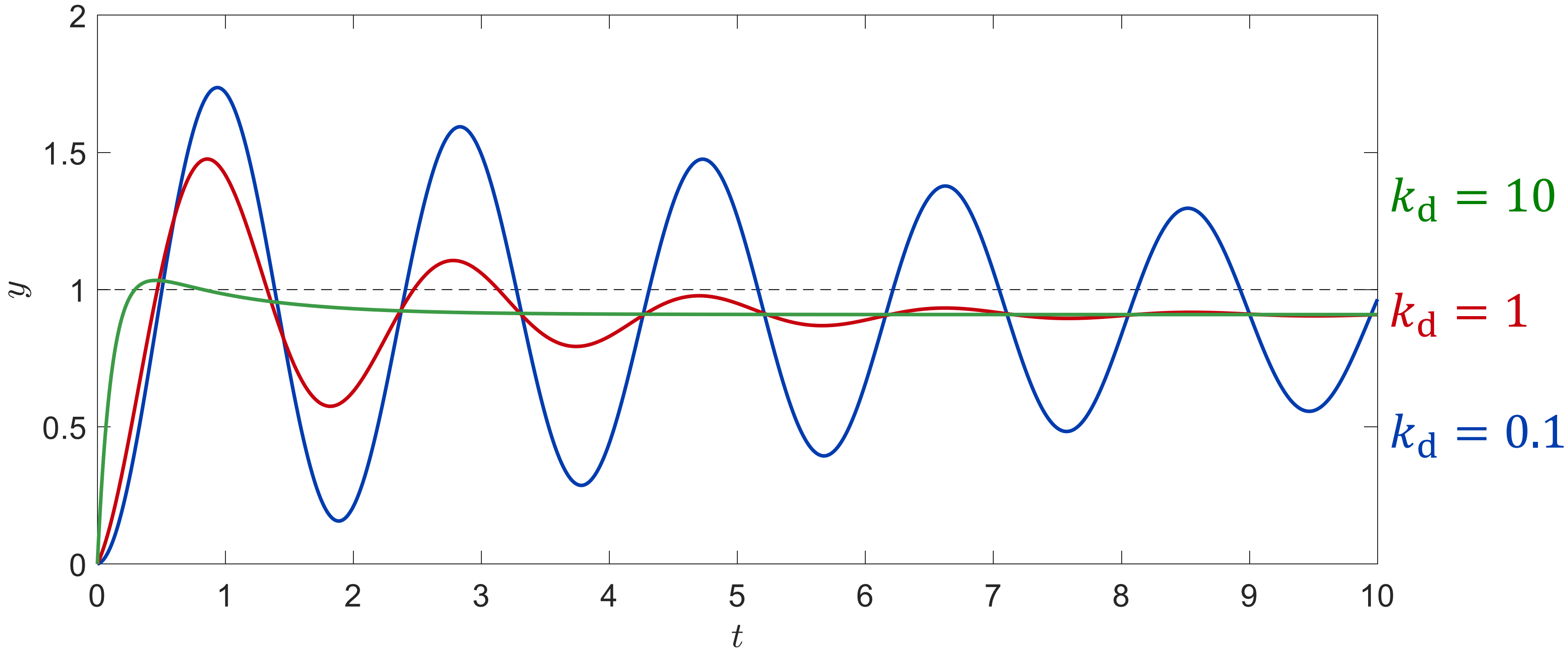


Increasing k_p leads to

- a faster response
- a smaller steady-state error
- more oscillations

PID control – derivative gain

$k_\mathrm{p} = 10, k_\mathrm{i} = 0$

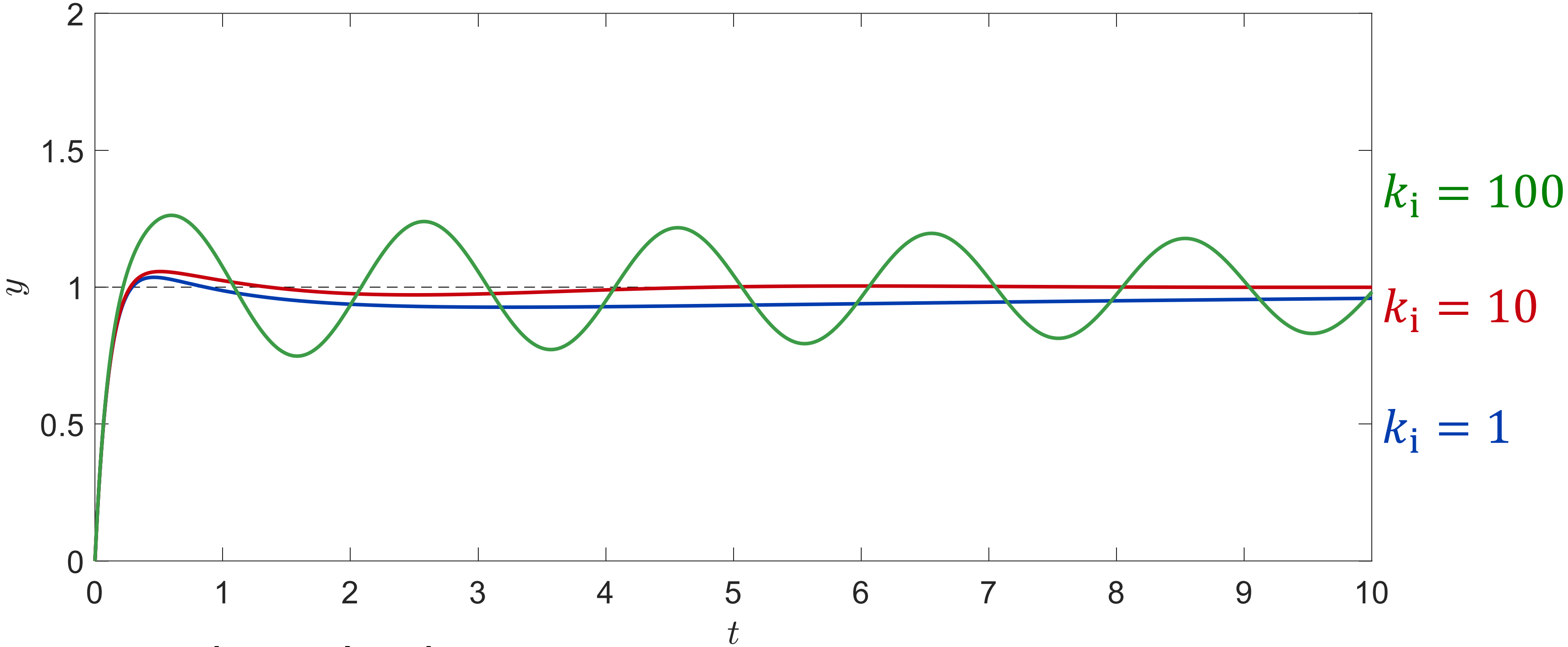


Increasing k_d leads to

- less oscillations

PID control – integral gain

$$k_{\mathrm{p}} = 10, k_{\mathrm{d}} = 10$$

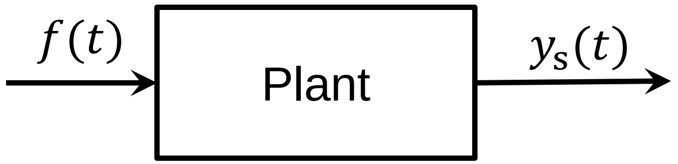


Increasing k_{i} leads to

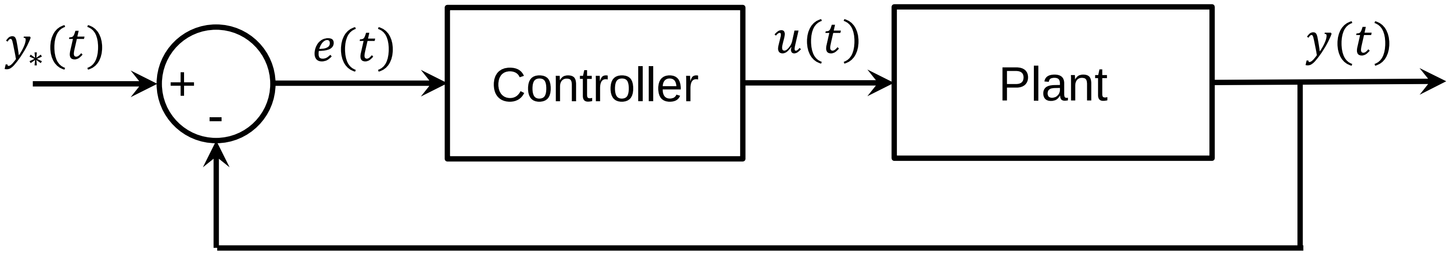
- a faster response
- zero steady-state error
- more oscillations

How to relate open- and closed-loop systems?

Open-loop system (what is of interest)



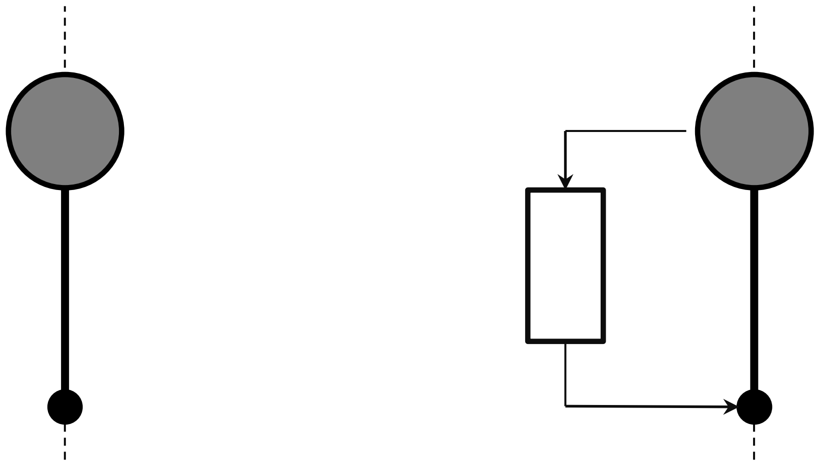
Closed-loop system (what we use for stabilization)



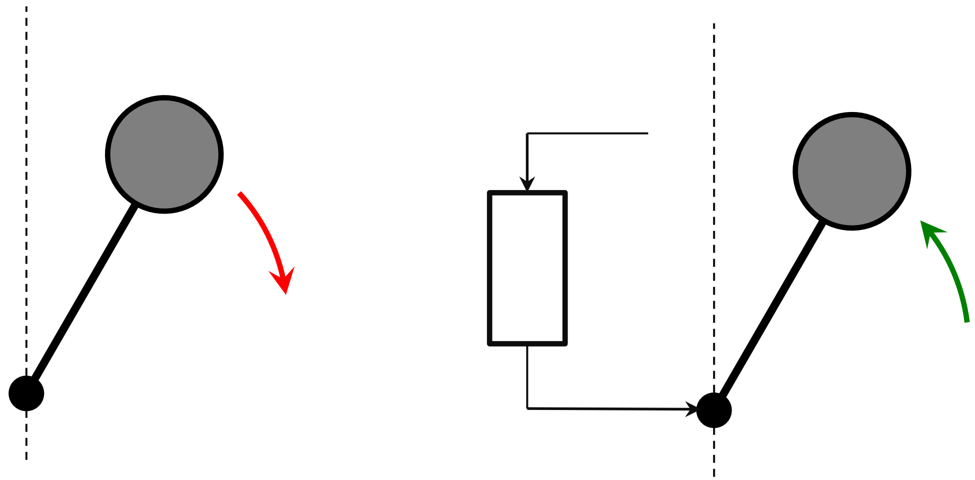
A sufficient condition for both systems to have the same steady-state response $y_s(t)$ is that $u(t) = f(t)$ when $y(t) = y_s(t)$.

In this case, the control is said to be **non-invasive**.

Non-invasive control?



At equilibrium, the controller has no action.



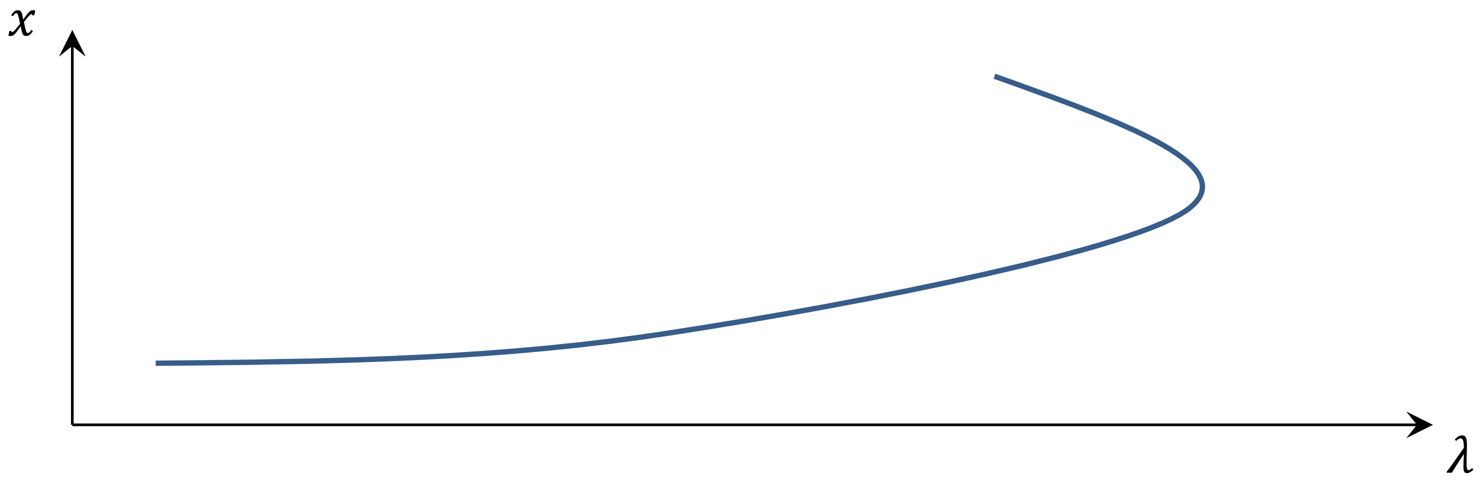
Away from equilibrium, the controller tends to bring the system back to equilibrium.

How can I measure complicated response curves?

Continuation approaches

Continuation problem

Our aim is to find a curve (the NFR) which has a complicated shape. It varies with a parameter λ (the frequency ω) but is not a function of it.



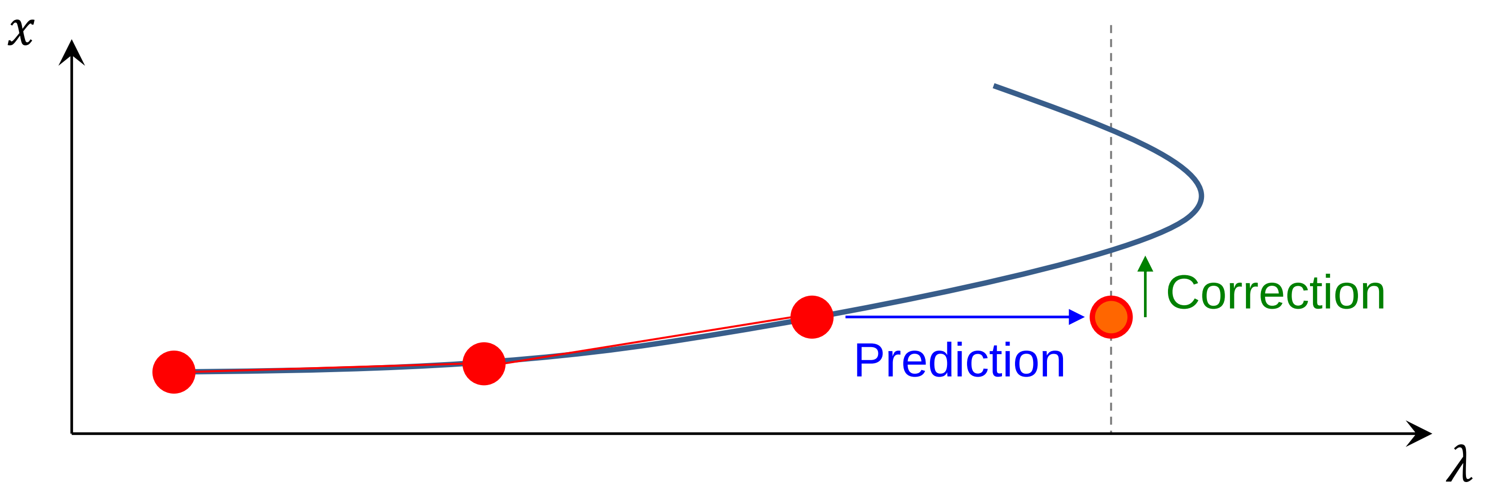
In a numerical setting, continuation is used to compute complicated responses. There exist two main variants.

Natural parameter continuation

In a numerical setting, continuation is used to compute complicated responses. There exist two main variants.

Natural parameter continuation works as follows:

1. Given a point (λ_n, x_n) on a curve, the next point is predicted as (λ_{n+1}, x_n), with $\lambda_{n+1} = \lambda_n + \Delta\lambda$.
2. This initial guess is corrected at $\lambda = \lambda_{n+1}$ until the point lies on the curve.
3. The procedure is restarted from (λ_{n+1}, x_{n+1}).



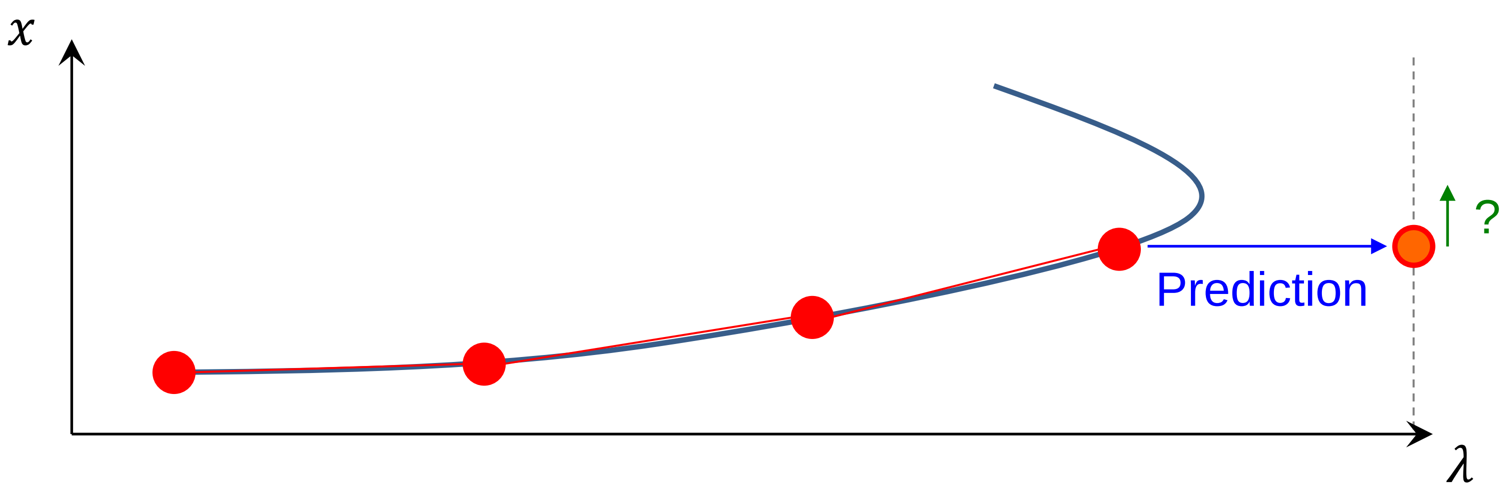
Natural parameter continuation

In a numerical setting, continuation is used to compute complicated responses. There exist two main variants

Natural parameter continuation works as follows:

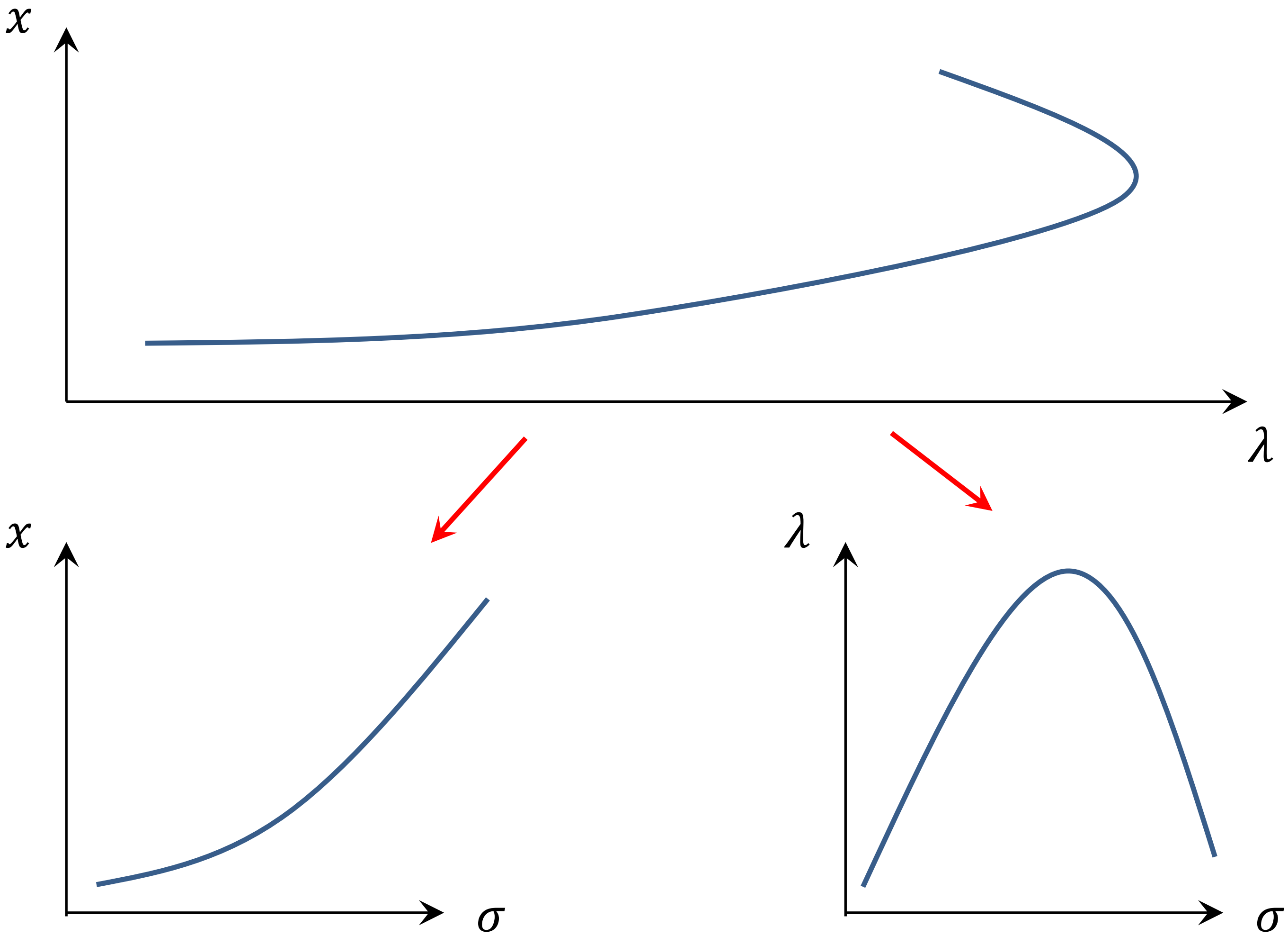
1. Given a point (λ_n, x_n) on a curve, the next point is predicted as (λ_{n+1}, x_n), with $\lambda_{n+1} = \lambda_n + \Delta\lambda$.
2. This initial guess is corrected at $\lambda = \lambda_{n+1}$ until the point lies on the curve.
3. The procedure is restarted from (λ_{n+1}, x_{n+1}).

Natural parameter continuation fails when the sought curve folds.



Reparametrization

One way to avoid folding issues is to reparametrize the curve (λ, x) with a parameter σ, so it is described by $(\lambda(\sigma), x(\sigma))$.

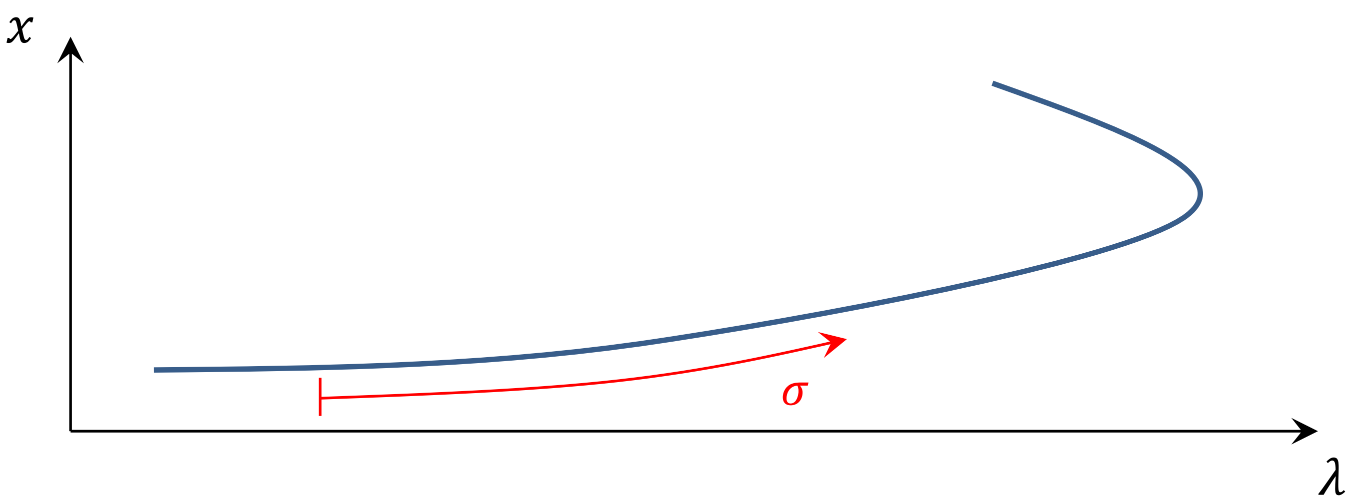


Arclength parametrization

Arclength parametrization can be used

$$\sigma = \int_{\mathcal{C}} \sqrt{dx^2 + d\lambda^2}$$

and uniquely parametrizes all points on the curve.



If we discretize this relation with finite increments,

$$\Delta\sigma = \sqrt{\Delta x^2 + \Delta\lambda^2}$$

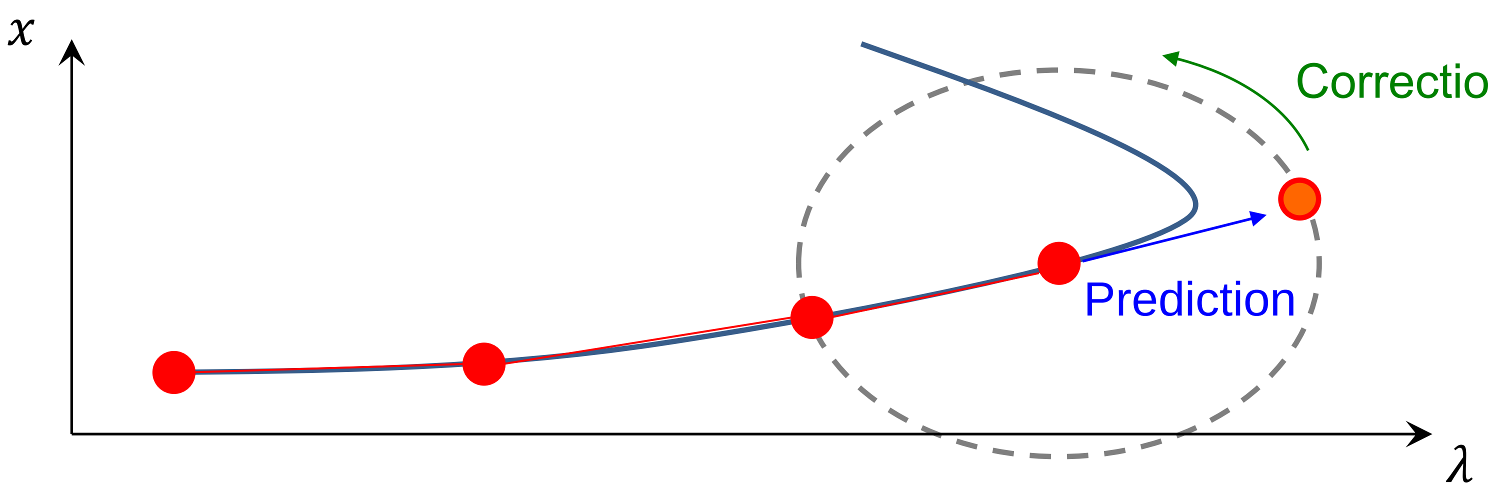
and continuation can be performed with finite increments in arclength $\Delta\sigma$.

Arclength continuation

Arclength continuation works as follows:

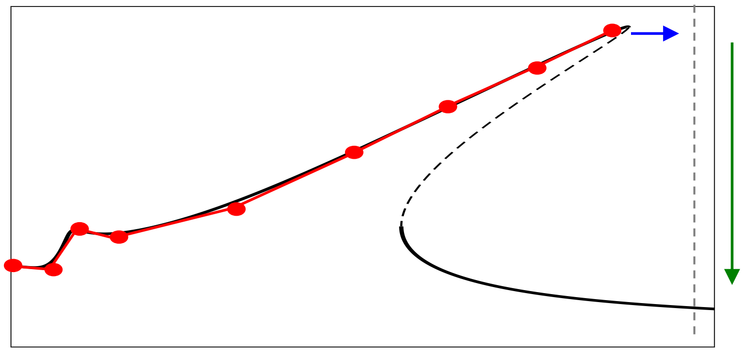
1. Given a point (λ_n, x_n) on a curve, the next point is predicted as (λ_{n+1}, x_{n+1}), with $x_{n+1} = x_n + \Delta x$, $\lambda_{n+1} = \lambda_n + \Delta\lambda$ and $\Delta\sigma^2 = \Delta x^2 + \Delta\lambda^2$. Typically, the prediction is made along the tangent of the curve.
2. This initial guess is corrected keeping $(x_{n+1} - x_n)^2 + (\lambda_{n+1} - \lambda_n)^2 = \Delta\sigma^2$ until the point lies on the curve.
3. The procedure is restarted from (λ_{n+1}, x_{n+1}).

This procedure is in general much more robust than natural parameter continuation.



Continuation in experiments

When we use swept or stepped sine excitation, we essentially perform natural parameter continuation with $\lambda \equiv \omega$. The correction is performed by the system itself.

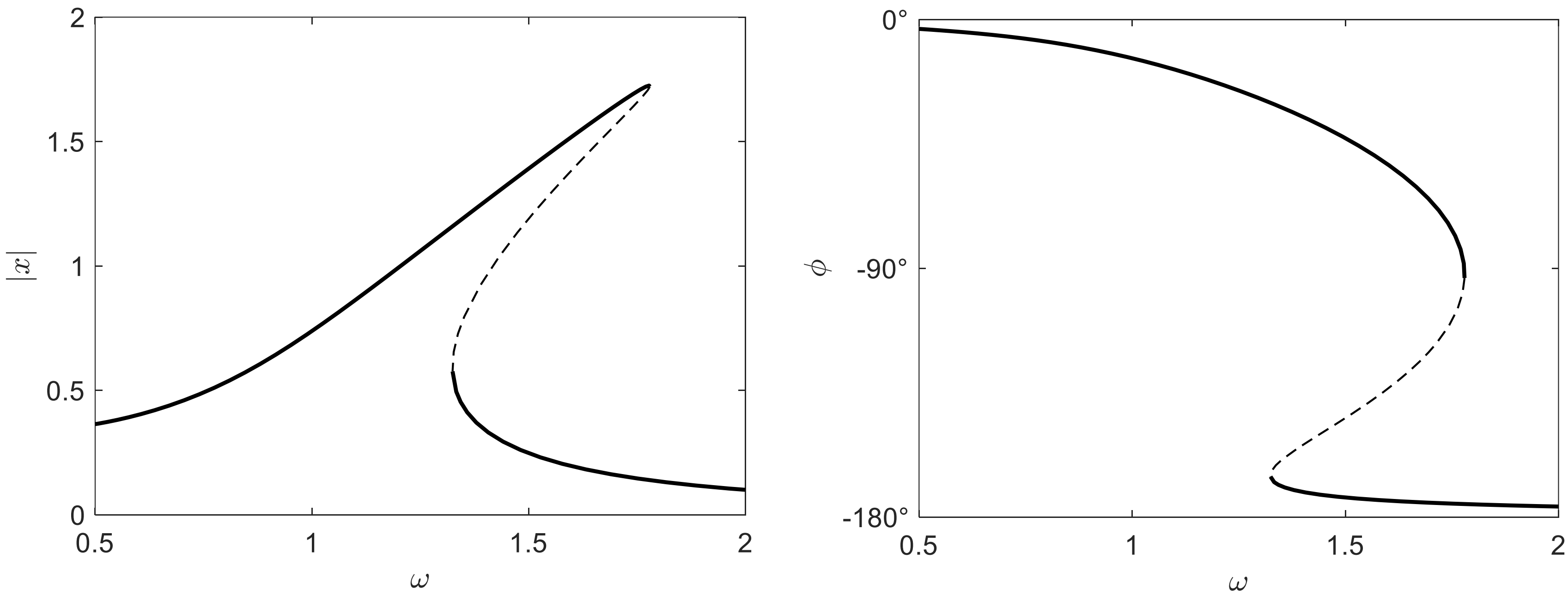


Arclength continuation (or the likes) can be used in experiments with control-based continuation (CBC) approaches, which we will not cover today.

In this lecture, we are going to focus on another natural continuation approach. We need to find a parametrization that (locally) uniquely defines the response.

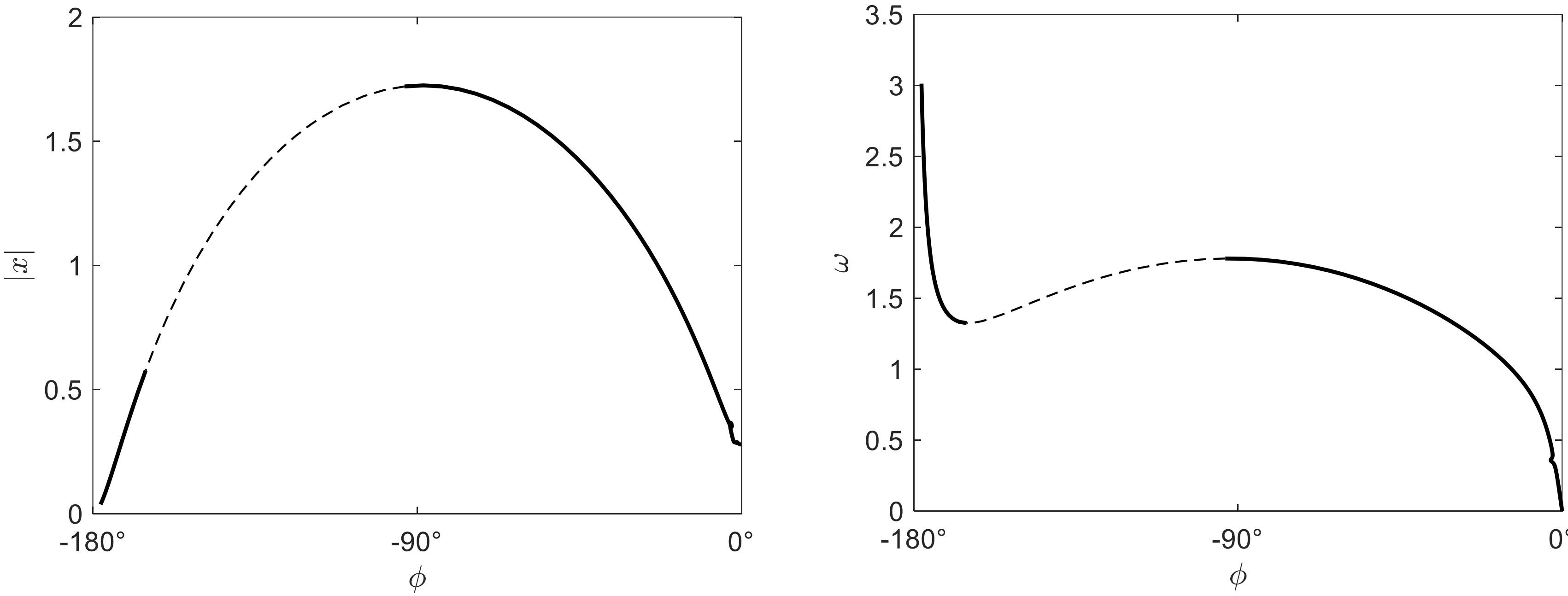
Frequency parametrization of NFRs

The response of a nonlinear system is traditionnally parametrized via its frequency…



Phase parametrization of NFRs

…but can also be parametrized via the phase difference (between first harmonics of the forcing and the response).



To one phase corresponds one response. We thus have our parametrization variable!

Phase parametrization of the NFR

To one phase corresponds one response. We thus have our parametrization variable!

$$\sigma \coloneqq \phi, \qquad a \coloneqq a(\phi), \qquad \omega \coloneqq \omega(\phi)$$

We have a way to locally uniquely define the response around a resonance. Two issues yet remain:

- How can we estimate ϕ during the experiment?
- How can we impose a phase lag between the first harmonics of the force and displacement?

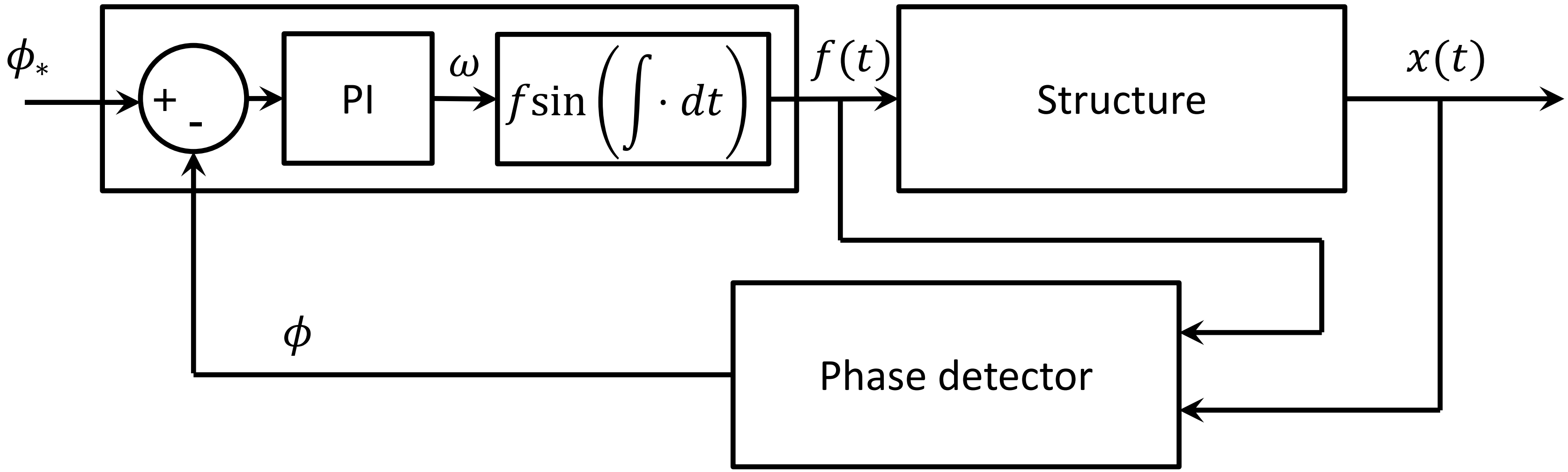
These two issues can be addressed with the use of phase-locked loops (PLLs).

How to use phase parametrization in an experiment?

Phase-locked loop control

Phase-locked loop (PLL) - implementation

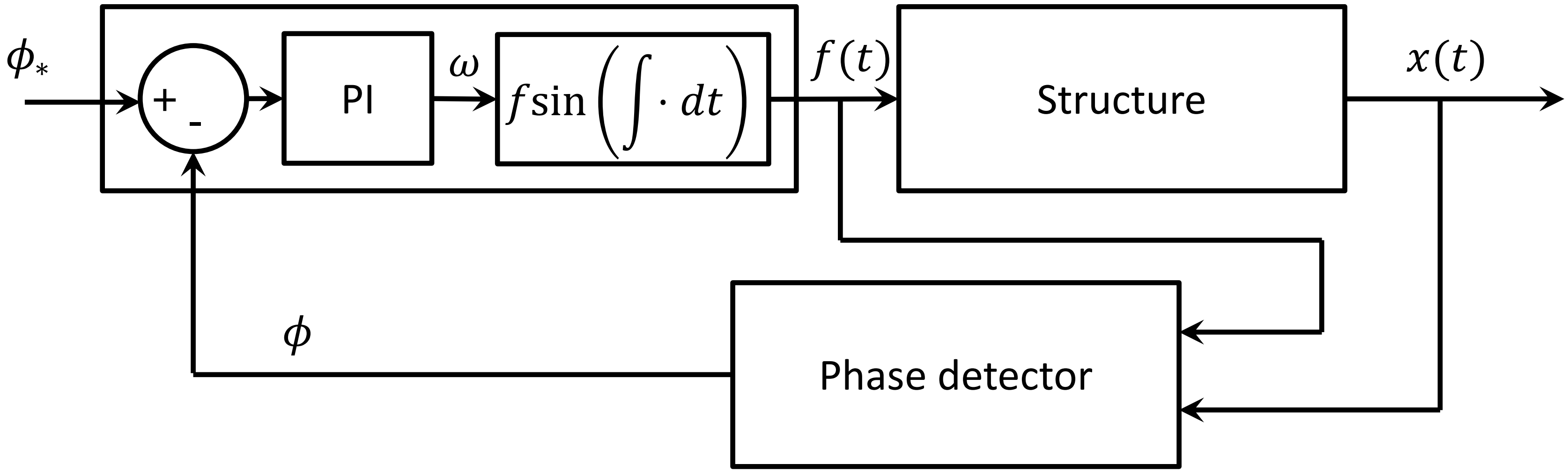
PLL are ubiquitous in electronics and can be used in an experimental continuation framework.



A feedback loop is used to enforce a prescribed phase lag ϕ_* between the forcing $f(t)$ and response $x(t)$.

This subjects the structure to the same conditions as an open-loop test, but modifies its stability. The PLL is able to stabilize responses that are open-loop unstable.

Phase-locked loop (PLL) – non-ivasiveness



The PI controller output is given by

$$\omega(t) = \omega(0) + \int_0^t k_\mathrm{i}\big(\phi_* - \phi(\tau)\big)d\tau + k_\mathrm{p}(\phi_* - \phi(t))$$

Once $\phi(t) = \phi_*$, ω becomes constant. Hence

$$f(t) = f\,\sin\left(\int_0^t \omega(\tau)d\tau\right) = f\,\sin(\omega t + \theta(0))$$

so the control is inherently noninvasive.

How can we estimate the phase of a signal?

Let

$$x(t) = A\sin(\omega + \phi)$$

How can we determine ϕ?

If $x(t)$ is multiplied by $\sin(\omega t)$ and $\cos(\omega t)$,

$$x_{\mathrm{s}}(t) = A\sin(\omega t + \phi)\sin(\omega t) = \frac{A}{2}(\cos(\phi) - \cos(2\omega t + \phi))$$

$$x_{\mathrm{c}}(t) = A\sin(\omega t + \phi)\cos(\omega t) = \frac{A}{2}(\sin(\phi) + \sin(2\omega t + \phi))$$

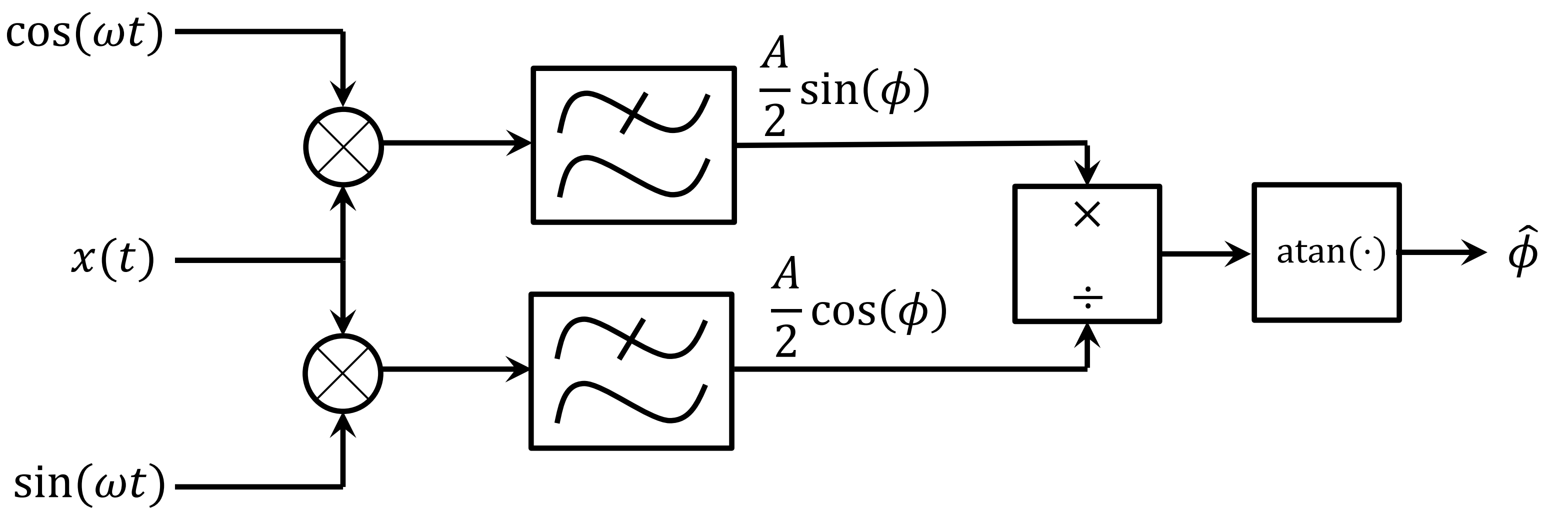
If we use a low-pass filter with cut-off frequency much smaller than 2ω, we get

$$\hat{x}_{\mathrm{s}}(t) \approx \frac{A}{2}\cos(\phi)$$

$$\hat{x}_{\mathrm{c}}(t) \approx \frac{A}{2}\sin(\phi)$$

Synchronous demodulation

The phase can be estimated with this trick, resulting in a method called synchronous demodulation.

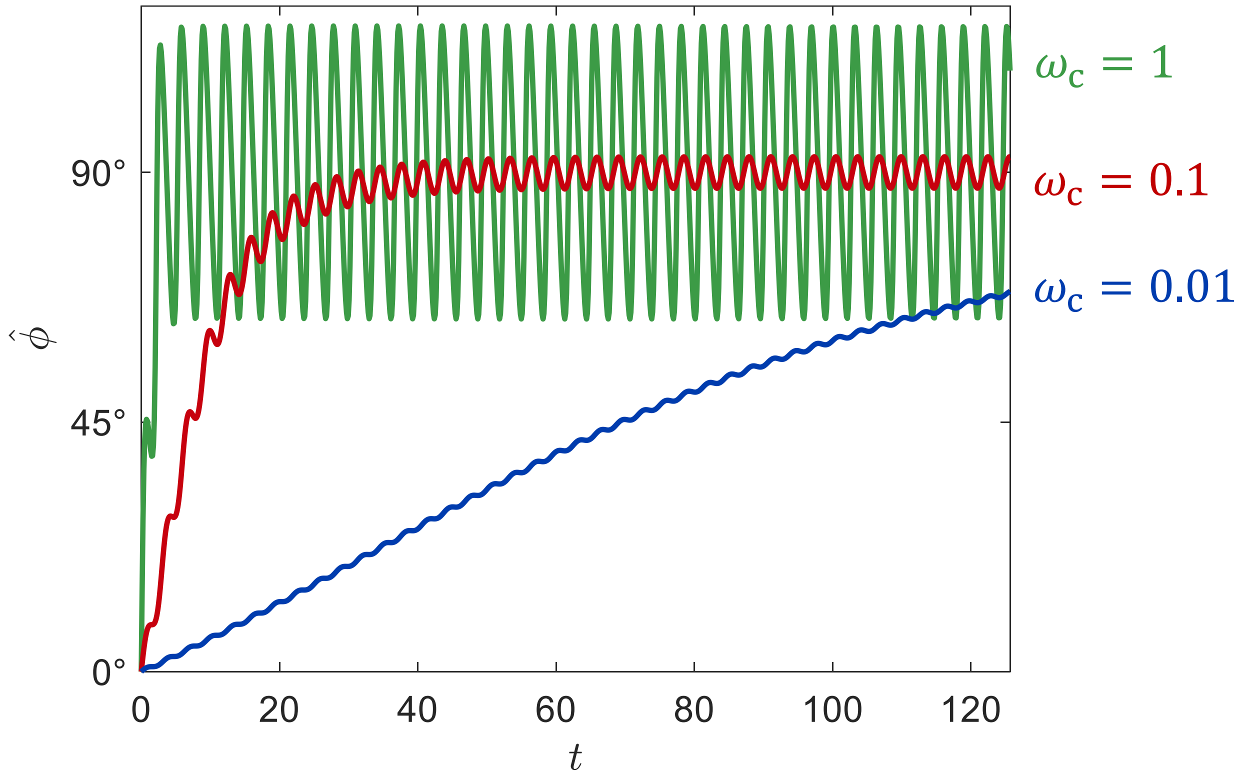


The phase of $x(t)$ can thus be estimated with simple operations on that signal.

This also works if $x(t)$ is multiharmonic.

Synchronous demodulation: tuning

$$x(t) = \cos(t), \qquad \hat{x}_\mathrm{s}(0) = \frac{1}{2}, \qquad \hat{y}_\mathrm{c}(0) = 0$$



The only parameter to tune is the filter cut-off frequency ω_c. Small cut-off frequencies lead to better rejection of the harmonics in the estimation of ϕ, but slow down its estimation.

Fourier decomposition to get the phase

The phase can also be determined from a truncated Fourier decomposition

$$x(t) \approx \mathbf{q}^T(\omega t)\mathbf{w}$$

where $\mathbf{q}$ is a vector containing harmonic functions

$$\mathbf{q}^T(\omega t) = [1 \quad \sin(\omega t) \quad \cos(\omega t) \quad \cdots \quad \sin(h\omega t) \quad \cos(h\omega t)]$$

and w is a vector containing the associated Fourier coefficients

$$\mathbf{w}^T = [w_0 \quad w_{\mathrm{s},1} \quad w_{\mathrm{c},1} \quad \cdots \quad w_{\mathrm{s},h} \quad w_{\mathrm{c},h}].$$

Indeed,

$$x(t) \approx w_0 + \sum_{n=1}^{h} \big(w_{\mathrm{s},n} \sin(n\omega t) + w_{\mathrm{c},n} \cos(n\omega t)\big)$$

$$= w_0 + \sum_{n=1}^{h} a_n \sin(n\omega t + \phi_n)$$

with

$$a_n = \sqrt{w_{\mathrm{s},n}^2 + w_{\mathrm{c},n}^2}, \qquad \phi_n = \operatorname{atan}\left(\frac{w_{\mathrm{c},n}}{w_{\mathrm{s},n}}\right)$$

Adaptive filters for online Fourier decomposition

We now wish to find $\mathbf{w} \coloneqq \mathbf{w}(t)$ such that

$$x(t) \approx \mathbf{q}^T(\omega t)\mathbf{w}(t)$$

The estimation of these coefficients can be performed online with adaptive filters.

The instantaneous squared error made by $\mathbf{q}^T(\omega t)\mathbf{w}(t)$ on $x(t)$ is

$$e^2(\mathbf{w}; t) = \left(x(t) - \mathbf{q}^T(\omega t)\mathbf{w}(t)\right)^2$$

and its gradient with respect to $\mathbf{w}$ is given by

$$\frac{\partial e^2}{\partial \mathbf{w}} = -2\left(x(t) - \mathbf{q}^T(\omega t)\mathbf{w}(t)\right)\mathbf{q}(\omega t).$$

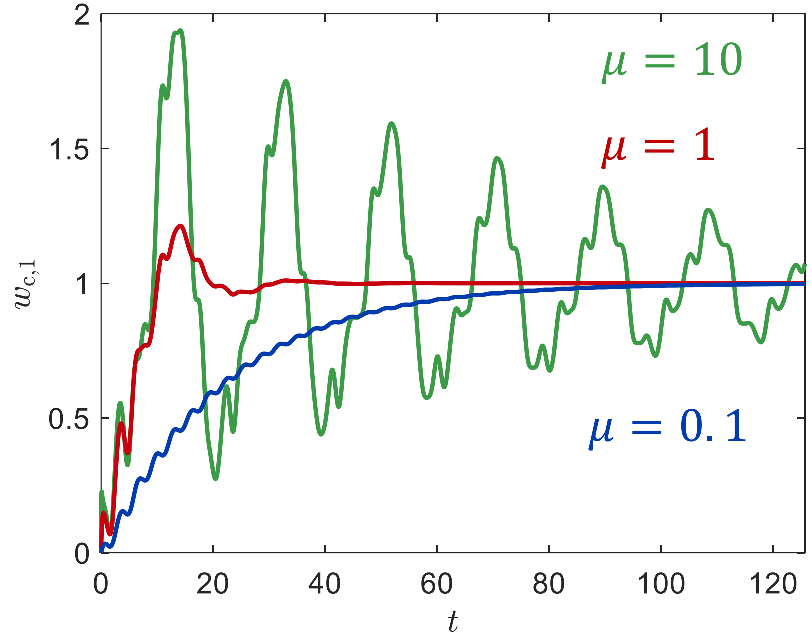
The vector $\mathbf{w}$ can be adapted with a continuous gradient descent algorithm. The adaptive filter is governed by the ODE

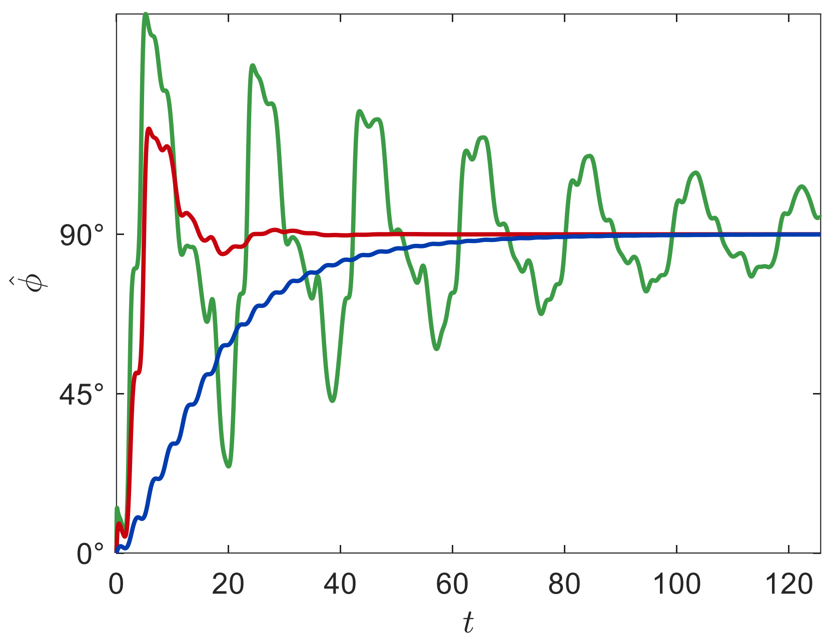
$$\dot{\mathbf{w}}(t) = \mu\left(x(t) - \mathbf{q}^T(\omega t)\mathbf{w}(t)\right)\mathbf{q}(\omega t)$$

where μ is the filter gain.

Adaptive filters: tuning

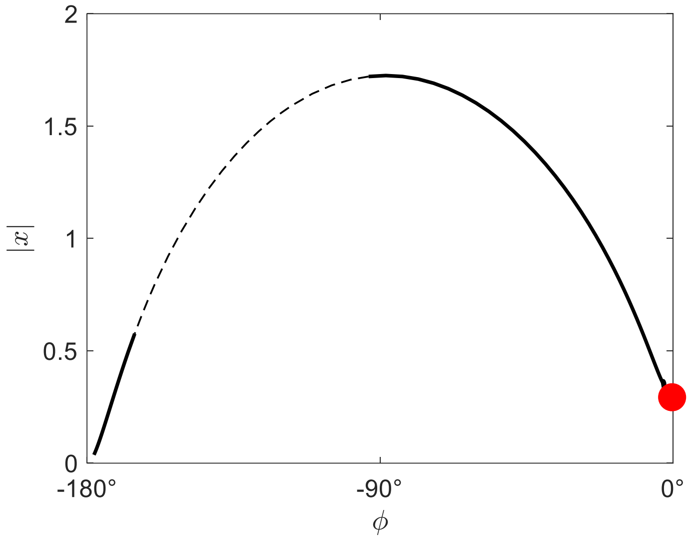
$$x(t) = \cos(t), \qquad \mathbf{w}^T(0) = [0 \quad 1 \quad 0 \quad \ldots \quad 0 \quad 0]$$

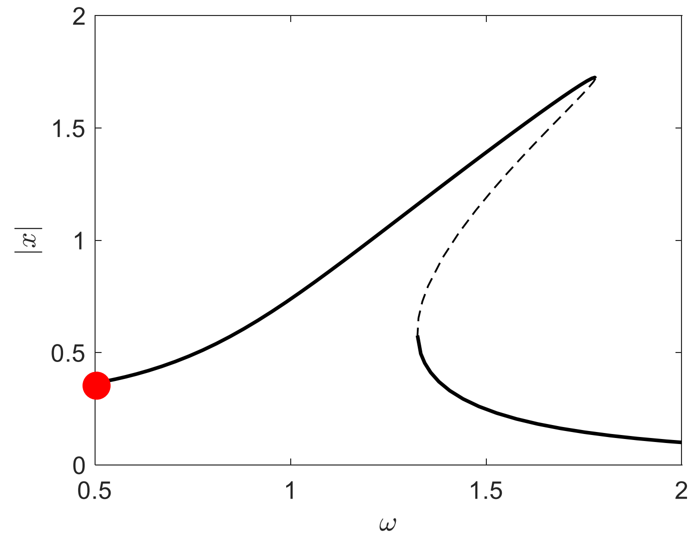


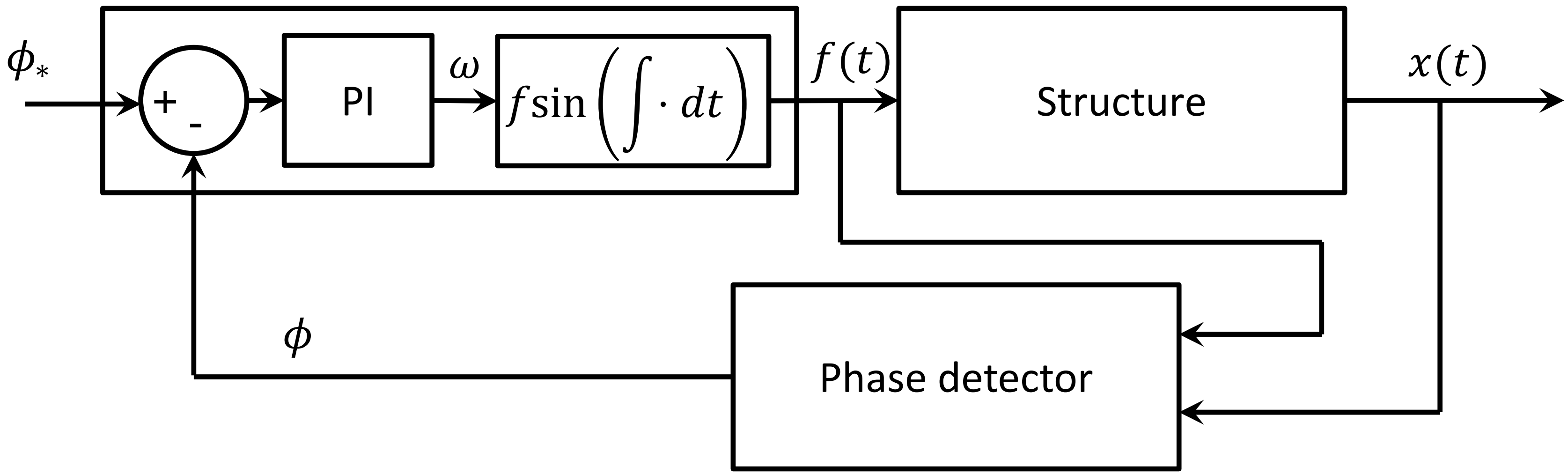


The only parameters to tune are the filter gain μ and number of harmonics h. Higher gains lead to a faster estimation but eventually result in oscillations. The number of harmonics can be tailored to the situation (strong or nonsmooth nonlinearities require a high number of harmonics).

Measuring NFRs with a PLL

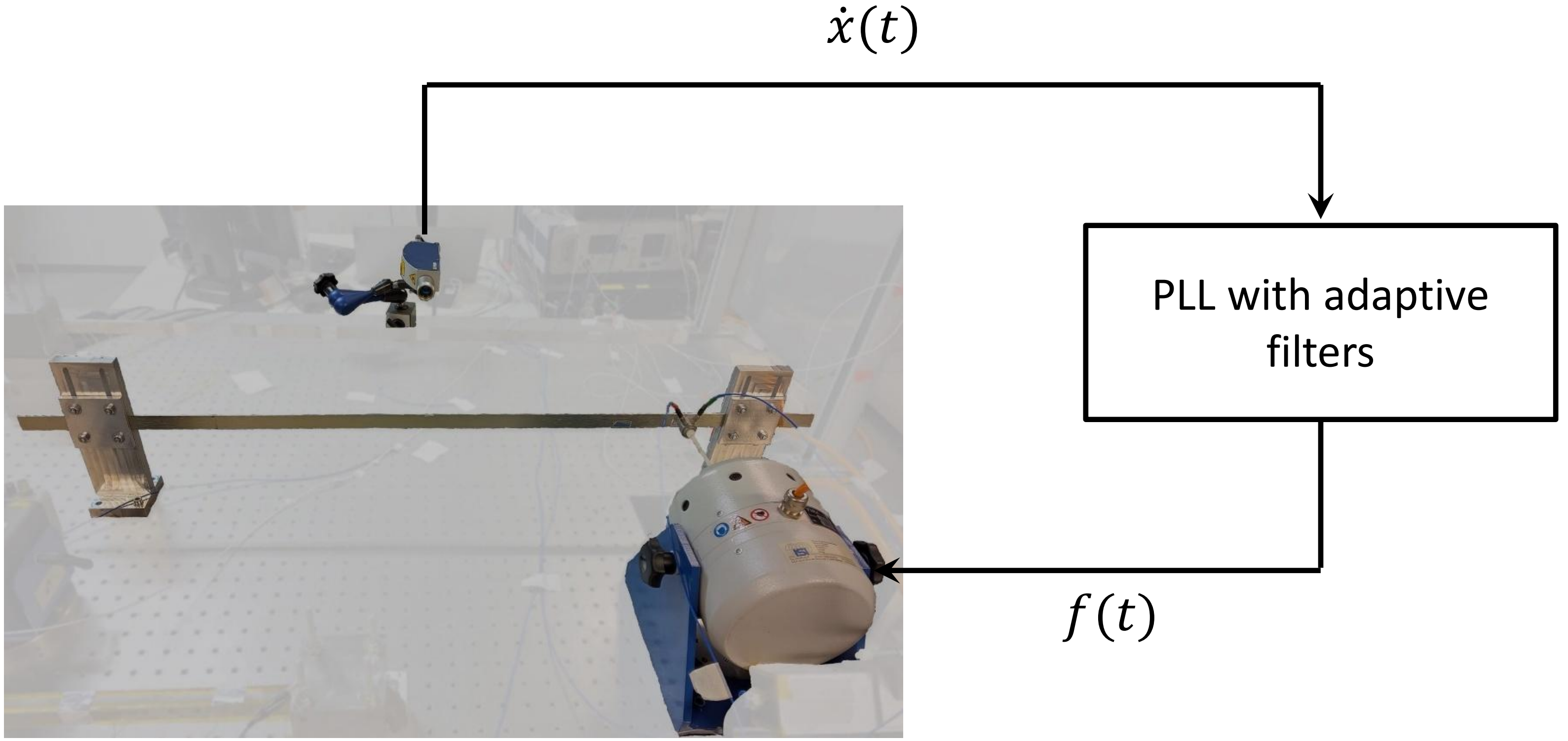






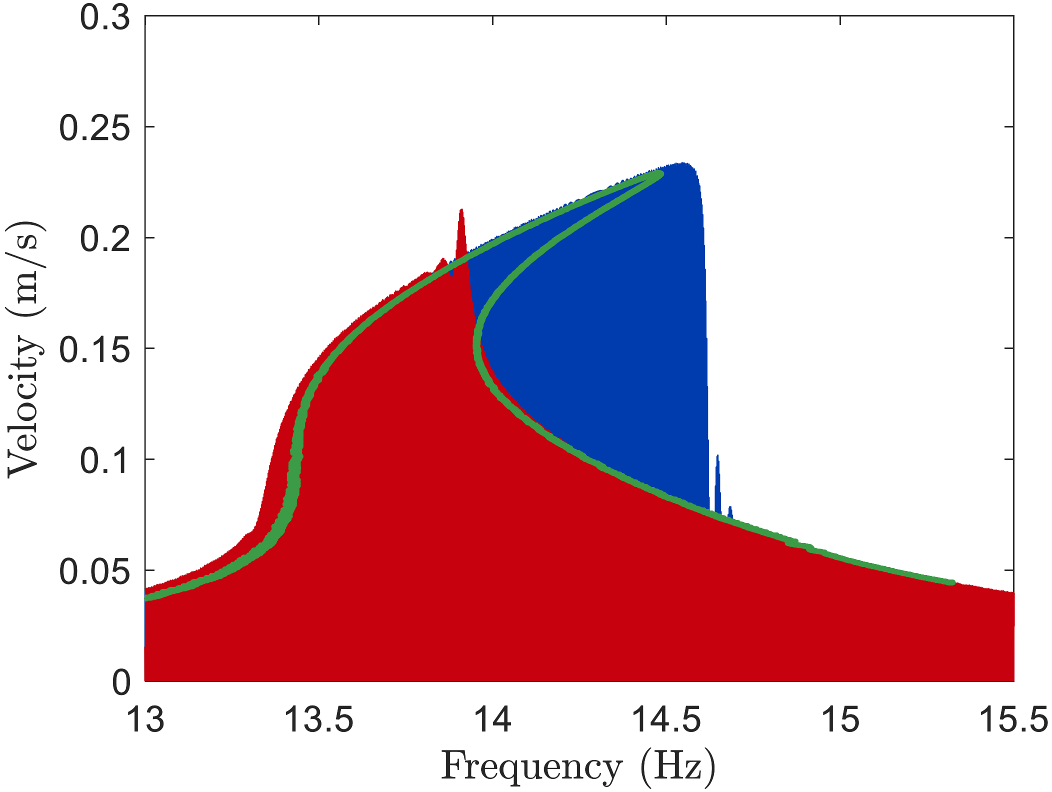
If our goal is to measure an NFR, we can vary ϕ_* between 0 and -180°.

Does it work with the thin clamped-clamped beam?



(NB: the PLL was implemented with a real-time controller – MicroLabBox from dSPACE)

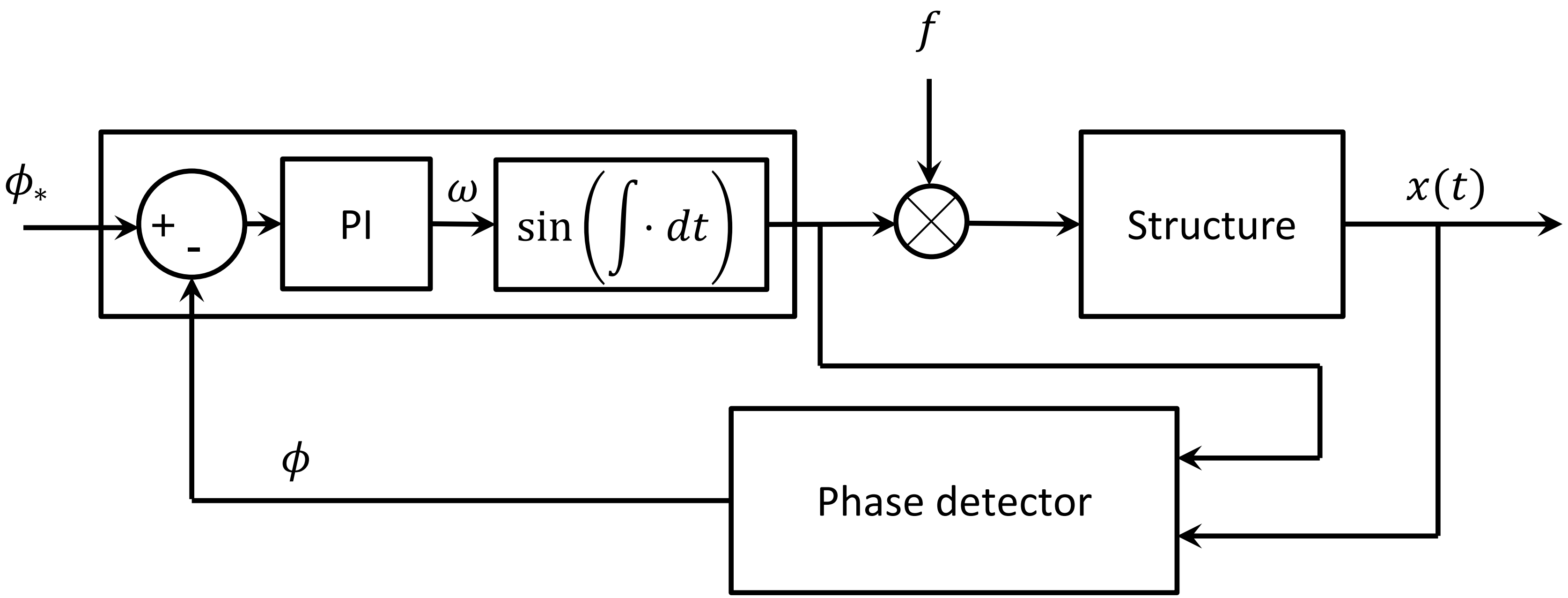
Measuring the beam NFR with a PLL



At $0.05\ V$, a sweep up leads to a jump down, and a sweep down leads to a jump up.

The PLL on the other hand is able to go through unstable branches without issues.

Resonance tracking: phase resonance backbone curve

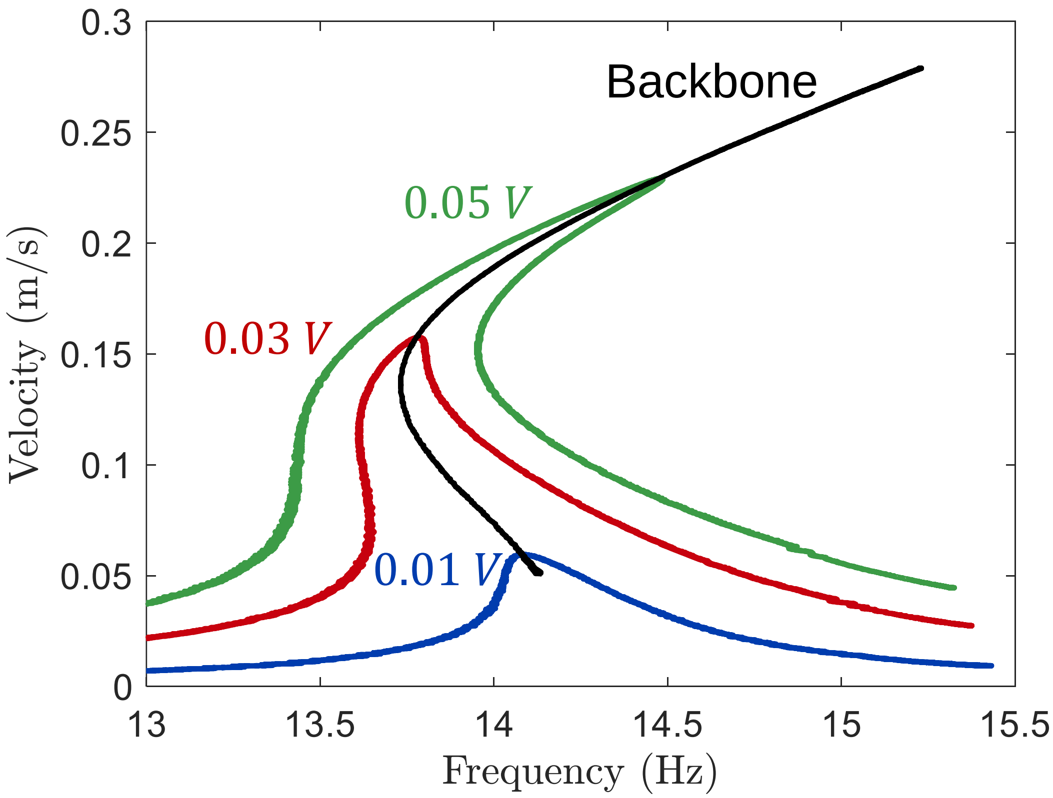


Resonances usually occur in the vicinity of $\phi = -90°$ (phase resonance).

We can track how this resonance evolves as f varies directly with a PLL. The resulting curve is usually called a (phase resonance) backbone curve.

Measuring the backbone curve of the beam with a PLL

The PLL is able to measure NFRs and backbone curves alike, with minor changes in implementation.

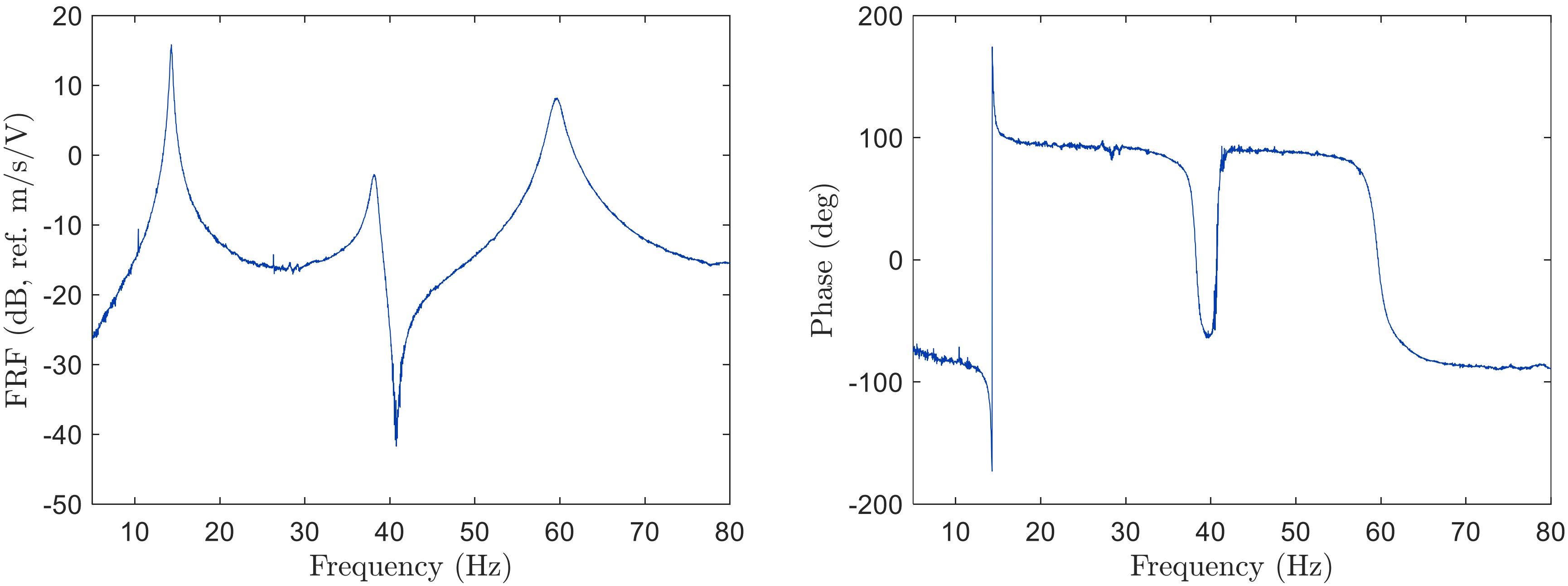


A backbone curve allows for a direct assessment of how the modal characteristics change with the forcing amplitude.

PLL – strengths and limitations

PLLs are very simple and efficient, making them a powerful tool to measure NFRs experimentally in the vicinity of a resonance.

Their main limitation comes from the need for the phase ϕ to (locally) uniquely parametrize the NFR. Unfortunately, this is not globally verified for multiple-degree-of-freedom structures (even linear ones!).



Summary

To address the challenges associated with nonlinear testing, experimental continuation combines

- feedback control to stabilize unstable branches
- continuation to trace out complicated NFRs

PLL leverages phase parametrization and PI control.

- NFRs are measured by fixing the forcing amplitude and varying the phase
- Backbone curves are measured by fixing the phase and varying the forcing amplitude

Other experimental continuation approaches exist, that rely on another type of reparametrization (SCBC, RCT) or on arclength continuation (CBC, ACBC).

References

S. Peter and R. I. Leine, “Excitation power quantities in phase resonance testing of nonlinear systems with phase-locked-loop excitation,” *Mechanical Systems and Signal Processing*, vol. 96, pp. 139–158, Nov. 2017, doi: 10.1016/j.ymssp.2017.04.011.

V. Denis, M. Jossic, C. Giraud-Audine, B. Chomette, A. Renault, and O. Thomas, “Identification of nonlinear modes using phase-locked-loop experimental continuation and normal form,” *Mechanical Systems and Signal Processing*, vol. 106, pp. 430–452, Jun. 2018, doi: 10.1016/j.ymssp.2018.01.014.

G. Abeloos *et al.*, “A consistency analysis of phase-locked-loop testing and control-based continuation for a geometrically nonlinear frictional system,” *Mechanical Systems and Signal Processing*, vol. 170, no. October 2021, p. 108820, May 2022, doi: 10.1016/j.ymssp.2022.108820.

G. Raze, G. Abeloos, and G. Kerschen, “Experimental continuation in nonlinear dynamics: recent advances and future challenges,” *Nonlinear Dynamics*, vol. 113, no. 6, pp. 4949–4997, Mar. 2025, doi: 10.1007/s11071-024-10543-9.