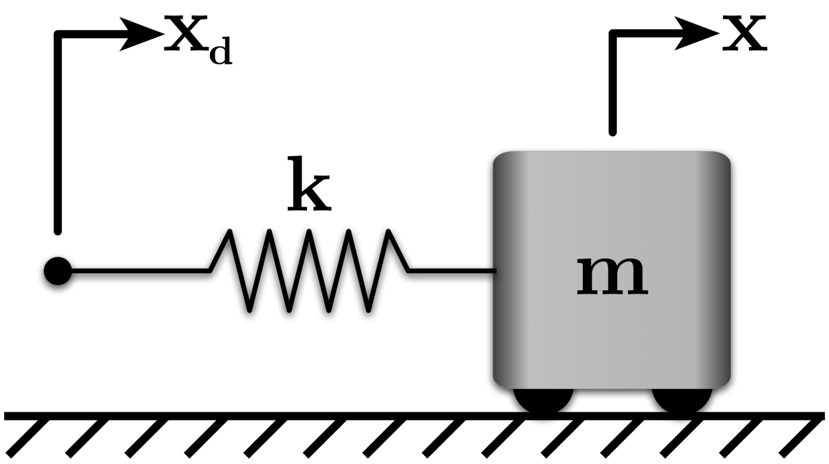
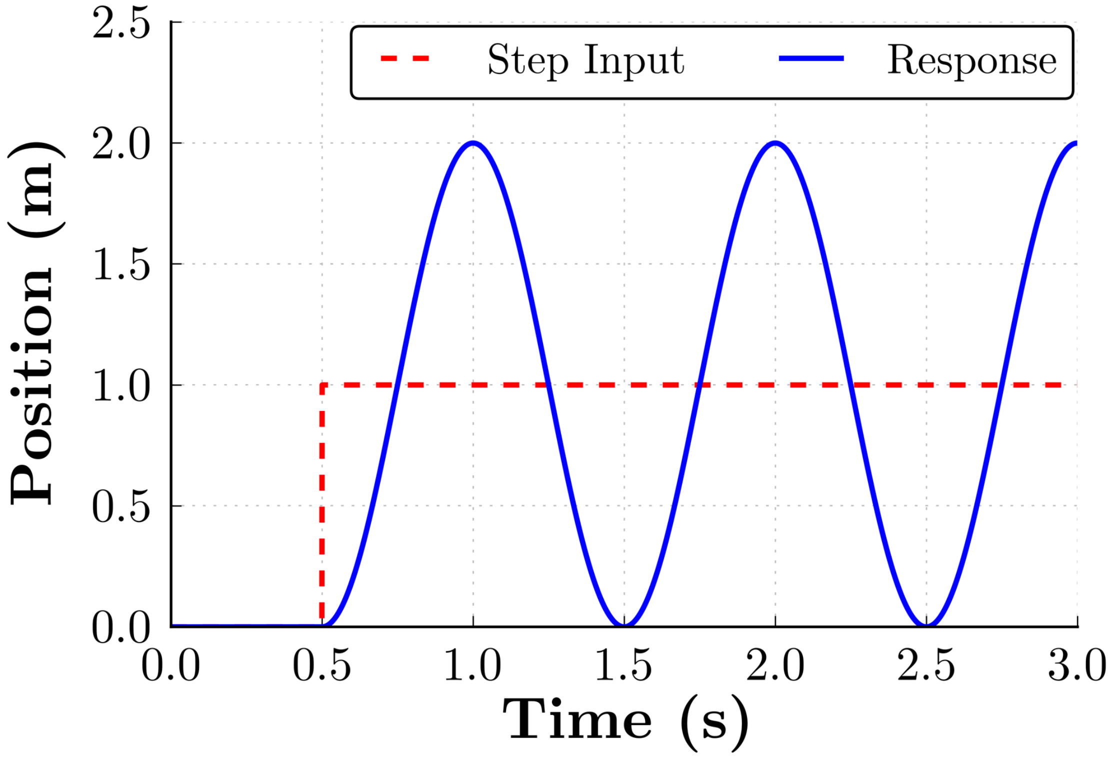
Command Generation for Flexible Systems



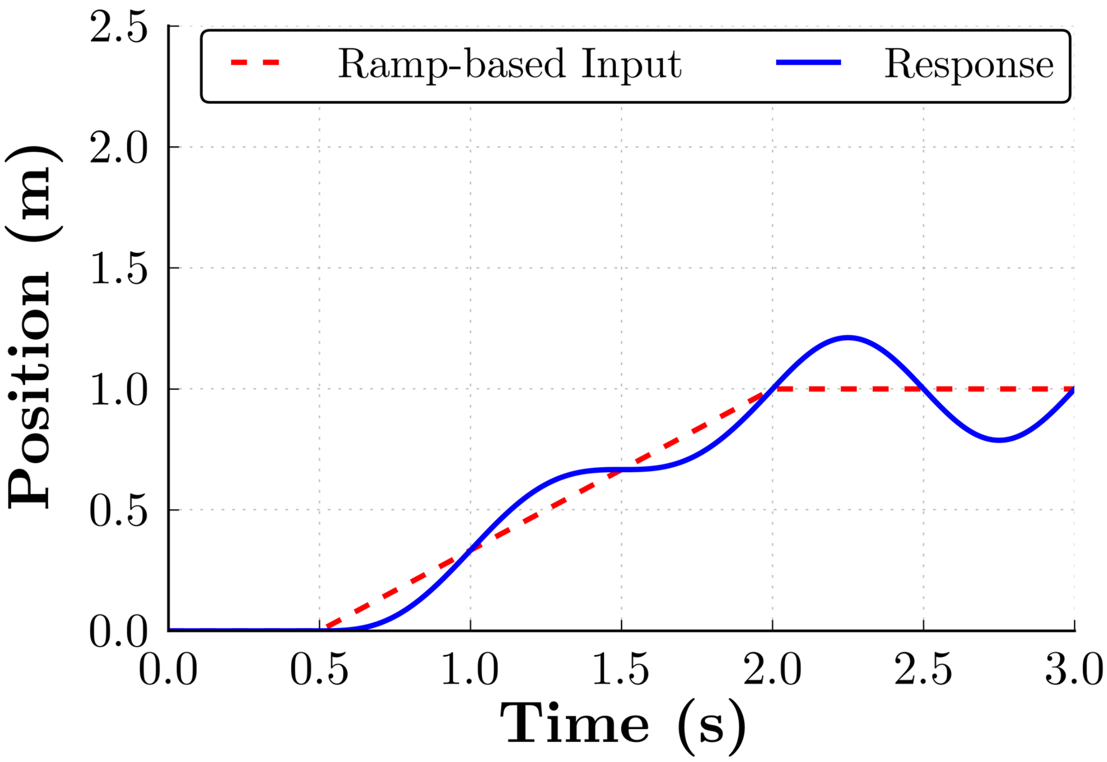
Q: What should $x_d(t)$ be to move x to a desired location as quickly as possible?

If we don't care (or know) about vibration, we might propose a step-input.



But, we do care about vibration.

Q: How can we move with low vibration? — move slow?



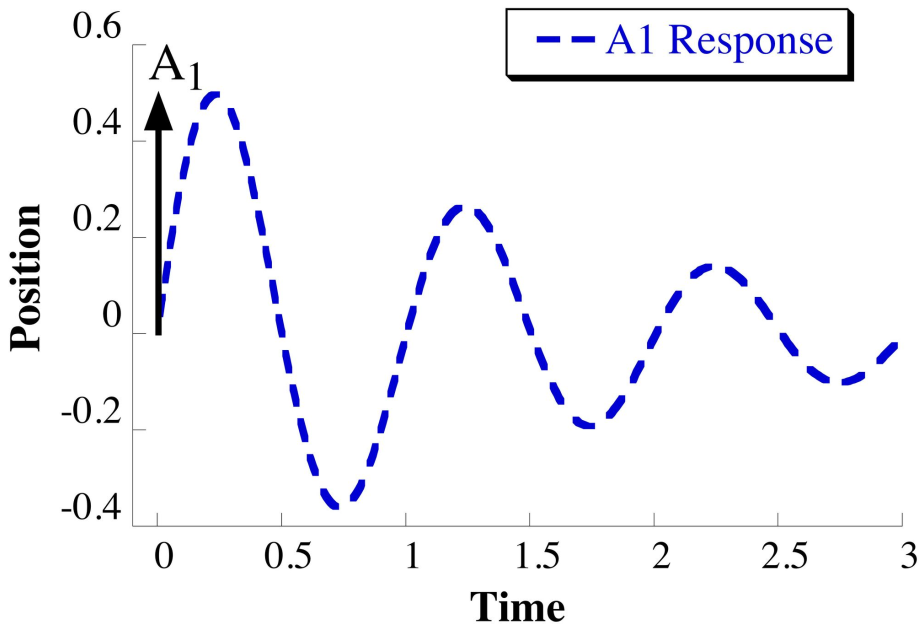
Try a ramp-based input.

Vibration is lower, but

1) maybe not low enough
2) it's slow.

Input Shaping

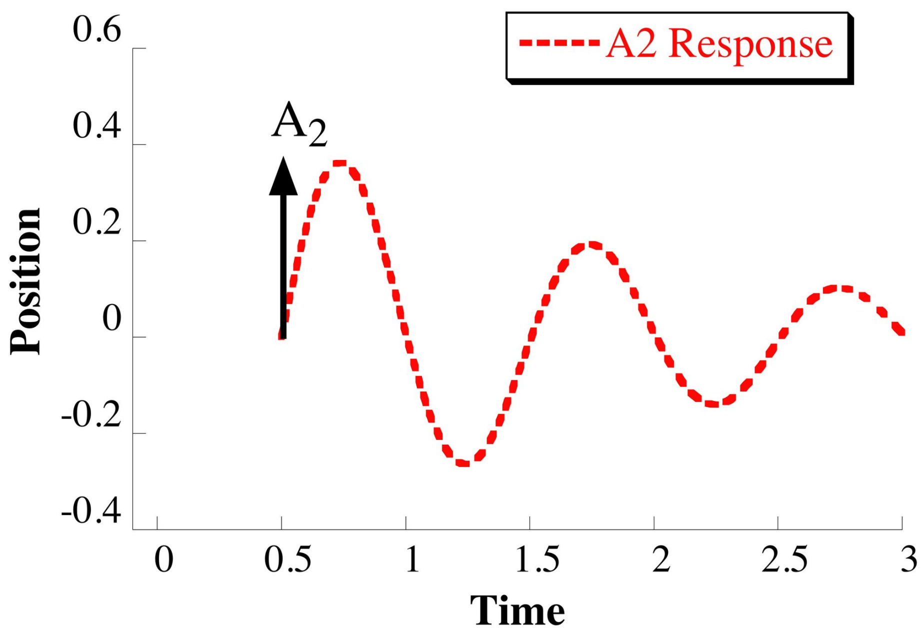
Let's Look at Impulse Responses



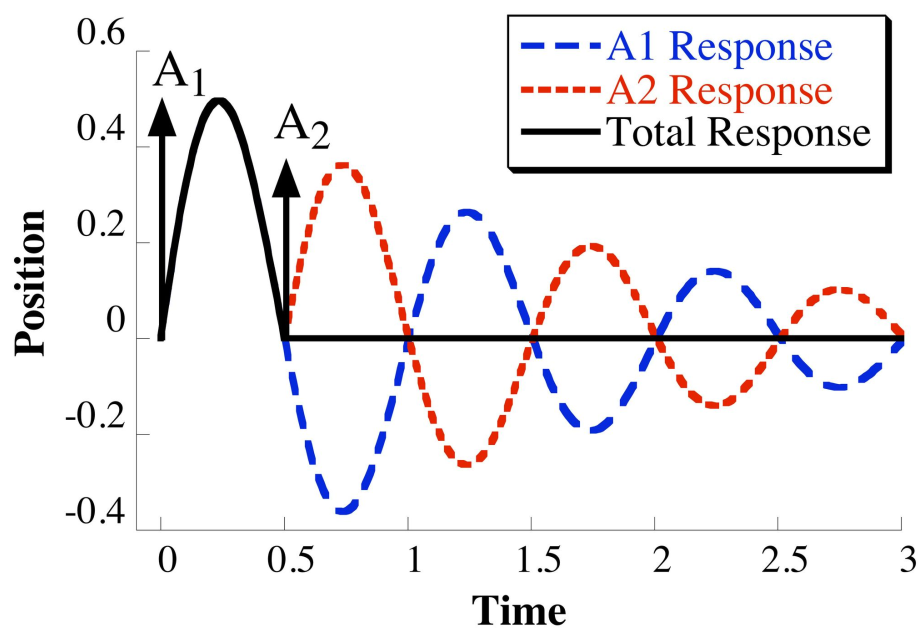
Q: If we issue an impulse to a flexible system, what happens?

- It vibrates at its natural freq.

Q: If we issue a different impulse some time later, what happens? ← It vibrates at its natural freq.



Q: If we issue 2 impulses to the same system, what happens? (assuming its linear) Superposition - the responses add



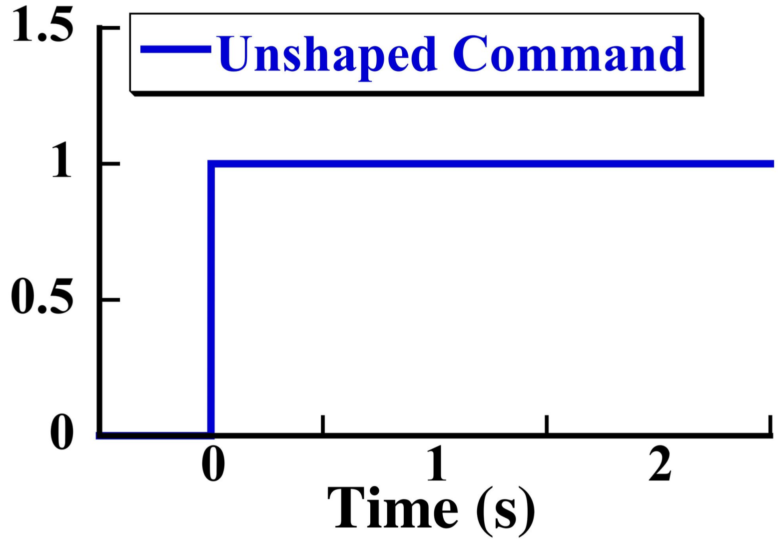
☆ If we choose the amplitudes and times correctly, we can create a zero-vibration total response!!! ☆

Q: What's the problem with this?

- linearity is necessary ← near linear is okay in practice
- We can't drive systems with impulses

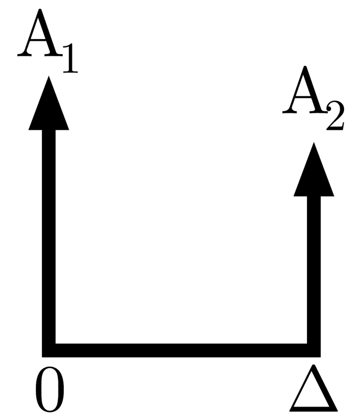
Convolution of Impulses

In rough summary, the properties of an impulse sequence remain when it is convolved with a reference command.



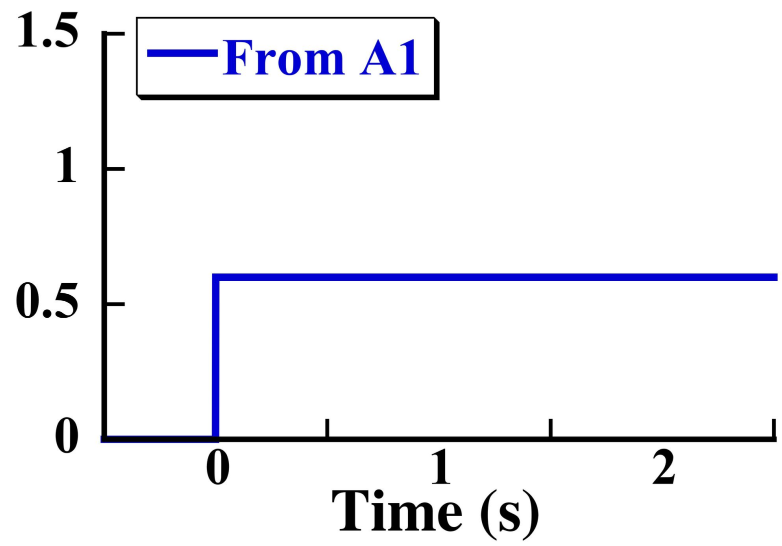
$*$

Convolution operator



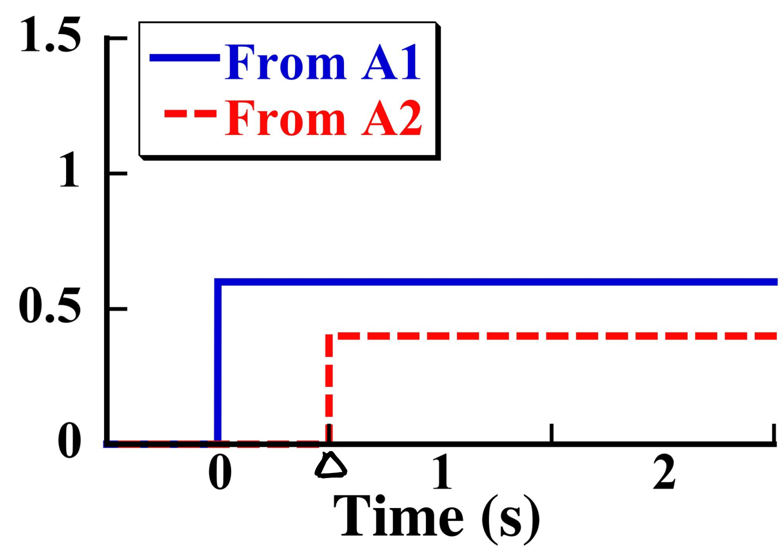
To convolve an impulse sequence with a reference command:

1) multiply original reference command by A_1 and apply this at t_1 ($t_1 = 0$ in this example)

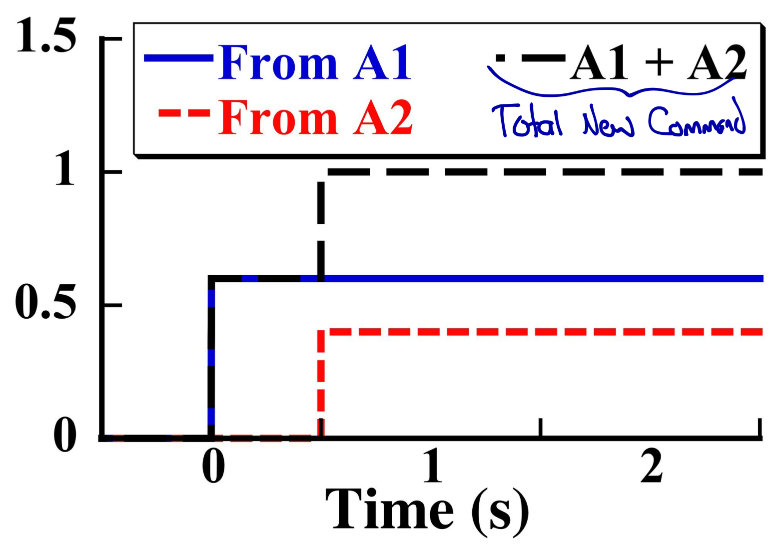


2) Multiply the original command by A_2 and apply this beginning at t_2 ($t_2 = \Delta$ in this example)

(repeat for all impulses in the sequence)



3) Sum the resulting components

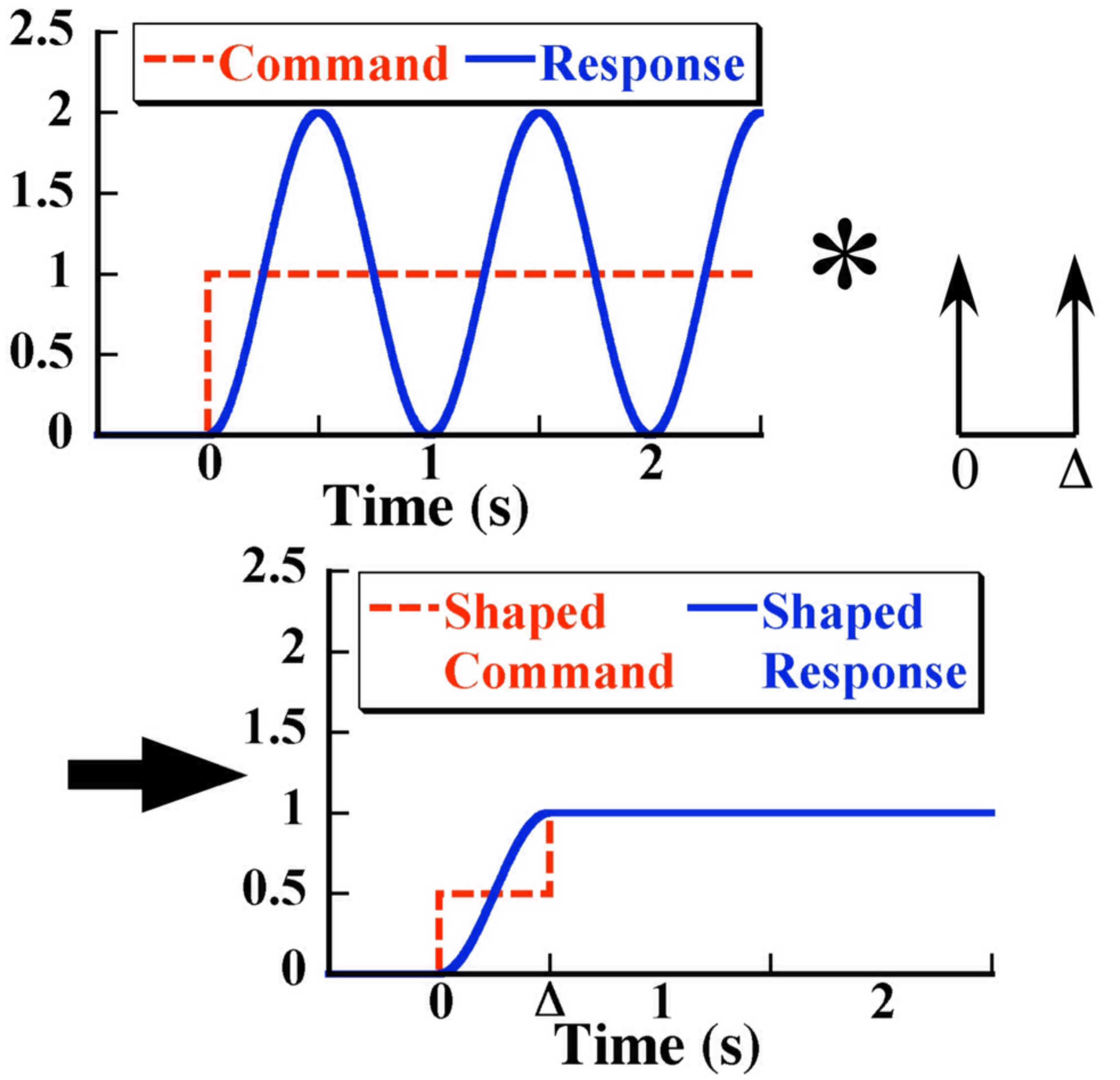


Note: The procedure is identical regardless of the "shape" of the original input.

Input Shaping

1) Determine an impulse sequence that results in low vib. (and other constraints, if desired)

2) Use the convolution of this impulse sequence and the original command to drive the system.



Q: What are the penalties for no vibration?

- command duration is increased by the duration of the impulse sequence
- Now have non-unity states (can be a problem for some systems... on/off → rockets, relays, etc.)

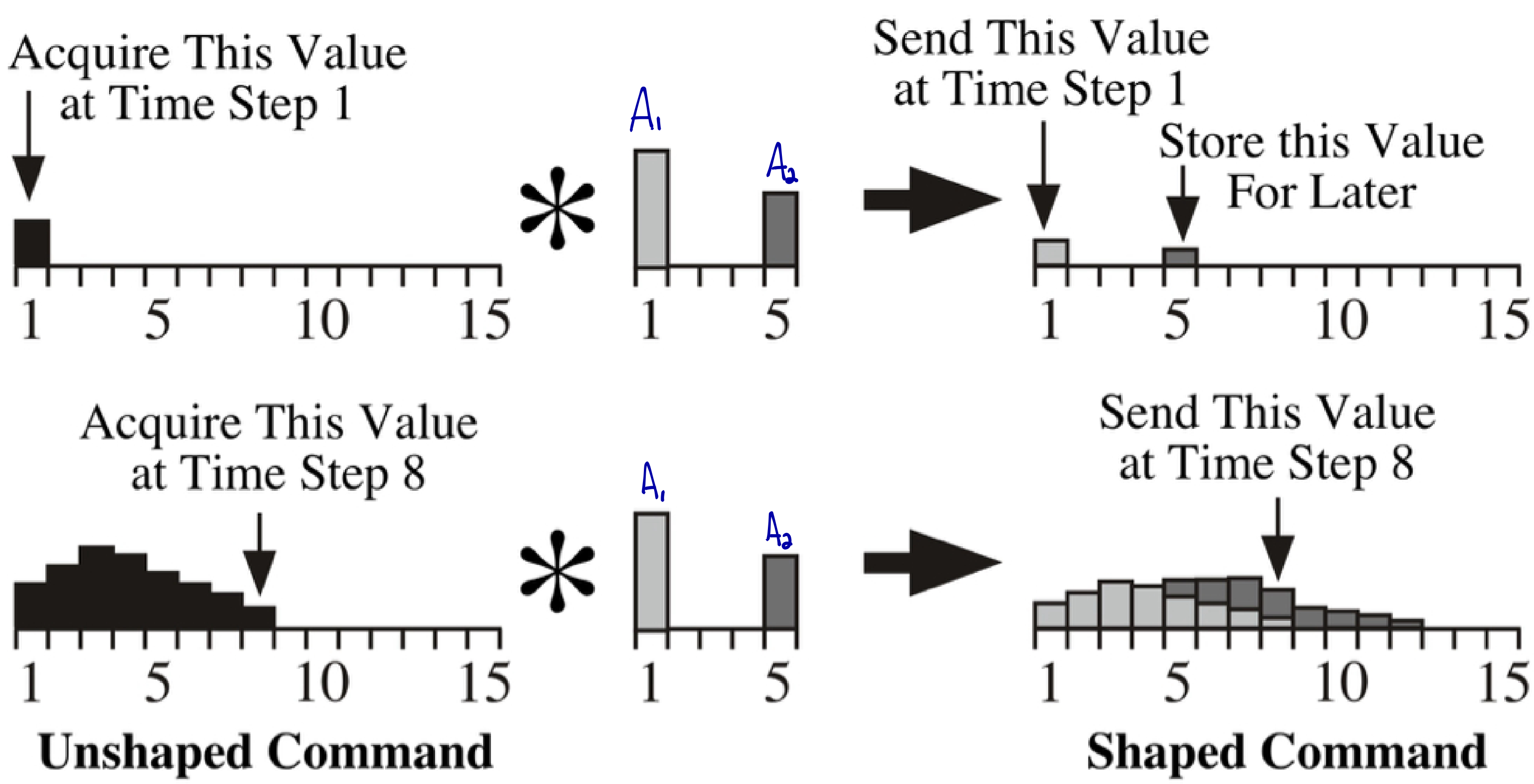
Q: What are the challenges of this approach?

- How do we select the right impulses?
- How can we implement this?
- What if our estimate of freq + damping is wrong?

Implementing Input Shaping

Simple! Just like before, but done at every time step.

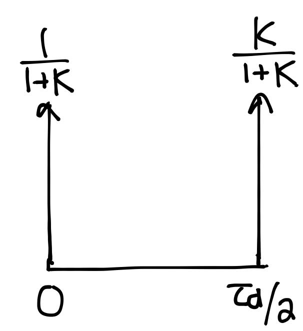
(1 mult and 1 addition per impulse ← computationally easy)



Many Input Shapers are known in closed form:

Zero-Vibration Shaper

- only positive impulses
- designed for 0 vibration at ω_n and ζ



$$\text{ZV} \equiv \begin{bmatrix} A_i \\ t_i \end{bmatrix} = \begin{bmatrix} \frac{1}{1+K} & \frac{K}{1+K} \\ 0 & \frac{\tau_d}{2} \end{bmatrix} \qquad K = e^{\frac{-\zeta\pi}{\sqrt{1-\zeta^2}}}$$

Zero-Vibration and Derivative (ZVD) Shaper

- only positive impulses
- zero vibration with robustness at ω_n and ζ

$$\text{ZVD} \equiv \begin{bmatrix} A_i \\ t_i \end{bmatrix} = \begin{bmatrix} \frac{1}{1+2K+K^2} & \frac{2K}{1+2K+K^2} & \frac{K^2}{1+2K+K^2} \\ 0 & \frac{\tau_d}{2} & \tau_d \end{bmatrix} \qquad K = e^{\frac{-\zeta\pi}{\sqrt{1-\zeta^2}}}$$