# Settling Control with Dual Stage Systems

**Santosh Devasia**

**Mechanical Engineering Department**

**U. Of Washington, Seattle**

# Overview of our work

**We investigated the benefits of using dual-stage systems**

**Clearly the second actuator should improve the servo performance.**

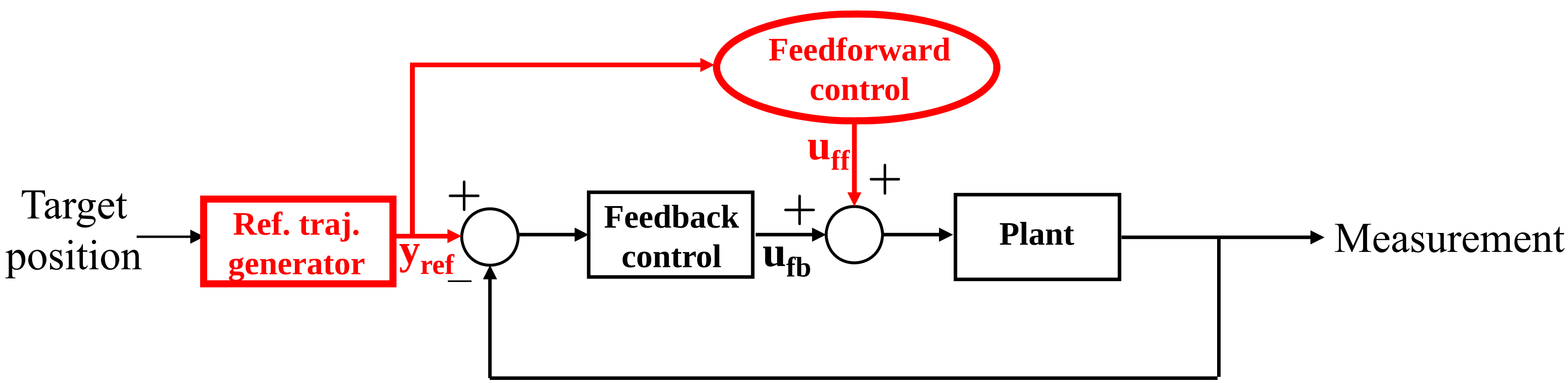
**The second actuator has**
**higher bandwidth and also higher resolution**
**both help to improve servo performance with increased data density.**

**Our research focused on two aspects**
**(a) Optimizing Seek**
**(b) Optimizing Settle**

# Our Research: Feedforward Design

**For both Seek (completing the articles in this area) and Settle (more recently)**

**Augments any existing feedback controller**

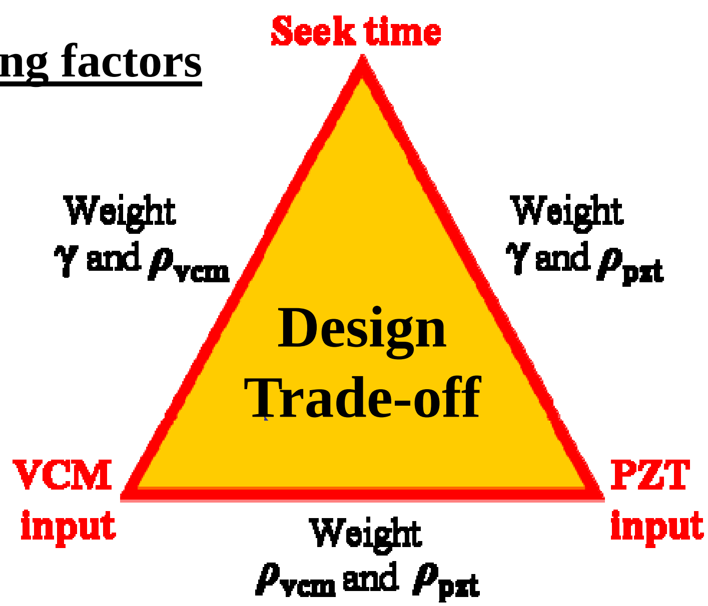


# Start with Seek control

**Main Issue: Trade-offs between VCM vs PZT vs Seek time?**

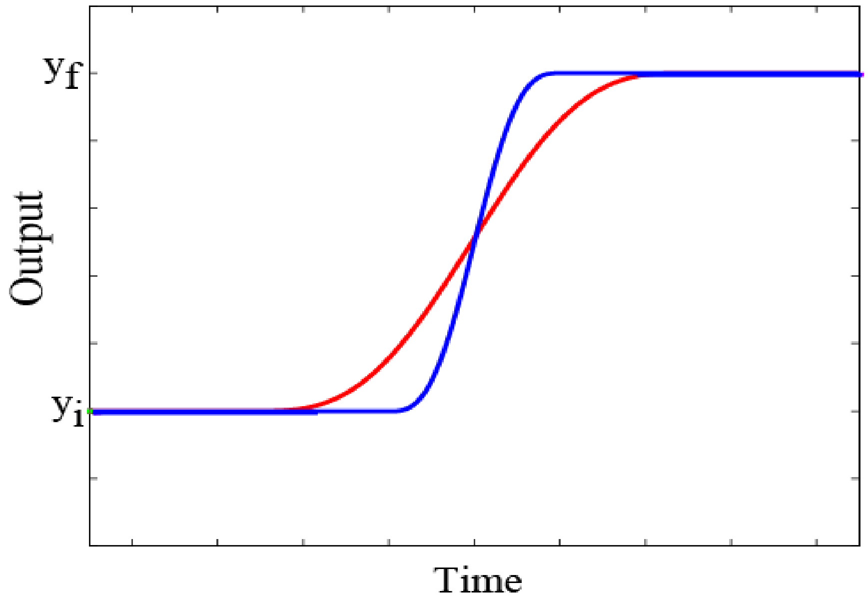
$$J = \underbrace{\gamma T}_{\substack{\text{Cost of} \\ \text{seek time}}} + \int \left\{ \underbrace{\rho_{vcm} u_{vcm}(t)^2}_{\substack{\text{Cost of} \\ \text{VCM input}}} + \underbrace{\rho_{pzt} u_{pzt}(t)^2}_{\substack{\text{Cost of} \\ \text{PZT input}}} \right\} dt$$

<u>**Weighting factors**</u>




**<u>Our contribution:</u>** Use pre/post actuation (input applied before and after seek time interval) to improve seek performance

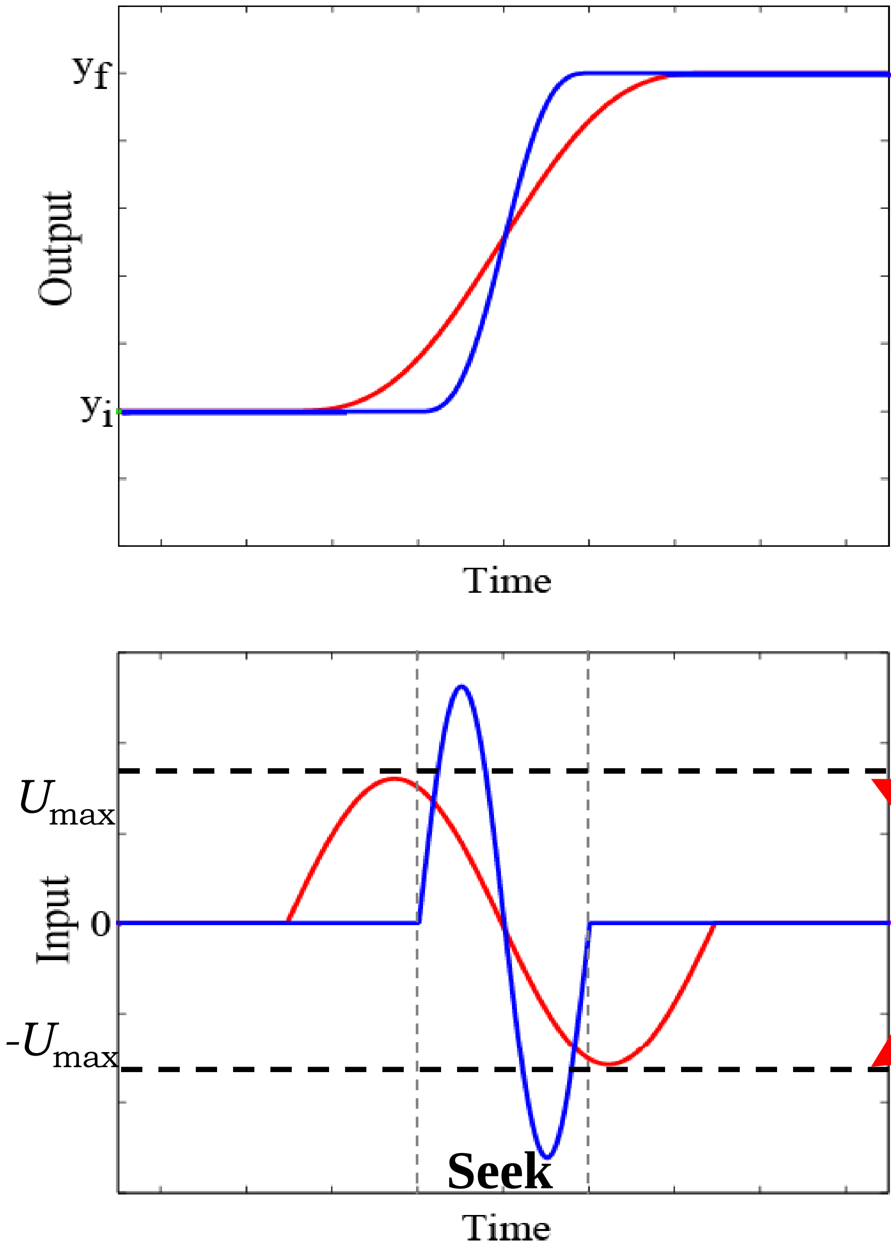
# How does pre/post-actuation help with dual-stage actuator?




Let's start with a single-stage VCM input

We'd like to design a fast seek trajectory (**blue line**)

# How does pre/post-actuation help with dual-stage actuator?

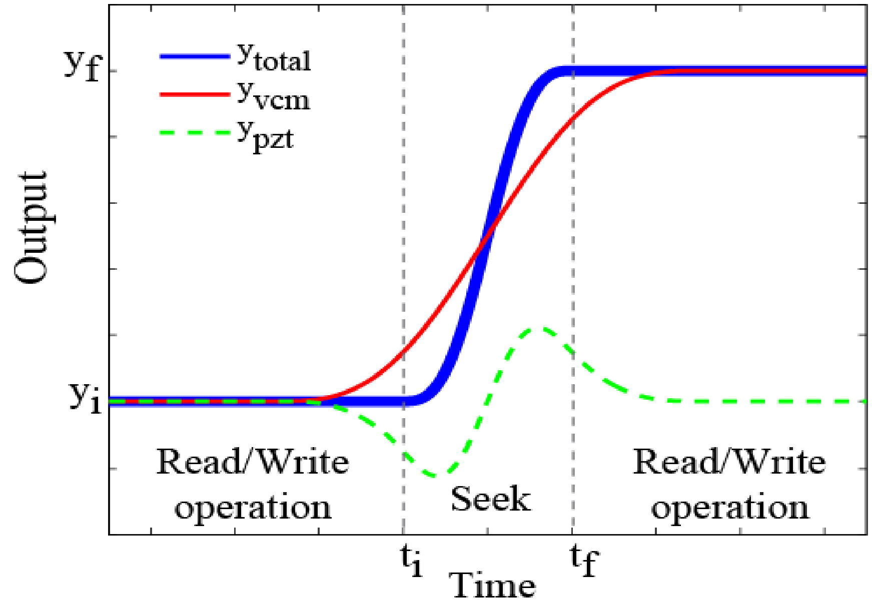



Let's start with a single-stage VCM input

We'd like to design a fast seek trajectory (**blue line**)

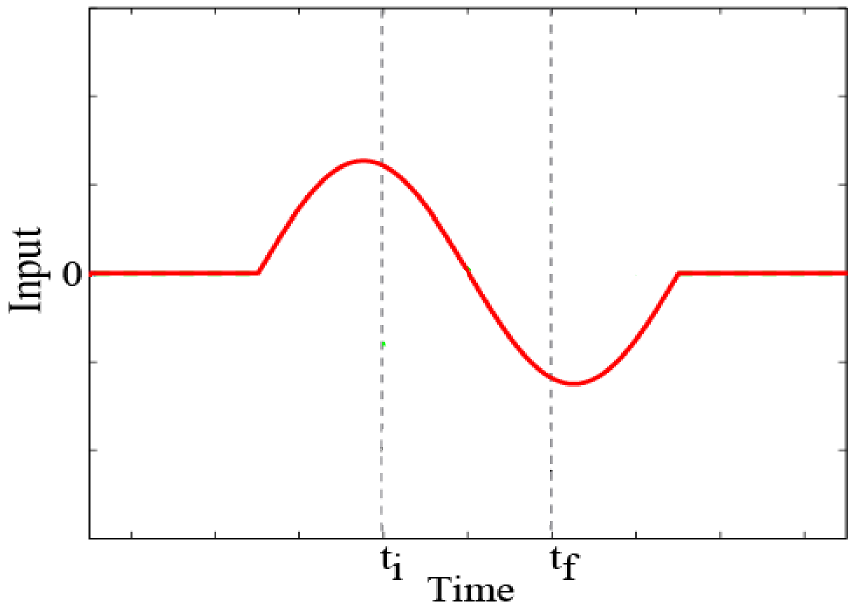
But the input is constrained by the saturation limit

# Use second actuator (PZT) to cancel the movement from VCM



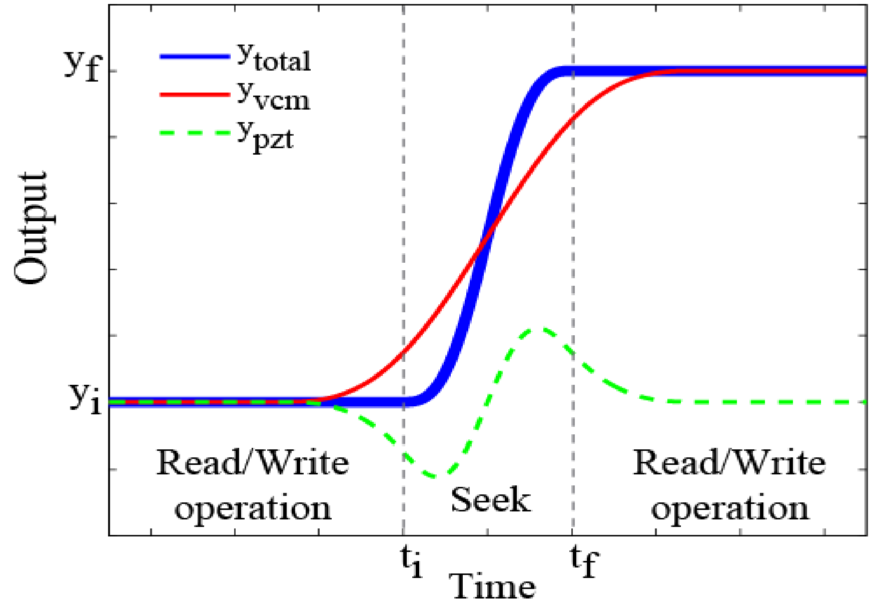

Design PZT trajectory (**green line**) to cancel the VCM motion outside the seek time

Effective time to change the output (seek time) is reduced



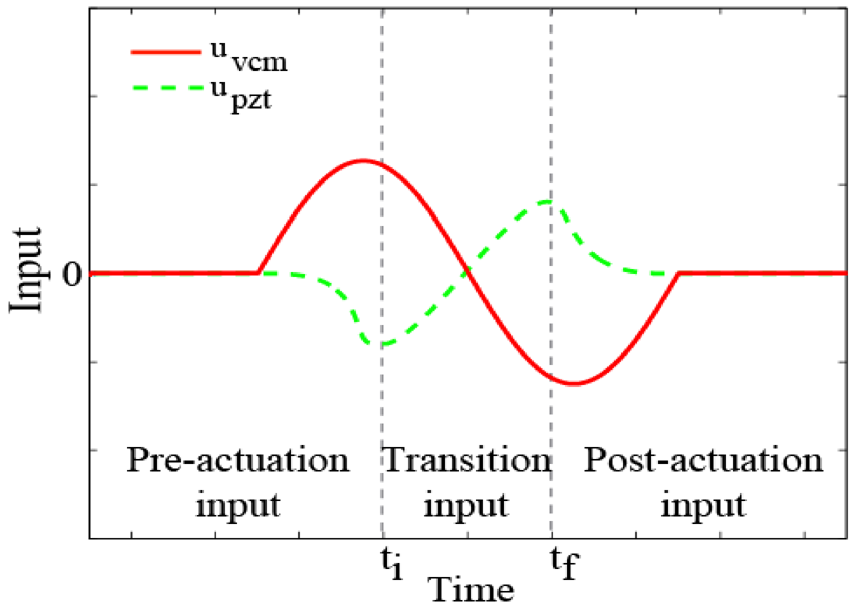

More time to apply the input without scarifying the seek time

PZT can help achieve faster seek with use of pre/post-actuation



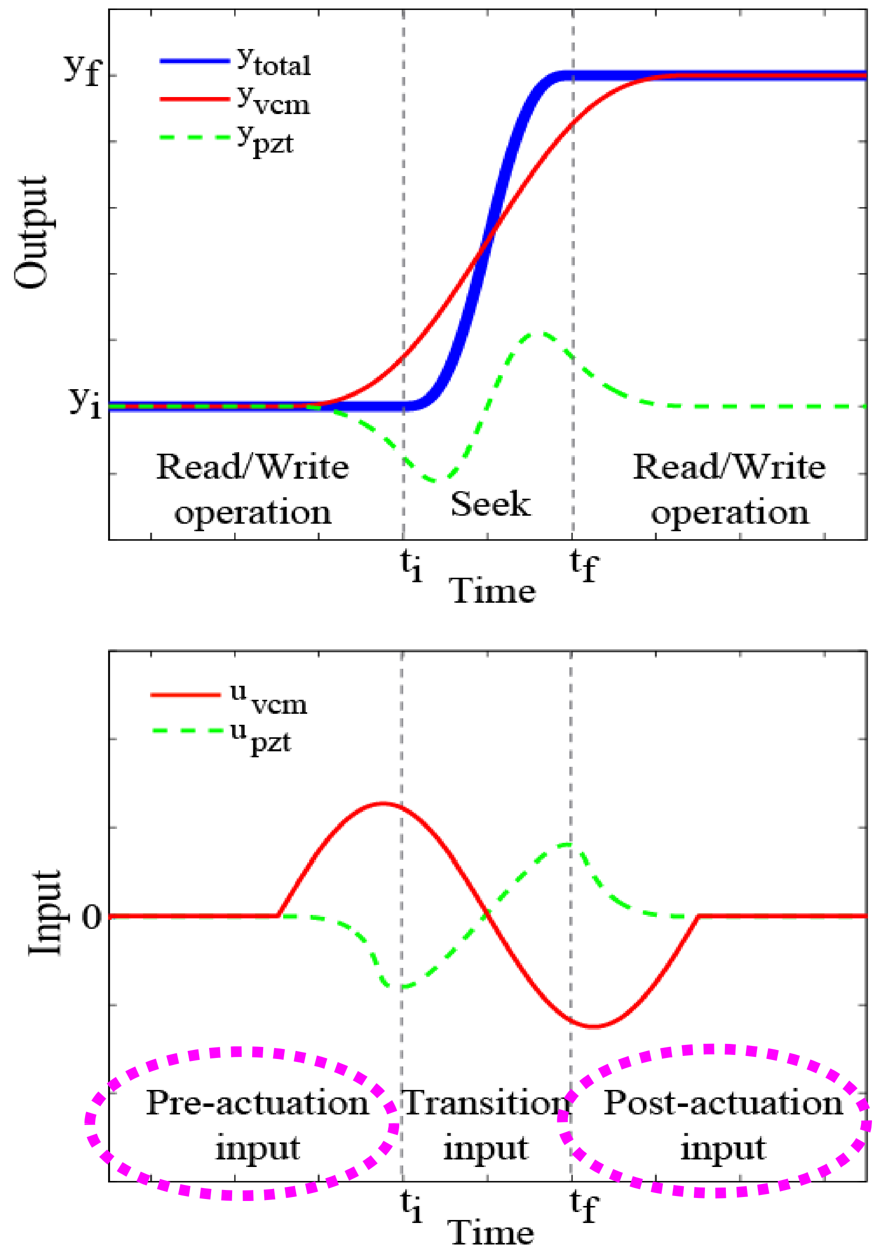

Design PZT trajectory (green line) to cancel the VCM motion outside the seek time

Effective time to change the output (seek time) is reduced




**Thus, for a fixed VCM input, the dual-stage can reduce the seek time.**

# Two main issues in pre/post-actuation:




1. Find pre- and post-actuation inputs to maintain constant output


**Publications:**

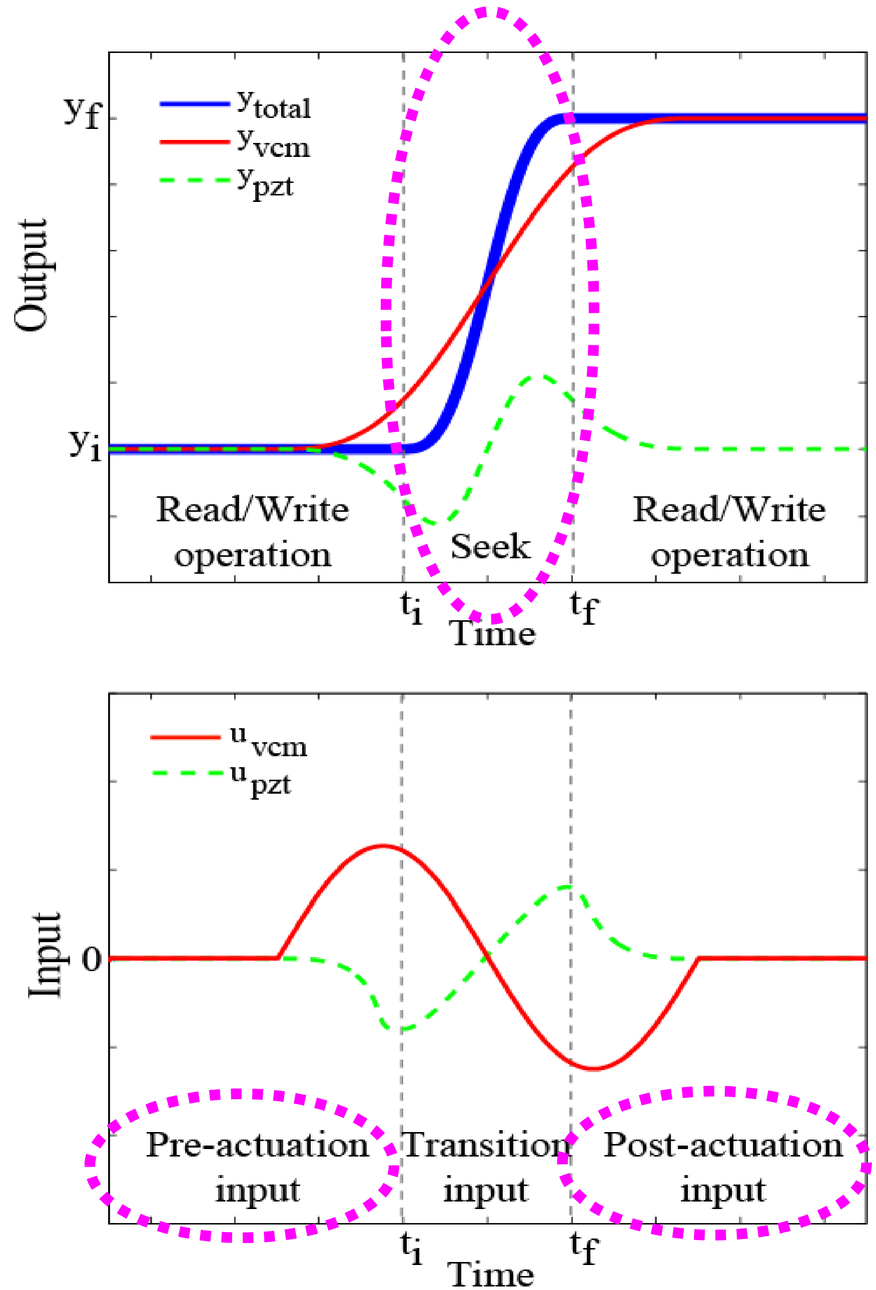
**1.) D. Iamratanakul**, B. Jordan, K. Leang, and S. Devasia "Optimal Output Transitions for Dual-Stage Systems," Revised version submitted to IEEE Trans. on Control System Technology.

**2.) D. Iamratanakul**, B. Jordan, K. Leang, and S. Devasia "Optimal Seek-Trajectory Design for Dual-Stage Systems," Proc. of the 2006 American Control Conference, Minneapolis MN.

**3.) D. Iamratanakul**, H. Perez and S. Devasia "Minimum-Energy Output- Transitions for Linear Discrete-Time Systems: Flexible Structure Applications," AIAA Journal of Guidance, Control, and Dynamics, Vol. 27(4), pp. 572-585, 2004.

**4.) D. Iamratanakul**, H. Perez and S. Devasia "Feedforward Trajectory Design for Output Transitions in Discrete-time Systems: Disk-Drive Example," Proc. of the 2003 American Control Conference, Volume: 4, Page(s): 3142-3147, 2003.

# Two main issues in pre/post-actuation:




1. Find pre- and post-actuation inputs to maintain constant output

2. What is a good seek trajectory? → Design trade off


**Publications:**

**1.)** **D. Iamratanakul**, B. Jordan, K. Leang, and S. Devasia "Optimal Output Transitions for Dual-Stage Systems," Revised version submitted to IEEE Trans. on Control System Technology.

**2.)** **D. Iamratanakul**, B. Jordan, K. Leang, and S. Devasia "Optimal Seek-Trajectory Design for Dual-Stage Systems," Proc. of the 2006 American Control Conference, Minneapolis MN.

**3.)** **D. Iamratanakul**, H. Perez and S. Devasia "Minimum-Energy Output- Transitions for Linear Discrete-Time Systems: Flexible Structure Applications," AIAA Journal of Guidance, Control, and Dynamics, Vol. 27(4), pp. 572-585, 2004.

**4.)** **D. Iamratanakul**, H. Perez and S. Devasia "Feedforward Trajectory Design for Output Transitions in Discrete-time Systems: Disk-Drive Example," Proc. of the 2003 American Control Conference, Volume: 4, Page(s): 3142-3147, 2003.

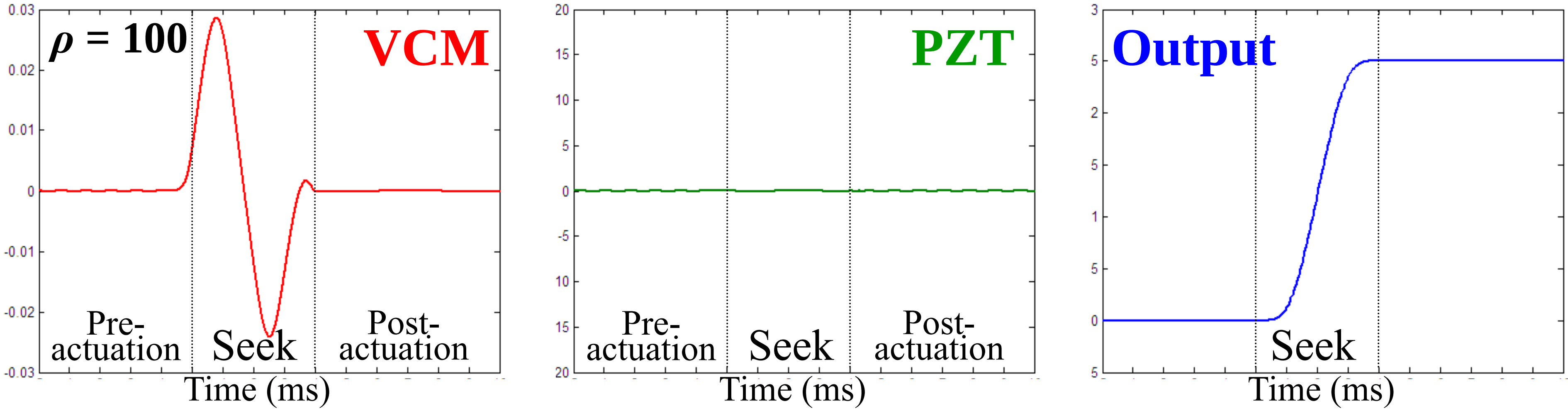
## Trade-off between VCM and PZT

$$\min_{u} J = \int \left\{ u_{vcm}(t)^2 + \rho u_{pzt}(t)^2 \right\} dt$$

## Trade-off between VCM and PZT

$$\min_{u} J = \int \left\{ u_{vcm}(t)^2 + \rho u_{pzt}(t)^2 \right\} dt$$

**Design Flexibility:** Choose large $\rho$ → single stage case with VCM input

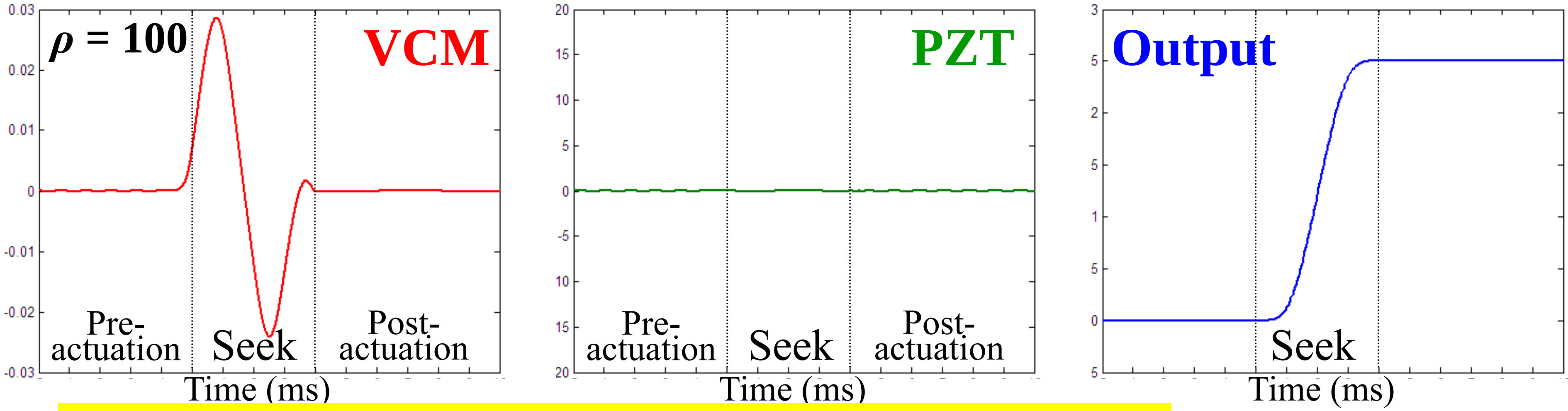



Seek length = 2.5 micron, Seek time = 4ms

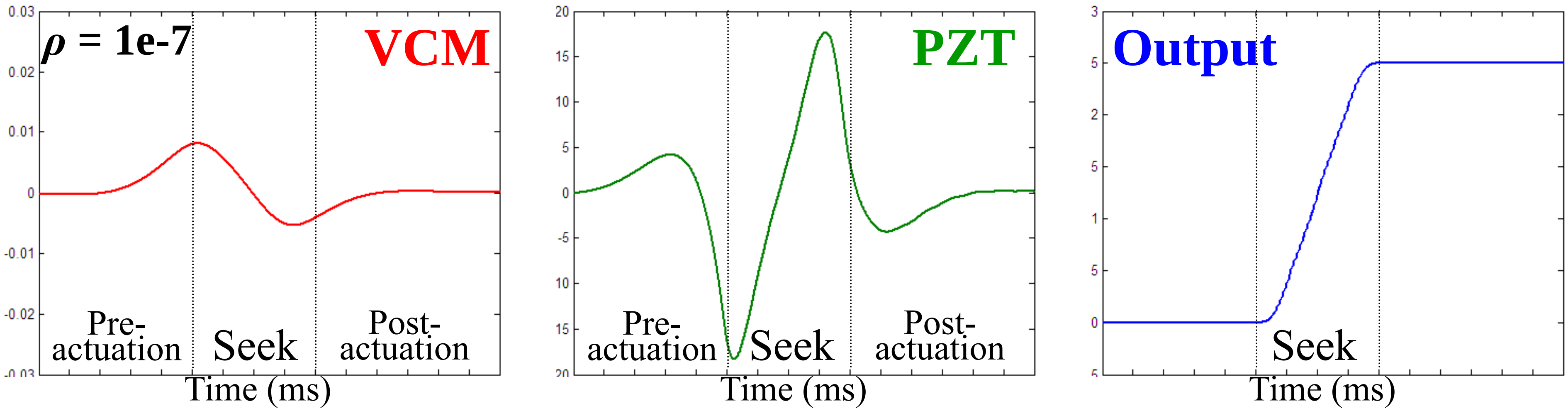
# Trade-off between VCM and PZT

$$\min_{u} J = \int \left\{ u_{vcm}(t)^2 + \rho u_{pzt}(t)^2 \right\} dt$$

**Design Flexibility:** Choose large $\rho$ → single stage case with VCM input




**Choose small $\rho$** → use PZT to augment the VCM input



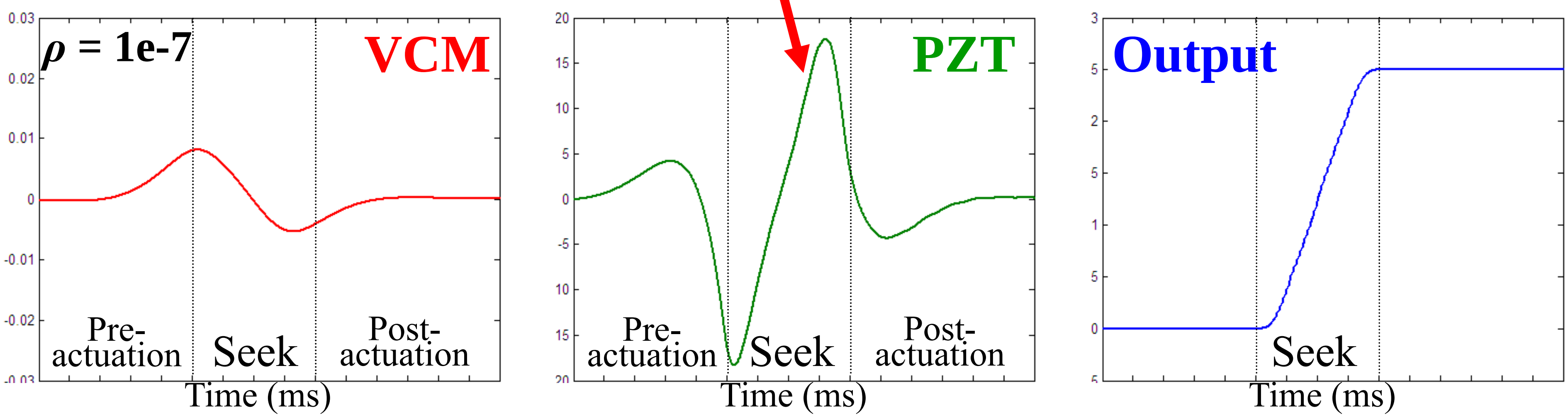
## Trade-off between VCM and PZT

$$\min_{u} J = \int \left\{ u_{vcm}(t)^2 + \rho u_{pzt}(t)^2 \right\} dt$$

What if PZT input is too high?

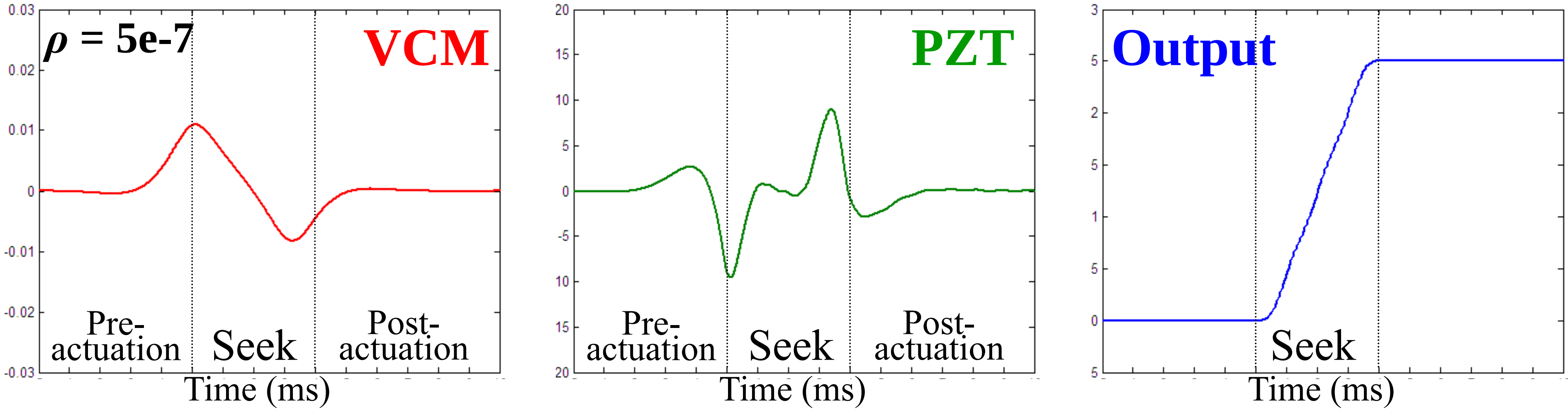
Then increase the weight on PZT

**Choose small $\rho$** → use PZT to help reduce VCM input

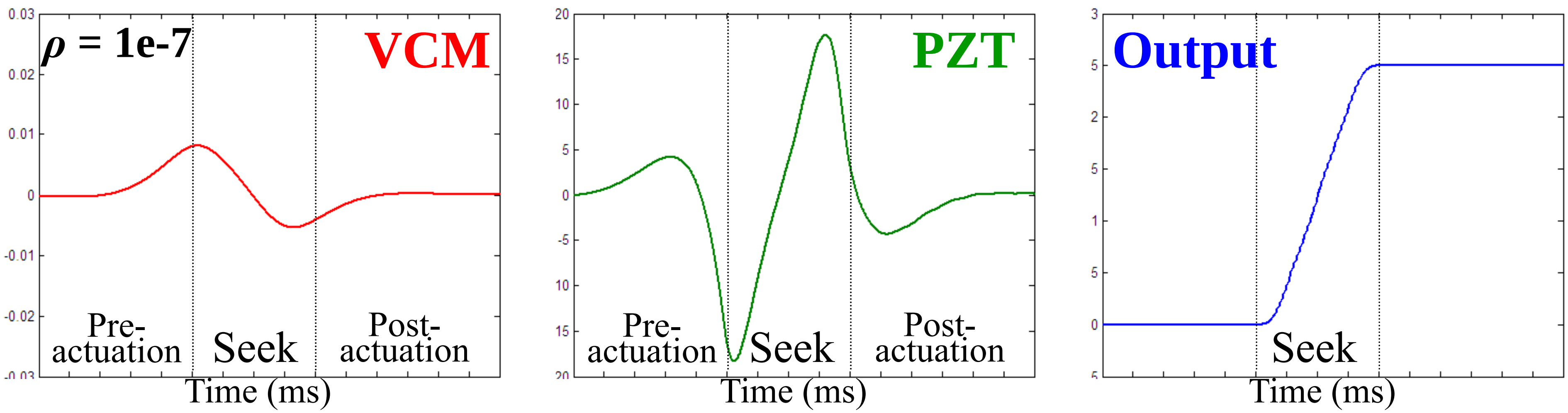


# Trade-off between VCM and PZT

PZT is now smaller -- How much improvement we can get
→ limit on PZT

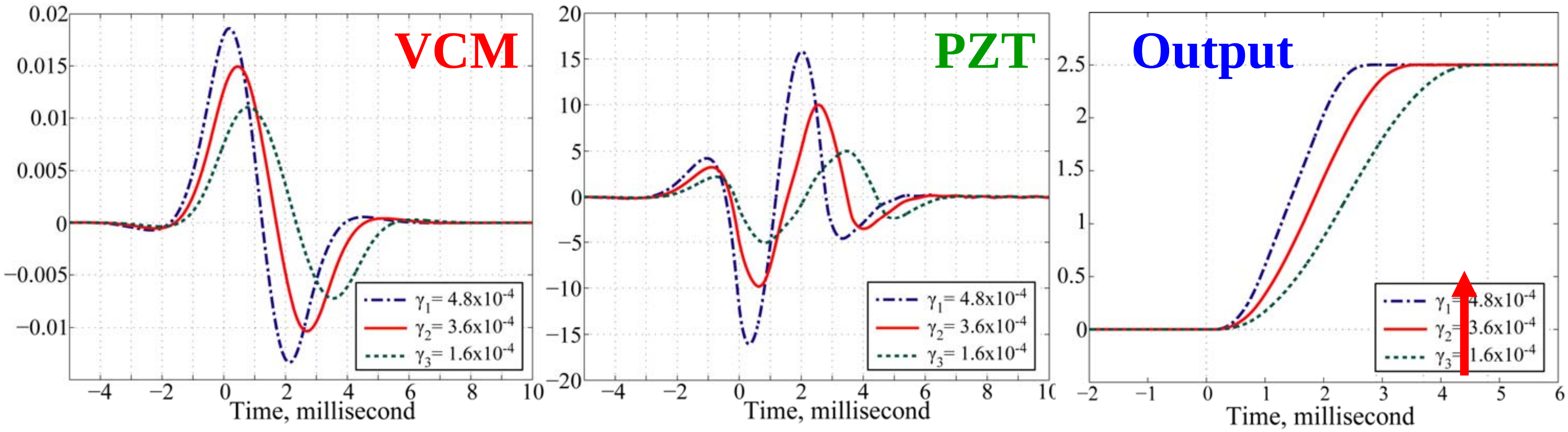



**Choose small $\rho$** → use PZT to help reduce VCM input



**Want faster seek → use larger weight on seek time**

$$\min \ J = \gamma T + \int \left\{ u_{\text{vcm}}(t)^2 + \rho u_{\text{pzt}}(t)^2 \right\} dt$$




**Faster seek comes at price of larger input**

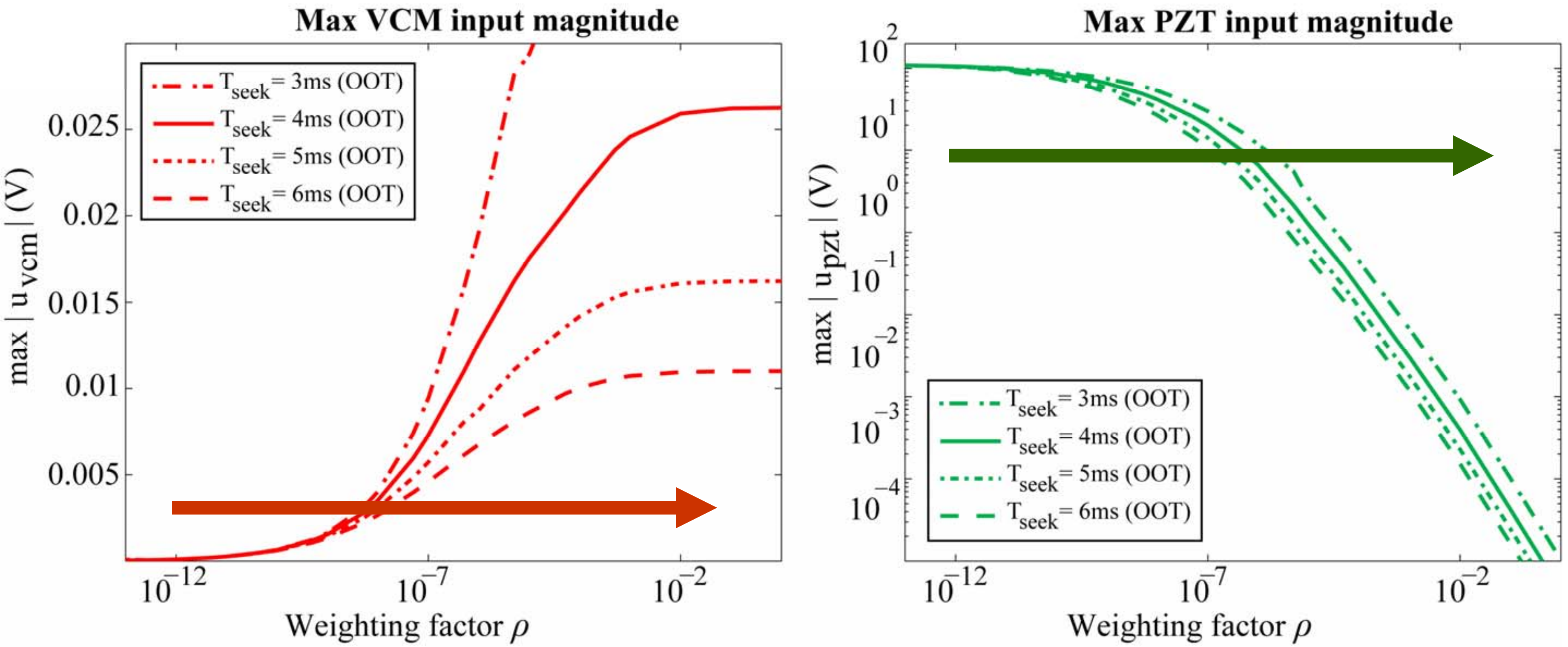
**Dhankorn completed this work --- got his PhD recently**

**Journal article with details is close to completion**

Seek length = 2.5 micron, ρ is fixed

## Summary so far

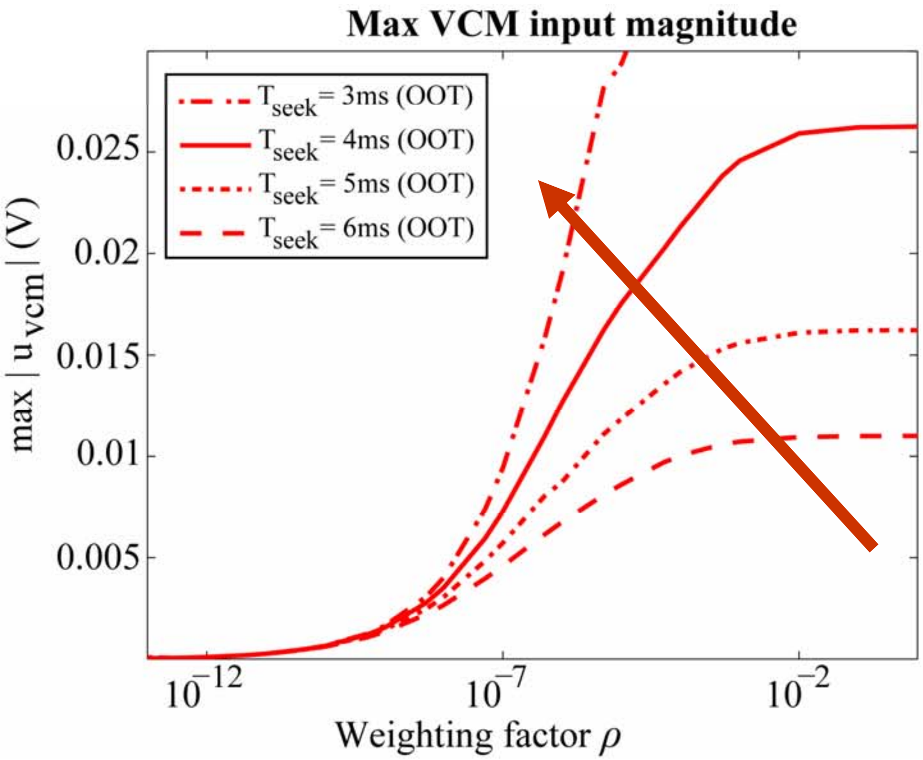
$$\min\ J = \gamma T + \int \left\{ u_{\mathrm{vcm}}(t)^2 + \rho u_{\mathrm{pzt}}(t)^2 \right\} dt$$

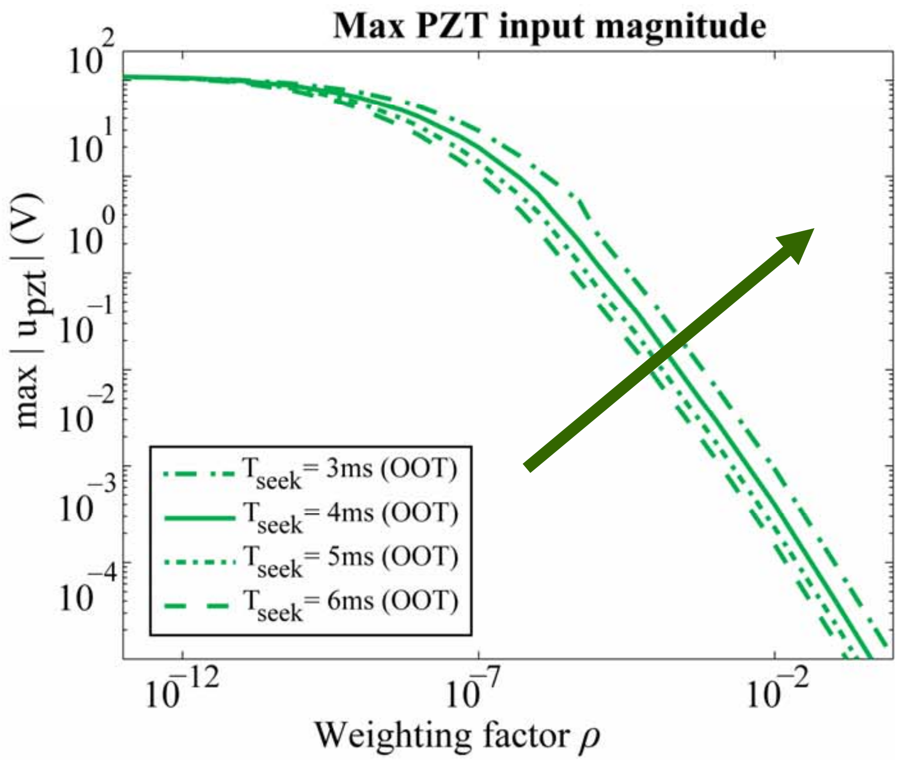



**We can vary the weighting factor to find optimal seek trajectory for a given limit on input**

## Summary so far

$$\min \ J = \gamma T + \int \left\{ u_{\mathrm{vcm}}(t)^2 + \rho u_{\mathrm{pzt}}(t)^2 \right\} dt$$

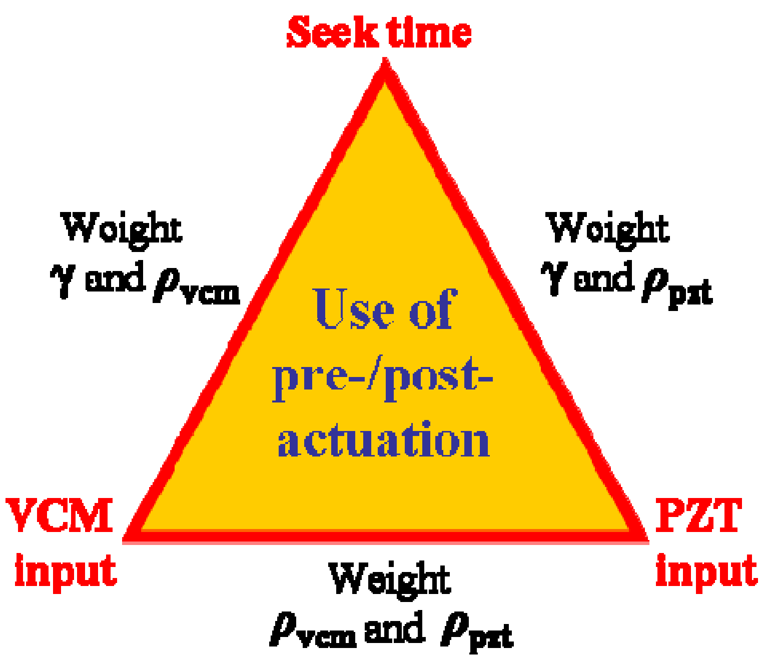



**Decreasing seek time of course increases the input**

# Main Results: trade-off design with VCM vs PZT vs Seek time

$$J = \underbrace{\gamma T}_{\text{Cost of seek time}} + \int \left\{ \underbrace{\rho_{vcm} u_{vcm}(t)^2}_{\text{Cost of VCM input}} + \underbrace{\rho_{pzt} u_{pzt}(t)^2}_{\text{Cost of PZT input}} \right\} dt$$





• **D. Iamratanakul** and S. Devasia "Minimum-Time/Energy Output Transitions for Dual-Stage Systems," To be submitted to ASME J. of Dynamic Systems, Measurement and Control.

• **D. Iamratanakul** and S. Devasia "Minimum-Time/Energy Output- Transitions in Linear Systems," Proc. of the 2004 American Control Conference, Page(s): 4831 – 4836, 2004.

# Implementation issues:

1. How much pre/post-actuation time do we need?

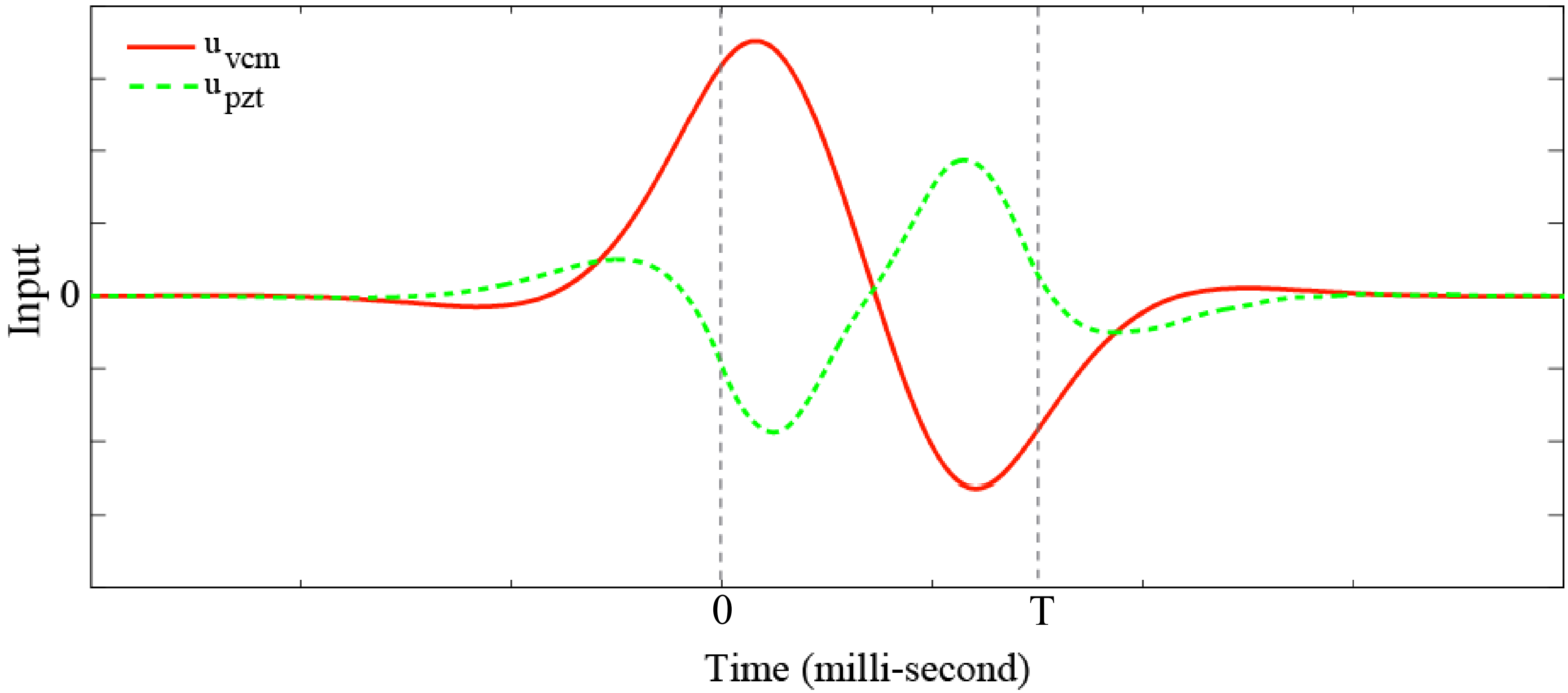
    What if we don't have time for pre-actuation?

2. Will post-actuation cause trouble after seek time?

    How do we use post-actuation for sequential seek?

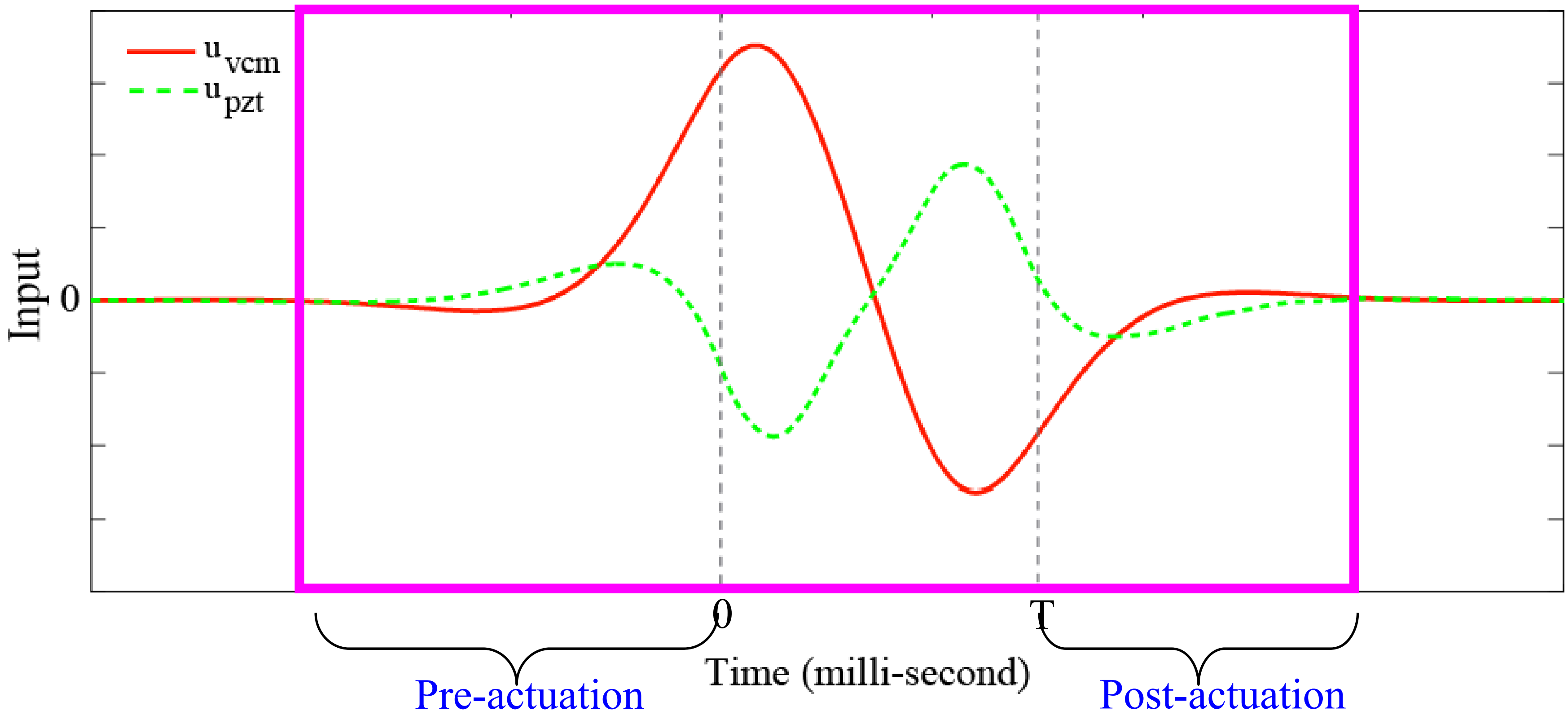
## How much pre/post-actuation time do we need?

Theoretically, pre- and post-actuation inputs require an infinite amount of time



## How much pre/post-actuation time do we need?

Theoretically, pre- and post-actuation inputs require an infinite amount of time




In practice, the pre- and post-actuation time is finite because the pre- and post-actuation inputs decay over time and can be truncated when the input signal becomes small

# Implementation issues:

1. How much pre/post-actuation time do we need?

What if we don't have time for pre-actuation?

## Solutions

- Pose a problem that only use post-actuation
- Add weighting factor to the pre/post-actuation cost


- **D. Iamratanakul**, B. Jordan, K. Leang, and S. Devasia "Optimal Output Transitions for Dual-Stage Systems," Revised version submitted to IEEE Trans. on Control System Technology.
- **D. Iamratanakul**, B. Jordan, K. Leang, and S. Devasia "Optimal Seek-Trajectory Design for Dual-Stage Systems," Proc. of the 2006 American Control Conference, Minneapolis MN.

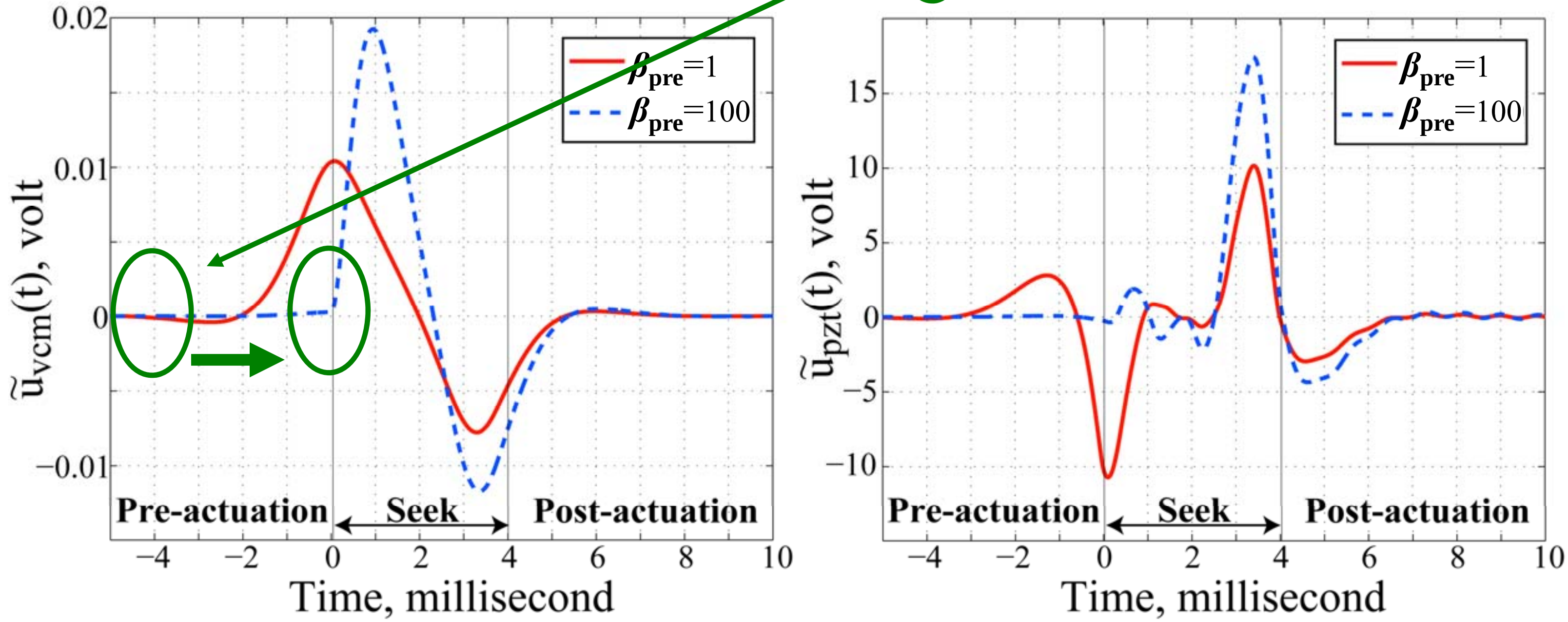
## Weighted pre- and post-actuation

$$J = \int_{-\infty}^{\infty} \beta(t)\left\{u_{vcm}(t)^2 + \rho u_{pzt}(t)^2\right\}dt = \beta_{\mathrm{pre}} J_{\mathrm{pre}} + J_{\mathrm{tran}} + \beta_{\mathrm{post}} J_{\mathrm{post}}$$

Weighting factor for the cost of pre/post-actuation
→ adjust the amount of pre/post-actuation inputs

# Example: Effect of weighting pre-actuation

$$J = \int_{-\infty}^{\infty} \beta(t)\left\{u_{vcm}(t)^2 + \rho u_{pzt}(t)^2\right\}dt = \beta_{\text{pre}} J_{\text{pre}} + J_{\text{tran}} + \beta_{\text{post}} J_{\text{post}}$$




- The pre-actuation time can be adjusted by choosing an appropriate choice of the weighting factor $\boldsymbol{\beta}_{\textbf{pre}}$

- Even without pre-actuation, post-actuation still helps improve the seek performance

| $\beta_{pre}$ | Pre-actuation time |
|---|---|
| 0.001 | 61.78 |
| 0.01 | 16.95 |
| 0.1 | 5.92 |
| 1 | 5.76 |
| 10 | 5.10 |
| 100 | 2.82 |
| 1000 | 0.30 |

## Implementation issues:

1. How much pre/post-actuation time do we need?

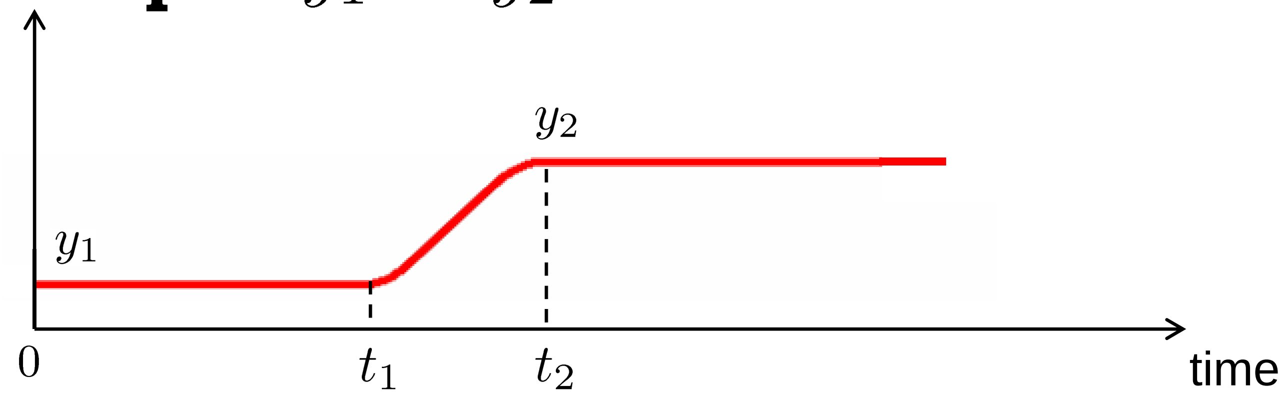
What if we don't have time for pre-actuation?

2. Will post-actuation cause trouble after seek time?

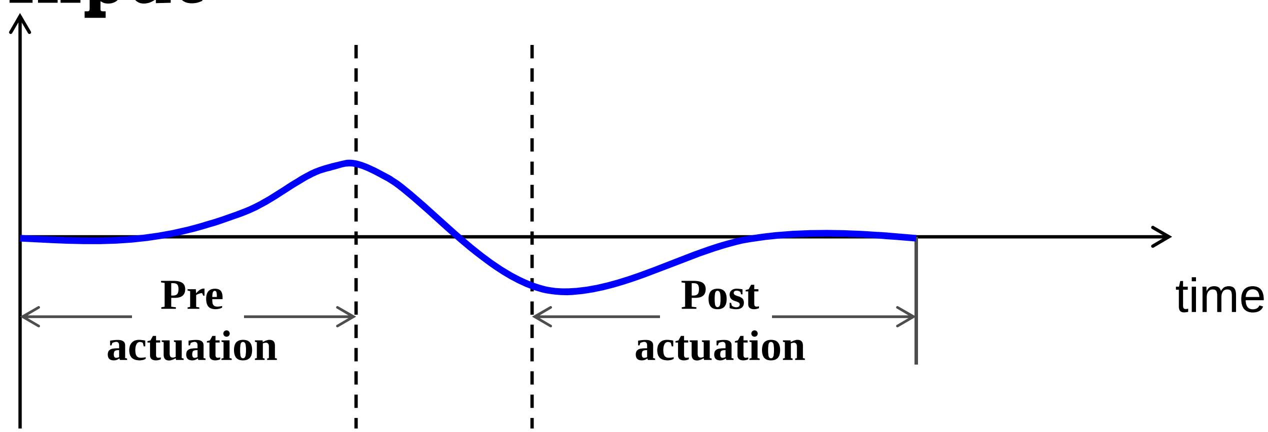
How do we use post-actuation for sequential seek?

# Sequential seek

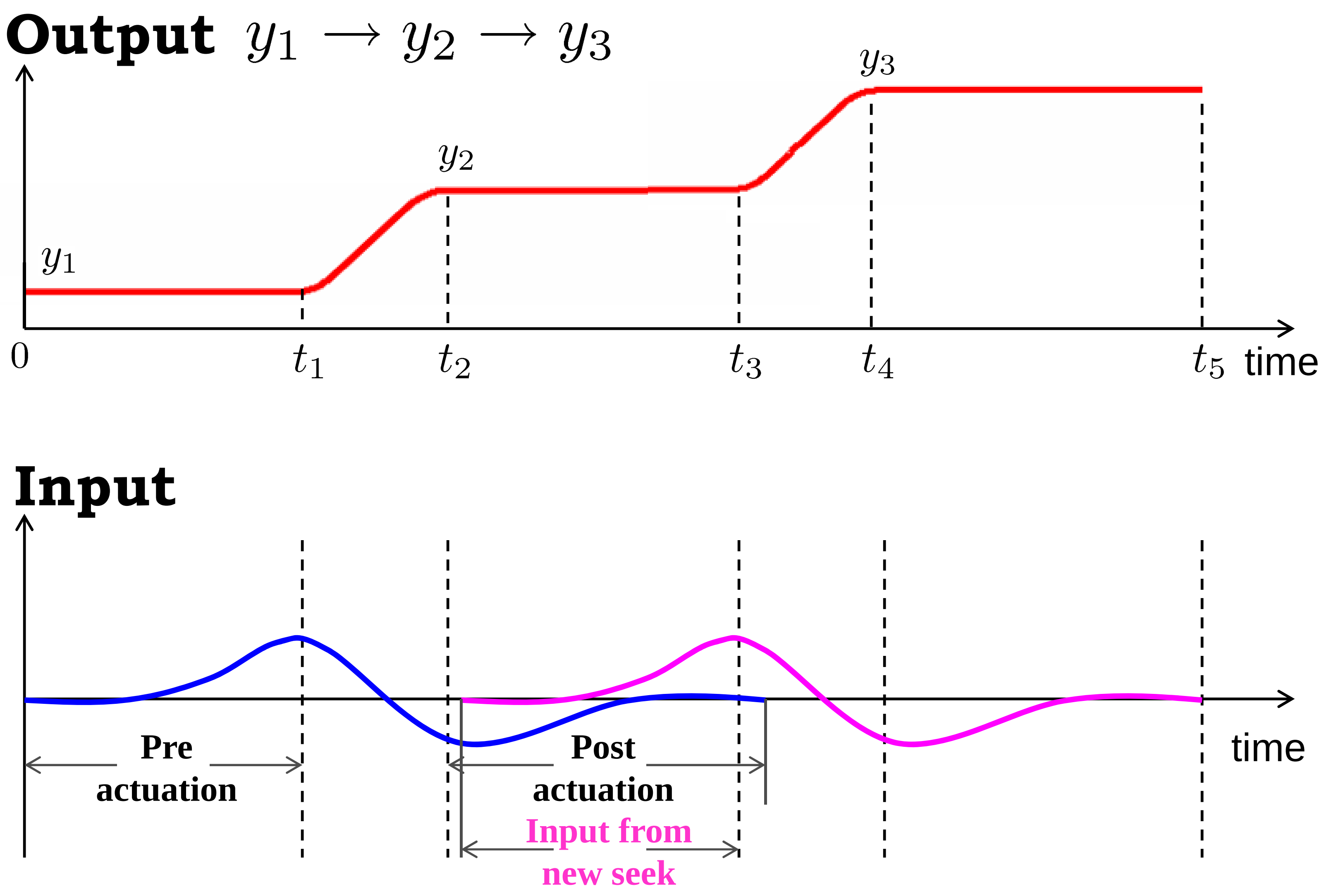
**Output** $y_1 \longrightarrow y_2$



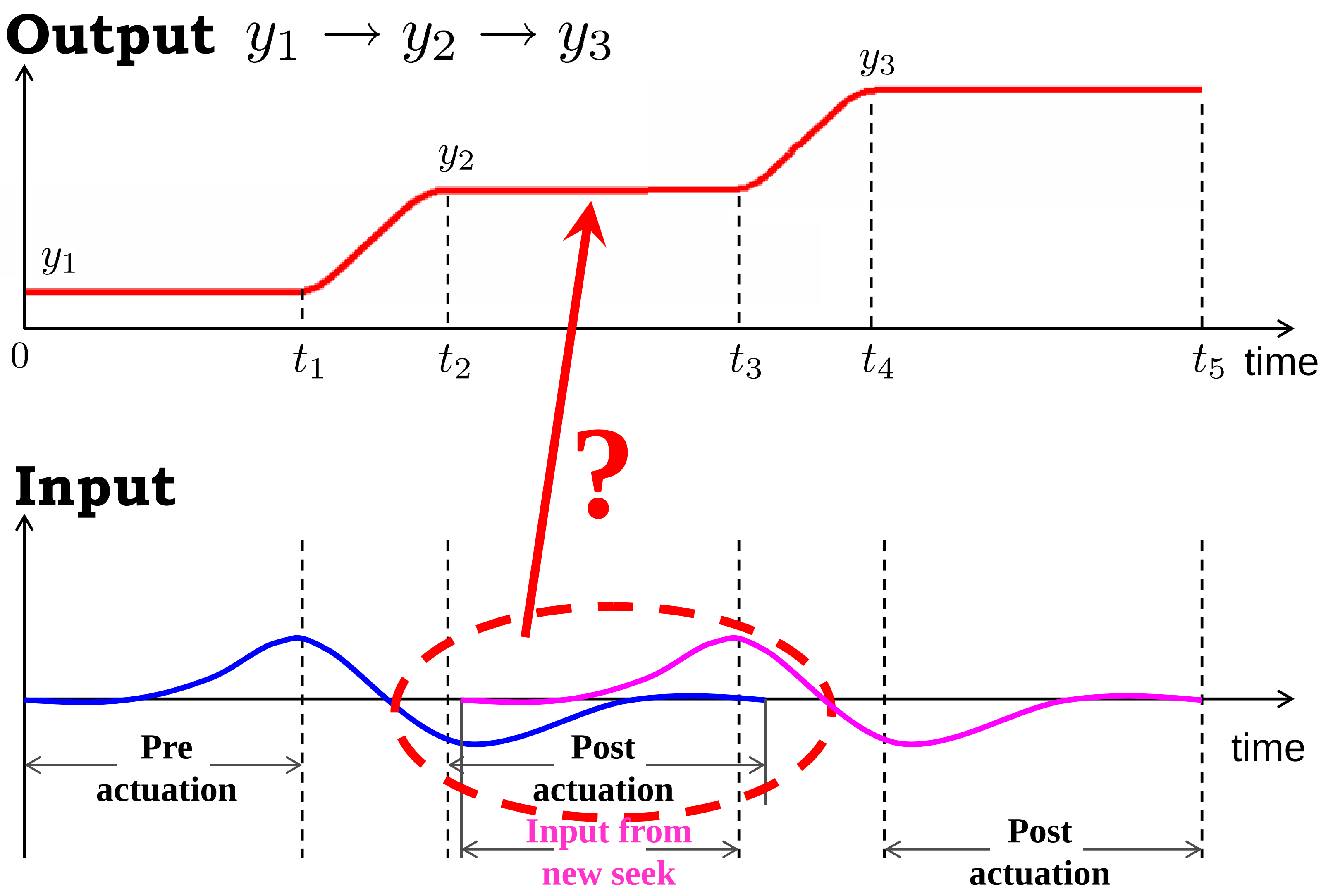

**Input**



# Sequential seek

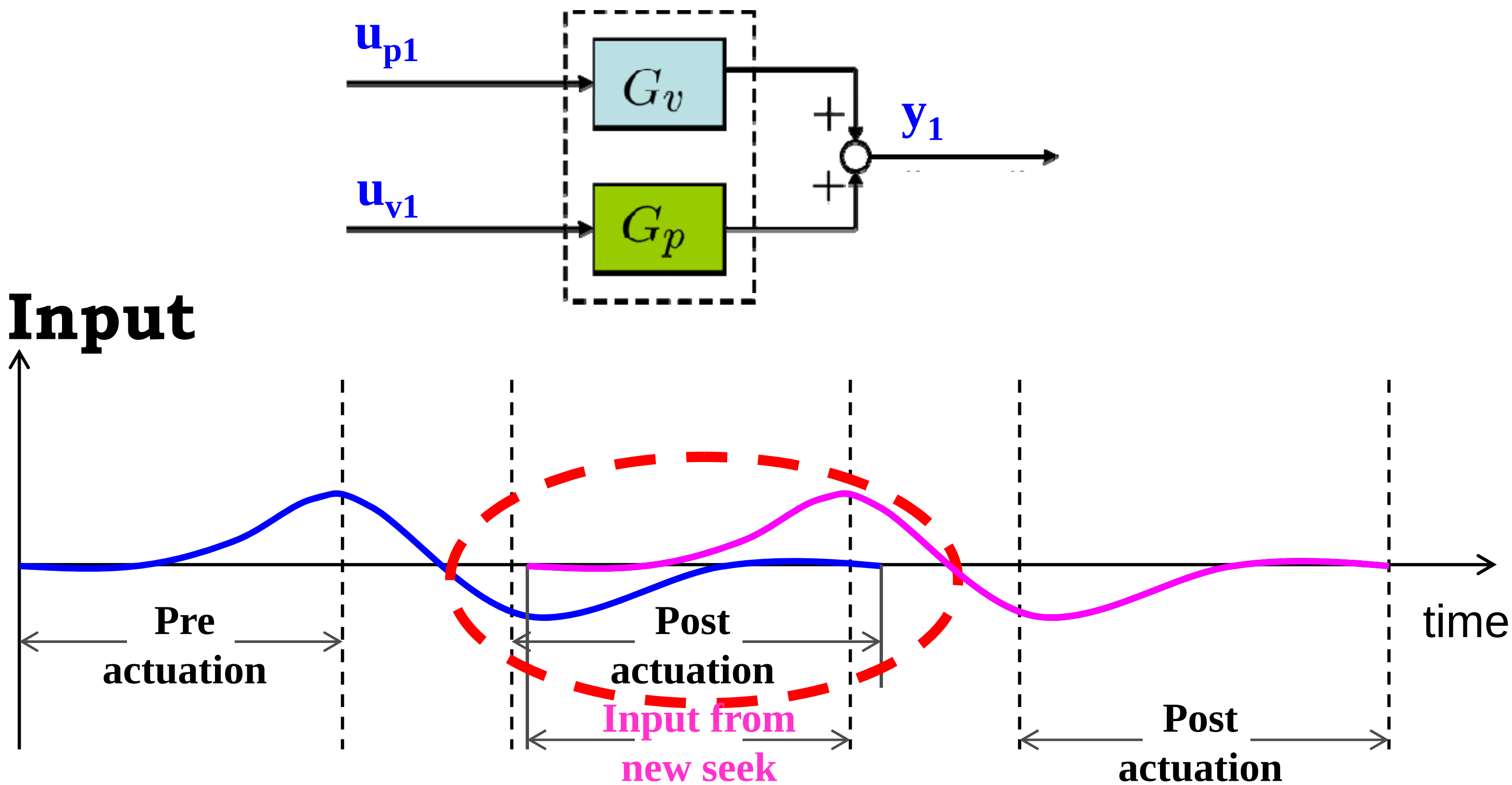


# Will the overlapped input cause problems?



# Will the overlapped input cause trouble?

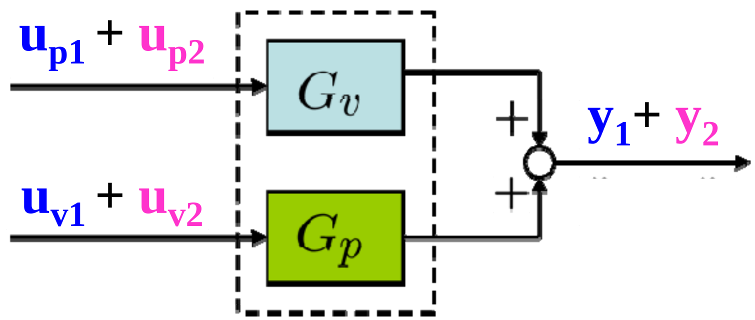
Answer: **NO**. Because the pre/post-actuation inputs do not affect the output, i.e. $y = \bar{y}, \;\ddot{y} = 0$

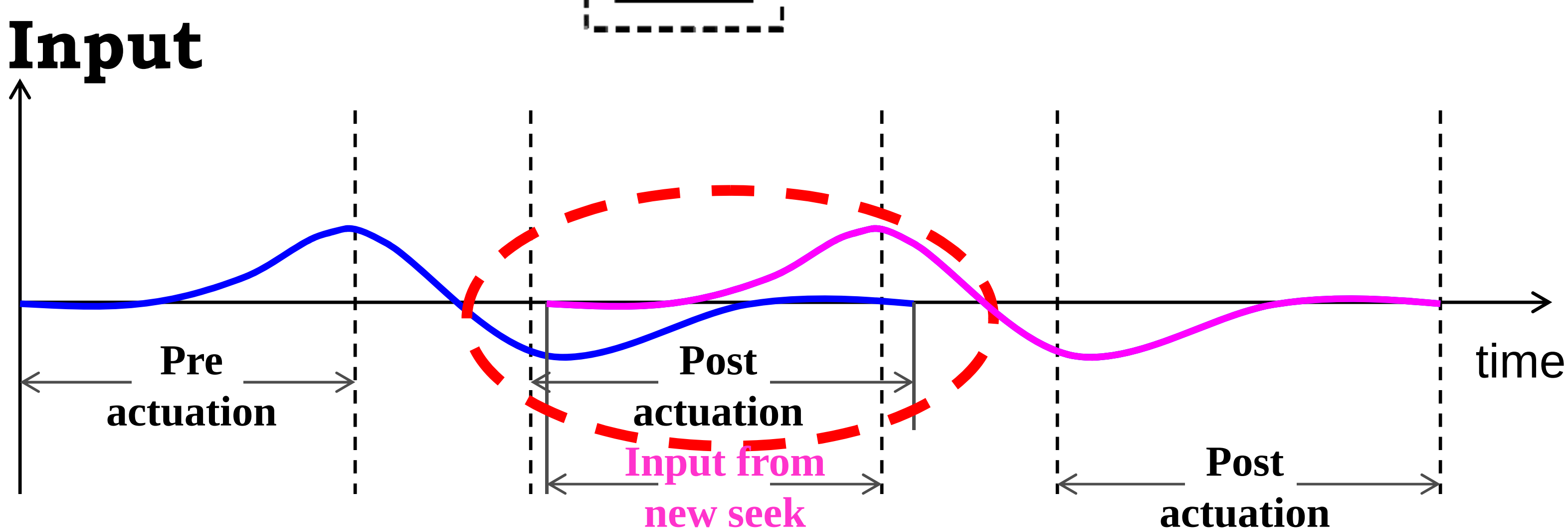


# Will the overlapped input cause trouble?

Answer: **NO**. Because the pre/post-actuation inputs do not affect the output, i.e. $y = \bar{y}, \ \ddot{y} = 0$

Assuming that the system is linear → superposition holds

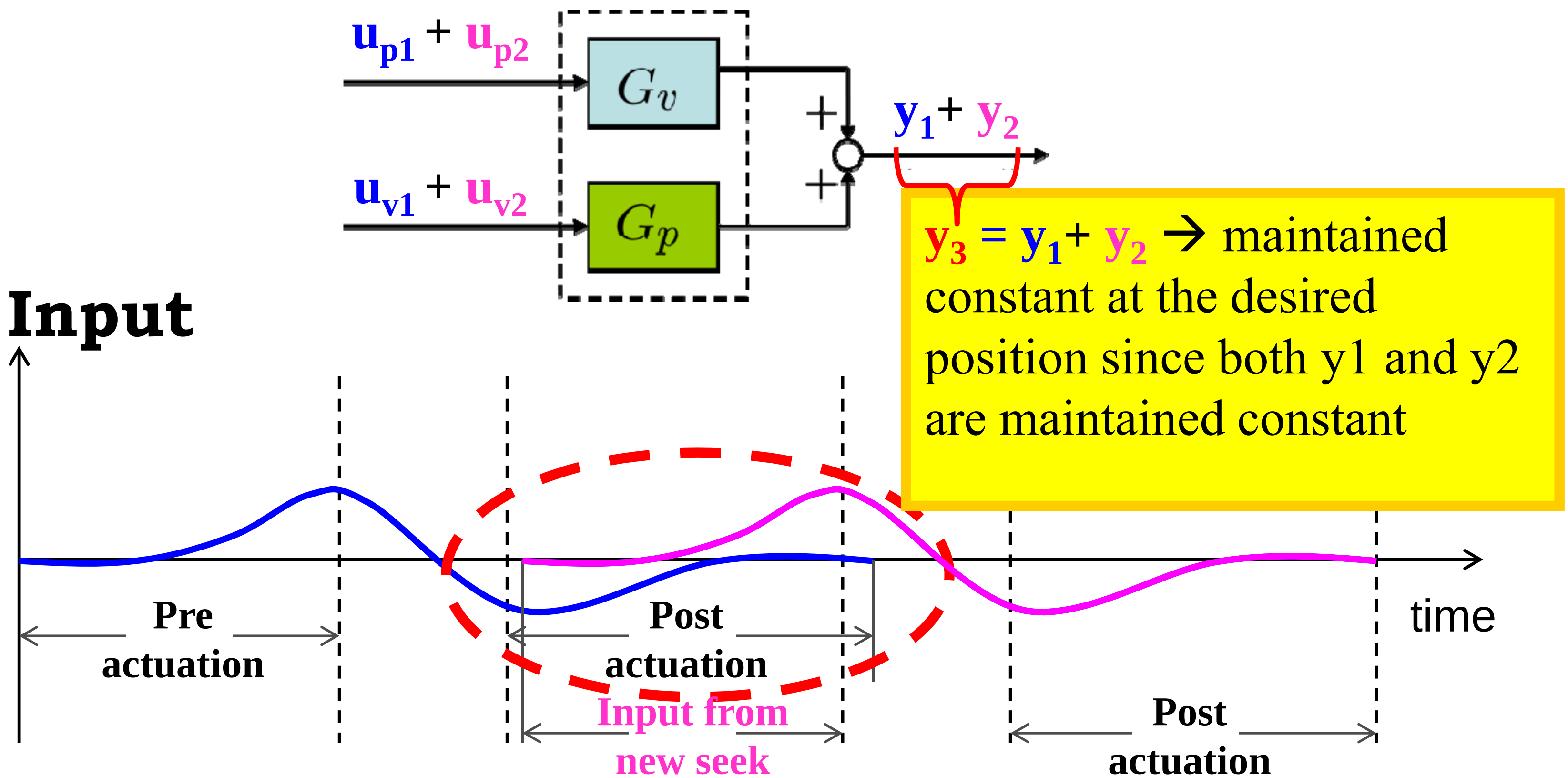


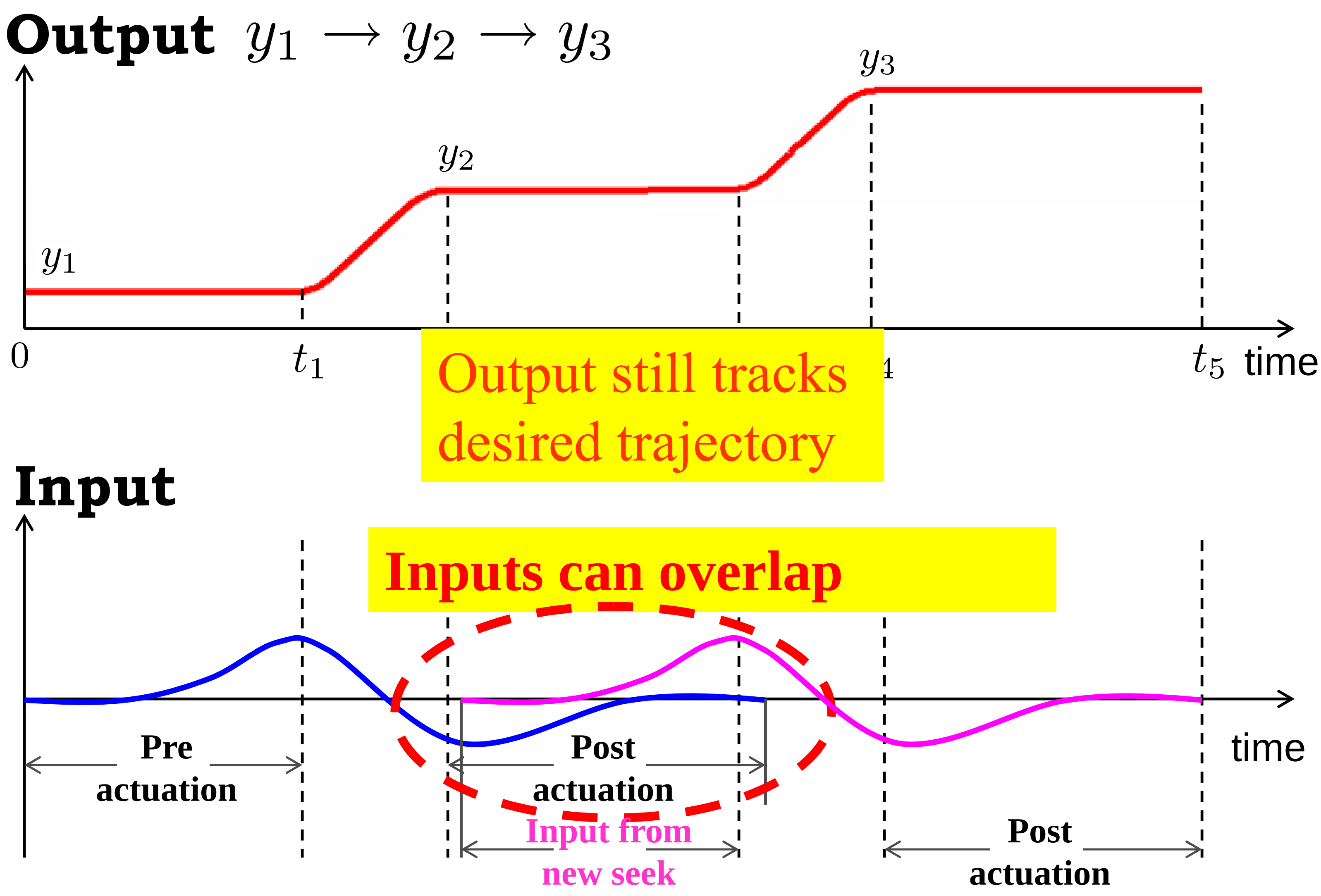
# Will the overlapped input cause trouble?

Answer: **NO**. Because the pre/post-actuation inputs do not affect the output, i.e. $y = \bar{y},\ \ddot{y} = 0$

Assuming that the system is linear → superposition holds

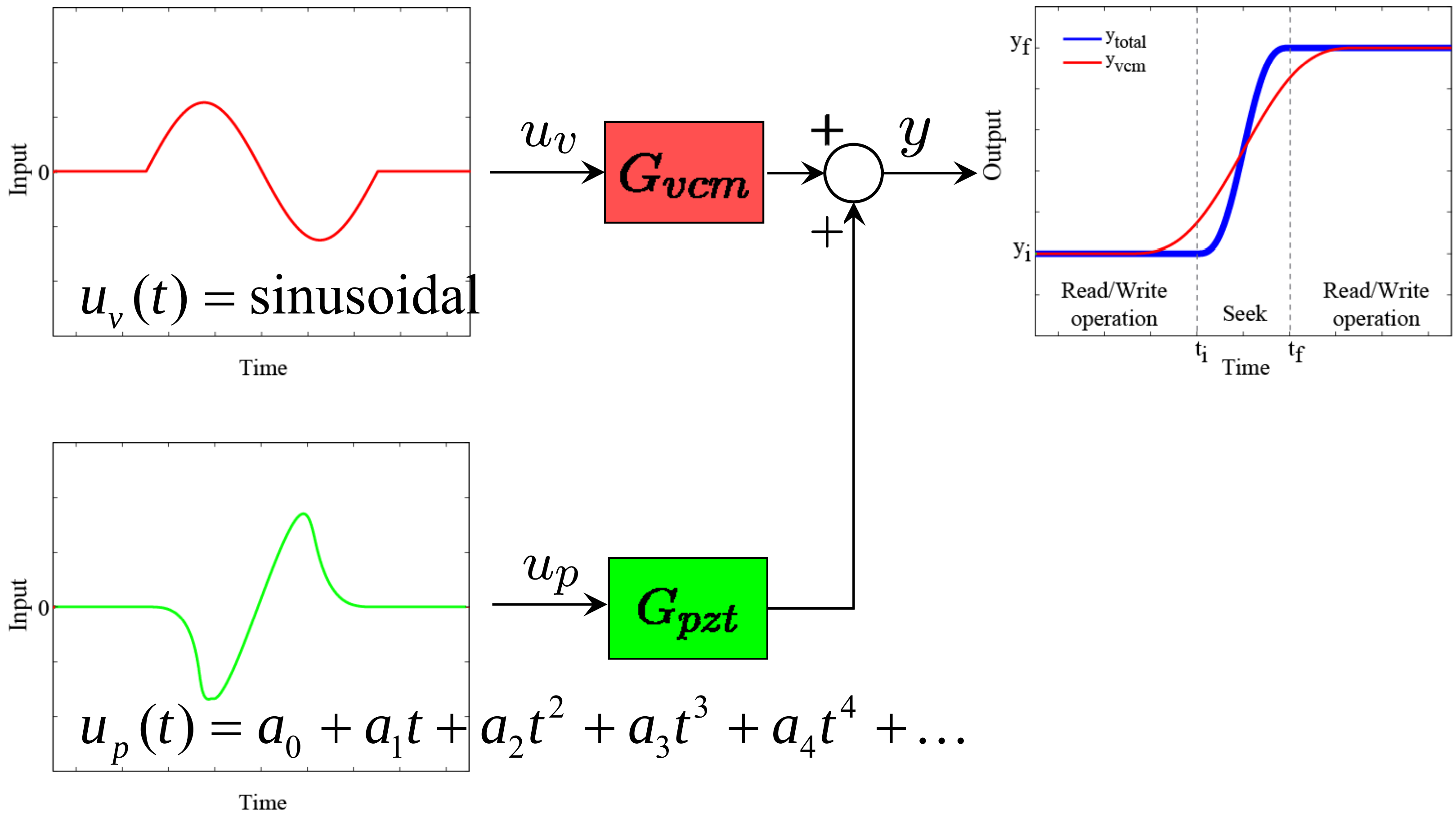


# Sequential seek



# Sinusoidal/polynomial input profile with dual-stage actuator

**Motivation:** Trade-off between optimality and ease in implementation of the control algorithm




$$u_v(t) = \text{sinusoidal}$$

$$u_p(t) = a_0 + a_1 t + a_2 t^2 + a_3 t^3 + a_4 t^4 + \ldots$$


• Hatano Y., Iamratanakul D., and Devasia S., "Seek Control for Dual-Stage Hard Disk Drives," Proc. of the 2006 International Symposium on Flexible Automation, Osaka, Japan.

# Key issues in polynomial input

- **Can we optimize the polynomial profile?**

If we use higher-order polynomial (higher than the minimum degree requirement), we can optimally choose the coefficient to minimize a cost function

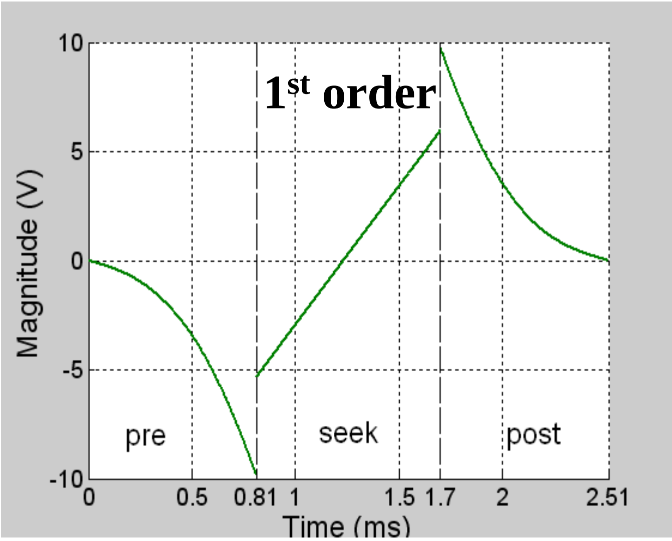
$$u_p(t) = a_0 + a_1 t + a_2 t^2 + a_3 t^3 + a_4 t^4 + \ldots$$

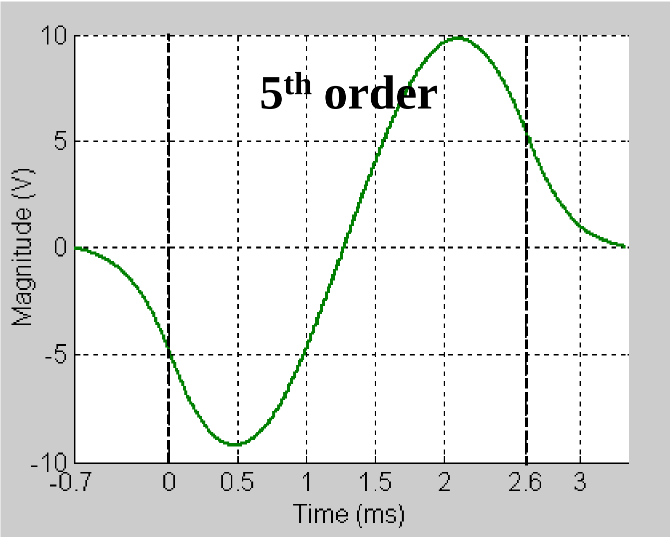
# Key issues in polynomial input

- **Can we optimize the polynomial profile?**

- **Can we obtain a smooth or continuous input trajectory?**

Extra constraints can be imposed at the cost of higher-order polynomial profile



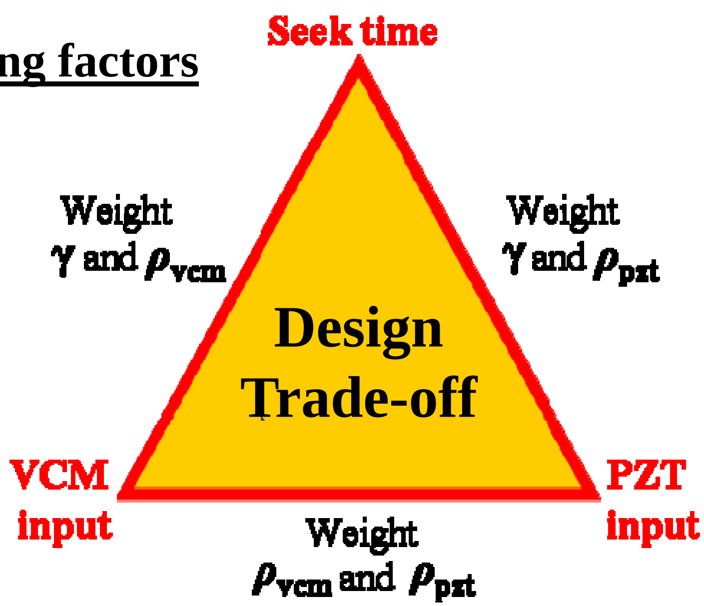



# Summary of Part 1: Seek control

**Main Issue: Trade-offs between VCM vs PZT vs Seek time?**

$$J = \underbrace{\gamma T}_{\text{Cost of seek time}} + \int \left\{ \underbrace{\rho_{vcm} u_{vcm}(t)^2}_{\text{Cost of VCM input}} + \underbrace{\rho_{pzt} u_{pzt}(t)^2}_{\text{Cost of PZT input}} \right\} dt$$

**Weighting factors**




**Our contribution:** Use pre/post actuation (input applied before and after seek time interval) to improve seek performance

**Studied Both:** Computational (Design Tradeoffs) and Implementation Issues

# Part 2: Settling Control

- **Second actuator cannot directly help during large seek (tends to be saturated).**

- **Settling for large seek with dual-stage system**
  - **Can second actuator help in settling?**
  - **Challenge: handle different initial conditions such as second actuator being saturated**

- **Our Approach**
  - **Yes. Second actuator can improve settling**
  - **Developed an optimal inverse feedforward approach**
  - **Proposed method avoids online computation**

# Background: Settling Control for Dual-Stage

- **Intuitive: Second stage should improve performance**

- **Previous works**
  - **Design of dual-stage feedback (as opposed to single stage)**
  - **Changing Initial Conditions of feedback controller for better settling**
  - **Adding feedforward to improve settling**

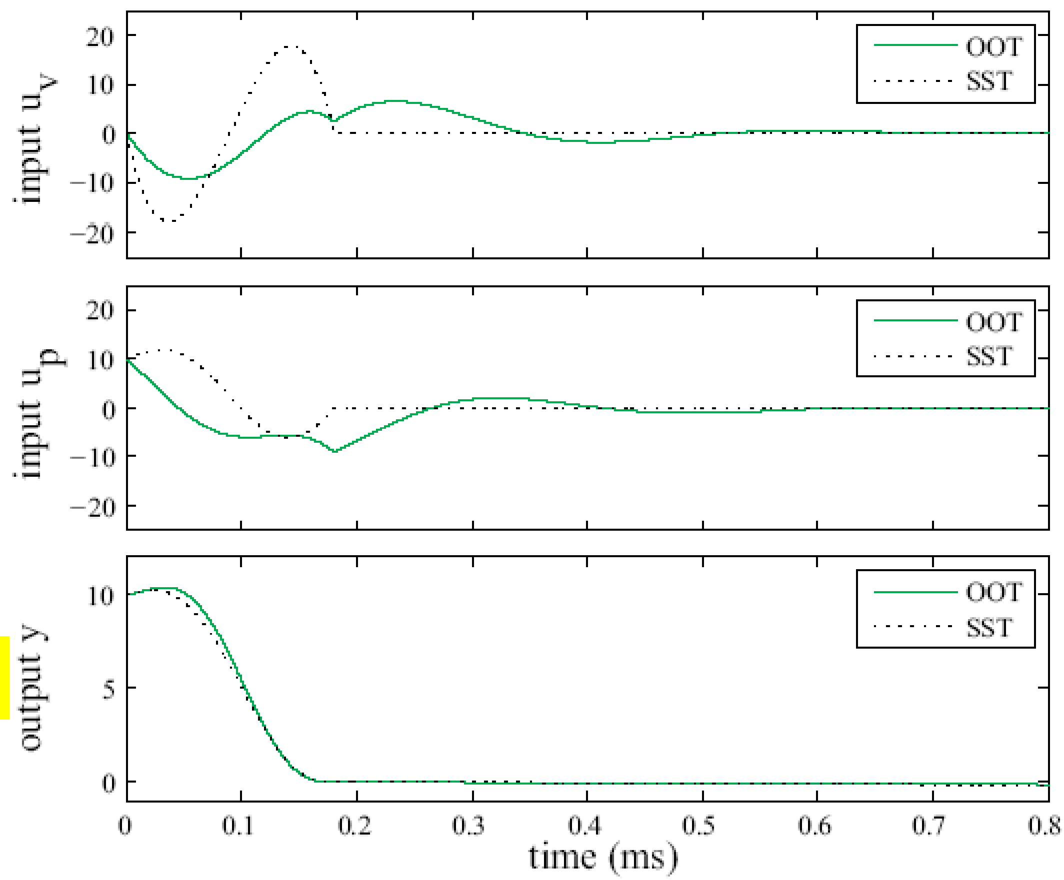
- **Our Approach**
  - **We used pre- and post-actuation to improve seek**
  - **Can such an approach improve settling?**
  - **Will it be prohibitively computationally intensive?**

# Main Idea? State vs. Output Settling

VCM

PZT

Output




Dotted line --- state settling without post-actuation

Green line --- output settling with post-actuation

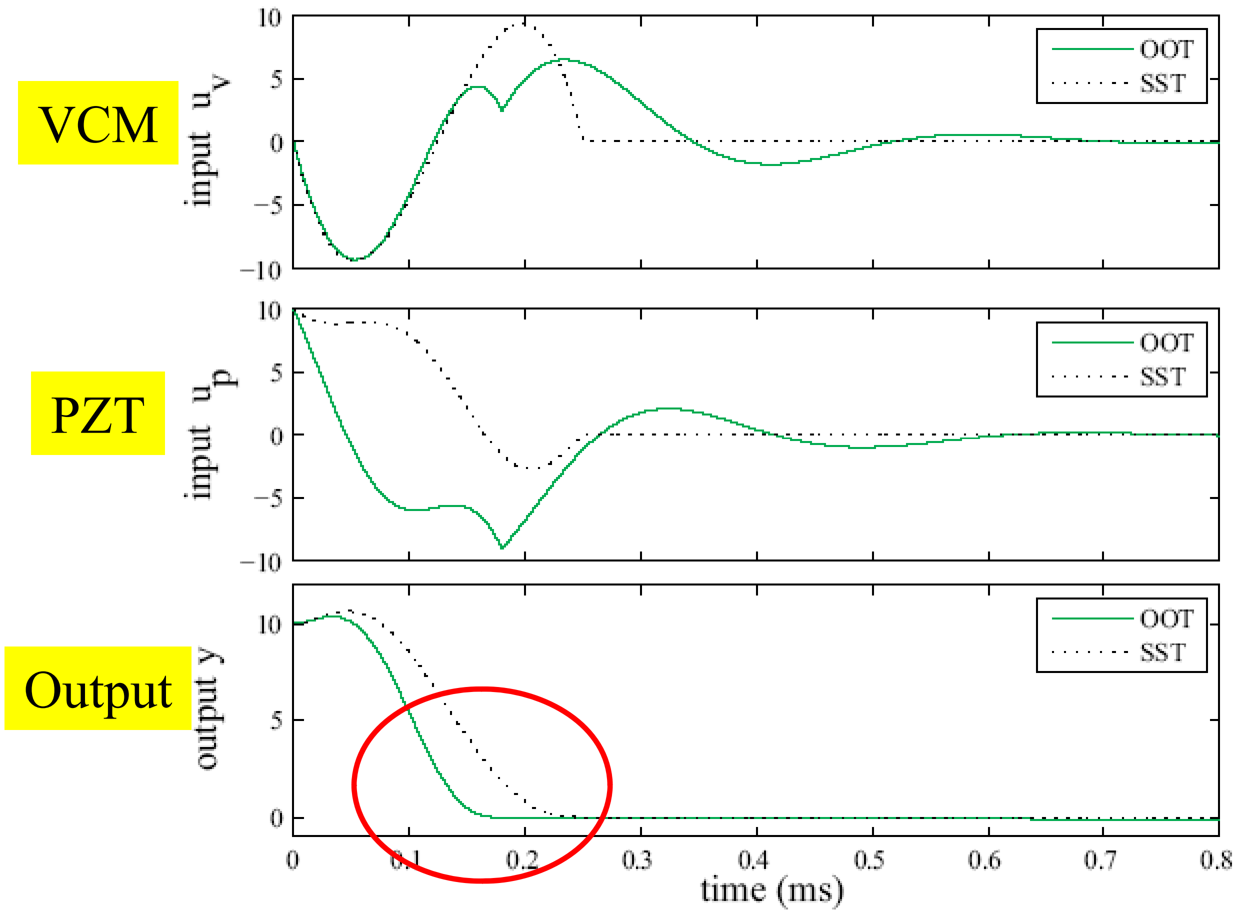
Use of Post-actuation (i.e., after bringing the output to zero and maintaining it zero)

Allows a smaller VCM Input when compared to State-settling

But output settling occurs in same amount of time

**Note: PZT is saturated at the beginning (It is still useful)**

# Advantage? Faster Settle




**In this simulation both Input magnitudes are similar**

**Faster settle for same input magnitudes**

**Note: PZT is saturated at the beginning (It is still useful)**

Dotted line --- state settling without post-actuation

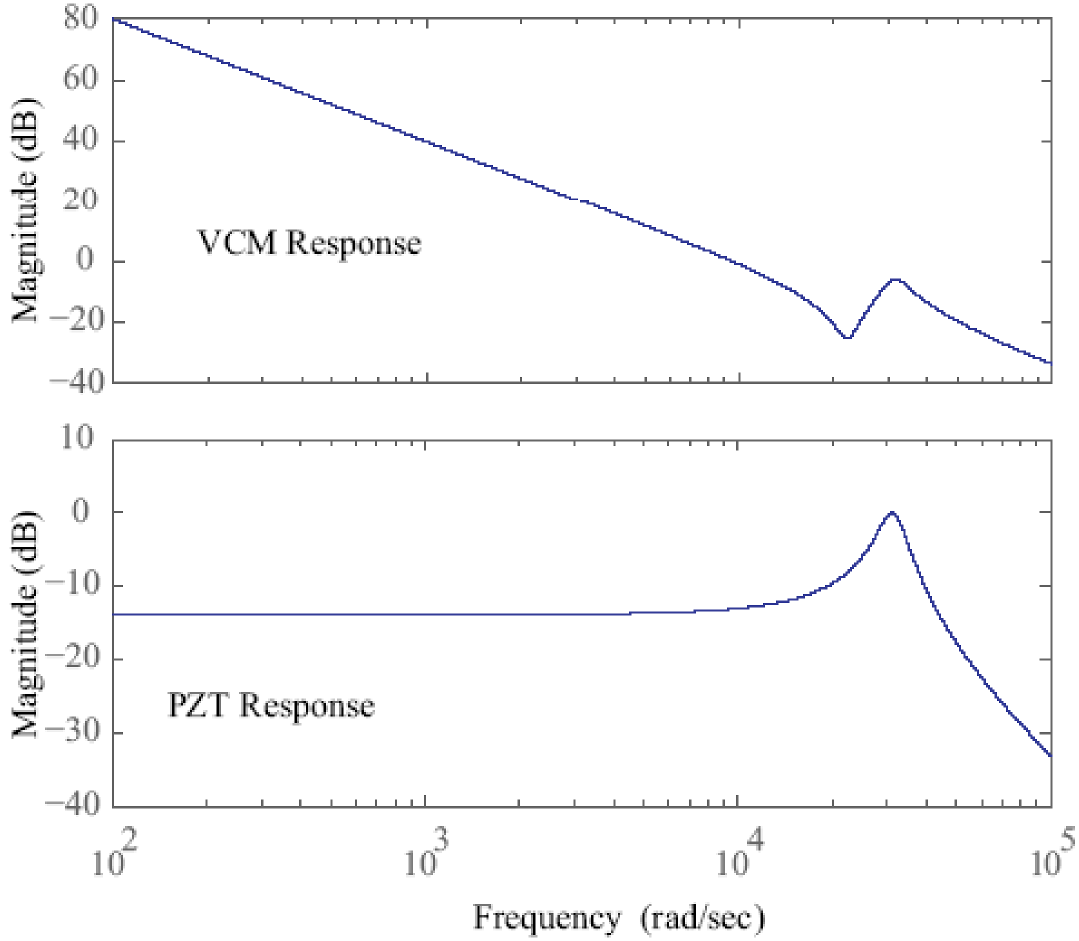
Green line --- output settling with post-actuation

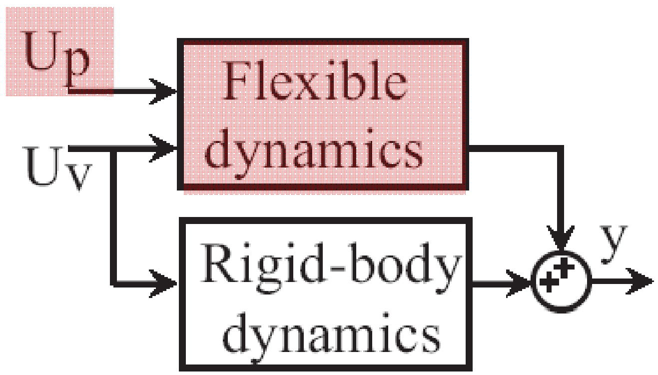
# Theory/Analysis

S Devasia. "Optimal Output Transition for Settling Control in Hard-Disk Drives with Dual-Stage Actuators."
To be presented at the IEEE Multi-Conference on Systems and Control, October, 2007.

# Simulation Results

# Simulation Results



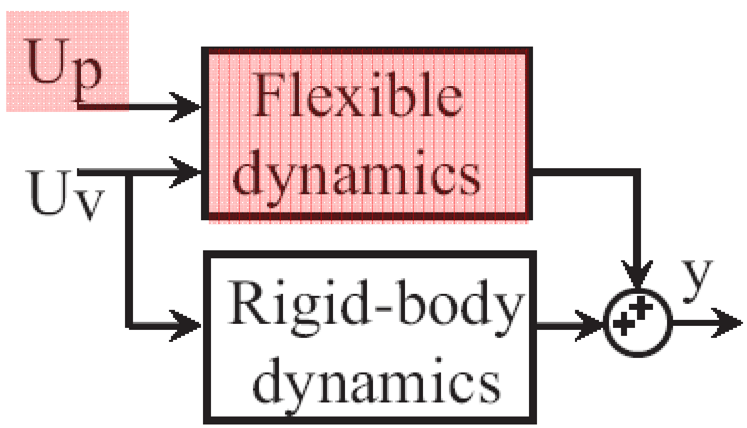
# Simplified Model

First Actuator, VCM

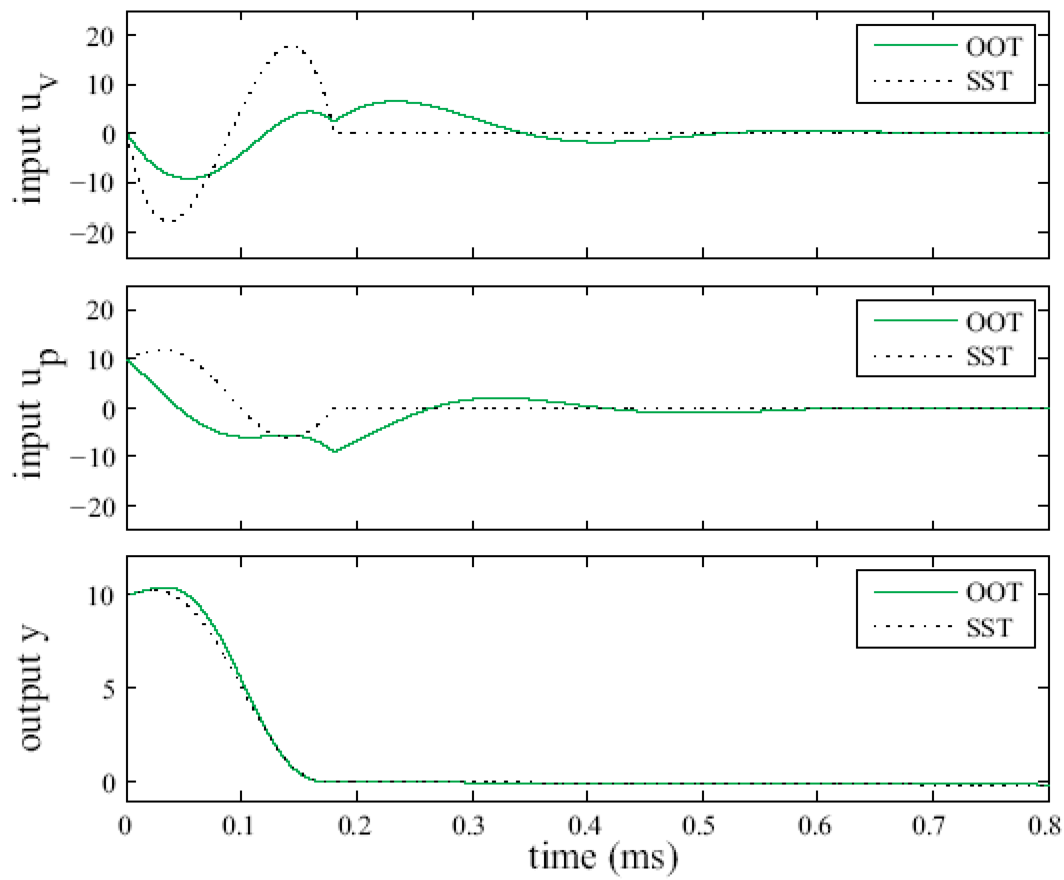
$$\begin{aligned} G_v(s) &= y(s)/u_v(s) \\ &= b_1/s^2 \; + b_2/(s^2 + a_1 s + a_0) \end{aligned} \qquad (1)$$

Second Actuator

$$\begin{aligned} G_p(s) &= y(s)/u_p(s) \\ &= b_3/(s^2 + a_1 s + a_0). \end{aligned}$$



# Main Idea! State vs. Output Settling




Use of Post-actuation (i.e., after bringing the output to zero and maintaining it zero)

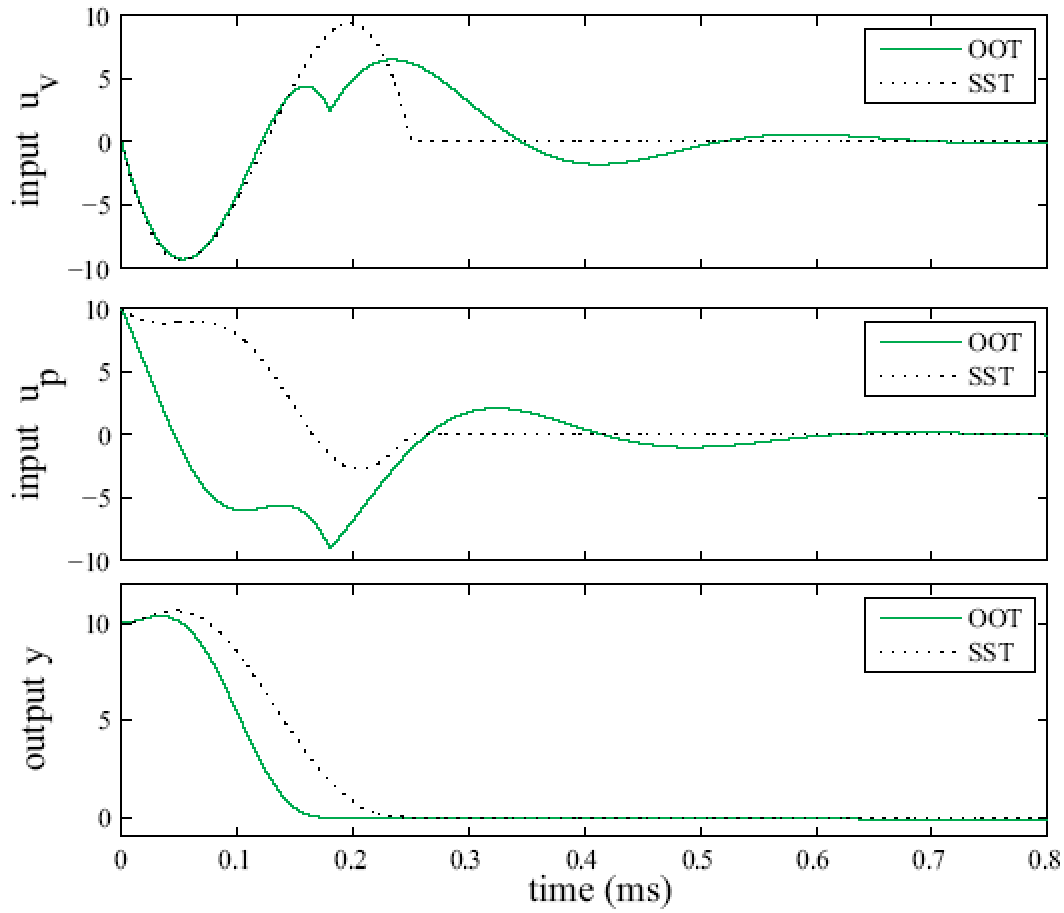
Allows a smaller VCM Input when compared to State-settling

But output settling occurs in same amount of time

**Note: PZT is saturated at the beginning (It is still useful)**

Dotted line --- state settling without post-actuation
Green line --- output settling with post-actuation

# Advantage? Faster Settle




Faster settle for same input magnitudes

**Note: PZT is saturated at the beginning (It is still useful)**

Dotted line --- state settling without post-actuation

Green line --- output settling with post-actuation

# Computational Issues

# Computational Issues

Inputs can be found explicitly.

**However, it still takes time to compute**

$$\boxed{\hat{u}^*(t) = R^{-1}\hat{B}^T e^{\hat{A}^T(t_f - t)}\hat{G}^{-1}\left[U_\eta \eta^* - e^{\hat{A}(t_f)}\hat{x}(0)\right]}$$

$$\begin{array}{rcl} \hat{u}_p^*(t) & = & -N^{-1}\left(B_\eta^T W + S\right)\left[e^{A_{CL}(t-t_f)}\eta^*\right] \\ \hat{u}_v^*(t) & = & C_\eta \left[e^{A_{CL}(t-t_f)}\eta^*\right] + D_\eta \hat{u}_p^*(t) \\ A_{CL} & = & A_\eta - B_\eta N^{-1}\left(B_\eta^T W + S\right) \end{array}$$

# Worse: Dependence on ICs

Inputs can be found explicitly.
However, it still takes time
**Choice of weights depends on initial conditions**

$$\hat{u}^*(t) = R^{-1}\hat{B}^T e^{\hat{A}^T(t_f - t)}\hat{G}^{-1}\left[U_\eta \eta^* - e^{\hat{A}(t_f)}\hat{x}(0)\right]$$

$$\begin{aligned}
\hat{u}_p^*(t) &= -N^{-1}\left(B_\eta^T W + S\right)\left[e^{A_{CL}(t-t_f)}\eta^*\right] \\
\hat{u}_v^*(t) &= C_\eta\left[e^{A_{CL}(t-t_f)}\eta^*\right] + D_\eta \hat{u}_p^*(t) \\
A_{CL} &= A_\eta - B_\eta N^{-1}\left(B_\eta^T W + S\right)
\end{aligned}$$

# Approach: Exploit Linearity

If initial condition at start of seek settle is linear combination of states then

$$X_3 = \gamma_1 X_1 + \gamma_2 X_2$$

Choose the input as the same linear combination of previous inputs

$$\hat{u}_3 = \gamma_1 \hat{u}_1 + \gamma_2 \hat{u}_2$$

# Does it work?

If initial condition at start of seek settle is linear combination of states then

$$X_3 = \gamma_1 X_1 + \gamma_2 X_2$$

Choose the input as the same linear combination of previous inputs

$$\hat{u}_3 = \gamma_1 \hat{u}_1 + \gamma_2 \hat{u}_2$$

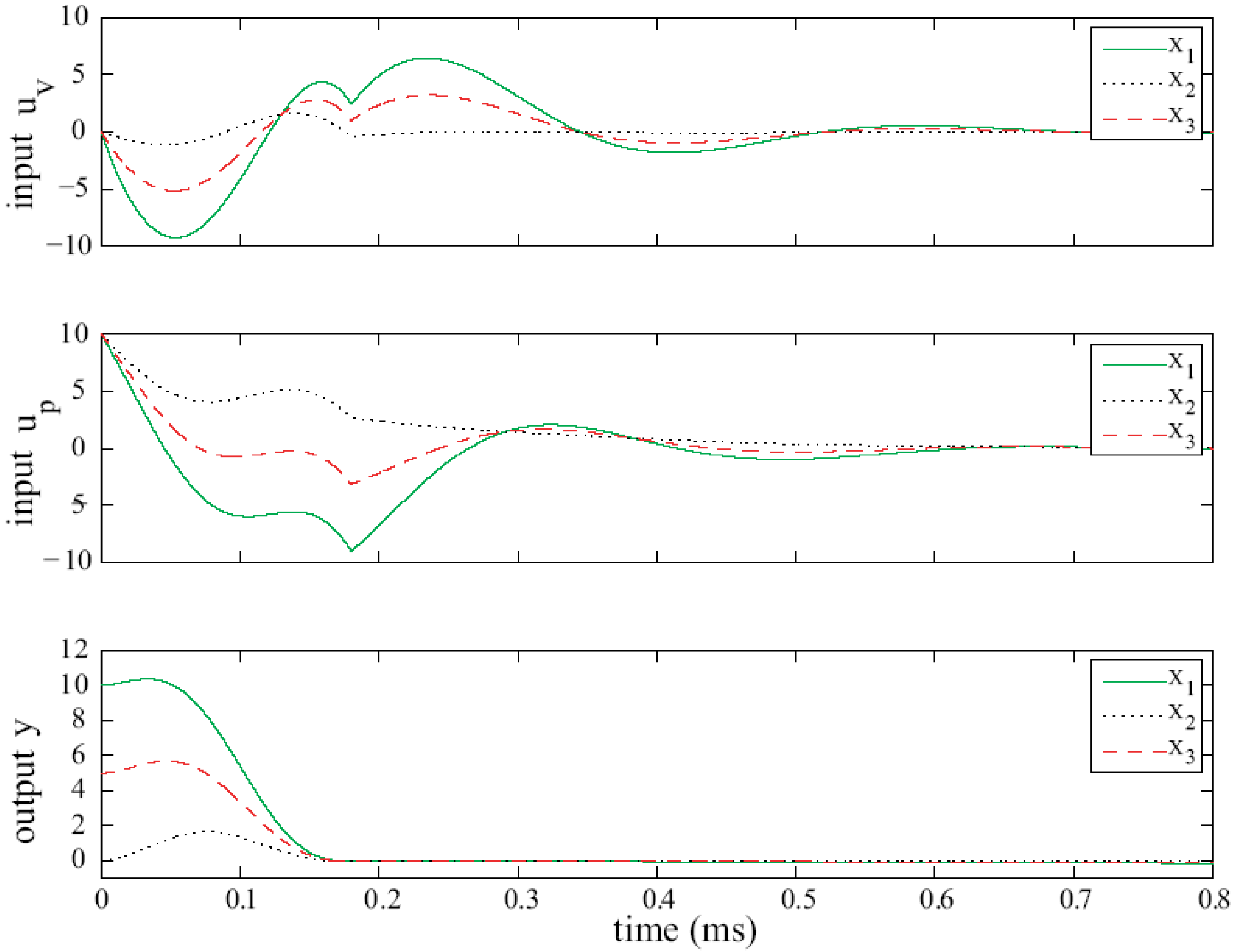
**Guarantees** settling time is not longer than previous ones!
**Convexity** can be used to ensure input is bounded
**Prefilter** allows inputs to be saturated at the start

# Example

VCM

PZT

Output



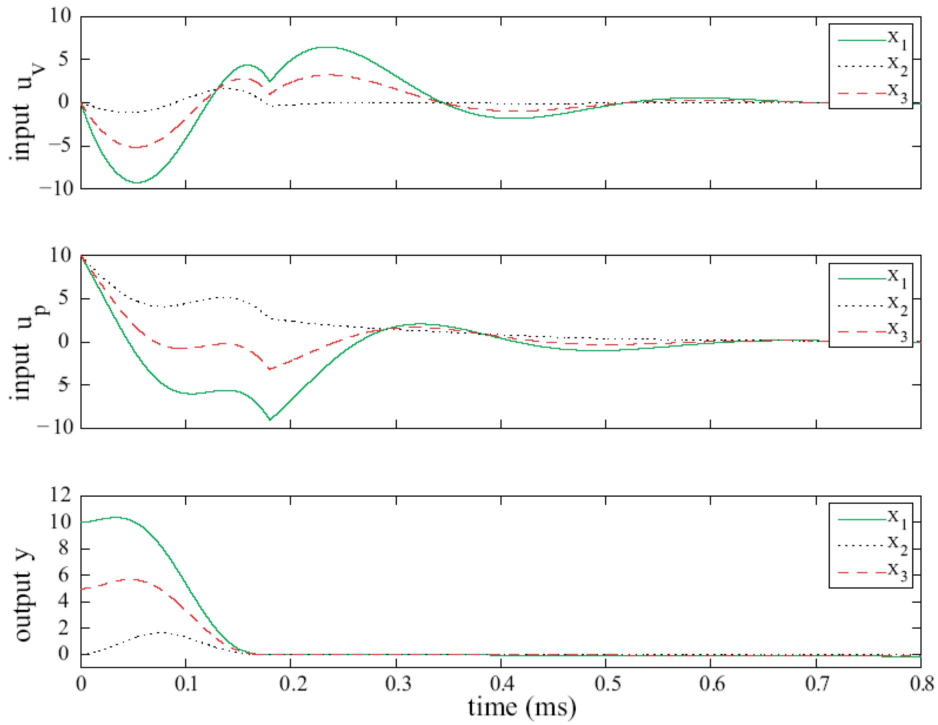

Red is linear combination of first two states

# Example

VCM

PZT

Output




**Guarantees** settling time is not longer than previous ones!

**Convexity** can be used to ensure input is bounded

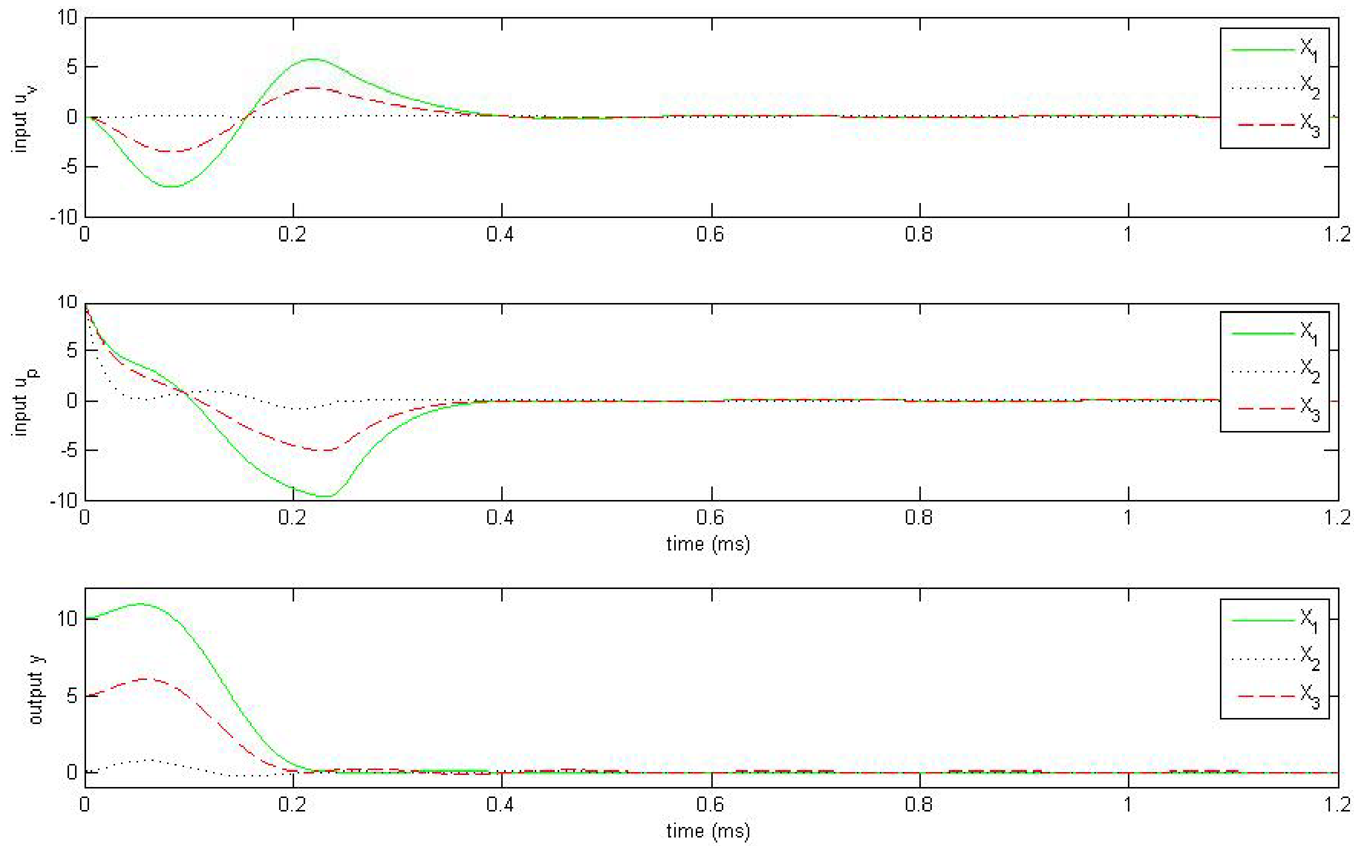
**Prefilter** allows inputs (PZT) to be saturated at the start

# Effect of higher-order pre-filters

VCM

PZT

Output




**Second order pre-filter allows smoother inputs**

**(when compared to first-order pre-filter)**

**(Settling takes more time with Smoother inputs 0.2**
**Was 0.18 before)**

# Conclusion for Part 2:
# Settling Control for Large-Seek

- **Settling for large seek with dual-stage system**
  - **Can second actuator help in settling?**
  - **Challenge: Second actuator saturated at start**

- **Our Approach**
  - **Yes. Second actuator can improve settling**
  - **Developed an optimal inverse feedforward approach which uses post-actuation idea!**
  - **Proposed method avoids online computation for different initial conditions**