Control of PDE Systems

Lecture 8 (Meetings 15 & 16)

Eugenio Schuster

schuster@lehigh.edu
Mechanical Engineering and Mechanics
Lehigh University

Material provided by Prof. Miroslav Krstic and Dr. Andrey Smyshlyaev (UCSD)

Motion Planning for PDEs

- Combined control solution: Feedforward (FF) + Feedback (FB)
- Trajectory generation, or motion planning $\rightarrow$ open-loop control for PDEs
- Reference trajectory + Combined FF+FB $\rightarrow$ Trajectory stabilization
- Trajectory tracking: state reference (relevant) $\rightarrow$ output reference (important)
  - Start from output reference trajectory, e.g. $u(0,t) = u^r(0,t)$
  - Generates the state trajectory $u^r(x,t)$ for all x
    - Including $x = 1 \rightarrow$ control reference $u^r(1,t)$
  - Combine this FF results with FB control law
    - Stabilize the trajectory $u^r(x,t)$
    - Force $u(0,t) = u^r(0,t)$

Trajectory Generation

We present the ideas through several examples.

Consider the heat equation

$$\begin{aligned} u_t &= u_{xx} \\ u_x(0) &= 0 \\ u(0) &= \text{system output} \\ u(1) &= \text{system input} \end{aligned}$$

The objective is to follow the **output** trajectory

$$u^r(0,t) = 2-(t-1)^2 = 1+2t-t^2$$

The first step is to construct the full state trajectory $u^r(x,t)$ which satisfies the PDE. Then add a feedback law that stabilizes that solution.

Motion Planning for PDEs

We postulate the state trajectory in the form:

$$u^r(x,t) = \sum_{k=0}^{\infty} a_k(t) \frac{x^k}{k!}.$$

This is a Taylor series in x with time varying coefficients $a_k(t)$ that will be determined from the PDE, boundary condition, and desired trajectory.

From the desired trajectory we get

$$u^r(0,t) = a_0(t) = 1 + 2t - t^2.$$

The boundary condition gives

$$u_x^r(0,t) = a_1(t) = 0.$$

Substituting $u^r(x,t)$ into the PDE we get:

$$\begin{aligned}
\sum_{k=0}^{\infty} \dot{a}_k(t)\frac{x^k}{k!} &= \frac{\partial^2}{\partial x^2}\sum_{k=0}^{\infty} a_k(t)\frac{x^k}{k!} \\
&= \sum_{k=2}^{\infty} a_k(t)\frac{k(k-1)x^{k-2}}{k!} \\
&= \sum_{k=2}^{\infty} a_k(t)\frac{x^{k-2}}{(k-2)!} \\
&= \sum_{k=0}^{\infty} a_{k+2}(t)\frac{x^k}{k!}.
\end{aligned}$$

We get the recursive relationship

$$a_{k+2}(t) = \dot{a}_k(t)$$

The recursion yields

$$\begin{array}{rclll} a_0 &=& 1+2t-t^2, & a_1 = 0 & \\ a_2 &=& 2-2t, & a_3 = 0 & \\ a_4 &=& -2, & a_5 = 0 & \\ a_6 &=& 0, & a_i = 0 & \text{for } i > 6. \end{array}$$

This gives the reference trajectory

$$u^r(x,t) = 1 + 2t + t^2 + (1-t)x^2 - \frac{1}{12}x^4,$$

and the input signal

$$u^r(1,t) = \frac{23}{12} + t - t^2.$$

Remark. Perfect trajectory obtained only if the initial condition of the plant agrees with the initial condition of the trajectory, that is, $u(x,0) = 1 + x^2 - \frac{1}{12}x^4$.

Example 2. Reaction-diffusion equation

$$\begin{aligned} u_t &= u_{xx} + \lambda u \\ u_x(0) &= 0 \end{aligned}$$

Desired output reference signal

$$u^r(0,t) = \mathrm{e}^{\alpha t}.$$

From the boundary condition we have

$$a_1(t) = 0,$$

and from the PDE

$$a_{k+2}(t) = \dot{a}_k(t) + \lambda a_k(t).$$

These conditions give

$$\begin{aligned} a_{2k+1} &= 0 \\ a_{2k+2} &= \dot{a}_{2k} - \lambda a_{2k} \\ a_2 &= (\alpha - \lambda)\mathrm{e}^{\alpha t} \\ a_4 &= (\alpha - \lambda)^2 \mathrm{e}^{\alpha t} \\ a_{2k} &= (\alpha - \lambda)^k \mathrm{e}^{\alpha t} \end{aligned}$$

Motion Planning for PDEs

The state trajectory is

$$
\begin{aligned}
u^r(x,t) &= \sum_{k=0}^{\infty} (\alpha-\lambda)^k \mathrm{e}^{\alpha t} \frac{x^{2k}}{2k!} \\
&= \mathrm{e}^{\alpha t} \sum_{k=0}^{\infty} \frac{(\sqrt{\alpha-\lambda}x)^{2k}}{2k!} \\
&= \mathrm{e}^{\alpha t} \begin{cases} \cosh(\sqrt{\alpha-\lambda}x) & \alpha \geq \lambda \\ \cos(\sqrt{\alpha-\lambda}x) & \alpha < \lambda \end{cases}
\end{aligned}
$$

The reference input is

$$
u^r(1,t) = \mathrm{e}^{\alpha t} \cosh(\sqrt{\alpha-\lambda}).
$$

Useful formulae when calculating trajectories for sinusoidal outputs:

$$
\begin{aligned}
\cosh(a) &= \sum_{k=0}^{\infty} \frac{a^{2k}}{(2k)!}, \qquad \sinh(a) = \sum_{k=0}^{\infty} \frac{a^{2k+1}}{(2k+1)!} \\
\cosh(ja) &= \cos(a), \qquad \sinh(ja) = j\sin(a), \qquad \sin(ja) = j\sinh(a).
\end{aligned}
$$

Example 8. First order hyperbolic PDE

$$\begin{aligned} u_t &= u_x + g u(0) \\ u^r(1,t) &= \text{control} \\ u^r(0,t) &= \sin(\omega t) = \operatorname{Im}\left\{ \mathrm{e}^{j\omega t} \right\} = a_0(t) \end{aligned}$$

Since this is a first order PDE, the only boundary condition is the one that is available for control.

From the PDE we get

$$\begin{aligned} a_1 &= \dot{a}_0 - g a_0 \\ &= \operatorname{Im}\left\{ (j\omega - g)\mathrm{e}^{j\omega t} \right\} \end{aligned}$$

$$\begin{aligned} a_{i+1} &= \dot{a}_i \\ a_k(t) &= \operatorname{Im}\left\{ (j\omega - g)(j\omega)^{k-1} \mathrm{e}^{j\omega t} \right\} \\ &= \operatorname{Im}\left\{ \left(1 - \frac{g}{j\omega}\right) (j\omega)^k \mathrm{e}^{j\omega t} \right\} \end{aligned}$$

Motion Planning for PDEs

State reference

$$
\begin{aligned}
u^r(x,t) &= \mathrm{Im}\left\{\left[1+\left(1-\frac{g}{j\omega}\right)\sum_{k=1}^{\infty}\frac{(j\omega x)^k}{k!}\right]\mathrm{e}^{j\omega t}\right\} \\
& \quad \text{add and subtract } \frac{g}{j\omega} \\
&= \mathrm{Im}\left\{\left[\frac{g}{j\omega}+\left(1-\frac{g}{j\omega}\right)\mathrm{e}^{j\omega x}\right]\mathrm{e}^{j\omega t}\right\} \\
&= \mathrm{Im}\left\{\frac{g}{j\omega}\mathrm{e}^{j\omega t}+\frac{j\omega-g}{j\omega}\mathrm{e}^{j\omega(t+x)}\right\}
\end{aligned}
$$

$$
u^r(x,t) = -\frac{g}{\omega}[\cos(\omega t)-\cos(\omega(t+x))]+\sin(\omega(t+x))
$$

Input reference

$$
u^r(1,t) = \frac{g}{\omega}[\cos(\omega(t+1))-\cos(\omega t)]+\sin(\omega(t+1)).
$$

Trajectory Tracking

Trajectory Tracking = stabilization of reference trajectory with feedback control.

Consider the previous example

$$\begin{aligned} \text{plant:} \quad u_t &= u_x + g u(0) \\ \text{trajectory:} \quad u^r(x,t) &= \frac{g}{\omega}[\cos(\omega(t+x)) - \cos(\omega t)] + \sin(\omega(t+x)) \end{aligned}$$

Stabilizing controller

$$u(1,t) - u^r(1,t) = \int_0^1 k(1,y)[u(y,t) - u^r(y,t)]dy$$

$$\begin{aligned} \text{transformation:} \quad w(x,t) &= u(x,t) - u^r(x,t) - \textstyle\int_0^x k(x,y)[u(y,t) - u^r(y,t)]dy \\ \text{kernel:} \quad k(x,y) &= -g e^{g(x-y)} \\ \text{target system:} \quad w_t &= w_x \qquad \text{(unit delay with zero input)} \\ w(1) &= 0 \end{aligned}$$

Motion Planning for PDEs

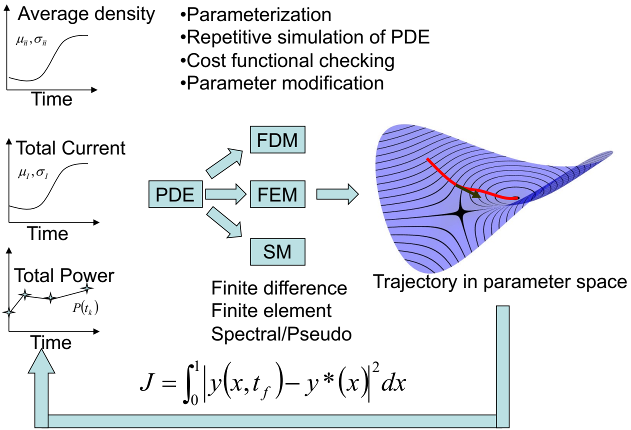
Stabilizing controller

$$
\begin{aligned}
u(1,t) - u^r(1,t) &= \int_0^1 k(1,y)[u(y,t) - u^r(y,t)]dy \\
u(1,t) &= u^r(1,t) + \int_0^1 k(1,y)[u(y,t) - u^r(y,t)]dy \\
&= \underbrace{u^r(1,t) - \int_0^1 k(1,y)u^r(y,t)dy}_{\text{feedforward (fcn of } t\text{)}} + \underbrace{\int_0^1 k(1,y)u(y,t)dy}_{\text{feedback}}
\end{aligned}
$$

$$
\begin{aligned}
\text{Ffwd} &= \frac{g}{\omega}[\cos(\omega(t+1)) - \cos(\omega t)] + \sin(\omega(t+1)) \\
&\quad + \int_0^1 g e^{g(1-y)} \left\{ \frac{g}{\omega}[\cos(\omega(t+y)) - \cos(\omega t)] + \sin(\omega(t+x)) \right\} dy \\
&= \frac{g}{\omega}[\cos(\omega(t+1)) - \cos(\omega t)] + \sin(\omega(t+1)) - \frac{g}{\omega}[\cos(\omega(t+1)) - \cos(\omega t)] \\
&= \sin(\omega(t+1))
\end{aligned}
$$

which is $u^r(0,t) = \sin(\omega t)$ advanced by one time unit!

Thus, it suffices to determine the reference trajectory for the target system (rather than for the complicated original system). This is true in general.



V. M. Becerra, "Solving Optimal Control Problems With State Constraints Using Nonlinear Programming and Simulation Tools," *IEEE Transactions on Education*, vol. 47, no. 3, August 2004.

Motion Planning for PDEs

W.P. Wehner, E. Schuster, M.D. Boyer, F. Poli, "TRANSP-based Optimization for Tokamak Scenario Development," *Fusion Engineering and Design*, 146 (2019) 547–550.

W.P. Wehner, J.E. Barton, E. Schuster, C.T. Holcomb, T.C. Luce, J.R. Ferron, M.L. Walker, D.A. Humphreys, B.G. Penaflor and R.D. Johnson, "Optimal Current Profile Control for Enhanced Repeatability of L-mode and H-mode Discharges in DIII-D," *Fusion Engineering and Design*, 123 (2017) 513-517.

H. Wang and E. Schuster, "Model-based Optimal Scenario Planning in EAST," *Fusion Engineering and Design*, 123 (2017) 569-573.

C. Xu, Y. Ou, E. Schuster and X. Yu, "Computing open-loop optimal control of the q-profile in ramp-up tokamak plasmas using the minimal-surface theory," *Plasma Science and Technology*, vol. 15, no. 5, pp. 403-410, 2013.

C. Xu, Y. Ou, J. Dalessio, E. Schuster, T.C. Luce, J.R. Ferron, M.L. Walker and D.A. Humphreys, "Ramp-Up-Phase Current-Profile Control of Tokamak Plasmas via Nonlinear Programming," IEEE Transactions on Plasma Science, vol. 38, no 2, pp. 163-173, February 2010.

Y. Ou, C. Xu, E. Schuster, T.C. Luce, J.R. Ferron, M.L. Walker and D.A. Humphreys, "Design and simulation of extremum-seeking open-loop optimal control of current profile in the DIII-D tokamak," *Plasma Physics and Controlled Fusion*, 50 (2008) 115001.

Motion Planning for PDEs

J. Barton, W. Shi, E. Schuster, T.C. Luce, J.R. Ferron, M.L. Walker, D.A. Humphreys, F. Turco, R.D. Johnson and B.G. Penaflor, "Nonlinear Physics-model-based Actuator Trajectory Optimization for Advanced Scenario Planning in the DIII-D Tokamak," Proceedings of the *19th IFAC World Congress*, Cape Town, South Africa, August 24-29, 2014.

C. Xu, E. Schuster, "Control of Ramp-Up Current Profile Dynamics in Tokamak Plasmas via the Minimal-Surface Theory", Proceedings of the *2009 IEEE Conference on Decision and Control*, Shanghai, China, December 16-18, 2009.

C. Xu, R. Arastoo and E. Schuster, "On Iterative Learning Control of Parabolic Distributed Parameter Systems," Proceedings of the *IEEE Mediterranean Conference on Automation and Control*, Greece, June 2009.

C. Xu, J. Dalessio, Y. Ou, E. Schuster, T.C. Luce, J. Ferron, M. Walker, D. Humphreys, T. Casper, W. Meyer, "POD-Based Optimal Control of Current Profile in Tokamak Plasmas via Nonlinear Programming," Proceedings of the *2008 American Control Conference*, Seattle, Washington, June 11-13, 2008.

Y. Ou, C. Xu, E. Schuster, T. Luce, J. R. Ferron, M. Walker, "Extremum-Seeking Finite-Time Optimal Control of Plasma Current Profile at the DIII-D Tokamak," in *Proceedings of the 2007 American Control Conference*, New York, New York, July 11-13, 2007.

Feedforward (FF) Control Optimization Problem Defined as Tradeoff Between Multiple Objectives

- Control Synthesis: feedforward controller, feedback controllers, state observers.

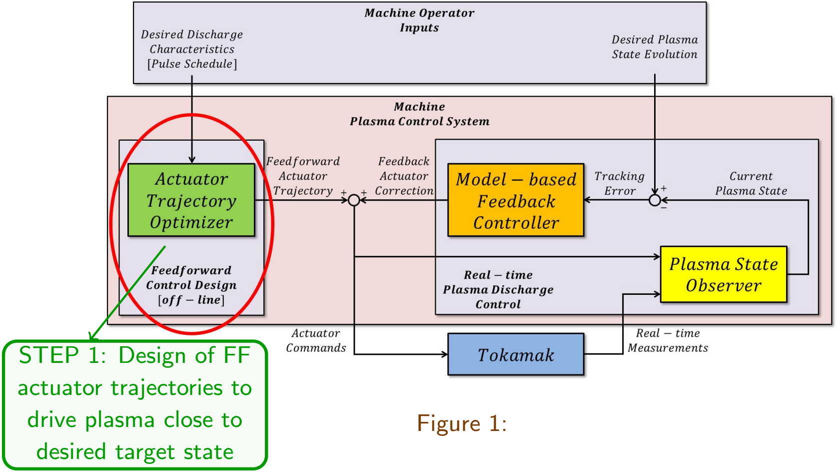


Figure 1:

Feedforward (FF) Control Optimization Problem Defined as Tradeoff Between Multiple Objectives

- **Objective:** Reach **target plasma state** at some time t_{targ} by designing **actuator waveforms** subject to **plasma dynamics** and **constraints**.
  - Defines nonlinear, constrained optimization problem.
- **Target state:** Defined in terms of q profile and β_N.
  - Proximity of achieved state to target formulated into cost functional (J).
  - To-be-minimized cost function is weighted sum of three objectives

$$J = k_q J_q(q(t_{\text{targ}})) + k_{ss} J_{ss}(\dot{\psi}(t_{\text{targ}})) + k_{\beta_N} J_{\beta_N}(\beta_N(t_{\text{targ}}))$$

  - Target q-profile objective

$$J_q(q(t_{\text{targ}})) = \int_0^1 W_q(\hat{\rho}) \left[q_{\text{targ}}(\hat{\rho}) - q(\hat{\rho}, t_{\text{targ}})\right]^2 d\hat{\rho}$$

  - Stationarity objective

$$J_{ss}(\dot{\psi}(t_{\text{targ}})) = \int_0^1 W_{ss}(\hat{\rho}) \left[g_{ss}(\hat{\rho}, t_{\text{targ}})\right]^2 d\hat{\rho}$$

$$g_{ss}(\hat{\rho}, t) = -2\pi \frac{\partial U_p}{\partial \hat{\rho}} \qquad U_p = \frac{\partial \psi}{\partial t} = \text{loop voltage}$$

  - Target β_N objective

$$J_{\beta_N}(\beta_N(t_{\text{targ}})) = \left[\beta_{N_{\text{targ}}} - \beta_N(t_{\text{targ}})\right]^2$$

Solution of Feedforward (FF) Control Optimization Problem Requires Minimization under Constraints

- An open-loop (feedforward) control policy u_{FF} is obtained via nonlinear optimization subject to constraints. Returns target trajectory ψ_{FF} (q_{FF}).

$$\min_{\boldsymbol{\alpha}} \quad J(\psi_{FF}, \dot{\psi}_{FF}, \beta_N) \quad \Big\} \text{ Cost Function (Optimization Objective)}$$

$$\text{s.t.} \quad \dot{\psi}_{FF} = f_\psi(\psi_{FF}, u_{\text{FF}}), \psi_{FF}(t_0) = \psi_0 \quad \Big\} \text{MDE}$$

$$\dot{W}_{FF} = f_W(W_{FF}, u_{\text{FF}}) \quad \Big\} \text{ Energy Balance}$$

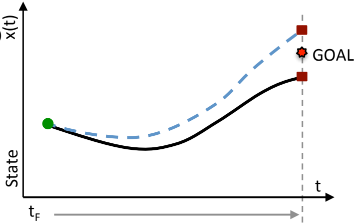
$$\beta_N(t) \le \beta_{N_{\text{max}}}, q \ge 1 \quad \Big\} \text{ State Constraint, MHD Stability Limit}$$

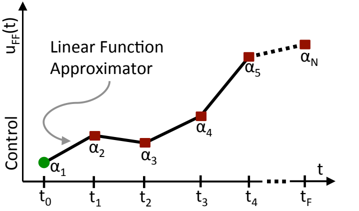
$$\bar{n}_e|_{20} \le I_{\text{p}}/\pi a^2 \quad \Big\} \text{ State Constraint, Density Limit}$$

$$P_{\text{tot}}(t) \ge P_{\text{tot}_{\text{min}}} \quad \Big\} \text{ State Constraint, Prevent H} \rightarrow \text{L Transition}$$

$$u_{\text{FF}}(t) \in \mathcal{U} \quad \Big\} \text{ Input Constraint, Saturation / Rate Limit}$$

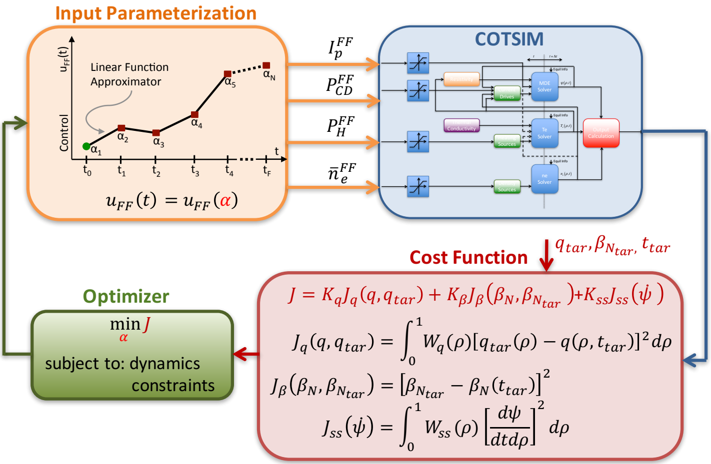
$$u_{\text{FF}}(t) = u_{\text{FF}}(\boldsymbol{\alpha}) \quad \Big\} \text{ Linear Function Approximator}$$





Solution of Feedforward (FF) Control Optimization Problem Requires Minimization under Constraints

- **COTSIM enables systematic model-based scenario planning based on nonlinear constrained optimization with arbitrary cost function.**



- **FF Design (offline): Arbitrary model complexity (COTSIM$\rightarrow$TRANSP)**