

# Reacting Flows and Control Theory\*

S. H. Lam<sup>†</sup>

Department of Mechanical and Aerospace Engineering  
Princeton University

January 28, 2008

## Abstract

Model reduction of reacting flows looks for “slow manifolds”—by exploiting the fast/slow speed gap of the reaction system—so that some of the original ODEs can be replaced by algebraic equations. Control theory strives to determine some unknown control forces in a set of given ODEs in order to honor some user-specified behaviors—such as restricting the solutions to stay on some user-specified phase space surface (i.e. manifold). It is shown that some model reduction methodologies in reacting flows can be applied to control theory.

## Introduction

Reacting flows with active chemistry and control theory are distinct disciplines. They have mathematical similarities, and some of the similarities can be exploited.

## Reacting flow problems

Consider a generic initial value problem:

$$\frac{d\mathbf{y}}{dt} = \mathbf{g}(\mathbf{y}), \quad (1a)$$

---

\*To be presented at the *Twelfth SIAM Conference on Numerical Combustion*, Monterey, CA, March 31-April 2, 2008.

<sup>†</sup>Professor emeritus. Email: lam@princeton.edu.

$$\mathbf{y}(0) = \mathbf{y}_o, \quad (1b)$$

where  $\mathbf{y}$  and  $\mathbf{g}$  are  $N$ -dimensional column vectors. For reacting flow problems,  $\mathbf{g}(\mathbf{y})$  is usually the sum of contributions from many elementary chemical reactions—some very fast, and others not so fast. For the present purpose,  $\mathbf{g}(\mathbf{y})$  is just any *given* differentiable function of  $\mathbf{y}$  and may be arbitrarily non-linear. The relevant Jacobian ( $\partial\mathbf{g}/\partial\mathbf{y}$ ) is denoted by  $\mathbf{J}$ —note that  $\mathbf{J}$  is  $\mathbf{y}$ -dependent when  $\mathbf{g}(\mathbf{y})$  is non-linear.

Now consider the case when the eigenvalues of  $\mathbf{J}$  have a wide gap, and that there are  $M$  “fast” modes ( $M < N$ ) which are all known to be decaying modes. After the  $M$  fast modes decay away, the solution will then evolve in a *slow manifold* defined by:

$$f^m(\mathbf{y}) \approx 0, \quad 1 \leq m \leq M, \quad (2)$$

for  $t \gg \tau_M$  where  $\tau_M$  is the time scale (reciprocal of the magnitude of the real part of the eigenvalue) of the slowest fast mode. When solutions in the slow evolution period are of interest, *finding the slow manifold—Eq.(2)—is a worthy objective*, because knowledge of the slow manifold can be exploited to provide various insights and useful simplifications.

## CSP on reacting flow problems

Let  $\mathbf{a}_n$  and  $\mathbf{b}^n$ ,  $n = 1, \dots, N$  be a full set of orthonormal (column and row) basis vectors spanning the full  $N$ -dimensional space. The right hand side of Eq.(1a) can always be expressed in terms of any full set of basis vectors:

$$\mathbf{g}(\mathbf{y}) = \sum_{n=1}^N \mathbf{a}_n f^n, \quad (3a)$$

where

$$f^n(\mathbf{y}) = \mathbf{b}^n \odot \mathbf{g}, \quad n = 1, \dots, N, \quad (3b)$$

and  $\odot$  is the  $N$ -dimensional inner product. For linear problems, the right and left eigenvectors of  $\mathbf{J}$  are excellent choices for  $\mathbf{a}_n$  and  $\mathbf{b}^n$ . If the eigenmodes are ordered with the fastest modes first, then after the first  $M$  fast modes have sufficiently decayed, the solution will subsequently evolve in the slow manifold, Eq.(2). For non-linear problems, CSP provides a two-step *refinement* algorithm to *improve* the fast/slow decoupling of any trial basis vectors set [1, 2]. Each CSP refinement cycle improves the *quality* of the fast/slow decoupling—i.e. the *smallness* of the  $M$   $f^m(\mathbf{y})$ 's—by a factor proportional to the eigenvalue gap between the fast and slow eigenmodes.

## Control problems

Consider the following dynamics system:

$$\frac{d\mathbf{x}}{dt} = \mathbf{A}(\mathbf{x}) + \mathbf{B}\mathbf{u}, \quad (4)$$

where  $\mathbf{x}$  and  $\mathbf{A}$  are  $N$ -dimensional column vectors,  $\mathbf{A}(\mathbf{x})$  is some *given* differentiable function of  $\mathbf{y}$  and may be arbitrarily non-linear,  $\mathbf{u}$  is a  $M$ -dimensional column vector ( $M \leq N$ ) representing the to-be-determined “control forces,” and  $\mathbf{B}$  is a  $N \times M$  known constant matrix. A typical *control objective* would be to find  $\mathbf{u}(t)$  (or  $\mathbf{u}(\mathbf{x})$ ) such that the solution  $\mathbf{x}(t)$ —after a brief transient—would evolve with acceptably small user-specified “tracking errors” i.e., the controlled  $\mathbf{x}(t)$  trajectories would stay inside a manifold defined by:

$$F^m(\mathbf{x}) \approx 0, \quad 1 \leq m \leq M, \quad (5)$$

where the  $F^m(\mathbf{x})$ 's are  $M$  *user-specified* algebraic tracking errors from some desired trajectories. See Eq.(11) later for more general control objectives.

A control force program  $\mathbf{u}(t)$  is called *open-loop* control while a control law  $\mathbf{u}(\mathbf{x}_*)$  is called *close-loop* control, where  $\mathbf{x}_*$  denotes sensor measurement of the actual true  $\mathbf{x}$  to provide “feedback” to the controller. Asterisks shall be used to identify sensor measurements provided to the controller. It is assumed that good sensor measurements for all components of  $\mathbf{x}$  are available.

### Restating control problems as reacting flow problems

For the sake of simplicity (and to sidestep many complications), this exposition shall be limited to the special case of  $M = 1$ . So  $F^1(\mathbf{x})$  is the sole user-specified algebraic function, and  $\mathbf{u}$  is a scalar to be denoted by  $u$ . For this special case, the control problem can be restated in the form of reacting flow problems (with  $N + 1$  species):

$$\mathbf{y} = \begin{bmatrix} \mathbf{x} \\ u \end{bmatrix}, \quad \mathbf{g}(\mathbf{y}) = \begin{bmatrix} \mathbf{A}(\mathbf{x}) + \mathbf{B}u \\ U(\mathbf{x}_*, u) \end{bmatrix}, \quad f^1(\mathbf{y}_*) = F^1(\mathbf{x}_*). \quad (6)$$

Thus the species  $u$  has the new “kinetics” equation:

$$\frac{du}{dt} = U(\mathbf{x}_*, u). \quad (7)$$

The problem now is to find  $U(\mathbf{x}_*, u)$  such that the tracking errors (as reported by the sensors)—starting with  $f^1(\mathbf{y}_*) = F^1(\mathbf{x}_*) \neq 0$  initially—would decay toward  $f^1(\mathbf{y}_*) = F^1(\mathbf{x}_*) \approx 0$  in some desirable manner after a brief transient.

## Fast reaction as *universal dynamic control law*

To achieve the control objective  $F^1(\mathbf{x}_*) \approx 0$  for  $t \gg \tau$ , the following  $U(\mathbf{x}_*, u)$  is recommended [3, 4] for Eq.(7) :

$$U(\mathbf{x}_*, u) = -\frac{1}{K\Delta t} \left( \frac{dF^1(\mathbf{x}_*)}{dt} + \frac{F^1(\mathbf{x}_*)}{\tau} \right) \quad (8a)$$

where

$$K = K(\mathbf{x}_*) \equiv \left( \frac{\partial F^1}{\partial \mathbf{x}} \odot \mathbf{B} \right)_* , \quad (8b)$$

and  $K(\mathbf{x}_*) \neq 0$  is assumed. Here,  $\tau$  is an user-specified time constant of the “brief transient,” and  $\Delta t$  is another user-specified positive time constant which must be smaller than  $\tau$ . *Both are at the disposal of the user.* While  $U(\mathbf{x}_*, u)$  in Eq.(7) can be interpreted as the chemical kinetics term for the chemical species  $u$ , it is actually a close loop *dynamic control law* for the control force  $u$ . From the reacting flows point of view, the QSSA (quasi-steady-state approximation) is applicable to the species  $u$  **only if** (i) Eq.(7) is stable **and** (ii)  $U(\mathbf{x}_*, u)$ ’s “reaction rate” is “asymptotically large.” Application of QSSA on Eq.(7) would yield an exponentially decaying  $F^1(\mathbf{x}_*)$  toward zero (for  $t \gg \Delta t$ )—with (user-specified) time constant  $\tau$ .

*The crucial question is then: how to ensure that QSSA is applicable to  $u$  (i.e. to Eq.(7) with Eq.(8a)) in the small  $\Delta t$  limit?*

How does  $U(\mathbf{x}_*, u)$  depend on  $u$ ? Neither  $F^1(\mathbf{x}_*)/\tau$  nor  $K$  depends on  $u$ , but  $(dF^1(\mathbf{x})/dt)_*$  does. Using straightforward mathematics, one obtains:

$$\begin{aligned} \left( \frac{dF^1(\mathbf{x})}{dt} \right)_* &= \frac{\partial F^1}{\partial \mathbf{x}_*} \odot \frac{d\mathbf{x}_*}{dt} \\ &= \frac{\partial F^1}{\partial \mathbf{x}_*} \odot \overbrace{(\mathbf{A}(\mathbf{x}_*) + \mathbf{B}u + O(\delta))}^{d\mathbf{x}_*/dt} \\ &= Ku + \frac{\partial F^1}{\partial \mathbf{x}_*} \odot (\mathbf{A}(\mathbf{x}_*) + O(\delta)), \end{aligned} \quad (9)$$

where  $O(\delta)$  represents the measurement errors of  $d\mathbf{x}_*/dt$  and they are assumed small (with zero mean). Thus, Eq.(8a) and Eq.(9) say that  $U(\mathbf{x}_*, u)$  contains—through  $d\mathbf{x}_*/dt$  provided by sensor measurements—a first order chemical reaction for species  $u$ , and its forward reaction rate (which is at the disposal of the user) is inversely proportional to  $\Delta t$ . It is now totally

straightforward to show that QSSA is indeed a valid approximation for  $u$  in the small  $\Delta t$  limit. Thus  $\Delta t$  should be as small as possible.

To achieve control objective Eq.(5) with  $M = 1$  for dynamic system Eq.(4), the recommended close loop dynamic control law is:

$$\frac{du}{dt} = -\frac{1}{K\Delta t} \left( \frac{\partial F^1}{\partial \mathbf{x}_*} \odot \frac{d\mathbf{x}_*}{dt} + \frac{F^1(\mathbf{x}_*)}{\tau} \right). \quad (10)$$

It is assumed that good quality sensor measurements of both  $\mathbf{x}_*(t)$  and  $d\mathbf{x}_*/dt$  are available to the controller to evaluate the right hand side. The initial condition for  $u(0)$  is quite irrelevant—any convenient value will do.

In actual implementation, the real world integrates Eq.(4)—using real world physics—while the “controller” (which is a microprocessor-based black box) integrates Eq.(10) numerically in real time—assuming that good quality real time sensor measurements of both  $\mathbf{x}_*$  and  $d\mathbf{x}_*/dt$  are available. Note that the precise numerical values of (small but finite)  $\Delta t$  used is irrelevant to the  $F^1(\mathbf{x}_*) \approx 0$  control objective (for  $t \gg \tau$ ). The smallness of  $\Delta t$  is limited by practical hardware considerations of the finite turn-around time of sensor measurements and the software computational time inside the black box.

## Can you control $\mathbf{x}$ if you don't know $\mathbf{A}(\mathbf{x})$ ?

It is important to note that this dynamics close loop control law Eq.(10) needs no information whatever on  $\mathbf{A}(\mathbf{x})$ , the open-loop dynamics of the original system—provided real time sensor measurements of both  $\mathbf{x}_*$  and  $(d\mathbf{x}/dt)_*$  are available. It does need some information on the column vector  $\mathbf{B}$ —its role here is merely to determine the sign of  $K$ . It is very easy to do computer simulations with this dynamics control law to confirm numerically that the *same* control law (for one specified control objective) can be used for *any* reasonable (unknown)  $\mathbf{A}(\mathbf{x})$ —its nonlinearity is quite irrelevant so long as its (open loop) time scale is much larger than the very small  $\Delta t$ . Hence Eq.(10) is a robust *universal dynamic control law*.

The exposition above was for the special control objective:  $F^1(\mathbf{x}) \approx 0$  after  $t \gg \tau$ . Generalization to the more general control objective:

$$\frac{dF^1}{dt} + H(F^1, \mathbf{x}; t) \approx 0, \quad t \gg \Delta t, \quad (11)$$

is completely straightforward—where  $H(F^1, \mathbf{x}; t)$  is free to be chosen by the users. Just replace  $F^1(\mathbf{x}_*)/\tau$  in Eq.(10) by the desired  $H(F^1, \mathbf{x}_*; t)$ .

## Optimal control

It is well known that optimal control theory usually generates two-point boundary value problems which are more cumbersome to solve than initial value problems. The methodology discussed here can be used to deal with the two-point boundary value issues, and was found successful [4] for a particular simple class of optimal control problems.

## Concluding remarks

When  $M \geq 1$ ,  $\mathbf{B}$  is a  $N \times M$  matrix, and there are  $M$  control objectives in Eq.(5). The important matrix for control theory is then the  $M \times M$  matrix  $\mathbf{K}$  defined by:

$$\mathbf{K} \equiv \frac{\partial F^m}{\partial \mathbf{x}} \odot \mathbf{B}, \quad m = 1, \dots, M. \quad (12)$$

When  $\mathbf{K}$  is non-singular the control problem is said to have unity *relative degree* [5]—it was so assumed in the  $M = 1$  case presented earlier. For problems with higher relative degrees (including the  $M = 1$  case with  $K = 0$ ), sensor measurements of higher time derivatives of  $\mathbf{x}$  are needed.

## References

- [1] Lam, S. H., Using CSP to Understand Complex Chemical Kinetics. *Combustion Science and Technology*, **89**, 5-6, pp.375-404, 1993.
- [2] Lam, S. H. and Goussis, D. A., The CSP Method for Simplifying Kinetics. *International Journal of Chemical Kinetics*, **26**, pp.461-486, 1994.
- [3] Lam, S. H., A Robust Universal Controller. Presented at the *International Mechanical Engineering Congress and Exposition*, Dallas, Texas, November 16-21, 1997.
- [4] Lam, S. H., A Robust Controller for Scalar, Autonomous Optimal Control Problems. Presented at the 1993 *American Control Conference*, San Diego, CA., June 2-4, 1999.
- [5] Isidori, A., *Nonlinear Control Systems: An Introduction*. Springer-Verlag, 1985.

All Lam references are available at <http://www.princeton.edu/~lam>