Identification of linear models for closed-loop flow control

Juan Guzmán Iñigo and Denis Sipp

ONERA DAFE
Model-based control

- For « model-based » closed-loop control, need of ROM capturing the input-output dynamics of flow systems
Data-based / Model-based ROMs

- Models may be obtained
  - Either by projecting a model on a basis (Galerkin Projection) => model-based ROMs
  - Or by processing input/output data => data-based ROMs

- This talk focuses on data-based ROMs
Boundary layer flow subject to Tollmien-Schlichting instabilities

\[ \partial_t v + V \cdot \nabla v + v \cdot \nabla V = -\nabla p + \nu \Delta v + w(t) \hat{f}(x, y), \nabla \cdot v = 0 \]
Closed-loop control setup

Boundary layer flow subject to Tollmien-Schlichting instabilities

Estimation sensor $s$  
Actuator $u$  
Performance sensor $m$
1/ Construct reduced order model between all inputs (noise, actuator) and all outputs (estimation sensor, performance sensor)
\[
\frac{dx}{dt} = AX + B_u u + B_w w, \quad s = C_s X, \quad m = C_m X
\]
Inputs: \((u, w)\), Outputs: \((s, m)\)

2/ Noise source term \(w\) is considered as external noise

3/ Dynamic observer takes advantage of estimation sensor \(s\) to reconstruct state.
\[
\frac{dY}{dt} = AY + B_u u - L(s - C_s Y), \quad m = C_m Y
\]
Inputs: \((u, s)\), Output: \((m)\)
Data-based models

- Linear relationship between \((u, s)\) and \((m)\)

- Either: go through the state if \((m)\) is the entire state

- Or: eliminate the state if \((m)\) is low-dimensional (wall-sensor)

\[
m(t) = \int_0^t C_m e^{(A+LC_s)(t-\tau)} B_u u(\tau) d\tau + \int_0^t -C_m e^{(A+LC_s)(t-\tau)} Ls(\tau) d\tau
\]

Low-dimensional temporal convolution kernels \(h_{mu}(t-\tau)\) replaces low-high-low dimensional expression \(C_m e^{(A+LC_s)(t-\tau)} B_u\).
Outline

1/ ROM with wall-sensor as output
2/ ROM with full velocity-field as output
3/ Conclusions and perspectives
Eliminate the state

Backward-facing step flow \( \text{Re}=500 \)
Open-loop simulations
Input-output dynamics

- Input-output dynamics fully characterized by impulse response (Markov parameters),
  \[ \begin{align*}
  u(0) &= 1, \quad u(k \geq 1) = 0 \\
  H_k &= m(k)
  \end{align*} \]

- Convolution:
  \[ m(k) = \sum_{j=0}^{\mu-1} H_j u(k - j) \]

- Data-based ROM: Find best \( H_j \)'s to fit input-output dataset.
Least-squares problem

- Parameters
  - Size of data sequence: \( N \)
  - Number of identified parameters (model order): \( n \)

\[
\begin{bmatrix}
m(n - 1) \\
m(n) \\
\vdots \\
m(N - 1)
\end{bmatrix}_M = \begin{bmatrix}
u_{n-1} & u_{n-2} & \cdots & u_0 \\
u_n & u_{n-1} & \cdots & u_1 \\
\vdots & \vdots & \ddots & \vdots \\
u_{(N-1)} & u_{(N-2)} & \cdots & u_{N-n}
\end{bmatrix}_\Phi \begin{bmatrix}
H_0 \\
H_1 \\
\vdots \\
H_{n-1}
\end{bmatrix}_H = \begin{bmatrix}
\end{bmatrix}_L
\]

- \( N - n + 1 \) equations for \( n \) unknowns with \( N - n + 1 \gg n \)
- Minimum of \( J(H) = \|M - \Phi H\|_2^2 \)
- Least squares problem solved with pseudo-inverse
- \( H = \Phi^\dagger M \) with \( \Phi^\dagger = (\Phi^T \Phi + \epsilon I)^{-1} \Phi^T \)
How to choose model order $n$?

- Akaike criterion: $\text{AIC} = \log J(H) + \frac{2n}{N} \frac{\text{Bias error}}{\text{Variance error}}$
- A posteriori criterion
- Choose model order to minimize AIC
How to decrease the number of model parameters $n$?

1/ Introduce auto-regressive (AR) term: $m(k - j) + \sum_{j=1}^{n_m} a_j m(k - j)$
2/ Introduce delays: $d_u$ and $d_s$

\[
m(k - j) + \sum_{j=1}^{n_m} a_j m(k - j) = \sum_{j=d_u}^{d_u+n_u} b^u_j u(k - j) + \sum_{j=d_s}^{d_s+n_s} b^s_j s(k - j) + e(k)\]

Residual of auto-regression
How to decrease the number of model parameters $n$?

Autocorrelation of $m$:
$R_{mm}(\Delta t) = \langle m(t)m(t + \Delta t) \rangle$
$\Rightarrow n_m$

Cross-correlation between $u$ and $m$:
$R_{um}(\Delta t) = \langle u(t)m(t + \Delta t) \rangle$
$\Rightarrow (d_u, n_u)$
Performance of ROM for estimation

ARX FIT=72%
How to deal with non-observable source noises?

Introduce moving average (MA) term to account for coloured noise:

\[ m(k - j) + \sum_{j=1}^{n_s} a_j m(k - j) = \sum_{j=d_u}^{d_u+n_u} b^u_j u(k - j) + \sum_{j=d_s}^{d_s+n_s} b^s_j s(k - j) + E(k) \]

\( E(k) = \sum_{j=1}^{n_e} c_j e(k - j) + e(k) \)

ARMAX FIT=96%
How to deal with non-observable source noises?

Additional non-observable (by s) noise source

Corrupted Learning dataset

Presence of additional external forcing source: No (%) Yes (%)

ARMAX fit 96 82
ARX fit 72 33
Control law

In frequency domain:

\[ m = H_{mu} u + H_{ms} s \]

\[ m = 0 \Rightarrow u = -H_{mu}^\perp H_{ms} s \]

\[ H_{mu}^\perp = (H_{mu}^T H_{mu} + \varepsilon I)^{-1} H_{mu}^T \]

\( \varepsilon \) used to tune control effort
Control performance in DNS
Outline

1/ ROM with wall-sensor as output

2/ ROM with full velocity-field as output

3/ Conclusions and perspectives
Performance sensor \( m \) placed at \( x = 800 \) (far from the actuator)
Performance sensor \( m \) placed at \( x = 260 \) (slightly downstream from the actuator)
\[ v(x, y, n) \equiv \text{Perturbation velocity field} \]

\[ m(n) = C_m \cdot v(n), \quad m(n) = 0 \]

1st case: \[ v(n) = 0 \]
2nd case: \[ C_m \perp v(n) \]

m(n) is a projection of the velocity field
• A more general control framework is introduced by targeting the perturbation kinetic energy

$$E(n) = \int_{\Omega} v(x, y, n) \cdot v(x, y, n) dV$$

• In this case, we want to obtain $v = f(s, u)$
• We replace the performance sensor \( (m) \) by the velocity field \( (v) \)
• The velocity field can be obtained in an experimental setup using PIV measurements
• The new objective is to obtain the velocity field \( v(n) \) from \( u(n) \) and \( s(n) \)

\[ v = f(s, u) \]
We replace the performance sensor (m) by the velocity field (v).

The velocity field can be obtained in an experimental setup using PIV measurements.

The new objective is to obtain the velocity field \( v(n) \) from \( u(n) \) and \( s(n) \), 

\[ v = f(s, u) \]

The velocity field is highly dimensional \( O(v) \approx 10^6 \).

We need to reduce the number of degrees of freedom.
Model order reduction

- A POD basis is computed in order to reduce the degrees of freedom of the velocity field $v$

$$v'(x, y, n) \approx \sum_{i=1}^{k} x_i(n) \phi_i(x, y)$$

- The POD decomposition guarantees that the error $\|v - v'\|^2$ is minimal

![Mode 1, $\phi_1$](image1.png)

![Mode 10, $\phi_{10}$](image2.png)
Reduced-order equations

- POD coefficients
  \[ x_i(n) = \langle v(n), \varphi_i \rangle, \]
  Velocity field \quad POD mode

- Reduced state
  \[ X(n) = (x_1, x_2, \ldots, x_N)^T, \]

- Reduced-order equations
  \[ X(n+1) = AX(n) + Ls(n) + B_u u(n) \]
Reduced-order equations

- POD coefficients
  \[ x_i(n) = \langle v(n), \varphi_i \rangle, \]
  Velocity field \quad \text{POD mode}

- Reduced state
  \[ X(n) = (x_1, x_2, \ldots, x_N)^T, \]

- Reduced-order equations
  \[ X(n+1) = AX(n) + Ls(n) + B_u u(n) \]
  \[ \text{• Matrices } A, L, B \text{ are the unknown coefficients} \]
We consider **subspace identification algorithms** to extract the system matrices \((A, L, B)\).

These techniques consider the general problem of computing the matrices \((A, B, C, D)\) from inputs-outputs datasets of the system

\[
x(n + 1) = Ax(n) + Bu(n) + w(n),
\]

\[
y(n) = Cx(n) + Du(n) + v(n).
\]
Subspace identification algorithms

- We consider **subspace identification algorithms** to extract the system matrices \((A, L, B)\).
- These techniques consider the general problem of computing the matrices \((A,B,C,D)\) from inputs-outputs dataset of the system

\[
\begin{align*}
  x(n+1) &= Ax(n) + Bu(n) + w(n), \\
  y(n) &= Cx(n) + Du(n) + v(n).
\end{align*}
\]
Subspace identification algorithms

- We consider **subspace identification algorithms** to extract the system matrices (A, L, B).
- These techniques consider the general problem of computing the matrices (A,B,C,D) from inputs-outputs dataset of the system

\[
\begin{align*}
x(n + 1) &= Ax(n) + Bu(n) + w(n), \\
y(n) &= Cx(n) + Du(n) + v(n).
\end{align*}
\]

- They are well-suited for **MIMO systems**
- Only one model parameter \( \rightarrow \) the order of the system

See

or
Estimation performance

• Need of a global variable to quantify the performance

\[ E(n) = \int_{\Omega} v(x, y) \cdot v(x, y) dV = \sum_{i=1}^{N_{pod}} x_i^2(n) \]

Learning dataset (90 modes)  FIT = 96.75%
Estimation performance

- Assessment of estimation capabilities using validation dataset

FIT = 89.5%
Model parameters

• The only model parameter to select is the number of POD modes
• The only model parameter to select is the number of POD modes

• 20 POD modes capture 99% of the energy
• However, 50 modes are required to obtain an accurate model

Observability of the sensor
Sensor observability

- Sensor $s$ must be accurately represented by the basis

$$s(n) = C_s X(n)$$

- Lack of observability of POD basis upstream
Find a control law \( u = -KX \) which minimizes the functional

\[
J = \int_0^\infty \left( X^* Q X + l u^2 \right) dt,
\]

where \( Q \) permits to select a control target (in this case \( Q=I \) to target the kinetic energy) and \( l \) weights the output and the control effort.

This problem implies the resolution of a Ricatti equation.
Control performance
Linear Stochastic Estimation (LSE)

• The estimation performance \( \nu = f(s) \) of the proposed model suggests a comparison with other estimation techniques used in fluid dynamics.

• The most widely used estimation technique is Linear Stochastic Estimation (LSE)

\[
X_{LSE}(n) = RS(n),
\]

where \( S(n) \) is a vector containing the measurement from several sensors and \( R \) is a matrix obtained by minimizing \( \|X^2 - X_{LSE}^2\| \).
Linear Stochastic Estimation (LSE)

- The estimation performance \( (v = f(s)) \) of the proposed model suggests a comparison with other estimation techniques used in fluid dynamics.
- The most widely used estimation technique is **Linear Stochastic Estimation (LSE)**.

\[
X_{LSE}(n) = RS(n),
\]

where \( S(n) \) is a vector containing the measurement from several sensors and \( R \) is a matrix obtained by minimizing \( \|X^2 - X_{LSE}^2\| \).

- LSE does not keep any temporal “memory”
- The lack of temporal information is palliated using several sensors.
• LSE works correctly when the velocity field is highly correlated in space (big spatial structures).

• In convective flows, two points situated in different streamwise positions are strongly correlated in time but not in space (due to the convection). Consequently, a great number of sensors will be required to obtain an accurate estimation.

• The dynamic model obtain an accurate estimation only from one sensor due to the temporal “memory”
• For the present configuration, an LSE estimator is obtained using 17 sensors equispaced between $x = 150-950$
1/ ROM with wall-sensor as output

2/ ROM with full velocity-field as output

3/ Conclusions and perspectives
Conclusion

• Valid for transitional (small amplitude perturbations), convectively unstable flows
• Data-based ROMs to deal with unknown input \( w(t) \) (replacement by upstream sensor \( s(t) \))
• Output: wall-sensor
  • Models: ARX, ARMAX
  • Algorithms for identification: Pseudo-inverse, ARMAX
  • Controller: inversion of transfer-function
• Output: full velocity-field
  • Model: State-Space
  • Algorithm: Subspace
  • Controller: LQR
Perspectives

• Dealing with non-linearities

• In experiment: **Feed-forward control of a perturbed backward-facing step flow, Gautier & Aider JFM 2014**

• Robust performance
Thank you for your attention!