River Flow Control using the Hayami Model

Tarek Rabbani

May 04, 2007
ME 236 – Professor Bayen
Project Purpose

- Irrigation systems waste too much water, especially during low water periods, when transported from the river to the crop fields.
- Identify a control strategy to convey the right amount of water
  - Open loop control
    - Differential flatness
  - Closed loop control
    - State-Space Model
Formulation of the Control Problem
Formulation of the Control Problem

\[ \partial_t Q + C \partial_x Q = D \partial_{xx} Q \]
\[ \partial_x Q (0, t) = u (t) \]
\[ \partial_x Q (L, t) = 0 \]
\[ Q (x, 0) = Q_0 (x) \]
\[ y (t) = Q (L, t) \]

Given a flow at the reach of the river, the control input is to be determined to produce the desired output.
Using Differential flatness

\[ Q = \sum_{k=0}^{\infty} a_k(t) \frac{(x - L)^k}{k!} \]

\[ \sum_{k=0}^{\infty} (x - L)^k [a'_{k+1} + Ca_{k+1} - Da_{k+2}] = 0 \]

\[ a_{k+2} = \frac{C}{D} a_{k+1} + \frac{1}{D} a'_{k+1} \quad k \geq 0 \]

\[ a_k = \frac{C}{D} a_{k-1} + \frac{1}{D} a'_{k-2} \quad k \geq 2 \]
Formulation of the Control Problem-Open Loop

- Verification of the results using Flatness-based Control of a Nonlinear Parabolic Equation Modeling of a Tubular Reactor.

\[ \partial_t C + \nu \partial_x C = \partial_{x,x} C \]
\[ \partial_x C (0, t) = 0 \]
\[ C (-1, t) = u (t) \]
\[ C (x, 0) = \psi (x) \]
\[ C (x, t) = \sum_{k=0}^{\infty} a_k (t) \frac{(x)^k}{k!} \]
\[ a_k = a'_{k-2} + \nu a_{k-1} \]

\[ \partial_T Q + C \partial_x Q = D \partial_{x,x} Q \]
\[ \partial_x Q (0, T) = u (T) \]
\[ \partial_x Q (L, T) = 0 \]
\[ Q (x, 0) = Q_0 (x) \]
\[ Q = \sum_{k=0}^{\infty} A_k (t) \frac{(X - L)^k}{k!} \]
\[ A_k = \frac{C}{D} A_{k-1} + \frac{1}{D} A'_{k-2} \]
Formulation of the Control Problem - Open Loop

Apply the transformation

\[
\begin{pmatrix}
    x \\
    t \\
    c
\end{pmatrix} = \begin{pmatrix}
    \frac{1}{L} & 0 & 0 \\
    0 & \frac{D}{L^2} & 0 \\
    0 & 0 & 1
\end{pmatrix} \begin{pmatrix}
    X \\
    T \\
    Q
\end{pmatrix}
\]

To get:

\[ a_k = A_k L^k \]

\[ a^{(n)}_k = \frac{L^k}{D} L^{(n)}_k \]

with \( \gamma = \frac{C L}{D} \)

Substitute in:

\[ a_{k+2} = a^{'}_k + \gamma a_{k+1} \]

To get:

\[ L^{k+2} A_{k+2} = \frac{L^k}{D} A^{'}_k + \frac{C L}{D} L^{k+1} A_{k+1} \]

\[ A_{k+2} = \frac{C}{D} A_{k+1} + \frac{1}{D} A^{'}_k \]

Same Result
Convergence of the series:

\[ Q = \sum_{k=0}^{\infty} a_k(t) \frac{(x - L)^k}{k!} \]

The formal series converges if \( y : R \rightarrow R \) is a Gevrey function of class \( \alpha \leq 2 \);

i.e. a \( C^\infty \) function which satisfies: \( \sup_{t \in R} \left| y^{(1)}(t) \right| \leq \frac{m!^\alpha}{\gamma^1} \) for \( 1 \geq 0 \) and \( \alpha \leq 2 \)

Proof:

1. Use the triangular inequality to show that \( \sup_{t \in R} \left| y^{(1)}(t) \right| \leq \frac{m!^\alpha}{\gamma^1} \) implies \( \sup_{t \in R} \left| a_k^{(1)}(t) \right| \leq \frac{m M^k (1 + k)!^\alpha}{\gamma^1 k!^{\alpha - 1}} \)

2. Apply the Cauchy - Hadamard Formula to compute the Radius of Convergence.

Result

The Radius of Convergence is \( R \geq \frac{1}{M} \) where \( M \) is the largest solution of \( \frac{1}{D} \frac{1}{\gamma M^2} + \frac{C}{D} \frac{1}{2 M} = 1 \)

or \( R \geq \frac{4}{C/D + \sqrt{16/\gamma D + (C/D)^2}} \)

Since the river has length of \( L \), we require an \( L \) radius of convergence. Hence,

\[ 2 \geq \frac{2 L^2}{D \gamma} + \frac{C L}{D} \quad \alpha \leq 2 \]

Infinite if \( \alpha < 2 \)
Simulation of the Diffusive Wave equation.

Solution at the end of simulation:
Q(x,END)

Actual Flow
Desired Flow

Control Effort u(t)

Time t
Distance x
Numerical solution of the Diffusive Wave equation.

Simulation-80 t=
Simulation - Unmatched initial conditions
Simulation - Low approximation (10 terms)

Solution at the end of simulation

Q(x, END)

Control Effort u(t)

Numerical solution of the Diffusive Wave equation.
Closed Loop Control—X. Litrico

\[ y(s) = F_{Hayami}(s)u(s) \]

\[ F_{Hayami}(s) = e^{\frac{\Theta e - \sqrt{\Theta^2 e + 4E e s}}{2E e}} X \]

\[ F_{2r}(s) = \frac{e^{-\tau s}}{1 + s^2 + Ps^2} \]

\[
\begin{cases}
\dot{\delta x}(t) = A(\lambda)\delta x(t) + B(\lambda)\delta u(t) \\
\delta y(t) = C(\lambda)\delta x(t - \tau(\lambda))
\end{cases}
\]

with \( A(\lambda) = \begin{pmatrix} \frac{S(\lambda)}{P(\lambda)} & -\frac{1}{P(\lambda)} \\ 1 & 0 \end{pmatrix}, \quad B(\lambda) = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \quad C(\lambda) = \begin{pmatrix} 0 \\ 1 \end{pmatrix} \) and \( \lambda = Q_e \).
Close loop Control – State Space Model using Differential Flatness

Think of the recursion $a_{k+2} = \frac{C}{D} a_{k+1} + \frac{1}{D} a'_k$ as a set of nonlinear ODEs:

- $a'[0] = D \cdot a[2] - C \cdot a[1]$

Due to boundary conditions, $a'[1] = a[1] = 0 \rightarrow a[3] = \frac{C}{D} a[2]$. Moreover, all odd number series can be expressed in terms of even ones.

For example,

- $\frac{C}{D} \cdot (a'[2]) = D \cdot a[5] - C \cdot a[4]$
- $\frac{C}{D} \cdot (D \cdot a[4] - C \cdot a[3]) = D \cdot a[5] - C \cdot a[4]$

Thus,

$$a[5] = \frac{\left( \frac{C}{D} \cdot (D \cdot a[4] - C \cdot \frac{C}{D} a[2]) + C \cdot a[4] \right)}{D} = -\frac{C^3 a[2]}{D^3} + \frac{2Ca[4]}{D}$$
Closed Loop Control – 2 Dimensional State Space

\[ a'[0] = D \ast a[2] \]
\[ a'[2] = \frac{-C^2}{D} \ast a[2] + D \ast a[4] \]
\[ u(t) = \sum_{k=0}^{3} a_{k+1}(t) \frac{(-L)^k}{k!} = a[4] \frac{(-L)^3}{3!} + a[1] \frac{(-L)^0}{0!} + a[2] \frac{(-L)^1}{1!} + a[3] \frac{(-L)^2}{2!} \]
\[ a[4] = \frac{3!}{(-L)^3} u(t) - \frac{3!}{(-L)^3} \left( a[1] \frac{(-L)^0}{0!} + a[2] \frac{(-L)^1}{1!} + \frac{C}{D} a[2] \frac{(-L)^2}{2!} \right) \]
\[ a[4] = -\frac{3!}{L^3} u(t) - \frac{3!}{L^2} a[2] - \frac{3!}{(-L)^3} \frac{C}{D} a[2] \frac{(-L)^2}{2!} \]
\[ a'[2] = \frac{-C^2}{D} \ast a[2] + D \ast \left( \frac{-3!}{L^3} u(t) - \frac{3!}{L^2} a[2] - \frac{3!}{(-L)^3} \frac{C}{D} a[2] \frac{(-L)^2}{2!} \right) \]
Closed Loop Control – 2 Dimensional State Space

\[ a'[0] = D \times a[2] \]

\[ a'[2] = \left( \frac{-C^2}{D} - \frac{6D}{L^2} + \frac{3C}{L} \right) \times a[2] - 6 \frac{D}{L^3} u \]

\[
\begin{pmatrix}
    a[0] \\
    a[2]
\end{pmatrix}
= \begin{pmatrix}
    0 & D \\
    0 & \left( \frac{-C^2}{D} - \frac{6D}{L^2} + \frac{3C}{L} \right)
\end{pmatrix}
\begin{pmatrix}
    a[0] \\
    a[2]
\end{pmatrix}
+ \begin{pmatrix}
    0 \\
    -6 \frac{D}{L^3}
\end{pmatrix} u
\]

\[ y = a[0] = (1 \ 0 \ 0) \begin{pmatrix}
    a[0] \\
    a[2]
\end{pmatrix} \]
Closed Loop Control – N Dimensional State Space

Determining the A, B, and C matrices for an N-state space boils down to determining a transformation between the odd and even series. i.e.

\[
\begin{pmatrix}
a[1] \\
a[3] \\
\vdots \\
a[2N - 1]
\end{pmatrix}
= T
\begin{pmatrix}
a[0] \\
a[1] \\
\vdots \\
a[2N - 2]
\end{pmatrix}
\]
Closed Loop Control – N Dimensional State Space

In order to find N series, we need N-1 explicit iterations and sum minor manipulations.

The iteration is given by:

\[
A_{i+1} = [I_{N \times N} - e_i^T e_{i+1}] A_i + e_i^T e_i A_i^2
\]

where 
\[
A_i = \begin{bmatrix}
1 & 1 & 0 & \ldots & 0 \\
0 & 1 & 1 & \ldots & 0 \\
\vdots & 1 & \ddots & \vdots \\
\vdots & \ddots & 1 \\
0 & 0 & 0 & \ldots & 1
\end{bmatrix}
\]

\[
e_i = [0 \ \cdots \ e(i) = 1 \ 0 \ \cdots \ 0]
\]
Closed Loop Control – N Dimensional State Space

```matlab
>> [A B C] = buildABC(2)

A =

[ 0, 0; -1/L^2 * (-3 * DD * L * C + 6 * DD^2 * Z + C * Z^2) / DD ]

B =

0
-6 * DD / L^3

C =

1   C
```
Closed Loop Control – N Dimensional State Space

Solution at the end of simulation

- N=2
- N=3
- N=4
- N=5
- N=6
- Exact

Q(x, END) vs. time t
## Comparison of Flatness based State Space with other methods

<table>
<thead>
<tr>
<th>Flatness Bases State Space</th>
<th>Others (X. Litrico)</th>
</tr>
</thead>
<tbody>
<tr>
<td>One approximation Used</td>
<td>More than one approximation</td>
</tr>
<tr>
<td>Observable, Controllable, Full State linearizable</td>
<td>There are no guarantees for any of the properties</td>
</tr>
<tr>
<td>Has information capable of constructing the whole flow, in space and time.</td>
<td>Just contains information of the reach flow</td>
</tr>
<tr>
<td>More accurate, easier to construct, can be used for other partial differential equations with negative coefficients</td>
<td>Limited to the diffusive wave equation with positive coefficients, needs hand calculations. i.e. can’t be automated</td>
</tr>
</tbody>
</table>
Conclusions

- The open-loop control has to be truncated at a certain integer, due to this fact and in view of disturbances and model errors, a closed loop control is needed.

- State space model using flatness can allow observers design, inspection of state, feedback stabilization.

- The discovered series for the transformation matrix allows the generation of an N symbolic state space equations.

- It is possible to extend the solution for the diffusive wave equation to others, like that of the tubular reactor.