Automatic generation of optimal synthesis for membrane filtration systems

Rémy Dutto [1,2] Jérôme Harmand [1] Alain Rapaport [2]

[1]LBE INRAE, Narbonne

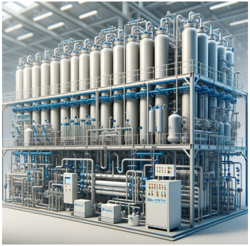
[2]MISTEA INRAE, Montpellier

Journées de Statistique et Optimisation en Occitanie (JS2O), Perpignan

Introduction

We are interested in :

- membrane filtration systems
- optimal synthesis
- automatic generation



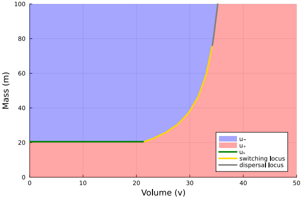


Table of Contents

Membrane filtration systems

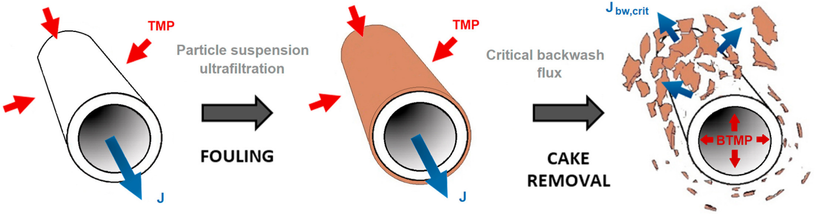


Figure: Extracted from [Vroman et al., 2021]

State/cost :

- m : mass of cake layer
- v : produced volume
- e : energy spend

Control :

- $u = 1$: filtration
- $u = -1$: backwash

Dynamics modeling

The dynamics for filtration and backwash mode are assumed to be given respectively by

- $m_f(m)$ and $m_b(m)$ for $\dot{m}$ (speed of variation of m),
- $v_f(m)$ and $v_b(m)$ for $\dot{v}$ (effective flow rate),
- $e_f(m)$ and $e_b(m)$ for $\dot{e}$ (instantaneous energy consumption).

Considering $u \in [-1, 1]$, the dynamics are modelled by

$$\begin{cases} \dot{m} &= \dfrac{1+u}{2} m_f(m) + \dfrac{1-u}{2} m_b(m), \\ \dot{v} &= \dfrac{1+u}{2} v_f(m) + \dfrac{1-u}{2} v_b(m), \\ \dot{e} &= \dfrac{1+u}{2} e_f(m) + \dfrac{1-u}{2} e_b(m). \end{cases}$$

Dynamics modeling

Denoting

$$\begin{aligned}
m_+(m) &= \frac{1}{2}\big(m_f(m) - m_b(m)\big), & m_-(m) &= \frac{1}{2}\big(m_f(m) + m_b(m)\big),\\
v_+(m) &= \frac{1}{2}\big(v_f(m) - v_b(m)\big), & v_-(m) &= \frac{1}{2}\big(v_f(m) + v_b(m)\big),\\
e_+(m) &= \frac{1}{2}\big(e_f(m) - e_b(m)\big), & e_-(m) &= \frac{1}{2}\big(e_f(m) + e_b(m)\big),
\end{aligned}$$

the dynamic of the system is

$$\begin{cases}
\dot{m} = u\, m_+(m) + m_-(m)\\
\dot{v} = u\, v_+(m) + v_-(m)\\
\dot{e} = u\, e_+(m) + e_-(m)
\end{cases}$$

Case #1 : Maximum volume

The goal is to maximise the filtered volume on a fixed time interval $[t_0, T]$:

$$(\#1) \qquad \begin{cases} \max\limits_{m,u} \displaystyle\int_{t_0}^{T} u(t)\, v_+(m(t)) + v_-(m(t))\, \mathrm{d}t, \\ \text{s.c. } \dot{m}(t) = u(t)\, m_+(m(t)) + m_-(m(t)), \\ \qquad u(t) \in [-1, 1], \quad t \in [t_0, T], \\ \qquad m(t_0) = m_0 \geq 0. \end{cases}$$

Case #2 : Minimum energy

The goal is to minimise the energy to provide a desired volume of filtered water v_f :

$$
(\#2) \quad \begin{cases}
\min\limits_{m,v,u,t_f} \displaystyle\int_{t_0}^{t_f} u(t)\, e_+(m(t)) + e_-(m(t))\, \mathrm{d}t, \\
\text{s.c. } \dot{m}(t) = u(t)\, m_+(m(t)) + m_-(m(t)), \\
\qquad \dot{v}(t) = u(t)\, v_+(m(t)) + v_-(m(t)), \\
\qquad u(t) \in [-1, 1], \quad t \in [t_0, t_f], \\
\qquad m(t_0) = m_0, \quad v(t_0) = 0, \quad v(t_f) = v_f .
\end{cases}
$$

Equivalent formulation

Let us consider the following general formulation

$$
\text{(OCP)} \quad \left\{
\begin{array}{l}
\displaystyle \min_{x,u,t_f} \int_{t_0}^{t_f} u(t)\, f_+^0(x_1(t)) + f_-^0(x_1(t))\, \mathrm{d}t, \\
\text{s.c. } \dot{x}_1(t) = u(t)\, f_+^1(x_1(t)) + f_-^1(x_1(t)), \\
\qquad \dot{x}_2(t) = u(t)\, f_+^2(x_1(t)) + f_-^2(x_1(t)), \\
\qquad u(t) \in [-1, 1], \quad t \in [t_0, t_f], \\
\qquad x_1(t_0) = x_0, \quad x_2(t_0) = 0, \quad x_2(t_f) = x_f,
\end{array}
\right.
$$

where $x = (x_1, x_2)$.

Equivalent formulation

Problem (#2) can be written as (OCP)

$$
\text{(OCP)} \quad \begin{cases} \displaystyle \min_{x,u,t_f} \int_{t_0}^{t_f} u(t)\, e_+(x_1(t)) + e_-(x_1(t))\, \mathrm{d}t, \\ \text{s.c. } \dot{x}_1(t) = u(t)\, m_+(x_1(t)) + m_-(x_1(t)), \\ \qquad \dot{x}_2(t) = u(t)\, v_+(x_1(t)) + v_-(x_1(t)), \\ \qquad u(t) \in [-1, 1], \quad t \in [t_0, t_f], \\ \qquad x_1(t_0) = m_0, \quad x_2(t_0) = 0, \quad x_2(t_f) = v_f, \end{cases}
$$

where $x = (x_1, x_2)$.

Equivalent formulation

Problem (#1) can be written as (OCP)

$$
\text{(OCP)} \qquad \begin{cases} \displaystyle \min_{x,u,t_f} - \int_{t_0}^{t_f} u(t)\, v_+(x_1(t)) + v_-(x_1(t))\,\mathrm{d}t, \\ \text{s.c. } \dot{x}_1(t) = u(t)\, m_+(x_1(t)) + m_-(x_1(t)), \\ \qquad \dot{x}_2(t) = \cancel{u(t)\,0} + 1, \\ \qquad u(t) \in [-1, 1], \quad t \in [t_0, t_f], \\ \qquad x_1(t_0) = m_0, \quad x_2(t_0) = 0, \quad x_2(t_f) = T, \end{cases}
$$

where $x = (x_1, x_2)$.

Objectives

Provide optimal synthesis of (OCP) "whatever" inputs functions and initial/final conditions are.

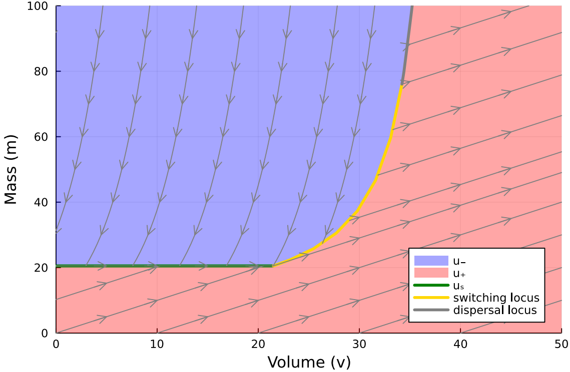


Figure: Example of optimal synthesis with trajectories

Pontryagin maximum principle

If (x, u, t_f) is a solution of (OCP), there exists a costate $p = (p_1, p_2)$ such that $p_1(t_f) = 0$, the *costate dynamic* is satisfied for almost every $t \in [t_0, t_f]$

$$\dot{p}(t) = -\frac{\partial H}{\partial x}(x(t), p(t), u(t))$$

as well as the *maximisation condition* for almost every $t \in [t_0, t_f]$

$$\max_{w \in [-1,1]} H(x(t), p(t), w) = H(x(t), p(t), u(t)) = 0$$

where H is the *hamiltonian* given by

$$H(x, p, u) = u\left(p_1 f_+^1(x_1) + p_2 f_+^2(x_1) - f_+^0(x_1)\right) + p_1 f_-^1(x_1) + p_2 f_-^2(x_1) - f_-^0(x_1)$$

Optimal control

Using the maximisation condition and the definition of H, we have

$$u(t) \begin{cases} = -1 & \text{if} \quad \phi\big(x_1(t), p(t)\big) < 0 \\ = \quad 1 & \text{if} \quad \phi\big(x_1(t), p(t)\big) > 0 \\ \in [-1, 1] & \text{if} \quad \phi\big(x_1(t), p(t)\big) = 0 \end{cases}$$

where the function ϕ is defined by

$$\phi(x_1, p) = p_1\, f_+^1(x_1) + p_2\, f_+^2(x_1) - f_+^0(x_1).$$

Lemma 1

There exists $\bar{t} \in [t_0, t_f[$ such that $u(t) = 1$ for almost every $t \in [\bar{t}, t_f]$.

Singular state and singular control

Let us suppose that there exists $I \subset [t_0, t_f]$ of non-zero mesure such that $\forall t \in I, \phi(x_1(t), p(t)) = 0$. Then we look for (x_1, p, u) such that

$$\begin{cases} \phi(x_1, p) = 0 \\ \dot{\phi}(x_1, p) = 0 \\ \ddot{\phi}(x_1, p, u) = 0 \\ H(x_1, p, u) = 0 \end{cases}$$

We can analytically have an expression of $p(x_1)$ and $u(x_1)$ such that

$$\phi(x_1, p(x_1)) = \dot{\phi}(x_1, p(x_1)) = \ddot{\phi}(x_1, p(x_1), u(x_1)) = 0$$

Singular state and control

Hypothesis 1

There exists exactly one state $x_s \in \mathbb{R}^+$ *such that*

$$H(x_s, p(x_s), u(x_s)) = 0$$

Under Hypothesis 1, we numerically find the singular state x_s by using a rootfinding method, the singular control $u_s = u(x_s)$, and the singular costate $p_s = p(x_s)$.

In Julia, we can use the **ForwardDiff** package to get the exact derivative of the inputs functions.

Optimal synthesis

We have computed the singular curve

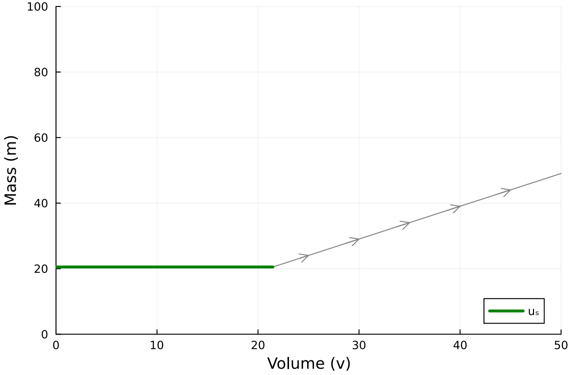


Figure: Optimal synthesis construction

Optimal synthesis

We have to compute the switching and dispersal locus curve

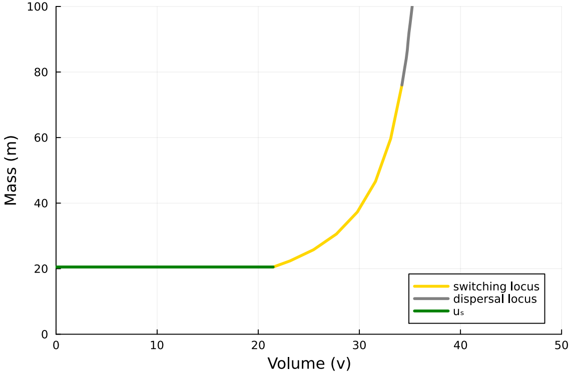


Figure: Optimal synthesis construction

Optimal synthesis

We have to compute the switching and dispersal locus curve

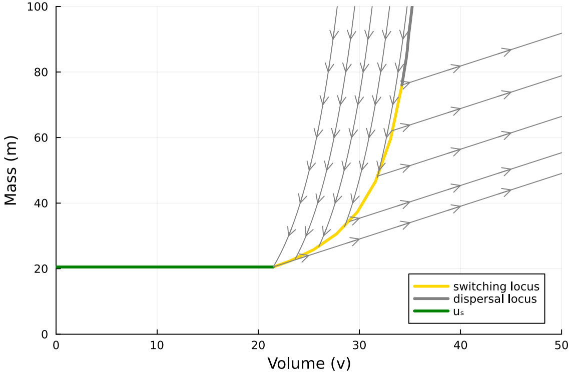


Figure: Optimal synthesis construction

Optimal synthesis

We have to compute the switching and dispersal locus curve

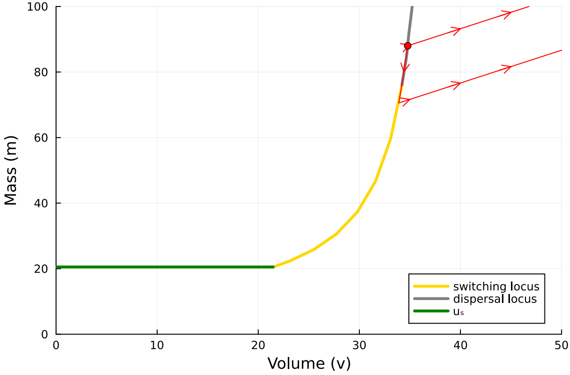


Figure: Optimal synthesis construction

Switching and dispersal curves

The switching and the dispersal curves are the solution of $S(x) = 0$, where S is a function $S\colon \mathbb{R}^2 \to \mathbb{R}, x \mapsto S(x)$.

For instance, for the dispersal curve, this function is defined by

$$S(x) = \varphi_0^+(x) - \varphi_0^{-+}(x)$$

where respectively φ_0^+ and φ_0^{-+} corresponds to the optimal cost associated to the trajectory starting from x with the control $u = +1$ (resp. $u = -1$ before hitting the switching curve, and the control $u = +1$ after).

Moreover, for both curves, we know a point (a, b) such that $S(a, b) = 0$.

Differential continuation method

Let us suppose that there exists a function $x_1(x_2)$ such that

$$S(x_1(x_2), x_2) = 0.$$

Since S is constant, we have

$$\frac{\partial S}{\partial x_1}(x_1(x_2), x_2)\, x_1'(x_2) + \frac{\partial S}{\partial x_2}(x_1(x_2, x_2)) = 0$$

Function $x_1(x_2)$ is the solution of the ODE

$$x_1'(x_2) = \left(\frac{\partial S}{\partial x_1}(x_1(x_2), x_2)\right)^{-1} \frac{\partial S}{\partial x_2}(x_1(x_2, x_2)), \quad x_1(b) = a.$$

Differential continuation method

In Julia, packages **ForwardDiff** and **OrdinaryDiffEq** work together.

- The gradient of S is computed thanks to the **ForwardDiff** package.
- Even if S contains a solution of an ODE, the derivative of S is computed properly (it uses variational equations).
- The numerical integration is stopped when a condition is satisfied by using **Callback**.

Conclusion

We can generate automatically optimal feedback map associated to Problem (OCP), used for membrame filtration systems.

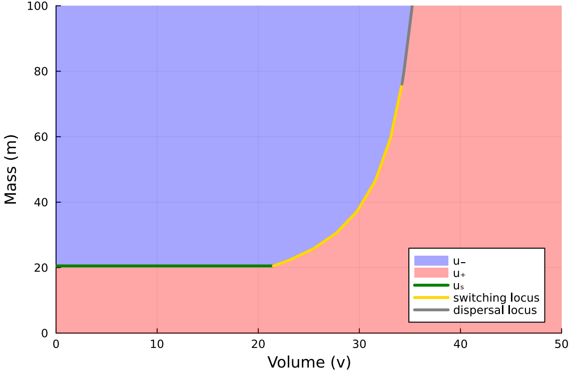


Figure: Optimal synthesis

Conclusion

We can easily go further and generate the optimal strategy classification associated to Problem (OCP).

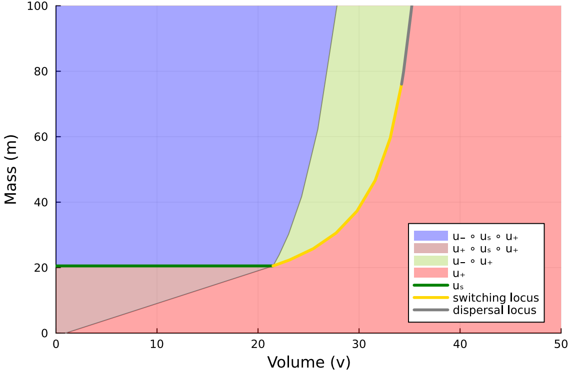


Figure: Classification of optimal strategies

References

Filtration.jl Package : Documentation and more examples

References :

Aichouche, F., Kalboussi, N., Rapaport, A., and Harmand, J. (2020).
Modeling and optimal control for production-regeneration systems - preliminary results -.
In *2020 European Control Conference (ECC)*.

Kalboussi, N., Rapaport, A., Bayen, T., Amar, N. B., Ellouze, F., and Harmand, J. (2019).
Optimal control of membrane-filtration systems.
IEEE Transactions on Automatic Control.

Vroman, T., Beaume, F., Armanges, V., Gout, E., and Remigy, J.-C. (2021).
Critical backwash flux for high backwash efficiency: Case of ultrafiltration of bentonite suspensions.
Journal of Membrane Science.