Optimal Control and Applications in Biomath

Thomas Götz
goetz@uni-koblenz.de

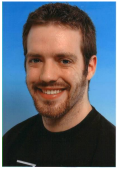
UNIVERSITÄT KOBLENZ · LANDAU

FACHBEREICH 3
MATHEMATIK | NATURWISSENSCHAFTEN

DSABNS–Workshop

Feb 2016
Evora, Portugal

...joint work of...



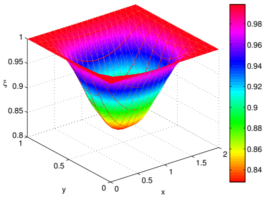
Contents

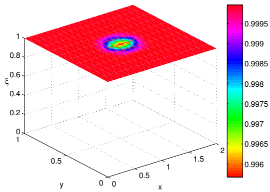
Task: Optimal Control

Real world process in nature or technology can be modeled with (partial) differential equations. However, one is often not just interested in the solution (=simulation) of the DE for a given set of parameters, but in identifying the parameters from measurements or optimizing the solution w.r.t. those parameters.

Task: Optimal Control

Real world process in nature or technology can be modeled with (partial) differential equations. However, one is often not just interested in the solution (=simulation) of the DE for a given set of parameters, but in identifying the parameters from measurements or optimizing the solution w.r.t. those parameters.





Task: Optimal Control

Real world process in nature or technology can be modeled with (partial) differential equations. However, one is often not just interested in the solution (=simulation) of the DE for a given set of parameters, but in identifying the parameters from measurements or optimizing the solution w.r.t. those parameters.

Mathematical Problem

Minimize the cost functional J on a set of admissible control parameters $u \in \mathcal{U}$ subject to the constraint that some differential equation is satisfied.

Model Problem

$$x' = f(t, x, u), \qquad x(0) = x_0$$

- Solution $x \in \mathcal{X} = H^1([0,1]) \subset C([0,1])$ (Sobolev–Lemma).

Model Problem

$$x' = f(t, x, u), \qquad x(0) = x_0$$

- Solution $x \in \mathcal{X} = H^1([0,1]) \subset C([0,1])$ (Sobolev–Lemma).
- Tracking–Problem: Find control $u \in \mathcal{U} = L^2([0,1])$ such that

$$J(x,u) = \frac{1}{2}\int_0^1 (x(t) - x_d(t))^2 + \alpha u(t)^2\, dt \longrightarrow \min$$

Model Problem

$$x' = f(t, x, u), \qquad x(0) = x_0$$

- Solution $x \in \mathcal{X} = H^1([0,1]) \subset C([0,1])$ (Sobolev–Lemma).
- Tracking–Problem: Find control $u \in \mathcal{U} = L^2([0,1])$ such that

$$J(x,u) = \frac{1}{2}\int_0^1 (x(t) - x_d(t))^2 + \alpha u(t)^2 \, dt \longrightarrow \min$$

- Desired trajectory x_d given

Model Problem

$$x' = f(t, x, u), \qquad x(0) = x_0$$

- Solution $x \in \mathcal{X} = H^1([0,1]) \subset C([0,1])$ (Sobolev–Lemma).
- Tracking–Problem: Find control $u \in \mathcal{U} = L^2([0,1])$ such that

$$J(x,u) = \frac{1}{2}\int_0^1 (x(t) - x_d(t))^2 + \alpha u(t)^2\, dt \longrightarrow \min$$

- Desired trajectory x_d given
- Regularization parameter $\alpha > 0$

Model Problem

$$x' = f(t, x, u), \qquad x(0) = x_0$$

- Solution $x \in \mathcal{X} = H^1([0,1]) \subset C([0,1])$ (Sobolev–Lemma).
- Tracking–Problem: Find control $u \in \mathcal{U} = L^2([0,1])$ such that

$$J(x,u) = \frac{1}{2}\int_0^1 (x(t) - x_d(t))^2 + \alpha u(t)^2 \, dt \longrightarrow \min$$

- Desired trajectory x_d given
- Regularization parameter $\alpha > 0$
- Minimization in ∞–dimensional spaces $\mathcal{X}$, $\mathcal{U}$

"Brute Force" Approach

First Discretize – Then Optimize

- Continuous state system $x' = f(t, x, u)$ and $x(0) = x_0$

"Brute Force" Approach

First Discretize – Then Optimize

- Continuous state system $x' = f(t, x, u)$ and $x(0) = x_0$
- Discretize time t_k, state $\boldsymbol{x} = (x_k) \in \mathbb{R}^m$, control $\boldsymbol{u} = (u_k) \in \mathbb{R}^n$

"Brute Force" Approach

First Discretize – Then Optimize

- Continuous state system $x' = f(t, x, u)$ and $x(0) = x_0$
- Discretize time t_k, state $\boldsymbol{x} = (x_k) \in \mathbb{R}^m$, control $\boldsymbol{u} = (u_k) \in \mathbb{R}^n$
- Discretize state system

$$x_{k+1} = x_k + h\varphi(t, x_k, u_k, u_{k+1}) \quad \text{or} \quad F(t, \boldsymbol{x}, \boldsymbol{u}) = 0$$

"Brute Force" Approach

First Discretize – Then Optimize

- Continuous state system $x' = f(t, x, u)$ and $x(0) = x_0$
- Discretize time t_k, state $\boldsymbol{x} = (x_k) \in \mathbb{R}^m$, control $\boldsymbol{u} = (u_k) \in \mathbb{R}^n$
- Discretize state system

$$x_{k+1} = x_k + h\varphi(t, x_k, u_k, u_{k+1}) \quad \text{or} \quad F(t, \boldsymbol{x}, \boldsymbol{u}) = 0$$

- Finite dimensional constrained minimization problem

$$\min_{\boldsymbol{u} \in \mathbb{R}^n} J(\boldsymbol{x}, \boldsymbol{u}) \quad \text{s.t.} \quad F(t, \boldsymbol{x}, \boldsymbol{u}) = 0$$

"Brute Force" Approach

First Discretize – Then Optimize

- Continuous state system $x' = f(t, x, u)$ and $x(0) = x_0$
- Discretize time t_k, state $\boldsymbol{x} = (x_k) \in \mathbb{R}^m$, control $\boldsymbol{u} = (u_k) \in \mathbb{R}^n$
- Discretize state system

$$x_{k+1} = x_k + h\varphi(t, x_k, u_k, u_{k+1}) \quad \text{or} \quad F(t, \boldsymbol{x}, \boldsymbol{u}) = 0$$

- Finite dimensional constrained minimization problem

$$\min_{\boldsymbol{u} \in \mathbb{R}^n} J(\boldsymbol{x}, \boldsymbol{u}) \quad \text{s.t.} \quad F(t, \boldsymbol{x}, \boldsymbol{u}) = 0$$

- Solve using method of your choice

"*Brute Force*" Approach

First Discretize – Then Optimize

- Continuous state system $x' = f(t, x, u)$ and $x(0) = x_0$
- Discretize time t_k, state $\mathbf{x} = (x_k) \in \mathbb{R}^m$, control $\mathbf{u} = (u_k) \in \mathbb{R}^n$
- Discretize state system

$$x_{k+1} = x_k + h\varphi(t, x_k, u_k, u_{k+1}) \quad \text{or} \quad F(t, \mathbf{x}, \mathbf{u}) = 0$$

- Finite dimensional constrained minimization problem

$$\min_{\mathbf{u} \in \mathbb{R}^n} J(\mathbf{x}, \mathbf{u}) \quad \text{s.t.} \quad F(t, \mathbf{x}, \mathbf{u}) = 0$$

- Solve using method of your choice
- ▸ Solution depends on discretization ↯

Constrained Minimization

Theorem (Lagrange–Multiplicator)

A local solution x^* of the problem

$$\min J(x) \; s.t. \; F(x) = 0$$

is a stationary point of the Lagrangian

$$\mathcal{L}(x, \lambda) := J(x) + \langle \lambda \, , \, F(x) \rangle$$

i.e.

$$\nabla \mathcal{L}(x^*, \lambda^*) = 0$$

We call λ^* the adjoint variable of x^* (Lagrange–multiplier).

Model Problem cont.

- State equation $F(x, u) := x' - f(t, x, u)$ where $x \in \mathcal{X}$, $u \in \mathcal{U}$

Model Problem cont.

- State equation $F(x,u) := x' - f(t,x,u)$ where $x \in \mathcal{X}$, $u \in \mathcal{U}$
- Multiply with test function $z \in \mathcal{Z} = \mathcal{X}'$, integrate by parts

$$\langle F\,,\,z\rangle := \int (x'-f)z = xz\Big|_0^1 - \int_0^1 xz' - \int_0^1 f(t,x,u)\cdot z = 0$$

Model Problem cont.

- State equation $F(x,u) := x' - f(t,x,u)$ where $x \in \mathcal{X}$, $u \in \mathcal{U}$
- Multiply with test function $z \in \mathcal{Z} = \mathcal{X}'$, integrate by parts

$$\langle F\,,\,z\rangle := \int (x'-f)z = xz\Big|_0^1 - \int_0^1 xz' - \int_0^1 f(t,x,u)\cdot z = 0$$

- Lagrangian

$$\begin{aligned}\mathcal{L}(x,u,z) &= \frac{1}{2}\left\|x-x_d\right\|^2 + \frac{\alpha}{2}\left\|u\right\|^2 + \langle F\,,\,z\rangle \\ &= xz\Big|_0^1 + \int_0^1 \frac{(x-x_d)^2+\alpha u^2}{2} - xz' - f(t,x,u)\cdot z\,dt\end{aligned}$$

KKT–System

$$\mathcal{L}(x, u, z) = xz\Big|_0^1 + \int_0^1 \frac{(x - x_d)^2 + \alpha u^2}{2} - xz' - f(t, x, u) \cdot z \, dt$$

KKT–System

$$\mathcal{L}(x,u,z) = xz\Big|_0^1 + \int_0^1 \frac{(x-x_d)^2 + \alpha u^2}{2} - xz' - f(t,x,u)\cdot z\,dt$$

- KKT–system: necessary condition for optimum at (x^*, u^*, z^*)

$$\nabla\mathcal{L}(x^*,u^*,z^*) = (\partial_x\mathcal{L}, \partial_u\mathcal{L}, \partial_z\mathcal{L})\,(x^*,u^*,z^*) \stackrel{!}{=} 0$$

KKT–System

$$\mathcal{L}(x,u,z) = xz\Big|_0^1 + \int_0^1 \frac{(x-x_d)^2 + \alpha u^2}{2} - xz' - f(t,x,u)\cdot z\,dt$$

- KKT–system: necessary condition for optimum at (x^*, u^*, z^*)

$$\nabla\mathcal{L}(x^*,u^*,z^*) = (\partial_x\mathcal{L}, \partial_u\mathcal{L}, \partial_z\mathcal{L})\,(x^*,u^*,z^*) \stackrel{!}{=} 0$$

- $\partial_z\mathcal{L} \stackrel{!}{=} 0$ $\qquad x' = f$ and $x(0) = x_0$

KKT–System

$$\mathcal{L}(x,u,z) = xz\Big|_0^1 + \int_0^1 \frac{(x-x_d)^2 + \alpha u^2}{2} - xz' - f(t,x,u)\cdot z\,dt$$

- KKT–system: necessary condition for optimum at (x^*, u^*, z^*)

$$\nabla\mathcal{L}(x^*,u^*,z^*) = (\partial_x\mathcal{L}, \partial_u\mathcal{L}, \partial_z\mathcal{L})(x^*,u^*,z^*) \stackrel{!}{=} 0$$

- $\partial_z\mathcal{L} \stackrel{!}{=} 0$ $\quad x' = f$ and $x(0) = x_0$
- $\partial_x\mathcal{L} \stackrel{!}{=} 0$ $\quad (x - x_d) - z' - \partial_x f \cdot z = 0$ and $z(T) = 0$

KKT–System

$$\mathcal{L}(x,u,z) = xz\Big|_0^1 + \int_0^1 \frac{(x-x_d)^2 + \alpha u^2}{2} - xz' - f(t,x,u)\cdot z\,dt$$

- KKT–system: necessary condition for optimum at (x^*, u^*, z^*)

$$\nabla\mathcal{L}(x^*,u^*,z^*) = (\partial_x\mathcal{L}, \partial_u\mathcal{L}, \partial_z\mathcal{L})\,(x^*,u^*,z^*) \stackrel{!}{=} 0$$

- $\partial_z\mathcal{L} \stackrel{!}{=} 0$ $\qquad x' = f$ and $x(0) = x_0$
- $\partial_x\mathcal{L} \stackrel{!}{=} 0$ $\qquad (x-x_d) - z' - \partial_x f\cdot z = 0$ and $z(T) = 0$
- $\partial_u\mathcal{L} \stackrel{!}{=} 0$ $\qquad \alpha u - \partial_u f\cdot z = 0$

Pontryagin Maximum (Minimum) Principle

$$x' = f(x,u) \quad x(0) = x_0, \quad u \in \mathcal{U}$$

- Solution x^*, u^*, z^* of Control Problem

$$J = \int_0^T g(x(t), u(t))\, dt \longrightarrow \min$$

leads to extremal value of *Hamiltonian* $H(z,x,u) = z \cdot f + g$.

- Costate / Adjoint z solves

$$-z' = zf_x + g_x, \quad z(T) = 0$$

Why using the L^2–Norm?

- So far: Cost function $J \simeq \|x\|_2^2 + \|u\|_2^2$

Why using the L^2–Norm?

- So far: Cost function $J \simeq \|x\|_2^2 + \|u\|_2^2$
- Why using the L^2–norm and not $\|u\|_1 = \int_0^T |u|$?

Why using the L^2–Norm?

- So far: Cost function $J \simeq \|x\|_2^2 + \|u\|_2^2$
- Why using the L^2–norm and not $\|u\|_1 = \int_0^T |u|$?

- Functional analytic answer: L_2 is Hilbert–space $\rightsquigarrow$ nice & simple

Why using the L^2–Norm?

- So far: Cost function $J \simeq \|x\|_2^2 + \|u\|_2^2$
- Why using the L^2–norm and not $\|u\|_1 = \int_0^T |u|$?

- Functional analytic answer: L_2 is Hilbert–space $\rightsquigarrow$ nice & simple
- Calculus answer: We need the derivatives J_x and J_u

Why using the L^2–Norm?

- So far: Cost function $J \simeq \|x\|_2^2 + \|u\|_2^2$
- Why using the L^2–norm and not $\|u\|_1 = \int_0^T |u|$?

- Functional analytic answer: L_2 is Hilbert–space $\rightsquigarrow$ nice & simple
- Calculus answer: We need the derivatives J_x and J_u
- Need subdifferentials $\lambda \in \partial \|x\|_{L^1}$ instead of derivatives $\nabla \|x\|_{L^2}$

$$\lambda(t) \begin{cases} = +1 & \text{where } x(t) > 0 \\ \in [-1, 1] & \text{where } x(t) = 0 \\ = -1 & \text{where } x(t) < 0 \end{cases}$$

Why using the L^2–Norm?

- So far: Cost function $J \simeq \|x\|_2^2 + \|u\|_2^2$
- Why using the L^2–norm and not $\|u\|_1 = \int_0^T |u|$?

- Functional analytic answer: L_2 is Hilbert–space $\rightsquigarrow$ nice & simple
- Calculus answer: We need the derivatives J_x and J_u
- Need subdifferentials $\lambda \in \partial \|x\|_{L^1}$ instead of derivatives $\nabla \|x\|_{L^2}$

$$\lambda(t) \begin{cases} = +1 & \text{where } x(t) > 0 \\ \in [-1, 1] & \text{where } x(t) = 0 \\ = -1 & \text{where } x(t) < 0 \end{cases}$$

- Leads to Bang–Bang–Control

Optimal Control Problem; Solution

First Discretize – Then Optimize

Optimal Control Problem

$$\min J(x,u) = \int_0^T j(x,u) = \frac{1}{2}\left(\|x - x_d\|_2^2 + \alpha \|u\|_2^2\right)$$
$$\text{s.t.} \quad x' = f(t,x,u), \quad x(0) = x_0$$

is equivalent to solving the KKT–system

$$\begin{aligned} x' &= f(t,x,u) & x(0) &= x_0 \\ z' &= j_x - \partial_x f \cdot z & z(T) &= 0 \\ 0 &= j_u - \partial_u f \cdot z & & \end{aligned}$$

How to Solve the KKT–System ?

Iterative Gradient algorithm

1. Solve $x' = f(t, x^{(k)}, u^{(k)})$, $x(0) = x_0$ for $x^{(k+1)}$

How to Solve the KKT–System ?

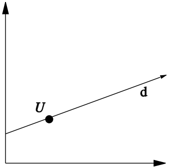
Iterative Gradient algorithm

1. Solve $x' = f(t, x^{(k)}, u^{(k)}),\ x(0) = x_0$ for $x^{(k+1)}$
2. Solve $z' = \partial_x j(x^{(k)}, u^{(k)}) - z\partial_x f(t, x^{(k)}, u^{(k)}),\ z(T) = 0$ for $z^{(k+1)}$

How to Solve the KKT–System ?

Iterative Gradient algorithm

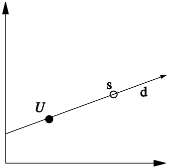
1. Solve $x' = f(t, x^{(k)}, u^{(k)})$, $x(0) = x_0$ for $x^{(k+1)}$
2. Solve $z' = \partial_x j(x^{(k)}, u^{(k)}) - z\partial_x f(t, x^{(k)}, u^{(k)})$, $z(T) = 0$ for $z^{(k+1)}$
3. Gradient $d^{(k+1)} = \nabla J = \partial_u j(x^{(k+1)}, u^{(k)}) - z^{(k)}\partial_u f(t, x^{(k+1)}, u^{(k)})$



How to Solve the KKT–System ?

Iterative Gradient algorithm

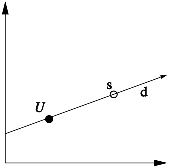
1. Solve $x' = f(t, x^{(k)}, u^{(k)}),\ x(0) = x_0$ for $x^{(k+1)}$
2. Solve $z' = \partial_x j(x^{(k)}, u^{(k)}) - z\partial_x f(t, x^{(k)}, u^{(k)}),\ z(T) = 0$ for $z^{(k+1)}$
3. Gradient $d^{(k+1)} = \nabla J = \partial_u j(x^{(k+1)}, u^{(k)}) - z^{(k)}\partial_u f(t, x^{(k+1)}, u^{(k)})$
4. Update $u^{(k+1)} = u^{(k)} - sd^{(k+1)})$ with step size $s > 0$ appropriate



How to Solve the KKT–System ?

Iterative Gradient algorithm

1. Solve $x' = f(t, x^{(k)}, u^{(k)})$, $x(0) = x_0$ for $x^{(k+1)}$
2. Solve $z' = \partial_x j(x^{(k)}, u^{(k)}) - z\partial_x f(t, x^{(k)}, u^{(k)})$, $z(T) = 0$ for $z^{(k+1)}$
3. Gradient $d^{(k+1)} = \nabla J = \partial_u j(x^{(k+1)}, u^{(k)}) - z^{(k)}\partial_u f(t, x^{(k+1)}, u^{(k)})$
4. Update $u^{(k+1)} = u^{(k)} - sd^{(k+1)})$ with step size $s > 0$ appropriate
5. Iterate, until $\|\nabla J\| < \varepsilon$



Approaches for Step Size Selection

$$\hat{J}(u) := J(x(u), u)$$

- Optimal: 1d–search: $s_{opt} = \operatorname{argmin}_{\delta > 0} \hat{J}\left(u^{(k)} - \delta d^{(k+1)}\right)$

Approaches for Step Size Selection

$$\hat{J}(u) := J(x(u), u)$$

- Optimal: 1d–search: $s_{opt} = \text{argmin}_{\delta>0}\, \hat{J}\left(u^{(k)} - \delta d^{(k+1)}\right)$
- Simple: Fixed step size $s = s_0$

Approaches for Step Size Selection

$$\hat{J}(u) := J(x(u), u)$$

- Optimal: 1d–search: $s_{opt} = \text{argmin}_{\delta>0}\, \hat{J}\left(u^{(k)} - \delta d^{(k+1)}\right)$
- Simple: Fixed step size $s = s_0$
- Better: Armijo–rule with back–tracking $s = s_0 \beta^k$

Approaches for Step Size Selection

$$\hat{J}(u) := J(x(u), u)$$

- Optimal: $1d$–search: $s_{opt} = \text{argmin}_{\delta>0} \hat{J}\left(u^{(k)} - \delta d^{(k+1)}\right)$
- Simple: Fixed step size $s = s_0$
- Better: Armijo–rule with back–tracking $s = s_0 \beta^k$
- Even better: Polynomial estimate

Approaches for Step Size Selection

$$\hat{J}(u) := J(x(u), u)$$

- Optimal: 1d–search: $s_{opt} = \text{argmin}_{\delta > 0} \hat{J}\left(u^{(k)} - \delta d^{(k+1)}\right)$
- Simple: Fixed step size $s = s_0$
- Better: Armijo–rule with back–tracking $s = s_0 \beta^k$
- Even better: Polynomial estimate
  - Given $u^{(k)}$, $\hat{J}'(u^{(k)})$

Approaches for Step Size Selection

$$\hat{J}(u) := J(x(u), u)$$

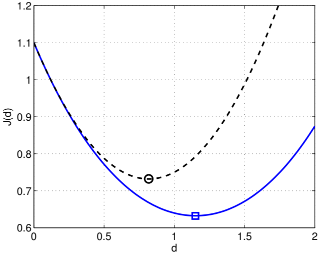
- Optimal: 1d–search: $s_{opt} = \text{argmin}_{\delta>0} \hat{J}\left(u^{(k)} - \delta d^{(k+1)}\right)$
- Simple: Fixed step size $s = s_0$
- Better: Armijo–rule with back–tracking $s = s_0 \beta^k$
- Even better: Polynomial estimate
  - Given $u^{(k)}$, $\hat{J}'(u^{(k)})$
  - Compute $\hat{J}(u^{(k)} - s_1 d^{(k+1)})$

Approaches for Step Size Selection

$$\hat{J}(u) := J(x(u), u)$$

- Optimal: 1d–search: $s_{opt} = \text{argmin}_{\delta>0} \hat{J}\left(u^{(k)} - \delta d^{(k+1)}\right)$
- Simple: Fixed step size $s = s_0$
- Better: Armijo–rule with back–tracking $s = s_0 \beta^k$
- Even better: Polynomial estimate
  - Given $u^{(k)}$, $\hat{J}'(u^{(k)})$
  - Compute $\hat{J}(u^{(k)} - s_1 d^{(k+1)})$
  - Polynomial approximation

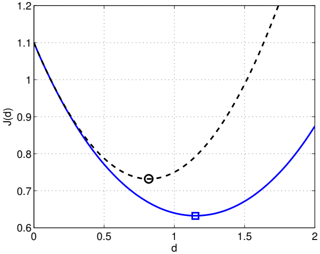
$$\tilde{J}(t) = \hat{J}(u^{(k)} + td^{(k+1)})$$



Approaches for Step Size Selection

$$\hat{J}(u) := J(x(u), u)$$

- Optimal: 1d–search: $s_{opt} = \text{argmin}_{\delta>0} \hat{J}\left(u^{(k)} - \delta d^{(k+1)}\right)$
- Simple: Fixed step size $s = s_0$
- Better: Armijo–rule with back–tracking $s = s_0 \beta^k$
- Even better: Polynomial estimate
  - Given $u^{(k)}$, $\hat{J}'(u^{(k)})$
  - Compute $\hat{J}(u^{(k)} - s_1 d^{(k+1)})$
  - Polynomial approximation
    $$\tilde{J}(t) = \hat{J}(u^{(k)} + t d^{(k+1)})$$
  - Use $s = \text{argmin}_t \tilde{J}(t)$.



Extensions to PDEs

- PDE–model $\partial_t y = \Delta y + f(t, y, u)$

Extensions to PDEs

- PDE–model $\partial_t y = \Delta\, y + f(t, y, u)$
- Domain $\Omega \subset \mathbb{R}_+ \times \mathbb{R}^n$ with suitable boundary conditions

Extensions to PDEs

- PDE–model $\partial_t y = \Delta\, y + f(t, y,\, u)$
- Domain $\Omega \subset \mathbb{R}_+ \times \mathbb{R}^n$ with suitable boundary conditions
- Cost–functional $J(y, u) = \frac{1}{2}\left(\|y - y_d\|^2 + \|u\|^2\right)$

Extensions to PDEs

- PDE–model $\partial_t y = \Delta\, y + f(t, y,\, u)$
- Domain $\Omega \subset \mathbb{R}_+ \times \mathbb{R}^n$ with suitable boundary conditions
- Cost–functional $J(y, u) = \frac{1}{2} \left(\|y - y_d\|^2 + \|u\|^2 \right)$
- Lagrangian $\mathcal{L}(y, u, z) = J(y, u) + \langle F(y, u)\,,\, z \rangle$

Extensions to PDEs

- PDE–model $\partial_t y = \Delta\, y + f(t, y,\, u)$
- Domain $\Omega \subset \mathbb{R}_+ \times \mathbb{R}^n$ with suitable boundary conditions
- Cost–functional $J(y, u) = \frac{1}{2}\left(\|y - y_d\|^2 + \|u\|^2\right)$
- Lagrangian $\mathcal{L}(y, u, z) = J(y, u) + \langle F(y, u)\,,\, z\rangle$
- Adjoint System

$$-\partial_t z = \Delta\, z + \partial_y f \cdot z + (y - y_d)$$

with $z(T) = 0$ and boundary conditions

Single–Objective vs. Multi–Objective Control

- So far: Objective = weighted sum of goal and cost

$$J(x,u) := \frac{1}{2}\left(\|x - x_d\|_2^2 + \alpha \|u\|_2^2\right)$$

weighting / regularization parameter $\alpha > 0$

Single–Objective vs. Multi–Objective Control

- So far: Objective = weighted sum of goal and cost

$$J(x,u) := \frac{1}{2}\left(\|x - x_d\|_2^2 + \alpha\,\|u\|_2^2\right)$$

  weighting / regularization parameter $\alpha > 0$
- How to choose α?

Single–Objective vs. Multi–Objective Control

- So far: Objective = weighted sum of goal and cost

$$J(x,u) := \frac{1}{2}\left(\|x - x_d\|_2^2 + \alpha\,\|u\|_2^2\right)$$

  weighting / regularization parameter $\alpha > 0$

- How to choose α?
- ▸ Choice of α influences result

Single–Objective vs. Multi–Objective Control

- So far: Objective = weighted sum of goal and cost

$$J(x,u) := \frac{1}{2}\left(\|x - x_d\|_2^2 + \alpha \|u\|_2^2\right)$$

weighting / regularization parameter $\alpha > 0$

- How to choose α?
  - Choice of α influences result
  - Optimize both criteria at the same time!

$$\min\left(J_1(x,u),\ J_2(x,u)\right)$$

Single–Objective vs. Multi–Objective Control

- So far: Objective = weighted sum of goal and cost

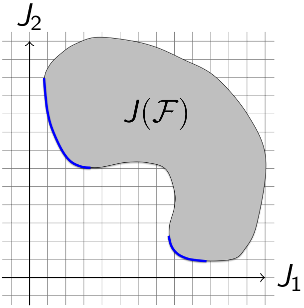
$$J(x,u) := \frac{1}{2}\left(\|x - x_d\|_2^2 + \alpha\,\|u\|_2^2\right)$$

  weighting / regularization parameter $\alpha > 0$

- How to choose α?
  - Choice of α influences result
  - Optimize both criteria at the same time!

$$\min\left(J_1(x,u),\ J_2(x,u)\right)$$

  - Find points/solutions such that for fixed J_1, there is no better J_2 and vice versa.



Multi–Objective Control

Typical Situation: *More than one optimization task*

- Dynamical system $x' = f(t, x, u)$, $x(0) = x_0$

Multi–Objective Control

Typical Situation: *More than one optimization task*

- Dynamical system $x' = f(t, x, u)$, $x(0) = x_0$
- Feasible set $\mathcal{F} = \{(x, u) \text{ solving the dynamical system}\} \subset \mathcal{X} \times \mathcal{U}$

Multi–Objective Control

Typical Situation: *More than one optimization task*

- Dynamical system $x' = f(t, x, u)$, $x(0) = x_0$
- Feasible set $\mathcal{F} = \{(x, u) \text{ solving the dynamical system}\} \subset \mathcal{X} \times \mathcal{U}$
- Objective functions $J_1(x, u)$, $J_2(x, u)$

Multi–Objective Control

Typical Situation: *More than one optimization task*

- Dynamical system $x' = f(t, x, u)$, $x(0) = x_0$
- Feasible set $\mathcal{F} = \{(x, u) \text{ solving the dynamical system}\} \subset \mathcal{X} \times \mathcal{U}$
- Objective functions $J_1(x, u)$, $J_2(x, u)$
- Feasible outcomes $J(\mathcal{F}) \subset \mathbb{R}^2$

Multi–Objective Control

Typical Situation: *More than one optimization task*

- Dynamical system $x' = f(t, x, u)$, $x(0) = x_0$
- Feasible set $\mathcal{F} = \{(x, u) \text{ solving the dynamical system}\} \subset \mathcal{X} \times \mathcal{U}$
- Objective functions $J_1(x, u)$, $J_2(x, u)$
- Feasible outcomes $J(\mathcal{F}) \subset \mathbb{R}^2$
- Task

$$\min J(x, u) = \min \left(J_1(x, u),\ J_2(x, u) \right)$$

Multi–Objective Control

Typical Situation: *More than one optimization task*

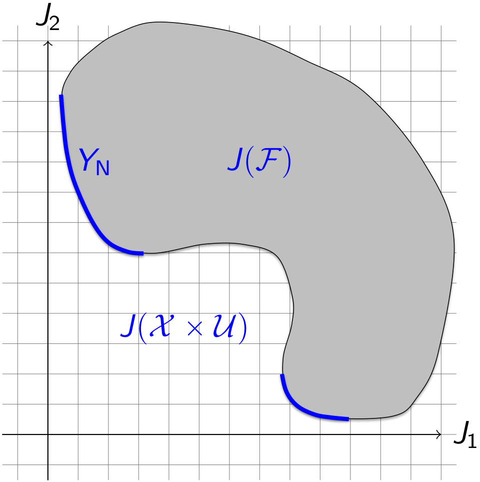
- Dynamical system $x' = f(t, x, u)$, $x(0) = x_0$
- Feasible set $\mathcal{F} = \{(x, u) \text{ solving the dynamical system}\} \subset \mathcal{X} \times \mathcal{U}$
- Objective functions $J_1(x, u)$, $J_2(x, u)$
- Feasible outcomes $J(\mathcal{F}) \subset \mathbb{R}^2$
- Task

$$\min J(x, u) = \min \left(J_1(x, u),\ J_2(x, u)\right)$$

- Ordering in $\mathbb{R}^2$, $\mathbb{R}^n$? $\rightsquigarrow$ Product order

$$(x_1, x_2) < (y_1, y_2) \text{ if } x_1 < y_1 \text{ and } x_2 < y_2$$

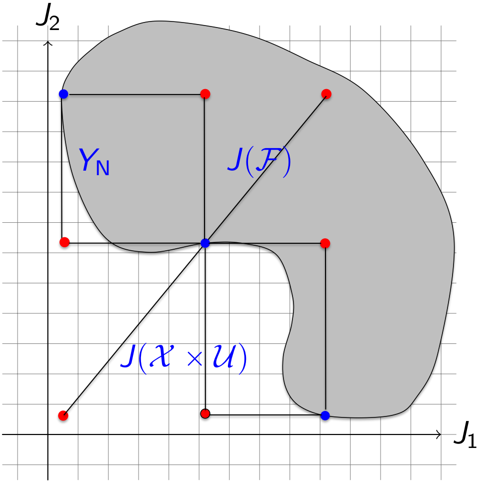
Non–dominated Points



- $J(\mathcal{F})$: set of feasible outcomes
- Y_N: set of non–dominated points
- Any point $y \in Y_N$ is an "optimal" solution of the problem

$$\min J = \min (J_1, J_2)$$

Computing the Set of Non–dominated Points



Rectangular framing

Divide–and–Conquer Method

- Generate ancor points
- Subdivide using mid–point
- Solve SO–problem
- Get two new rectangles
- Iterate

Idea: K.Putra

Fishing Model

[adapted from toonpool.com]

Model

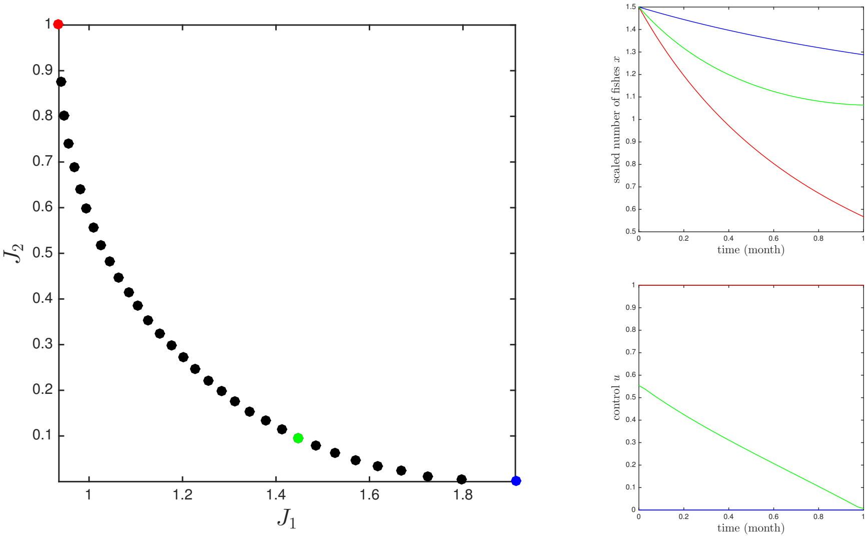
Objective:

$$\min_{(x,u)\in\mathcal{X}\times\mathcal{U}} \left(\int_0^T x^2\,dt, \int_0^T u^2\,dt \right)$$

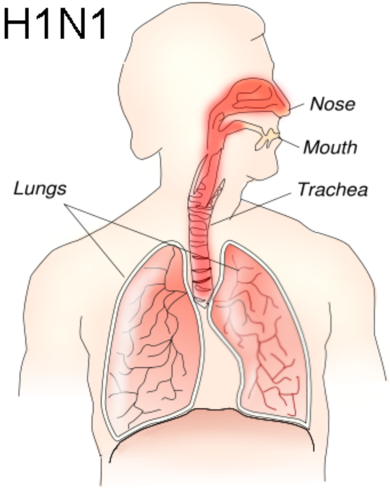
Constraints:

$$x' = r\,x\,(1-x) - u(t)x$$
$$x(0) = x_0$$
$$u(t) \in [0,1]$$
$$t \in [0,T]$$

Fishing Model – Pareto Front



Influenza Model



©Wikipedia 2009

Model

Objective:

$$\min_{(x,u)\in\mathcal{X}\times\mathcal{U}} \left(\int_0^T I^2\,dt, \int_0^T u^2\,dt \right)$$

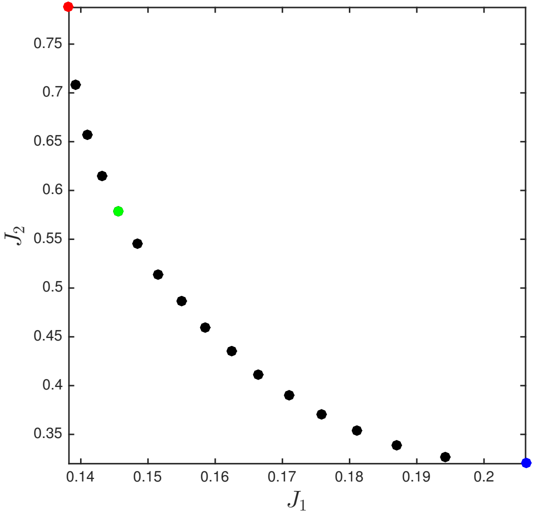
Constraints:

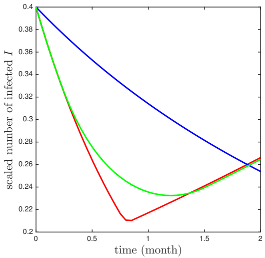
$$\begin{aligned} S' &= -b\,S\,I + c\,I + u\,I \quad & S(0) &= S_0 \\ I' &= b\,S\,I - c\,I - u\,I \quad & I(0) &= I_0 \end{aligned}$$

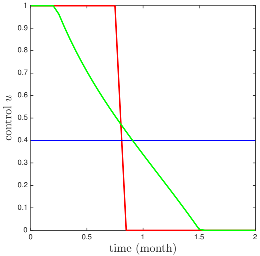
$u(t) \in [0, 1]$

$$A = \int_0^T u(t)\,dt$$

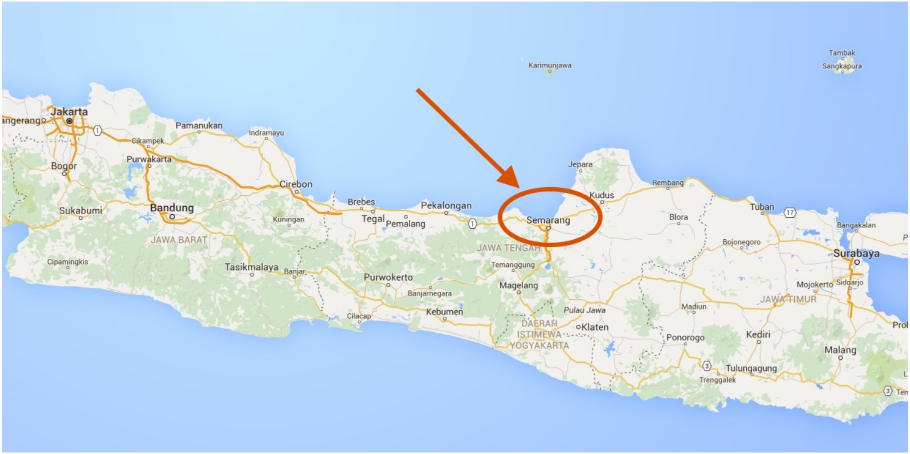
Influenza Model – Pareto Front





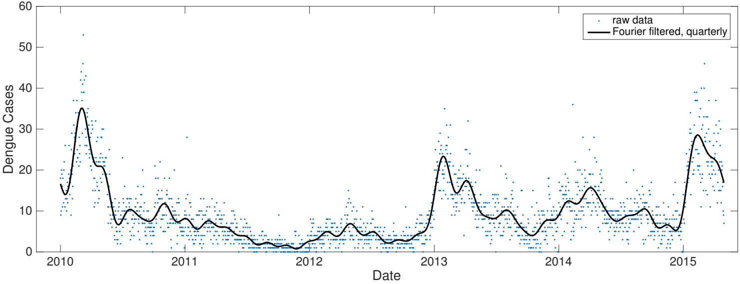


Dengue in Semarang, Java, Indonesia



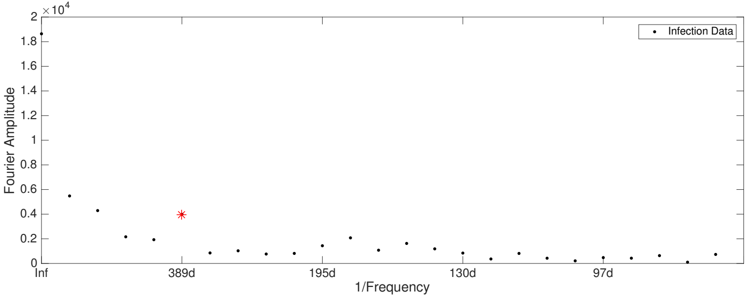
- Population: approx. 1.5 Mio.
- Tropical climate, wet season in Dec–Mar, dry season Jun–Aug

Dengue Data from Semarang



- Hospitalized cases of Dengue
  - DF: Dengue Fever
  - DHF: Dengue Haemorrhagic Fever
  - DSS: Dengue Shock syndrome
- January 2010 – April 2015
- Age structured data also available (?)

Data Analysis



- $n = 1946$ data
- Low–pass filter
- Fourier–transform shows peak at period $\tau = 389d$

SIR UV–Model

$$
\begin{aligned}
S' &= \mu(N - S) - \frac{\beta(t)}{M} S \cdot V \\
I' &= \frac{\beta(t)}{M} S \cdot V - (\gamma + \mu) I \\
R' &= \gamma I - \mu R \\
(U', V') &= \dots
\end{aligned}
$$

Idea from time–scale separation (Rocha, Aguiar, e.a.)

$$
V \simeq \frac{I}{I + \nu N} M, \quad \nu = \frac{1}{2}
$$

Time–dependent *biting–rate* $\beta(t)$

IR–Model

$$I' = \beta(t)(N - I - R)\frac{I}{I + \nu N} - (\gamma + \mu)I$$
$$R' = \gamma I - \mu R$$

Parameters

$$\mu = 1/65a \qquad \gamma = 1/30d \qquad N = 1.5 \cdot 10^6 \qquad I(0) = \text{given}$$

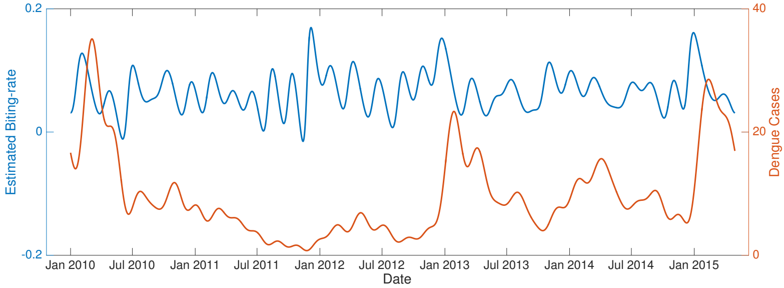
$$R_0 \stackrel{?}{=} R^* = \rho N \text{ from equilibrium}, \; \rho = \frac{3\delta^2 - \delta}{4\delta^2 + 6\delta + 2}, \quad \delta = \frac{\mu}{\gamma} \gg 1$$

- Biting–rate $\beta(t)$ as *Control–function*

A–priori Estimate for β

$$\beta_{est} = \frac{\nu}{1-\rho} \frac{I' + (\gamma+\mu)I}{I}$$

with average $$\overline{\beta}_{est} = \frac{\nu}{1-\rho}\left[\frac{1}{T}\ln\frac{I_0}{I_n} + \gamma + \mu\right] \to 2\gamma$$



Estimating β using Optimal Control

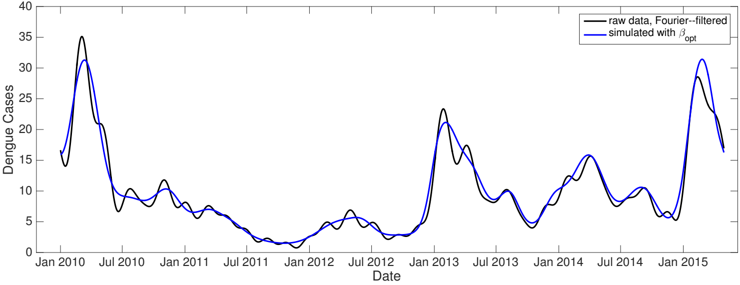
Find $\beta(t)$ such, that $\|I - I_{obs}\|_2 \to \min$, where I solves IR–system

$$I' = \beta(t)(N - I - R) \cdot \frac{I}{I + N/2} - (\gamma + \mu)I$$
$$R' = \gamma I - \mu R$$

Parameter

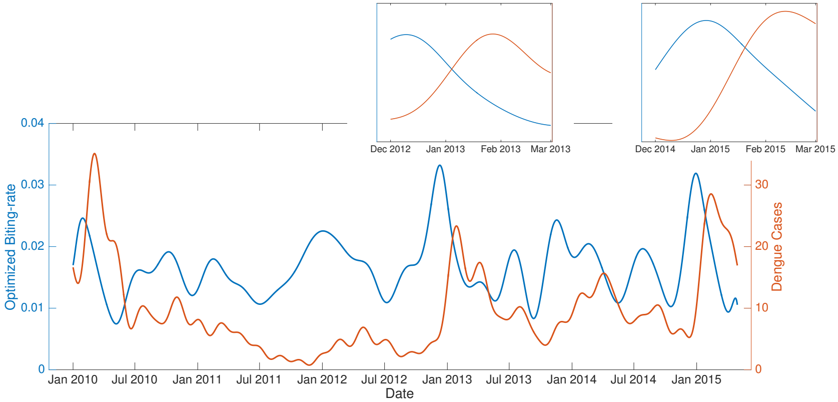
- $\gamma = 1/(30\,\mathrm{d})$
- $\mu = 1/(365 \cdot 65\,\mathrm{d})$
- $N = 1.5 \cdot 10^6$
- $I_0 = 16$
- $R_0 = \sum_{k=1}^{30} I_{obs,k}$ (?)

Optimal Control Solution vs. Data



- Observed data (Fourier–filtered)
- Simulation result with β_{opt}
- Good agreement — as to be expected

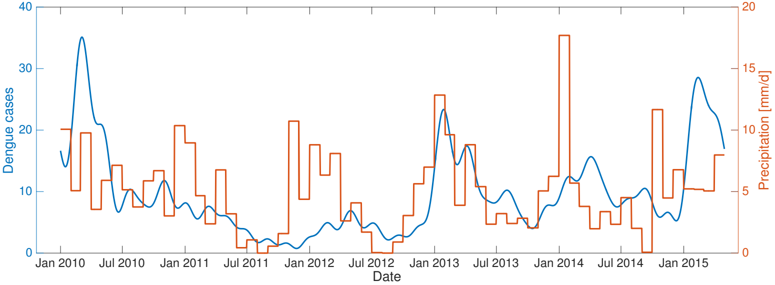
Optimal Control Solution vs. Data



- Biting rate β_{opt} from Optimal Control vs. Dengue Data
- Peaks in Dengue lag behind peaks in β_{opt}
- Delay t$\simeq 40d$ (Jan 2013 and Jan 2015)

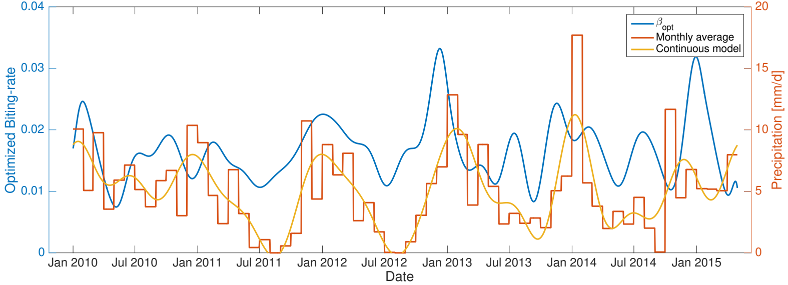
Influence of Precipitation

- *Aedes aegyptii* breed in stagnant water
- Eggs viable in dry state, can re–emerge later on
- Pronounced wet season in Semarang
- Historic precipitation data (monthly) $p(t)$

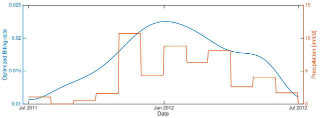


- Good agreement in 2013

Influence of Precipitation



- Biting–rate $\beta(t)$ depends on precipitation data $p(t)$
- July 2011 – July 2012



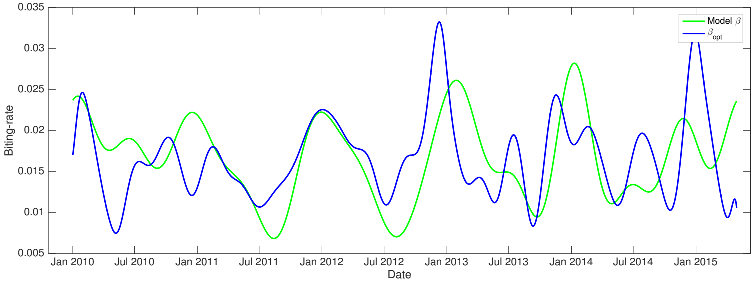
Modeling the Influence of Precipitation

- Normalize precipitation data

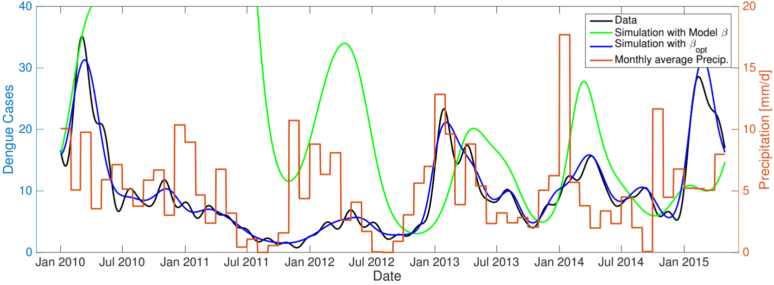
$$p_0 := \frac{p(t) - \overline{p}}{\sigma(p)}$$

with average $\overline{p} = \frac{1}{T}\int_0^T p(t)\,dt$ and standard deviation $\sigma(p)$

- Model *biting–rate* $\beta(t) := \overline{\beta_{opt}} + \sigma(\beta_{opt})\,p_0$



Modeling the Influence of Precipitation II



- Observed Dengue Cases
- Simulation using precipitation data and $\beta = \overline{\beta}_{opt} + \sigma(\beta_{opt})\, p_0$
- Simulation using β_{opt} from Optimal Control
- Precipitation Data

Conclusion & Outlook

- Optimal Control & Pontryagin

Conclusion & Outlook

- Optimal Control & Pontryagin
- Solution using Adjoint Equations

Conclusion & Outlook

- Optimal Control & Pontryagin
- Solution using Adjoint Equations
- Multi Objective Control

Conclusion & Outlook

- Optimal Control & Pontryagin
- Solution using Adjoint Equations
- Multi Objective Control
- Dengue Simulations for Semarang

Conclusion & Outlook

- Optimal Control & Pontryagin
- Solution using Adjoint Equations

- Multi Objective Control

- Dengue Simulations for Semarang
- Modeling the Influence of Precipitation

Conclusion & Outlook

- Optimal Control & Pontryagin
- Solution using Adjoint Equations

- Multi Objective Control

- Dengue Simulations for Semarang
- Modeling the Influence of Precipitation

- Improve this model !

Conclusion & Outlook

- Optimal Control & Pontryagin
- Solution using Adjoint Equations
- Multi Objective Control
- Dengue Simulations for Semarang
- Modeling the Influence of Precipitation

- Improve this model !
- Use age–structure data

Conclusion & Outlook

- Optimal Control & Pontryagin
- Solution using Adjoint Equations
- Multi Objective Control
- Dengue Simulations for Semarang
- Modeling the Influence of Precipitation
- Improve this model !
- Use age–structure data

Thanks.