Universidade do Minho

Optimal control of fed-batch processes with particle swarm optimization

A. Ismael F. Vaz

Departamento de Produção e Sistemas

Escola de Engenharia, Universidade do Minho

aivaz@dps.uminho.pt

Eugénio C. Ferreira

Centro de Engenharia Biológica

Escola de Engenharia, Universidade do Minho

ecferreira@deb.uminho.pt

Universidade do Minho

Contents

- The fed-batch optimal control problem
- The used approach to solve the optimal control problem
- The particle swarm paradigm
- Numerical results and conclusions

Universidade do Minho

Contents

Universidade do Minho

Contents

Contents

Universidade do Minho

Optimal control

Universidade do Minho

Motivation

- A great number of valuable products are produced using fermentation processes and thus optimizing such processes is of great economic importance.
- Fermentation modeling process involves, in general, highly nonlinear and complex differential equations.
- Often optimizing these processes results in control optimization problems for which an analytical solution is not possible.

Universidade do Minho

Motivation

- A great number of valuable products are produced using fermentation processes and thus optimizing such processes is of great economic importance.
- Fermentation modeling process involves, in general, highly nonlinear and complex differential equations.
- Often optimizing these processes results in control optimization problems for which an analytical solution is not possible.

Universidade do Minho

Motivation

- A great number of valuable products are produced using fermentation processes and thus optimizing such processes is of great economic importance.
- Fermentation modeling process involves, in general, highly nonlinear and complex differential equations.
- Often optimizing these processes results in control optimization problems for which an analytical solution is not possible.

Universidade do Minho

The control problem

- The optimal control problem is described by a set of differential equations $\dot{x} = f(x,u,t), \;\; x(t_0) = x^0, \;\; t_0 \leq t \leq t_f$.
- The performance index J can be generally stated as

$$J(t_f) = \varphi(x(t_f), t_f) + \int_{t_0}^{t_f} \phi(x,u,t)dt,$$

where φ is the performance index of the state variables at final time t_f and ϕ is the integrated performance index during the operation.
- constraints that often reflet some physical limitation of the system are imposed.

Universidade do Minho

The control problem

- The optimal control problem is described by a set of differential equations $\dot{x} = f(x,u,t), \quad x(t_0) = x^0, \quad t_0 \leq t \leq t_f$.
- The performance index J can be generally stated as

$$J(t_f) = \varphi(x(t_f), t_f) + \int_{t_0}^{t_f} \phi(x,u,t)dt,$$

where φ is the performance index of the state variables at final time t_f and ϕ is the integrated performance index during the operation.
- constraints that often reflet some physical limitation of the system are imposed.

Universidade do Minho

The control problem

- The optimal control problem is described by a set of differential equations $\dot{x} = f(x,u,t), \;\; x(t_0) = x^0, \;\; t_0 \leq t \leq t_f$.
- The performance index J can be generally stated as

$$J(t_f) = \varphi(x(t_f), t_f) + \int_{t_0}^{t_f} \phi(x,u,t)dt,$$

  where φ is the performance index of the state variables at final time t_f and ϕ is the integrated performance index during the operation.
- constraints that often reflet some physical limitation of the system are imposed.

The control problem

The general maximization problem (P) can be posed as

$$\begin{aligned} &\max \quad J(t_f) && (1)\\ &s.t. \quad \dot{x} = f(x,u,t) && (2)\\ &\qquad \underline{x} \leq x(t) \leq \overline{x}, && (3)\\ &\qquad \underline{u} \leq u(t) \leq \overline{u}, && (4)\\ &\qquad \forall t \in [t_0, t_f] && (5) \end{aligned}$$

Where the state constraints (3) and control constraints (4) are to be understood as componentwise inequalities.

Universidade do Minho

The control problem

The general maximization problem (P) can be posed as

$$\max \quad J(t_f) \tag{1}$$

$$s.t. \quad \dot{x} = f(x,u,t) \tag{2}$$

$$\underline{x} \leq x(t) \leq \overline{x}, \tag{3}$$

$$\underline{u} \leq u(t) \leq \overline{u}, \tag{4}$$

$$\forall t \in [t_0, t_f] \tag{5}$$

Where the state constraints (3) and control constraints (4) are to be understood as componentwise inequalities.

How we addressed problem (P)?

Universidade do Minho

Approach

- Imposing the penalty function for state constraints results in redefining the objective function as

$$\hat{J}(t_f) = \begin{cases} J(t_f) & \text{if } \underline{x} \leq x(t) \leq \overline{x}, \forall t \in [t_0, t_f] \\ -\infty & \text{otherwise} \end{cases}$$

- We will use a linear interpolating function $w(t)$ (linear spline) to approximate the feeding trajectory function $u(t)$.
The spline segment $w^i(t)$, $i = 1, \ldots, n$, is defined as:

$$w^i(t) = u_{i-1} + (u_i - u_{i-1})(t - t_{i-1})/(t_i - t_{i-1}), \text{ for } t \in [t_{i-1}, t_i].$$

where t_i, $i = 0, \ldots, n$, are the time instants and $u_{i-1} = u(t_{i-1})$.

Universidade do Minho

Approach

- Imposing the penalty function for state constraints results in redefining the objective function as

$$\hat{J}(t_f) = \begin{cases} J(t_f) & \text{if } \underline{x} \leq x(t) \leq \overline{x}, \forall t \in [t_0, t_f] \\ -\infty & \text{otherwise} \end{cases}$$

- We will use a linear interpolating function $w(t)$ (linear spline) to approximate the feeding trajectory function $u(t)$.
  The spline segment $w^i(t)$, $i = 1, \ldots, n$, is defined as:

  $$w^i(t) = u_{i-1} + (u_i - u_{i-1})(t - t_{i-1})/(t_i - t_{i-1}), \text{ for } t \in [t_{i-1}, t_i].$$

  where t_i, $i = 0, \ldots, n$, are the time instants and $u_{i-1} = u(t_{i-1})$.

Universidade do Minho

Nonlinear programming (NLP)

The semi-infinite programming problem is then defined as:

$$\begin{array}{rl} & \max \hat{J}(t_f) \\ s.t. & \dot{x} = f(x,w,t) \\ & \underline{u} \leq w(t) \leq \overline{u}. \end{array}$$

and by using the optimality conditions the SIP is redefine as the following nonlinear programming problem.

Universidade do Minho

Nonlinear programming (NLP)

The semi-infinite programming problem is then defined as:

$$\begin{aligned} &\max \hat{J}(t_f) \\ s.t. \quad &\dot{x} = f(x, w, t) \\ &\underline{u} \leq w(t) \leq \overline{u}. \end{aligned}$$

and by using the optimality conditions the SIP is redefine as the following nonlinear programming problem.

$$\begin{aligned} &\max_{u \in R^{n+1}} \hat{J}(t_f) \\ s.t. \quad &\dot{x} = f(x, w, t) \\ &\underline{u} \leq u(t_i) \leq \overline{u}, \quad i = 1, \ldots, n. \end{aligned}$$

Universidade do Minho

Nonlinear optimization

- $u(t_i)$ are variables to be optimized.
- The initial dynamic system conditions ($x(t_0)$) can be considered as variable.
- $h \in R^{n+1}$ and t_f can also be considered as variables to be optimized. $h_i = t_i - t_{i-1}$, $i = 1, \ldots, n$.

Universidade do Minho

Nonlinear optimization

- $u(t_i)$ are variables to be optimized.
- The initial dynamic system conditions ($x(t_0)$) can be considered as variable.
- $h \in R^{n+1}$ and t_f can also be considered as variables to be optimized. $h_i = t_i - t_{i-1}$, $i = 1, \ldots, n$.

Universidade do Minho

Nonlinear optimization

- $u(t_i)$ are variables to be optimized.
- The initial dynamic system conditions ($x(t_0)$) can be considered as variable.
- $h \in R^{n+1}$ and t_f can also be considered as variables to be optimized. $h_i = t_i - t_{i-1}$, $i = 1, \ldots, n$.

Universidade do Minho

Nonlinear optimization

- $u(t_i)$ are variables to be optimized.
- The initial dynamic system conditions ($x(t_0)$) can be considered as variable.
- $h \in R^{n+1}$ and t_f can also be considered as variables to be optimized. $h_i = t_i - t_{i-1}, i = 1, \dots, n$.

$w(t)$ is not differentiable and we will apply a derivative free algorithm.

Global optimum is most desirable and we will apply a stochastic algorithm.

Universidade do Minho

The PSP

The Particle Swarm Paradigm (PSP)

The PSP is a population (swarm) based algorithm that mimics the social behavior of a set of individuals (particles).

An individual behavior is a combination of its past experience (cognition influence) and the society experience (social influence).

In the optimization context a particle p, at time instant k, is represented by its current position ($u^p(k)$), its best ever position ($y^p(k)$) and its travelling velocity ($v^p(k)$).

Universidade do Minho

The new travel position and velocity

The new particle position is updated by

$$u^p(k+1) = u^p(k) + v^p(k+1),$$

where $v^p(k+1)$ is the new velocity given by

$$v_j^p(k+1) = \iota(k)v_j^p(k) + \mu\omega_{1j}(k)\left(y_j^p(k) - u_j^p(k)\right) + \nu\omega_{2j}(k)\left(\hat{y}_j(k) - u_j^p(k)\right)$$

for $j = 1, \ldots, n$.

- $\iota(k)$ is a weighting factor (inertial)
- μ is the *cognition* parameter and ν is the *social* parameter
- $\omega_{1j}(k)$ and $\omega_{2j}(k)$ are random numbers drawn from the uniform $(0,1)$ distribution.

Universidade do Minho

The new travel position and velocity

The new particle position is updated by

$$u^p(k+1) = u^p(k) + v^p(k+1),$$

where $v^p(k+1)$ is the new velocity given by

$$v_j^p(k+1) = \iota(k)v_j^p(k)+\mu\omega_{1j}(k)\left(y_j^p(k) - u_j^p(k)\right)+\nu\omega_{2j}(k)\left(\hat{y}_j(k) - u_j^p(k)\right)$$

for $j = 1, \dots, n$.

- $\iota(k)$ is a weighting factor (inertial)
- μ is the *cognition* parameter and ν is the *social* parameter
- $\omega_{1j}(k)$ and $\omega_{2j}(k)$ are random numbers drawn from the uniform $(0, 1)$ distribution.

Universidade do Minho

The new travel position and velocity

The new particle position is updated by

$$u^p(k+1) = u^p(k) + v^p(k+1),$$

where $v^p(k+1)$ is the new velocity given by

$$v_j^p(k+1) = \iota(k)v_j^p(k) + \mu\omega_{1j}(k)\left(y_j^p(k) - u_j^p(k)\right) + \nu\omega_{2j}(k)\left(\hat{y}_j(k) - u_j^p(k)\right)$$

for $j = 1, \dots, n$.

- $\iota(k)$ is a weighting factor (inertial)
- μ is the *cognition* parameter and ν is the *social* parameter
- $\omega_{1j}(k)$ and $\omega_{2j}(k)$ are random numbers drawn from the uniform $(0, 1)$ distribution.

Universidade do Minho

The new travel position and velocity

The new particle position is updated by

$$u^p(k+1) = u^p(k) + v^p(k+1),$$

where $v^p(k+1)$ is the new velocity given by

$$v_j^p(k+1) = \iota(k)v_j^p(k)+\mu\omega_{1j}(k)\left(y_j^p(k) - u_j^p(k)\right)+\nu\omega_{2j}(k)\left(\hat{y}_j(k) - u_j^p(k)\right)$$

for $j = 1, \ldots, n$.

- $\iota(k)$ is a weighting factor (inertial)
- μ is the *cognition* parameter and ν is the *social* parameter
- $\omega_{1j}(k)$ and $\omega_{2j}(k)$ are random numbers drawn from the uniform $(0, 1)$ distribution.

Universidade do Minho

The best ever particle

$\hat{y}(k)$ is a particle position with global best function value so far, *i.e.*,

$$\hat{y}(k) = \arg\min_{a \in \mathcal{A}} f(a)$$
$$\mathcal{A} = \left\{ y^1(k), \ldots, y^s(k) \right\}.$$

where s is the number of particles in the swarm.

Universidade do Minho

The best ever particle

$\hat{y}(k)$ is a particle position with global best function value so far, *i.e.*,

$$\hat{y}(k) = \arg\min_{a \in \mathcal{A}} f(a)$$

$$\mathcal{A} = \left\{ y^1(k), \ldots, y^s(k) \right\}.$$

where s is the number of particles in the swarm.

In an algorithmic point of view we just have to keep track of the particle with the best ever function value.

Universidade do Minho

Features

Population based algorithm.

1. Good
   (a) Easy to implement.
   (b) Easy to parallelize.
   (c) Easy to handle discrete variables.
   (d) Only uses objective function evaluations.
2. Not so good
   (a) Slow rate of convergence near an optimum.
   (b) Quite large number of function evaluations.
   (c) In the presence of several global optima the algorithm may not converge.

Universidade do Minho

Features

Population based algorithm.

1. Good
   (a) Easy to implement.
   (b) Easy to parallelize.
   (c) Easy to handle discrete variables.
   (d) Only uses objective function evaluations.
2. Not so good
   (a) Slow rate of convergence near an optimum.
   (b) Quite large number of function evaluations.
   (c) In the presence of several global optima the algorithm may not converge.

Universidade do Minho

Features

Population based algorithm.

1. Good
   (a) Easy to implement.
   (b) Easy to parallelize.
   (c) Easy to handle discrete variables.
   (d) Only uses objective function evaluations.
2. Not so good
   (a) Slow rate of convergence near an optimum.
   (b) Quite large number of function evaluations.
   (c) In the presence of several global optima the algorithm may not converge.

Universidade do Minho

Features

Population based algorithm.

1. Good
   (a) Easy to implement.
   (b) Easy to parallelize.
   (c) Easy to handle discrete variables.
   (d) Only uses objective function evaluations.
2. Not so good
   (a) Slow rate of convergence near an optimum.
   (b) Quite large number of function evaluations.
   (c) In the presence of several global optima the algorithm may not converge.

Features

Population based algorithm.

1. Good
   (a) Easy to implement.
   (b) Easy to parallelize.
   (c) Easy to handle discrete variables.
   (d) Only uses objective function evaluations.
2. Not so good
   (a) Slow rate of convergence near an optimum.
   (b) Quite large number of function evaluations.
   (c) In the presence of several global optima the algorithm may not converge.

Universidade do Minho

Features

Population based algorithm.

1. Good
   (a) Easy to implement.
   (b) Easy to parallelize.
   (c) Easy to handle discrete variables.
   (d) Only uses objective function evaluations.
2. Not so good
   (a) Slow rate of convergence near an optimum.
   (b) Quite large number of function evaluations.
   (c) In the presence of several global optima the algorithm may not converge.

Features

Population based algorithm.

1. Good
   (a) Easy to implement.
   (b) Easy to parallelize.
   (c) Easy to handle discrete variables.
   (d) Only uses objective function evaluations.
2. Not so good
   (a) Slow rate of convergence near an optimum.
   (b) Quite large number of function evaluations.
   (c) In the presence of several global optima the algorithm may not converge.

Universidade do Minho

Environment

Universidade do Minho

Modeling language - AMPL

AMPL is a modeling language for mathematical programming.

www.ampl.com

AMPL is commercial software, but a student edition is freely available.

The possibility to load an external dynamic library is exploited in this paper in order to solve ordinary differential equations.

A short example is presented next regarding the external function `chemotherapy`.

Example

```
function chemotherapy;          # external function to be called
param Tumor_mass := log(100); # Tumor cells N=10^12*exp(-x1)
param Drug       := 0;          # Drug concentration in the body
param Cumulative := 0;          # Cumulative effect of the drug
param n := 4;                   # Number of times displacements (knots-1)
param h{1..n} := 21;            # Time displacements, could be variables.
var u{1..n+1};                  # Spline knots

maximize obj:            # maximize objective function
    chemotherapy(0, n, {i in 1..n} h[i], {i in 1..n+1} u[i],
       Tumor_mass, Drug, Cumulative);
subject to hbounds {i in 1..n}:  # constraints on time instants
    1<= h[i] <= 100;             # AMPL just checks for correctness
subject to ubounds {i in 1..n+1}: # constraints on drug delivery
    0.01<= u[i] <= 50;            # problem constraints
option solver mlocpsoa;    # mlocpsoa solver
option mlocpsoa_options 'mlocal=0 size=60 maxiter=1000';
         # global search, population size of 60, maximum of 1000 iterations
solve;                     # solve problem
```

Universidade do Minho

Additional constraints

Additional constraints can easily be incorporated into the model. If, for example, a constraint in the total allowed glucose addition (t_G) is to be imposed, the constraint

$$\sum_{i=0}^{n-1} h_{i+1}(u_i + u_{i+1})/2 \leq t_G$$

can easily be considered in the model file by adding

```
subject to totalfeed:
   sum {i in 0..n-1} (h[i+1]*(u[i]+u[i+1])/2)<=t_G;
```

and to properly define the `t_G` parameter.

Universidade do Minho

Some details

- The ordinary differential equations are solved by calling the CVODE package where the Newton iteration with the CVDiag module was selected.
- At each call to the `chemotherapy` function the linear spline is computed with the provided data and the objective function value is returned. The objective function expression is therefore coded in the external library.
- MLOCPSOA stands for Multi-LOCal Particle Swarm Optimization Algorithm.
- MLOCPSOA provides an interface to AMPL, allowing problems to be easily coded and solved in this modeling language.
- The NLOCPSOA allows a wide variety of algorithm parameters to be set. The used parameters are `size` for the population size (defaults to $\min(6^n, 1000)$), `maxiter` for the maximum allowed iterations (defaults to 2000) and `mlocal` for multi-local search (defaults to 0 – global search instead of multi-local search).

Universidade do Minho

Some details

- The ordinary differential equations are solved by calling the CVODE package where the Newton iteration with the CVDiag module was selected.
- At each call to the `chemotherapy` function the linear spline is computed with the provided data and the objective function value is returned. The objective function expression is therefore coded in the external library.
- MLOCPSOA stands for Multi-LOCal Particle Swarm Optimization Algorithm.
- MLOCPSOA provides an interface to AMPL, allowing problems to be easily coded and solved in this modeling language.
- The NLOCPSOA allows a wide variety of algorithm parameters to be set. The used parameters are `size` for the population size (defaults to $\min(6^n, 1000)$), `maxiter` for the maximum allowed iterations (defaults to 2000) and `mlocal` for multi-local search (defaults to 0 – global search instead of multi-local search).

Universidade do Minho

Some details

- The ordinary differential equations are solved by calling the CVODE package where the Newton iteration with the CVDiag module was selected.
- At each call to the `chemotherapy` function the linear spline is computed with the provided data and the objective function value is returned. The objective function expression is therefore coded in the external library.
- MLOCPSOA stands for Multi-LOCal Particle Swarm Optimization Algorithm.
- MLOCPSOA provides an interface to AMPL, allowing problems to be easily coded and solved in this modeling language.
- The NLOCPSOA allows a wide variety of algorithm parameters to be set. The used parameters are `size` for the population size (defaults to $\min(6^n, 1000)$), `maxiter` for the maximum allowed iterations (defaults to 2000) and `mlocal` for multi-local search (defaults to 0 – global search instead of multi-local search).

Universidade do Minho

Some details

- The ordinary differential equations are solved by calling the CVODE package where the Newton iteration with the CVDiag module was selected.
- At each call to the `chemotherapy` function the linear spline is computed with the provided data and the objective function value is returned. The objective function expression is therefore coded in the external library.
- MLOCPSOA stands for Multi-LOCal Particle Swarm Optimization Algorithm.
- MLOCPSOA provides an interface to AMPL, allowing problems to be easily coded and solved in this modeling language.
- The NLOCPSOA allows a wide variety of algorithm parameters to be set. The used parameters are `size` for the population size (defaults to $\min(6^n, 1000)$), `maxiter` for the maximum allowed iterations (defaults to 2000) and `mlocal` for multi-local search (defaults to 0 – global search instead of multi-local search).

Universidade do Minho

Some details

- The ordinary differential equations are solved by calling the CVODE package where the Newton iteration with the CVDiag module was selected.
- At each call to the `chemotherapy` function the linear spline is computed with the provided data and the objective function value is returned. The objective function expression is therefore coded in the external library.
- MLOCPSOA stands for Multi-LOCal Particle Swarm Optimization Algorithm.
- MLOCPSOA provides an interface to AMPL, allowing problems to be easily coded and solved in this modeling language.
- The NLOCPSOA allows a wide variety of algorithm parameters to be set. The used parameters are `size` for the population size (defaults to $\min(6^n, 1000)$), `maxiter` for the maximum allowed iterations (defaults to 2000) and `mlocal` for multi-local search (defaults to 0 – global search instead of multi-local search).

Universidade do Minho

Numerical results

Parameters

- Numerical results were obtained for the five case studies of fed-batch fermentation processes.
- The time displacements were kept fixed and the best control feeding trajectory was approximated by computing the knots function value.
- MLOCPSOA solver used a population size of 60 and a maximum of 1000 iterations (reaching a maximum of 60000 function evaluations).
- Since MLOCPSOA is a stochastic algorithm we performed 10 solver runs for each problem and the best solutions obtained are report.

Parameters

- Numerical results were obtained for the five case studies of fed-batch fermentation processes.
- The time displacements were kept fixed and the best control feeding trajectory was approximated by computing the knots function value.
- MLOCPSOA solver used a population size of 60 and a maximum of 1000 iterations (reaching a maximum of 60000 function evaluations).
- Since MLOCPSOA is a stochastic algorithm we performed 10 solver runs for each problem and the best solutions obtained are report.

Universidade do Minho

Parameters

- Numerical results were obtained for the five case studies of fed-batch fermentation processes.
- The time displacements were kept fixed and the best control feeding trajectory was approximated by computing the knots function value.
- MLOCPSOA solver used a population size of 60 and a maximum of 1000 iterations (reaching a maximum of 60000 function evaluations).
- Since MLOCPSOA is a stochastic algorithm we performed 10 solver runs for each problem and the best solutions obtained are report.

Universidade do Minho

Parameters

- Numerical results were obtained for the five case studies of fed-batch fermentation processes.
- The time displacements were kept fixed and the best control feeding trajectory was approximated by computing the knots function value.
- MLOCPSOA solver used a population size of 60 and a maximum of 1000 iterations (reaching a maximum of 60000 function evaluations).
- Since MLOCPSOA is a stochastic algorithm we performed 10 solver runs for each problem and the best solutions obtained are report.

Universidade do Minho

Results

			MLOCPSOA		Previous	
Problem	NT	n	$\hat{J}(t_f)$	t_f	$\hat{J}(t_f)$	t_f
`penicillin` [1]	1	5	88.29	132.00	87.99	132.00
`ethanol` [1]	1	5	20379.50	61.20	20839.00	61.17
`chemotherapy` [1]	1	4	16.83	84.00	17.48	84.00
`hprotein` [2]	1	5	32.73	15.00	32.40	15.00
`rprotein` [3]	2	5	0.12	10.00	0.16	10.00

[1] J.R. Banga, E.Balsa-Canto, C.G. Moles, and A.A. Alonso. Dynamic optimization of bioprocesses: Efficient and robust numerical strategies. *Journal of Biotechnology*, 117:407–419, 2005.
[2] S. Park and W.F. Ramirez. Optimal production of secreted protein in fed-batch reactors. *AIChE Journal*, 34(9):1550–1558, 1988.
[3] J. Lee and W.F. Ramirez. Optimal fed-batch control of induced foreign protein-production by recombinant bacteria. *AIChE Journal*, 40(5):899–907, 1994.

Results

By allowing the spline displacements to be variable and using the objective function $\bar{J}(t_f, h) = (\hat{J}(t_f))/(\sum_{i=1}^{n} h_i)$ we can easily compute the profile with the best ratio per unit time.

	Fixed time		Variable time			
Problem	$\hat{J}(t_f)$	t_f	$\bar{J}(t_f)$	$\hat{J}(t_f)$	t_f	h_i^{max}
`penicillin`	88.29	132.00	0.92	76.16	83.20	60
`ethanol`	20229.50	61.20	604.20	14417.50	23.86	20
`chemotherapy`	16.83	84.00	0.70	8.06	11.52	25
`hprotein`	32.73	15.00	17.57	439.18	25	5
`rprotein`	0.12	10.00	2.38	59.61	25	5

With the additional constraints $0.01 \leq h_i \leq h_i^{max}, i = 1, \ldots, n$.

Universidade do Minho

Conclusions and future work

- We have shown an environment for solving optimal control problems
- We applied the Particle Swarm paradigm to optimal control problems
- The Particle Swarm paradigm proved to be a valuable tool in solving these optimal control problems

Universidade do Minho

Conclusions and future work

- We have shown an environment for solving optimal control problems
- We applied the Particle Swarm paradigm to optimal control problems
- The Particle Swarm paradigm proved to be a valuable tool in solving these optimal control problems

Universidade do Minho

Conclusions and future work

- We have shown an environment for solving optimal control problems
- We applied the Particle Swarm paradigm to optimal control problems
- The Particle Swarm paradigm proved to be a valuable tool in solving these optimal control problems

Universidade do Minho

Conclusions and future work

- We have shown an environment for solving optimal control problems
- We applied the Particle Swarm paradigm to optimal control problems
- The Particle Swarm paradigm proved to be a valuable tool in solving these optimal control problems

As a future research we intend to use cubic splines instead of linear splines to approximate the trajectories.

Universidade do Minho

The End

email:	aivaz@dps.uminho.pt
Web	http://www.norg.uminho.pt/aivaz
email:	ecferreira@deb.uminho.pt
Web	http://www.deb.uminho.pt/ecferreira

Universidade do Minho

Case studies

Universidade do Minho

Fed-batch fermentor for penicillin

The optimization problem (in (P) formulation) is:

$$\max_{u(t)} \; J(t_f) \equiv x_2(t_f)x_4(t_f)$$

$$\begin{aligned} s.t. \quad & \dot{x}_1 = h_1x_1 - ux_1/(500x_4), \;\; \dot{x}_2 = h_2x_1 - 0.01x_2 - ux_2/(500x_4) \\ & \dot{x}_3 = -(h_1x_1)/0.47 - h_2x_1/1.2 - 0.029x_1x_3/(0.0001 + x_3) + \\ & \quad + u(1 - x_3/500)/x_4, \;\; \dot{x}_4 = u/500 \\ & 0 \le x_1(t) \le 40, \;\; 0 \le x_3(t) \le 25, \;\; 0 \le x_4(t) \le 10, \;\; 0 \le u(t) \le 50, \\ & \forall t \in [t_0, t_f] \end{aligned}$$

with

$h_1 = 0.11(x_3/(0.006x_1 + x_3))$ and $h_2 = 0.0055(x_3/(0.0001 + x_3(1 + 10x_3))$

where x_1, x_2 and x_3 are the biomass, penicillin and substrate concentrations (g/L), and x_4 is the volume (L). The initial conditions are $x(t_0) = (1.5, 0, 0, 7)^T$.

Universidade do Minho

Fed-batch reactor for ethanol production

The optimization problem is:

$$\max_{u(t)} \quad J(t_f) \equiv x_3(t_f)x_4(t_f)$$

$$\begin{aligned} s.t. \quad & \dot{x}_1 = g_1 x_1 - u x_1/x_4, \quad \dot{x}_2 = -10 g_1 x_1 + u(150 - x_2)/x_4, \quad \dot{x}_4 = u \\ & \dot{x}_3 = g_2 x_1 - u x_3/x_4, \quad 0 \le x_4(t_f) \le 200, \quad 0 \le u(t) \le 12, \quad \forall t \in [t_0, t_f] \end{aligned}$$

with

$$\begin{aligned} g_1 &= (0.408/(1 + x_3/16))(x_2/(0.22 + x_2)) \\ g_2 &= (1/(1 + x_3/71.5))(x_2/(0.44 + x_2)) \end{aligned}$$

where x_1, x_2 and x_3 are the cell mass, substrate and product concentrations (g/L), and x_4 is the volume (L). The initial conditions are $x(t_0) = (1, 150, 0, 10)^T$.

Drug scheduling for cancer chemotherapy

The optimization problem is:

$$\max_{u(t)} \quad J(t_f) \equiv x_1(t_f)$$

$$s.t. \quad \dot{x}_1 = -k_1 x_1 + k_2(x_2 - k_3) \times H\{x_2 - k_3\}$$

$$\dot{x}_2 = u - k_4 x_2, \quad \dot{x}_3 = x_2$$

$$x_2(t) \leq 50 \quad x_3(t) \leq 2.1 \times 10^3, \quad 0 \leq u(t), \quad \forall t \in [t_0, t_f]$$

with $H\{x_2 - k_3\} = 1$ if $x_2 \geq k_3$ and $H\{x_2 - k_3\} = 0$ if $x_2 < k_3$, where the tumor mass cells is given by $N = 10^{12} \times exp(-x_1)$, x_2 is the drug concentration in the body in drug units [D] and x_3 is the cumulative effect of the drug. The parameters are: $k_1 = 9.9 \times 10^{-4}$days, $k_2 = 8.4 \times 10^{-3}$days^{-1}[D^{-1}], $k_3 = 10$[D^{-1}] and $k_4 = 0.27$days^{-1}. The initial conditions are $x(t_0) = (\ln(100), 0, 0)^T$.
Some extra constraints are imposed as there should be at least a 50% reduction in the size of the tumor every three weeks. The treatment period considered is 84 days and therefore the extra constraints are $x_1(21) \geq \ln(200)$, $x_1(42) \geq \ln(400)$ and $x_1(63) \geq \ln(800)$.

Universidade do Minho

Fed-batch bioreactor for protein production

The optimization problem is:

$$\max_{u(t)} \; J(t_f) \equiv x_4(t_f)x_5(t_f)$$

$$s.t. \quad \dot{x}_1 = \mu x_1 - Dx_1, \quad \dot{x}_2 = -7.3\mu x_1 - D(x_2 - x_2^0)$$

$$\dot{x}_3 = f_P x_1 - Dx_3, \quad \dot{x}_4 = \chi(x_3 - x_4) - Dx_4, \quad \dot{x}_5 = u$$

$$0 \leq u(t) \leq 10, \quad \forall t \in [t_0, t_f]$$

with

$$\mu = 21.87x_2/((x_2+0.4)(x_2+62.5)), \quad f_P = x_2 exp(-5x_2)/(x_2+0.1)$$

$$\chi = 4.75\mu/(0.12+\mu), \quad D = u/x_5$$

where x_1, x_2, x_3 and x_4 are the biomass, glucose, total protein and secreted protein concentrations (g/L), and x_5 is the volume (L). The parameter x_2^0 is 20g/L and the initial conditions are $x(t_0) = (1.0, 5.0, 0.0, 0.0, 1.0)^T$.

Universidade do Minho

Fed-batch fermentation for protein

The optimization problem is:

$$\max_{u(t)} \; J(t_f) \equiv x_3(t_f)x_7(t_f)/Q - \int_{t_0}^{t_f} u_2(\tau)x_4^F \, d\tau$$

$$\begin{aligned} s.t. \quad & \dot{x}_1 = \mu x_1 - Dx_1, \quad \dot{x}_2 = -Y^{-1}\mu x_1 - Dx_2 + u_1 x_2^F/x_7 \\ & \dot{x}_3 = R_{fp}x_1 - Dx_3, \quad \dot{x}_4 = -Dx_4 + u_2 x_4^F/x_7 \\ & \dot{x}_5 = -a_1 x_5, \quad \dot{x}_6 = a_2(1 - x_6), \quad \dot{x}_7 = u_1 + u_2 \\ & 0 \le u_1(t) \le 1, \quad 0 \le u_2(t) \le 1, \quad \forall t \in [t_0, t_f] \end{aligned}$$

with

$$\mu = 0.407\psi(x_5 + 0.22x_6/(0.22 + x_4)), \; R_{fp} = 0.095\psi(0.0005 + x_4)/(0.022 + x_4)$$

$$D = (u_1 + u_2)/x_7, \quad \psi = x_2/(0.108 + x_2 + x_2^2/14814.8)$$

$$a_1 = a_2 = 0.09x_4/(0.034 + x_4)$$

where $Y = 0.51$ is the growth yield coefficient, $Q = 5$ is the ratio of protein value to inducer cost, $x_2^F = 100.0$g/L, $x_4^F = 4.0$g/L, x_1 is the biomass (g/L), x_2, x_3, and x_4 are the glucose, protein and inducer concentrations (g/L), x_5 and x_6 are the inducer shock and inducer recovery factors, and x_7 is the volume (L). The initial conditions are $x(t_0) = (0.1, 40, 0.0, 0.0, 1.0, 0.0, 1.0)^T$.