Vehicle Propulsion Systems
Lecture 5
Deterministic Dynamic Programming and Some Examples

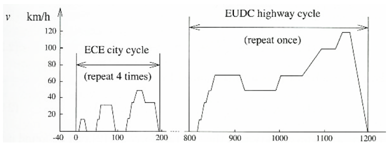
Lars Eriksson
Professor

Vehicular Systems
Linköping University

April 5, 2019

Outline

Energy consumption for cycles



Numerical values for MVEG-95, ECE, EUDC

$$\text{air drag} = \frac{1}{x_{tot}} \sum_{i \in trac} \bar{v}_i^3\, h = \quad \{319, 82.9, 455\}$$

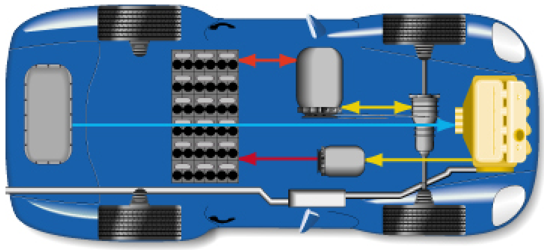
$$\text{rolling resistance} = \frac{1}{x_{tot}} \sum_{i \in trac} \bar{v}_i\, h = \quad \{.856, 0.81, 0.88\}$$

$$\text{kinetic energy} = \frac{1}{x_{tot}} \sum_{i \in trac} \bar{a}_i\, \bar{v}_i\, h = \quad \{0.101, 0.126, 0.086\}$$

$$\bar{E}_{\text{MVEG-95}} \approx A_f\, c_d\, 1.9 \cdot 10^4 + m_v\, c_r\, 8.4 \cdot 10^2 + m_v\, 10 \qquad kJ/100km$$

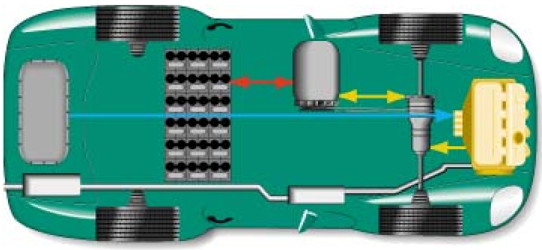
Hybrid Electrical Vehicles – Serial

- Two paths working in parallel
- Decoupled through the battery



Hybrid Electrical Vehicles – Parallel

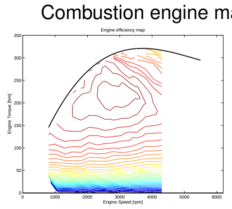
- Two parallel energy paths



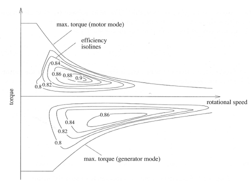
Component modeling

- Model energy (power) transfer and losses
- Using maps $\eta = f(T, \omega)$

Combustion engine map



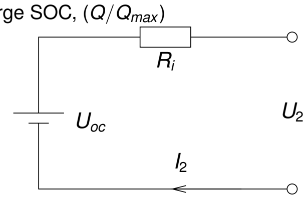
Electric motor map



- Using parameterized (scalable) models –Willans approach

Standard model

Simple model for the battery
–Open circuit voltage $U_{oc}(SOC)$
–State of charge SOC, (Q/Q_{max})

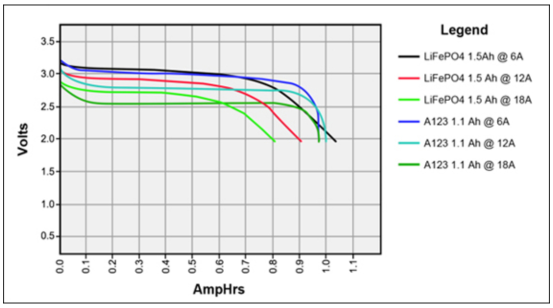


Output voltage

$$U_2 = U_{oc}(SOC) - R_i\, I_2 \qquad \frac{dQ}{dt} = -I_2$$

C-rate, how fast is the battery (pack) charged.
C=1, full capacity in 1 hour.

Voltage and SOC

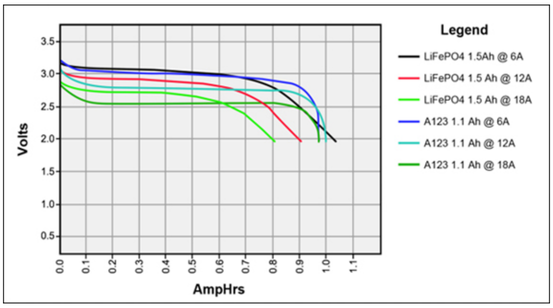


Typical characteristics. Can extract inner resistance, and capacity.

(Source: batteryuniversity.com)

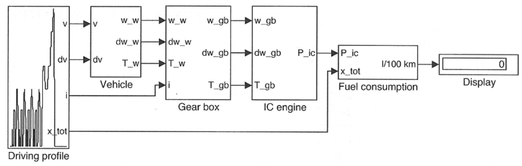
Two important battery estimation problems

- SOC – State of Charge. Current and voltage sensing.
- SOH – State of Health. Cycle monitoring, current and voltage sensing.
- Prolonging life: Temperature monitoring and current limits important.



Model implemented in QSS

Conventional powertrain



Efficient computations are important
–For example if we want to do optimization and sensitivity studies.

Outline

Repetition

“Traditional” Optimization
  Problem motivation
  Different Classes of Problems

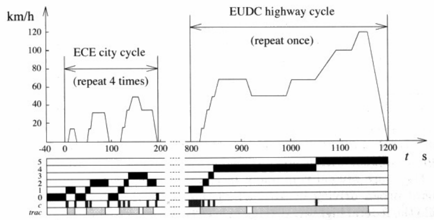
Optimal Control
  Problem Motivation

Deterministic Dynamic Programming
  Problem setup and basic solution idea
  Cost Calculation – Two Implementation Alternatives
  The Provided Tools

Case Studies
  Energy Management of a Parallel Hybrid

Problem motivation

What gear ratios give the lowest fuel consumption for a given drivingcycle?
–Problem presented in appendix 8.1



Problem characteristics

- Countable number of free variables, $i_{g,j}$, $j \in [1,5]$
- A “computable” cost, $m_f(\cdots)$
- A “computable” set of constraints, model and cycle
- The formulated problem

$$\begin{array}{ll} \min\limits_{i_{g,j},\, j\in[1,5]} & m_f(i_{g,1}, i_{g,2}, i_{g,3}, i_{g,4}, i_{g,5}) \\ \text{s.t.} & \text{model and cycle is fulfilled} \end{array}$$

Optimization – Linear Programming

- Linear problem

$$\begin{array}{llcl} \min\limits_{x} & c^T x & & \\ \text{s.t.} & A\,x & = & b \\ & x & \geq & 0 \end{array}$$

- Convex problem
- Much analyzed: existence, uniqueness, sensitivity
- Many algorithms: Simplex the most famous

- About the word *Programming*
  –The solution to a problem was called a program

Optimization – Non-Linear Programming

- Non-linear problem

$$\begin{array}{llcl} \min\limits_{x} & f(x) & & \\ \text{s.t.} & g(x) & = & 0 \\ & x & \geq & 0 \end{array}$$

- For convex problems
  –Much analyzed: existence, uniqueness, sensitivity
  –Many algorithms
- For non-convex problems
  –Some special problems have solutions
  –Local optimum is not necessarily a global optimum

Mixed Integer and Combinatorial Optimziation

- Problem

$$\begin{array}{llcl} \min\limits_{x} & f(x,y) & & \\ \text{s.t.} & g(x,y) & = & 0 \\ & x & \geq & 0 \\ & y & \in & Z^+ \end{array}$$

- Inherently non-convex y
  Generally hard problems to solve.
- Much analyzed
  –Existence, uniqueness, sensitivity
  –Many types of problems
  –Many different algorithms

Some comments on problem solver

- Find the “right” problem formulation
- Use the right solver for the problem

Outline

Repetition

"Traditional" Optimization
- Problem motivation
- Different Classes of Problems

Optimal Control
- Problem Motivation

Deterministic Dynamic Programming
- Problem setup and basic solution idea
- Cost Calculation – Two Implementation Alternatives
- The Provided Tools

Case Studies
- Energy Management of a Parallel Hybrid

Optimal Control – Problem Motivation

Car with gas pedal $u(t)$ as control input:
How to drive from A to B on a given time with minimum fuel consumption?

- Infinite dimensional decision variable $u(t)$.
- Cost function $\int_0^{t_f} \dot{m}_f(t)dt$
- Constraints:
  - Model of the car (the vehicle motion equation)

$$\begin{aligned} m_v \tfrac{d}{dt} v(t) &= F_t(v(t), u(t)) \quad -(F_a(v(t)) + F_r(v(t)) + F_g(x(t))) \\ \tfrac{d}{dt} x(t) &= v(t) \\ \dot{m}_f &= f(v(t), u(t)) \end{aligned}$$

  - Starting point $x(0) = A$
  - End point $x(t_f) = B$
  - Speed limits $v(t) \leq g(x(t))$
  - Limited control action $0 \leq u(t) \leq 1$
- Difficult (impossible) problem to solve analytically

General problem formulation

- Performance index

$$J(u) = \phi(x(t_b), t_b) + \int_{t_a}^{t_b} L(x(t), u(t), t)dt$$

- System model (constraints)

$$\frac{d}{dt} x = f(x(t), u(t), t), \qquad x(t_a) = x_a$$

- State and control constraints

$$u(t) \in U(t)$$

$$x(t) \in X(t)$$

Optimal Control – Historical Perspective

- Old subject
- Rich theory
  - Old theory from calculus of variations
  - Much theory and many methods were developed during 50's-70's
  - Theory and methods are still being actively developed
- Dynamic programming, Richard Bellman, 50's.
- A modern success story:
  –Model predictive control (MPC)
- Now a new interest for collocation methods:
  –A few during 1990's
  –Much interest 2000–

$\Rightarrow$ TSRT08 Optimal Control

Outline

Repetition

"Traditional" Optimization
- Problem motivation
- Different Classes of Problems

Optimal Control
- Problem Motivation

Deterministic Dynamic Programming
- Problem setup and basic solution idea
- Cost Calculation – Two Implementation Alternatives
- The Provided Tools

Case Studies
- Energy Management of a Parallel Hybrid

Dynamic programming – Problem Formulation

- Optimal control problem

$$\begin{aligned} \min\ & J(u) = \phi(x(t_b), t_b) + \int_{t_a}^{t_b} L(x(t), u(t), t)dt \\ s.t.\ & \frac{d}{dt} x = f(x(t), u(t), t) \\ & x(t_a) = x_a \\ & u(t) \in U(t) \\ & x(t) \in X(t) \end{aligned}$$

- $x(t)$, $u(t)$ functions on $t \in [t_a, t_b]$
- Search an approximation to the solution by discretizing
  - the state space $x(t)$
  - and maybe the control signal $u(t)$

  in both amplitude and time.
- The result is a combinatorial (network) problem

Dynamic Programming (DP) – Problem Formulation

- Find the optimal control sequence $\pi^0(x_0) = \{u_0, u_1, \ldots, u_{N-1}\}$ minimizing:

$$J(x_0) = g_N(x_N) + \sum_{k=0}^{N-1} g_k(x_k, u_k, w_k)$$

- subject to:

$$\begin{aligned} x_{k+1} &= f_k(x_k, u_k, w_k) \\ x_0 &= x(t=0) \\ x_k &\in X_k \\ u_k &\in U_k \end{aligned}$$

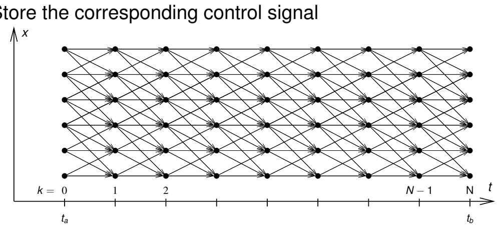
- Disturbance w_k
- Stochastic vs Deterministic DP

DDP – Basic Algorithm

$$J(x_0) = g_N(x_N) + \sum_{k=0}^{N-1} g_k(x_k, u_k)$$

$$x_{k+1} = f_k(x_k, u_k)$$

Bellman's Theory and Algorithm:
–Start at the end and proceed backward in time
–Determine the optimal cost-to-go
–Store the corresponding control signal



DDP – Basic algorithm

$$J(x_0) = g_N(x_N) + \sum_{k=0}^{N-1} g_k(x_k, u_k)$$

$$x_{k+1} = f_k(x_k, u_k)$$

Algorithm:

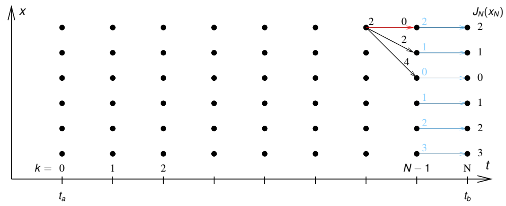
1. Set $k = N$, and assign final cost $J_N(x_N) = g_N(x_N)$
2. Set $k = k - 1$
3. For all points in the state-space grid, find the optimal cost to go

$$J_k(x_k) = \min_{u_k \in U_k(x_k)} g_k(x_k, u_k) + J_{k+1}(f_k(x_k, u_k))$$

4. If $k = 0$ then return solution
5. Go to step 2

Deterministic Dynamic Programming – Basic Algorithm

Graphical illustration of the solution procedure



Arc Cost Calculations

There are two ways for calculating the arc costs

- Calculate the exact control signal and cost for each arc.
  –Quasi-static approach
- Make a grid over the control signal and interpolate the cost for each arc.
  –Forward calculation approach

Matlab implementation – it is important to utilize matrix calculations

- Calculate the whole bundle of arcs in one step
- Add boundary and constraint checks

Pros and Cons with Dynamic Programming

Pros

- Globally optimal, for all initial conditions
- Can handle nonlinearities and constraints
- Time complexity grows linearly with horizon
- Use output and solution as reference for comparison

Cons

- Non causal
- Time complexity grows “exponentially” with number of states, curse of dimensionality
- 2-3 states are often at the limit

Calculation Example

- Problem 200s with discretization $\Delta t = 1$s.
- Control signal discretized with 10 points.
- Statespace discretized with 1000 points.
- One evaluation of the model takes 1μs
- Solution time:
  - Brute force:
    Evaluate all possible combinations of control sequences.
    Number of evaluations, 10^{200} gives $\approx 3 \cdot 10^{186}$ years.
  - Dynamic programming:
    Number of evaluations: $200 \cdot 10 \cdot 1000$ gives 2 s.

(Example contributed by ETH)

The Provided Tools for Hand-in Assignment 2

Task:
Investigate optimal control of one parallel and one series hybrid configuration in different driving profiles.

- Some Matlab-functions provided
  - Skeleton file for defining the problems
  - 2 DDP solvers, 1-dim and 2-dim.
  - 2 skeleton files for calculating the arc costs for parallel and serial hybrids

Outline

Repetition

“Traditional” Optimization
    Problem motivation
    Different Classes of Problems

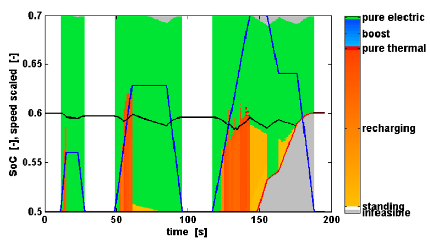
Optimal Control
    Problem Motivation

Deterministic Dynamic Programming
    Problem setup and basic solution idea
    Cost Calculation – Two Implementation Alternatives
    The Provided Tools

Case Studies
    Energy Management of a Parallel Hybrid

Parallel Hybrid Example

- Fuel-optimal torque split factor $u(SOC, t) = \frac{T_{e-motor}}{T_{gearbox}}$
- ECE cycle
- Constraints $SOC(t = t_f) \geq 0.6$, $SOC \in [0.5, 0.7]$



Parallel Hybrid Example

- Fuel-optimal torque split factor $u(SOC, t) = \frac{T_{e-motor}}{T_{gearbox}}$
- NEDC cycle
- Constraints $SOC(t = t_f) = 0.6$, $SOC \in [0.5, 0.7]$