Time Optimal Control and Dynamic Programming

Hugh Runs the Marathon

Shoudong Huang

Centre of Excellence in Autonomous Systems
The University of Technology, Sydney
The Problem

Hugh runs the Marathon

Basic Assumptions:

- Hugh has finite Energy
  \[ p - \text{Energy level}, \ 0 \leq p \leq 1 = 100\% \]
The Problem

Hugh runs the Marathon

Basic Assumptions:

- Hugh has finite Energy
  \( p \) – Energy level, \( 0 \leq p \leq 1 = 100\% \)
- Run faster – Energy dissipates more quickly
The Problem

Hugh runs the Marathon

Basic Assumptions:

- Hugh has finite Energy
  $p$ – Energy level, $0 \leq p \leq 1 = 100\%$
- Run faster – Energy dissipates more quickly
- Energy can not be negative at any time

Constraints: $p_k \geq 0$ for all time step $k$.

$p = 0.001$ – Hugh has to run very slowly or stop
The Problem

How to run the Marathon with Finite Energy?

Choose different speed at different time

For simplicity:

- All the possible speeds
  0 m/s, 1 m/s, 2 m/s, 3 m/s, 4 m/s, 5 m/s
The Problem

How to run the Marathon with Finite Energy?

Choose different speed at different time

For simplicity:

- All the possible speeds
  0 m/s, 1 m/s, 2 m/s, 3 m/s, 4 m/s, 5 m/s
- We can choose a (different) speed at every second
  (or every 10 seconds)
The Problem

How to choose the speed?

- Run very fast at the beginning – silly
  - Local optimization doesn’t work
The Problem

How to choose the speed?

- Run very fast at the beginning – silly
  – Local optimization doesn’t work
- Minimize the total running time
  – Global optimization
The Problem

How to choose the speed?

- Run very fast at the beginning – silly
  - Local optimization doesn’t work
- Minimize the total running time
  - Global optimization
- At time step $k$, choose the speed based on:
  - the current Energy level – $p_k$
    (low energy – choose lower speed)
  - the current Distance to the destination – $d_k$
    (run 50m and run 1000m – different speeds)
The Problem – RL view

This is a **Reinforcement Learning (RL)** problem – *Agent learning from interaction with environment to achieve goal.*

- The goal – minimized the total time used
The Problem – RL view

This is a Reinforcement Learning (RL) problem – Agent learning from interaction with environment to achieve goal.

- The goal – minimized the total time used
- Agent – Hugh’s brain (the learner and decision maker)
The Problem – RL view

This is a **Reinforcement Learning (RL)** problem – *Agent learning from interaction with environment to achieve goal.*

- The goal – minimized the total time used
- Agent – Hugh’s brain (the learner and decision maker)
- Environment – Hugh’s body + road
The Problem – RL view

This is a Reinforcement Learning (RL) problem – Agent learning from interaction with environment to achieve goal.

- The goal – minimized the total time used
- Agent – Hugh’s brain (the learner and decision maker)
- Environment – Hugh’s body + road
- Actions – different speeds
This is a Reinforcement Learning (RL) problem – Agent learning from interaction with environment to achieve goal.

- The goal – minimized the total time used
- Agent – Hugh’s brain (the learner and decision maker)
- Environment – Hugh’s body + road
- Actions – different speeds
- Rewards – 1 s (or −1 s)
This is a Reinforcement Learning (RL) problem – Agent learning from interaction with environment to achieve goal.

- The goal – minimized the total time used
- Agent – Hugh’s brain (the learner and decision maker)
- Environment – Hugh’s body + road
- Actions – different speeds
- Rewards – 1 s (or −1 s)
- (Expected) Return (value function) – the total time $T$ (or $−T$) (Goal: maximize $−T$)
The Problem – Control view

This is a Control problem – select the control inputs such that the system has the best performance

- Performance – the total time used
This is a **Control** problem – *select the control inputs such that the system has the best performance*

- **Performance** – the total time used
- **Controller** – Hugh’s brain
The Problem – Control view

This is a Control problem – select the control inputs such that the system has the best performance

- Performance – the total time used
- Controller – Hugh’s brain
- Plant (System) – Hugh’s body + road
The Problem – Control view

This is a Control problem – select the control inputs such that the system has the best performance

- Performance – the total time used
- Controller – Hugh’s brain
- Plant (System) – Hugh’s body + road
- Control signals – different speeds
This is a Control problem – *select the control inputs such that the system has the best performance*

- **Performance** – the total time used
- **Controller** – Hugh’s brain
- **Plant (System)** – Hugh’s body + road
- **Control signals** – different speeds
- **Constraints** – $p_k \geq 0$
The Problem – Control view

This is a **Control** problem – *select the control inputs such that the system has the best performance*

- **Performance** – the total time used
- **Controller** – Hugh’s brain
- **Plant (System)** – Hugh’s body + road
- **Control signals** – different speeds
- **Constraints** – $p_k \geq 0$
- **Measurements (= States)** – Energy level $p$ and Distance $d$ (*perfect measurement*)
The Problem – Control+RL view

Both Control and RL

- focus on Key Factors
  Environment States (RL)/System States (Control) –
  Energy level $p$ and Distance $d$
The Problem – Control+RL view

Both Control and RL

- focus on Key Factors
  Environment States (RL) / System States (Control) – Energy level $p$ and Distance $d$

- aim to find a
  Policy (RL) / State Feedback Control law (Control) – a mapping from $p, d$ to speed
The Problem – Control+RL view

Both Control and RL

- focus on Key Factors
  - Environment States (RL) / System States (Control) – Energy level $p$ and Distance $d$

- aim to find a
  - Policy (RL) / State Feedback Control law (Control) – a mapping from $p, d$ to speed

- such that
  - the Expected Return is maximized (RL) / the Performance is optimized (Control)
The Solution of the Problem

Hugh runs the Marathon

- Give us the Model (nonlinear, deterministic)
The Solution of the Problem

Hugh runs the Marathon

- Give us the Model (nonlinear, deterministic)
- the problem is a
The Solution of the Problem

Hugh runs the Marathon

- Give us the Model (nonlinear, deterministic)
- the problem is a
- State Feedback (Perfect Measurement Feedback)
The Solution of the Problem

Hugh runs the Marathon

- Give us the Model (nonlinear, deterministic)
- the problem is a
- State Feedback (Perfect Measurement Feedback)
- Nonlinear Optimal Control Problem
Hugh runs the Marathon

- Give us the Model (nonlinear, deterministic)
- the problem is a
- State Feedback (Perfect Measurement Feedback)
- Nonlinear Optimal Control Problem
- with States Constraints
The Model

States – Distance to destination $d$, Energy level $p$

- The model of the Distance $d$
  
  $d_0 = 42195$ (m), $d_{k+1} = d_k - i$ when speed $= i$ m/s
The Model

States – Distance to destination $d$, Energy level $p$

- The model of the Distance $d$
  \[ d_0 = 42195 \text{(m)}, \ d_{k+1} = d_k - i \text{ when speed} = i \text{ m/s} \]

- The model of Energy level $p$ (Assumption)
  \[ p_0 = 1, \ p_{k+1} = f_i(p_k) \text{ when speed} = i \text{ m/s} \]
  e.g.
  \[ p_{k+1} = p_k - 0.005(1 + p_k^2) \text{ when speed} = 5 \text{ m/s} \]
  \[ p_{k+1} = p_k - 0.001 \text{ when speed} = 2 \text{ m/s} \]
  \[ p_{k+1} = p_k + 0.002 \text{ when speed} = 0 \text{ m/s} \]
Value Function and Optimal Policy

Define the Value Function $V(p, d)$ – minimal time needed to reach the destination when the current Energy level is $p$ and the current Distance to the destination is $d$.

Optimal Policy (Control) – $u^*(p, d)$ (the best speed)

Both $V(p, d)$ and $u^*(p, d)$ depend on

- the current energy level $p$
  lower energy – longer time, choose lower speed

- the current distance $d$
  longer distance – longer time, choose lower speed
Dynamic Programming Equation

- \( V(p_k, d_k) \) – minimal total time at step \( k \)
Dynamic Programming Equation

- $V(p_k, d_k)$ – minimal total time at step $k$
- $V(p_{k+1}, d_{k+1})$ – minimal total time at step $k + 1$
Dynamic Programming Equation

- \( V(p_k, d_k) \) – minimal total time at step \( k \)
- \( V(p_{k+1}, d_{k+1}) \) – minimal total time at step \( k + 1 \)
- from step \( k \) to step \( k + 1 \) – need 1 second
Dynamic Programming Equation

- $V(p_k, d_k)$ – minimal total time at step $k$
- $V(p_{k+1}, d_{k+1})$ – minimal total time at step $k + 1$
- from step $k$ to step $k + 1$ – need 1 second
- $V(p_k, d_k) \leq 1 + V(p_{k+1}, d_{k+1})$
Dynamic Programming Equation

- $V(p_k, d_k) - \text{minimal total time at step } k$
- $V(p_{k+1}, d_{k+1}) - \text{minimal total time at step } k + 1$
- From step $k$ to step $k + 1 - \text{need 1 second}$
- $V(p_k, d_k) \leq 1 + V(p_{k+1}, d_{k+1})$
- Why not equal?
Dynamic Programming Equation

- $V(p_k, d_k) - \text{minimal total time at step } k$
- $V(p_{k+1}, d_{k+1}) - \text{minimal total time at step } k + 1$
- from step $k$ to step $k + 1 - \text{need 1 second}$
- $V(p_k, d_k) \leq 1 + V(p_{k+1}, d_{k+1})$
- why not equal?
- $1 + V(p_{k+1}, d_{k+1}) - a \text{ possible minimal total time at step } k$ (the chosen speed may not be optimal)
Dynamic Programming Equation

- \( V(p_k, d_k) \) – minimal total time at step \( k \)
- \( V(p_{k+1}, d_{k+1}) \) – minimal total time at step \( k + 1 \)
- from step \( k \) to step \( k + 1 \) – need 1 second
- \( V(p_k, d_k) \leq 1 + V(p_{k+1}, d_{k+1}) \)
- why not equal?
- \( 1 + V(p_{k+1}, d_{k+1}) \) – a possible minimal total time at step \( k \) (the chosen speed may not be optimal)
- when the chosen speed is optimal
  \( V(p_k, d_k) = 1 + V(p_{k+1}, d_{k+1}) \)
Dynamic Programming Equation

- We have

\[ V(p_k, d_k) = 1 + \min_{u_k \in U(p_k)} V(p_{k+1}, d_{k+1}) \]

where \( U(p_k) = \{ i \text{ m/s} : f_i(p_k) \geq 0 \} \) is the set of all the admissible speeds at time \( k \) (depends on \( p_k \))
Dynamic Programming Equation

- We have

\[ V(p_k, d_k) = 1 + \min_{u_k \in U(p_k)} V(p_{k+1}, d_{k+1}) \]

where \( U(p_k) = \{ i \text{ m/s} : f_i(p_k) \geq 0 \} \) is the set of all the admissible speeds at time \( k \) (depends on \( p_k \))

- Using the dynamics of \( p_k \) and \( d_k \), i.e.

\[ p_{k+1} = f_i(p_k), \quad d_{k+1} = d_k - i \text{ when speed} = i \text{ m/s} \]
Dynamic Programming Equation

- We have

\[ V(p_k, d_k) = 1 + \min_{u_k \in \mathcal{U}(p_k)} V(p_{k+1}, d_{k+1}) \]

where \( \mathcal{U}(p_k) = \{ i \text{ m/s} : f_i(p_k) \geq 0 \} \) is the set of all the admissible speeds at time \( k \) (depends on \( p_k \))

- Using the dynamics of \( p_k \) and \( d_k \), i.e.

\[ p_{k+1} = f_i(p_k), \quad d_{k+1} = d_k - i \text{ when speed } = i \text{ m/s} \]

- We can obtain the Dynamic Programming Equation

\[ V(p, d) = 1 + \min_{i \in \mathcal{U}(p)} V(f_i(p), d - i) \]
The Dynamic Programming Equation can be solved numerically

\[ V_{n+1}(p, d) = 1 + \min_{i \in U(p)} V_n(f_i(p), d - i) \]
The Dynamic Programming Equation can be solved numerically

\[ V_{n+1}(p, d) = 1 + \min_{i \in U(p)} V_n(f_i(p), d - i) \]

in a few seconds time (because \( p \) and \( d \) are scalars)
The Dynamic Programming Equation can be solved numerically

\[ V_{n+1}(p, d) = 1 + \min_{i \in \mathcal{U}(p)} V_n(f_i(p), d - i) \]

in a few seconds time (because \( p \) and \( d \) are scalars)

simultaneously obtain the Value Function and the Optimal Policy (Control) law \( u^*(p, d) \) (offline)
The Dynamic Programming Equation can be solved numerically

\[ V_{n+1}(p, d) = 1 + \min_{i \in U(p)} V_n(f_i(p), d - i) \]

- in a few seconds time (because \( p \) and \( d \) are scalars)
- simultaneously obtain the Value Function and the Optimal Policy (Control) law \( u^*(p, d) \) (offline)
- in the Marathon running, compute \( p_k \) and \( d_k \) at each time step \( k \), and apply the Optimal Policy (Control) law \( u^*(p_k, d_k) \)
Some Remarks on the Model

- The model of $p_k$ is one dimensional, higher dimensional model is more accurate – tradeoff between model accuracy and computation complexity.
Some Remarks on the Model

- The model of $p_k$ is one dimensional, higher dimensional model is more accurate – tradeoff between model accuracy and computation complexity

- The model can be learned by experiments (Hugh runs at different speed at different energy level) – Nonlinear Regression
Some Remarks on the Model

- The model of $p_k$ is one dimensional, higher dimensional model is more accurate – tradeoff between model accuracy and computation complexity

- The model can be learned by experiments (Hugh runs at different speed at different energy level) – Nonlinear Regression

- The model can be learned in the real Marathon running – update the model and the Optimal Policy (Control) law from time to time
Some Remarks on the Model

- The model of $p_k$ is one dimensional, higher dimensional model is more accurate – tradeoff between model accuracy and computation complexity

- The model can be learned by experiments (Hugh runs at different speed at different energy level) – Nonlinear Regression

- The model can be learned in the real Marathon running – update the model and the Optimal Policy (Control) law from time to time

- Tradeoff between exploration (learn the model) and exploiting (apply the current optimal control law) – A Key Issue in RL
Real Problems I would like to tackle

Possible applications in robot activities:

- **Minimal time SLAM problem** – obtain an acceptable map in minimal time (Shoudong, Zhan, Dissa)

Hugh $\rightarrow$ robot

Energy Level $\rightarrow$ Error bound in the estimation
Real Problems I would like to tackle

Possible applications in robot activities:

- **Minimal time SLAM problem** – obtain an acceptable map in minimal time (Shoudong, Zhan, Dissa)
  
  Hugh $\rightarrow$ robot
  Energy Level $\rightarrow$ Error bound in the estimation

- Hugh $\rightarrow$ robot
  Energy Level $\rightarrow$ a bound of uncertainty of the information gathered
  Minimal time exploration problem?
Real Problems I would like to tackle

Possible applications in robot activities:

- Minimal time SLAM problem – obtain an acceptable map in minimal time (Shoudong, Zhan, Dissa)
  Hugh → robot
  Energy Level → Error bound in the estimation

- Hugh → robot
  Energy Level → a bound of uncertainty of the information gathered
  Minimal time exploration problem?

- Hugh → robot
  Energy Level → ???
  Minimal time ?? problem?
Conclusion ? – My feeling

Control and RL

- using different languages
Conclusion ? – My feeling

Control and RL

- using different languages
- address almost the same problems
Conclusion ? – My feeling

Control and RL

- using different languages
- address almost the same problems
- with slightly different focus

- **Control**: rigorous proof; rely on model; separate the whole task into several smaller tasks – system identification, theoretical analysis and synthesis, numerical algorithms
- **RL**: tractable numerical results; try to get rid of model; consider the task in whole
Conclusion ? – My feeling

Control and RL

- using different languages
- address almost the same problems
- with slightly different focus

Control: rigorous proof; rely on model; separate the whole task into several smaller tasks – system identification, theoretical analysis and synthesis, numerical algorithms

RL: tractable numerical results; try to get rid of model; consider the task in whole
Conclusion ? – My feeling

Control and RL

- using different languages
- address almost the same problems
- with slightly different focus

- Control: rigorous proof; rely on model; separate the whole task into several smaller tasks – system identification, theoretical analysis and synthesis, numerical algorithms
- RL: tractable numerical results; try to get rid of model; consider the task in whole