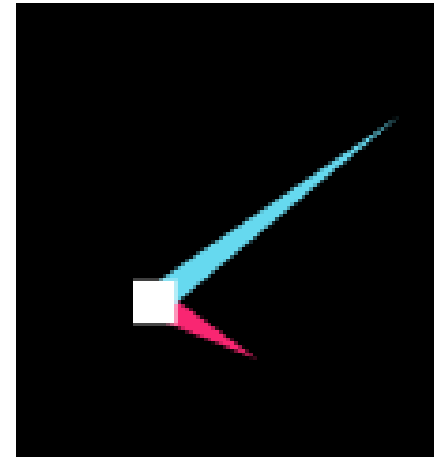


Assignment 6: Trajectory Optimization

TA: Miguel Zamora (mimora@ethz.ch)

Assignment 6

- Assignment credits to James Bern
- Use Github Issues to raise questions
- Use **Release** or **RelWithDebInfo** mode otherwise too slow!



disclaimer

Terminology for trajectory optimization is very confusing and not particularly consistent across sources.

Good resource: <http://underactuated.mit.edu/trajopt.html>

state variables ξ

$$\xi = \begin{bmatrix} x \\ v \end{bmatrix}$$

Note: More complicated systems possible...

$$\begin{pmatrix} \dot{R} \\ \dot{b} \\ \dot{\Omega} \\ \dot{v} \\ \dot{r}_p \\ \dot{r}_b \\ \dot{r}_w \\ \dot{P}_p \\ \dot{P}_b \\ \dot{P}_w \\ \dot{m}_b \end{pmatrix} = \begin{pmatrix} R\hat{\Omega} \\ Rv \\ J^{-1}\bar{T} \\ M^{-1}\bar{F} \\ \frac{1}{m}P_p - v - \Omega \times r_p \\ \frac{1}{m_b}P_b - v - \Omega \times r_b \\ \frac{1}{m_w}P_w - v - \Omega \times r_w \\ \bar{u} \\ u_b \\ u_w \\ u_{ballast_rate} \end{pmatrix}$$

first order dynamics $\dot{\xi}$

$$\mathbf{F} = m\mathbf{a}$$

$$\frac{1}{m}\mathbf{F} = \mathbf{a}$$

$$\dot{\xi} = \begin{bmatrix} \dot{x} \\ \dot{v} \end{bmatrix}$$

integrating forward in time

input

- control trajectory $\mathbf{u} = (\mathbf{u}_0, \dots, \mathbf{u}_{K-1})$
- initial conditions ξ_0

algorithm-ish

- apply physics ($\mathbf{F}_k = m\mathbf{a}_k$) for $k = 0, \dots, K - 1$

output

- physically-valid state trajectory $\xi(\mathbf{u}) = (\xi_1, \dots, \xi_K)$

what is trajectory optimization?

Find the right control trajectory u to achieve a given task.

some approaches

Trajectory Optimization Terminology

Direct shooting

- Optimization
- $\xi(t)$

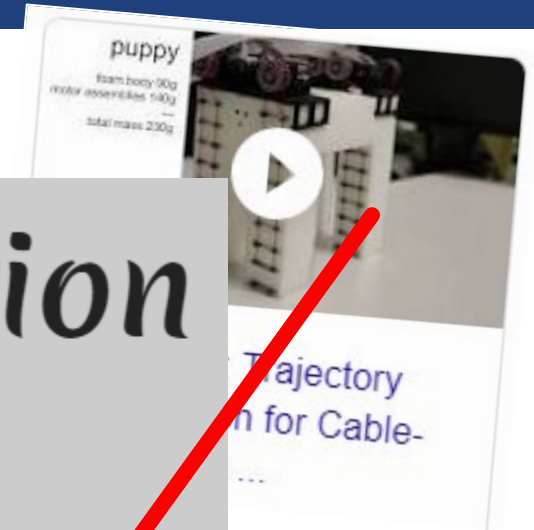
Direct transcription

All trajectory methods can either be described as **shooting** methods or **simultaneous (collocation)** methods. The key difference is that shooting methods use an explicit integration scheme, whereas simultaneous methods use an implicit integration scheme to solve the dynamics.

Direct collocation

- Represent u, ξ as splines and optimize over spline knots
 - Physics imposed as constraints

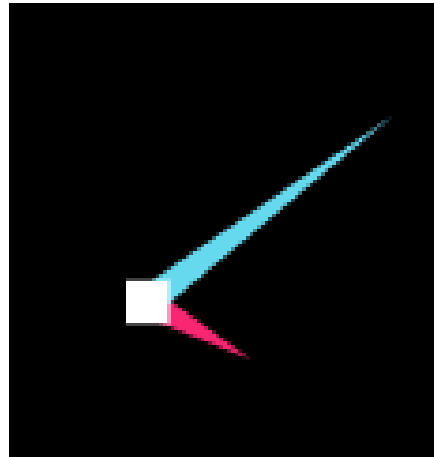
NOTE: Some sources use these terms, “direct transcription”, even though one of them is already called direct transcription.



other approaches

- Direct multiple shooting
- Orthogonal collocation
- Differential Dynamic Programming (DDP)
- ...

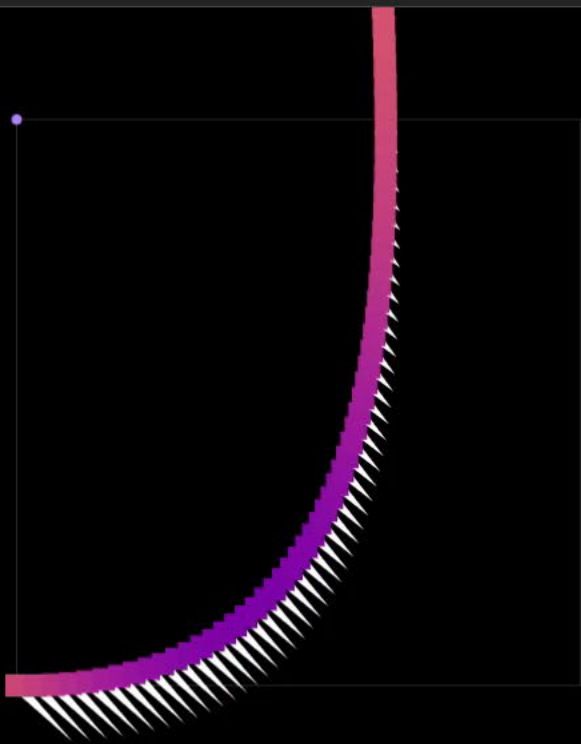
spaceship (no orientation)



shooting-app.cpp

Direct shooting

- Optimize over u
 - $\xi(u)$ found by solving physics



▼ NOTE: Press r to reset (optimization)

PLANET

OPTIMIZE // Keyboard Shortcut: o

SECOND_ORDER_SOVLER

-3.000000

log_c_reg

SIMULATE // Keyboard Shortcut: s

SLOMO

PRINT O

PRINT |G|

✓ DRAW_THRUSTER

DRAW_VELOCITY

1.000000

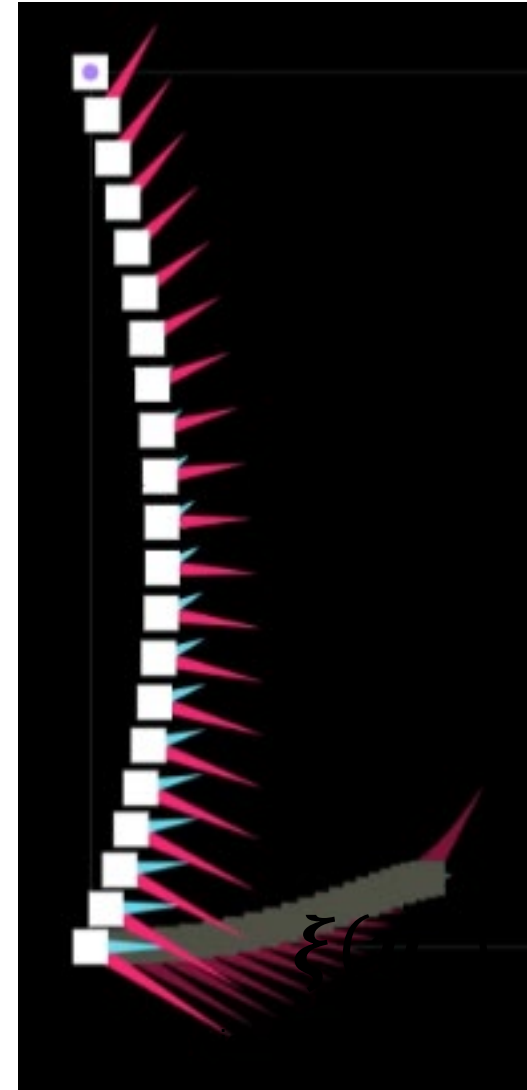
ZOOM

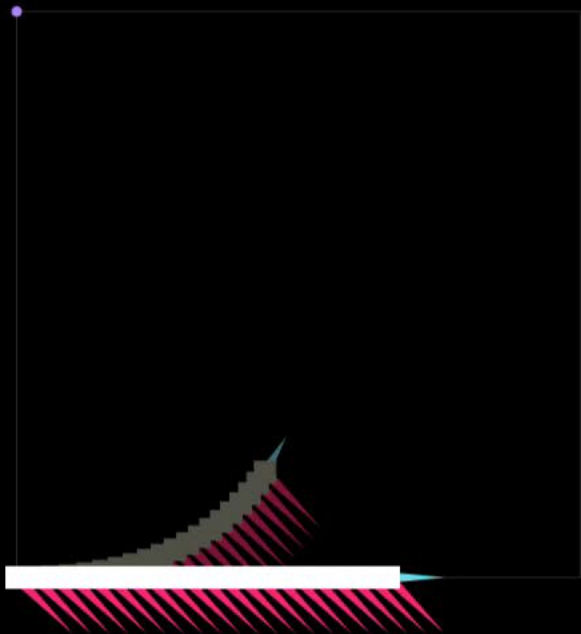
O(u_curr)

transcription-app.cpp

Direct transcription

- Optimize over \mathbf{u}, ξ
 - Physics imposed as constraints





▼ NOTE: Press r to reset (optimization or simulation).

OPTIMIZE // Keyboard Shortcut: o

3.000000 log_c_x0

3.000000 log_c_physics

VISUALIZE // Keyboard Shortcut: s

SLOMO

PRINT o

PRINT |G|

☒ DRAW_REAL_PHYSICS_FOR_COMPARISON

☒ DRAW_THRUSTER

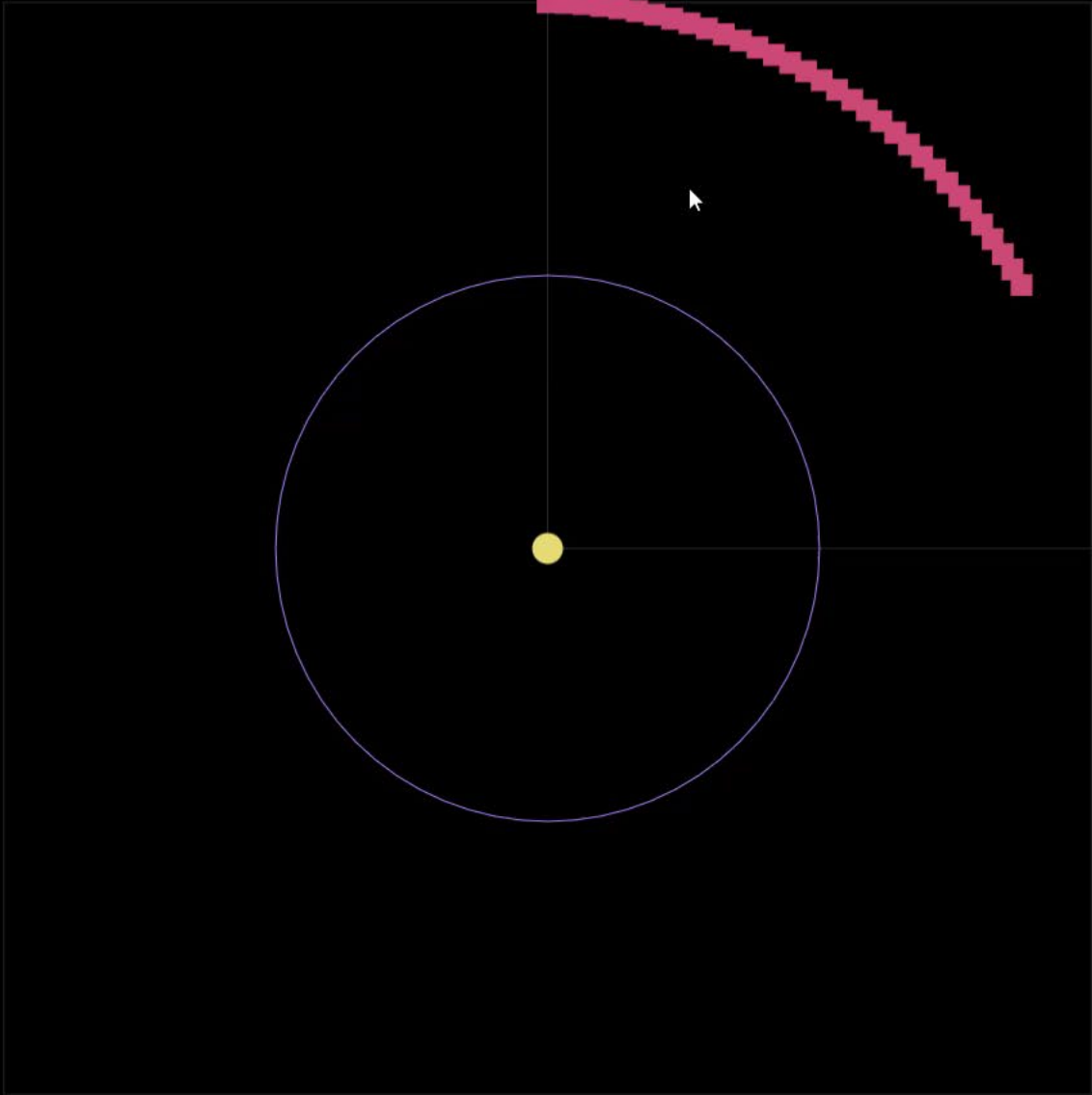
☒ DRAW_VELOCITY

1.000000 ZOOM

O(u_curr)

challenge-app.cpp

Use direct shooting to put spaceship into orbit.



▼ NOTE: Press r to reset (optimization)

☐ OPTIMIZE // Keyboard Shortcut: o

☐ SIMULATE // Keyboard Shortcut: s

☐ SLOMO

☐ PRINT O

☐ PRINT |G|

☒ DRAW_THRUSTER

☐ DRAW_VELOCITY

1.000000

ZOOM

O(u_curr)

Thanks!