3D Translational Rocket Landing LCvx

3DoF (all translational) rocket landing using lossless convexification

Initialize

```
% Physical Parameters
m_0 = 2600; % [kg]
g_0 = 9.81e-3; % [km / s2]
g = [0; 0; -9.81e-3]; % [km / s2]
Isp = 200; % [s]
alpha = 1 / (Isp * g_0); % [s / km]
T_max = 3 * m_0 * 9.81e-3; % [kg km / s2]
T_min = 0.55 * T_max; % [kg km / s2]

% Problem Parameters
N = 50; % [] Number of timesteps
delta_t = 30/(N - 1); % [s]
r_0 = [0; 0; 4.6]; % [km]
theta_0 = deg2rad(120); % [rad]
v_0 = [1 0; 0 0; 0 1] * make_R2(deg2rad(-60)) * [0.306; 0]; % [km / s]
glideslope_angle_max = deg2rad(60); % [rad]

L = 3e-3;
vehicle = Vehicle(m_0 - 600, L, L * 3, 0, T_min, T_max, alpha = alpha);

x_0 = [r_0; v_0; log(m_0)];
x_f = zeros(6, 1);

tspan = [0, (N - 1) * delta_t];
t_k = linspace(tspan(1), tspan(2), N);

u_hold = "FOH";
Nu = (u_hold == "ZOH") * (N - 1) + (u_hold == "FOH") * N;

nx = 7;
nu = 4;

% PTR algorithm parameters
ptr_ops.iter_max = 20;
ptr_ops.iter_min = 2;
ptr_ops.Delta_min = 5e-5;
ptr_ops.w_vc = 1e2;
ptr_ops.w_tr = ones(1, Nu) * 5e-4;
ptr_ops.w_tr_p = 1e-1;
ptr_ops.update_w_tr = false;
ptr_ops.delta_tol = 3e-2;
ptr_ops.q = 2;
ptr_ops.alpha_x = 1;
ptr_ops.alpha_u = 1;
ptr_ops.alpha_p = 0;

scale = true; % scale all variables so that they are around 1
```

Get Dynamics

```
f = @(t, x, u, p) SymDynamics3DoF_linear(t, x, u, 1, alpha, -g(3));
```

Specify Constraints

```
z_lb = @(t) log(m_0 - alpha * T_max * t);
z_lb_k = z_lb(t_k);

% Convex state path constraints
glideslope_constraint = {1:N, @(t, x, u, p) norm(x(1:3)) - x(3) /
cos(glideslope_angle_max)};
min_mass_constraint = @(t, x, u, p) z_lb(t) - x(7);
state_convex_constraints = {glideslope_constraint};

% Convex control constraints
max_thrust_constraint = {1:N, @(t, x, u, p) u(4) - T_max * exp(-z_lb(t)) * (1 -
(x(7) - z_lb(t)))};
min_thrust_constraint = {1:N, @(t, x, u, p) T_min * exp(-x(7)) - u(4)};
lcvx_thrust_constraint = {1:N, @(t, x, u, p) norm(u(1:3)) - u(4)};
control_convex_constraints = {min_thrust_constraint, max_thrust_constraint,
lcvx_thrust_constraint};

% Combine convex constraints
convex_constraints = [state_convex_constraints, control_convex_constraints];

% Terminal boundary condition
terminal_bc = @(x, p, x_ref, p_ref) [x(1:6) - x_f; 0];
```

Specify Objective

```
if u_hold == "ZOH"
    min_fuel_objective = @(x, u, p) sum(u(4, :)) * delta_t;
elseif u_hold == "FOH"
    min_fuel_objective = @(x, u, p) sum((u(4, 1:(end - 1)) + u(4, 2:end)) / 2) *
delta_t;
end
```

Create Guess

Guess doesn't matter since problem is totally convex and the dynamics are linear. Create a straight line guess anyways

```
guess = straight_line_3DoF_rocket_landing_linear(x_0, x_f, N, Nu, delta_t, vehicle,
u_hold);
```

Construct Problem Object

```
prob_3DoF = DeterministicProblem(x_0, x_f, N, u_hold, tspan(end), f, guess,
convex_constraints, min_fuel_objective, scale = scale, terminal_bc = terminal_bc,
discretization_method = "error", N_sub = 1);
```

Solve Problem with PTR

```
ptr_sol = ptr(prob_3DoF, ptr_ops);
```

```
 k |       status       |   vd  |   vs  |  vbc_0 |  vbc_N |    J    |   J_tr   |   J_vc    |   dJ %  |   dx   |   du
 1 |             Solved | 9e-15 |     0 | 1e-15 | 3e-13 | 0.638 | 0.125 | 4.44e-11 |          |  9e+01 |      2 |
 2 |             Solved | 6e-15 |     0 | 1e-15 | 2e-13 | 0.636 | 8.76e-05 | 2.92e-11 | -0.295 |      3 |   0.08 |
 3 |             Solved | 6e-15 |     0 | 1e-15 | 2e-13 | 0.636 | 4.44e-07 | 3.01e-11 | -0.001 |    0.2 |  0.006 |
```

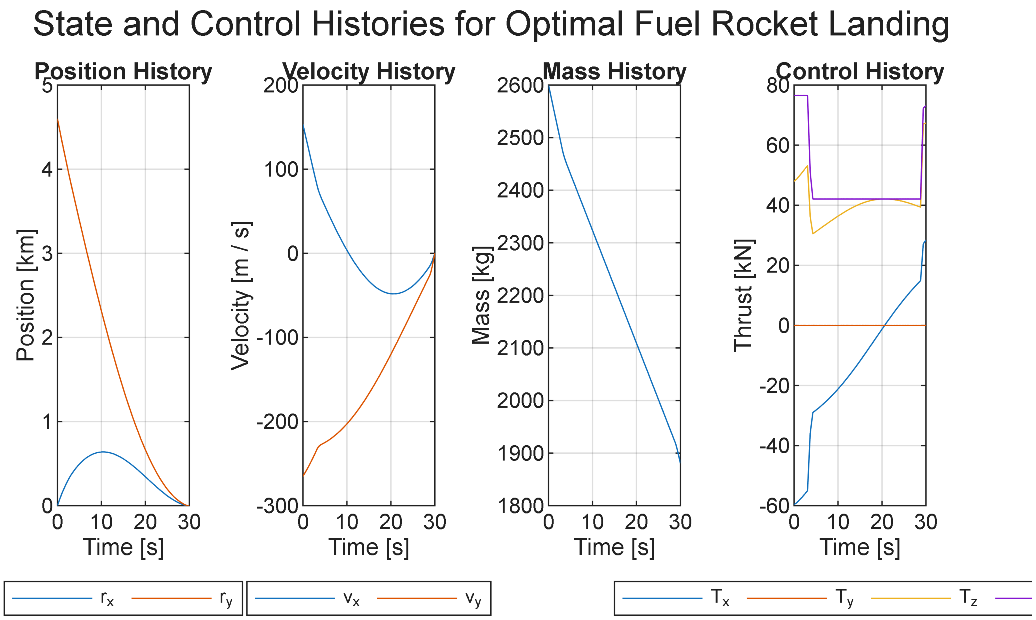
```


% Extract solution
X = ptr_sol.x(:, :, ptr_sol.converged_i + 1);
U = ptr_sol.u(:, :, ptr_sol.converged_i + 1);
p = ptr_sol.p(:, :, ptr_sol.converged_i + 1); % not used for this problem
```

Plot Solution

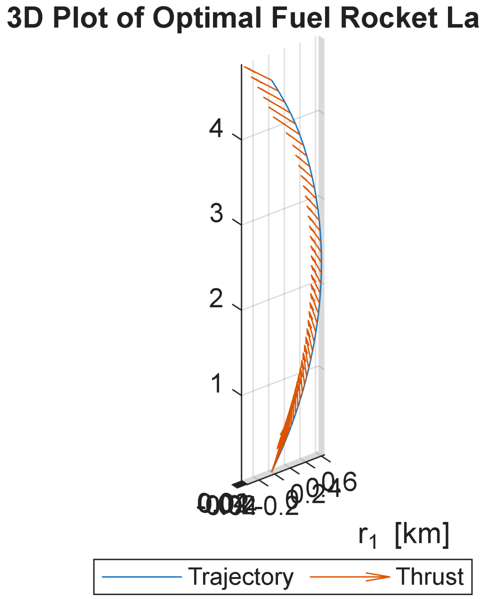
```
% Propagate the solution to make sure the solution is good
[t_cont_sol, x_cont_sol, u_cont_sol] = prob_3DoF.cont_prop(U, p);

% Plot state and control histories
plot_histories_3DoF_trans_rocket_LCvx(t_cont_sol, x_cont_sol, u_cont_sol, u_hold)
```



Plot Solution 2D

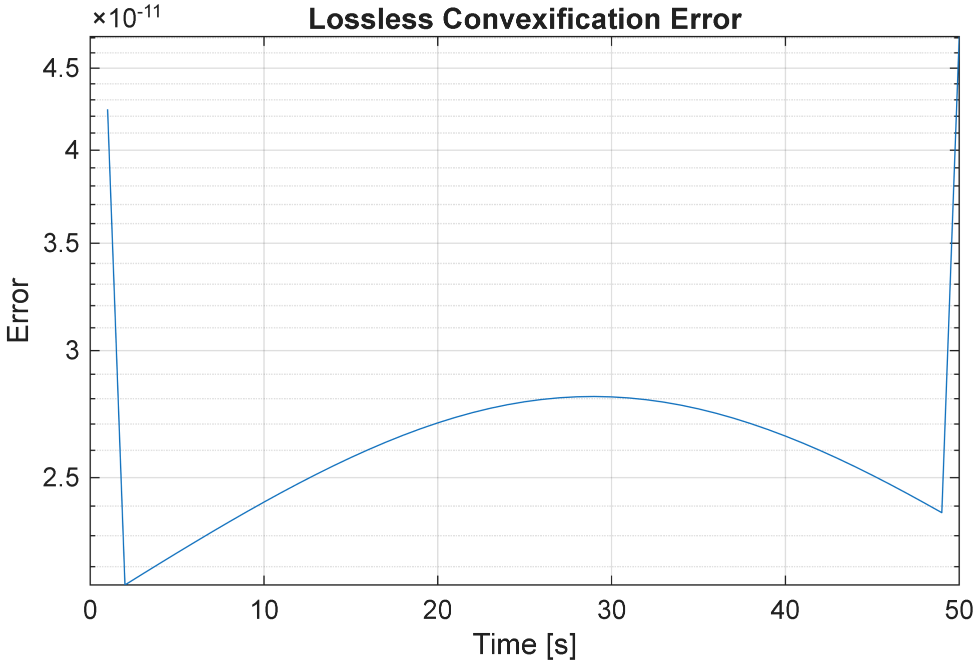
```
plot_trajectory_3DoF_trans_rocket_LCvx(X, U, Nu)
```



Quantify Lossiness from Convexification

```
figure
lcvx_err = abs(vecnorm(U(1:3, :)) - U(4, :));
plot(1:Nu, lcvx_err);
yscale("log")
```

```
grid on
xlabel("Time [s]")
ylabel("Error")
title("Lossless Convexification Error")
```



Helper Functions

```
function [sl_guess] = straight_line_3DoF_rocket_landing_linear(x_0, x_f, N, Nu,
delta_t, vehicle, u_hold)
    sl_guess = guess_3DoF([x_0(1:6); zeros([6, 1])], [x_f; zeros([6, 1])], N, Nu,
delta_t, vehicle);
    sl_guess.x = sl_guess.x(1:6, :);

    % Add mass to guess
    m_0 = x_0(7);
    if u_hold == "ZOH"
        sl_guess.x = [sl_guess.x; m_0 - vehicle.alpha * [cumsum(sl_guess.u(3, :) *
delta_t), sum(sl_guess.u(3, :)) * delta_t]];
    elseif u_hold == "FOH"
        sl_guess.x = [sl_guess.x; m_0 - vehicle.alpha * cumsum(sl_guess.u(3, :) *
delta_t)];
    end

    sl_guess.u = sl_guess.u ./ sl_guess.x(7, 1:Nu);
    sl_guess.u(3:4, :) = [zeros([1, Nu]); sl_guess.u(3, :)];
    sl_guess.x(7, :) = log(sl_guess.x(7, :));
end

function [] = plot_histories_3DoF_trans_rocket_LCvx(t_cont_sol, x_cont_sol,
u_cont_sol, u_hold)
```

```
    tiledlayout(1, 4)
    nexttile
    plot(t_cont_sol, x_cont_sol([1,3], :))
    title("Position History")
    xlabel("Time [s]")
    ylabel("Position [km]")
    legend("r_x", "r_y", Location="southoutside", Orientation="horizontal")
    grid on

    nexttile
    plot(t_cont_sol, x_cont_sol([4, 6], :) * 1000)
    title("Velocity History")
    xlabel("Time [s]")
    ylabel("Velocity [m / s]")
    legend("v_x", "v_y", Location="southoutside", Orientation="horizontal")
    grid on

    nexttile
    plot(t_cont_sol, exp(x_cont_sol(7, :)))
    title("Mass History")
    xlabel("Time [s]")
    ylabel("Mass [kg]")
    grid on

    nexttile
    if u_hold == "ZOH"
        stairs(t_cont_sol(1:size(u_cont_sol, 2)), (u_cont_sol(:, :) .*
exp(x_cont_sol(end, 1:size(u_cont_sol, 2))))')
    elseif u_hold == "FOH"
        plot(t_cont_sol(1:size(u_cont_sol, 2)), (u_cont_sol(:, :) .*
exp(x_cont_sol(end, 1:size(u_cont_sol, 2))))')
    end
    title("Control History")
    xlabel("Time [s]")
    ylabel("Thrust [kN]")
    legend("T_x", "T_y", "T_z", "\sigma", Location="southoutside",
Orientation="horizontal")
    grid on

    sgtitle("State and Control Histories for Optimal Fuel Rocket Landing")
end

function [] = plot_trajectory_3DoF_trans_rocket_LCvx(X, U, Nu)
    figure
    plot3(X(1, :), X(2, :), X(3, :), DisplayName="Trajectory"); hold on
    quiver3(X(1, 1:Nu), X(2, 1:Nu), X(3, 1:Nu), U(1, :), U(2, :), U(3, :),
DisplayName = "Thrust")
    grid on
    title("3D Plot of Optimal Fuel Rocket Landing")
    xlabel("r_1 [km]")
    ylabel("r_2 [km]")
    ylabel("r_3 [km]")
    legend(Location="southoutside", Orientation="horizontal")
    axis equal
end
```