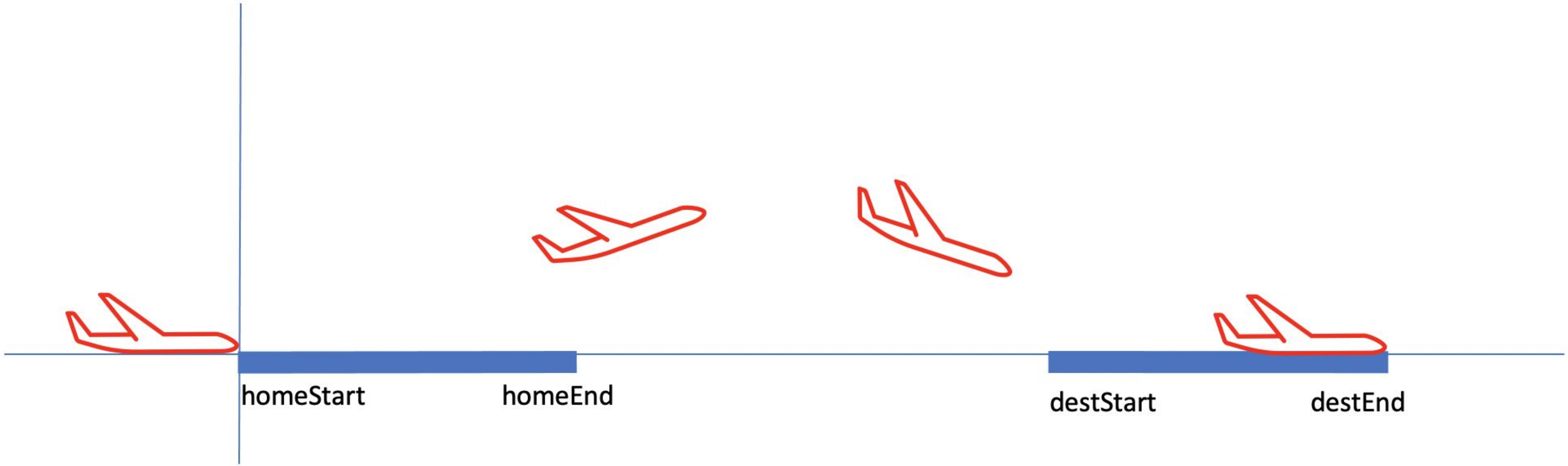
Modeling the Safe Control of an Aircraft in 2-D

Nikita Rupani & Brian Wei
Fall 2020

Motivation

NASA is preparing to fly the latest small plane technologies to demonstrate the advantages of a Small Aircraft Transportation System (SATS). This Lancair Columbia has recently been added to the research fleet of aircraft at NASA Langley Research Center. As envisioned, SATS would support prop and jet aircraft for business and personal transportation for on-demand, point-to-point trips, as well as scheduled service.

Our project



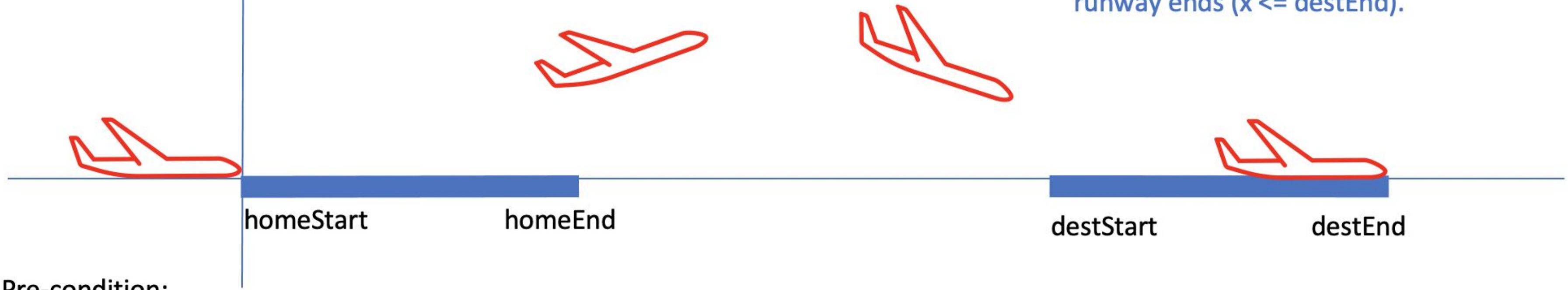
Our project

Safety Property #1:

We lift-off (y>0) before leaving home runway (x <= homeEnd).

Safety Property #2:

We brake (vx = 0) on ground (y = 0) before destination runway ends (x <= destEnd).



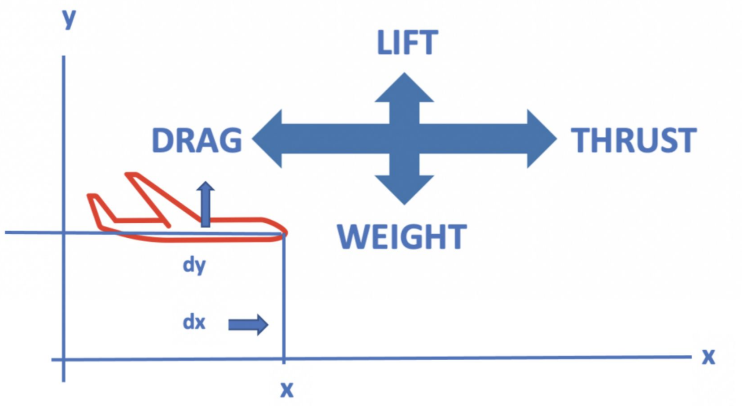
Pre-condition:
Plane is stationary at the origin.

Our project

Pre-condition:
Plane is stationary at the origin.

High-level Model Design

```
(preconditions) ->
[{  takeoff controls
    {takeoff evolution & not taken off}
 }*
 ?(check that we have taken off);
 {
    inflight controls
    {inflight evolution & not landed}
 }*
 ?(check that we have landed);
 {
    landing controls
    {landing evolution}
  }*
](postconditions)
```



Assumptions & Abstractions

- We live in a 2-D world
- No ground friction or air resistance
- No drag during takeoff or thrust during landing allowed
- No collision or interaction

Takeoff

- Takeoff phase is up until the plane is about to leave the ground
- Only need horizontal component of motion

```
t' = 1,
x' = vx, y' = vy,
vx' = (dx * thrust)/planeMass,
vy' = 0,
dx' = 0, dy' = 0
&
t <= T & !liftGEQWeight(vx, L)
```

Takeoff

- Takeoff phase is up until the plane is about to leave the ground
- Only need horizontal component of motion

```
t := 0;
thrust := *;
?(0 <= thrust & thrust <= maxThrust);

L := *;
?(0 <= L & L <= maxL);

if (!liftGEQWeight(vx, L)) {
    ?(
        (liftGEQWeight(vx + thrust/planeMass*T, L)
            & (x + distTravel(vx, thrust/planeMass, T) <= homeEnd))
    |
        (x + distTravel(vx, thrust/planeMass, T) <= homeEnd)
            & canTakeoff(x + distTravel(vx, thrust/planeMass, T), vx + thrust/planeMass*T)
    );
    { ODE }
}
```

Landing

- Landing is once the plane hits the ground, until it comes to a stop
- Only need horizontal component of motion
- Set vertical components discreetly

```
t'=1,
x'= vx, y'= vy,
vx' = -B / planeMass,
vy' = 0,
dx'= 0, dy'= 0
&
t <= T & vx >= 0
```

```
t := 0;
vy := 0;
dy := 0;
dx := 1;

thrust := 0;
L := 0;
B := *;
?(0 < B & B <= maxB);

?(canStopBeforeEnd(x, vx, B));
```

In-flight

- Right after the plane leaves the ground, until it lands
- Dynamics are much more complicated

```
x'= vx, y'= vy,
vx' = dx * thrust / planeMass,
vy' = ((vx^2 + vy^2) * L * C1 - g * planeMass + dy * thrust) / planeMass,
dx'= vx / (vx^2 + vy^2)^(1/2),
dy'= vy / (vx^2 + vy^2)^(1/2)
```

Conclusion

Questions?