Unified Path Following Guidance for hybrid VTOLs

**Supervisors:** [jaeyoung.lim@ethz-asl.ch](mailto:jaeyoung.lim@ethz-asl.ch) [florian.achermann@ethz-asl.ch](mailto:florian.achermann@ethz-asl.ch) [david.rohr@ethz-asl.ch](mailto:david.rohr@ethz-asl.ch) [thomas.stastny@auterion.com](mailto:thomas.stastny@auterion.com) [hnkim@korea.ac.kr](mailto:hnkim@korea.ac.kr)

**End Date:** 2022.03.08

**Regular meeting time:** Monday, 4pm CET

**Report template and wiki:** <https://github.com/ethz-asl/asl-student-templates/wiki>

(Add new meetings to the top)

END

## 

## Apr 11, 2023 - Wrapup

* Meeting with Prof. Kim at KU set for Apr 20, 2023
* Motivation
  + Tackling multiple things at once is hard.
  + Need to anlyze and evaluate one step at a time.
    - I should start understanding the problem first - David
* Overall
  + Feedback: Understanding the problem wasn’t uncear until the middl stage of the projet
    - Maybe because I didn’t takes all the couses already
    - Definition of problem, how to ceck if I solved it systematically would have beennice
    - Evaluaion: Should be clear, wheter it was “done or not”
    - Even in the simplest form of the theory, the problem may already havebeen ‘solved’.
      * Not just engineering, but doing the evaluation
* Implemenation
  + Decent?
* Presentation
  + Motivation was a bit confusing.
  + Otherwise was communicated
* Report
  + Information missing
  + Was in a rush
  + Problem approach
  + **Comments**: For helping out final submission.
* Grade
  + 5.5 / 6.0
    - Lucy
* **My feedback**
  + Encouraging comments
  + For some the harsh comments don’t get taken so well
  + Even if you try and the outcome is minimal, if it’s ‘some’ progress, there won’t be reason to give harsh comments - Florian
  + Even for MS and PhD, it’s all similar
  + Lower expect than MS.
  + BA has to learn a lot of tools and research things, etc. So it makes sense to take more ‘time’.

## Mar 9, 2023 - 17th Week, Final Presentation

### Presentation Material

Presentation can be found here: [Junwoo Hwang | Final Presentation | BA](https://docs.google.com/presentation/u/0/d/1mIPViBMY6p6bgkq33CLrOFOGcFkYFee1oSspzAbUab8/edit)

### Presentation review

## Mar 6, 2023 - 16th Week (Final)

### Progress update

* Kinematic simulation based evaluation: [16\_Final\_Evaluation\_Ground\_Velocity\_Vector\_Fields](https://docs.google.com/presentation/u/0/d/1eZZ8hmKbt09BQcU_sKQEGicRKV8nPBRugR2AwNV5mOw/edit)

### Discussions

* **Hybrid Unicyclic**
  + ~~It is questionable whether actually adopting higher V\_approach in case of high V\_path (> V\_nom). As it doesn’t provide a consistent approaching behavior (e.g. V\_nom).~~
* **V\_nom vs V\_path vs V\_approach**
  + “i think it does not make sense to distinguish between a vnom and vpath as you've defined them. "efficient speed vs user definable speed". this just complicates things. if the user wants a certain speed.. ok, that's the speed. whether we call it nominal or path speed, doesnt matter. it's a single speed setpoint.
  + So if they want efficient speed, then that's the path speed. if they want a faster segment, that's the path speed. it was called nominal in the npfg formulation because npfg was allowed to \*increase the current setpoint above this value only in the extreme wind cases to enable it to \*first be on the path, then try to get as close to the nominally defined speed (desired speed, path speed, all the same) as possible. but this was a mitigation strategy. where the nominal speed would not enable path tracking at the same time. in any case, the naming doesnt matter.
  + what does matter, is that the path speed is a single setpoint that anyone can define, user, path segment, cruise efficiency, etc. \*how the vehicle reaches the path and simultaneously achieves the desired path speed, is an independent question. and that is what you are designing velocity profiles for.
  + Notes
    - V\_approach & V\_path definition only (exclude other variables for presentation)
* Which metrics should I prepare for the presentation?
  + Path track error
  + Time to achieving convergence on path
  + Path length
* Evaluation

### TODOs

* ~~Start with High-level to Low-level, s~~**~~o start with problem of unifying the MC vs FW PF controller~~**~~, instead of starting from unicyclic npfg limitations~~
* ~~Give more context in final presentation on how vehicle capability differs, and why~~
* ~~Talk about ground speed for presentation~~
* Figure out more intuitive figure on showing iterative vector field formulation
* First introduce all the methods, to give overview (baseline method vs proposed **methods**)
* Incorporate trac kerror boundary diagram on the VF
* Discuss how VF methods work
* Showo case where unicyclic PF breaks
* Showcase case where having adjustable V\_path matters
* Showcase 2D animations, instead of 1D
* First showcase: only show the velocity curves only (minimize information)
* Combine legends to increase size
* ~~Change font size to be bigger (plot)~~
* Metrics should be done from first oder integrated
* Evaluation
  + Ideally for different PF formulations, vehicle tuning should also change
* Show different PF formulations in consistent style
* Conclusion
* **Problem definition**
  + MC
    - Approach and Path speed
  + FW
    - Unicyclic
  + ? Do the formulations solve this problem / satisfy the constraints?
* Evaluation comes after checking if it solves the problem (above).

*END*

## Feb 27, 2023 - 15th Week

### Progress Update

* **Evaluation** on the path following ground velocity vector field: [15\_Multirotor\_NPFG\_Formulation](https://docs.google.com/document/u/0/d/15Rz8NqcWSnZr-yy6gBaD0D3bD1l49U7zyp_r5yKA2U8/edit)
* Varying V\_path problem
  + Now we should focus on constant V\_path
  + Change to **state** constant V\_path case.
* Naming of formulations
  + TJ NPFG > ‘Unicyclic NPFG’
* Criteria
  + Track error boundary: Does not matter so much as ‘time to convergence’
  + Can have multiple criterias
    - E.g. Max acc utilization \*could be desired (performance)
      * May be nice for ‘racing quads’, but bad for ‘big drones’, etc.
  + Generally preferable criterias:
    - Monotonicity
    - Achieving V\_path on the path
  + Metrics
    - Don’t show too many metrics
    - Ones to show
      * E.g. achieving V\_path on the path
      * Having 0 velocity on the path (edge case)
    - I can showcase different algorithms and show the pros/cons
* Convergence
  + Should it not converge (t\_conv = infinite)?
  + Asymptotic convergence: ‘Settling time’
    - Set another boundary to determine convergence
  + On the path, when we deviate to left/right, we need immediate acc input?
    - Defines the first order system around the path (kinematic)
  + Time constant
    - Having simulation to calculate the time would be better
* Presentation
  + **Explain why the new formulation is needed**
  + **Hybrid NPFG (Squashed) introduction**
  + For MC: Introduce max acc formulation
  + Speed monoticity: Better to show the curves to explain behavior
* David implementation
  + Defines ground velocity range and takes average for track error boundary calculation
  + Low speed = matching exactly on path
  + Pick up = ground speed range given
  + High speed = Vg VF
  + Having bearing feasibility isn’t necessary (conceptually)
  + Getting off position point when hovering = under certain speed, allows non-yawed multirotor

### Next Steps

* Construct a **storyline** with various cases with plots (Complete **Evaluation**)

### Discussions

* 9th 11-12: Pre-presentation (trial)
* **13th : Presentation**
  + 15min Presentation / 5min QnA
  + Need to introduce topic / be general
* 27th : Actual submission deadline

## Feb 20, 2023 - 14th Week

Weekly progress summarized in: [14\_Quantitative\_Analysis\_of\_Vel\_Curves](https://docs.google.com/presentation/u/0/d/14smjdtz6QsPEF7p_vCwpgjgrVPIJ9Vns8JUFv4poGP4/edit)

### Discussions

1. **Removing V\_tk is ok**, as it indeed shouldn’t interfere since we are only formulating ground velocity curves
2. **Ideally, we would have unified formulation with just ‘V\_approach, V\_path’ as a variable** that can define the velocity curves for both MC and FW
   1. Answer: We can have a velocity cutoff, above which we can consider FW dynamics & under we consider MC dynamics. This will be fastest way to showcase this ‘unified path following’.
   2. Information of V\_nom and V\_min\_approach is at the end calculating sane ‘V\_approach’
   3. For FW, we can set very conservative acceleration (even in longitudinal direction, which is very limited) as the whole acc limit - David
3. Having **mathematical** description of the algorithms would be nice
4. For **Fixed Wing, we should NOT change the V\_nom**, to respect the unicyclic motion (constant speed).
   1. Thus this can be the ‘special’ case of the unified formulation, where the V\_path == V\_approach == V\_nom
5. **TJ NPFG Squashed is basically equal to TJ NPFG when V\_nom is set to the corresponding V\_path** (so V\_nom == V\_path case)
6. **Drawing the centripetal / longitudinal (relative to velocity course direction) acceleration**, instead of relative to ‘path’ unit tangent vector would be interesting
   1. That would reveal more on how much \*forward and \*sideways acceleration the curves are demanding (assuming no wind)
7. In actual application, **for FW it would be desirable to overwrite the V\_path by ‘V\_nom’**, as FW doesn’t have (or we want to assume it has no) capability of changing speed
8. **We can remove the yaw rate limit**, and just focus on acceleration limit, as yaw-rate limit is only relevant to FW, and it is already coupled with acc limit.

### Question

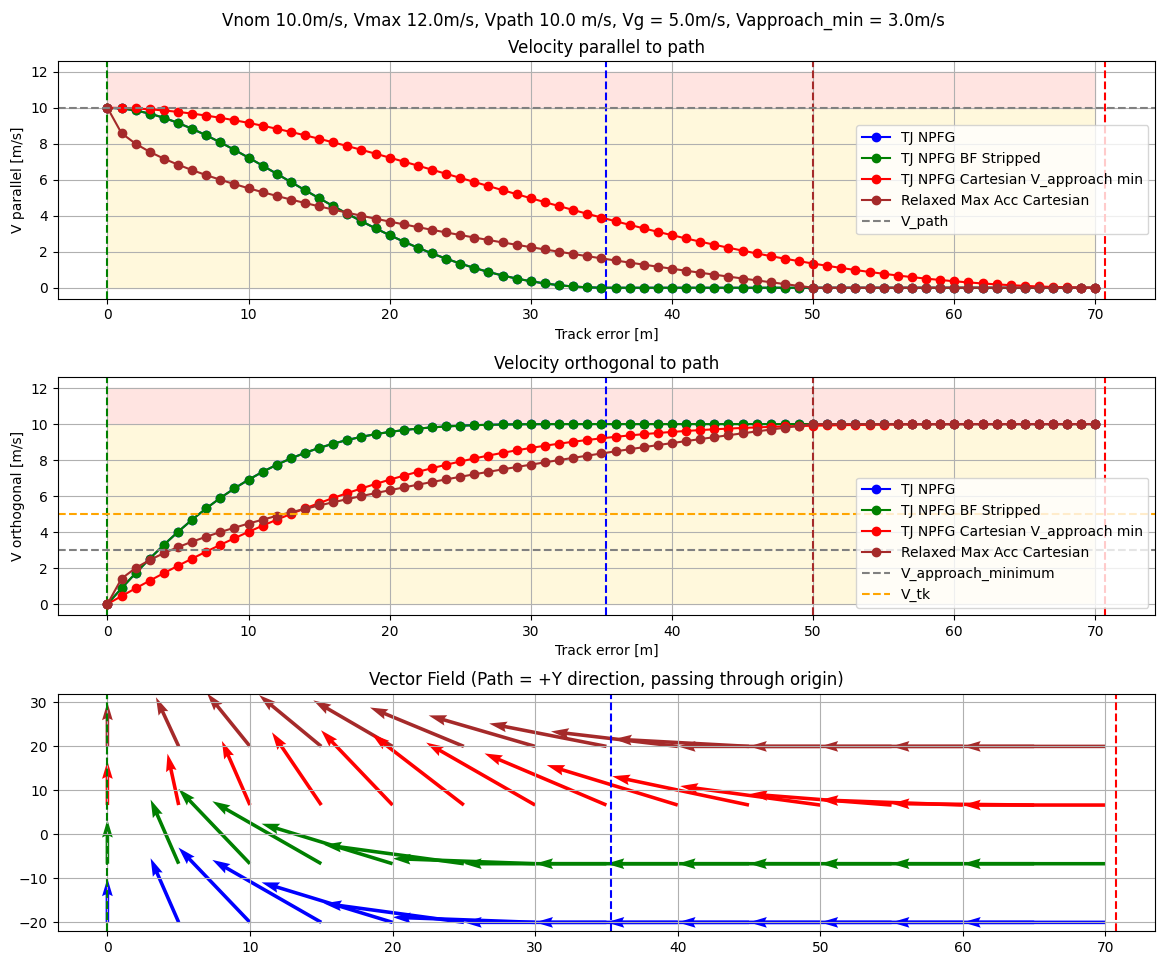
1. Handling path with **curvatures**
   1. ~~TJ NPFG dealed with this by altering look-ahead angle even further, but with current formulation the Squashed formulation would provide a parallel velocity component \*less than V\_path (as it will get rotated further), how do we solve this?~~
      1. Answer: it will be an \*independent feedforward acceleration, and is more relevant for ‘tracking’ regime, not the Vector Field generation.
2. **Handling VTOL transitions**
   1. e.g. when approaching from far, how do we decide which velocity vector field should be given? Or should PF algorithm just know about the [0, V\_max] range, and throw whatever speed, to which vehicle should transition / semi-transition?)
   2. Even so, how do we handle discrepancy in V\_nom for MC / FW? (How do we decide which one to give priority to, in different scenarios?)
      1. Answer: First formulate MC case, then compare method difference on MC vs FW, then check how we can \*generalize them.
3. Considering wind
   1. Now we have rough Ground velocity vector field, but how do we extend this to air-relative frame?
      1. Is this feasible for MC?
      2. Answer: This is low priority as of now. Let’s get analysis right first, then try out jerk limited formulation, then \*maybe airspeed formulation.

### TODOs

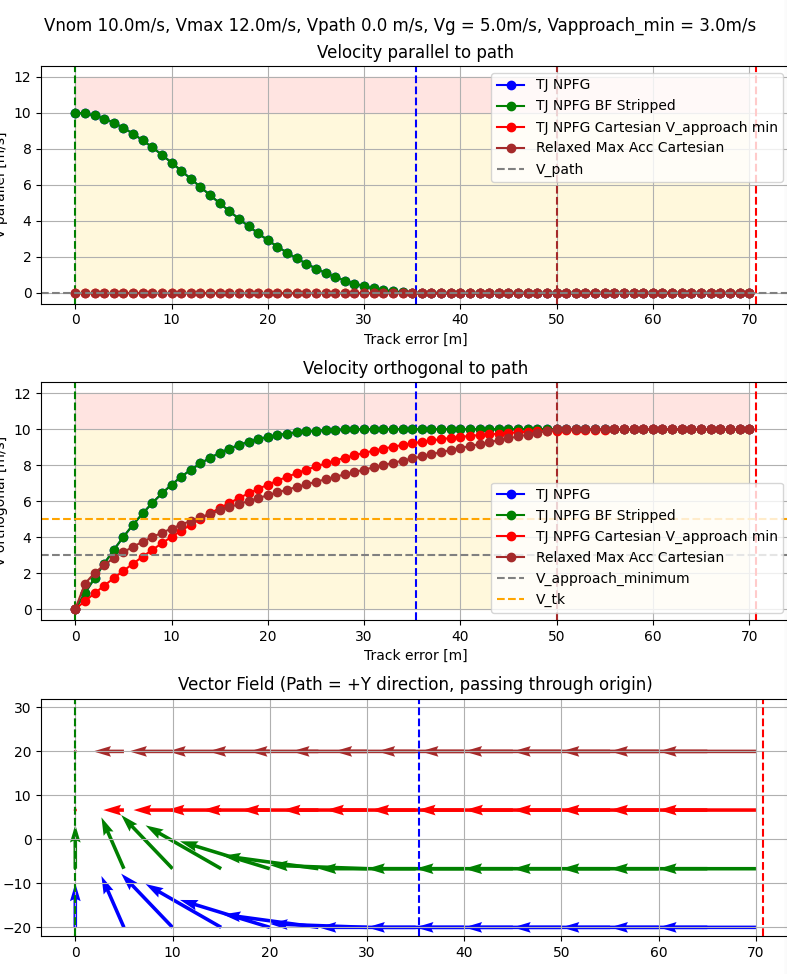
1. Defining the FW/MC mode, how they get differentiated
   1. **Qualitative** measurement of what is better / why
2. Compared original vs squash (extended) for showing improvement
3. Acceleration bound application
   1. Make max acc one monotonic 🤔
4. Definition of the ‘ideal’ case of Velocity Curve (**metric**)
   1. Time / Monoticity /

END

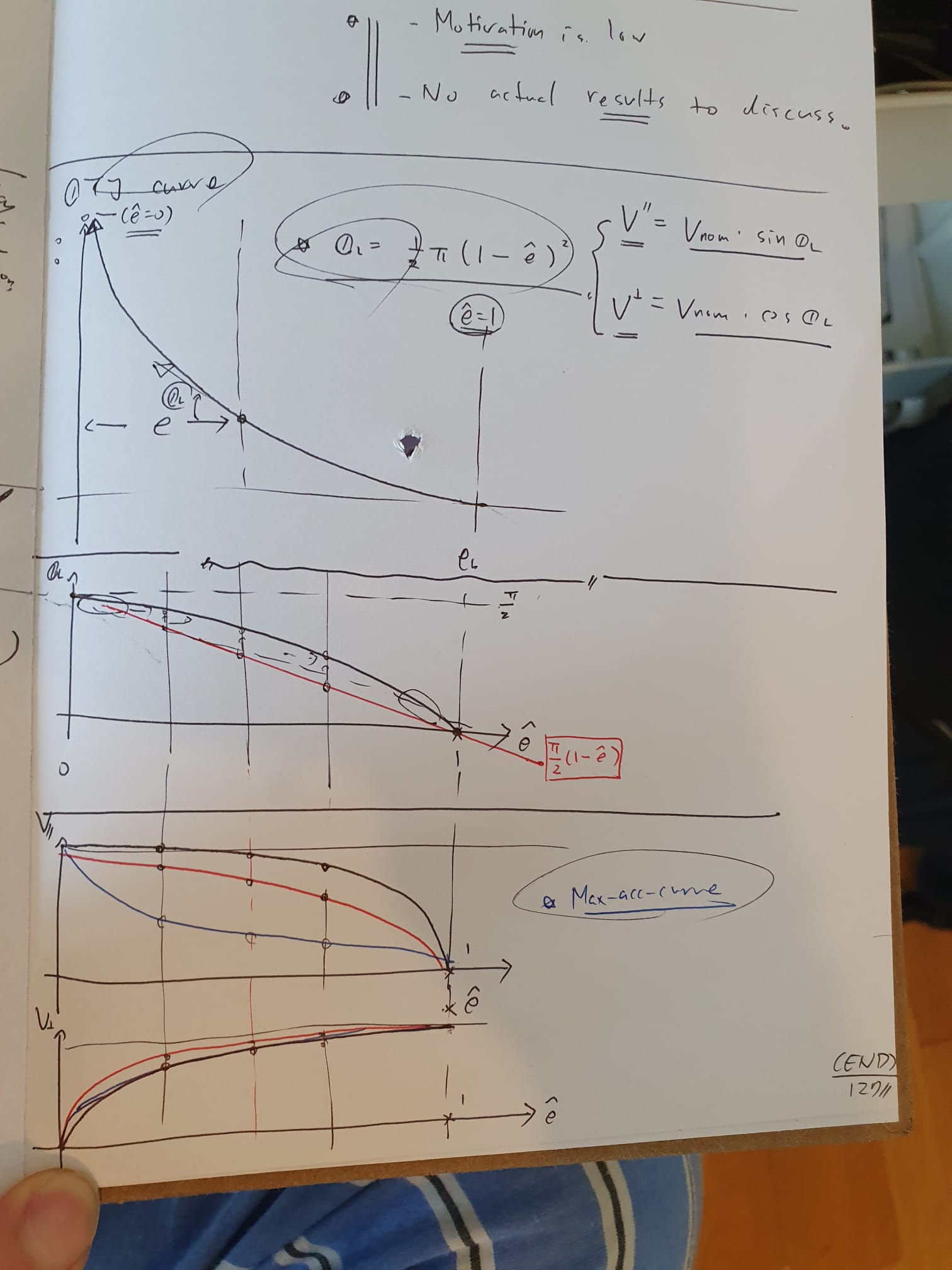
## Feb 13, 2023 - 13th Week

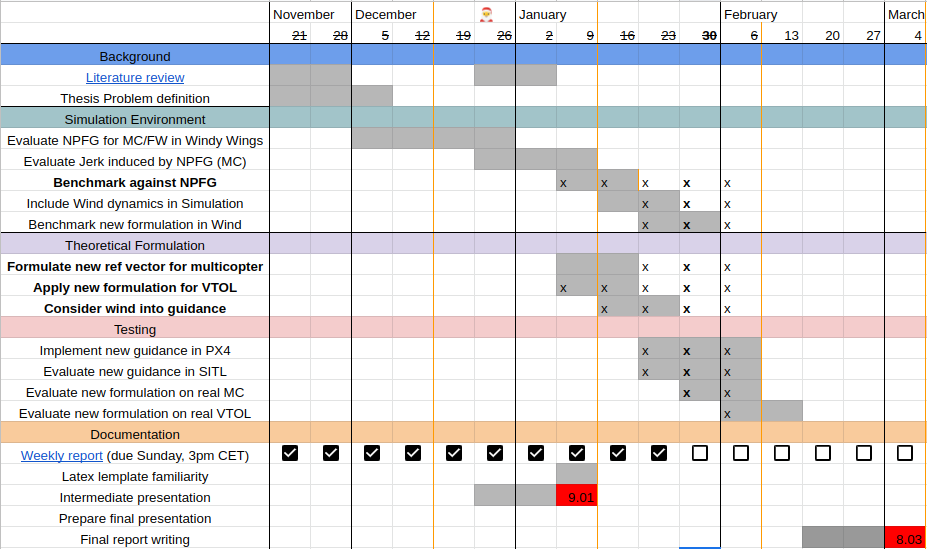


* Vel Magnitude is increasing, why?
* Currently was transitioning the curve based position array to simulation
  + But having crude integration could have been ok - David
* Complexity of formulation
  + With Min approach speed & V\_path, velocity decoupling is already done via NPFG
  + Complexity is in Accel limits
* Squashing formulation suggestion: This is partly done by the Cartesian V approach min method.



* TJ’s NPFG is the one with least track error boundary with low V\_path, as it needs real time vehicle ground speed as input to get track error boundary.
* Analyzing Accelerations would be in any case make sense
* The cartesian V\_approach min ramp in formulation may be exceeding acceleration limit (indicated by brown graph: Max Acc curve)
* Tiltwing VTOL UAV
  + When flying arc, lateral acceleration is high & longitudinal acc = 0
  + So stopping with V\_path = 0 would be very aggressive on longitudinal axis of the vehicle





Thesis ends in less than a month! (4 weeks left)

**Ways to move forward**

* Implemenation aspect
  + PX4 implementation & test real life cases
* **Algorithmic aspect <<<**
  + Wind & Corner cases
  + Velocity Monoticity & Wind
  + Analysis on the Curves
    - Characteristics & Pros/Cons
  + Behavior based on constraint changes (accel)

### 🦾 TODOs for the week

1. **Plots** to have
   1. Path curve (2D)
   2. Acceleration curve
   3. Norm velocity
2. Varying the ratio of velocity constraints and drawing curves
   1. This will give good insight
3. Constraint
   1. Ideal: Longitudinal / Lateral acceleration limit in **body** frame
   2. Certain stiffness around the path (upper limit), to reduce oscillation

END

## Feb 6, 2023 - 12th Week (Skipped, as Junwoo wasn’t available)

1. No update other than drawing more curves 😢

## Jan 30, 2023 - 11th Week: Accel limit math formulation

[Week10\_Summary](https://docs.google.com/presentation/u/0/d/1YukWM_7taye_1Svm4yUeeppgHexCuLdYiNcxM93-1jc/edit)

1. Don’t formulate everything on LaTeX, it takes a lot of time (advice from phD friend)
2. Need to have a curve that would assimilate the NPFG unicyclic curve, not the other way around as the V\_path approaches higher value.

## Jan 23, 2023 - 10th Week

### Incorporating Acceleration limit to Curves

1. By parametrizing the **acc** limit, can derive the time constant for track error boundary, to limit the stiffness
2. Can formulate such that norm of the velocity setpoint vector doesn’t brake as the vehicle approaches that path (**monotonicity**)
3. Can check acc on both parallel / orthogonal to the velocity vector component individually

Can solve for V\_orthogonal first, then come up with most aggressive parallel velocity curve, then can vary the V\_parallel to relax the constraints.

1. Passivity:
2. VF methods with acc limits in previous literature (can reference) / briefly

Next steps (further):

* Acc limits for different vehicles
* Wind (will be more complicated)

### TODOs

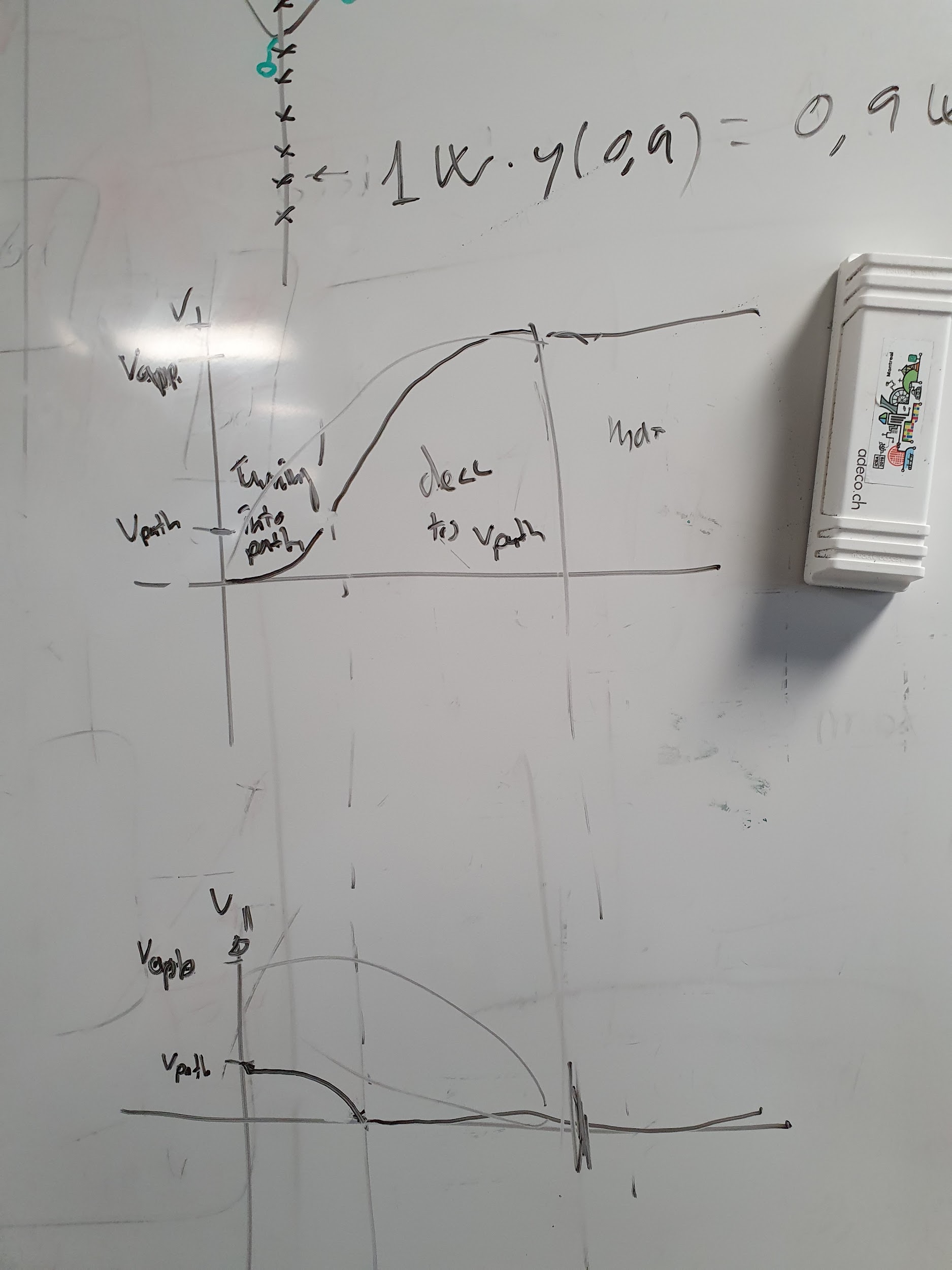
1. Mathematically **formulate** the formulation methods, and evaluate:
   1. **Monotonicity** of the magnitude of the velocity curve as it approaches the path
   2. **Acceleration** induced by the curve, and how it relates to vehicle’s capabilities
2. Based on that, come up with **different** curves that satisfies the above conditions
3. **Benchmark** against original TJ NPFG on metrics:
   1. Convergence time to path
   2. Vehicle brake/accleeration command

## Jan 16, 2023 - 9th Week, focusing again on Vel Curves

### New formulation

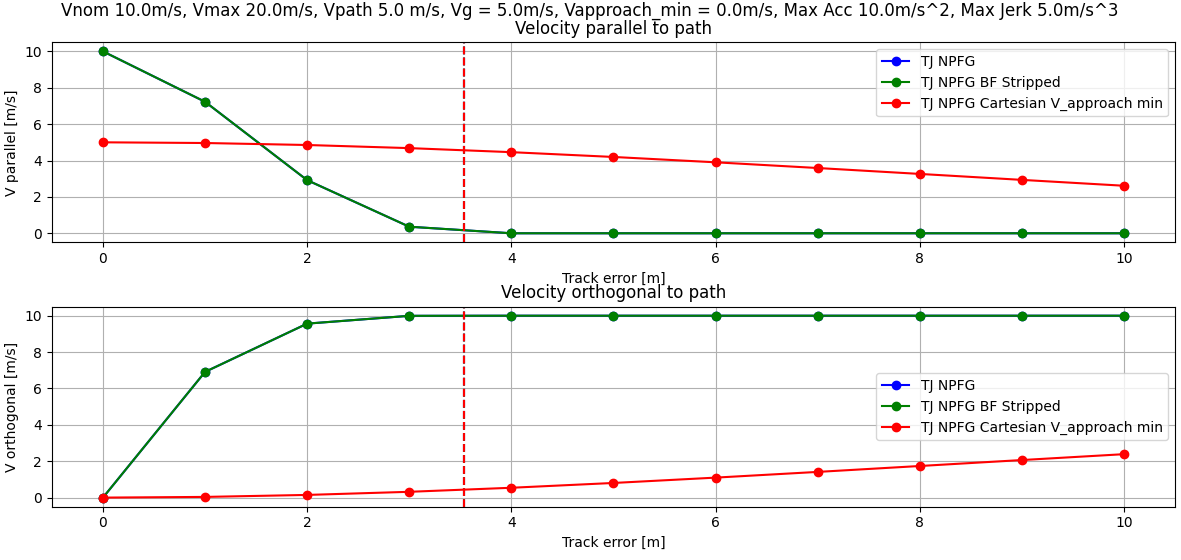
* Vnom vs Vpath
  + If (0, 0), we may even not move, and that may be the correct thing to do.
* V\_approach or V\_ground for **track error boundary formulation**?
* V\_approach\_min
* When we are far away
  + We are guaranteed to approach at minimum, at V\_approach\_min
    - **V\_approach** = max(V\_nom, V\_approach\_min, **V\_path**)
      * We could even ‘lower-bound’ the approach speed to the V\_path. Could be desired.
* The modification will take into effect only when ***V\_path*** *< V\_approach*
  + When **V\_path > max(V\_nom, V\_approach\_min)**, we handle as unicyclic motion
  + When **V\_path < max(V\_nom, V\_approach\_min)**, we handle in cartesian coordinate form
  + The V\_approach is continuous to V\_path as it crosses the boundary
    - Continous but not differentiable
    - Since variables are constants, this doesn’t matter (won’t cause weird behavior)
* Additional constraints
  + V\_path = V\_nom (Fixed Wing)
* The desired metric
  + *Minimizing braking*
  + *No brake & accelerate in the single velocity curve*
* Further questions
  + What happens when ‘Vnom > **Vpath’**? (Need to deccelerate in parallel vel)
    - It won’t be unicyclic motion
* **TODO**
  + ~~VF & show how this algorithm handles different velocity plots~~
  + Make sure the stiffness of the velocity curve has significan amount (so it actually drives the vehicle to the path)
  + David
    - Linear regime used in his project: Is it better than the square-root curve (const acceleration)?
      * Too stiff at origin

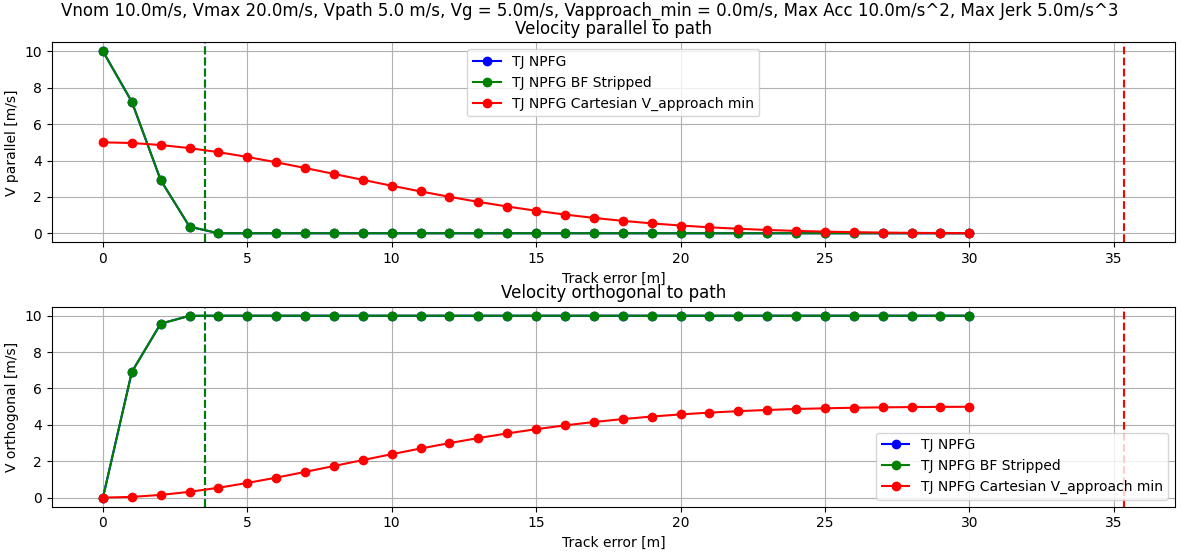
We could even try always finishing up approaching the path in unicyclic motion like this:



Where there is 2 different track error boundaries, where, velocity orthogonal to path follows:

1. Vehicle maintains V\_approach
2. Vehicle slows down to V\_path until certain point
3. Then vehicle ramps in, fully transitioning the V\_path into a parallel component to the path itself.



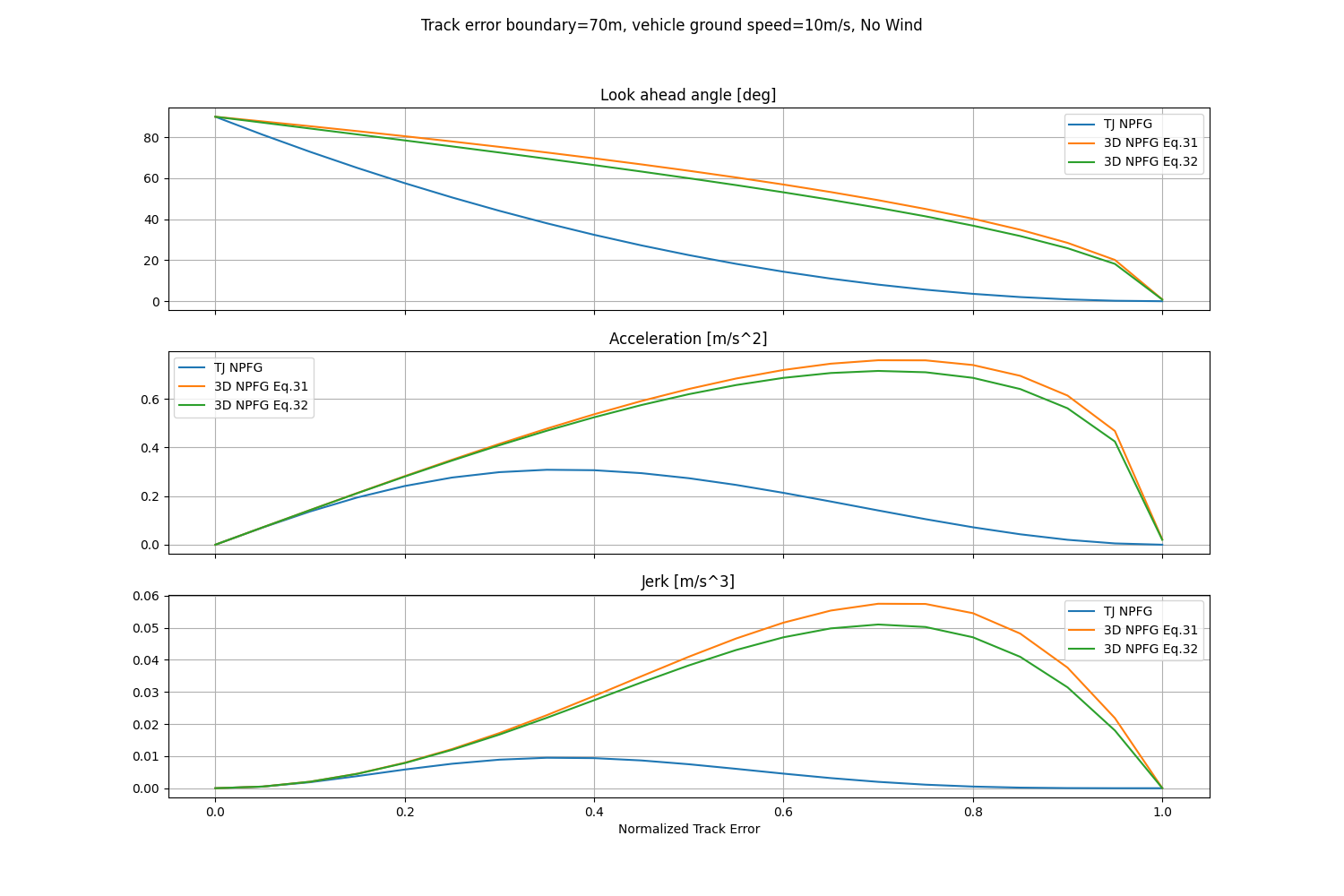


## Jan 9, 2023 - 8th Week, investigation into Jerks

### Intermediate Presentation

* Any feedbacks? [8\_Intermediate\_Presentation](https://docs.google.com/presentation/u/0/d/1gdoeVaushm6uMP61ysq2CldqRS2swrc1veAt2cJ-c5w/edit)

### Jerk limited trajectory in NPFG



Created with [this script](https://github.com/Jaeyoung-Lim/windywings-gym/pull/10/commits/a783e9a4c95ec958126d9b9afa3aabdd28d46e38):

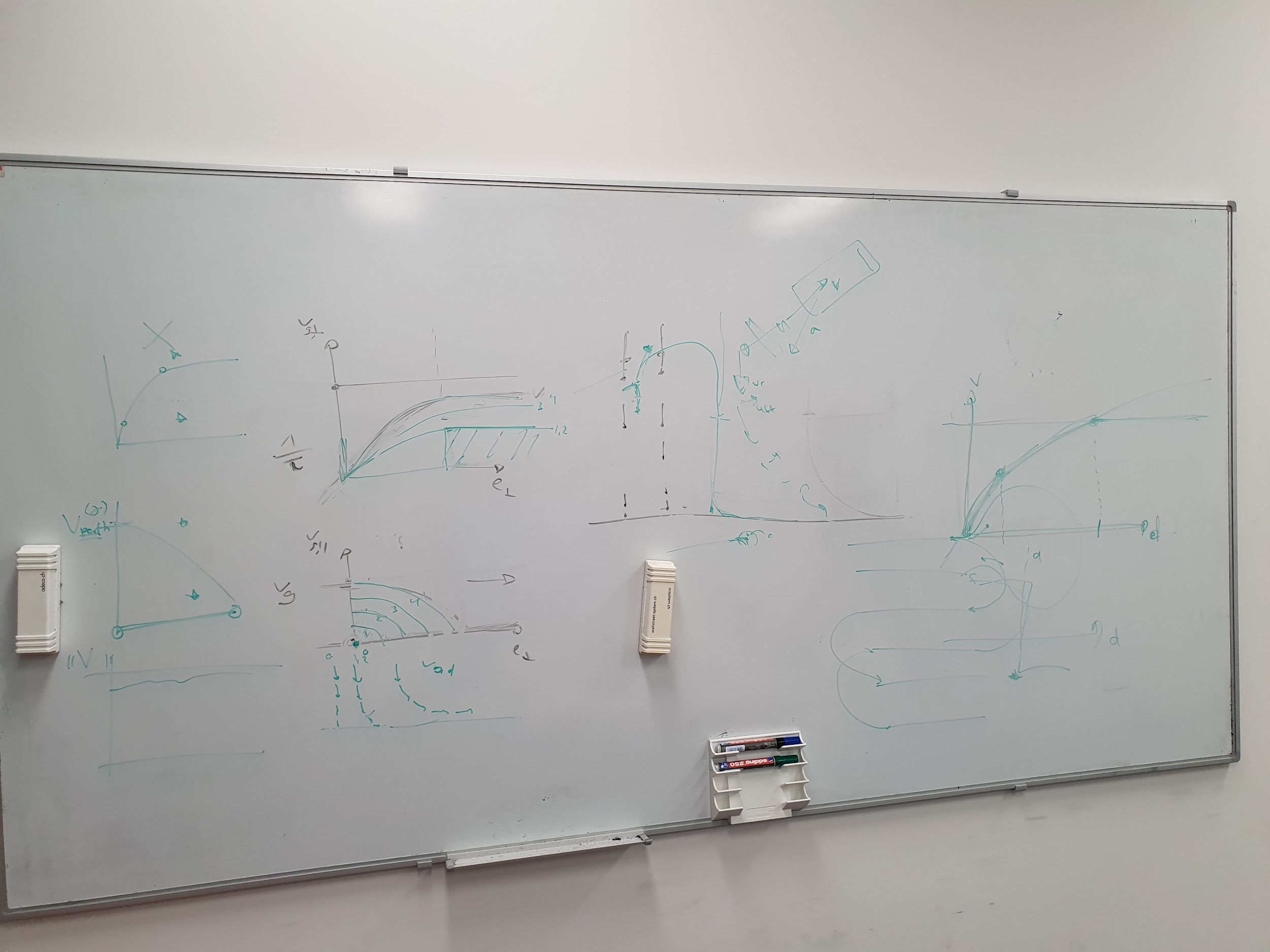
* With a no-wind, constant vehicle velocity assumption, the vehicle’s desired trajectory in NPFG (if followed exactly) requires the above plotted Look-ahead angle, Acceleration, Jerk setpoints to follow the path

### Timeline



* Replace the real vehicle flight with SITL flight, to reduce complexity / time commitment

### Discussions



Above drawing was generated during the discussion of today.

In **Summary**:

1. We can define parallel / vertical (to path) velocity curves that defines the Vector Field around the path
2. We can define the parallel velocity to 0 for a multicopter (to just go and stop at the path).
3. Where vehicle actually is at, shouldn’t affect the guidance algorithm’s Vector Field generation. The curve / field generated is parametrized by vehicle constraints alone.
4. We can define different curves that can have constant magnitude for the vector (fixed-wing case, for cruise-speed), or a changing / fast accelerating case (for multicopter)
5. We can utilize VTOL maneuverability, so that we can choose to come back to path as a FW with high speed, and after that transition into MC and track the path slowly (if desired)
6. VTOL, when transitioned into MC or FW, has a ‘negligible effect’ coming from the opposite configuration (e.g. on a high-speed navigation in MC mode, the FW effect can be ignored, for now / longitudinal acceleration using MC actuators, when in FW mode can be ignored)
7. Path following doesn’t include temporal information. So technically, we shouldn’t really define the ‘desired speed on path’ in the problem definition. The path following behavior should rather be determined by e.g. vehicle nominal airspeed parameter.

### TODOs

* First, make sure the PF guidance itself can work universally (before applying jerk limited vehicle constraints) - MC/FW
  + **Fix** the MC slowing when approaching path, due to bearing feasibility dependency - [Week 4 - Summary](https://docs.google.com/presentation/u/0/d/1ySzieS5Zy0es1_EpFP5weXv18kx0m2MDeSgIchy8acY/edit)
  + **Formulate** Horizontal/ Vertical velocity **curves** (which will be different MC / FW)
    - **Visualize in Vector Field**
      * Consider on track 0 speed / cruise speed
  + Maximum velocity constraint
  + Parametrized behavior on path approaching

### Questions

* [3D NPFG Paper](https://arc.aiaa.org/doi/10.2514/1.G001060), revisited
  + It seems like I can improve the following, is this correct?:
    - Incorporating Wind information
    - Constraint on only applying a ‘lateral’ acceleration to the inertial velocity
      * Is ‘lateral acceleration’ to inertial velocity always feasible?
      * ~~Even if so, is that the optimal solution for PF?~~
        + It is true that lateral acceleration isn’t optimal / the only solution. However, for simplicity and maintaining the airspeed (for optimal cruise of a fixed-wing), the design choice was made.
        + But overall, this for a geometry based PF algorithm like NPFG, shouldn’t be a concern, and the lower level controllers should handle the control to match the desired reference air velocity: David
    - Airspeed setpoint manipulation (vehicle is assumed to move at a somewhat \*constant speed)
* ~~Waypoint reaching characteristics~~
  + ~~Refer to:~~ [~~9\_RealWorldProblems\_of\_PathFollowing~~](https://docs.google.com/document/u/0/d/14qoll_E4UWZVMi8Vwlz4xxujNcvyesCcgKCDO9ctsLc/edit)
  + ~~When using native NPFG with L- look ahead distance constant, vehicle will never really reach the dis-continuous points (‘waypoints’ in missions)~~
    - ~~Is this desired?~~
  + ~~Can we categorize / select what kind of ‘waypoint reaching’ beahvior the algorithm should behave on?~~

*END*

## Jan 4, 2023 - 7th Week, New Year!

### Updates

* Intermediate presentation on next Monday
  + 5 min pres, 5 min question, informal
  + Talk about: What I want to do with thesis & my status in the project & what I plan to do until the end
  + 2 slides (Goal / What I did / What I want to do)
  + Goal: Professor to get to know what’s happening

### Survey Papers (Copied from Weekly Report)

As I didn’t get to devote much time to actual simulations and programming side, I mainly focused on reading more path-following method survey papers, which were:

* “A Survey of Path Following Control Strategies for UAVs Focused on Quadrotors” / [link](http://link.springer.com/10.1007/s10846-019-01085-z) / 2019
* “A Survey and Analysis of Algorithms for Fixed-Wing Unmanned Aerial Vehicles” / [link](https://ieeexplore.ieee.org/document/6712082?reload=true) / 2014
* “A review of path following control strategies for autonomous robotic vehicles: theory, simulations, and experiments” / [link](http://arxiv.org/abs/2204.07319) / 2022

Whom I will refer to as **Quad**, **FixedWing** and **Submarine** papers respectively, as they each focused on the mentioned platforms path following scenarios.

After going through them carefully, I arrived at the following conclusions:

1. All the mentioned PF algorithms don’t consider the wind (external disturbance) estimate
2. TJ’s NPFG (NLGL) is indeed similar to the Vector-Field approach (includes ‘track error boundary’, just like NPFG), but with extra feed-forward acceleration to track curvature
3. Line-of-sight and Pure-pursuit are also similar to NLGL, and it performs well (**FixedWing**)
4. All the algorithms focus on the 2D- path following and augment 3D PF by setting extra height setpoint, but it isn’t embedded into the path-following aspect as coarsely as in the 2D path (**Quad**)
5. The vehicle never reached a velocity of 0 on the path for Quadrotor paper, because the velocity on path was never varied. However, this is where the PF discrepancies would arise (**Quad**)

Therefore I thought that I would like to do the following with the thesis:

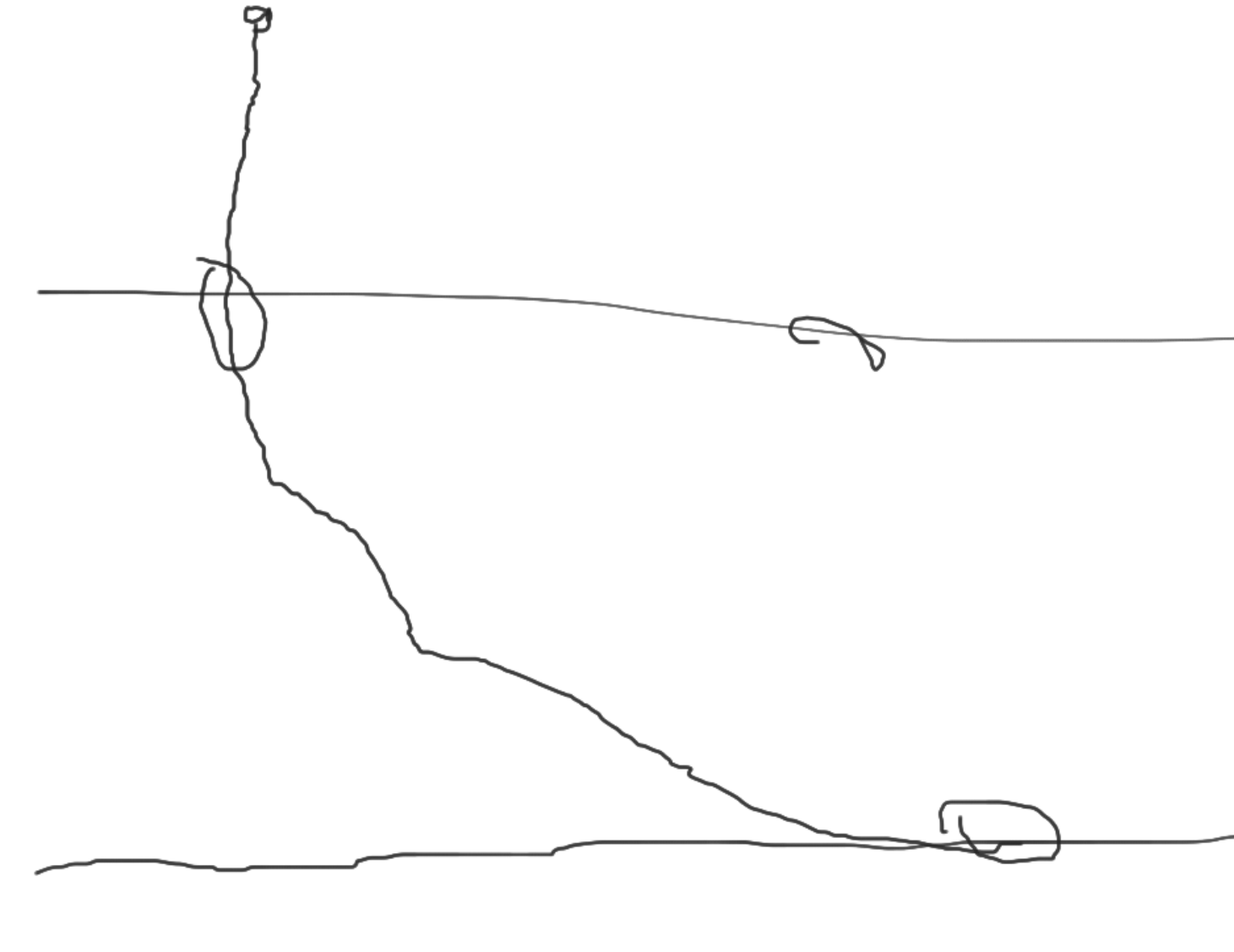
1. First develop the guidance law in a wind-less environment, on a 2D path
2. Have PF be able to deal with 0 velocity setpoint on path
3. Evaluate PF with and without wind estimate

### Questions

1. Can we assume that we can have a wind estimation reliably? Otherwise, how would we deal with this in the PF logic?
   1. Doesn’t matter that much, controller can deal with the external disturbance
   2. **First, should focus on no-wind controller**
   3. The only difference between having wind esimtaion and not, is the airspeed setpoint & yaw setpoint (for fixed-wing).
2. **Do we develop on the 2D path?**
   1. Yes, that will be the focus of the project
   2. 2D -> 3D is straightforward, but for fixed-wing, the vertical dimension dynamics is different from 2D
3. Velocity of 0 on the path
   1. For PF, this doesn’t necessarily makes sense, but to upstream it is relevant.
4. Entry angle to track error boundary
   1. Having it as perpendicular to path: It would be ok to approach (for line-path)
5. L parameter to the NLGL formulation
   1. Previous study had it as a constant (not desired)
   2. For NPFG, we are using VF: The Unit Path Tangent serves a purpose to guide the vehicle,

### 🦾 TODOs

1. Drawing the vector field around the path
   1. Gives sense of how vehicle behaves under different initial conditions
2. Velocity reference vector generation
   1. **Can first start with a pre-existing algorithm**
   2. Try out and during testing, catch the edge cases, and come up with different formulation
   3. Jerk-limited formulation
      1. In a second iteration (future): Can add ‘jerk’ parameter to adjust the path
      2. For now, I don’t have to rigorously monitor the jerk
   4. Speed variation
      1. Current papers/survey assumes a ‘unicyclic’ motion (constant speed)
      2. We have been trying V\_path == 0, as an extreme case.
   5. Track error boundary
      1. It helps vehicle decellerate when approaching the path
      2. With unicyclic motion
         1. We can have ‘minimum turn radius’ as the track error boundary
      3. With quadrotor model
         1. Based on current velocity, we adjust the track error boundary
   6. **Try to find when the current guidance breaks**
3. **Write** down the track error boundary & velocity ramp-in for different vehicles & formulate whether it can be unified
4. Quadrotor: How to deal with bearing setpoint. For now the bearing feasibility part (as showin in last meeting) can be removed for generating velocity setpoint.
   1. Chat with TJ regarding this as well
5. Can send slides of intermediate presentation before next monday for a review



END

## Dec 19, 2022 - Fifth Week

### Summary

Slides: [Week 4 - Summary](https://docs.google.com/presentation/u/0/d/1ySzieS5Zy0es1_EpFP5weXv18kx0m2MDeSgIchy8acY/edit)

### Environment

* More relevant to have acceleration as input to the environment, rather than velocity setpoint
* Multicopter: Should have acceleration input instead of velocity?
  + They are rather controlled via acceleration

### Multicopter vs Fixed-wing

Multicopter: We may want to have a velocity field that is completely orthogonal to the path, instead of ramping-in, along the unit tangent vector

TODO:

* **Draw** the air velocity setpoint on the world (pyplot)
  + Nominal airspeed, Cross track error, Unit Path tangent, Ground speed (can fix it for a single drawing)
  + Can also do ‘random initialized position’ and plot the vehicle position history
* **Velocity reference formulation**
  + Currently: It is using ground speed based track error boundary based bearing-adjustments
  + Current implementation assumes an ‘infinite jerk’, in a sense that bearing setpoint starts turning immediately after entering track error boundary
  + Desired state: Vehicle on the path & moving at nominal airspeed
  + Try first on the ‘cartesian’ air velocity setpoint.
* Ask TJ on how to improve current status of path following behavior
  + Get feedback on what could be ‘better’

### Questions

* Acceleration from look-ahead vector, Acceleration from curvature (feed-forward)
* Acceleration Feed-forward formulation
  + Right now it is assuming yaw rate == ground velocity’s rate of change, following the arc
* Why is the bearing feasibility function formulated in a way it is now?
  + With airspeed buffer

END

## Dec 14, 2022 - Fourth Week

### NPFG Implementation in WindyWings

1. Fixed wing dynamics (lateral acceleration control) & NPFG is fully implemented with no wind.
2. Multicopter dynamics & NPFG is still in testing, with some [multicopter dynamics formulation](https://docs.google.com/document/u/0/d/1lU97lqo_B2BgYdSqMDfGO0l5trjmtktYcmiQ-dZJQ4Y/edit) available.
3. Implementation is [here](https://github.com/Jaeyoung-Lim/windywings-gym/pull/10), as a draft PR.
4. *TODO: Implement Airspeed regulation logic (which would take charge of slowing the vehicle down to nominal speed, e.g. Multicopter -> come to a stop)*

### Simulation results

| **V = 20 m/s, Bearing = PI/4 (Right Up)** | **V = 20 m/s, Bearing = 0 (Right)** |
| --- | --- |
|  |  |
| **V = 20 m/s, Bearing = -3/4PI (Left Down)** | **V = 20 m/s, Bearing = 3/4PI (Left Up)** |
|  |  |
| **Visualization of NPFG** | |
| **\* Gray: Line to closest point on path**  **\* Green: Unit tangent vector (e.g. Going Right)**  **\* Purple: Track error boundary (around closest point on path)**  **\* Pink: Bearing setpoint (from look-ahead angle)**  **\* Black: Lateral acceleration setpoint (on body frame)**  **\* T: Simulation time in seconds**  **\* Acc: Lateral acceleration in m/s^2**  **\* tE: Track Error in meters**  **\* te: Normalized Track Error**  **\* tp: Track proximity (from `te`)**  **\* at: Lateral acceleration from no-curvature assumption**  **\* ac: Lateral acceleration feed-forward from path curvature**  **\* Ax: Resulting Vehicle longitudinal acceleration in m/s^2**  **\* Ay: Resulting Vehicle lateral acceleration in m/s^2** | |

### NPFG on Fixed Wing vs Multicopter

#### ✈️ Fixed Wing

NPFG on a fixed-wing works well, as it was built to do so. It slowly ramps-in the air-velocity reference vector in the direction of the path, as the vehicle enters the track error bound and gets closer to the path.

*Q. However, I did question why the unicyclic control was used, instead of the Velocity^2/L1 type of acceleration command, in the “Gone with the wind” paper.*

#### 🔜 NPFG Lateral acceleration, but to which axis?

For a fixed-wing, the lateral acceleration is applied on the lateral axis in the body frame. Which is possible only because the acceleration is meant to be orthogonal to the air mass relative velocity (air-velocity), which for a fixed-wing is (almost) equal to longitudinal axis in body frame.

But for a Multicopter, as it can have a freely moving heading, this ‘lateral’ acceleration can’t just be applied in the body frame, but should be transformed appropriately.

More discussion on Multicopter NPFG control is continued below.

#### 🛑 Effect of setting Nominal Airspeed to 0.0

Setting the ‘nominal’ airspeed to 0 means:

* When on track, vehicle will come to a stop (maintain the ‘nominal’ speed)

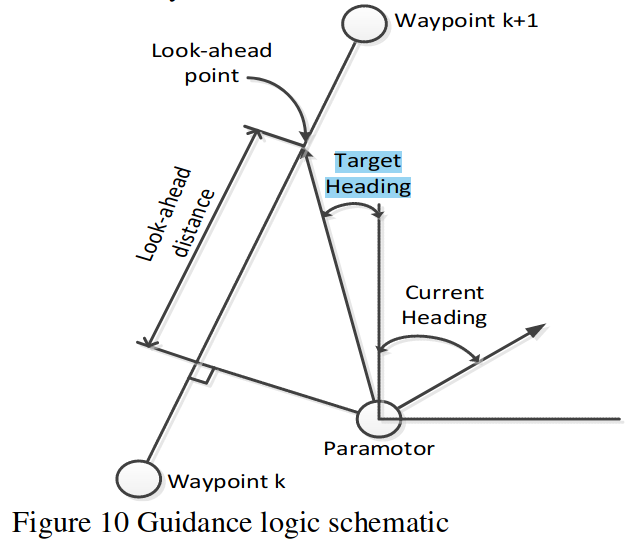
In practice, due to the **unicyclic** control logic inside NPFG on calculating the lateral acceleration proportional to the inverse of nominal airspeed, this results in a blown-up lateral acceleration control command.

Interestingly though, having nominal airspeed set to 0.0 doesn’t have a big effect on the fixed-wing, as the lateral command gets saturated and as soon as the vehicle (heading) catches the desired air velocity reference vector, acceleration gets settled (to 0.0), and excessive control therefore doesn’t occur.

Solution:

* Unicycle can’t stay still just using lateral acceleration! We need to use a different dynamic/control schema for vehicles that can have a low nominal airspeed.

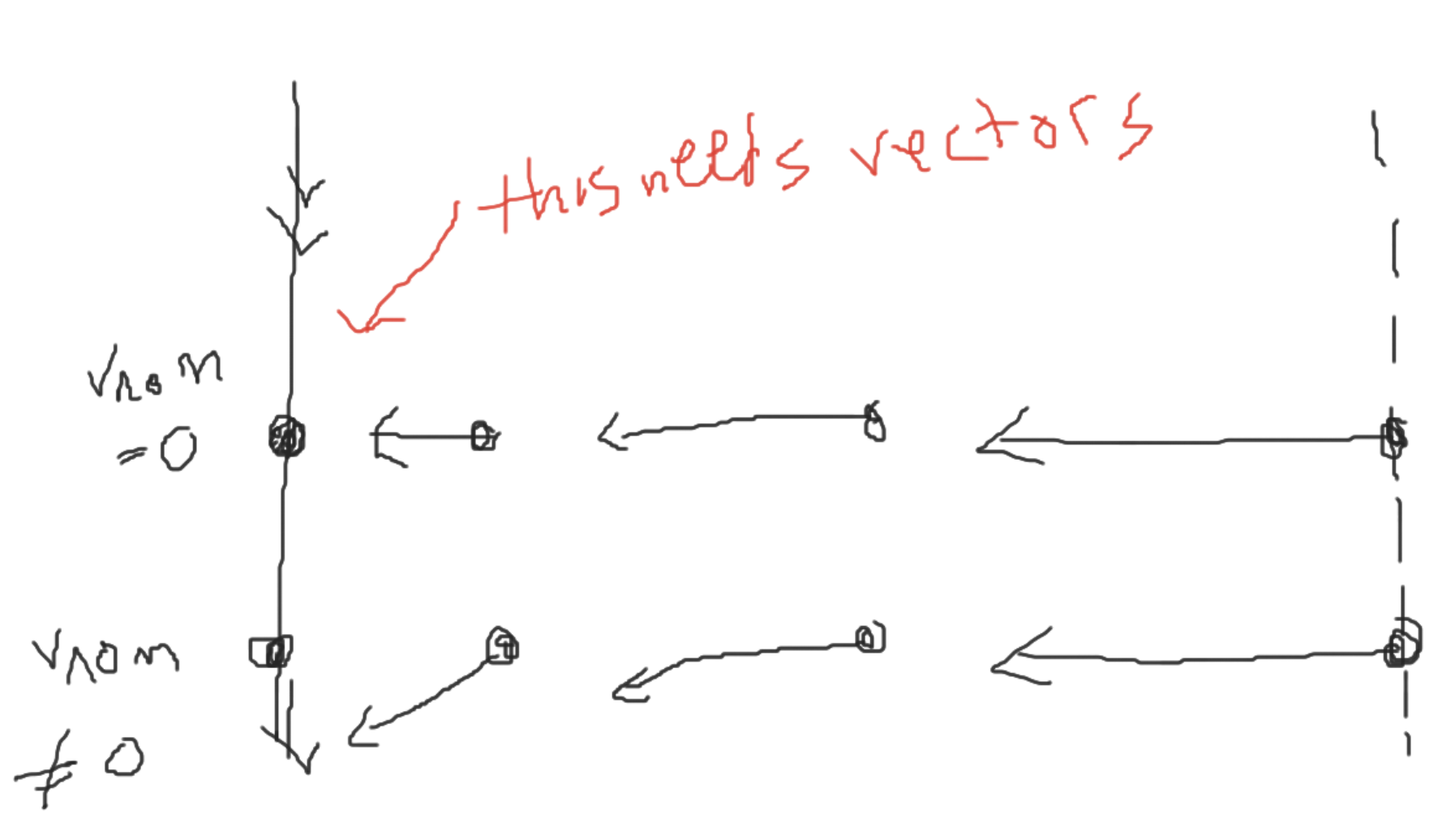
### Discussions

1. If for a multicopter, we set the velocity setpoint, instead of using lateral acceleration control only (for Fixed-Wing), *isn’t this going against the ‘unified’ path following algorithm methodology?*
   1. For a VTOL, wouldn’t this discrepancy result in the jump in the actuator controls when the vehicle is in transition period?
   2. Is it not possible to truly ‘unify’ the control? In some form that any type of vehicle is able to process? (E.g. by having proper bounds/limits on acceleration, etc, can’t we come up with a generic control output?)
2. If we use the Air velocity reference vector from NPFG, and use a velocity controller, how can the **‘stability’** of such control be verified as well?
   1. Like done in TJ’s [Doc for NPFG](https://drive.google.com/file/d/1jW01k246Y7KO57p9eeFZN8tuoAUwVp05/view?usp=share_link)
3. For some systems (e.g. RC Paramotor), the **‘bearing** (deduced from `air velocity ref vector`)’ command is the most relevant control for the path following logic.
   1. ‘Lateral acceleration’ is not always achievable to a same degree that Fixed-Wing can (via initiating a roll), on certain platforms
   2. Would it make sense to provide multiple ‘setpoint’ types (lateral acceleration/ air velocity reference vector)?
   3. Also, having the ‘air mass relative velocity’ correctly modeled would be crucial for solving wind triangles. How is this dealt with in NPFG (for bearing feasibility calculation)? Can we do online estimation on this?
   4. 
      1. Figure from ["Guidance, Navigation and Control of a Small-Scale Paramotor"](https://www.araa.asn.au/acra/acra2012/papers/pap151.pdf)
4. Still confused about the exact concept of ‘setting’
5. Velocity control on MC using e.g. PD vel controller, to follow the ‘velocity reference’ vector
   1. Acceleration should only be used when we \*want to use unicyclic motion (e.g. Fixed-wing)
6. If we set velocity setpoint so that vehicle would always be at the edge of track error bound, the vehicle’s airspeed velocity reference will always be perpendicular to the path, and start to converge to path
   1. Vector field approach (desired velocity vectors)
      1. Derivative of the vector field -> can calculate acceleration feed-forward
         1. But can be ignored for the initial velocity setpoint
   2. Different velocity setpoint calculation algorithm
      1. Maybe we can just ramp-in the
7. [Track keeping in NPFG](https://github.com/PX4/PX4-Autopilot/blob/6c7702b906d7e9dde98e25a4d0cc7af6accc077e/src/lib/npfg/npfg.cpp#L327-L349)
   1. When far away from the track, the ‘minimum ground speed’ will be higher.
8. NPFG doesn’t have to deal with all the different vehicle platforms natively!
   1. We \*can use different reference/setpoints for guidance, depending on vehicle type

### Questions

1. What kind of ‘quality’ metric can we use, to determine which path following algorithm behaves better?
2. **Reference airspeed vs Nominal airspeed, how do we define them exactly?**
   1. In my current script, I can set the ‘reference airspeed’ directly, which causes the diverging / exploding acceleration setpoint, since it is using a unicyclic motion assumption.
   2. PX4> `airspeed\_ref\_` variable
      1. Norm (size) of the `air\_vel\_ref` vector
         1. Calculated from the `refAirVelocity` function, which takes into account min/max/nominal airspeed.
         2. **So, norm value shouldn’t be set manually!!**

### TJ Comments



* Clothoid

### TODOs

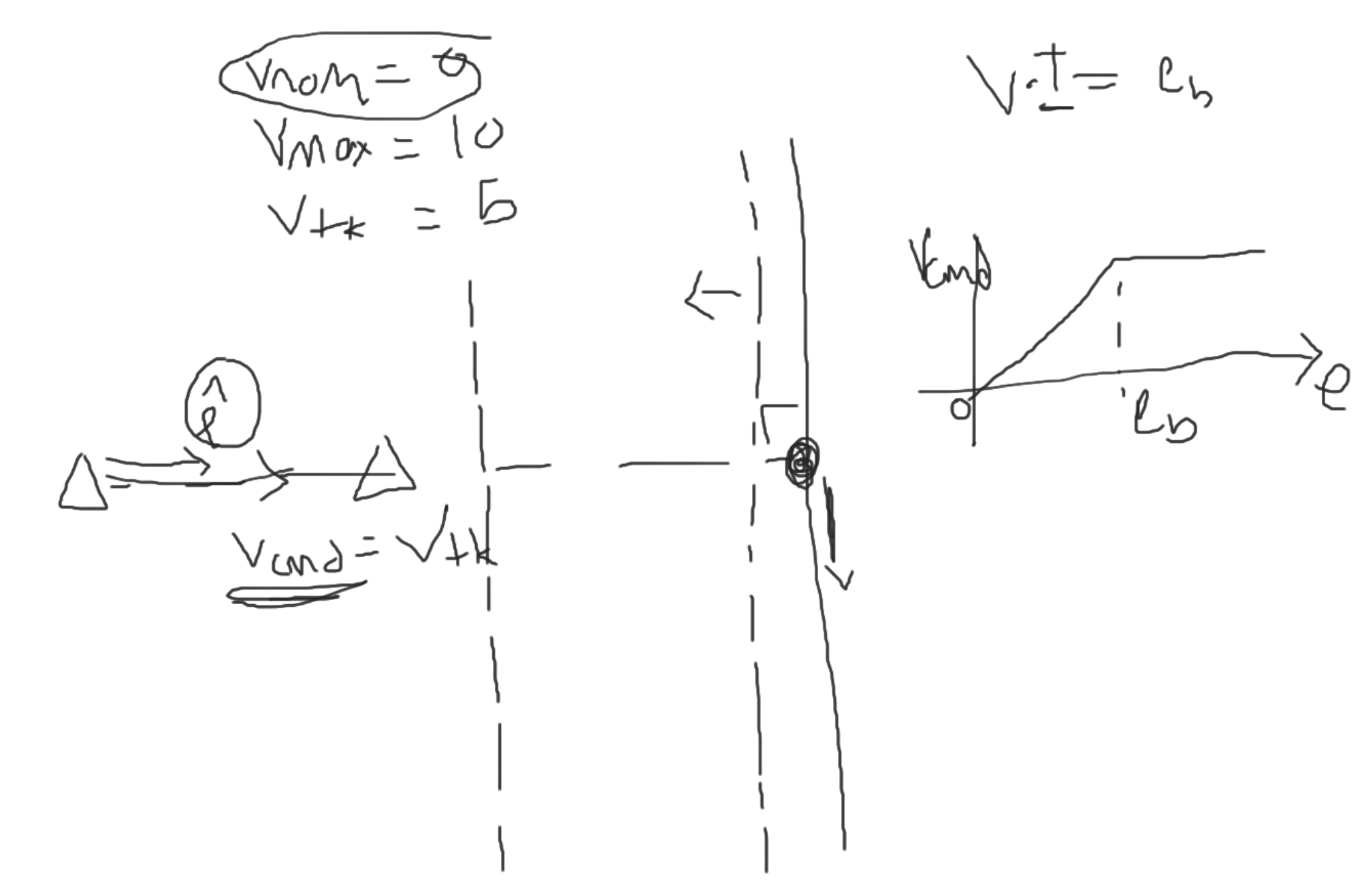
1. Core Thesis
   1. Don’t worry about in-the-middle transition state of a VTOL yet
   2. Don’t worry about jerk limited velocity ramp-in limiting yet
2. Vehicle testing environment
   1. ~~Strip out the rendering from the environment code, and only include vehicle ‘dynamics’~~
   2. Render the graphs at the end of the simulation with matplotlib. It does not have to be real time
   3. ~~My take: Strip out path setpoint from the environment as well! It does not have to do anything with the environment (vehicle dynamics)!~~
3. NPFG
   1. Question what each step of NPFG is doing (with no wind)
   2. Question why certain choices were made, and whether they were optimized for fixed-wing
   3. **Implement track-error based minimum ground speed setpoint (track-keeping) logic.**
      1. This will even prevent the blowup of the acceleration command when the nominal velocity setpoint is 0.0, for a fixed-wing.
   4. Come up with different ramp-in/ramp-out of the path-parallel / path-vertical velocity setpoints
   5. Implement Multicopter dynamics with a point-mass assumption & using velocity-setpoint based navigation

END

## ✍️ Dec 5, 2022 - Third Week

Summary can be found in [🎅JunwooHwang\_WeeklyReports](https://docs.google.com/document/u/0/d/14rIaHRVA3reV4YcAOHXqJK1KwITvM051xEHCWstiejw/edit).

1. We should start with a wind-less environment for formulating the guidance algorithm
2. Path following: If we generate the ‘accleration setpoint’, the end result is achieved.
   1. Exact vehicle control / roll setpoint isn’t related to path following itself
   2. Setting the maximum acceleration boundary is vehicle specific
3. Fixed Wing
   1. Lower / Upper bound for longitudinal velocity
   2. Upper bound for lateral accleration
4. Multicopter
   1. Can ‘stop’ (Fixed-wing isn’t capable of this)
   2. Jerk limited: B/c, rate acceleration setpoint is limited
   3. Can handle different longituidinal
5. **Unified constraints**
   1. Bounds for velocity setpoints
6. TJ Comments
   1. Can try using NPFG out of the box with airspeed setpoint = 0 & omitting velocity error based acceleration feed-back control (abused in NPFG) for multicopter
   2. Nominal / Maximal air ‘speed’ / Airspeed ‘reference’ = output of the guidance
   3. For multicopter, it would \*kind of stop at the waypoint if we set the airspeed reference to 0. This is what ‘track keeping’ is about.
   4. Point of ‘track keeping’ parameter is to ‘enable’ stopping on the track.
   5. It’s an edge case!
7. **What to do for this week 🦾**
   1. Applying NPFG, on a multicopter for evaluating path following performence without a wind (Line-following first, then with curvature)
   2. Use the WindyWings gym environment
   3. Implement NPFG in Windywings (not everything!!!) 🔥
      1. Can omit **winds** for now
      2. With Airspeed ref = 0, we are always assuming of the \*excess wind



### NPFG Analysis Result

1. Input: Wind (2D), Ground Velocity (2D), ~~Airspeed~~ (1D), Path unit tangent vector (2d), Path position setpoint (2D), Curvature, Vehicle position (2D)
2. Output: **Lateral Acceleration (body frame, 1D), Airspeed reference setpoint (1D)**

### Questions

1. Comments in the [NPFG Code Analysis document](https://docs.google.com/document/u/0/d/1fMfKQlW77NaglUZXBY433rGhBw4mINd2B1nGXgftLW4/edit)
   1. (Critical) Wind factor, how is it defined?

### NPFG Limitations/Discussion points (extension from the [weekly report](https://docs.google.com/document/u/0/d/14rIaHRVA3reV4YcAOHXqJK1KwITvM051xEHCWstiejw/edit))

1. Lateral acceleration setpoint
   1. Vehicle dynamics
      1. **If the vehicle is capable of generating a lateral acceleration without rolling, shouldn’t this be taken into account?**
         1. Could we have a control allocation in the global frame, that would then determine which roll (or activating lateral acceleration actuator) / actuator setpoints make the most sense (/ is energy efficient?)
      2. **Why was the logic confined to this ‘lateral acceleration’** (including airspeed compensation logic, which does include ‘forward’ acceleration’) convention?
         1. Is this a extension of the idea from [original L1 paper](https://drive.google.com/file/d/1j1z4ZxkFJLl83ViqZruRpTVKQW0LlP5L/view?usp=share_link)?
         2. Could we have something less limiting (but as simple), for the guidance algorithm?
   2. Roll calculation
      1. The Roll setpoint logic follows the ‘coordinated turn’ convention, where a zero vertical acceleration is assumed (thus, roll = atan(acc/G))
      2. However, in a 3D-path, this assumption is expected to be broken (E.g. on a almost free-fall like behavior for an aggressive path, vehicle should maybe rather roll 90 degrees, to only apply force in horizontal plane)
2. NPFG Implementation

## 

## 🤹 2022.11.28 - Second Week

1. Terminology
   1. Pure pursuit / Line Of Sight
      1. Need to read the reference paper, if I need to understand this (duh, not enough time invested 🤦)
      2. “G. Conte, S. Duranti , and T. Merz, “Dynamic 3D path following for an autonomous helicopter,”
   2. **L1 adaptive control != l1 guidance**
      1. L1 norm is a terminology: <https://montjoile.medium.com/l0-norm-l1-norm-l2-norm-l-infinity-norm-7a7d18a4f40c>
         1. Lk norm = sum (e^k for e in elements)!
   3. NPFG is a generic term
      1. We can refer to “On flying backwards” paper for now
      2. E.g. changing behavior when the vehicle approaches the path
         1. We define time constants, etc for tuning the controller
2. Scope of Literature Review
   1. Trying to understand everything is time consuming
   2. “L1 Adaptive Path-Following of Small Fixed-wing Unmanned Aerial Vehicles in Wind”
      1. Suspected to be a paper that is describing a problem that was already solved in another paper, 10 years ago
      2. Need to be careful not to spend too much time on these kinds of papers. Need to take them with a grain of salt.
      3. Always check the year it was published as well. Most likely there’s another paper that solved the exact same problem.
3. Path following algorithms
   1. L1
   2. Vector Field: NPFG is also included
      1. NPFG is likely good enough to use
4. Problem
   1. Challenge of unifying the guidance
      1. Step 1: Velocity control problem
      2. Step 2: Yaw control problem (6 DoF pose considered)
5. TODOs
   1. **Draw a diagram of NPFG on fixed wing & multicopter, with clear input/output**
      1. Understand the concept, Consider different cases/tweaks for different vehicle types
      2. What is the limiting factor for using the NPFG for Quadcopter?
      3. *E.g. Minimum Ground Speed Requirement manipulation*
      4. Figure out where the ‘fixed wing’ constraint comes into the Fixed-Wing
         1. E.g. Constant velocity assumption
      5. What happens when we set ground speed to 0, and set a desired path, does NPFG follow that?
         1. Desired Ground speed / Airspeed set to 0?
            1. TJ: Yes, it can work

There’s a feasibility option for 0 airspeed (NaN)

Bearing may not be weird (control may not be implemented)

Quad: Need to check if it would work well (unsure)

Would be nice to have a jerk limited calculations (example)

* 1. **TODO: Check PX4 NPFG code, for up to date implementation**
     1. <https://github.com/PX4/PX4-Autopilot/blob/main/src/lib/npfg/npfg.cpp>

## 

## 🤸‍♂️ 2022.11.21 - Kickoff meeting

1. ~~ETHZ myStudies: Book 15 ECTS Course (16 is typical)~~
2. **Big thanks to Thomas & Florian for organizing this Scholarship!!! - Junwoo**
3. **Unified path following**
   1. How to generalize the guidance, regardless of the vehicle, depending on maneuverability
   2. David: Guidance of hybrids in progress
      1. Velocity envelope / Flight state
      2. Yaw rate control (may or may not have control authority)
4. Feel free to add some literature papers for reference (to TJ)
   1. Non-[Holonomic](https://en.wikipedia.org/wiki/Holonomic_constraints) (can’t accelerate / move in any step input direction) control.
5. **TJ**: We can start with ‘kinematics’, instead of low-level
   1. Logic would remain consistent (multicopter doesn’t \*require forward speed, but logic for control can be applied same as for Fixed Wing)
   2. \*chosen function: (e.g. Lateral control for NPFG) can be chosen differently, depending on the vehicle.
   3. High level: (Input: Path) / (Output: Bearing command)
      1. **Having velocity setpoint / bearing vectors that ramps-in**
         1. FixedWing: Like a car & airspeed is more relevant (and mostly stays \*fixed)
         2. Multicopter: Groundspeed is more relevant (e.g. staying on one place)
         3. We could have depending on ‘speed’, switching the logic (e.g. Fixed wing, which can result in low speed can take a long time to reach a point of interest)
         4. Like Jerk-limited control of PX4 (Speed regulation), we can do \*intermediate transition to a fixed-wing, etc.
            1. (PX4) Motion planning library: How do we change the bearing?

Currently it’s a quadratic function.

Using Accel & Jerk limits

* + 1. Next level: How do we regulate this error
  1. Sticky point: We may not have enough space for the transition / velocity moderation
  2. Speed:
     1. 3rd chapter of (NPFG) TJ Dissertation would be a good reference
  3. Could be fun to pipe-in the PX4’s logic to simulate could be fun

1. **David**:
   1. Had a hack to have bearing setpoint modified?

* Goals for next week:
  + **David**
    - Draw up a diagram explaining potential guidance law he has in mind
  + **Student**:
    - Read references, and define:
      * Vehicle constraints
      * Which information is needed for Guidance
    - **Define “what I am trying to do”**
      * I can reference papers to get ‘ideas’ on how to solve my problem.
      * I can try out NPFG on a normal quadcopter in SITL as well
    - Ask myself: Can I use NPFG in Multicopter?