

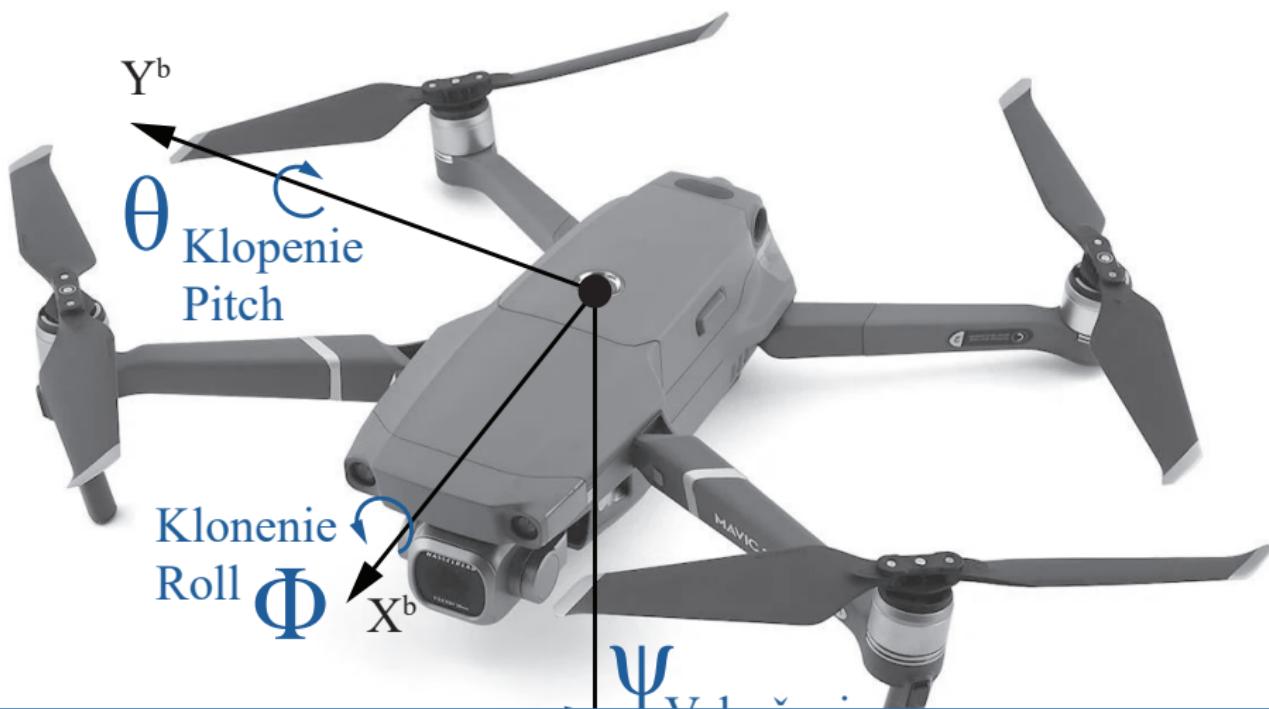
Working title drone control

Riadenie dronov

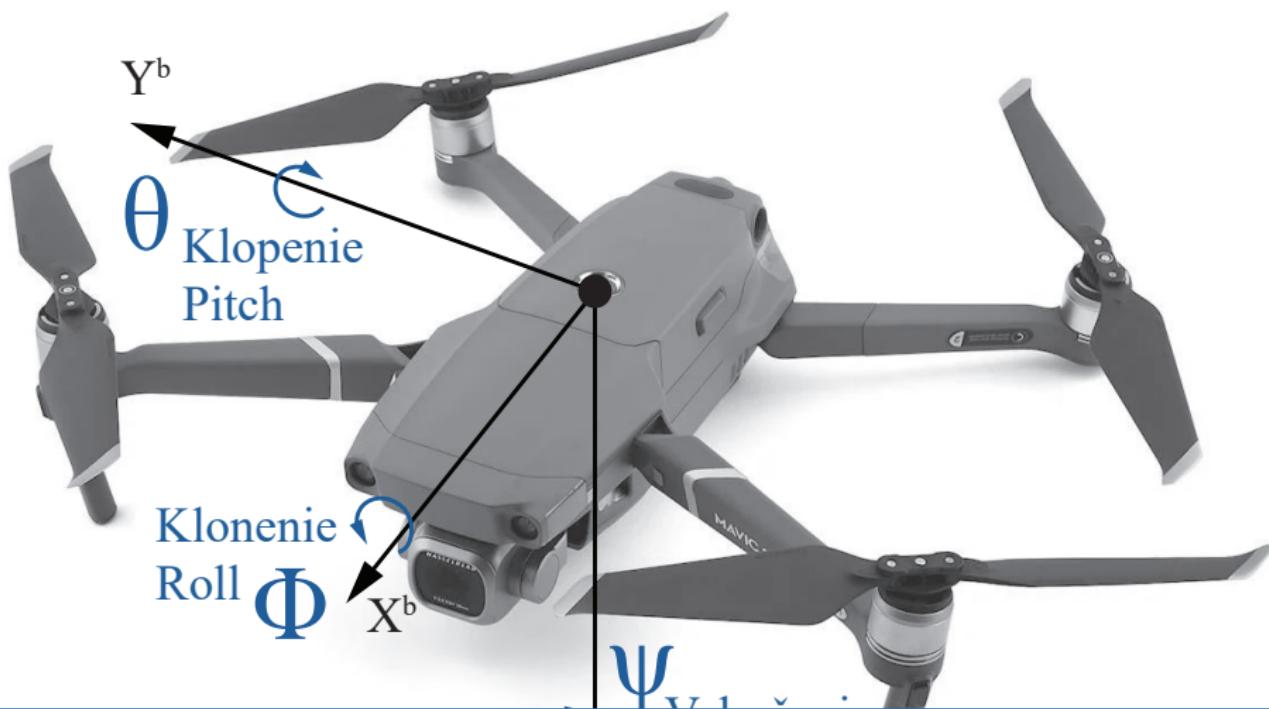
prof. Ing. Gergely Takács, PhD.



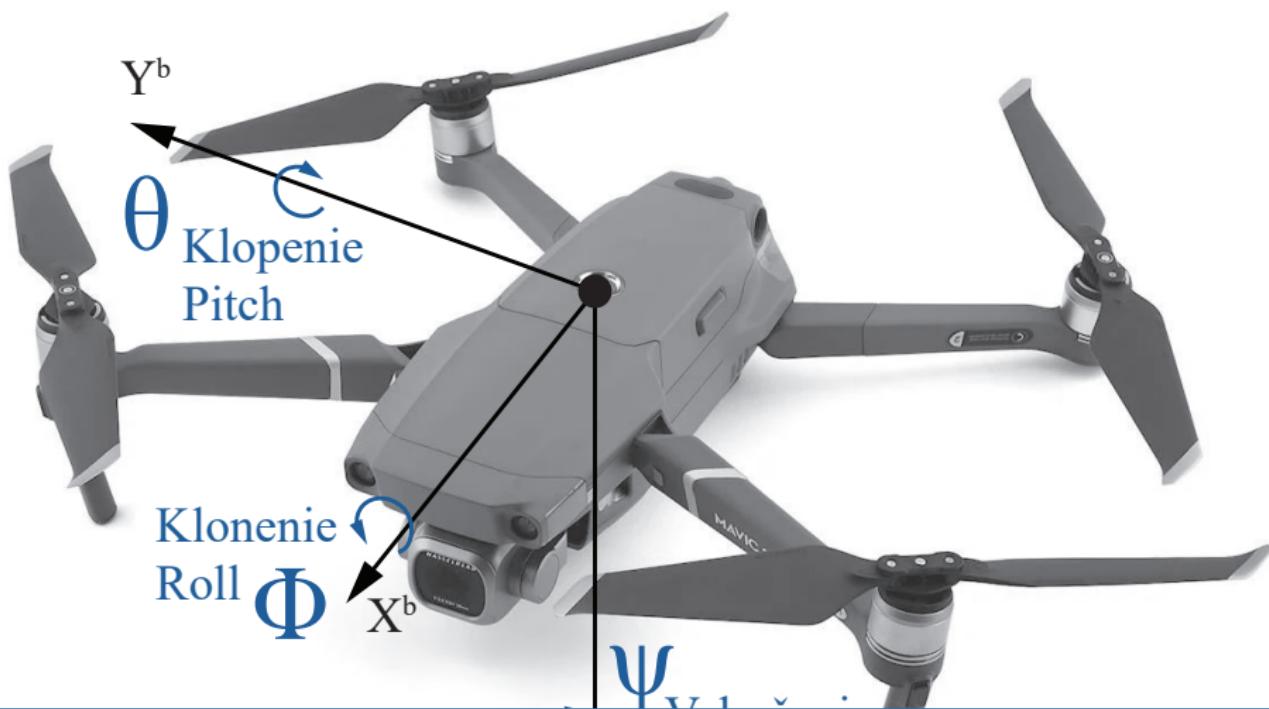
- Nestabilná (*angl.*: unstable) - duh...
- Nelineárna (*angl.*: nonlinear) - uhly, atd'.
- (*angl.*: underactuated) - 4 vstupy, 6 DOF výstup [Douglas 2018]



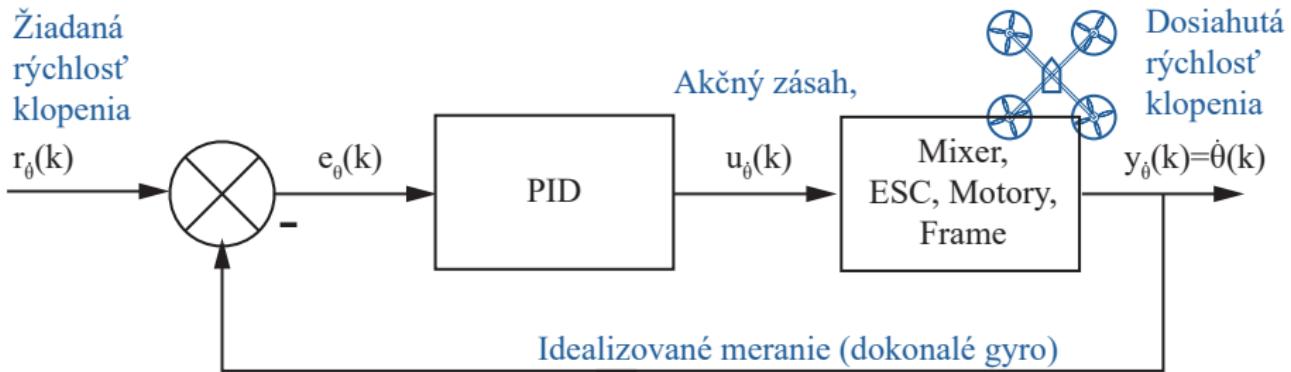
- Nestabilná (*angl.*: unstable) - duh...
- Nelineárna (*angl.*: nonlinear) - uhly, atď.
- (*angl.*: underactuated) - 4 vstupy, 6 DOF výstup [Douglas 2018]



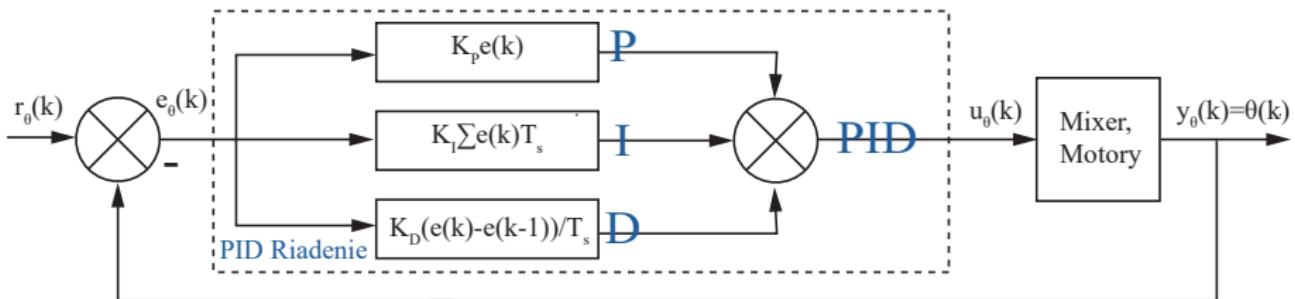
- Nestabilná (*angl.*: unstable) - duh...
- Nelineárna (*angl.*: nonlinear) - uhly, atď'.
- (*angl.*: underactuated) - 4 vstupy, 6 DOF výstup [Douglas 2018]



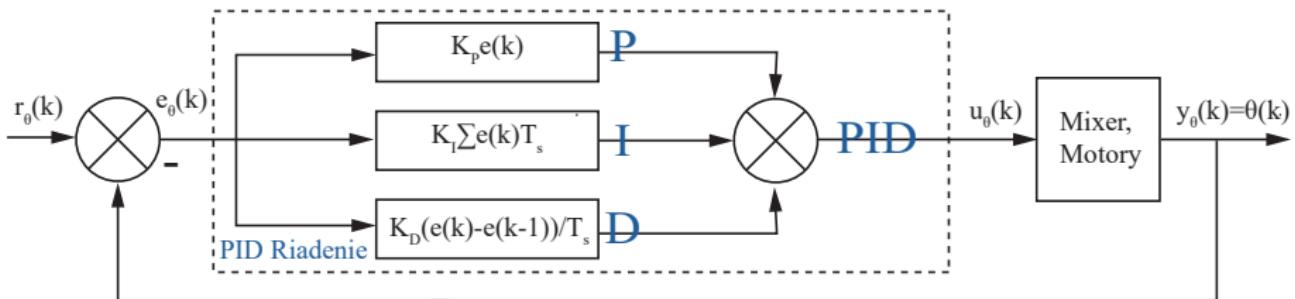
- Tvárame sa, že poznáme žiadanú uhlovú rýchlosť orientácie  $r_{\dot{\Theta}}$ ,  $r_{\dot{\phi}}$  a  $r_{\dot{\psi}}$  a pre jednoduchosť sústredme len na rýchlosť zmeny klopenia ( $\dot{\Theta}$ ).
- Sme na najnižšej úrovni, t.j. riadenie uhlovej rýchlosť (angl.: rate controller) — je to zároveň aj najdôležitejšia slučka, a máme 3 nezávislých slučiek [Hall 2018; PX4 Autopilot 2021b] .
- Je to aj najrýchlejšia slučka (cca.  $f_s=400\text{--}1000\text{ Hz}$  [Hall 2018; PX4 Autopilot 2021b] ) Meranie je komplexný problém sám v sebe, napr. low-pass gyro + notch [Bresciani a kol. 2020] alebo odhad



- Tvárame sa, že poznáme žiadanú uhlovú rýchlosť orientácie  $r_{\dot{\Theta}}$ ,  $r_{\dot{\phi}}$  a  $r_{\dot{\psi}}$  a pre jednoduchosť sústredme len na rýchlosť zmeny klopenia ( $\dot{\Theta}$ ).
- Sme na najnižšej úrovni, t.j. riadenie uhlovej rýchlosť (*angl.*: rate controller) — je to zároveň aj najdôležitejšia slučka, a máme 3 nezávislých slučiek [Hall 2018; PX4 Autopilot 2021b].
- Je to aj najrýchlejšia slučka (cca.  $f_s=400\text{--}1000$  Hz [Hall 2018; PX4 Autopilot 2021b] ) Meranie je komplexný problém sám v sebe, napr. low-pass gyro + notch [Bresciani a kol. 2020] alebo odhad



- Tvárame sa, že poznáme žiadanú uhlovú rýchlosť orientácie  $r_{\dot{\Theta}}$ ,  $r_{\dot{\phi}}$  a  $r_{\dot{\psi}}$  a pre jednoduchosť sústredme len na rýchlosť zmeny klopenia ( $\dot{\Theta}$ ).
- Sme na najnižšej úrovni, t.j. riadenie uhlovej rýchlosť (*angl.*: rate controller) — je to zároveň aj najdôležitejšia slučka, a máme 3 nezávislých slučiek [Hall 2018; PX4 Autopilot 2021b].
- Je to aj najrýchlejšia slučka (cca.  $f_s=400\text{--}1000$  Hz [Hall 2018; PX4 Autopilot 2021b]) Meranie je komplexný problém sám v sebe, napr. low-pass gyro + notch [Bresciani a kol. 2020] alebo odhad



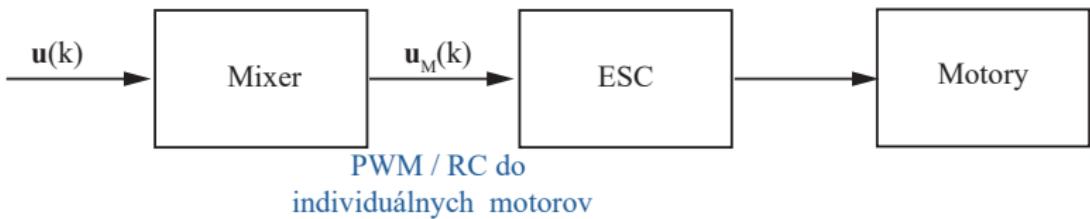
- Aj keď slučky PID na tejto úrovni definujú žiadanú zmenu  $u(k)$ , musíme implementovať pridelenie riadenia (*angl.*: control allocation)  $u_M(k)$  (PWM, RC impulzy) do individuálnych motorov cez elektronické riadenie rýchlosťi (*angl.*: electronic speed control, ESC).
- V najjednoduchšom prípade máme mapujeme pre každý motor 1-4 (alebo viac)

$$u_{M1}(k) = u_\tau + u_\phi + u_\dot{\theta} + u_\dot{\Sigma} \quad (1)$$

$$u_{M2}(k) = u_\tau - u_\phi + u_\dot{\theta} - u_\dot{\Sigma} \quad (2)$$

$$u_{M3}(k) = u_\tau + u_\phi - u_\dot{\theta} - u_\dot{\Sigma} \quad (3)$$

$$u_{M4}(k) = u_\tau - u_\phi - u_\dot{\theta} + u_\dot{\Sigma} \quad (4)$$



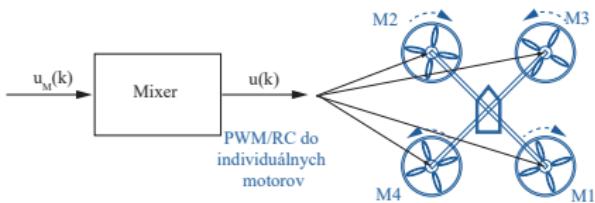
- Aj keď slučky PID na tejto úrovni definujú žiadanú zmenu  $u(k)$ , musíme implementovať pridelenie riadenia (*angl.*: control allocation)  $u_M(k)$  (PWM, RC impulzy) do individuálnych motorov cez elektronické riadenie rýchlosťi (*angl.*: electronic speed control, ESC).
- V najjednoduchšom prípade máme mapujeme pre každý motor 1-4 (alebo viac)

$$u_{M1}(k) = u_\tau + u_\phi + u_\theta + u_\Sigma \quad (1)$$

$$u_{M2}(k) = u_\tau - u_\phi + u_\theta - u_\Sigma \quad (2)$$

$$u_{M3}(k) = u_\tau + u_\phi - u_\theta - u_\Sigma \quad (3)$$

$$u_{M4}(k) = u_\tau - u_\phi - u_\theta + u_\Sigma \quad (4)$$





$$\begin{bmatrix} u_{M1}(k) \\ u_{M2}(k) \\ u_{M3}(k) \\ u_{M4}(k) \end{bmatrix} = \underbrace{\begin{bmatrix} +1 & +1 & +1 & +1 \\ +1 & -1 & +1 & -1 \\ +1 & +1 & -1 & -1 \\ +1 & -1 & -1 & +1 \end{bmatrix}}_P \begin{bmatrix} u_\tau \\ u_\phi \\ u_\dot{\theta} \\ u_\dot{\Sigma} \end{bmatrix} \quad (5)$$

- Mapa  $u_M(k) = Pu(k)$  kde  $P$  matica pridelenia riadenia (*angl.: control allocation matrix*). Nemusí obsahovať iba jednotky, môže byť aj neštvorcová ale našťastie transformácia je stále lineárna [Bresciani a kol. 2020].
- Môžeme priamo namapovať maticu efektivity akčných členov (*angl.: actuator effectiveness matrix*)  $B$ , kde  $P = B^{-1}$ . Ak počet akčných zásahov = počtu akčných členov, používame inverziu ináč napr. Moore-Penroseovu pseudoinverziu  $P = B^\dagger$  [Bresciani a kol. 2020].



$$\begin{bmatrix} u_{M1}(k) \\ u_{M2}(k) \\ u_{M3}(k) \\ u_{M4}(k) \end{bmatrix} = \underbrace{\begin{bmatrix} +1 & +1 & +1 & +1 \\ +1 & -1 & +1 & -1 \\ +1 & +1 & -1 & -1 \\ +1 & -1 & -1 & +1 \end{bmatrix}}_P \begin{bmatrix} u_\tau \\ u_\phi \\ u_\dot{\theta} \\ u_\dot{\Sigma} \end{bmatrix} \quad (5)$$

- Mapa  $u_M(k) = Pu(k)$  kde  $P$  matica pridelenia riadenia (*angl.: control allocation matrix*). Nemusí obsahovať iba jednotky, môže byť aj neštvorcová ale našťastie transformácia je stále lineárna [Bresciani a kol. 2020].
- Môžeme priamo namapovať maticu efektivity akčných členov (*angl.: actuator effectiveness matrix*)  $B$ , kde  $P = B^{-1}$ . Ak počet akčných zásahov = počtu akčných členov, používame inverziu ináč napr. Moore-Penroseovu pseudoinverziu  $P = B^\dagger$  [Bresciani a kol. 2020].

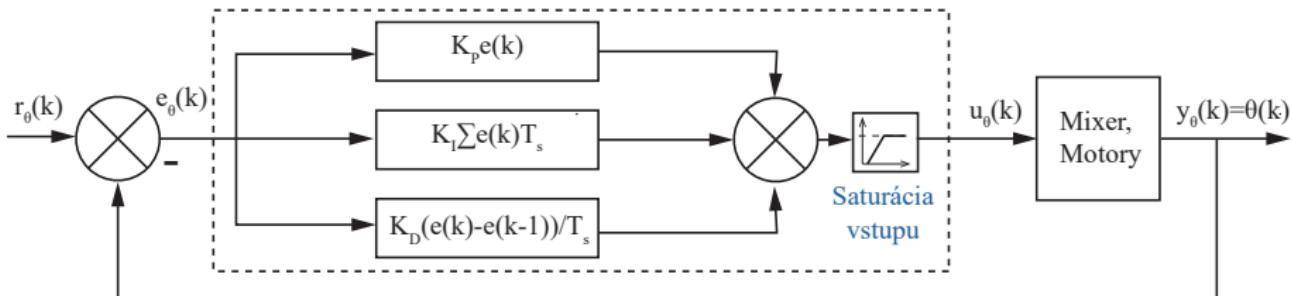


$$\begin{bmatrix} u_{M1}(k) \\ u_{M2}(k) \\ u_{M3}(k) \\ u_{M4}(k) \end{bmatrix} = \underbrace{\begin{bmatrix} +1 & +1 & +1 & +1 \\ +1 & -1 & +1 & -1 \\ +1 & +1 & -1 & -1 \\ +1 & -1 & -1 & +1 \end{bmatrix}}_P \begin{bmatrix} u_\tau \\ u_\phi \\ u_\dot{\theta} \\ u_\dot{\Sigma} \end{bmatrix} \quad (5)$$

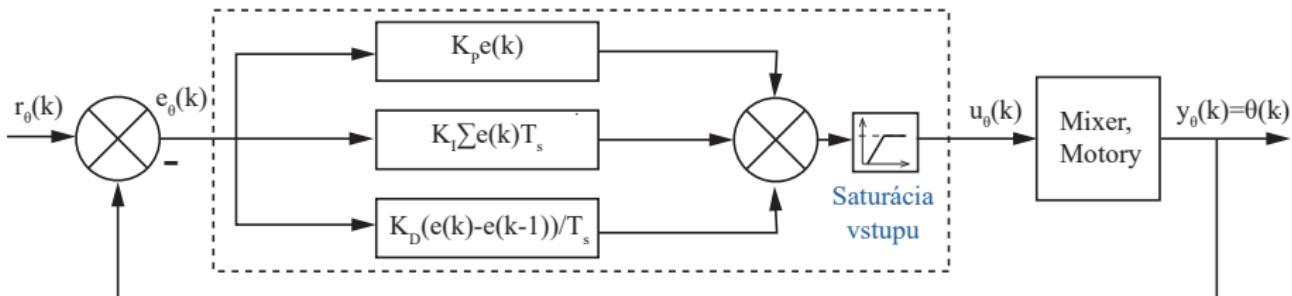
- Mapa  $u_M(k) = Pu(k)$  kde  $P$  matica pridelenia riadenia (*angl.: control allocation matrix*). Nemusí obsahovať iba jednotky, môže byť aj neštvorcová ale našťastie transformácia je stále lineárna [Bresciani a kol. 2020].
- Môžeme priamo namapovať maticu efektivity akčných členov (*angl.: actuator effectiveness matrix*)  $B$ , kde  $P = B^{-1}$ . Ak počet akčných zásahov = počtu akčných členov, používame inverziu ináč napr. Moore-Penroseovu pseudoinverziu  $P = B^\dagger$  [Bresciani a kol. 2020].

- Akčné zásahy majú svoje ohraničenia ((angl.: constraints))
- Ako donútime ich dodržanie? Saturáciou (orezávaním) hodnôt, ktoré skutočne vypočíta PID.
- Saturácia vnáša nelinearitu, vplýva na výkon riadenia aj stabilitu.
- Aj iné veličiny môžeme saturovať, napr. žiadane hodnoty. Ako by sme ohraničili výstup?

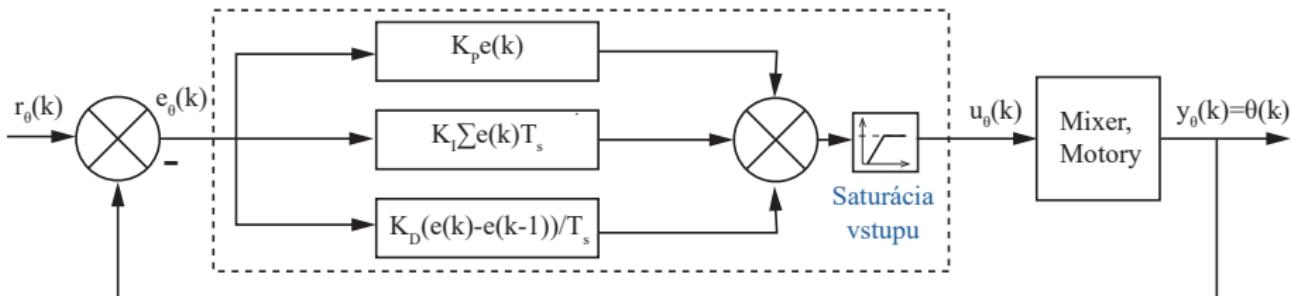
- Akčné zásahy majú svoje ohraničenia ((angl.: constraints))
- Ako donútime ich dodržanie? Saturáciou (orezávaním) hodnôt, ktoré skutočne vypočítava PID.
- Saturácia vnáša nelinearitu, vplýva na výkon riadenia aj stabilitu.
- Aj iné veličiny môžeme saturovať, napr. žiadane hodnoty. Ako by sme ohraničili výstup?



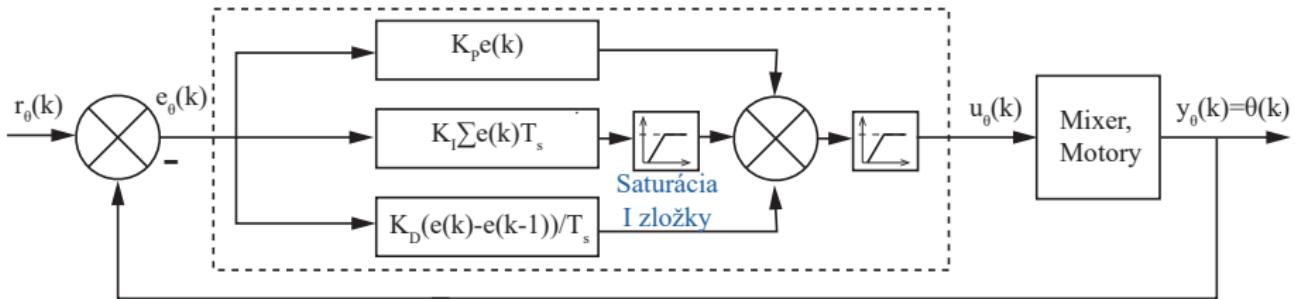
- Akčné zásahy majú svoje ohraničenia ((angl.: constraints))
- Ako donútime ich dodržanie? Saturáciou (orezávaním) hodnôt, ktoré skutočne vypočítava PID.
- Saturácia vnáša nelinearitu, vplýva na výkon riadenia aj stabilitu.
- Aj iné veličiny môžeme saturovať, napr. žiadane hodnoty. Ako by sme ohraničili výstup?



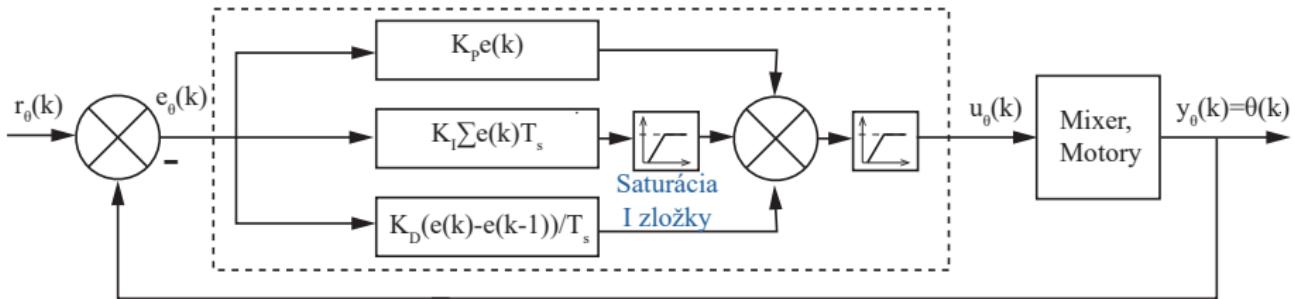
- Akčné zásahy majú svoje ohraničenia ((angl.: constraints))
- Ako donútime ich dodržanie? Saturáciou (orezávaním) hodnôt, ktoré skutočne vypočítava PID.
- Saturácia vnáša nelinearitu, vplýva na výkon riadenia aj stabilitu.
- Aj iné veličiny môžeme saturovať, napr. žiadane hodnoty. Ako by sme ohraničili výstup?



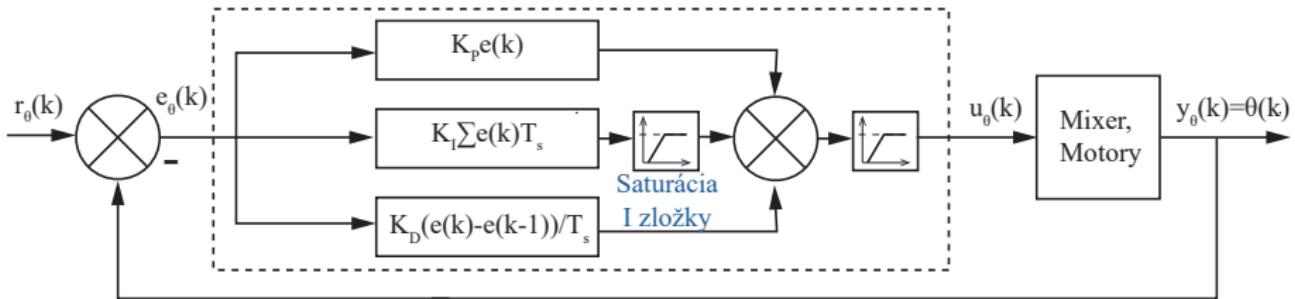
- Integračná zložka ráta odchýlku v minulosti, preto ak akčné členy sú už na hraniciach možností - začína sa nahromadovať (*angl.*: windup).
- Akonáhle sa vrátia akčné zásahy pod ohraničenia, nahromadená I zložka stále bude tlačiť systém na hranice možností, musí sa to chvíľu "uvolňovať" (*angl.*: unwind) a tým pádom prestrelíme (*angl.*: overshoot) žiadané hodnoty
- To je saturácia integračnej zložky (*angl.*: integral windup).
- Môžeme používať rôzne triky, napr. ohraničiť veľkosť integračnej zložky, resp. vypnúť zložku pri určitých podmienkach.



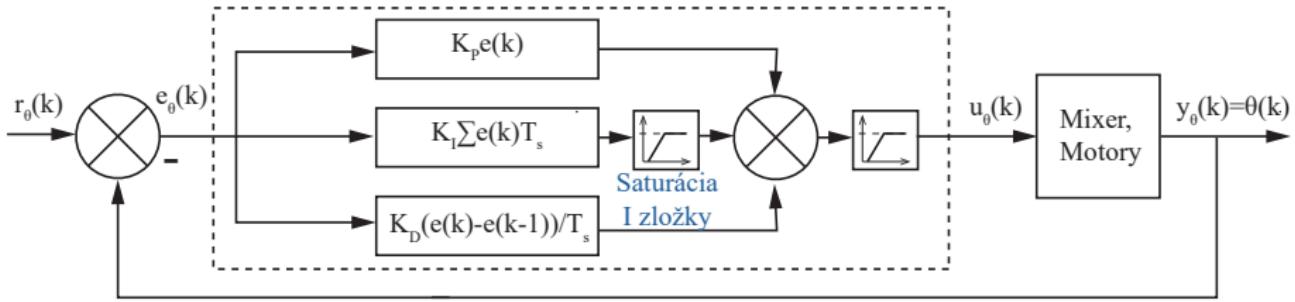
- Integračná zložka ráta odchýlku v minulosti, preto ak akčné členy sú už na hraniciach možností - začína sa nahromadať (*angl.*: windup).
- Akonáhle sa vrátia akčné zásahy pod ohraničenia, nahromadená I zložka stále bude tlačiť systém na hranice možností, musí sa to chvíľu "uvolňovať" (*angl.*: unwind) a tým pádom prestrelíme (*angl.*: overshoot) žiadané hodnoty
- To je saturácia integračnej zložky (*angl.*: integral windup).
- Môžeme používať rôzne triky, napr. ohraničiť veľkosť integračnej zložky, resp. vypnúť zložku pri určitých podmienkach.



- Integračná zložka ráta odchýlku v minulosti, preto ak akčné členy sú už na hraniciach možností - začína sa nahromadať (*angl.*: windup).
- Akonáhle sa vrátia akčné zásahy pod ohraničenia, nahromadená I zložka stále bude tlačiť systém na hranice možností, musí sa to chvíľu "uvolňovať" (*angl.*: unwind) a tým pádom prestrelíme (*angl.*: overshoot) žiadané hodnoty
- To je saturácia integračnej zložky (*angl.*: integral windup).
- Môžeme používať rôzne triky, napr. ohraničiť veľkosť integračnej zložky, resp. vypnúť zložku pri určitých podmienkach.

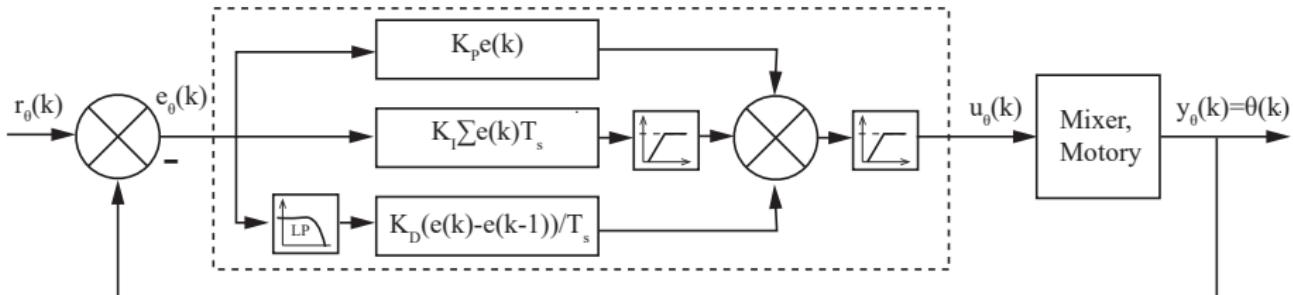


- Integračná zložka ráta odchýlku v minulosti, preto ak akčné členy sú už na hraniciach možností - začína sa nahromadať (*angl.*: windup).
- Akonáhle sa vrátia akčné zásahy pod ohraničenia, nahromadená I zložka stále bude tlačiť systém na hranice možností, musí sa to chvíľu "uvolňovať" (*angl.*: unwind) a tým pádom prestrelíme (*angl.*: overshoot) žiadané hodnoty
- To je saturácia integračnej zložky (*angl.*: integral windup).
- Môžeme používať rôzne triky, napr. ohraničiť veľkosť integračnej zložky, resp. vypnúť zložku pri určitých podmienkach.

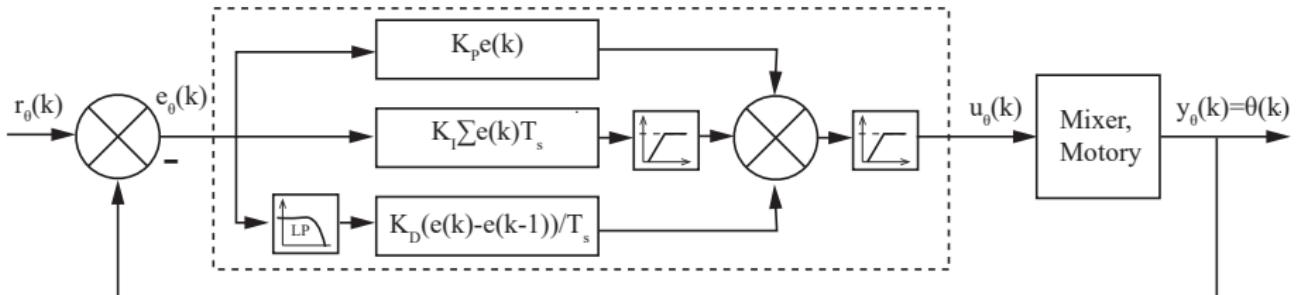


- Integračná zložka ráta odchýlku v minulosti, preto ak akčné členy sú už na hraniciach možností - začína sa nahromadovať (*angl.*: windup).
- Akonáhle sa vrátia akčné zásahy pod ohraničenia, nahromadená I zložka stále bude tlačiť systém na hranice možností, musí sa to chvíľu "uvolňovať" (*angl.*: unwind) a tým pádom prestrelíme (*angl.*: overshoot) žiadane hodnoty
- To je saturácia integračnej zložky (*angl.*: integral windup).
- Môžeme používať rôzne triky, napr. ohraničiť veľkosť integračnej zložky, resp. vypnúť zložku pri určitých podmienkach.
- ArduPilot - Ak akčný člen je akurát saturovaný, podrž hodnotu I zložky. Nižšie to môže ísť, vyššie nie [\[Hall 2018\]](#).
- ArduPilot - Keďže máme kaskádnu konfiguráciu PID slučiek, saturačný znak postupuje cez hierarchiu nižšie a nižšie aby zastavil nahromadenie I zložky [\[Hall 2018\]](#).

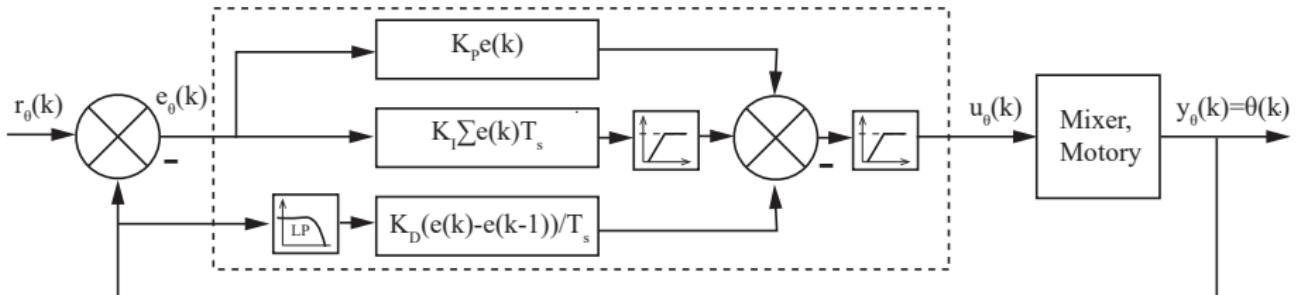
- Šum zo snímačov môže propagovať cez výpočet odchýlky riadenia do derivačnej zložky Vysokofrekvenčné zložky potom navýšia D zložku (čo je derivácia impulzu?)
- Riešenie: Odchýlku riadenia pustíme cez dolnopriepustný (angl.: low-pass) filter (LPF) — Aj ArduCopter (20 Hz LPF) aj PX4 Autopilot používa [Hall 2018; PX4 Autopilot 2021a]
- Náhle zmeny spôsobia “kopnutie” riadenia (angl.: derivative kick). (čo je derivácia impulzu?)
- Môžeme celkovo obísť zmenu žiadanej hodnoty a tým odchýlky  $e_\Theta(k)$  tak, že derivujeme výstup  $y_\Theta(k)^1$



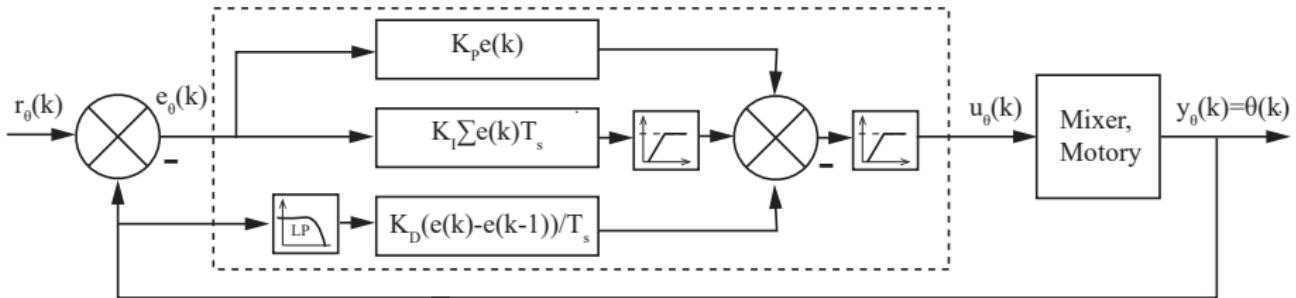
- Šum zo snímačov môže propagovať cez výpočet odchýlky riadenia do derivačnej zložky Vysokofrekvenčné zložky potom navýšia D zložku (čo je derivácia impulzu?)
- Riešenie: Odchýlku riadenia pustíme cez dolnopriepustný (angl.: low-pass) filter (LPF) — Aj ArduCopter (20 Hz LPF) aj PX4 Autopilot používa [Hall 2018; PX4 Autopilot 2021a]
- Náhle zmeny spôsobia “kopnutie” riadenia (angl.: derivative kick). (čo je derivácia impulzu?)
- Môžeme celkovo obísť zmenu žiadanej hodnoty a tým odchýlky  $e_\theta(k)$  tak, že derivujeme výstup  $y_\theta(k)^1$

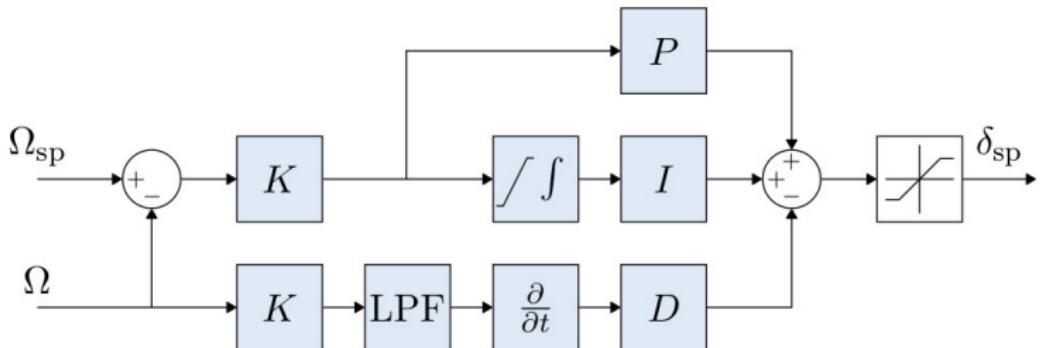


- Šum zo snímačov môže propagovať cez výpočet odchýlky riadenia do derivačnej zložky Vysokofrekvenčné zložky potom navýšia D zložku (čo je derivácia impulzu?)
- Riešenie: Odchýlku riadenia pustíme cez dolnopriepustný (*angl.*: low-pass) filter (LPF) — Aj ArduCopter (20 Hz LPF) aj PX4 Autopilot používa [Hall 2018; PX4 Autopilot 2021a]
- Náhle zmeny spôsobia “kopnutie” riadenia (*angl.*: derivative kick). (čo je derivácia impulzu?)
- Môžeme celkovo obísť zmenu žiadanej hodnoty a tým odchýlky  $e_\Theta(k)$  tak, že derivujeme výstup  $y_\Theta(k)^1$

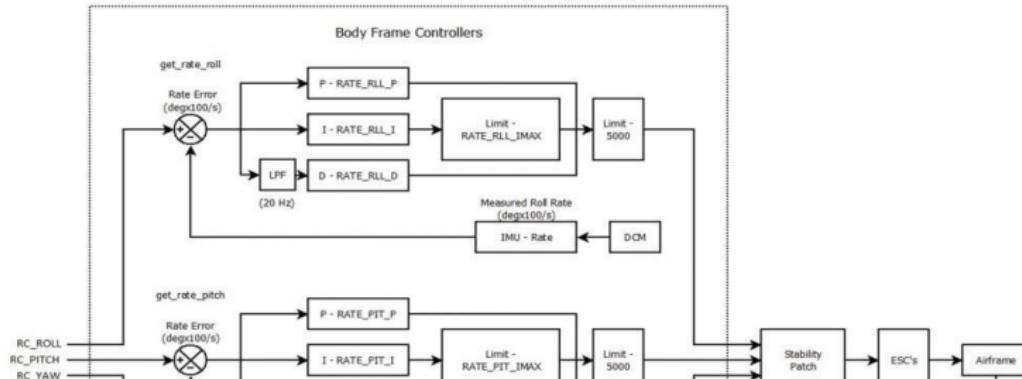


- Šum zo snímačov môže propagovať cez výpočet odchýlky riadenia do derivačnej zložky Vysokofrekvenčné zložky potom navýšia D zložku (čo je derivácia impulzu?)
- Riešenie: Odchýlku riadenia pustíme cez dolnopriepustný (*angl.*: low-pass) filter (LPF) — Aj ArduCopter (20 Hz LPF) aj PX4 Autopilot používa [Hall 2018; PX4 Autopilot 2021a]
- Náhle zmeny spôsobia “kopnutie” riadenia (*angl.*: derivative kick). (čo je derivácia impulzu?)
- Môžeme celkovo obísť zmenu žiadanej hodnoty a tým odchýlky  $e_\theta(k)$  tak, že derivujeme výstup  $y_\theta(k)$ <sup>1</sup>





ArduCopter V2.9 STABILIZE Roll, Pitch & Yaw PID's  
100 Hz Update Rate



- Napr. v Mission Planner nájdeme všetky Eulerove uhly + výšku, polohu, rýchlosť



- Heuristika<sup>2</sup> vs. pseudo-heuristický prístup: Ziegler-Nicholsova (ZN) metóda: [Ziegler a kol. 1942]
- Nastavme  $K_I$ ,  $K_D$  zložky nulové a zvyšujme  $K_P$ , kým nedostaneme na hranu stability s konzistentnými osciláciami. Tam sme našli  $K_{P\max}$  a períoda oscilácií bude  $T_{\max}$ .
- Opakujeme pre každú slučku, poprípade pre každú os (dá sa aj priviazať dron)
- Na základe týchto hodnôt môžeme z tabuľky na základe pseudoheuristiky vybrať ladenie.

---

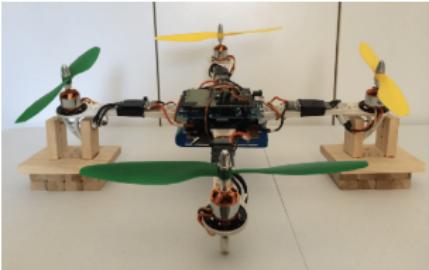
<sup>2</sup>fuck around, find out

- Heuristika<sup>2</sup> vs. pseudo-heuristický prístup: Ziegler-Nicholsova (ZN) metóda: [Ziegler a kol. 1942]
- Nastavme  $K_I$ ,  $K_D$  zložky nulové a zvyšujme  $K_P$ , kým nedostaneme na hranu stability s konzistentnými osciláciami. Tam sme našli  $K_{P\max}$  a períoda oscilácií bude  $T_{\max}$ .
- Opakujeme pre každú slučku, poprípade pre každú os (dá sa aj priviazať dron)
- Na základe týchto hodnôt môžeme z tabuľky na základe pseudoheuristiky vybrať ladenie.

---

<sup>2</sup>fuck around, find out

- Heuristika<sup>2</sup> vs. pseudo-heuristický prístup: Ziegler-Nicholsova (ZN) metóda: [Ziegler a kol. 1942]
- Nastavme  $K_I$ ,  $K_D$  zložky nulové a zvyšujme  $K_P$ , kým nedostaneme na hranu stability s konzistentnými osciláciami. Tam sme našli  $K_{P\max}$  a períoda oscilácií bude  $T_{\max}$ .
- Opakujeme pre každú slučku, poprípade pre každú os (dá sa aj priviazať dron)
- Na základe týchto hodnôt môžeme z tabuľky na základe pseudoheuristiky vybrať ladenie.



<sup>2</sup>fuck around, find out

- Heuristika<sup>2</sup> vs. pseudo-heuristický prístup: Ziegler-Nicholsova (ZN) metóda: [Ziegler a kol. 1942]
- Nastavme  $K_I$ ,  $K_D$  zložky nulové a zvyšujme  $K_P$ , kým nedostaneme na hranu stability s konzistentnými osciláciami. Tam sme našli  $K_{P\max}$  a períoda oscilácií bude  $T_{\max}$ .
- Opakujeme pre každú slučku, poprípade pre každú os (dá sa aj priviazať dron)
- Na základe týchto hodnôt môžeme z tabuľky na základe pseudoheuristiky vybrať ladenie.

	$K_P$	$T_I$	$T_D$	$K_I$	$K_D$
P	$0.5K_{P\max}$	-	-	-	-
PI	$0.45K_{P\max}$	$0.80T_{\max}$	-	$0.54K_{P\max}/T_{\max}$	
PD	$0.8K_{P\max}$	-	$0.125T_{\max}$	-	$0.10K_{P\max}$
PID	$0.6K_{P\max}$	$0.5T_{\max}$	$0.125T_{\max}$	$1.2K_{P\max}/T_{\max}$	$0.075K_{P\max}$

<sup>2</sup>fuck around, find out

- **Výhody:** Jednoduché
- **Nevýhody:** tlačí to quad na hranu stability, určite nebude “optimálne” pre všetky prípady a od 1942 máme lepšie metódy. Nastavené na maximálne odstránenie porúch, náchylné na agresívne riadenie s prestrelom Bequette 2010
- Môžeme podobné experimentálne dáta získať aj zo skokovej odozvy
- ArduPilot (ArduCopter) — “AutoTuning” Napodobňuje túto heuristiku (ale nie je to ZN) [Tridgell 2021]
  - ▶ Zvyšuje P parameter pomaly
  - ▶ Akonáhle deteguje osciláciu, zníži P parameter
  - ▶ Opakuje pre iné zložky/slučky
- PX4 dlho nemal vstavanú “autotune” funkciu [Bolton a kol. 2018] , momentálne využíva na odhad SISO linearizovaný model 2. rádu (bez prepojenia medzi osami) [PX4 Autopilot 2021b]

- **Výhody:** Jednoduché
- **Nevýhody:** tlačí to quad na hranu stability, určite nebude “optimálne” pre všetky prípady a od 1942 máme lepšie metódy. Nastavené na maximálne odstránenie porúch, náchylné na agresívne riadenie s prestrelom [Bequette 2010](#)
- Môžeme podobné experimentálne dáta získať aj zo skokovej odozvy
- ArduPilot (ArduCopter) — “AutoTuning” Napodobňuje túto heuristiku (ale nie je to ZN) [\[Tridgell 2021\]](#)
  - ▶ Zvyšuje P parameter pomaly
  - ▶ Akonáhle deteguje osciláciu, zníži P parameter
  - ▶ Opakuje pre iné zložky/slučky
- PX4 dlho nemal vstavanú “autotune” funkciu [\[Bolton a kol. 2018\]](#), momentálne využíva na odhad SISO linearizovaný model 2. rádu (bez prepojenia medzi osami) [\[PX4 Autopilot 2021b\]](#)

- **Výhody:** Jednoduché
- **Nevýhody:** tlačí to quad na hranu stability, určite nebude “optimálne” pre všetky prípady a od 1942 máme lepšie metódy. Nastavené na maximálne odstránenie porúch, náchylné na agresívne riadenie s prestrelom [Bequette 2010](#)
- Môžeme podobné experimentálne dáta získať aj zo skokovej odozvy
- ArduPilot (ArduCopter) — “AutoTuning” Napodobňuje túto heuristiku (ale nie je to ZN) [Tridgell 2021]
  - ▶ Zvyšuje P parameter pomaly
  - ▶ Akonáhle deteguje osciláciu, zníži P parameter
  - ▶ Opakuje pre iné zložky/slučky
- PX4 dlho nemal vstavanú “autotune” funkciu [Bolton a kol. 2018] , momentálne využíva na odhad SISO linearizovaný model 2. rádu (bez prepojenia medzi osami) [PX4 Autopilot 2021b]

- **Výhody:** Jednoduché
- **Nevýhody:** tlačí to quad na hranu stability, určite nebude “optimálne” pre všetky prípady a od 1942 máme lepšie metódy. Nastavené na maximálne odstránenie porúch, náchylné na agresívne riadenie s prestrelom [Bequette 2010](#)
- Môžeme podobné experimentálne dáta získať aj zo skokovej odozvy
- ArduPilot (ArduCopter) — “AutoTuning” Napodobňuje túto heuristiku (ale nie je to ZN) [\[Tridgell 2021\]](#)
  - ▶ Zvyšuje P parameter pomaly
  - ▶ Akonáhle deteguje osciláciu, zníži P parameter
  - ▶ Opakuje pre iné zložky/slučky
- PX4 dlho nemal vstavanú “autotune” funkciu [\[Bolton a kol. 2018\]](#), momentálne využíva na odhad SISO linearizovaný model 2. rádu (bez prepojenia medzi osami) [\[PX4 Autopilot 2021b\]](#)

- **Výhody:** Jednoduché
- **Nevýhody:** tlačí to quad na hranu stability, určite nebude “optimálne” pre všetky prípady a od 1942 máme lepšie metódy. Nastavené na maximálne odstránenie porúch, náchylné na agresívne riadenie s prestrelom [Bequette 2010](#)
- Môžeme podobné experimentálne dáta získať aj zo skokovej odozvy
- ArduPilot (ArduCopter) — “AutoTuning” Napodobňuje túto heuristiku (ale nie je to ZN) [\[Tridgell 2021\]](#)
  - ▶ Zvyšuje P parameter pomaly
  - ▶ Akonáhle deteguje osciláciu, zníži P parameter
  - ▶ Opakuje pre iné zložky/slučky
- PX4 dlho nemal vstavanú “autotune” funkciu [\[Bolton a kol. 2018\]](#), momentálne využíva na odhad SISO linearizovaný model 2. rádu (bez prepojenia medzi osami) [\[PX4 Autopilot 2021b\]](#)

- Cohen Coon je založené na ZN, pridá aj oneskorenie sústavy [Joseph a kol. 2018] (nie je typicky problém pri quadoch)
- Chien, Hrones a Reswickova (CHR) metóda, taktiež založené na ZN. Vyladené na maximalizáciu rýchlosťi odozvy bez prestrelu (alebo 20%) [Hambali a kol. 2014] .

- Cohen Coon je založené na ZN, pridá aj oneskorenie sústavy [Joseph a kol. 2018] (nie je typicky problém pri quadoch)
- Chien, Hrones a Reswickova (CHR) metóda, taktiež založené na ZN. Vyladené na maximalizáciu rýchlosťi odozvy bez prestrelu (alebo 20%) [Hambali a kol. 2014] .

Notes: [Hall 2018](#)

User (ROS) → Shaping → PID → Actuators Yaw je prioritizovanych nad Pitch Roll, lebo to drzi dron v lufte [\[Erasmus 2020\]](#) 50 Hz -> 400 Hz Min 24 tazsie uchopytelne koncepty pre prezentaciu, dava menej konkretnosti Velocity prioritizuje vertikalnu rychlosť [\[Erasmus 2020\]](#)

Ďakujem za Vašu pozornosť.

- [1] B. W. Bequette. *Process Control: Modeling, Design, and Simulation*. Prentice Hall PTR, 2010.
- [2] Barry Bolton a PX4 Dev Team. *Feature Request: AutoTune*. Online. PX4 GitHub [cited 3.12.2021]; Available from <https://github.com/PX4/PX4-Autopilot/issues/8472>. 2018.
- [3] Mathieu Bresciani a Matthias Grob. *Overview of multicopter control from sensors to motors*. Online. PX4 Developer Summit Virtual 2020. [cited 1.12.2021]; Available from [https://www.youtube.com/watch?v=orvng\\_11ngQ](https://www.youtube.com/watch?v=orvng_11ngQ). 2020.
- [4] Brian Douglas. *Drone Simulation and Control*. Online. From the series Matlab Tech Talks [cited 3.12.2021]; Available from <https://www.mathworks.com/videos/series/drone-simulation-and-control.html>. 2018.

- [5] Anton Erasmus. *An In-depth Look at the Multicopter Control System Architecture*. Online. PX4 Developer Summit Virtual 2020. [cited 1.12.2021]; Available from <https://www.youtube.com/watch?v=nEo4WG14Lgc>. 2020.
- [6] Leonard Hall. *Practical PID implementation and the new Position Controller*. [online]. ArduPilot UnConference 2018, uploaded Feb 22, 2018 [cited 24.11.2021]; Available from <https://www.youtube.com/watch?v=-PC69jcMizA>. 2018.
- [7] Najidah Hambali a kol. “Process controllability for flow control system using Ziegler-Nichols (ZN), Cohen-Coon (CC) and Chien-Hrones-Reswick (CHR) tuning methods”. In: *2014 IEEE International Conference on Smart Instrumentation, Measurement and Applications (ICSIMA)*. 2014, s. 1–6.
- [8] Elisha Joseph a O OlaiyaO. “Cohen-Coon PID Tuning Method: A Better Option to Ziegler Nichols-Pid Tuning Method”. In: *Computer Engineering and Intelligent Systems* 9 (2018), s. 33–37.

- [9] **PX4 Autopilot.** *Controller Diagrams — Multicopter Control Architecture*. Online. [cited 29.11.2021]; Available from [https://docs.px4.io/master/en/flight\\_stack/controller\\_diagrams.html](https://docs.px4.io/master/en/flight_stack/controller_diagrams.html). 2021.
- [10] **PX4 Autopilot.** *Multicopter PID Tuning Guide (Advanced/Detailed)*. Online. [cited 30.11.2021]; Available from <https://docs.px4.io/master/en/config/autotune.html>. 2021.
- [11] **Andrew Tridgell.** *ArduPilot Log Analysis Seminar*. Online. [cited 10.11.2021]; Available from [https://www.youtube.com/watch?v=WcfLTW\\_qZ08&list=PLC8WVaJJhN4ya\\_HDxh6qGBT6VbjXR6L5p](https://www.youtube.com/watch?v=WcfLTW_qZ08&list=PLC8WVaJJhN4ya_HDxh6qGBT6VbjXR6L5p). 2021.
- [12] **J. G. Ziegler a N. B. Nichols.** “Optimum settings for automatic controllers”. In: *Transactions of the ASME* 64.- (1942), s. 759–768.