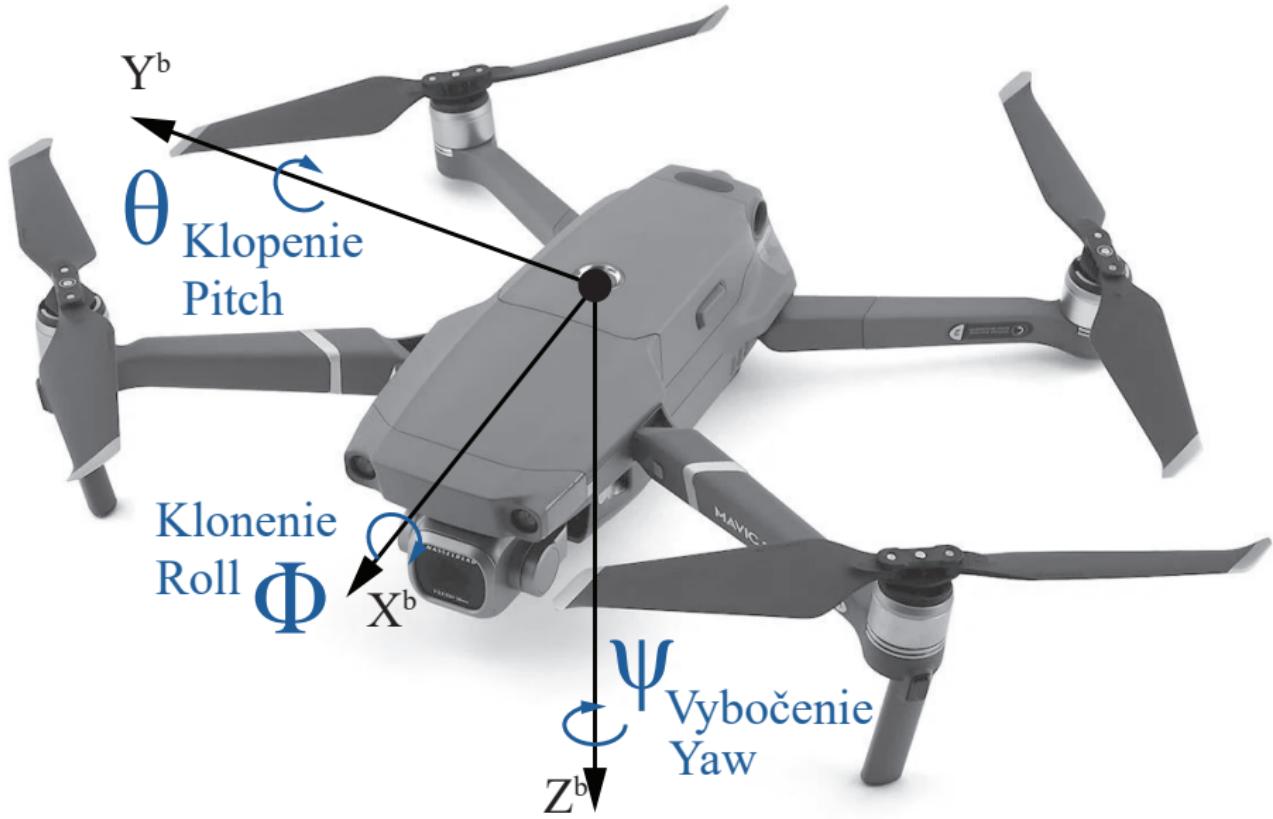


Working title drone control

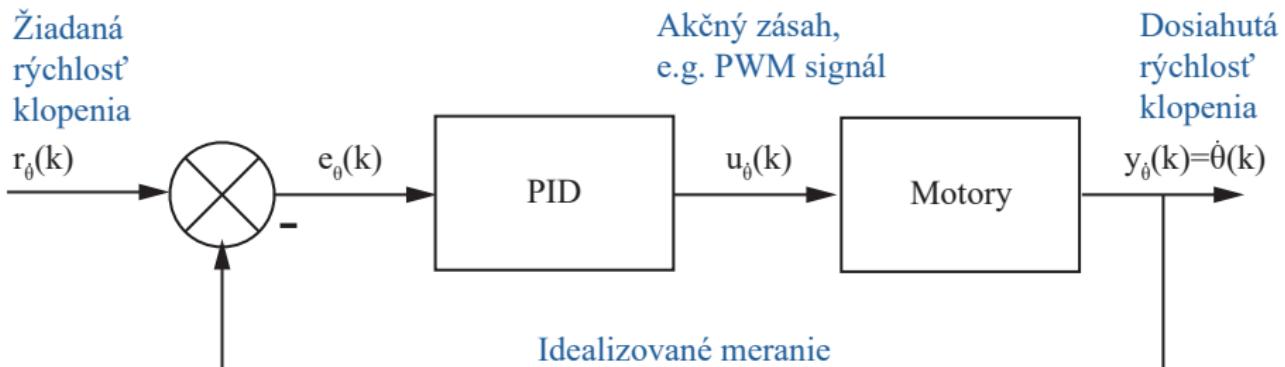
Riadenie dronov

prof. Ing. Gergely Takács, PhD.

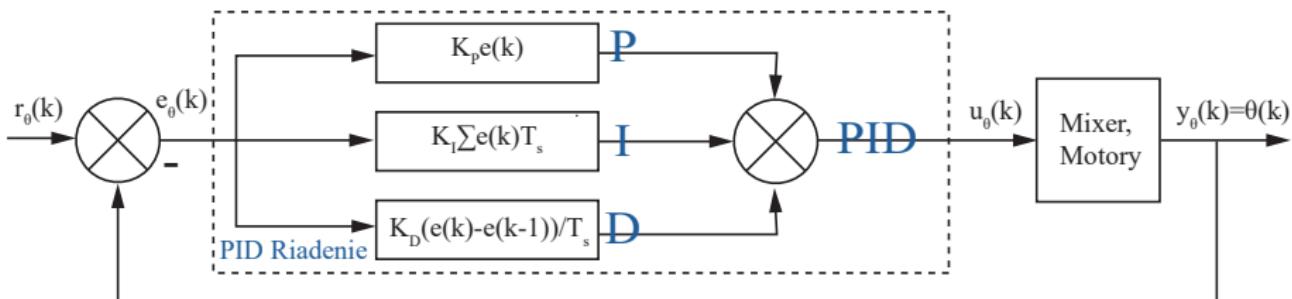




- Tvárme sa, že poznáme žiadanú uhlovú rýchlosť orientácie $r_{\dot{\Theta}}$, $r_{\dot{\phi}}$ a $r_{\dot{\psi}}$ a pre jednoduchosť sústredme len na rýchlosť zmeny klopenia ($\dot{\Theta}$).
- Sme na najnižšej úrovni riadenia — je to zároveň aj najdôležitejšia slučka [Hall 2018]

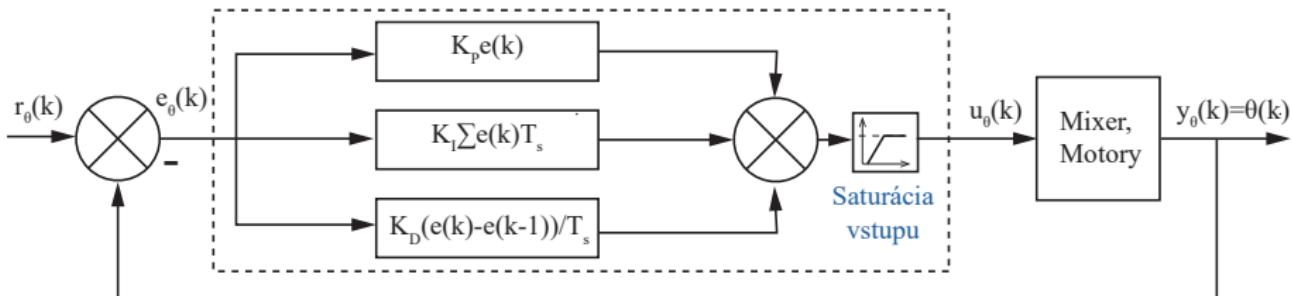


- Tvárame sa, že poznáme žiadanú uhlovú rýchlosť orientácie $r_{\dot{\Theta}}$, $r_{\dot{\phi}}$ a $r_{\dot{\psi}}$ a pre jednoduchosť sústredme len na rýchlosť zmeny klopenia ($\dot{\Theta}$).
- Sme na najnižšej úrovni riadenia — je to zároveň aj najdôležitejšia slučka [Hall 2018]

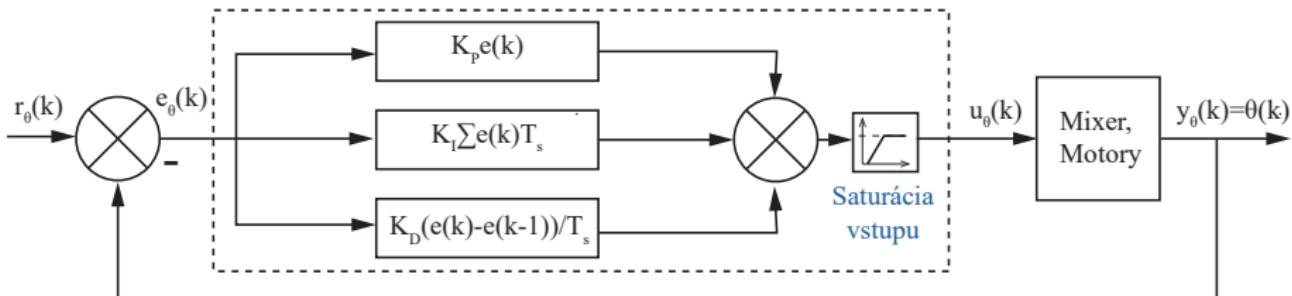


- Akčné zásahy majú svoje ohraničenia ((angl.: constraints))
- Ako donútime ich dodržanie? Saturáciou (orezávaním) hodnôt, ktoré skutočne vypočíta PID.
- Saturácia vnáša nelinearitu, vplýva na výkon riadenia aj stabilitu.
- Aj iné veličiny môžeme saturovať, napr. žiadane hodnoty. Ako by sme ohraničili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

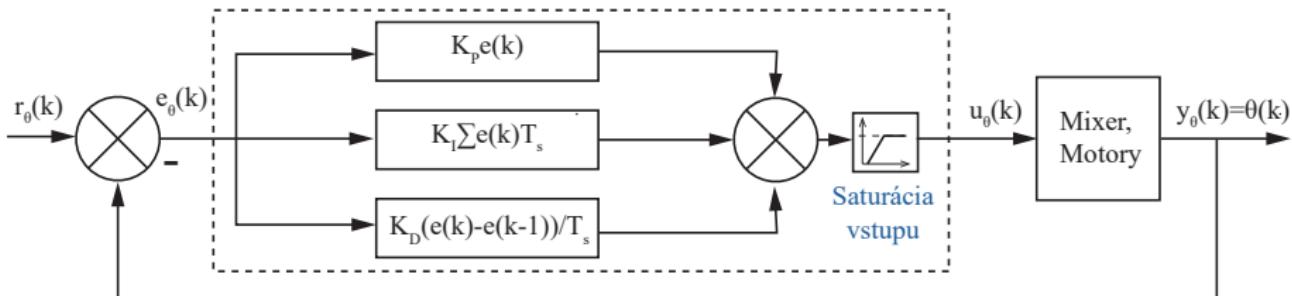
- Akčné zásahy majú svoje ohraničenia ((angl.: constraints))
- Ako donútime ich dodržanie? Saturáciou (orezávaním) hodnôt, ktoré skutočne vypočítava PID.
- Saturácia vnáša nelinearitu, vplýva na výkon riadenia aj stabilitu.
- Aj iné veličiny môžeme saturovať, napr. žiadane hodnoty. Ako by sme ohraničili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



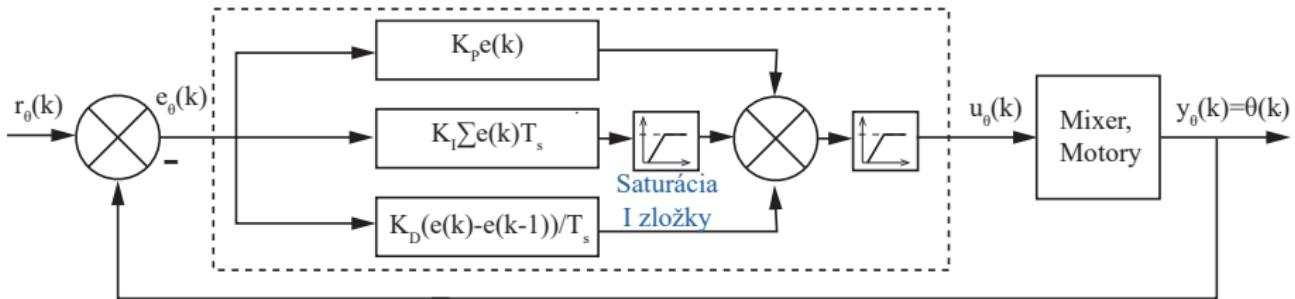
- Akčné zásahy majú svoje ohraničenia ((angl.: constraints))
- Ako donútime ich dodržanie? Saturáciou (orezávaním) hodnôt, ktoré skutočne vypočítava PID.
- Saturácia vnáša nelinearitu, vplýva na výkon riadenia aj stabilitu.
- Aj iné veličiny môžeme saturovať, napr. žiadane hodnoty. Ako by sme ohraničili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



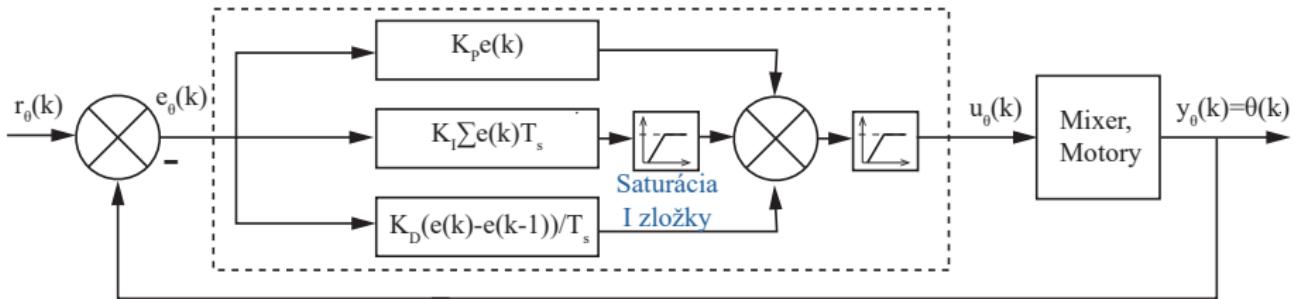
- Akčné zásahy majú svoje ohraničenia ((angl.: constraints))
- Ako donútime ich dodržanie? Saturáciou (orezávaním) hodnôt, ktoré skutočne vypočítava PID.
- Saturácia vnáša nelinearitu, vplýva na výkon riadenia aj stabilitu.
- Aj iné veličiny môžeme saturovať, napr. žiadane hodnoty. Ako by sme ohraničili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



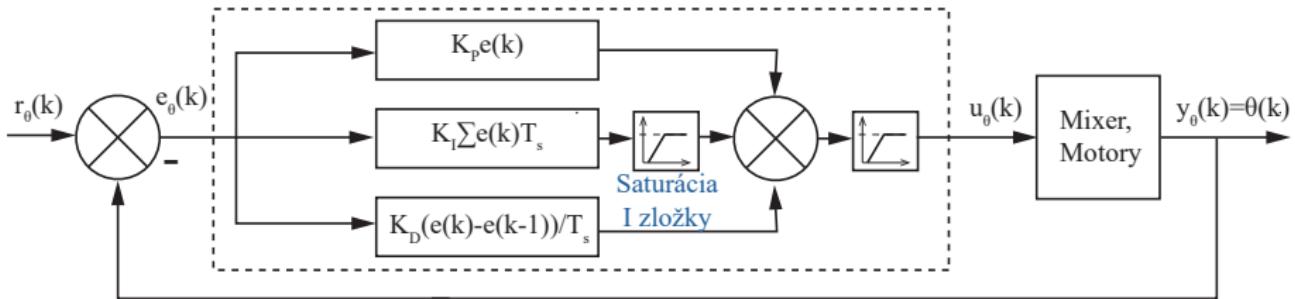
- Integračná zložka ráta odchýlku v minulosti, preto ak akčné členy sú už na hraniciach možností - začína sa nahromadovať (*angl.*: windup).
- Akonáhle sa vrátia akčné zásahy pod ohraničenia, nahromadená I zložka stále bude tlačiť systém na hranice možností, musí sa to chvíľu "uvolňovať" (*angl.*: unwind) 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.



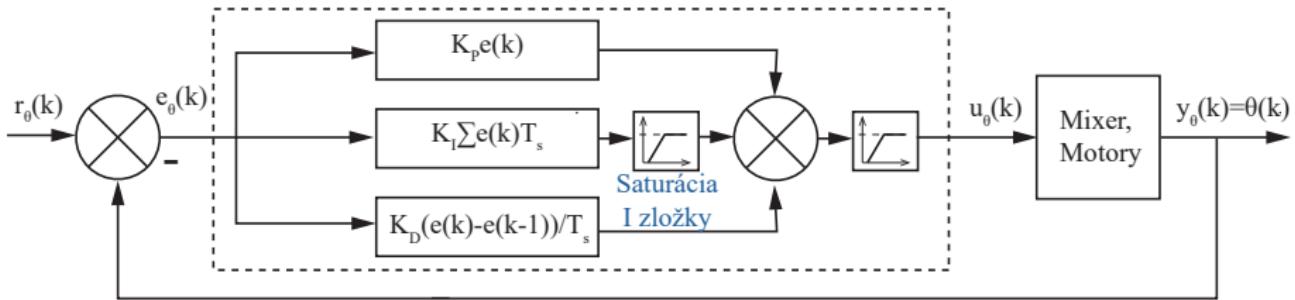
- Integračná zložka ráta odchýlku v minulosti, preto ak akčné členy sú už na hraniciach možností - začína sa nahromadať (*angl.*: windup).
- Akonáhle sa vrátia akčné zásahy pod ohraničenia, nahromadená I zložka stále bude tlačiť systém na hranice možností, musí sa to chvíľu "uvolňovať" (*angl.*: unwind) 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.



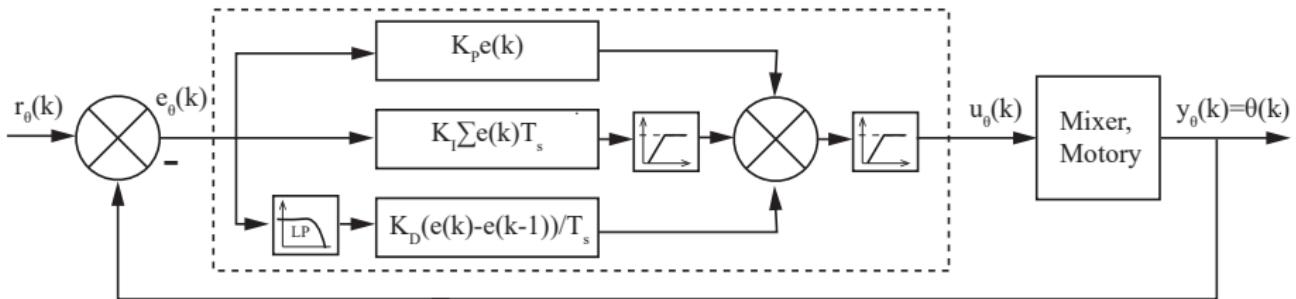
- Integračná zložka ráta odchýlku v minulosti, preto ak akčné členy sú už na hraniciach možností - začína sa nahromadať (*angl.*: windup).
- Akonáhle sa vrátia akčné zásahy pod ohraničenia, nahromadená I zložka stále bude tlačiť systém na hranice možností, musí sa to chvíľu "uvolňovať" (*angl.*: unwind) 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.



- ArduPilot - Ak akčný člen je akurát saturovaný, podrž hodnotu I zložky. Nižšie to môže ísť, vyššie nie [Hall 2018].
- ArduPilot - Ked'že máme kaskádnu konfiguráciu PID slučiek, saturačný znak postupuje cez hierarchiu nižšie a nižšie aby zastavil nahromadenie I zložky [Hall 2018].

- 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].

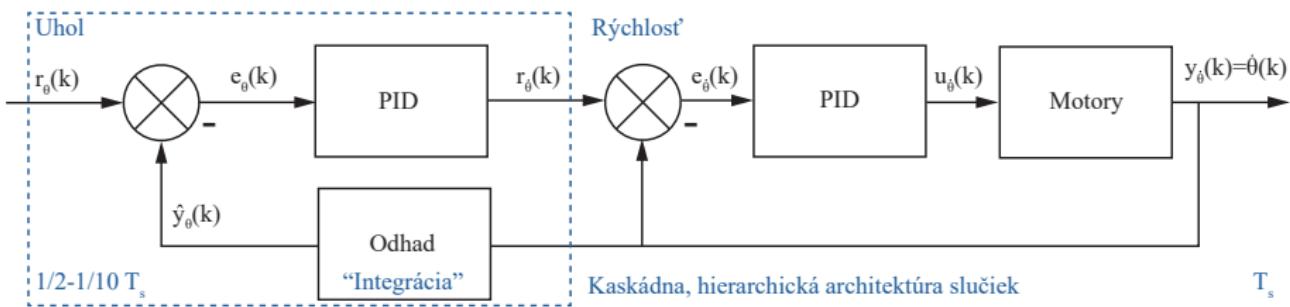
- aaaaaa



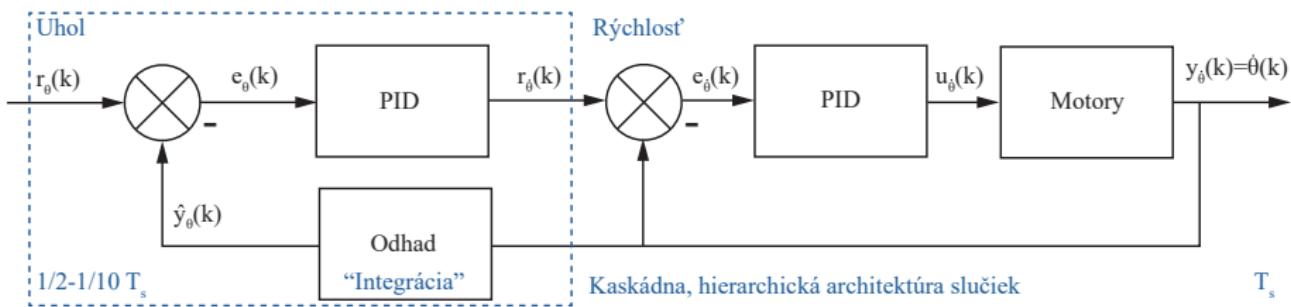
- Riadiť orientáciu len na základe zmeny rýchlosťi by bolo dosť neintuitívne, potrebujeme prepočítať $r_\Theta \rightarrow r_{\dot{\Theta}}$
- Tvárme sa, že poznáme žiadane orientácie, napr. klopenie r_Θ .
- Riadenie orientácie môže byť riešené ďalšou, nadradenou regulačnou slučkou - hovoríme o tzv. kaskádnom riadení (*angl.*: nested control).
- Nadradené slučky sú pomalšie [Hall 2018], vytvára to istý “filter”, t.j. nemôžeme rýchlejšie ovládať rýchlosť ako polohu (cca. o rád, min polovicu pomalšie)

- Riadiť orientáciu len na základe zmeny rýchlosťi by bolo dosť neintuitívne, potrebujeme prepočítať $r_\Theta \rightarrow r_{\dot{\Theta}}$
- Tvárme sa, že poznáme žiadane orientácie, napr. klopenie r_Θ .
- Riadenie orientácie môže byť riešené ďalšou, nadradenou regulačnou slučkou - hovoríme o tzv. kaskádnom riadení (*angl.*: nested control).
- Nadradené slučky sú pomalšie [Hall 2018], vytvára to istý “filter”, t.j. nemôžeme rýchlejšie ovládať rýchlosť ako polohu (cca. o rád, min polovicu pomalšie)

- Riadiť orientáciu len na základe zmeny rýchlosi by bolo dosť neintuitívne, potrebujeme prepočítať $r_\Theta \rightarrow r_{\dot{\Theta}}$
- Tvárme sa, že poznáme žiadane orientácie, napr. klopenie r_Θ .
- Riadenie orientácie môže byť riešené ďalšou, nadradenou regulačnou slučkou - hovoríme o tzv. kaskádnom riadení (*angl.*: nested control).
- Nadradené slučky sú pomalšie [Hall 2018], vytvára to istý "filter", t.j. nemôžeme rýchlejšie ovládať rýchlosť ako polohu (cca. o rád, min polovicu pomalšie)



- Riadiť orientáciu len na základe zmeny rýchlosi by bolo dosť neintuitívne, potrebujeme prepočítať $r_\Theta \rightarrow r_{\dot{\Theta}}$
- Tvárme sa, že poznáme žiadane orientácie, napr. klopenie r_Θ .
- Riadenie orientácie môže byť riešené ďalšou, nadradenou regulačnou slučkou - hovoríme o tzv. kaskádnom riadení (*angl.*: nested control).
- Nadradené slučky sú pomalšie [Hall 2018], vytvára to istý "filter", t.j. nemôžeme rýchlejšie ovládať rýchlosť ako polohu (cca. o rád, min polovicu pomalšie)

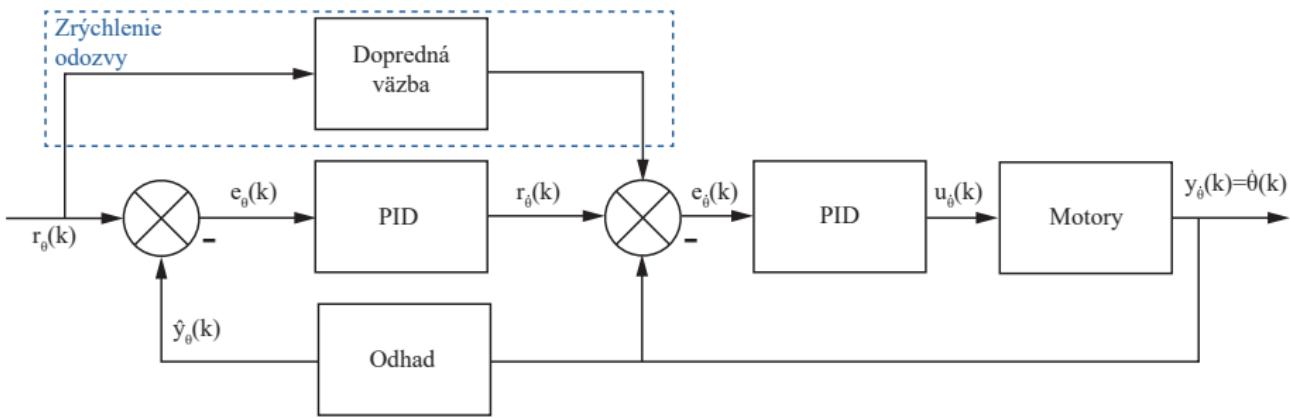


- Pri PID skôr PI¹, lebo
 - ▶ iba P neodstráni, trvalú odchýlku
 - ▶ D reaguje príliš agresívne na šum.
- Do slučky dopracujeme doprednú väzbu (*angl.*: feedforward) ktorá zrýchli odozvu regulácie
- “Whatever works” - netreba mystifikovať, je to len matematika²

¹ArduCopter a PX4 Autopilot používa P regulátor [PX4 Autopilot 2021; ArduPilot 2021]

²napr.

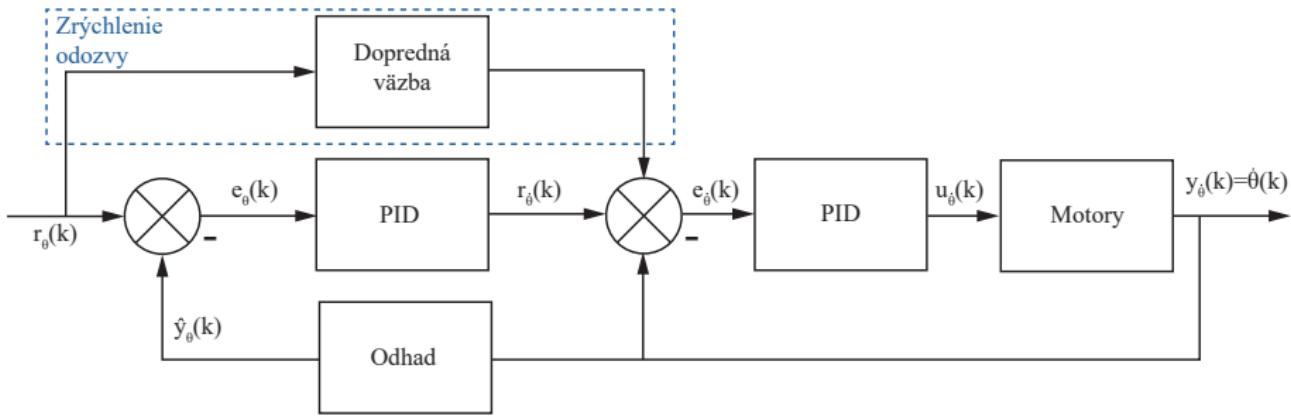
- Pri PID skôr PI¹, lebo
 - ▶ iba P neodstráni trvalú odchýlku
 - ▶ D reaguje príliš agresívne na šum.
- Do slučky dopracujeme doprednú väzbu (*angl.*: feedforward) ktorá zrýchli odozvu regulácie
- “Whatever works” - netreba mystifikovať, je to len matematika²



¹ArduCopter a PX4 Autopilot používa P regulátor [PX4 Autopilot 2021; ArduPilot 2021]

²napr.

- Pri PID skôr PI¹, lebo
 - ▶ iba P neodstráni trvalú odchýlku
 - ▶ D reaguje príliš agresívne na šum.
- Do slučky dopracujeme doprednú väzbu (*angl.*: feedforward) ktorá zrýchli odozvu regulácie
- “Whatever works” - netreba mystifikovať, je to len matematika²



¹ArduCopter a PX4 Autopilot používa P regulátor [PX4 Autopilot 2021; ArduPilot 2021]

²napr.

« « « < Updated upstream

- Integračná zložka ráta odchýlku v minulosti, preto ak akčné členy sú už na hraniciach možností - začína sa to zbytočne hromadiť (*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
- Tento nežiadaný efekt 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 to zbytočne hromadiť (*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
- Tento nežiadaný efekt je saturácia integračnej zložky (*angl.*: integral windup).
- Môžeme používať rôzne triky, napr. ohraničiť veľkosť integračnej zložky, resp. vypnúť zložku pri určitých podmienkach.

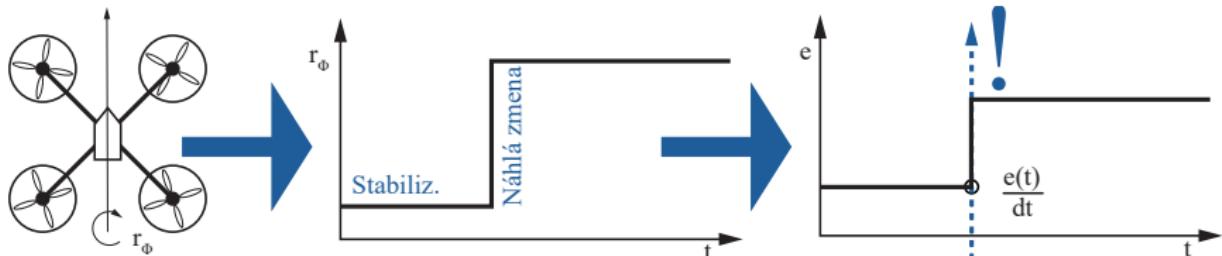
- Integračná zložka ráta odchýlku v minulosti, preto ak akčné členy sú už na hraniciach možností - začína sa to zbytočne hromadiť (*angl.*: windup).
- Akonáhle sa vrátia akčné zásahy pod ohraničenia, nahromadená I zložka stále bude tlačiť systém na hranice možností, musí sa to chvíľu "uvolňovať" (*angl.*: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

- Dron je stabilizovaný, pilot/ROS náhle chce $r_\phi = +30^\circ$ klonenie. Čo sa stane s riadením?
- Žiadana hodnota je skokový signál. Derivácia skoku je... D zložka a tým aj vstup do akčných členov vystrelí!
- Potrebujeme tvarovať vstupy do regulácie (*angl.: input shaping*), t.j. tvarovať žiadane hodnoty:
 - ▶ Pomalšie vzorkovanie na rýchlejšie (interpolácia) ³
 - ▶ Vyhladenie filtráciou, saturácie
 - ▶ Ak hovoríme o manuálnom pilotovaní, tvarovanie určuje aký "pocit" je riadiť stroj

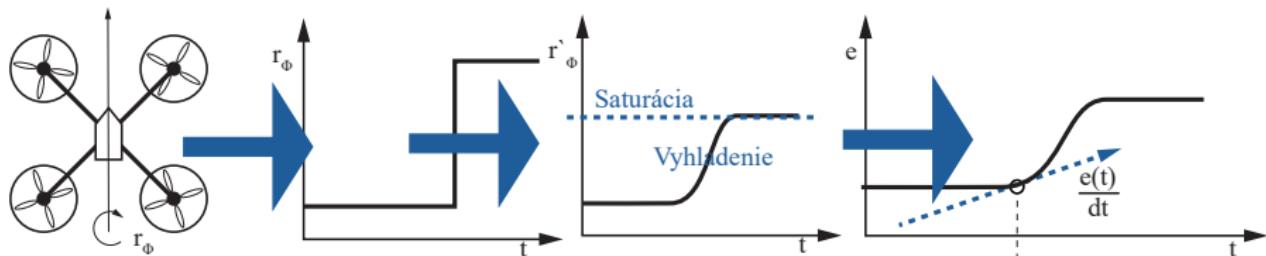
³ArduCopter 50 Hz → 400 Hz [Hall 2018]

- Dron je stabilizovaný, pilot/ROS náhle chce $r_\phi = +30^\circ$ klonenie. Čo sa stane s riadením?
- Žiadana hodnota je skokový signál. Derivácia skoku je... D zložka a tým aj vstup do akčných členov vystrelí!
- Potrebujeme tvarovať vstupy do regulácie (angl.: input shaping), t.j. tvarovať žiadane hodnoty:
 - ▶ Pomalšie vzorkovanie na rýchlejšie (interpolácia)³
 - ▶ Vyhladenie filtráciou, saturácie
 - ▶ Ak hovoríme o manuálnom pilotovaní, tvarovanie určuje aký "počit" je riadiť stroj



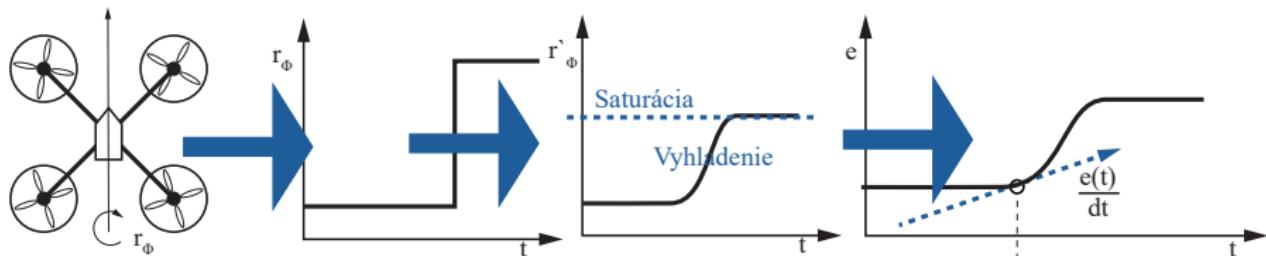
³ArduCopter 50 Hz → 400 Hz [Hall 2018]

- Dron je stabilizovaný, pilot/ROS náhle chce $r_\phi = +30^\circ$ klonenie. Čo sa stane s riadením?
- Žiadana hodnota je skokový signál. Derivácia skoku je... D zložka a tým aj vstup do akčných členov vystrelí!
- Potrebujeme tvarovať vstupy do regulácie (angl.: input shaping), t.j. tvarovať žiadane hodnoty:
 - ▶ Pomalšie vzorkovanie na rýchlejšie (interpolácia) ³
 - ▶ Vyhladenie filtráciou, saturácie
 - ▶ Ak hovoríme o manuálnom pilotovaní, tvarovanie určuje aký "pocit" je riadiť stroj



³ArduCopter 50 Hz → 400 Hz [Hall 2018]

- Dron je stabilizovaný, pilot/ROS náhle chce $r_\phi = +30^\circ$ klonenie. Čo sa stane s riadením?
- Žiadana hodnota je skokový signál. Derivácia skoku je... D zložka a tým aj vstup do akčných členov vystrelí!
- Potrebujeme tvarovať vstupy do regulácie (angl.: input shaping), t.j. tvarovať žiadane hodnoty:
 - ▶ Pomalšie vzorkovanie na rýchlejšie (interpolácia) ³
 - ▶ Vyhľadenie filtráciou, saturácie
 - ▶ Ak hovoríme o manuálnom pilotovaní, tvarovanie určuje aký "pocit" je riadiť stroj



³ArduCopter 50 Hz → 400 Hz [Hall 2018]

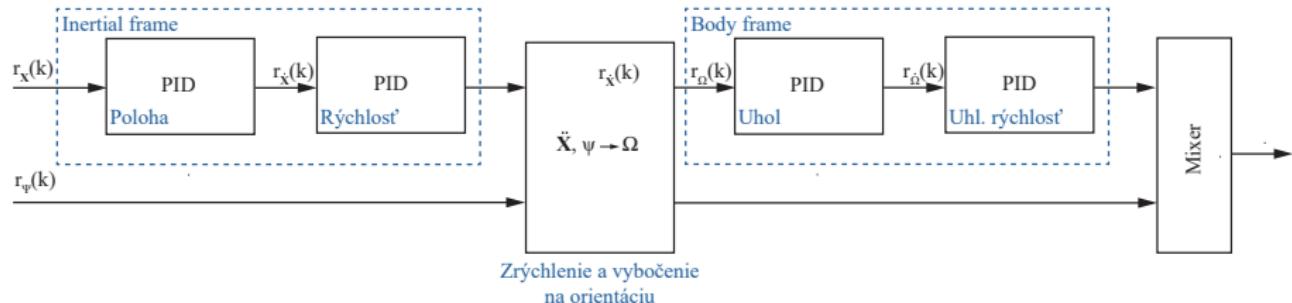
- Ako naštartujeme riadenie? Aká je odchýlka $e(0)$ pri štarte?
- Kritické pri zmene letových módov.
- Pre ArduCopter Hall 2018
 - ▶ Nadradené riadenie poloha vs. rýchlosť (P) — tak aby vstup bol konštantný
 - ▶ Rýchlosť (PID) — $e(0) = 0$, I zložka s konštantným vstupom, zrátať D pri nastavení žiadanej hodnoty

- Ako naštartujeme riadenie? Aká je odchýlka $e(0)$ pri štarte?
- Kritické pri zmene letových módov.
- Pre ArduCopter Hall 2018
 - ▶ Nadradené riadenie poloha vs. rýchlosť (P) — tak aby vstup bol konštantný
 - ▶ Rýchlosť (PID) — $e(0) = 0$, I zložka s konštantným vstupom, zrátať D pri nastavení žiadanej hodnoty

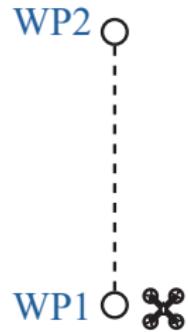
- Ako naštartujeme riadenie? Aká je odchýlka $e(0)$ pri štarte?
- Kritické pri zmene letových módov.
- Pre ArduCopter [Hall 2018](#)
 - ▶ Nadradené riadenie poloha vs. rýchlosť (P) — tak aby vstup bol konštantný
 - ▶ Rýchlosť (PID) — $e(0) = 0$, I zložka s konštantným vstupom, zrátať D pri nastavení žiadanej hodnoty

- Najnižšia úroveň je riadenie orientácie, tj. uhly $\Omega = \Phi, \Theta, \Sigma$ a uhlové rýchlosťi $\dot{\Omega}$ v "body frame"

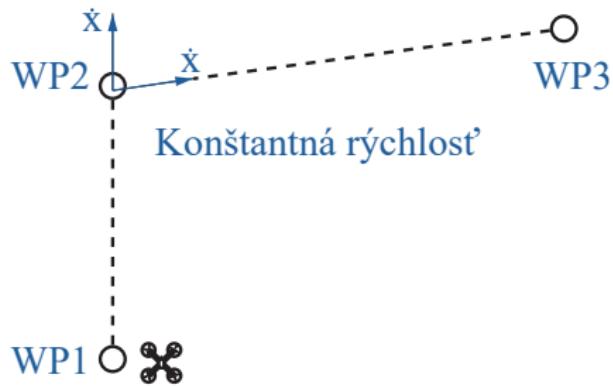
- Najnižšia úroveň je riadenie orientácie, tj. uhly $\Omega = \Phi, \Theta, \Sigma$ a uhlové rýchlosťi $\dot{\Omega}$ v "body frame"



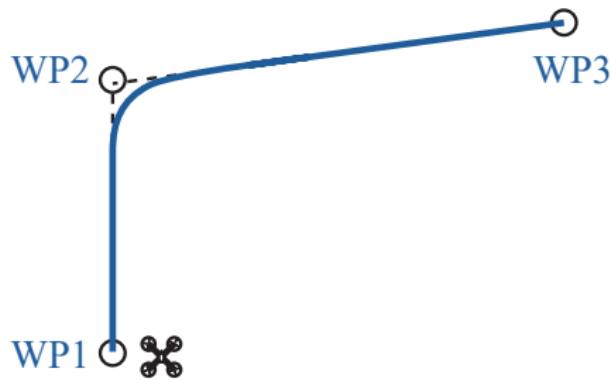
- Majme trasu WP1 do WP2, všetko je v poriadku.
- Pridajme WP3 a rozmýšľajme čo sa deje pri WP2. Je možné preletieť nad WP2?
- Bud' musíme úplne sa zastaviť alebo nemôžeme priamo preletieť — ináč by sme potrebovali nekonečne veľké zrýchlenia



- Majme trasu WP1 do WP2, všetko je v poriadku.
- Pridajme WP3 a rozmýšľajme čo sa deje pri WP2. Je možné preletieť nad WP2?
- Bud' musíme úplne sa zastaviť alebo nemôžeme priamo preletieť — ináč by sme potrebovali nekonečne veľké zrýchlenia



- Majme trasu WP1 do WP2, všetko je v poriadku.
- Pridajme WP3 a rozmýšľajme čo sa deje pri WP2. Je možné preletieť nad WP2?
- Bud' musíme úplne sa zastaviť alebo nemôžeme priamo preletieť — ináč by sme potrebovali nekonečne veľké zrýchlenia



- ArduPilot (ArduCopter) — Napodobňuje heuristiku [Tridgell 2021]
 - ▶ Zvyšuje D parameter pomaly
 - ▶ Akonáhle deteguje osciláciu, zníži D parameter
 - ▶ Opakuje pre iné zložky/slučky
- Intuitívne ale konzervatívne

Notes: Hall 2018

User (ROS) → Shaping → PID → Actuators

50 Hz -> 400 Hz Min 24 tazsie uchopytelne koncepty pre prezentaciu, dava menej konkretnosti

Ďakujem za Vašu pozornosť.

- [1] **ArduPilot.** *CCopter Attitude Control.* Online. [cited 29.11.2021]; Available from <https://ardupilot.org/dev/docs/apmcopter-programming-attitude-control-2.html>. 2021.
- [2] **Leonard Hall.** *Pratical 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.
- [3] **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. 2021.
- [4] **Andrew Tridgell.** *ArduPilot Log Analysis Seminar.* Online. [cited 10.11.2021]; Available from https://www.youtube.com/watch?v=WcfLTW_qZ08&list=PLC8WVaJJhN4ya_HDxh6qGBT6VbjXR6L5p. 2021.