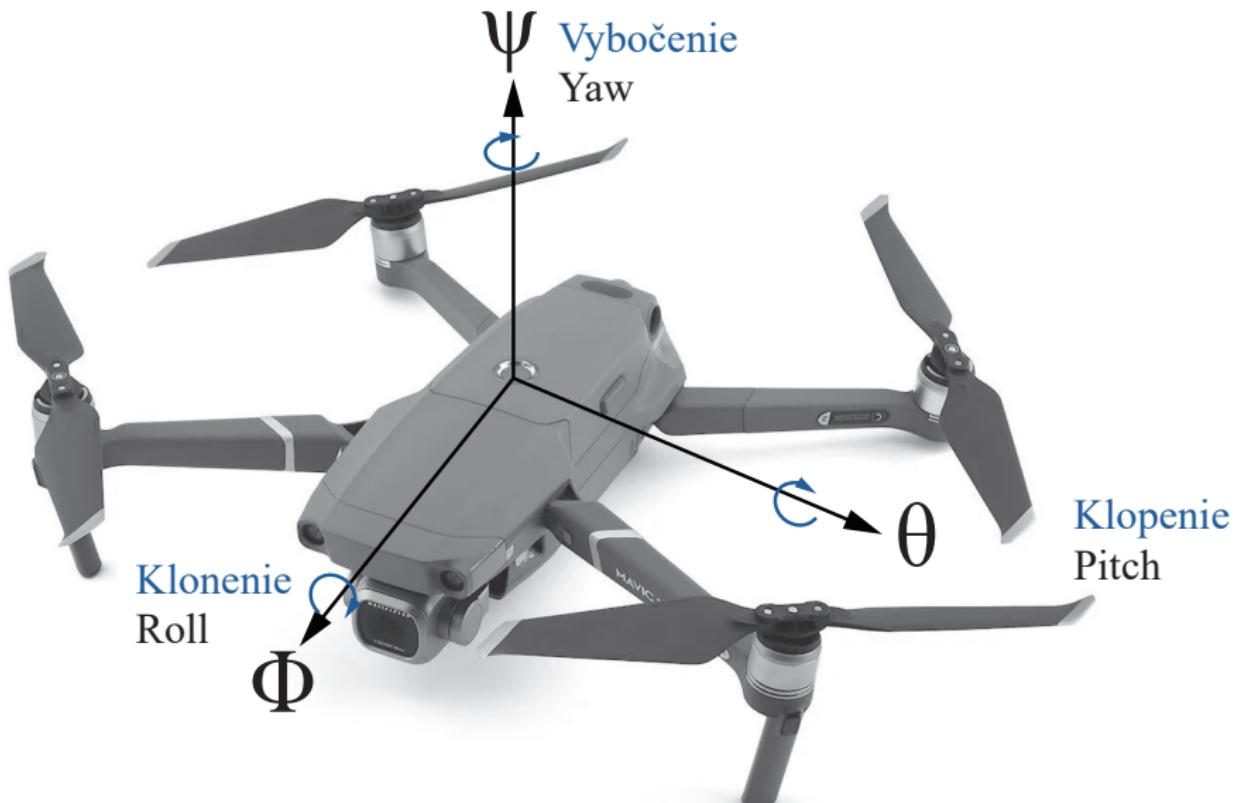


Working title drone control

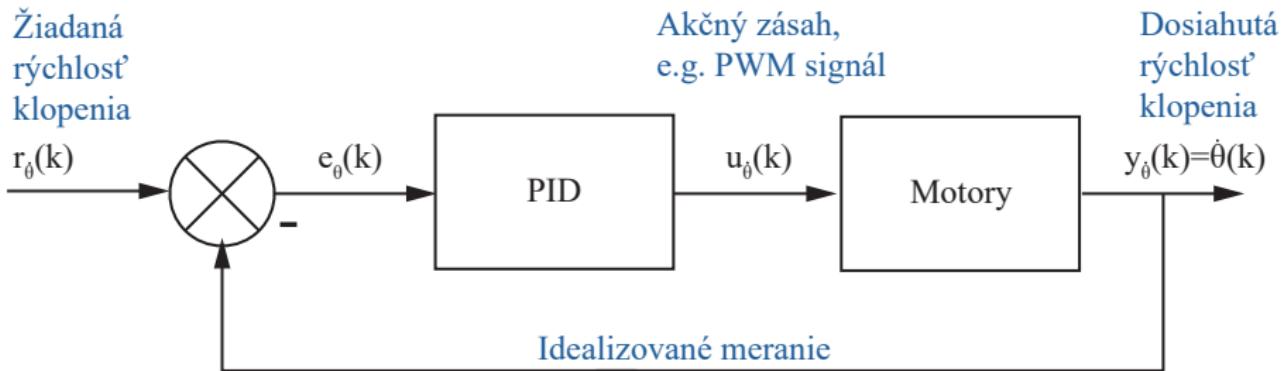
Riadenie dronov

prof. Ing. Gergely Takács, PhD.

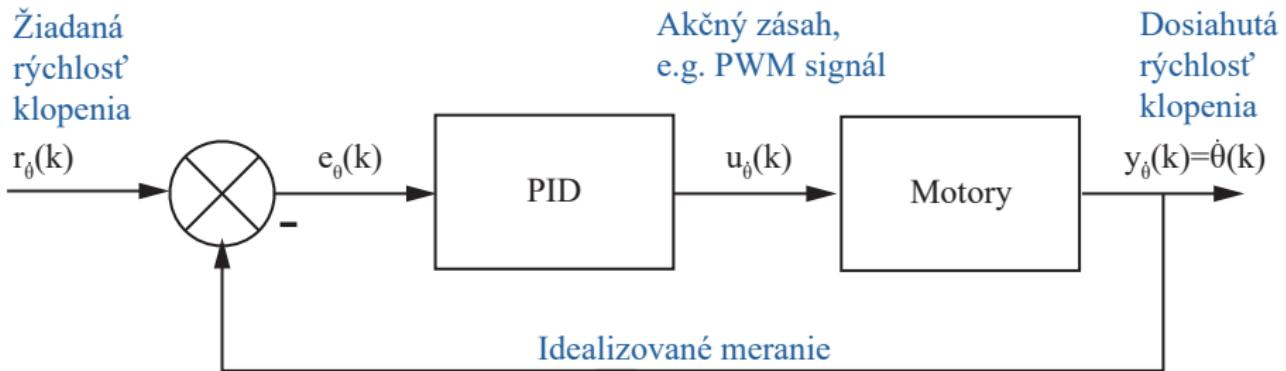




- Tvoríme sa, že poznáme žiadanú uhlovú rýchlosť orientácie $r_{\dot{\Theta}}$, $r_{\dot{\phi}}$ a $r_{\dot{\psi}}$ a pre jednoduchosť sústredme len na rýchlosť zmeny klopenia ($\dot{\Theta}$).
- Sme na najnižšej úrovni riadenia — je to zároveň aj najdôležitejšia slučka [Hall 2018]



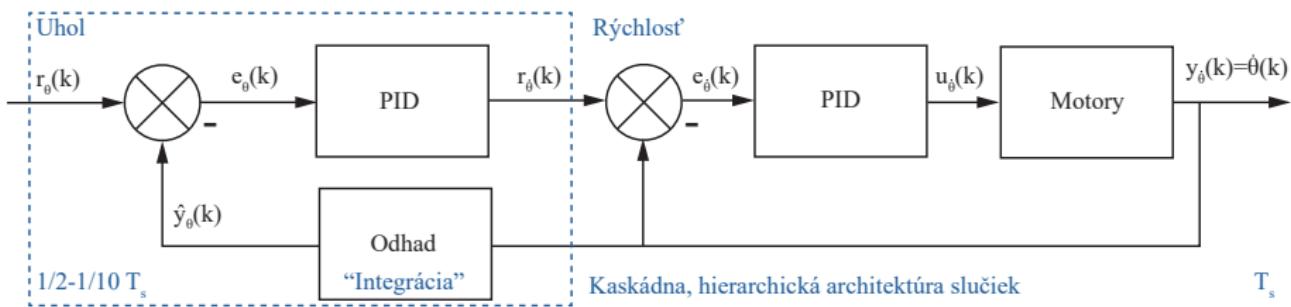
- Tvoríme sa, že poznáme žiadanú uhlovú rýchlosť orientácie $r_{\dot{\Theta}}$, $r_{\dot{\phi}}$ a $r_{\dot{\psi}}$ a pre jednoduchosť sústredme len na rýchlosť zmeny klopenia ($\dot{\Theta}$).
- Sme na najnižšej úrovni riadenia — je to zároveň aj najdôležitejšia slučka [Hall 2018]



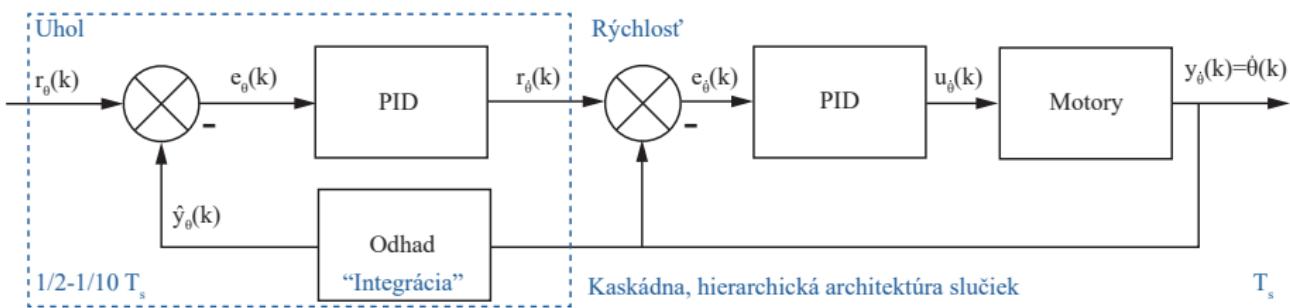
- Riadiť orientáciu len na základe zmeny rýchlosťi by bolo dosť neintuitívne, potrebujeme prepočítať $r_\Theta \rightarrow r_{\dot{\Theta}}$
- Tvárme sa, že poznáme žiadane orientácie, napr. klopenie r_Θ .
- Riadenie orientácie môže byť riešené ďalšou, nadradenou regulačnou slučkou - hovoríme o tzv. 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)

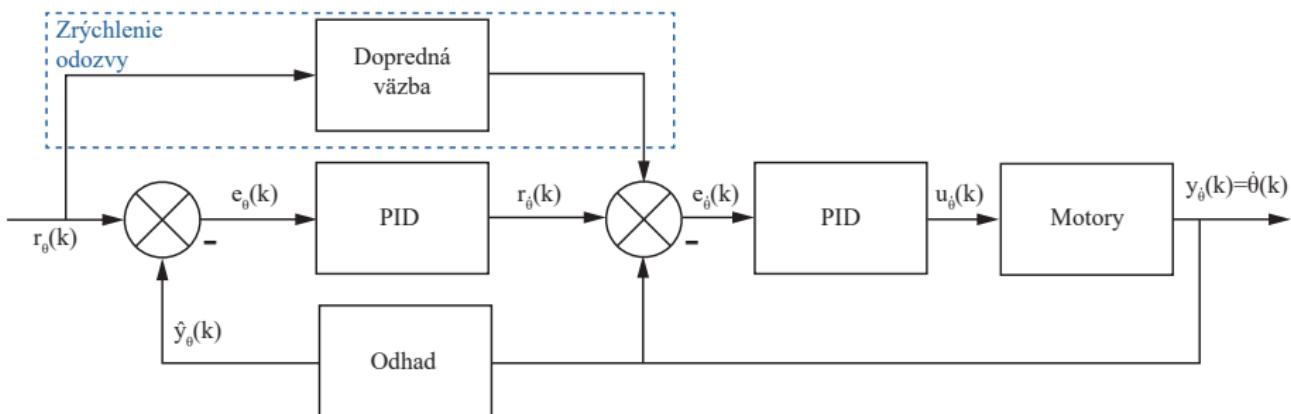


- Pri PID skôr PI¹, lebo
 - ▶ iba P neodstráni, trvalú odchýlku
 - ▶ D reaguje príliš agresívne na šum.
- Do slučky dopracujeme doprednú väzbu (*angl.*: feedforward) ktorá zrýchli odozvu regulácie
- “Whatever works” - netreba mystifikovať, je to len matematika²

¹ArduCopter používa P regulátor

²napr.

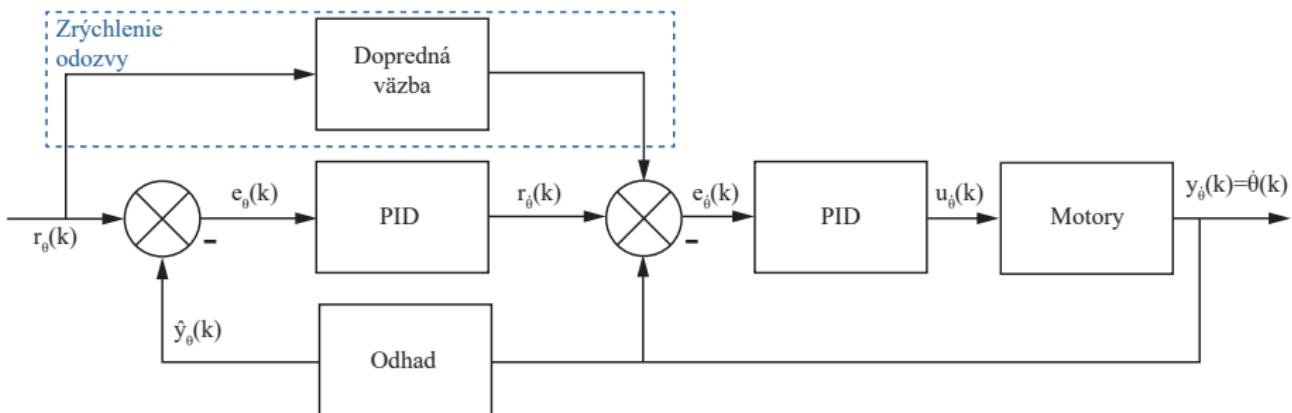
- Pri PID skôr PI¹, lebo
 - ▶ iba P neodstráni, trvalú odchýlku
 - ▶ D reaguje príliš agresívne na šum.
- Do slučky dopracujeme doprednú väzbu (*angl.: feedforward*) ktorá zrýchli odozvu regulácie
- “Whatever works” - netreba mystifikovať, je to len matematika²



¹ArduCopter používa P regulátor

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 používa P regulátor

²napr.

- Akčné zásahy majú svoje ohraničenia ((*angl.*: constraints)), ako donútime ich dodržanie? Saturáciou (orezávaním) hodnôt, ktoré skutočne vypočíta PID.
- Saturácia vnáša nelinearitu, vplyva 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 vypočíta PID.
- Saturácia vnáša nelinearitu, vplyva 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 vypočíta 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 to zbytočne nahromadiť (*angl.*: windup).
- Akonáhle 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 nahromadiť (*angl.*: windup).
- Akonáhle 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
- 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 nahromadiť (*angl.*: windup).
- Akonáhle 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
- 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 nahromadiť (*angl.*: windup).
- Akonáhle 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
- 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.

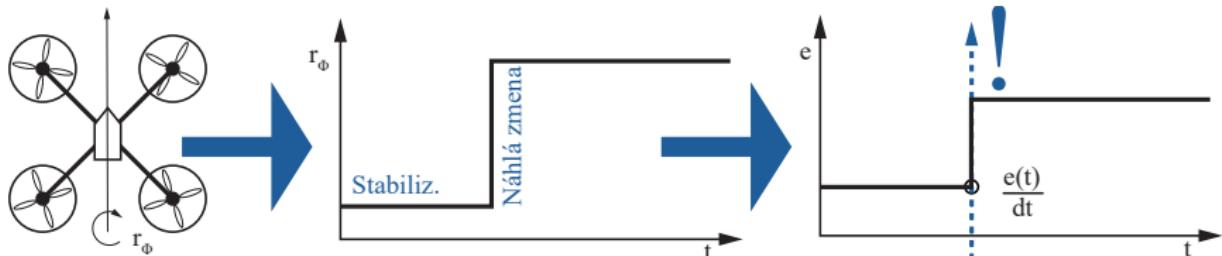
- ArduPilot - Ak akčný člen je akurát saturovaný, podrž hodnotu I zložky. Nižšie to môže ísť, vyššie nie [Hall 2018].
- ArduPilot - Ked'že máme kaskádnu konfiguráciu PID slučiek, saturačný znak postupuje cez hierarchiu nižšie a nižšie aby zastavil nahromadenie I zložky [Hall 2018].

- 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].

- 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 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

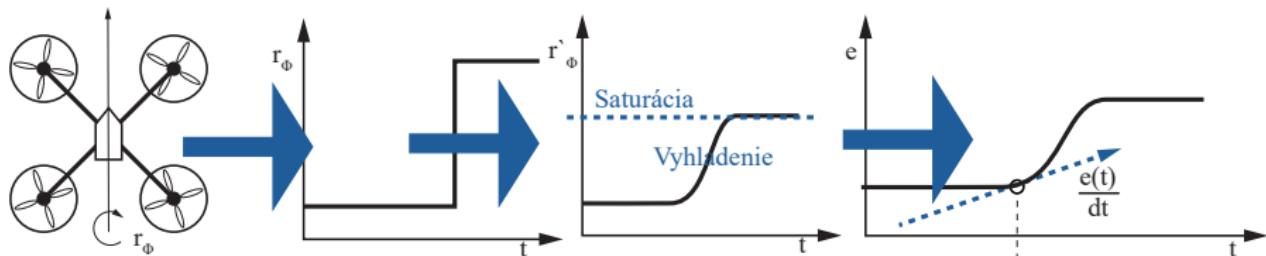
³ArduCopter 50 Hz → 400 Hz [Hall 2018]

- Dron je stabilizovaný, pilot/ROS náhle chce $r_\phi = +30^\circ$ klonenie. Čo sa stane s riadením?
- Žiadaná hodnota je skokový signál. Derivácia skoku je... D zložka 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



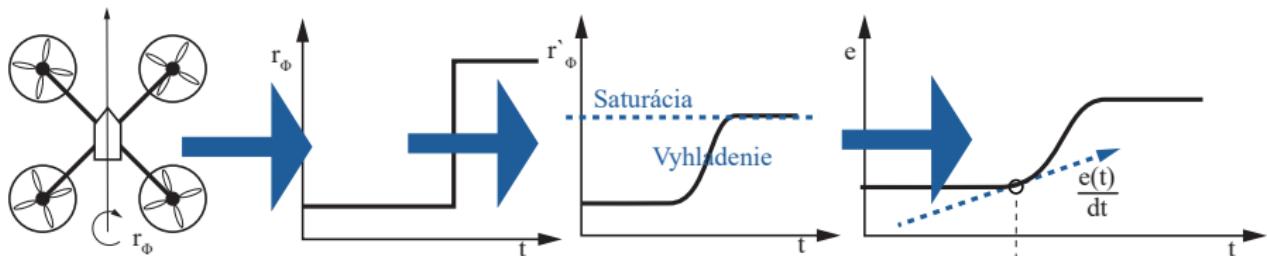
³ArduCopter 50 Hz → 400 Hz [Hall 2018]

- 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 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 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]

Notes: [Hall 2018](#)

User (ROS) → Shaping → PID → Actuators

50 Hz -> 400 Hz

Ďakujem za Vašu pozornosť.

- [1] 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.