

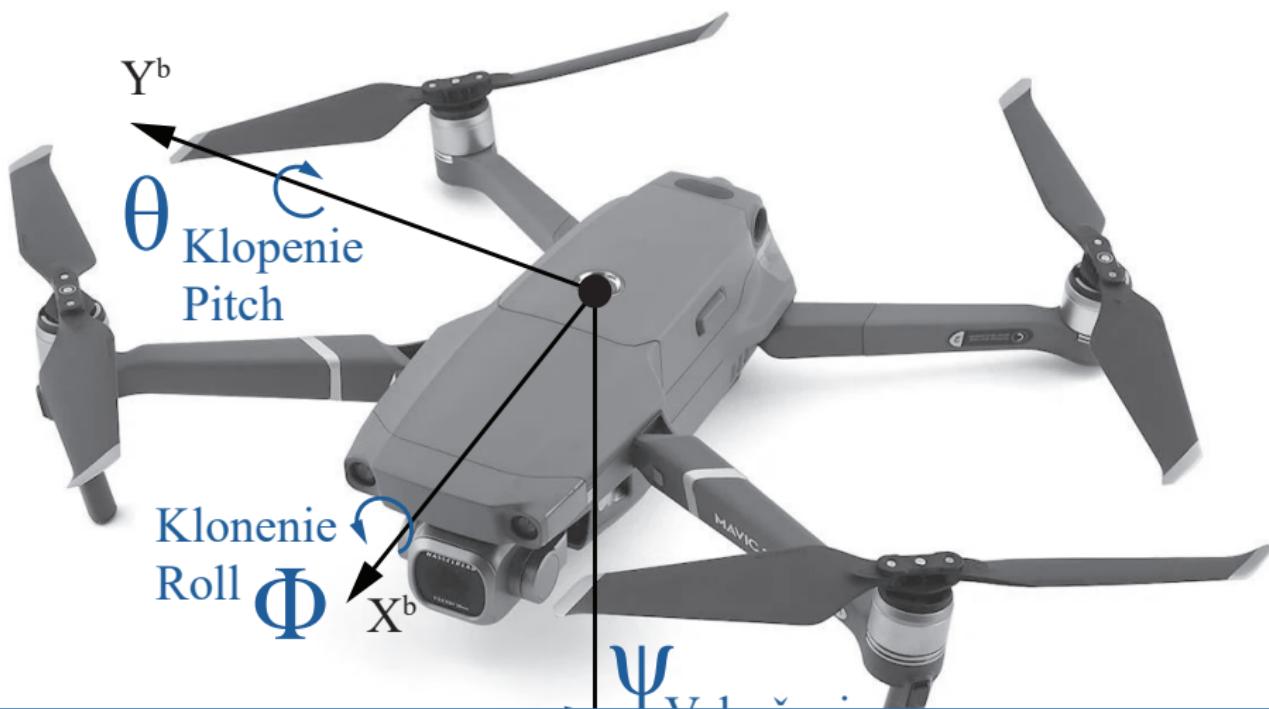
Working title drone control

Riadenie dronov

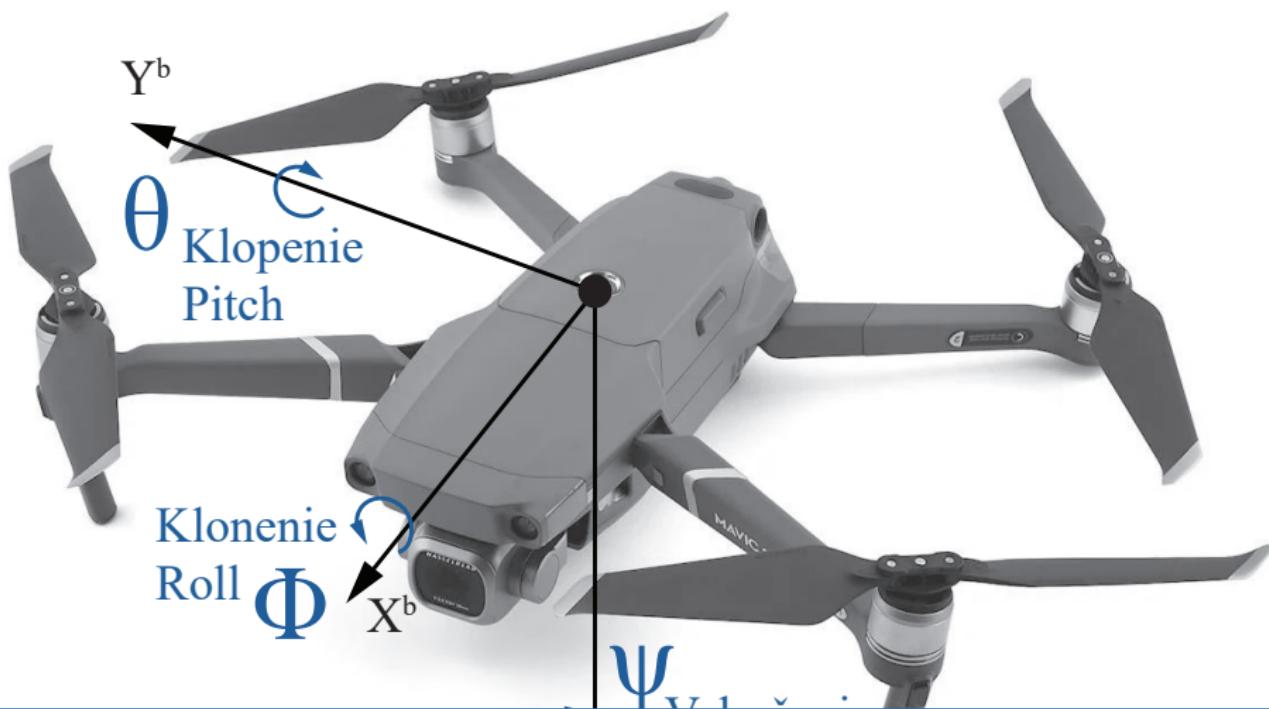
prof. Ing. Gergely Takács, PhD.



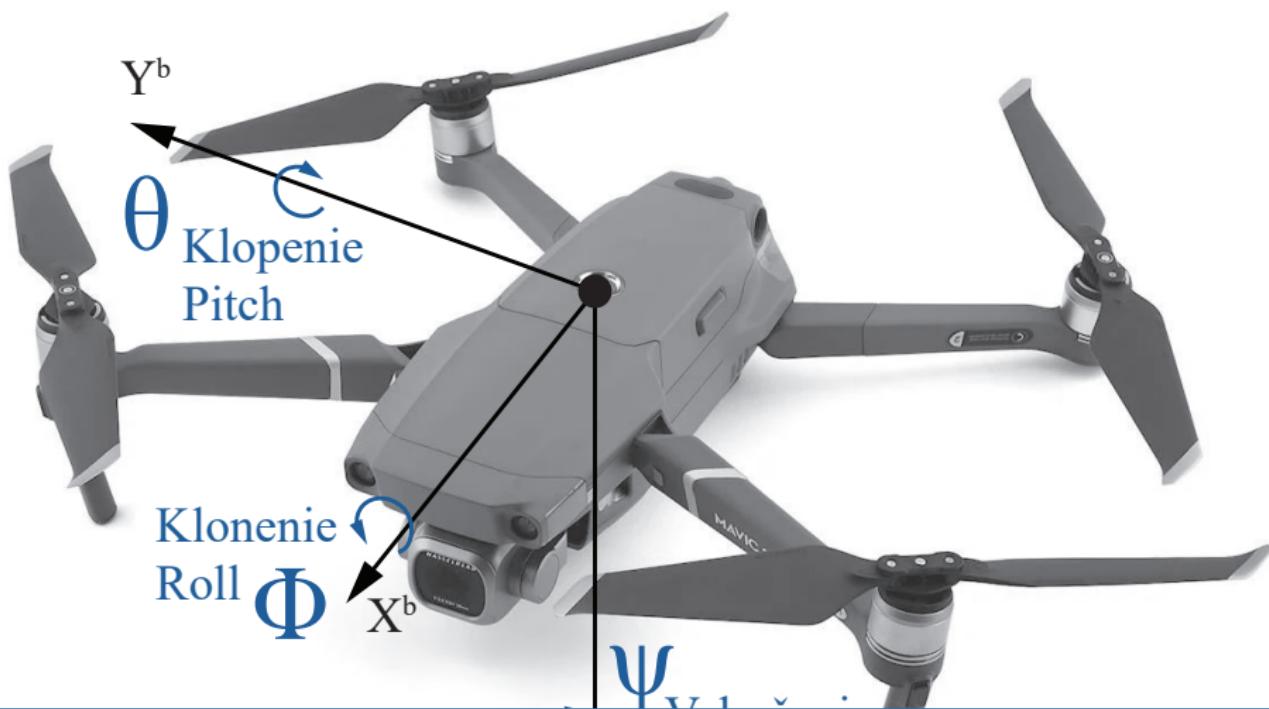
- Nestabilná (*angl.*: unstable) - duh...
- Nelineárna (*angl.*: nonlinear) - uhly, atd'.
- (*angl.*: underactuated) - 4 vstupy, 6 DOF výstup [Douglas 2018]



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

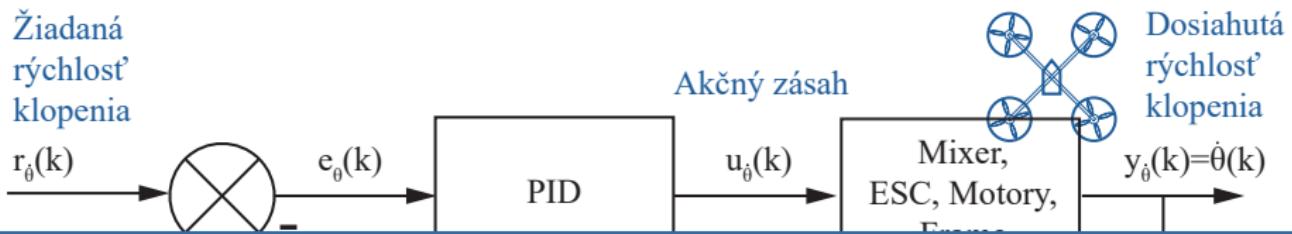


- Nestabilná (*angl.*: unstable) - duh...
- Nelineárna (*angl.*: nonlinear) - uhly, atď'.
- (*angl.*: underactuated) - 4 vstupy, 6 DOF výstup [Douglas 2018]

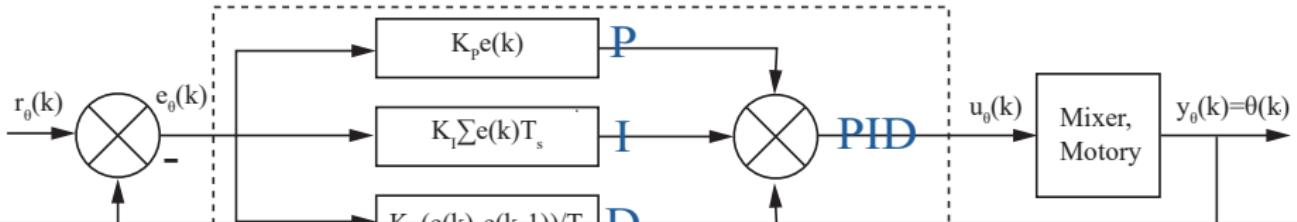


Rýchlosť zmeny uhlov

- Tvárame sa, že poznáme žiadanú uhlovú rýchlosť orientácie $r_{\dot{\Theta}}$, $r_{\dot{\phi}}$ a $r_{\dot{\psi}}$ resp r_h (vert. rýchlosť) 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 4 slučky [Hall 2018; PX4 Autopilot 2021b] .
- Slučky sú nezávislé len pre malé uhly, lebo keď zvýšime letovú výšku h pri väčších e.g. Θ tak zvýšime aj vertikálnu aj horizontálnu rýchlosť [Douglas 2018] !
- 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

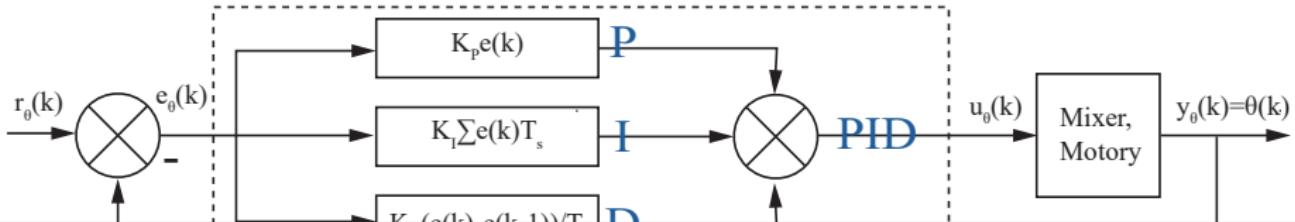


- Tvárame sa, že poznáme žiadanú uhlovú rýchlosť orientácie $r_{\dot{\Theta}}$, $r_{\dot{\phi}}$ a $r_{\dot{\psi}}$ resp r_h (vert. rýchlosť) a pre jednoduchosť sústredme len na rýchlosť zmeny klopenia ($\dot{\Theta}$).
- Sme na najnižšej úrovni, t.j. riadenie uhlovej rýchlosť (*angl.*: rate controller) — je to zároveň aj najdôležitejšia slučka, a máme 4 slučky [Hall 2018; PX4 Autopilot 2021b] .
- Slučky sú nezávislé len pre malé uhly, lebo keď zvýšime letovú výšku h pri väčších e.g. Θ tak zvýšime aj vertikálnu aj horizontálnu rýchlosť [Douglas 2018] !
- 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



Rýchlosť zmeny uhlov

- Tvárame sa, že poznáme žiadanú uhlovú rýchlosť orientácie $r_{\dot{\Theta}}$, $r_{\dot{\phi}}$ a $r_{\dot{\psi}}$ resp r_h (vert. rýchlosť) 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 4 slučky [Hall 2018; PX4 Autopilot 2021b].
- Slučky sú nezávislé len pre malé uhly, lebo keď zvýšime letovú výšku h pri väčších e.g. Θ tak zvýšime aj vertikálnu aj horizontálnu rýchlosť [Douglas 2018] !
- 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 $r_{\dot{\psi}}$ resp r_h (vert. rýchlosť) a pre jednoduchosť sústredme len na rýchlosť zmeny klopenia ($\dot{\Theta}$).
- Sme na najnižšej úrovni, t.j. riadenie uhlovej rýchlosťi (*angl.*: rate controller) — je to zároveň aj najdôležitejšia slučka, a máme 4 slučky [Hall 2018; PX4 Autopilot 2021b] .
- Slučky sú nezávislé len pre malé uhly, lebo keď zvýšime letovú výšku h pri väčších e.g. Θ tak zvýšime aj vertikálnu aj horizontálnu rýchlosť [Douglas 2018] !
- 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

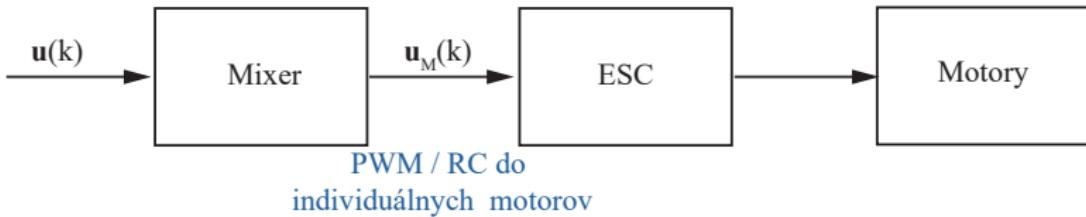
- Aj keď slučky PID na tejto úrovni definujú žiadanú zmenu $u(k)$, musíme implementovať pridelenie riadenia (*angl.*: control allocation) $u_M(k)$ (PWM, RC impulzy) do individuálnych motorov cez elektronické riadenie rýchlosťi (*angl.*: electronic speed control, ESC).
- V najjednoduchšom prípade máme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

$$u_{M1}(k) = u_h + u_\phi + u_\Theta + u_\Sigma \quad (1)$$

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

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

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



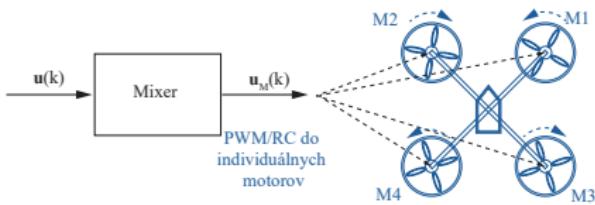
- Aj keď slučky PID na tejto úrovni definujú žiadanú zmenu $u(k)$, musíme implementovať pridelenie riadenia (*angl.*: control allocation) $u_M(k)$ (PWM, RC impulzy) do individuálnych motorov cez elektronické riadenie rýchlosťi (*angl.*: electronic speed control, ESC).
- V najjednoduchšom prípade máme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

$$u_{M1}(k) = u_h + u_\phi + u_\Theta + u_\Sigma \quad (1)$$

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

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

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





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

- 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].



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

- 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].

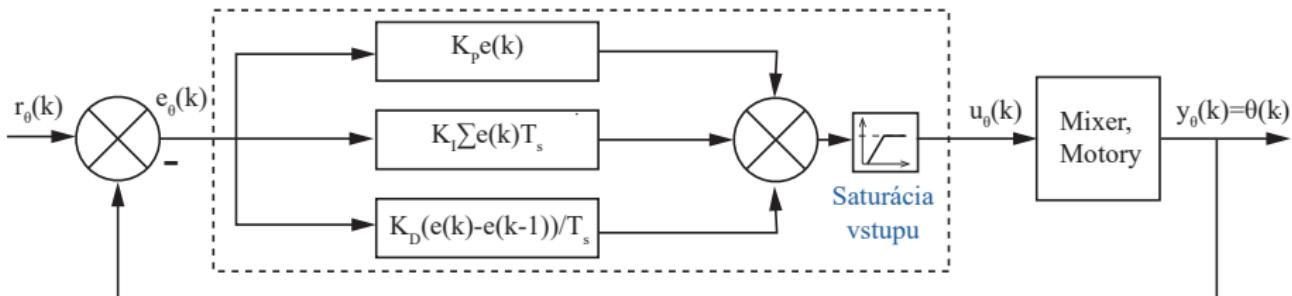


$$\begin{bmatrix} u_{M1}(k) \\ u_{M2}(k) \\ u_{M3}(k) \\ u_{M4}(k) \end{bmatrix} = \underbrace{\begin{bmatrix} +1 & +1 & +1 & +1 \\ +1 & -1 & +1 & -1 \\ +1 & +1 & -1 & -1 \\ +1 & -1 & -1 & +1 \end{bmatrix}}_P \begin{bmatrix} u_h \\ u_\phi \\ u_\theta \\ u_\Sigma \end{bmatrix} \quad (5)$$

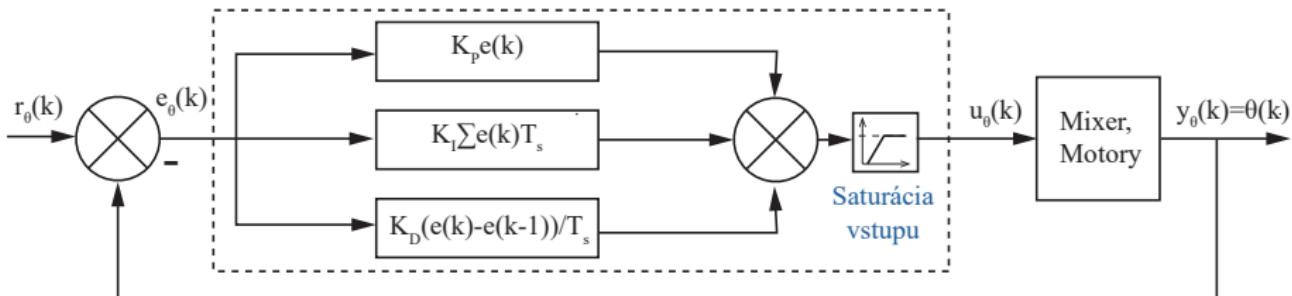
- Mapa $u_M(k) = Pu(k)$ kde P matica pridelenia riadenia (*angl.: control allocation matrix*). Nemusí obsahovať iba jednotky, môže byť aj neštvorcová ale našťastie transformácia je stále lineárna [Bresciani a kol. 2020].
- Môžeme priamo namapovať maticu efektivity akčných členov (*angl.: actuator effectiveness 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?

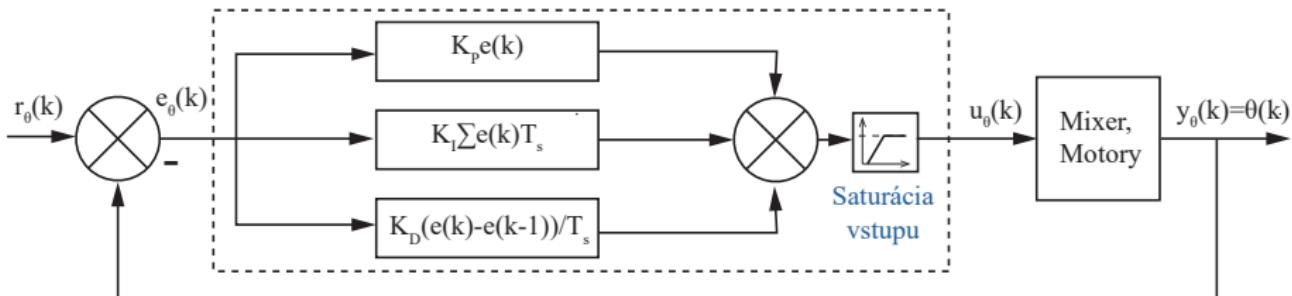
- Akčné zásahy majú svoje ohraničenia ((angl.: constraints))
- Ako donútime ich dodržanie? Saturáciou (orezávaním) hodnôt, ktoré skutočne vypočítava PID.
- Saturácia vnáša nelinearitu, vplýva na výkon riadenia aj stabilitu.
- Aj iné veličiny môžeme saturovať, napr. žiadane hodnoty. Ako by sme ohraničili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



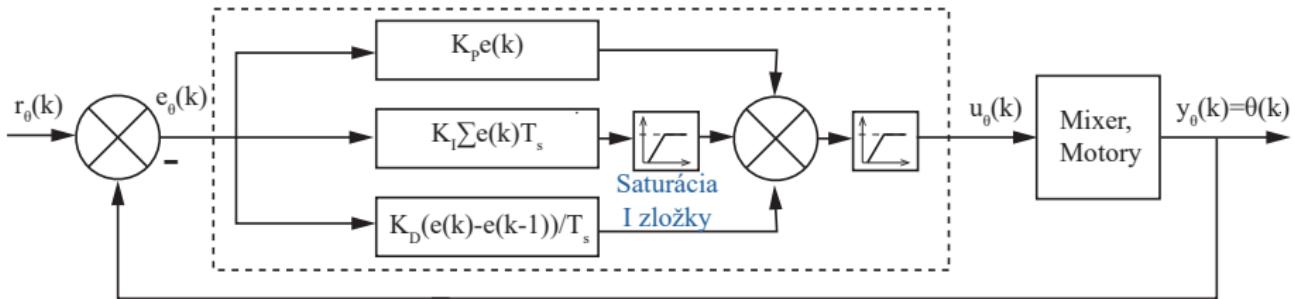
- Akčné zásahy majú svoje ohraničenia ((angl.: constraints))
- Ako donútime ich dodržanie? Saturáciou (orezávaním) hodnôt, ktoré skutočne vypočítava PID.
- Saturácia vnáša nelinearitu, vplýva na výkon riadenia aj stabilitu.
- Aj iné veličiny môžeme saturovať, napr. žiadane hodnoty. Ako by sme ohraničili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



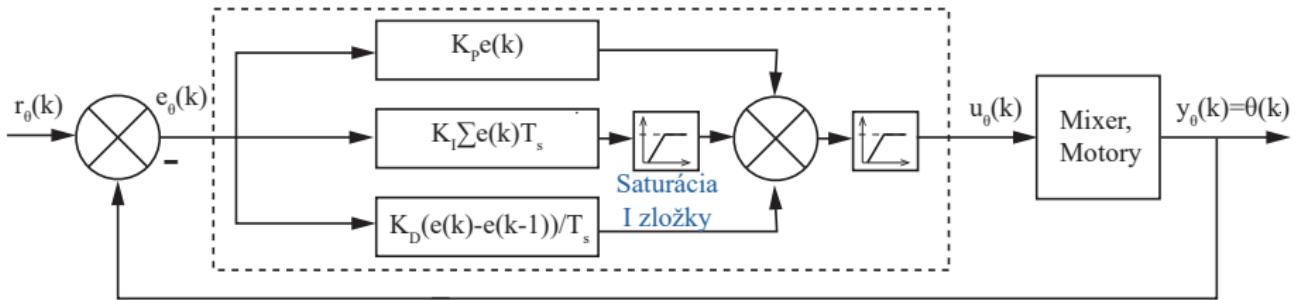
- Akčné zásahy majú svoje ohraničenia ((angl.: constraints))
- Ako donútime ich dodržanie? Saturáciou (orezávaním) hodnôt, ktoré skutočne vypočítava PID.
- Saturácia vnáša nelinearitu, vplýva na výkon riadenia aj stabilitu.
- Aj iné veličiny môžeme saturovať, napr. žiadane hodnoty. Ako by sme ohraničili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



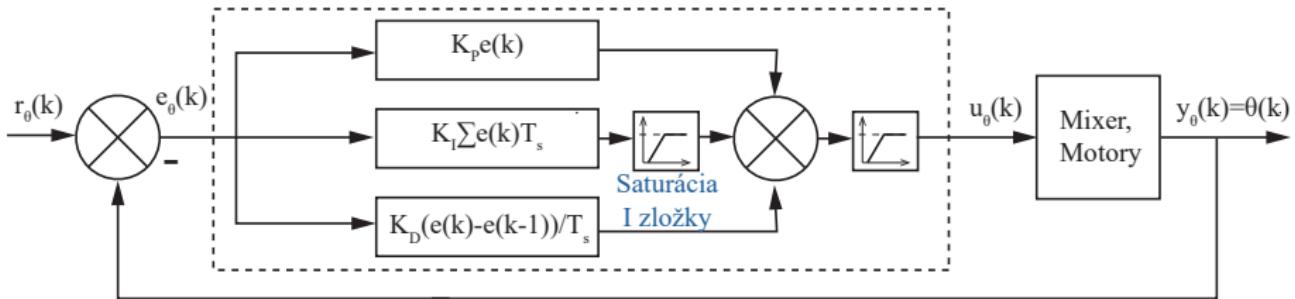
- Integračná zložka ráta odchýlku v minulosti, preto ak akčné členy sú už na hraniciach možností - začína sa nahromadovať (*angl.*: windup).
- Akonáhle sa vrátia akčné zásahy pod ohraničenia, nahromadená I zložka stále bude tlačiť systém na hranice možností, musí sa to chvíľu "uvolňovať" (*angl.*: unwind)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



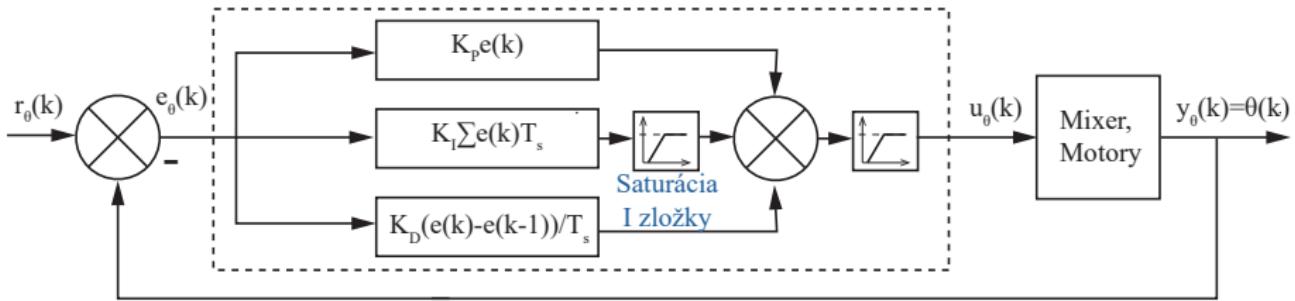
- Integračná zložka ráta odchýlku v minulosti, preto ak akčné členy sú už na hraniciach možností - začína sa nahromadať (*angl.*: windup).
- Akonáhle sa vrátia akčné zásahy pod ohraničenia, nahromadená I zložka stále bude tlačiť systém na hranice možností, musí sa to chvíľu "uvolňovať" (*angl.*: unwind)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

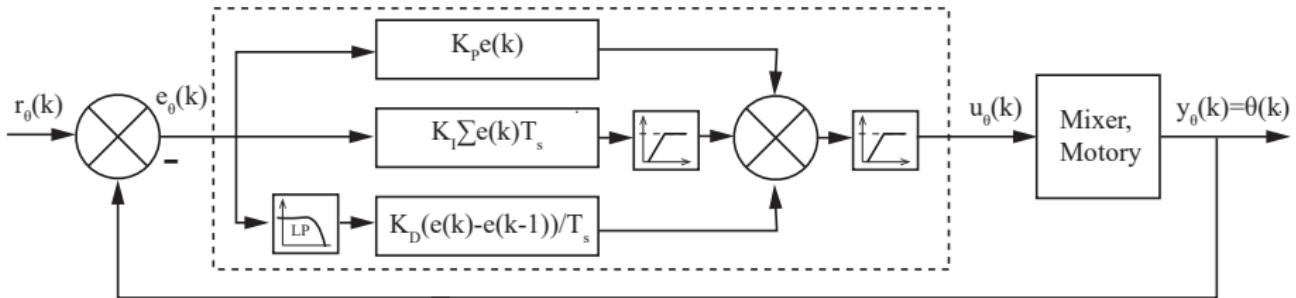


- Integračná zložka ráta odchýlku v minulosti, preto ak akčné členy sú už na hraniciach možností - začína sa nahromadať (*angl.*: windup).
- Akonáhle sa vrátia akčné zásahy pod ohraničenia, nahromadená I zložka stále bude tlačiť systém na hranice možností, musí sa to chvíľu "uvolňovať" (*angl.*: unwind)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

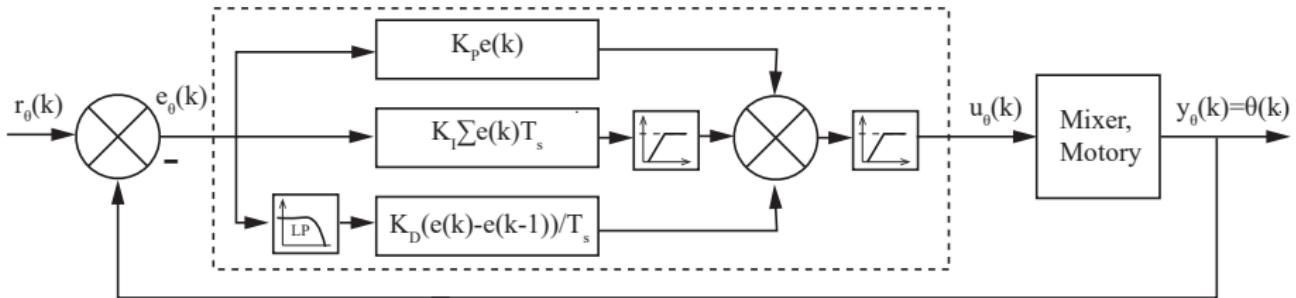


- Integračná zložka ráta odchýlku v minulosti, preto ak akčné členy sú už na hraniciach možností - začína sa nahromadovať (*angl.*: windup).
- Akonáhle sa vrátia akčné zásahy pod ohraničenia, nahromadená I zložka stále bude tlačiť systém na hranice možností, musí sa to chvíľu "uvolňovať" (*angl.*: unwind) 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 [\[Hall 2018\]](#).

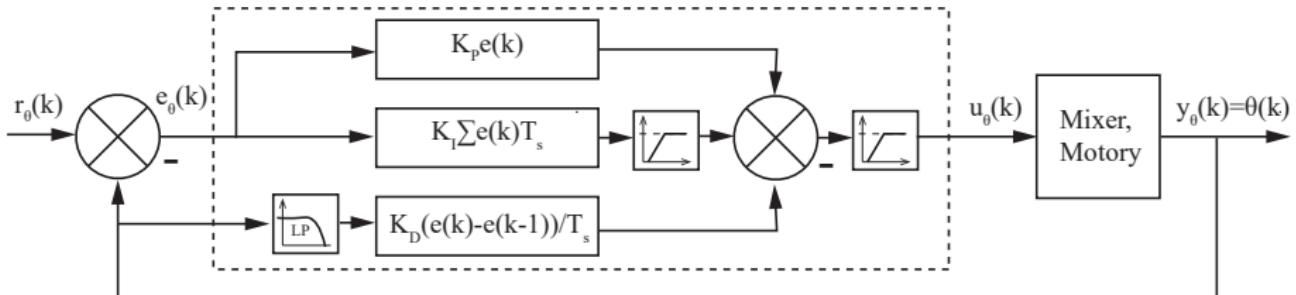
- Šum zo snímačov môže propagovať cez výpočet odchýlky riadenia do derivačnej zložky Vysokofrekvenčné zložky potom navýšia D zložku (čo je derivácia impulzu?)
- Riešenie: Odchýlku riadenia pustíme cez dolnopriepustný (angl.: low-pass) filter (LPF) — Aj ArduCopter (20 Hz LPF) aj PX4 Autopilot používa [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 $y_\Theta(k)^1$



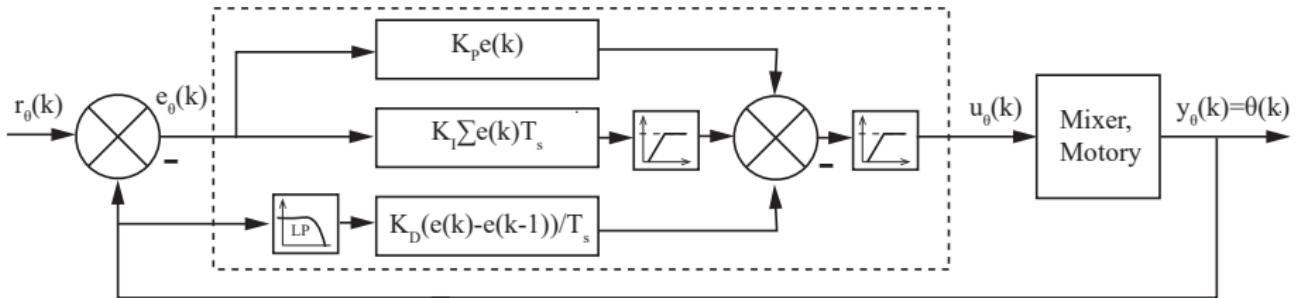
- Šum zo snímačov môže propagovať cez výpočet odchýlky riadenia do derivačnej zložky Vysokofrekvenčné zložky potom navýšia D zložku (čo je derivácia impulzu?)
- Riešenie: Odchýlku riadenia pustíme cez dolnopriepustný (angl.: low-pass) filter (LPF) — Aj ArduCopter (20 Hz LPF) aj PX4 Autopilot používa [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 $y_\theta(k)^1$

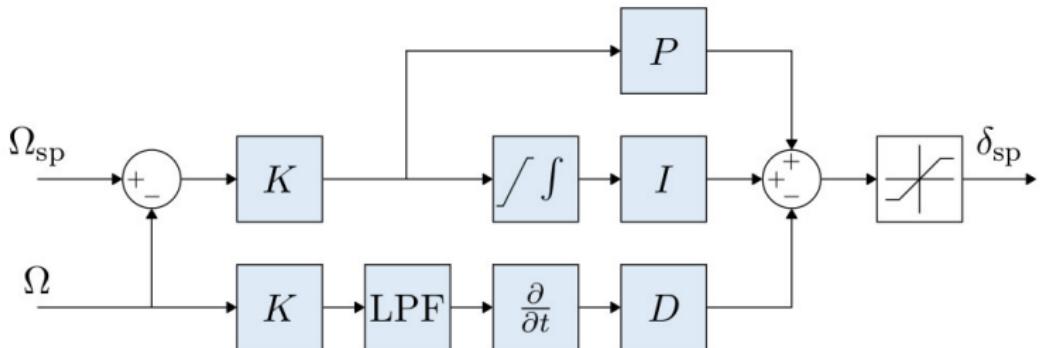


- Šum zo snímačov môže propagovať cez výpočet odchýlky riadenia do derivačnej zložky Vysokofrekvenčné zložky potom navýšia D zložku (čo je derivácia impulzu?)
- Riešenie: Odchýlku riadenia pustíme cez dolnopriepustný (*angl.*: low-pass) filter (LPF) — Aj ArduCopter (20 Hz LPF) aj PX4 Autopilot používa [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 $y_\Theta(k)^1$

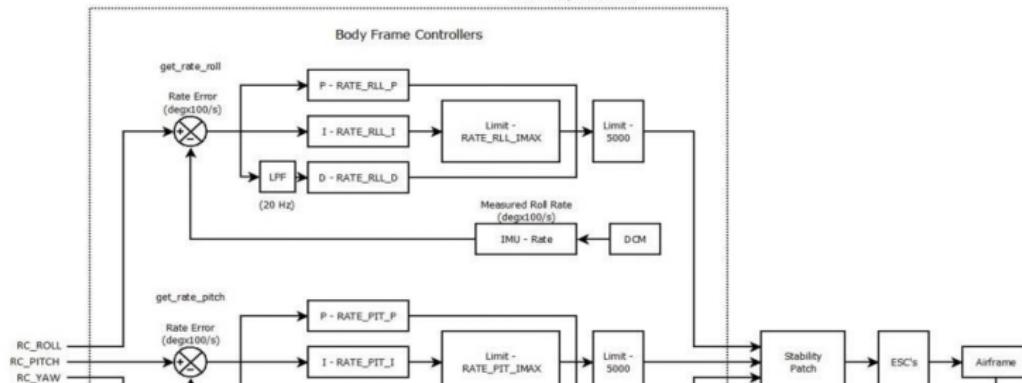


- Šum zo snímačov môže propagovať cez výpočet odchýlky riadenia do derivačnej zložky Vysokofrekvenčné zložky potom navýšia D zložku (čo je derivácia impulzu?)
- Riešenie: Odchýlku riadenia pustíme cez dolnopriepustný (*angl.*: low-pass) filter (LPF) — Aj ArduCopter (20 Hz LPF) aj PX4 Autopilot používa [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 $y_\theta(k)^1$





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



- riadenie orientácie, tj. uhly $\Omega = \Phi, \Theta, \Sigma$ a uhlové rýchlosťi $\dot{\Omega}$ v "body frame"
- Pre RC dron by to aj stačilo s plynom $h \leftarrow \tau$ (angl.: throttle)² [Boland 2015]
- Riadiť orientáciu (angl.: attitude) len na základe zmeny uhl. rýchlosťi (angl.: rate) by bolo dosť neintuitívne, potrebujeme prepočítať $r_\Theta \rightarrow r_{\dot{\Theta}}$
- Táto slučka je často nazvaná ako stabilizačné riadenie (angl.: stabilize) alebo orientačné riadenie (angl.: attitude control) (predošlá je (angl.: rate))
- Nezabudnime na štvrtú kaskadovanú slučku: výška \rightarrow rýchlosť zmeny výšky resp. vertikálna rýchlosť (angl.: climb rate) \rightarrow zrýchlenie \rightarrow motor

²Síce rate-control je tiež možné [Boland 2015] !

- riadenie orientácie, tj. uhly $\Omega = \Phi, \Theta, \Sigma$ a uhlové rýchlosťi $\dot{\Omega}$ v "body frame"
- Pre RC dron by to aj stačilo s plynom $h \leftarrow \tau$ (angl.: throttle)² [Boland 2015]
- Riadiť orientáciu (angl.: attitude) len na základe zmeny uhl. rýchlosťi (angl.: rate) by bolo dosť neintuitívne, potrebujeme prepočítať $r_\Theta \rightarrow r_{\dot{\Theta}}$
- Táto slučka je často nazvaná ako stabilizačné riadenie (angl.: stabilize) alebo orientačné riadenie (angl.: attitude control) (predošlá je (angl.: rate))
- Nezabudnime na štvrtú kaskadovanú slučku: výška \rightarrow rýchlosť zmeny výšky resp. vertikálna rýchlosť (angl.: climb rate) \rightarrow zrýchlenie \rightarrow motor

²Síce rate-control je tiež možné [Boland 2015] !

- riadenie orientácie, tj. uhly $\Omega = \Phi, \Theta, \Sigma$ a uhlové rýchlosťi $\dot{\Omega}$ v "body frame"
- Pre RC dron by to aj stačilo s plynom $h \leftarrow \tau$ (angl.: throttle)² [Boland 2015]
- Riadiť orientáciu (angl.: attitude) len na základe zmeny uhl. rýchlosťi (angl.: rate) by bolo dosť neintuitívne, potrebujeme prepočítať $r_\Theta \rightarrow r_{\dot{\Theta}}$
- Táto slučka je často nazvaná ako stabilizačné riadenie (angl.: stabilize) alebo orientačné riadenie (angl.: attitude control) (predošlá je (angl.: rate))
- Nezabudnime na štvrtú kaskadovanú slučku: výška \rightarrow rýchlosť zmeny výšky resp. vertikálna rýchlosť (angl.: climb rate) \rightarrow zrýchlenie \rightarrow motor

²Síce rate-control je tiež možné [Boland 2015] !

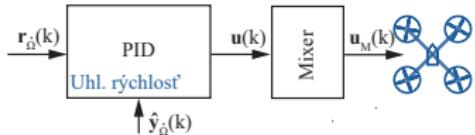
- riadenie orientácie, tj. uhly $\Omega = \Phi, \Theta, \Sigma$ a uhlové rýchlosťi $\dot{\Omega}$ v "body frame"
- Pre RC dron by to aj stačilo s plynom $h \leftarrow \tau$ (angl.: throttle)² [Boland 2015]
- Riadiť orientáciu (angl.: attitude) len na základe zmeny uhl. rýchlosťi (angl.: rate) by bolo dosť neintuitívne, potrebujeme prepočítať $r_\Theta \rightarrow r_{\dot{\Theta}}$
- Táto slučka je často nazvaná ako stabilizačné riadenie (angl.: stabilize) alebo orientačné riadenie (angl.: attitude control) (predošlá je (angl.: rate))
- Nezabudnime na štvrtú kaskadovanú slučku: výška \rightarrow rýchlosť zmeny výšky resp. vertikálna rýchlosť (angl.: climb rate) \rightarrow zrýchlenie \rightarrow motor

²Síce rate-control je tiež možné [Boland 2015] !

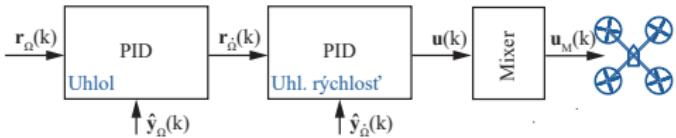
- riadenie orientácie, tj. uhly $\Omega = \Phi, \Theta, \Sigma$ a uhlové rýchlosťi $\dot{\Omega}$ v "body frame"
- Pre RC dron by to aj stačilo s plynom $h \leftarrow \tau$ (angl.: throttle)² [Boland 2015]
- Riadiť orientáciu (angl.: attitude) len na základe zmeny uhl. rýchlosťi (angl.: rate) by bolo dosť neintuitívne, potrebujeme prepočítať $r_\Theta \rightarrow r_{\dot{\Theta}}$
- Táto slučka je často nazvaná ako stabilizačné riadenie (angl.: stabilize) alebo orientačné riadenie (angl.: attitude control) (predošlá je (angl.: rate))
- Nezabudnime na štvrtú kaskadovanú slučku: výška \rightarrow rýchlosť zmeny výšky resp. vertikálna rýchlosť (angl.: climb rate) \rightarrow zrýchlenie \rightarrow motor

²Síce rate-control je tiež možné [Boland 2015] !

- Tvárme sa, že poznáme žiadané orientácie (napr. RC), napr. klopenie r_Θ a chceme riadiť y_Θ . V skutočnosti riešené s kvaternióny aby sme obišli singularitu Eulerových uhlov pri odhadе [Erasmus 2020]. Ak by sme priamo pilotovali RC , bol by to tzv. stabilizovaný letový mód [Boland 2015].
- Riadenie orientácie môže byť riešené ďalšou, nadradenou regulačnou slučkou - hovoríme o tzv. kaskádnom riadení (angl.: nested, cascaded).



- Tvárme sa, že poznáme žiadané orientácie (napr. RC), napr. klopenie r_Θ a chceme riadiť y_Θ . V skutočnosti riešené s kvaternióny aby sme obišli singularitu Eulerových uhlov pri odhadе [Erasmus 2020]. Ak by sme priamo pilotovali RC , bol by to tzv. stabilizovaný letový mód [Boland 2015].
- Riadenie orientácie môže byť riešené ďalšou, nadradenou regulačnou slučkou - hovoríme o tzv. kaskádnom riadení (*angl.*: nested, cascaded).

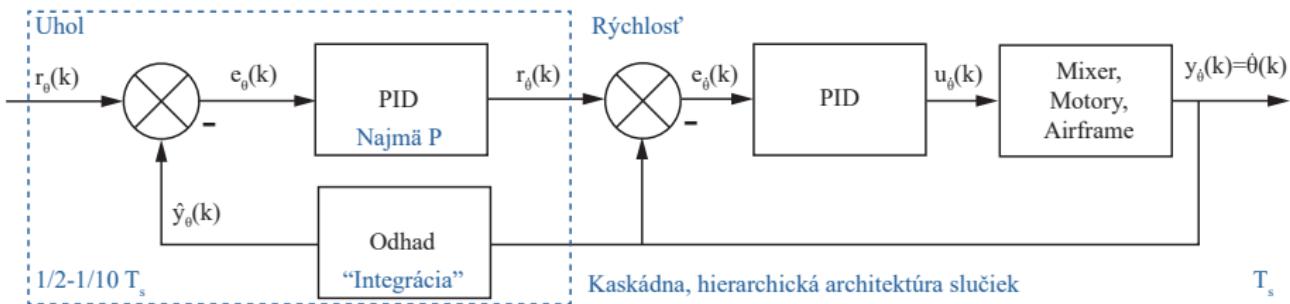


- Nadradené slučky sú pomalšie, vytvára to istý "filter", t.j. nemôžeme rýchlejšie ovládať rýchlosť ako polohu (cca. o rád, min polovicu pomalšie³) [Hall 2018; PX4 Autopilot 2021a]
- Pri PID skôr P⁴, lebo reaguje príliš agresívne na šum.
- Do slučky dopracujeme doprednú väzbu (*angl.*: feedforward) ktorá zrýchli odozvu regulácie. "Whatever works" - netreba mystifikovať.

³ArduCopter 40 Hz vs 400 Hz, PX4 250 vs. 1000 Hz [Hall 2018; PX4 Autopilot 2021a]

⁴ArduCopter a PX4 Autopilot používa P regulátor [PX4 Autopilot 2021a; ArduPilot 2021]

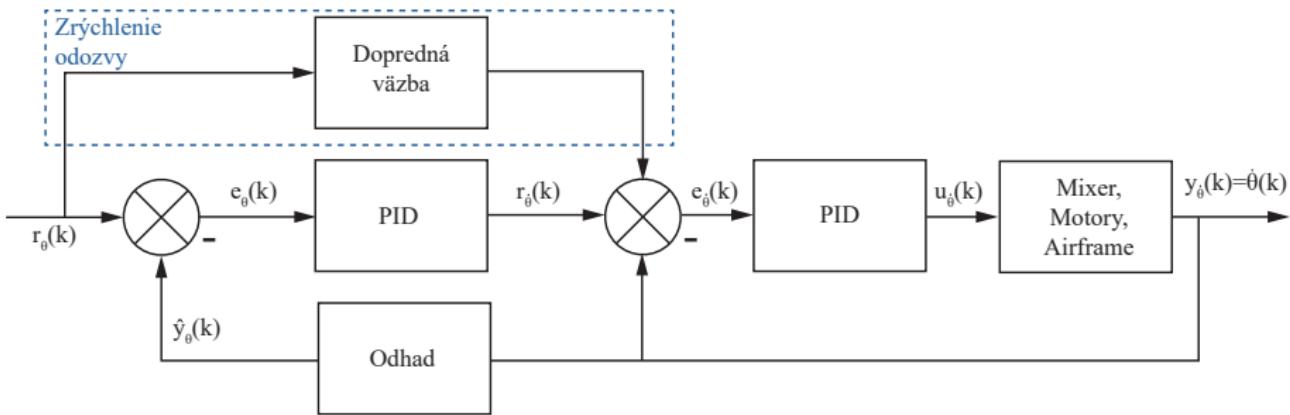
- Nadradené slučky sú pomalšie, vytvára to istý "filter", t.j. nemôžeme rýchlejšie ovládať rýchlosť ako polohu (cca. o rád, min polovicu pomalšie³) [Hall 2018; PX4 Autopilot 2021a]
- Pri PID skôr P⁴, lebo reaguje príliš agresívne na šum.
- Do slučky dopracujeme doprednú väzbu (*angl.*: feedforward) ktorá zrýchli odozvu regulácie. "Whatever works" - netreba mystifikovať.



³ArduCopter 40 Hz vs 400 Hz, PX4 250 vs. 1000 Hz [Hall 2018; PX4 Autopilot 2021a]

⁴ArduCopter a PX4 Autopilot používa P regulátor [PX4 Autopilot 2021a; ArduPilot 2021]

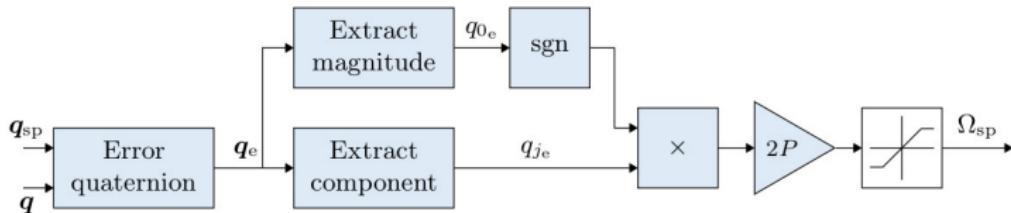
- Nadradené slučky sú pomalšie, vytvára to istý "filter", t.j. nemôžeme rýchlejšie ovládať rýchlosť ako polohu (cca. o rád, min polovicu pomalšie³) [Hall 2018; PX4 Autopilot 2021a]
- Pri PID skôr P⁴, lebo reaguje príliš agresívne na šum.
- Do slučky dopracujeme doprednú väzbu (*angl.*: feedforward) ktorá zrýchli odozvu regulácie. "Whatever works" - netreba mystifikovať.



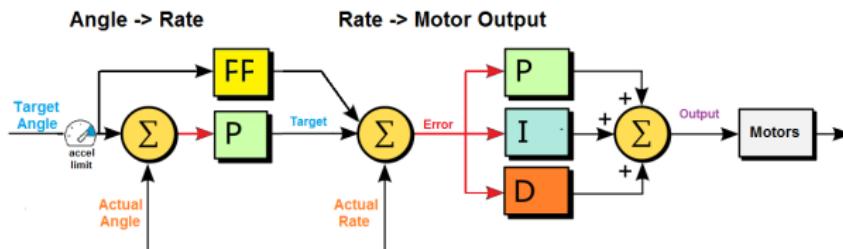
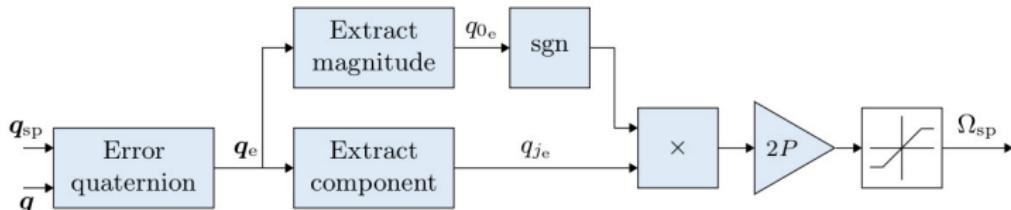
³ArduCopter 40 Hz vs 400 Hz, PX4 250 vs. 1000 Hz [Hall 2018; PX4 Autopilot 2021a]

⁴ArduCopter a PX4 Autopilot používa P regulátor [PX4 Autopilot 2021a; ArduPilot 2021]

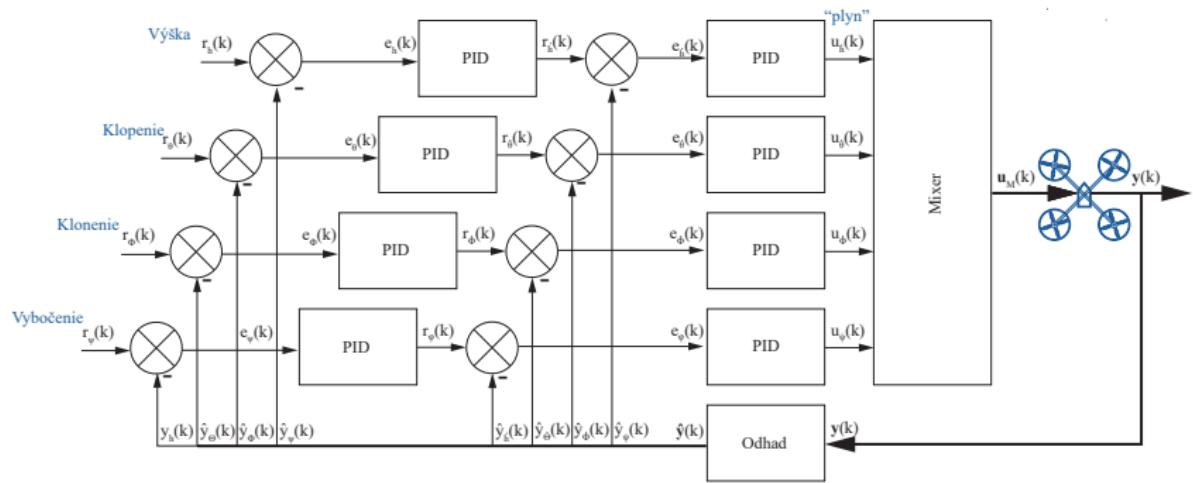
- PX4 používa kvaternióny, ale je to iba P regulátor
- ArduCopter v podstate taktiež tam má P regulátor + FF



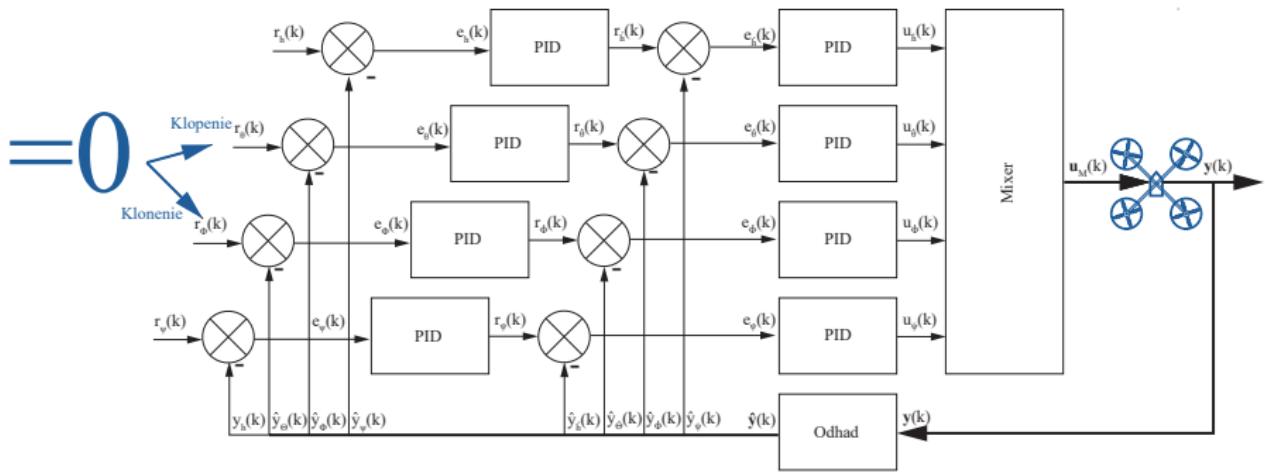
- PX4 používa kvaternióny, ale je to iba P regulátor
- ArduCopter v podstate taktiež tam má P regulátor + FF



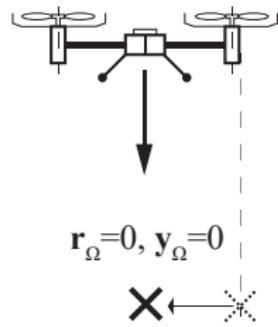
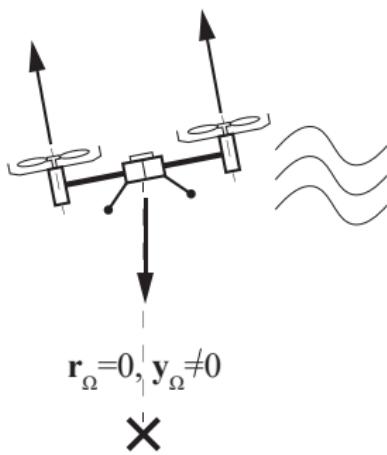
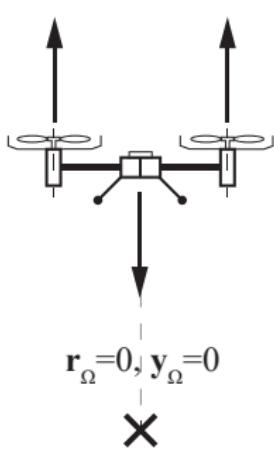
- Máme 4 nezávislých slučiek
- Čo sa stane ak chceme dron stabilizovať? Majme 0-vé Eulerove uhly!
- Určite to drží na jednom mieste?
- Potrebujeme nenulové klopenie a klonenie, a referenciu na základe polohy!



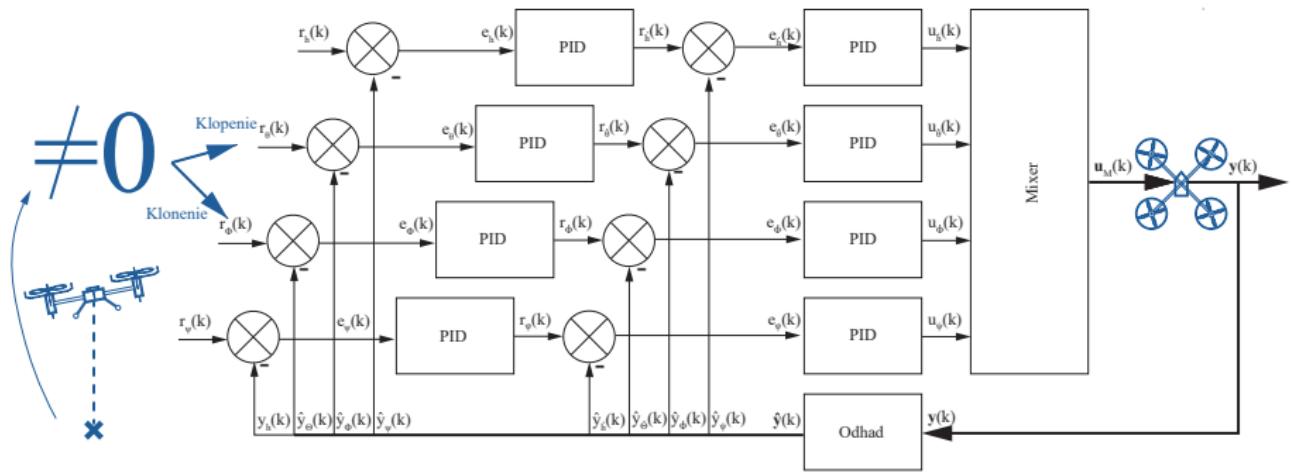
- Máme 4 nezávislých slučiek
- Čo sa stane ak chceme dron stabilizovať? Majme 0-vé Eulerove uhly!
- Určite to drží na jednom mieste?
- Potrebujeme nenulové klopenie a klonenie, a referenciu na základe polohy!



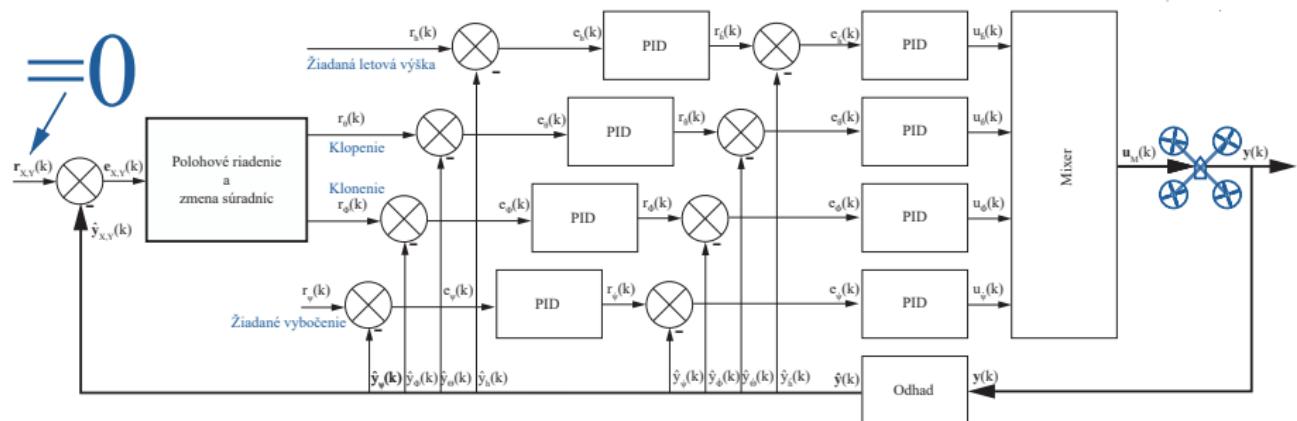
- Máme 4 nezávislých slučiek
- Čo sa stane ak chceme dron stabilizovať? Majme 0-vé Eulerove uhly!
- Určite to drží na jednom mieste?
- Potrebujeme nenulové klopenie a klonenie, a referenciu na základe polohy!



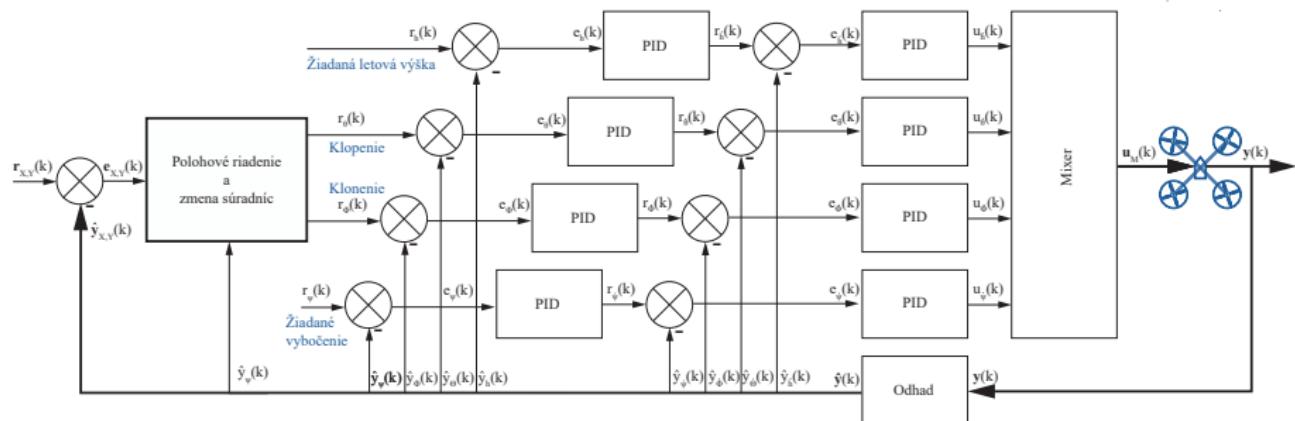
- Máme 4 nezávislých slučiek
- Čo sa stane ak chceme dron stabilizovať? Majme 0-vé Eulerove uhly!
- Určite to drží na jednom mieste?
- Potrebujeme nenulové klopenie a klonenie, a referenciu na základe polohy!



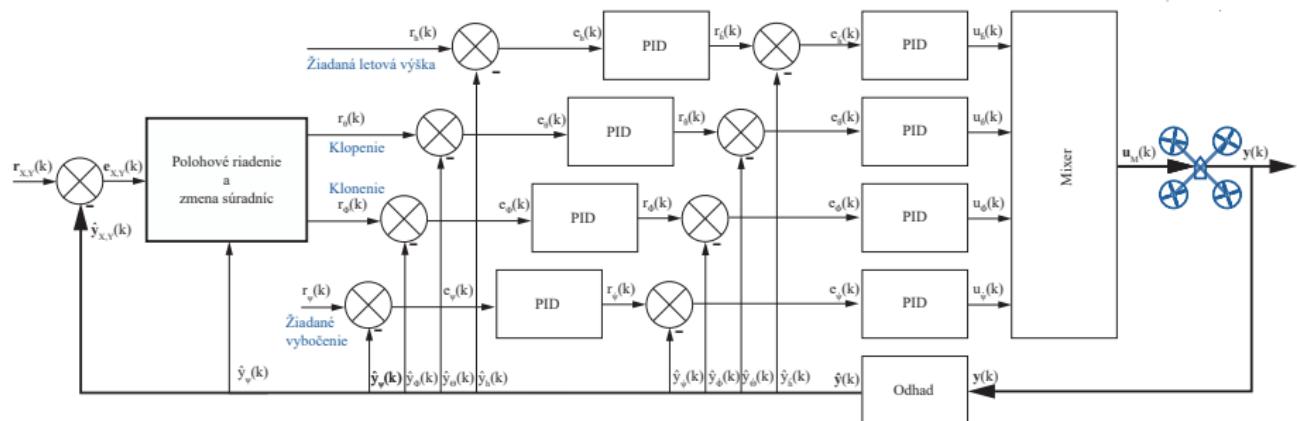
- Úlohou je držať polohu na základe globálnych súradníc (GS) napr. GPS
- Klopenie a klonenie určíme na základe GS súradníc, t.j. premeníme $X, Y \rightarrow \Theta, \Phi\dots$
- ... ale potrebujeme na výpočet aj želané/aktuálne vybočenie, t.j. $X, Y, \Psi \rightarrow \Theta, \Phi,$



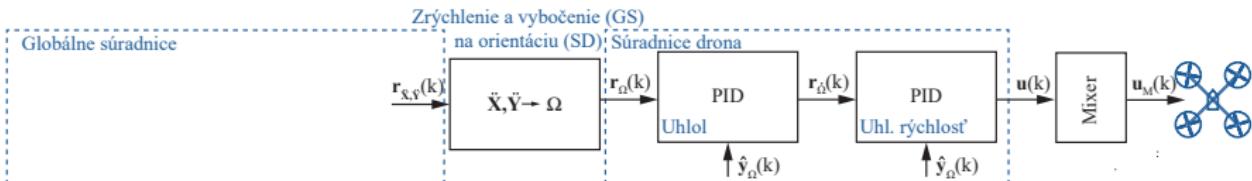
- Úlohou je držať polohu na základe globálnych súradníc (GS) napr. GPS
- Klopenie a klonenie určíme na základe GS súradníc, t.j. premeníme $X, Y \rightarrow \Theta, \Phi\dots$
- ... ale potrebujeme na výpočet aj želané/aktuálne vybočenie, t.j. $X, Y, \Psi \rightarrow \Theta, \Phi,$



- Úlohou je držať polohu na základe globálnych súradníc (GS) napr. GPS
- Klopenie a klonenie určíme na základe GS súradníc, t.j. premeníme $X, Y \rightarrow \Theta, \Phi\dots$
- ... ale potrebujeme na výpočet aj želané/aktuálne vybočenie, t.j. $X, Y, \Psi \rightarrow \Theta, \Phi,$

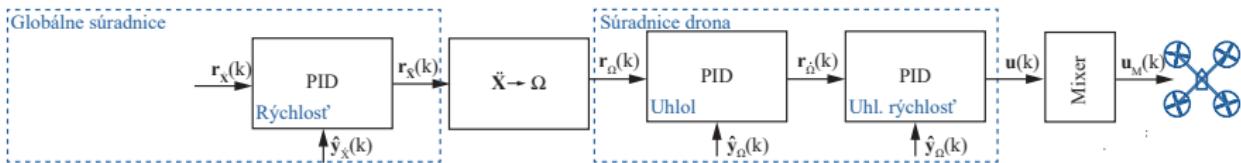


• aaaa



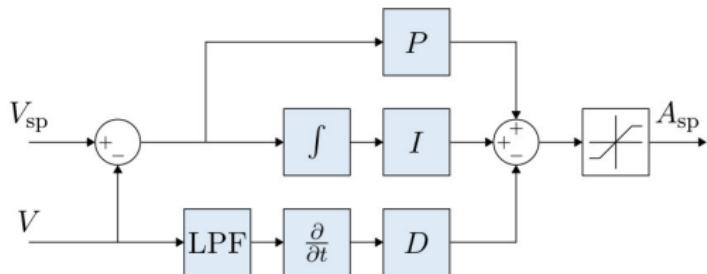
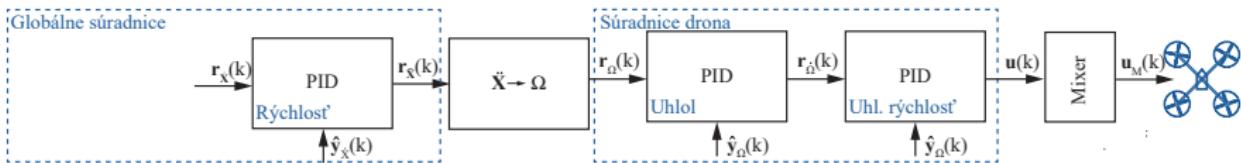


- Väčšinou je to PID riadenie, ako aj pri ArduCopter, PX4 a inde (c.f. Saha 2020).

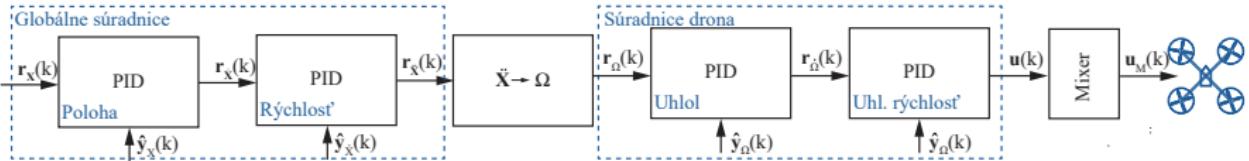




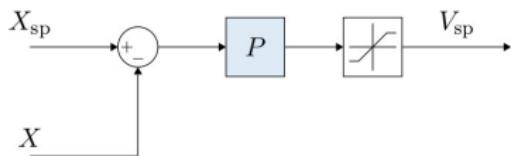
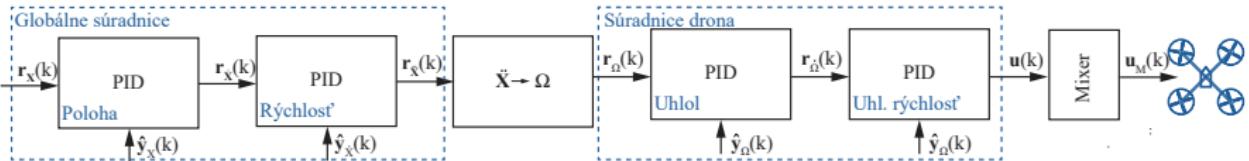
- Väčšinou je to PID riadenie, ako aj pri ArduCopter, PX4 a inde (c.f. Saha 2020).



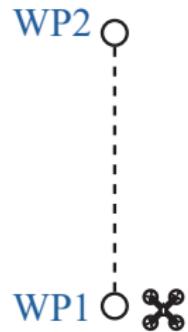
- Väčšinou je to P riadenie, ako aj pri ArduCopter, PX4 a inde (c.f. Saha 2020).



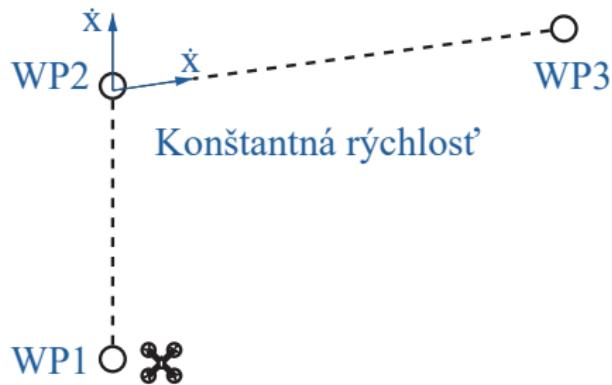
- Väčšinou je to P riadenie, ako aj pri ArduCopter, PX4 a inde (c.f. Saha 2020).



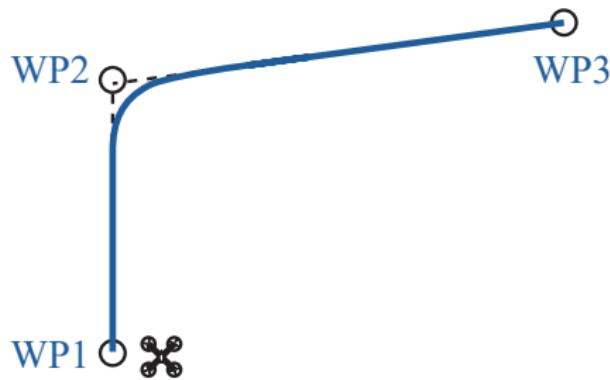
- Majme trasu WP1 do WP2, všetko je v poriadku.
- Pridajme WP3 a rozmýšľajme čo sa deje pri WP2. Je možné preletieť nad WP2?
- Bud' musíme úplne sa zastaviť alebo nemôžeme priamo preletieť — ináč by sme potrebovali nekonečne veľké zrýchlenia



- 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



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] ArduPilot. *Copter Attitude Control*. Online. [cited 29.11.2021]; Available from <https://ardupilot.org/dev/docs/apmcopter-programming-attitude-control-2.html>. 2021.
- [2] Ryan Boland. *Embedded Programming for Quadcopters*. Online. [cited 2.12.2021]; Available from <https://www.youtube.com/watch?v=CHSYgLfhwUo&t=3s>. 2015.
- [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. 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] **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.
- [8] **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.

- [9] Dola Saha. *Precise Unmanned Aerial Vehicle (UAV) Flight Control*. Online. 2020 Fall ECE553 Resources [cited 3.12.2021]; Available from https://www.albany.edu/faculty/dsaha/teach/2020Fall_ECE553/. 2020.