Lelkce 4

Cíl lekce

Naučit robota jezdit po připravené pásce.

Úvodní nastavení, která se nám budou hodit

- `vision_ctrl.enable_detection(rm_define.vision_detection_line)` - zapne detekci čar
- `vision_ctrl.line_follow_color_set(rm_define.line_follow_color_blue)` - nastaví robota, aby sledoval modrou čáru
- `media_ctrl.exposure_value_update(rm_define.exposure_value_large)` - nastaví senzor robota tak, aby lépe rozpoznával čáru na zemi
- `gimbal_ctrl.pitch_ctrl(-15)` - nastaví hlavu robota, aby se díval dolů - lépe uvidí na čáru
- `robot_ctrl.set_mode(rm_define.robot_mode_chassis_follow)` - tohle už známe, nastaví aby tělo následovalo v otáčení hlavu

Teorie řízení - PID controller (proportional–integral–derivative controller)

Tvůrci robota pro nás připravili funkcionalitu zvanou `PIDCtrl()`, která nám sama vypočte, jak robota řídit, aby jezdil po čáře (nebo zaměřoval objekt). Tato funkce má tři koeficienty, které dopředu nastavíme a ona už se sama postará o správný výpočet na zákdladě dodaného vizuálního vstupu.

Inicializace funkce řízení

```
pid=rm_ctrl.PIDCtrl()
pid.set_ctrl_params(115, 0, 5) # tyto parametry můžete měnit
```

Jako příklad použijte tyto nastavené parametry, můžete je ovšem zkusit měnit a sledovat jak se mění chování robota.

Využívání výpočtu PID controlleru

Když už máme controller nastavený, můžeme do něj dávat vstupy z kamery. Výstup rozpoznání kamery získáme zavoláním funkce `vision_ctrl.get_line_detection_info()`.
Pokud nám tato funkce vrátí list (pole) delší než 2 elementy, víme že se rozpoznání čáry povedlo a můžeme si vzít parametr, který nás pro řízení robota po čáře zajímá.

```
line = vision_ctrl.get_line_detection_info()

if len(line) > 2:
    # nějakou čáru jsme rozpoznali
    print(line)
else:
    print("caru nevidim :(")
```

Popis výstupu funkce `get_line_detection_info()` nalezneme v [dokumentaci (https://www.dji.com/cz/robomaster-s1/programming-guide)], záložka 17 `identified line info`.

Nyní můžeme zavolat funkci pro výpočet správného natočení hlavy robota a funkci na otáčení hlavy:

```
pid.set_error(RmList(line)[19] - 0.5)
gimbal_ctrl.rotate_with_speed(pid.get_output(),0)
```

Co je na těchto dvou řádkách děje?

- `RmList` - robomaster konstruktor na list (tento list se pak zobrazí v debug info)
- `RmList(line)[19]` - vezmeme 20. prvek listu (pole). Na kterém je uložena informace o úhlu natočení robota vůči čáře
- `- 0.5` - od výsledku odečteme konstantu 0.5, čímž dostaneme čáru do centra obrazu
- `pid.set_error()` - výsledek výpočtu dáme do této funkce. Tato funkce bere naměřenou odchylku (chybu), na jejímž základě vypočte správný úhel natočení hlavy
- `pid.get_output()` - tato funkce vypočte správné natočení hlavy na základě naměřené odchylky (chyby), minulým chybám a koeficientech nastavení (viz výše)

Co je to vlastně PID controller /můžete přeskočit/

Zjednodušeně se jedná o vzorec (algoritmus), který nám pomůže regulovat systém (např. otáčet věž robota), tak abychom minimalizovali odchylku.
Je složený ze tří částí (proporcionální, integrační a derivační). Nenechte se zmást neznámými výrazy, nejedná se o nic složitého.

- proporcionální část je naměřená chyba pronásobená koeficientem K_p
- integrační část je *plocha* dosud naměřené chyby vynásobená koeficientem K_i
- derivační část je tečna (směrnice) chyby vynásobená koeficientem K_d

Co je to plocha a tečna chyby si můžeme společně vysvětlit na flipchartu.

Koeficienty K_p, K_i, K_d si můžete zvolit tak, aby regulátor dával vhodné výsledky pro vaši úholu.

[Vizualizace chyby v závislosti na zvolených parametrech (https://en.wikipedia.org/wiki/PID_controller#Controller_theory)]

Úkol 1

1. Napište cyklus ve kterém robot bude rozpoznávat čáru. Výsledky rozpoznávání vypište a dívejte se, které parametry se mění a jak.
2. S pomocí PID controlleru (viz výše) vypočtěte úhel, jak se má robot natáčet vůči čáře
3. Robota správně natočte a v cyklu jeďte blíže k čáře a nebo přímo po čáře. (pokud bude potřeba počkejte na výsledek natočení funkcí `time.sleep()`)

Úkol 2

- Rozšiřte první úkol tak, aby robot rozpoznával křižovatky (typu "T") a na křižovatce se vydal cestou doleva
- Když rozpoznáte křižovatku můžete na ni dojet (odhad vzdálenosti je na třetím prvku listu rozpoznaného objektu) a pak se otočit na požadovanou stranu

Jak poznat křižovatku?

Pokud funkce na rozpoznání čáry vrátí na druhém indexu číslo dva, jedná se o křižovatku typu "T" (tj. z jednoho bodu jsou 3 cesty, ale jednou jsme již přijeli)

Úkol 3 (bonus)

- Rozšiřte předchozí úkol tak, aby robot když stojí na křižovatce našel marker (můžete si to zjednodušit, že marker bude rovnou vidět a robot se nebude muset nijak otáčet).
  Pokud je na markeru liché číslo, pojede robot doleva, jinak doprava.