TECHNICKÁ UNIVERZITA V KOŠICIACH
FAKULTA ELEKTROTECHNIKY A INFORMATIKY

Riešenie úlohy súčasnej lokalizácie a mapovania objektov v prostredí systému ROS

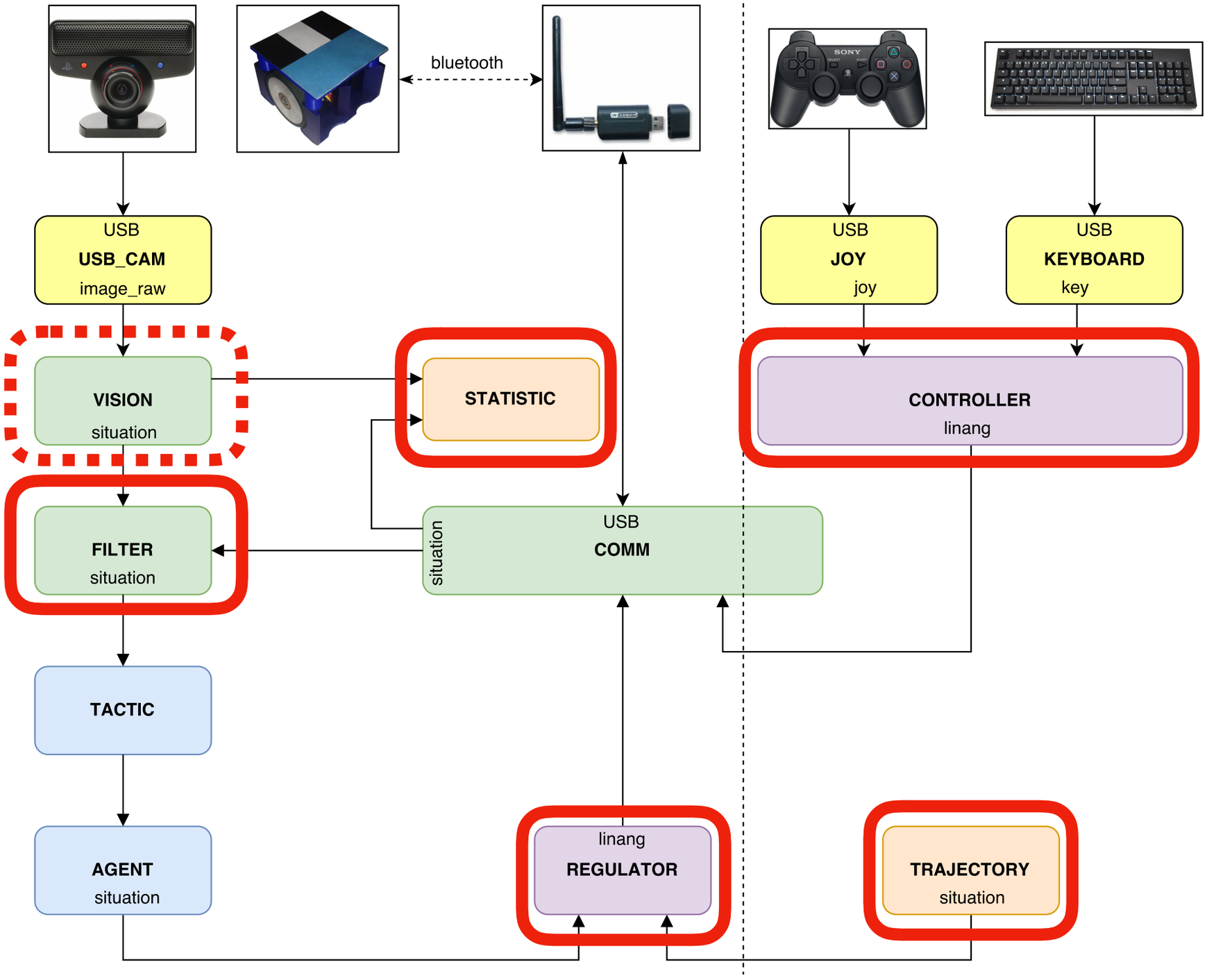
Bakalárska práca

Vedúci práce: doc. Ing. Ján Jadlovský, CSc.
Konzultant: Ing. Michal Kopčík

Adam Březina
3. ročník Bc. štúdia, Inteligentné Systémy

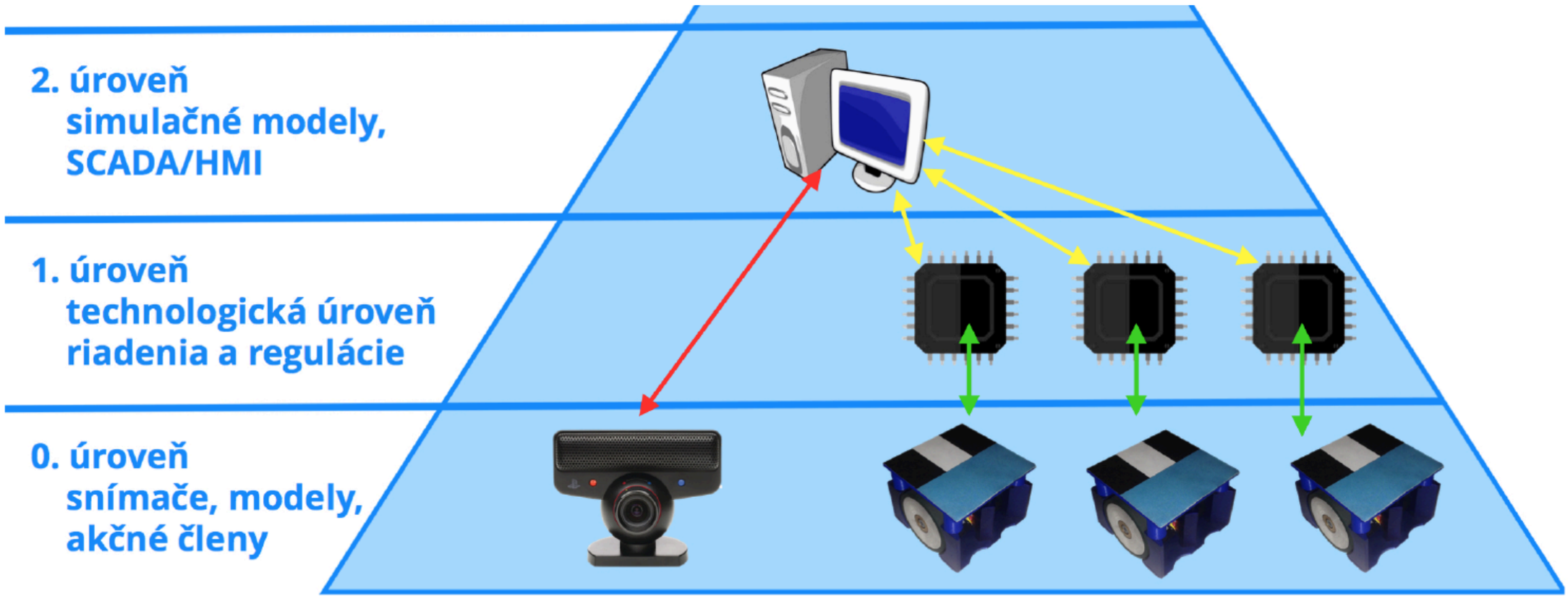
Obsah

1 Návrh infraštruktúry

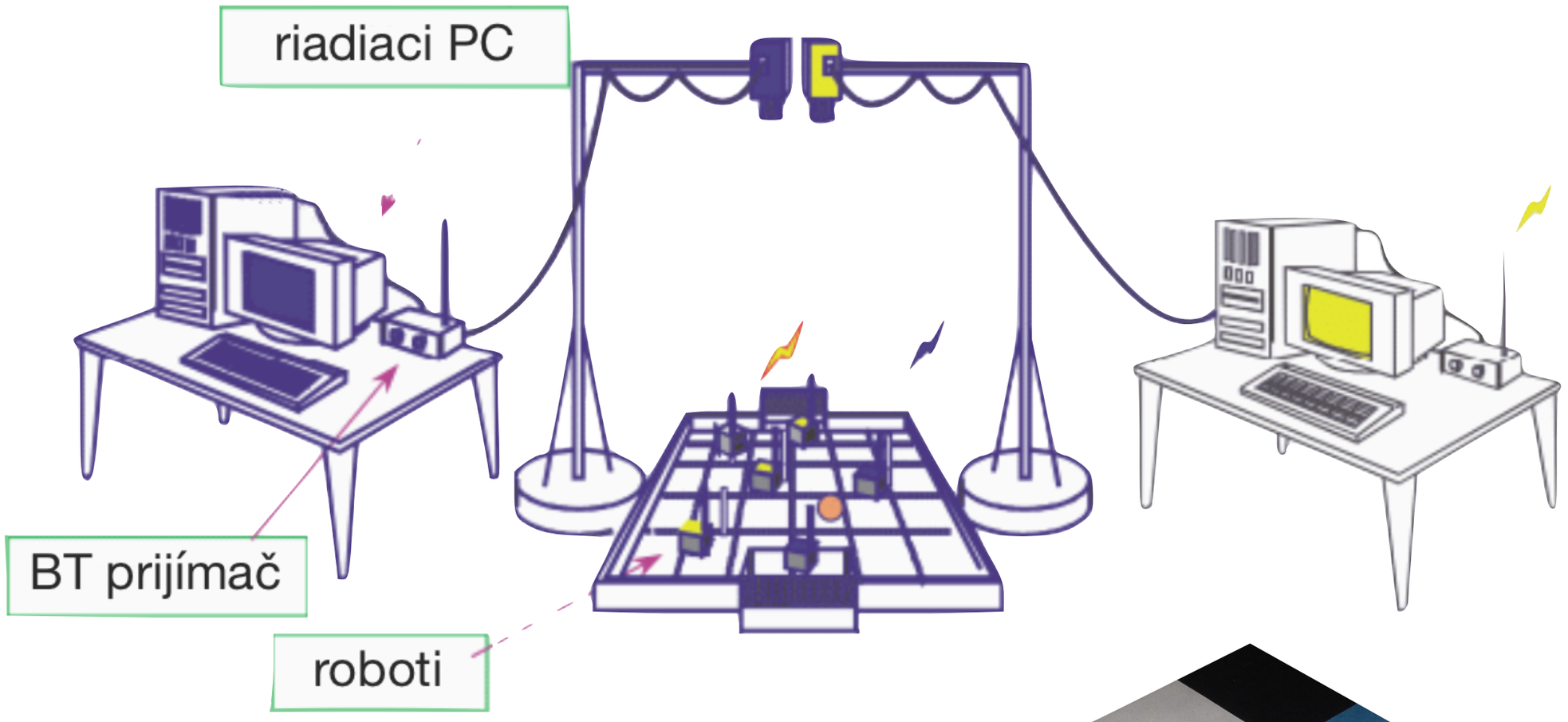


2 Popis robotického futbalu

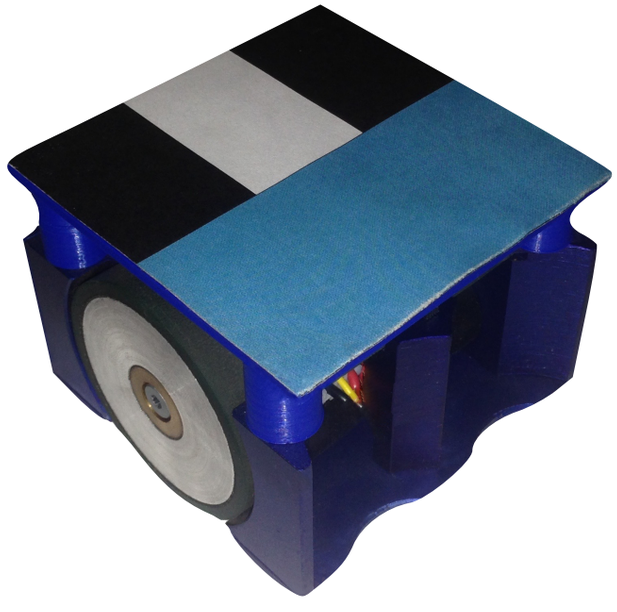
Robotický futbal v rámci pyramídy DSR



Náčrt MiRoSoT zápasu

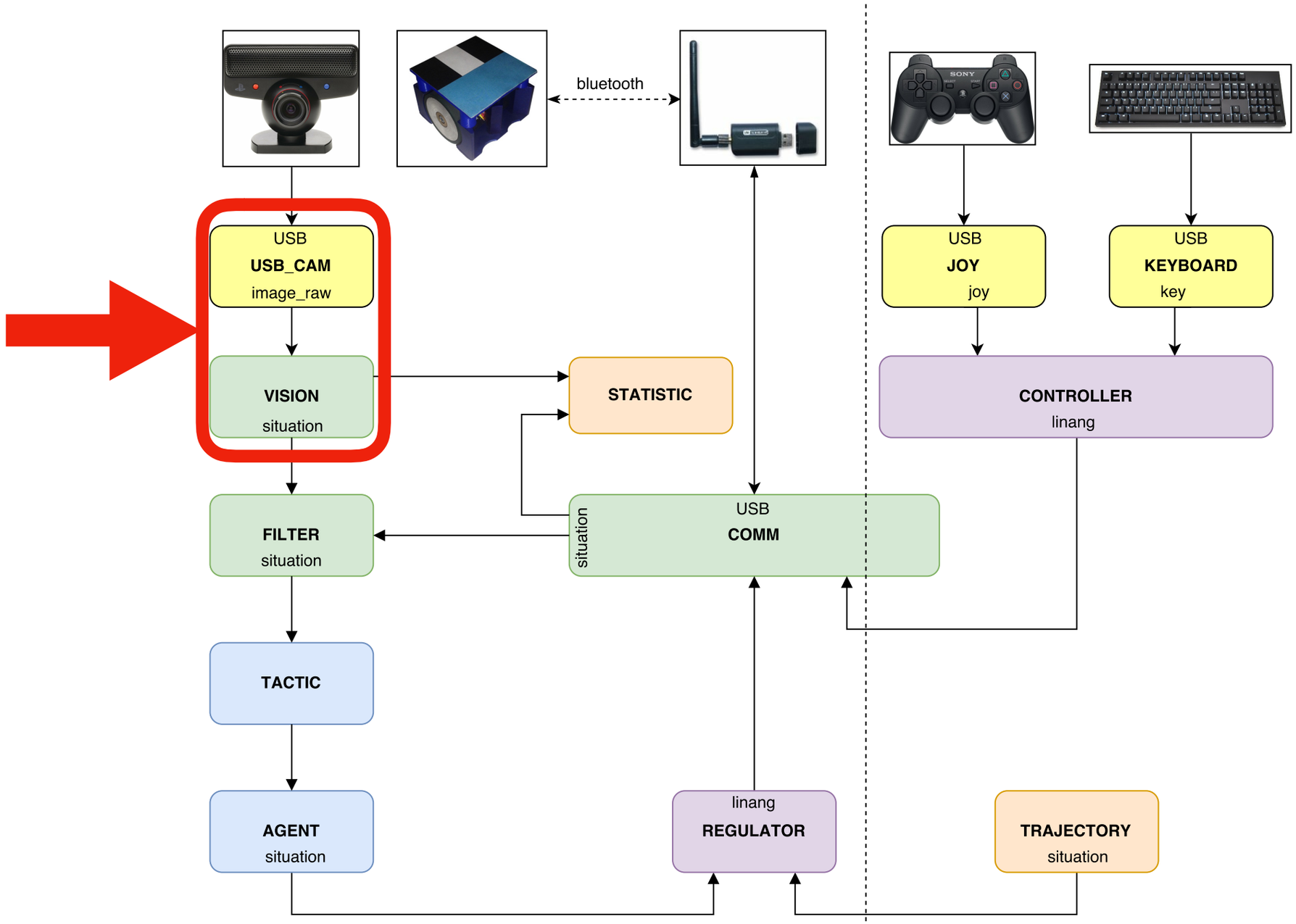


Náš MiRoSoT futbalista



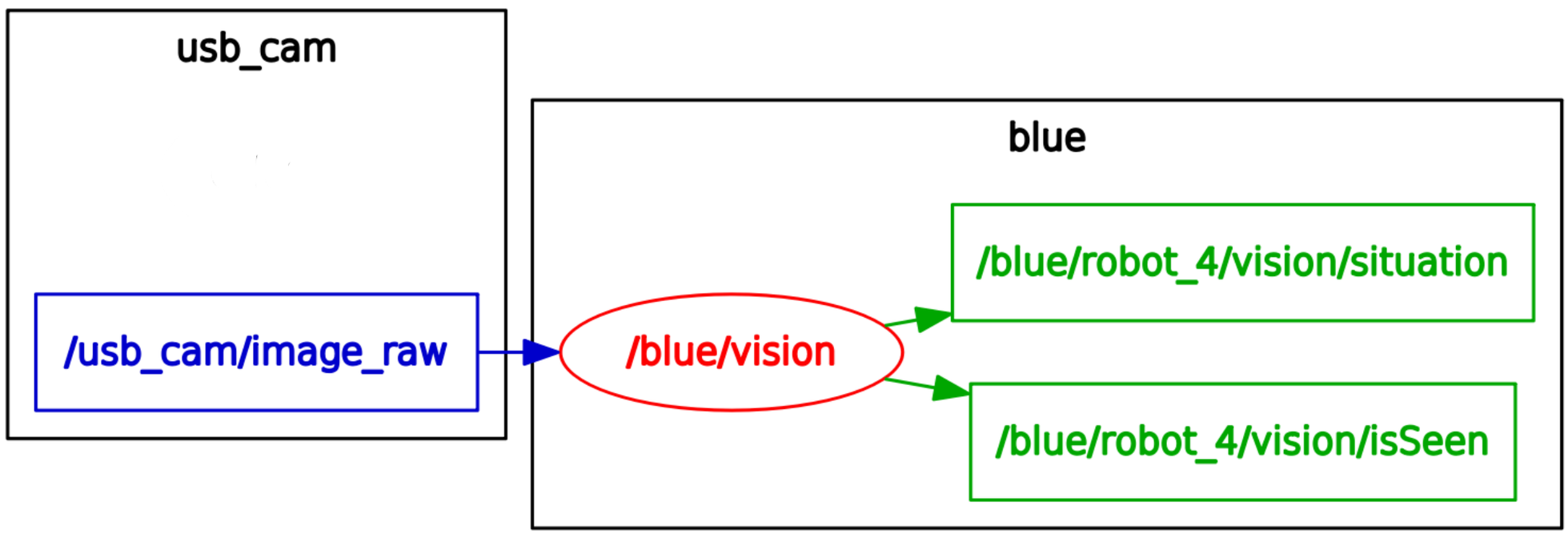
3 Rozpoznávanie obrazu

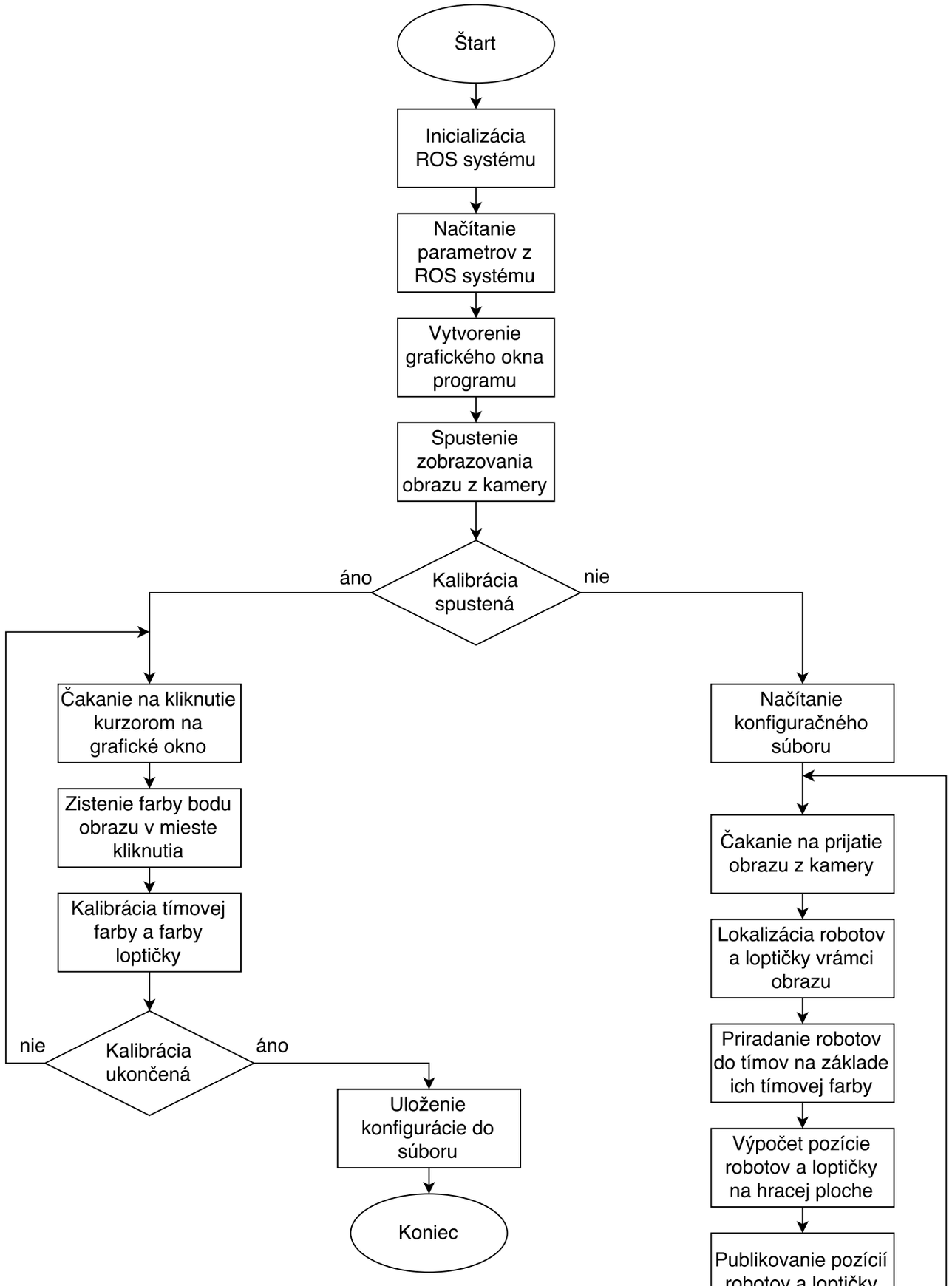
Základné informácie



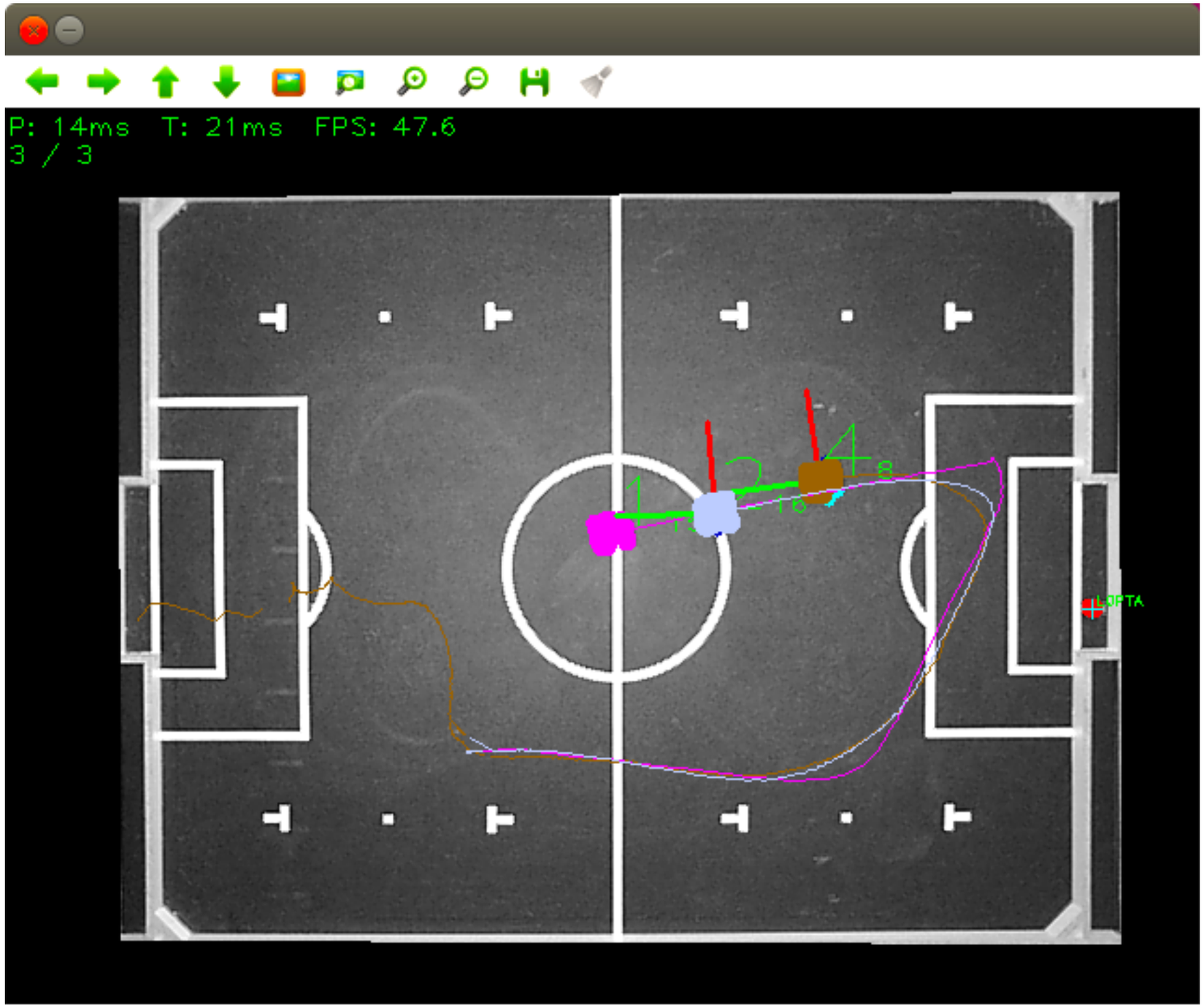
Implementácia modulu Vision

- Rozpoznávania obrazu Ing. M. Vargu
- Vstup:
  - obraz z kamery - **/usb_cam/image_raw**
- Výstup:
  - pozícia robota - **/blue/robot_x/vision/situation**
  - spozorovanie robota - **/blue/robot_x/vision/isSeen**




Štart
Inicializácia ROS systému
Načítanie parametrov z ROS systému
Vytvorenie grafického okna programu
Spustenie zobrazovania obrazu z kamery
Kalibrácia spustená
áno
nie
Čakanie na kliknutie kurzorom na grafické okno
Zistenie farby bodu obrazu v mieste kliknutia
Kalibrácia tímovej farby a farby loptičky
Kalibrácia ukončená
nie
áno
Uloženie konfigurácie do súboru
Koniec
Načítanie konfiguračného súboru
Čakanie na prijatie obrazu z kamery
Lokalizácia robotov a loptičky vrámci obrazu
Priradanie robotov do tímov na základe ich tímovej farby
Výpočet pozície robotov a loptičky na hracej ploche
Publikovanie pozícií robotov a loptičky

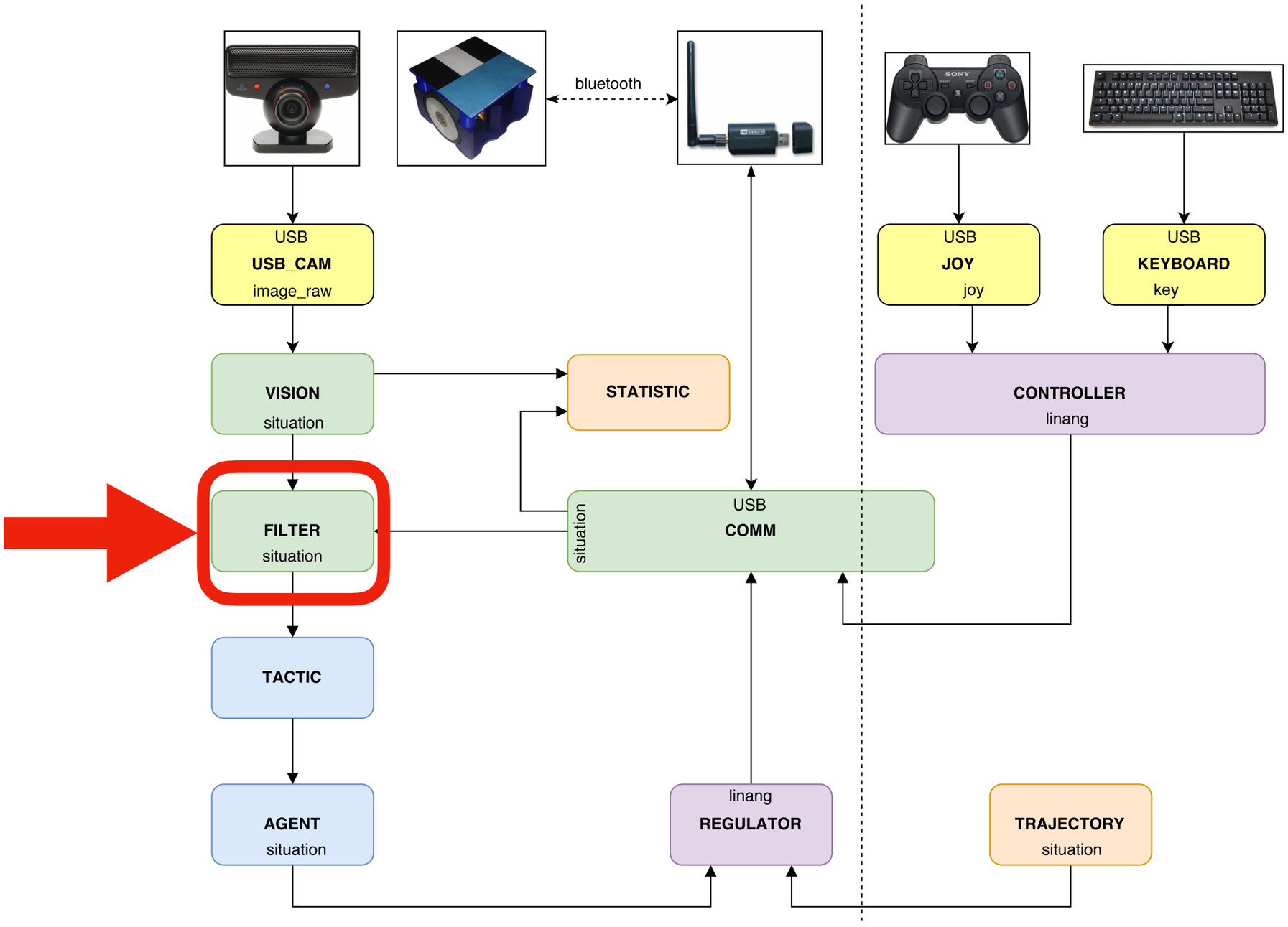
Výstup z rozpoznávania obrazu



(x=414, y=327) ~ R:77 G:77 B:77

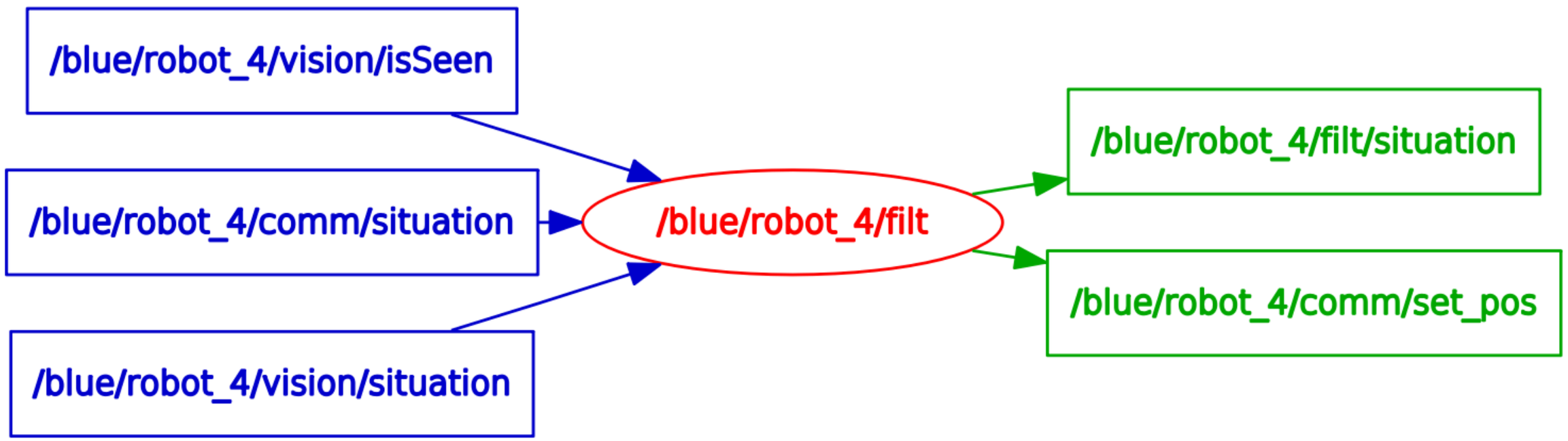
Filtrovanie údajov

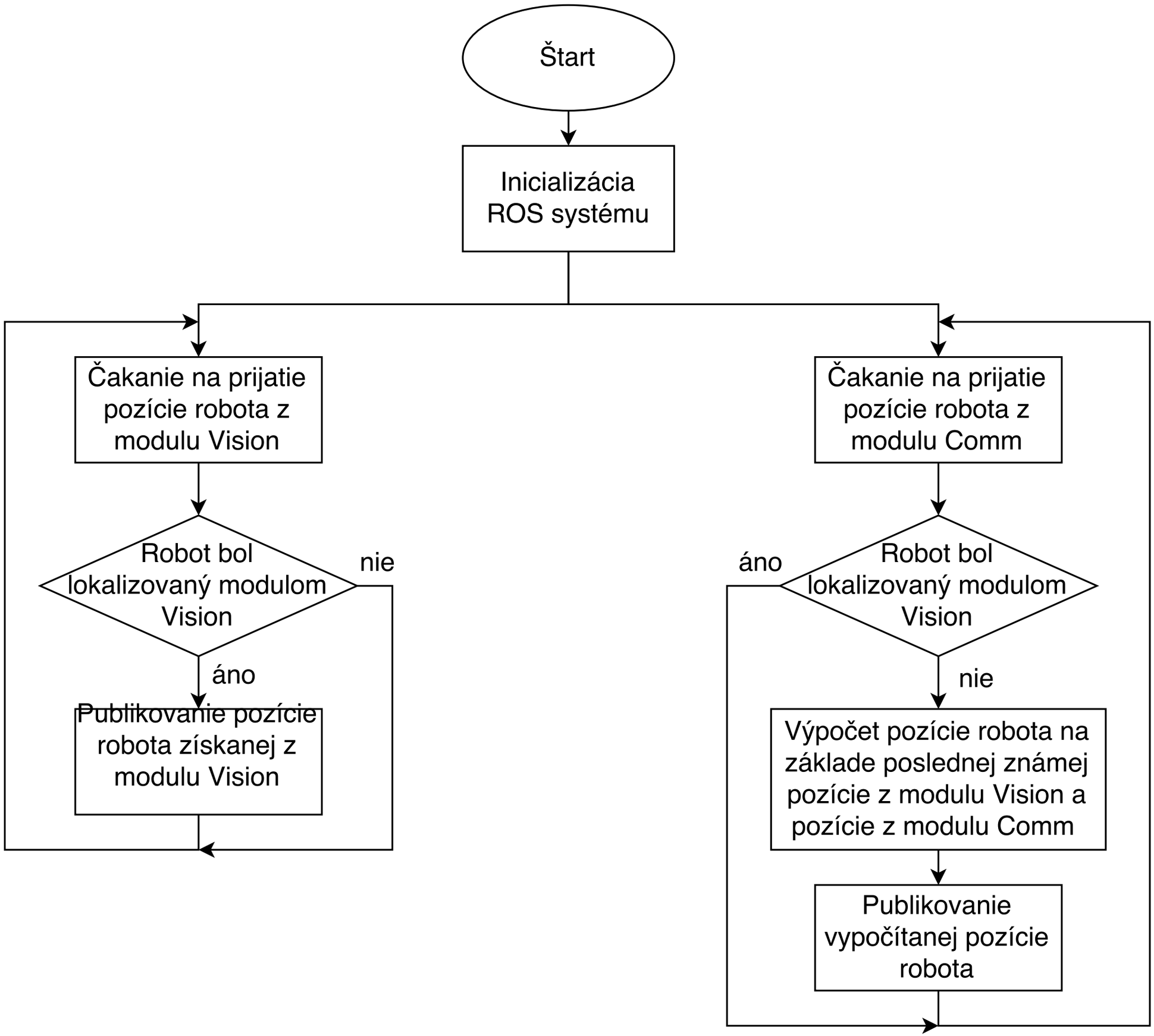
Základné informácie



Implementácia modulu **Filter**

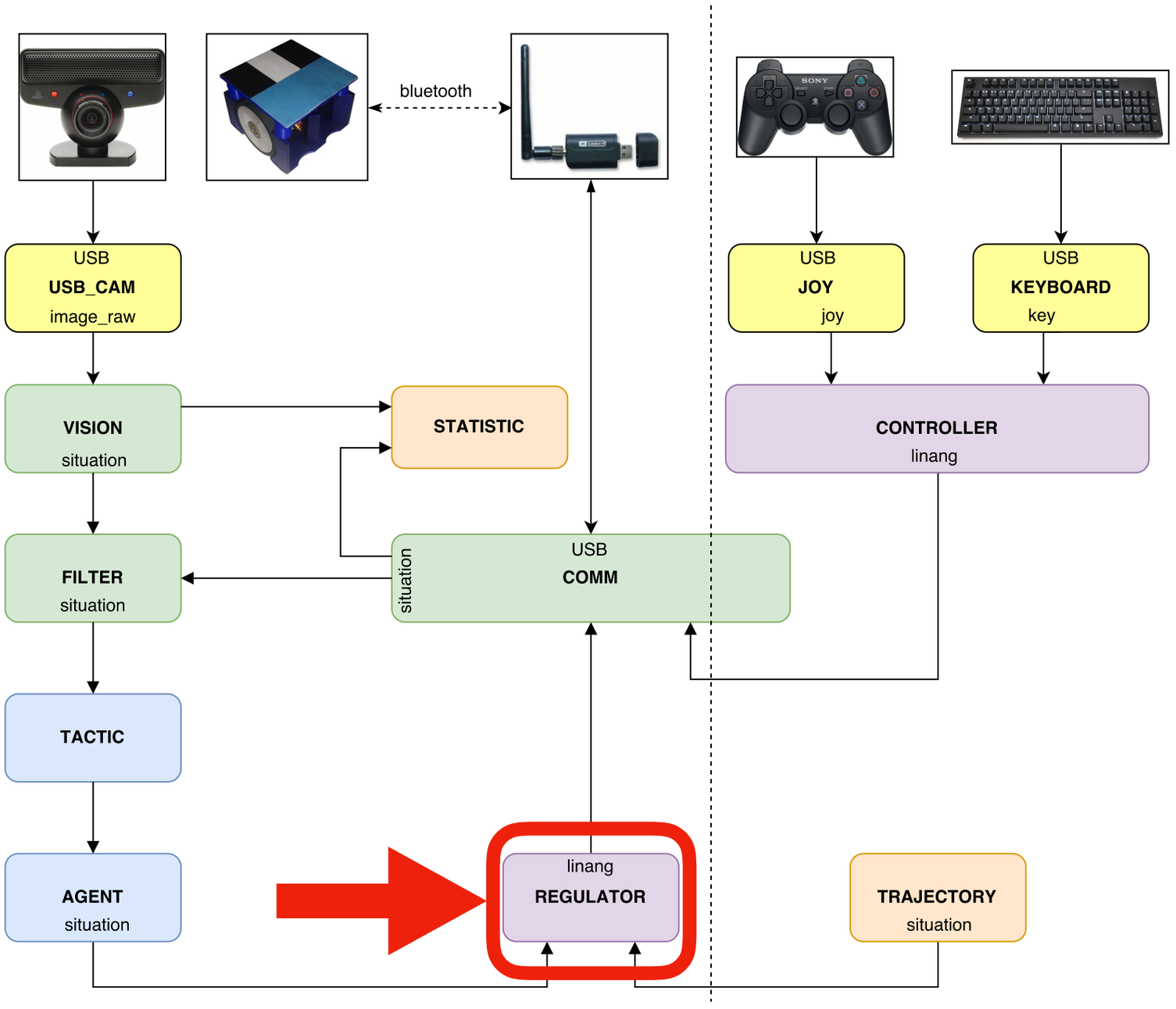
- Filtrovanie dostupných údajov
- Vstup:
  - pozícia robota - **/blue/robot_x/vision/situation**
  - spozorovanie robota - **/blue/robot_x/vision/isSeen**
  - odometria robota - **/blue/robot_x/comm/situation**
- Výstup:
  - filtrovaná pozícia - **/blue/robot_x/filt/situation**
  - zmena polohy v robotovi - **/blue/robot_x/comm/set_pos**



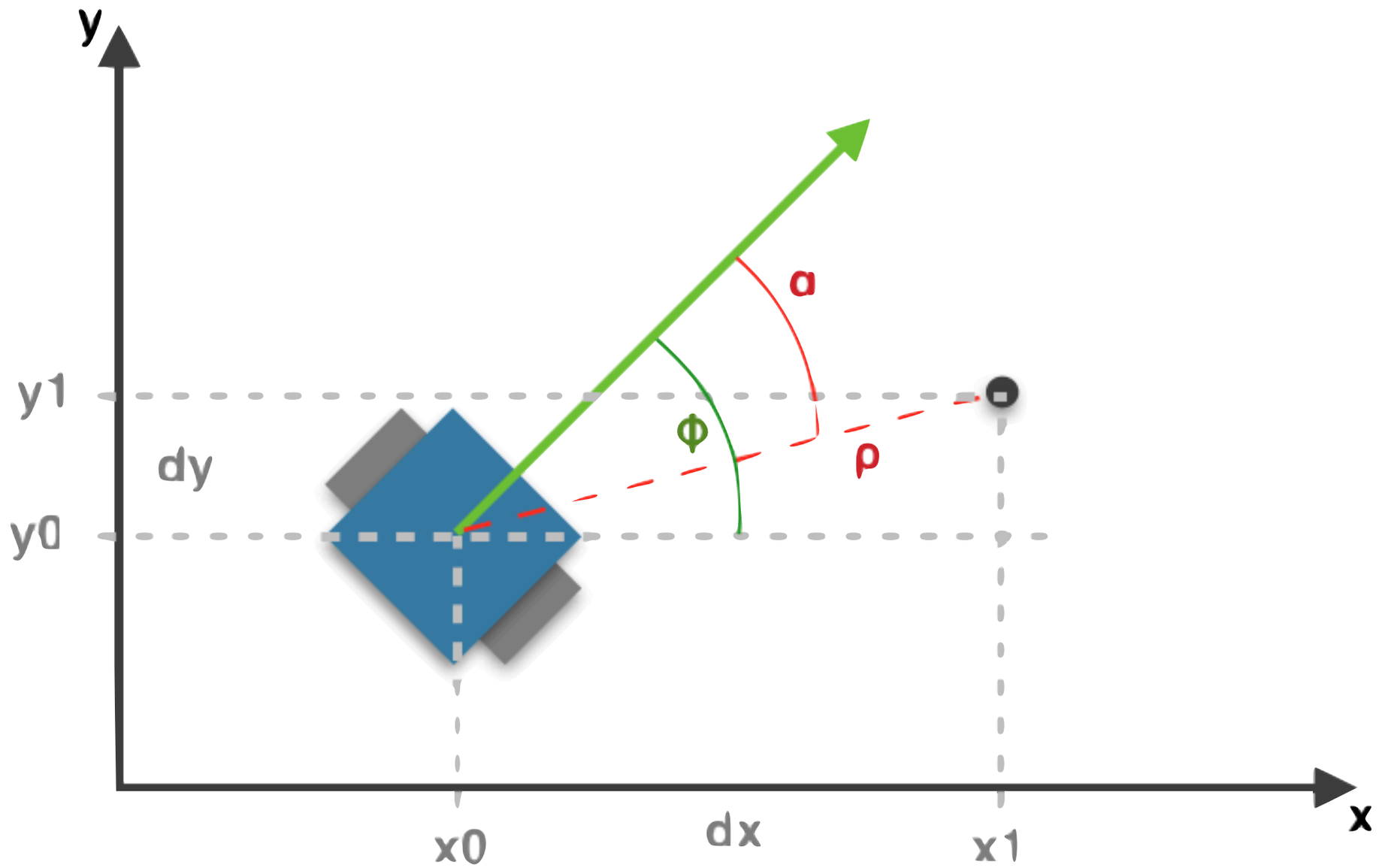

Štart
Inicializácia ROS systému
Čakanie na prijatie pozície robota z modulu Vision
Robot bol lokalizovaný modulom Vision
nie
áno
Publikovanie pozície robota získanej z modulu Vision
Čakanie na prijatie pozície robota z modulu Comm
Robot bol lokalizovaný modulom Vision
áno
nie
Výpočet pozície robota na základe poslednej známej pozície z modulu Vision a pozície z modulu Comm
Publikovanie vypočítanej pozície robota

4 Riadenie robotov

Základné informácie



Znázornenie robota v priestore

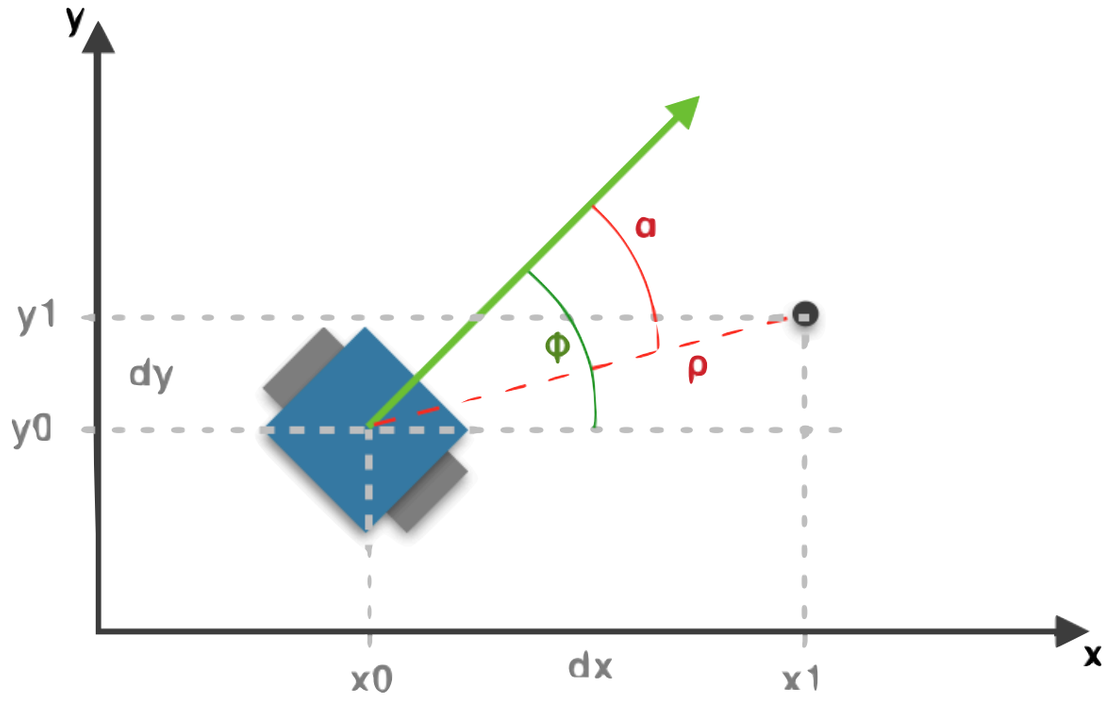


Výpočet výstupu regulátora

Výpočet chyby:

$$\rho = \sqrt{dx^2 + dy^2}$$

$$\alpha = atan2(dy, dx) - \phi_0$$

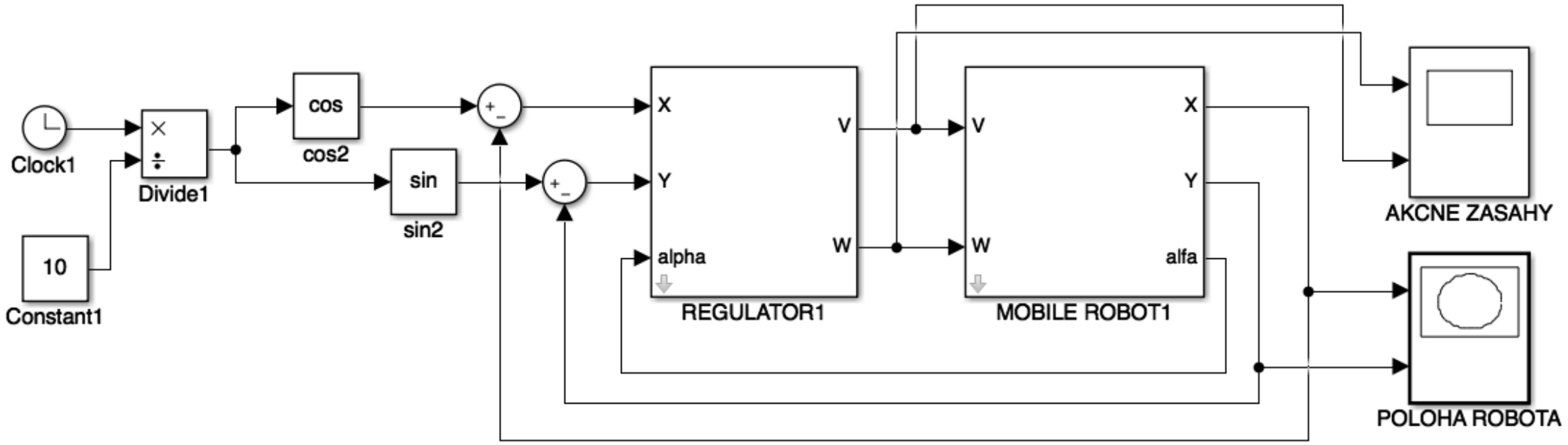


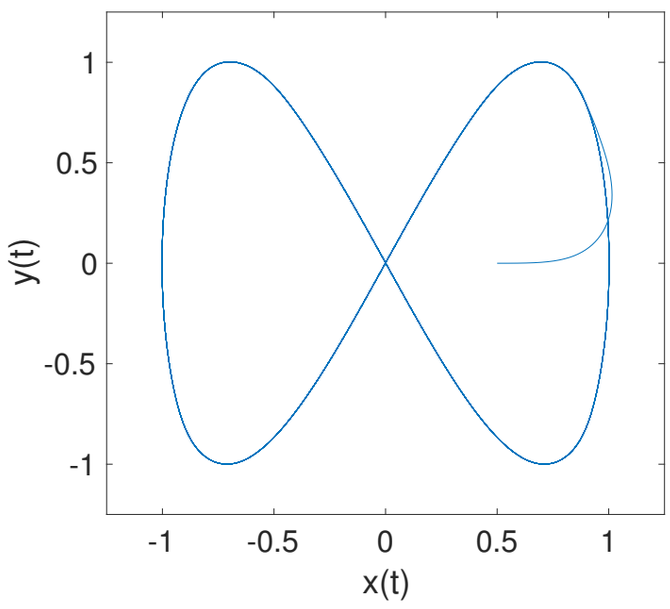
Výpočet výstupu:

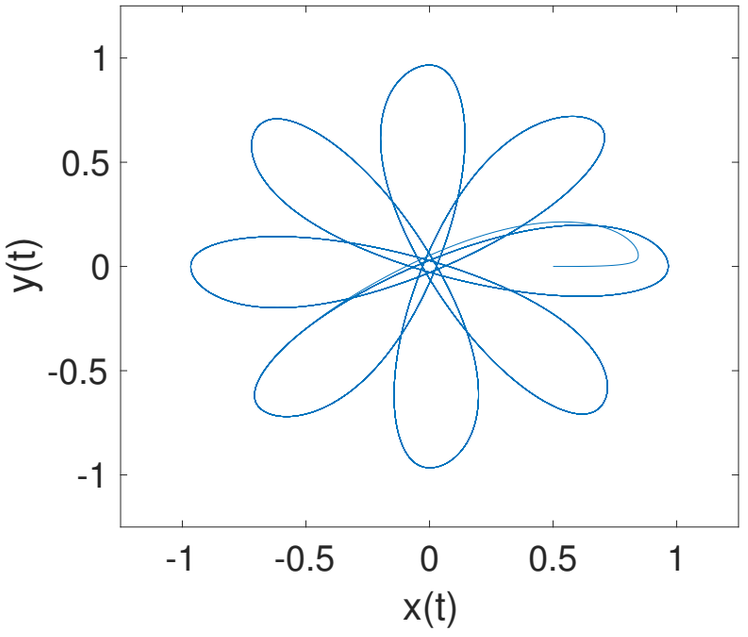
$$v_{lin} = K_\rho \rho \, |(cos(\alpha)|$$

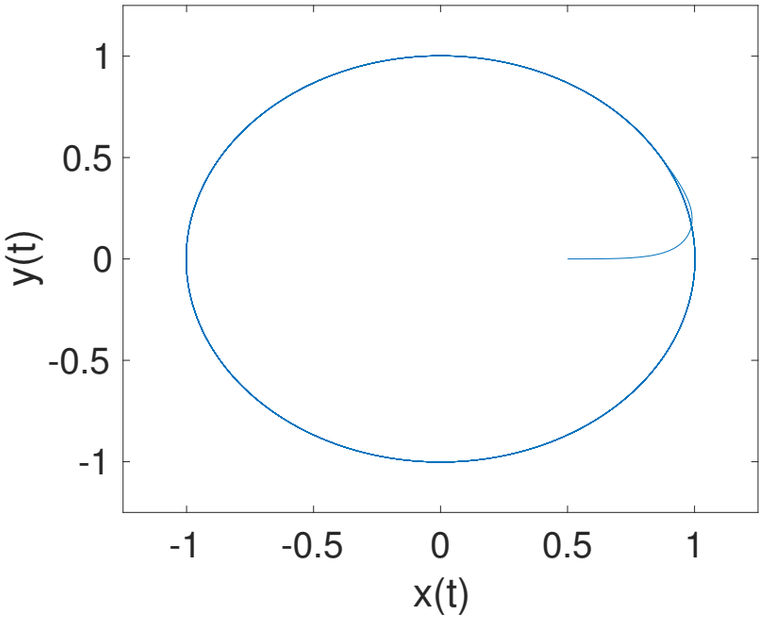
$$v_{ang} = K_\alpha \alpha$$

Simulačné overenie návrhu



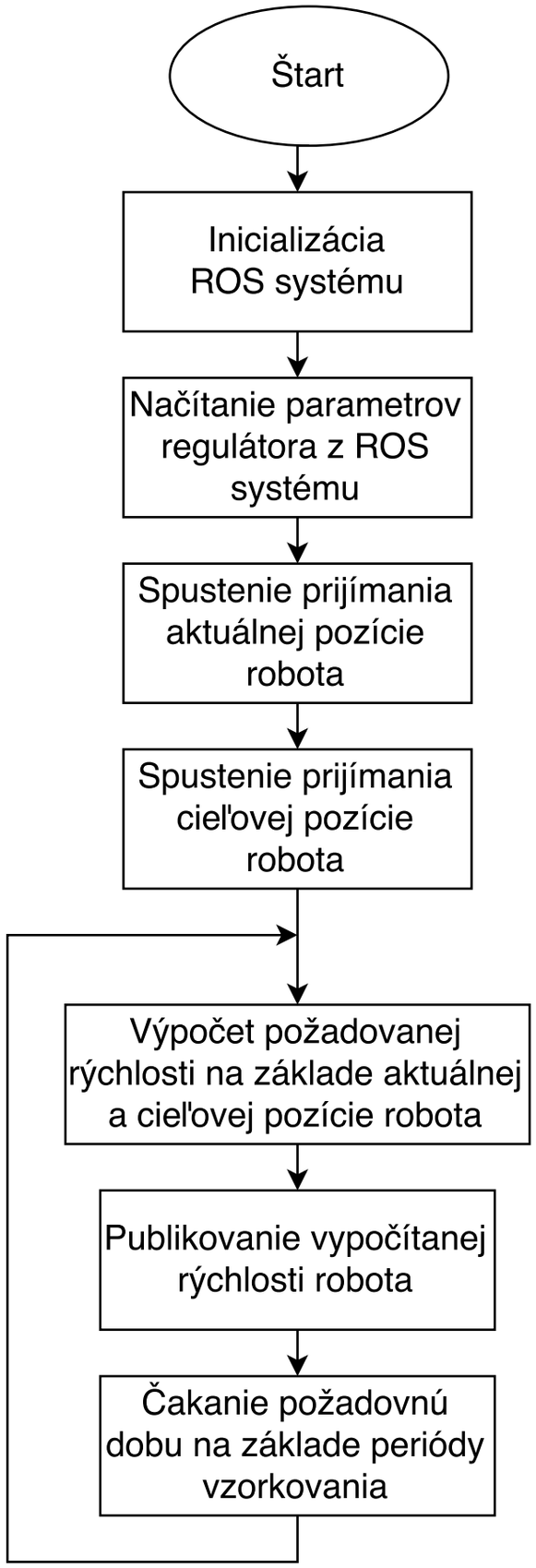




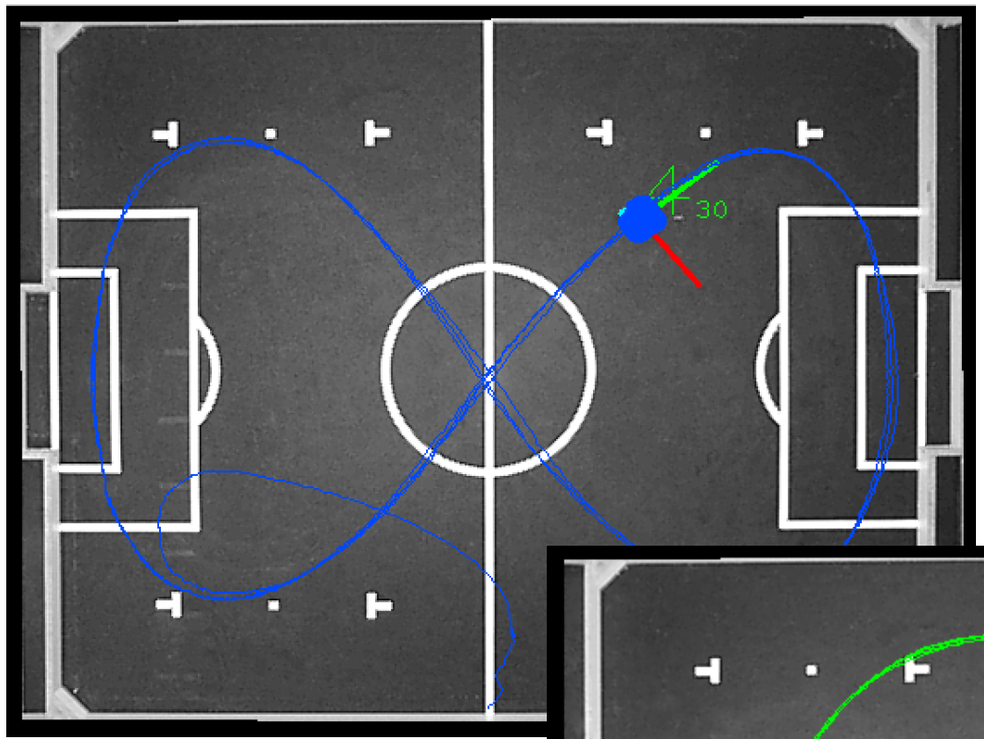


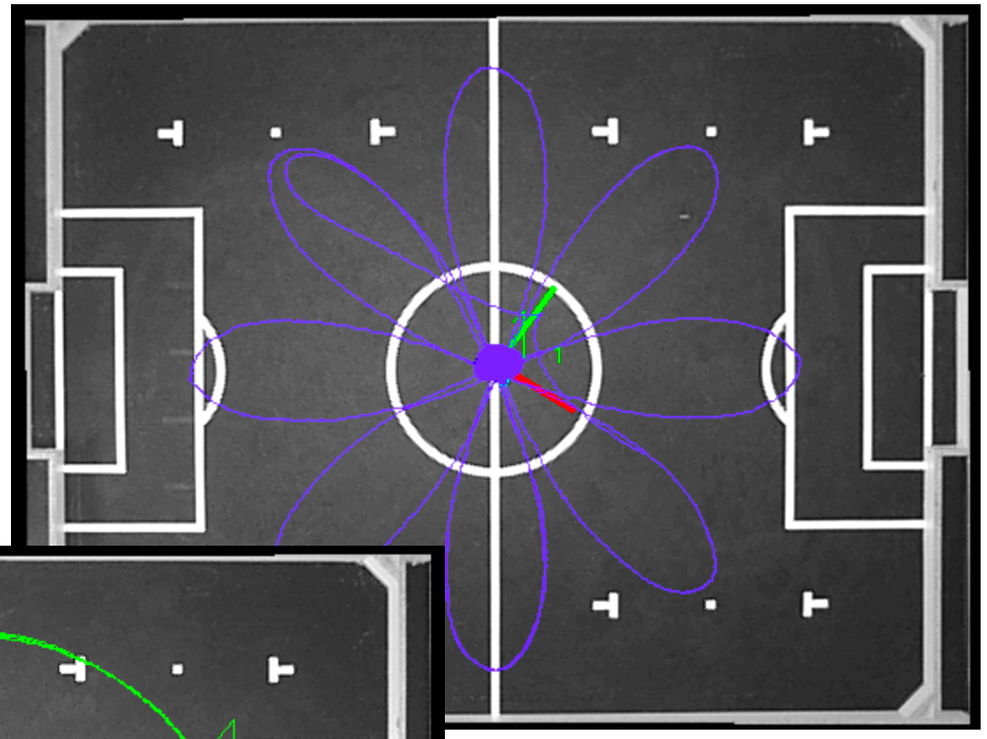
Implementácia modulu Regulator

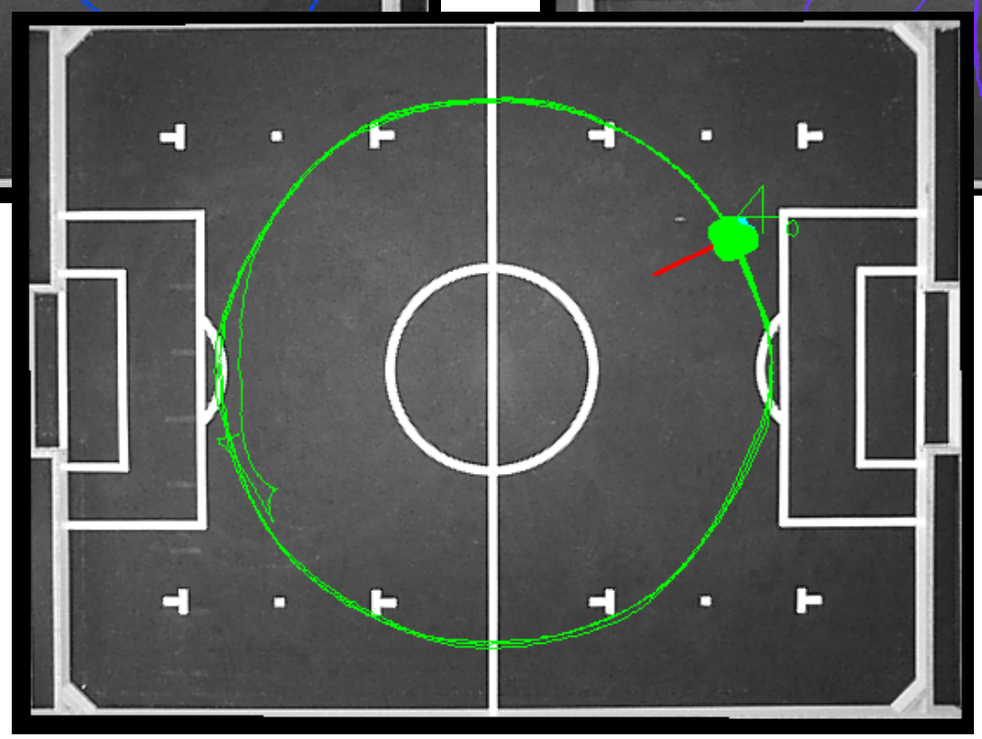
- Filtrovanie údajov z robota a kamery
- Vstup:
  - požadovaná pozícia robota - **/blue/robot_x/traj/situation**
  - filtrovaná pozícia - **/blue/robot_x/filt/situation**
- Výstup:
  - nastavovanie rýchlosti robota cez bluetooth
    - **/blue/robot_x/comm/linang_speed**


Štart
Inicializácia ROS systému
Načítanie parametrov regulátora z ROS systému
Spustenie prijímania aktuálnej pozície robota
Spustenie prijímania cieľovej pozície robota
Výpočet požadovanej rýchlosti na základe aktuálnej a cieľovej pozície robota
Publikovanie vypočítanej rýchlosti robota
Čakanie požadovnú dobu na základe periódy vzorkovania

Overenie návrhu na reálnych robotoch

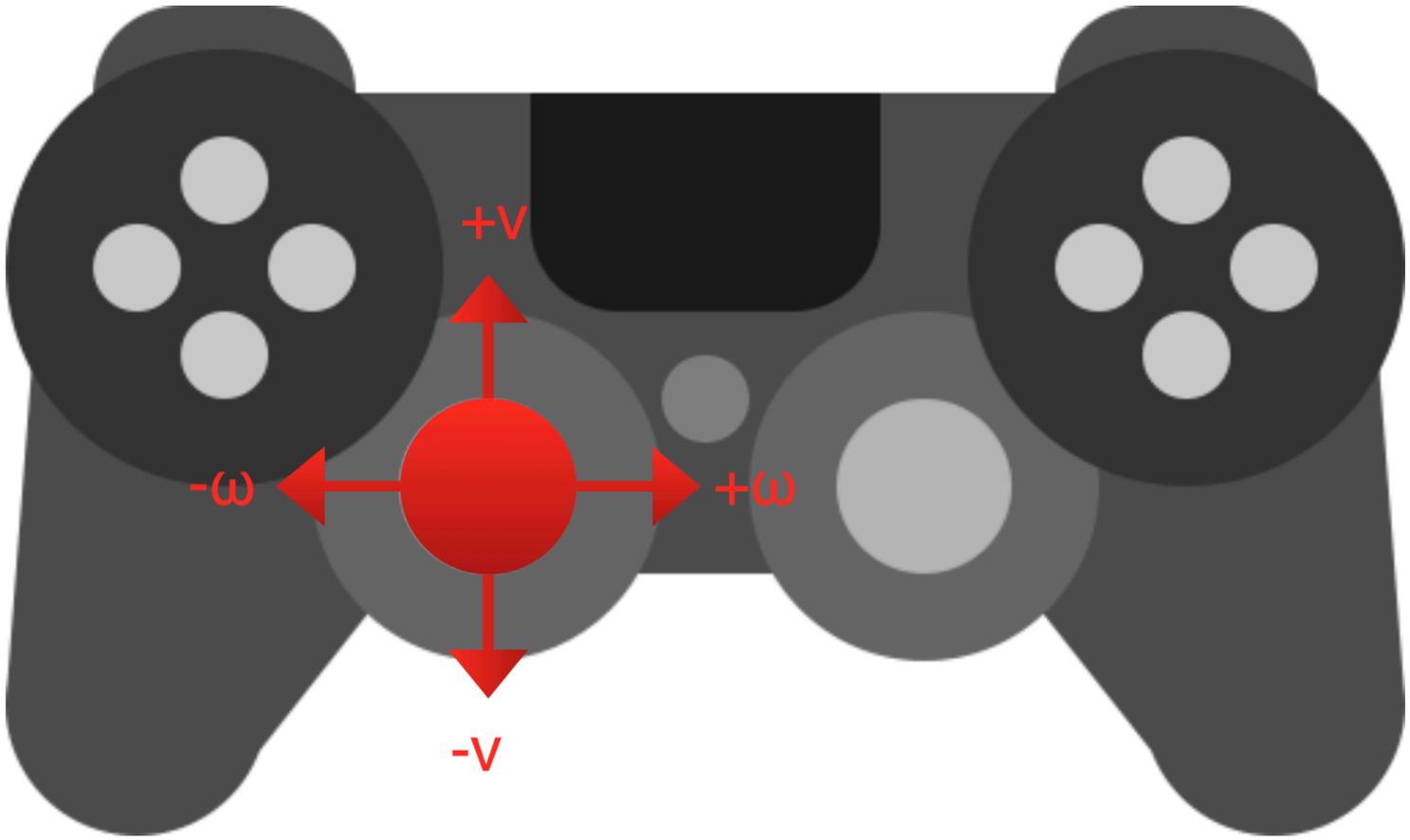






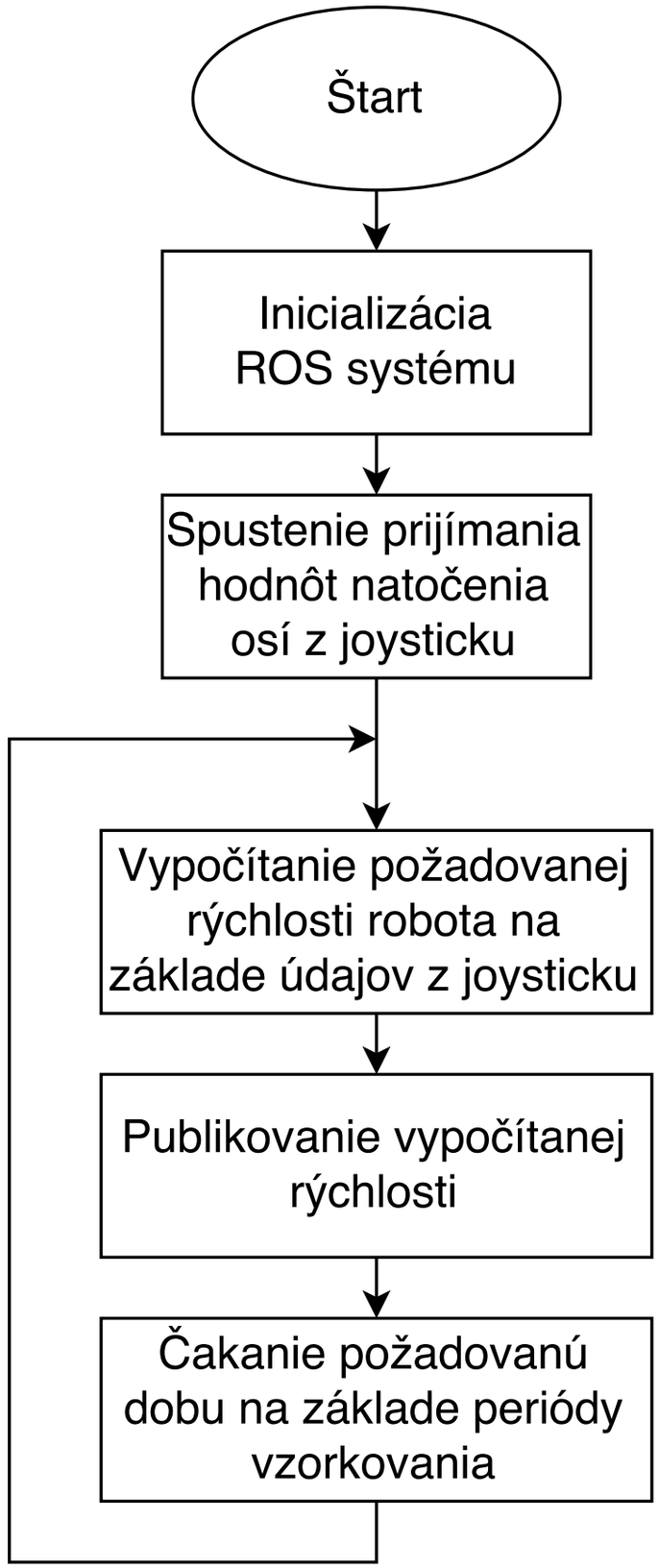
5 Pomocné moduly

Manuálne ovládanie robotov



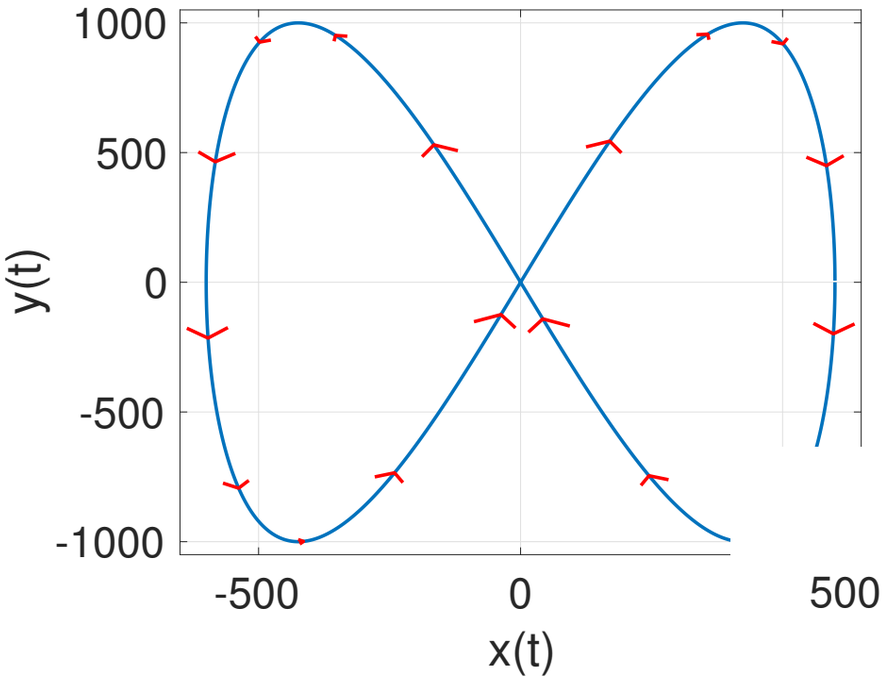
Implementácia modulu Controller

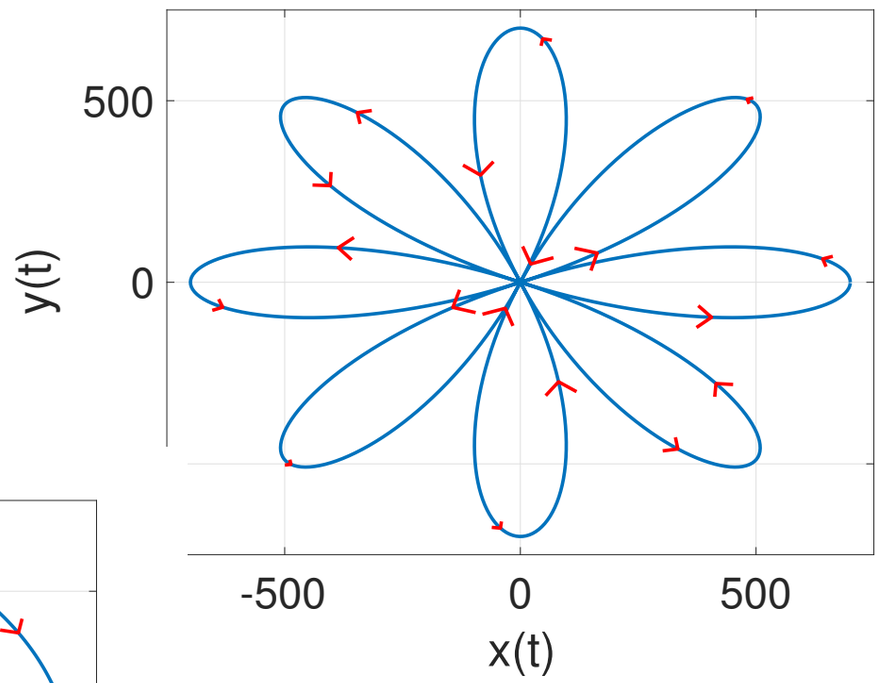
- Úprava údajov z joysticka
- Vstup:
  - hodnoty ovládaných prvkov joysticku - **/blue/robot_x/joy**
- Výstup:
  - prepočet na rýchlosti - **/blue/robot_x/comm/linang_speed**

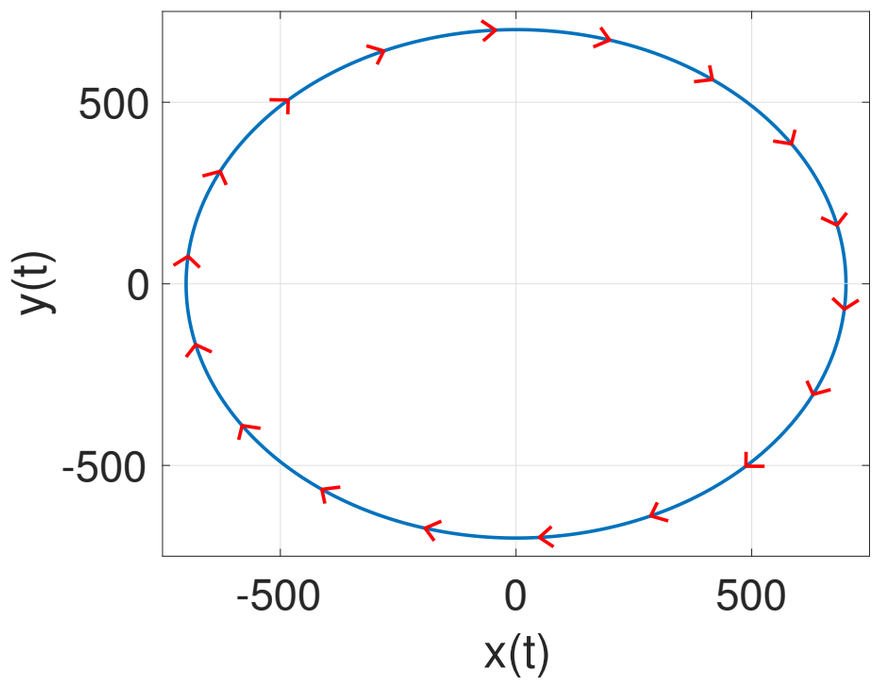

Štart
Inicializácia ROS systému
Spustenie prijímania hodnôt natočenia osí z joysticku
Vypočítanie požadovanej rýchlosti robota na základe údajov z joysticku
Publikovanie vypočítanej rýchlosti
Čakanie požadovanú dobu na základe periódy vzorkovania

Pomocné moduly

Generovanie trajektorií

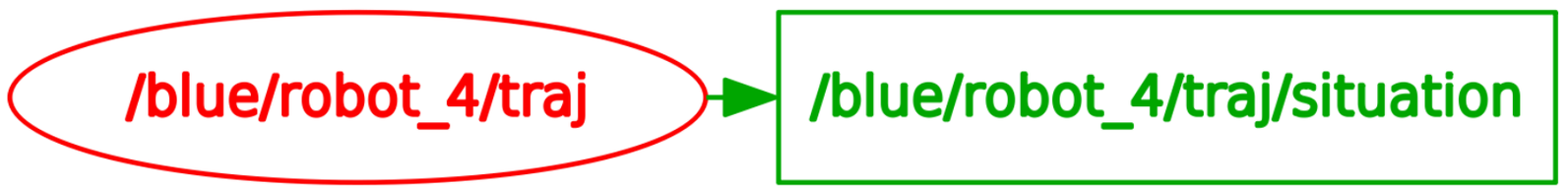


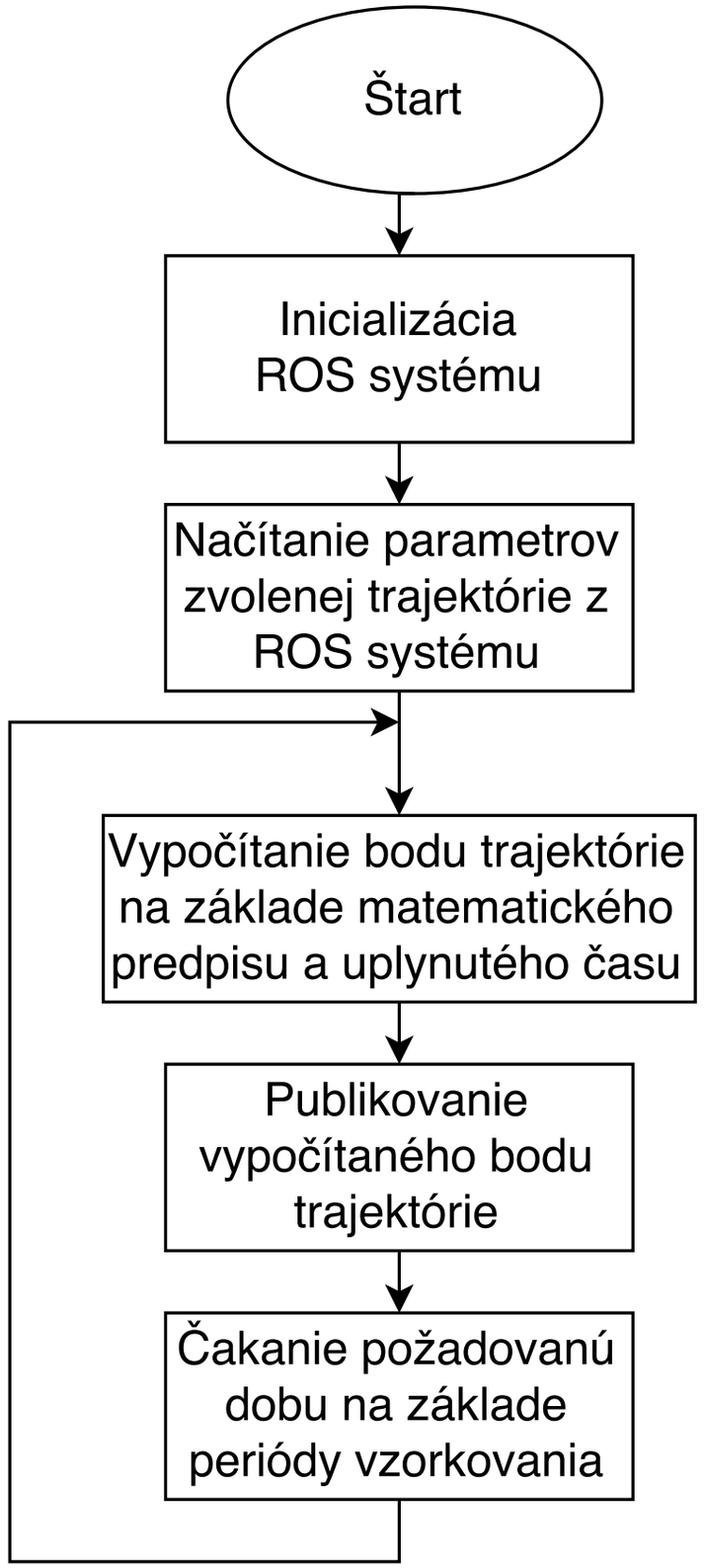




Implementácia modulu Trajectory

- Generovanie zvolených trajektórií
- Vstup:
  - –
- Výstup:
  - generované body trajektórie - **/blue/robot_x/traj/situation**




Štart
Inicializácia ROS systému
Načítanie parametrov zvolenej trajektórie z ROS systému
Vypočítanie bodu trajektórie na základe matematického predpisu a uplynutého času
Publikovanie vypočítaného bodu trajektórie
Čakanie požadovanú dobu na základe periódy vzorkovania

Pomocné moduly

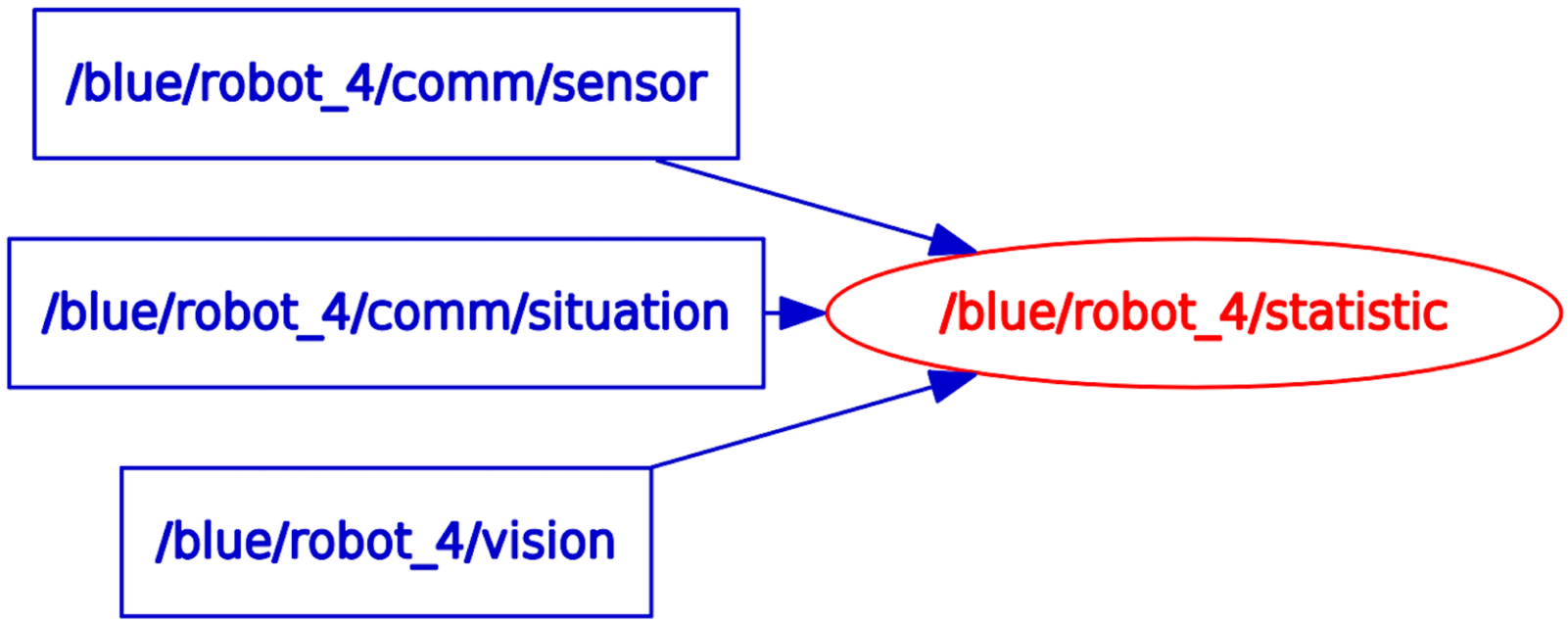
Ukladanie údajov

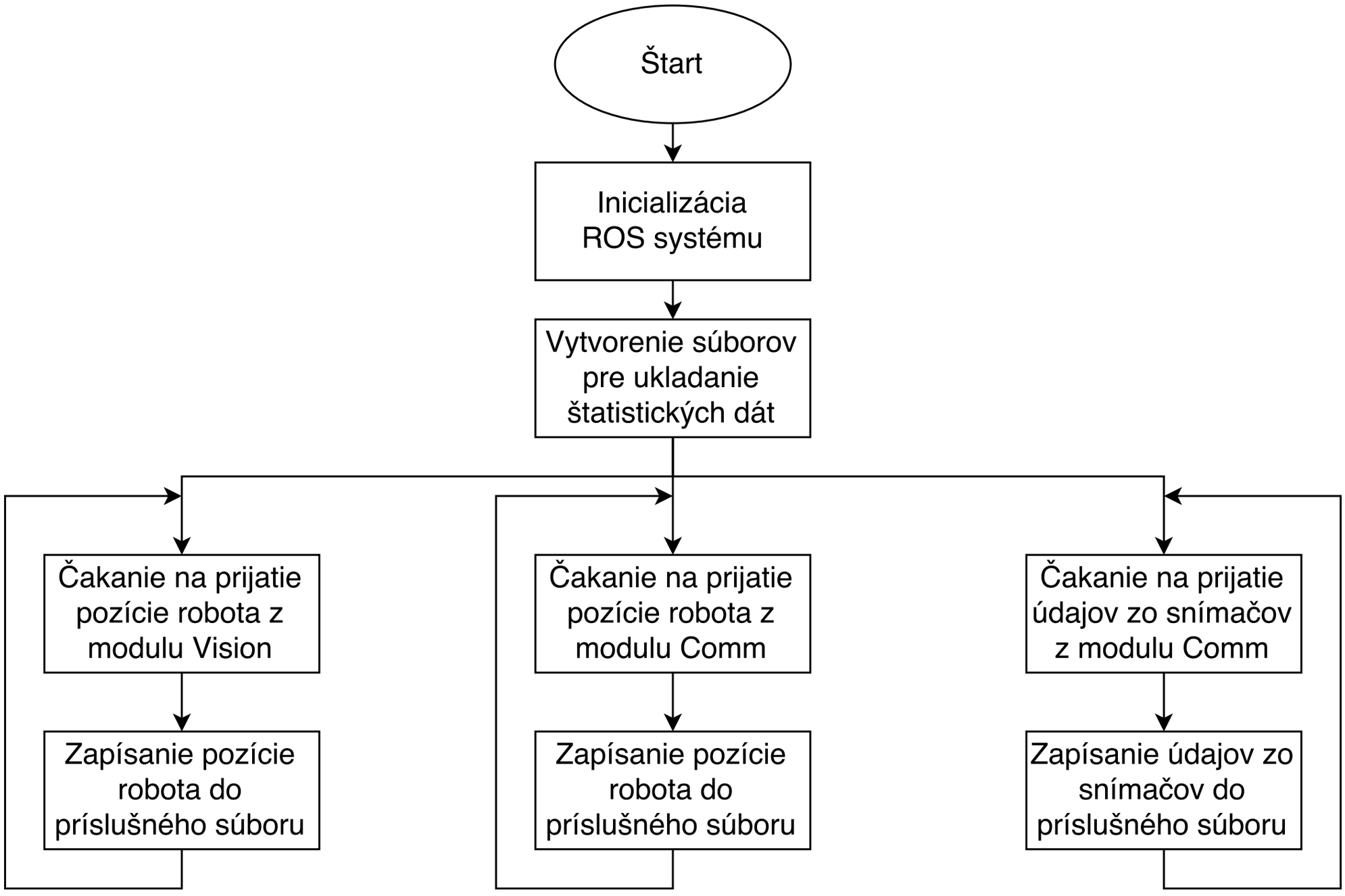
Table 1

T	x	y	a	rm_u	rm_i	rm_v	lm_u	lm_i	lm_v	acc_x	gyr_z	mag_x	mag_y
-912916507	0	0	0	0	0	0	0	0	-32	-1	-1	-1	-1
-912912873	0	0	0	0	0	0	0	0	-170	-1	-1	-1	-1
-912909058	0	0	0	0	0	0	0	0	-162	-1	-1	-1	-1
-912905028	0	0	0	0	0	0	0	0	-140	-1	-1	-1	-1
-912901435	0	0	0	0	0	0	0	0	-124	-1	-1	-1	-1
-912897993	0	0	0	0	0	0	0	0	-109	-1	-1	-1	-1
-912897921	0	0	0	0	0	0	0	0	-103	-1	-1	-1	-1
-912894277	0	0	0	0	0	0	0	0	-88	-1	-1	-1	-1
-912894200	0	0	0	0	0	0	0	0	-75	-1	-1	-1	-1
-912888782	0	0	0	0	0	0	0	0	-57	-1	-1	-1	-1

Implementácia modulu **Statistic**

- Ukladanie dostupných údajov do .csv
- Vstup:
  - údaje z robota - **/blue/robot_x/comm/sensor**
  - odometria robota - **/blue/robot_x/comm/situation**
  - poloha z kamery **- /blue/robot_x/vision/situation**
- Výstup:
  - –




Štart
Inicializácia ROS systému
Vytvorenie súborov pre ukladanie štatistických dát
Čakanie na prijatie pozície robota z modulu Vision
Zapísanie pozície robota do príslušného súboru
Čakanie na prijatie pozície robota z modulu Comm
Zapísanie pozície robota do príslušného súboru
Čakanie na prijatie údajov zo snímačov z modulu Comm
Zapísanie údajov zo snímačov do príslušného súboru

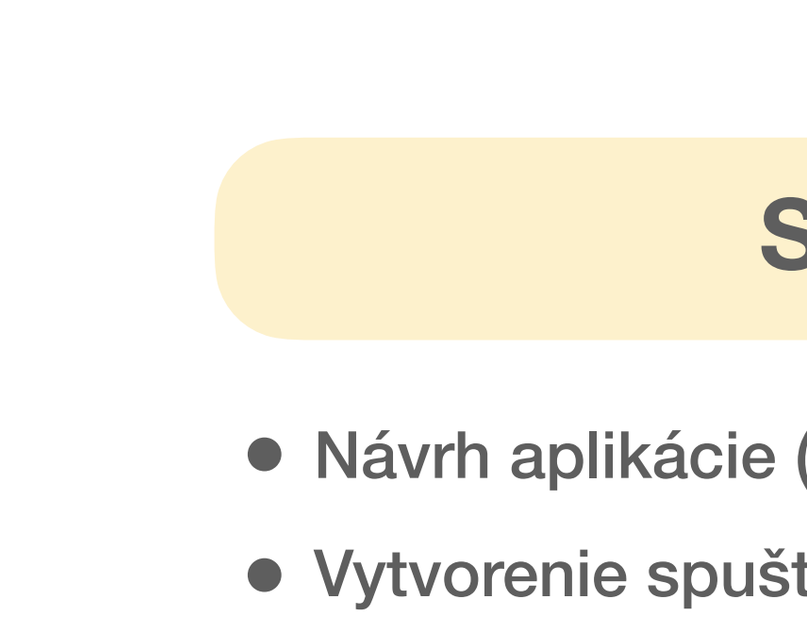
6 Aplikácia

Potrebné prostriedky

- PC s Linux Ubuntu
- Nainštalovaná platforma ROS
- USB kamera (PS3eye kamera)
- MiRoSoT robotický futbalisti
- Stiahnuté a skompilované moduly

Spúšťanie aplikácie

- Návrh aplikácie (prepojenie správ)
- Vytvorenie spušťacieho xml skriptu (skupiny modulov)
- Spustenie aplikácie pomocou platformy ROS
- Ladenie ROS parametrov v spúštacích skriptoch

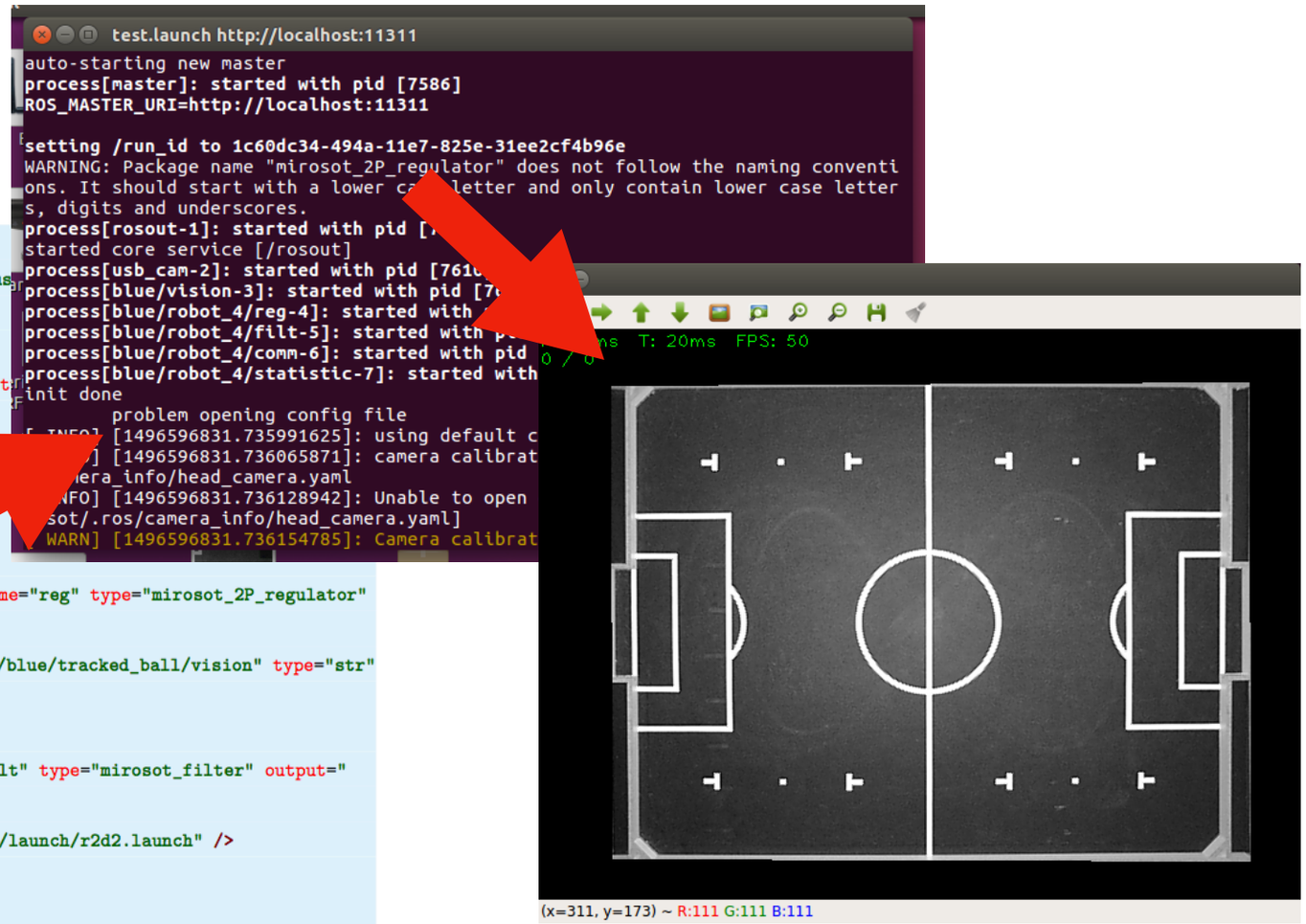



```
<launch>
  <include file="$(find usb_cam)/launch/us

  <group ns="blue">
    <param name="blue/team" value="blue" t

    <node pkg="mirosot_vision" name="vi
    <param name="calibration" value="

    <group ns="robot_4">
      <node pkg="mirosot_2P_regulator" name="reg" type="mirosot_2P_regulator"
            output="screen">
        <param name="traj_source" value="/blue/tracked_ball/vision" type="str"
              />
      </node>
      <node pkg="mirosot_filter" name="filt" type="mirosot_filter" output="
          screen" />
      <include file="$(find mirosot_comm)/launch/r2d2.launch" />
    </group>
  </group>
</launch>
```

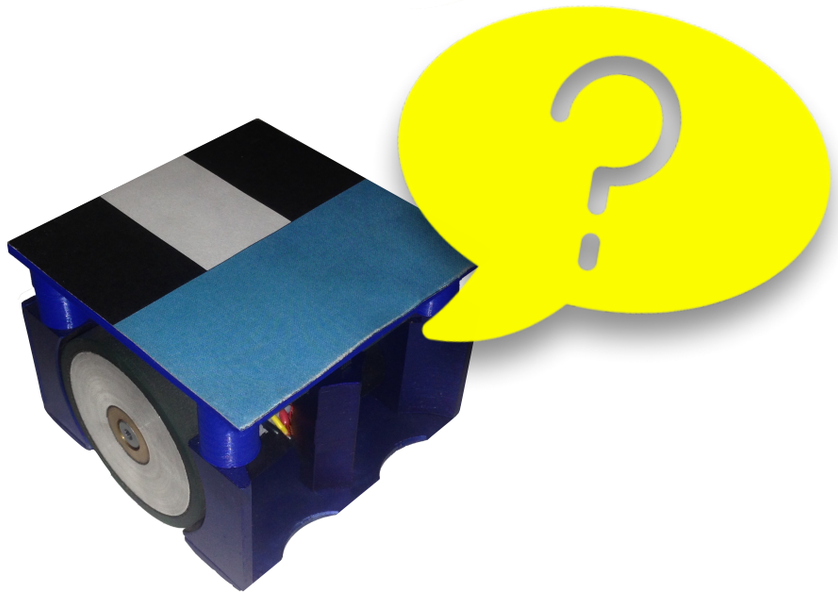


Vytvorené aplikácie

- Sledovanie referenčnej trajektórie (video)
- Ovládanie robota joystickom
- Sériové nasledovanie robotov - vláčik (video)
- Sledovanie loptičky robotom

Publikácie

- JADLOVSKÝ, J et al. 2017. Výskumné aktivity CMMRaPI na KKUI FEI TU v Košiciach. In: ATP Journal - článok v recenznom konaní
  - časť venujúca sa robotickému futbalu



Otázky oponenta

1. V akom vzťahu je ROS a linux? Existuje verzia ROS aj pre iné OS?

- Platforma ros vznikla pod OS Linux Ubuntu, kde pre jednotlivé verzie Ubuntu vychádzali špecifické verzie platformy ROS. Napr. pre OS **Linux Ubuntu 16.04** je to **ROS Kinetic Kame**.
- Vznikli rôzne experimentálne verzie platformy ROS pre OS **Windows** a **macOS**, ktoré však niesú podporované.

2. Ako by bolo možné uložiť dáta, ktoré zaznamenáva programový modul Statistic do Databázy v platforme ROS. Kde by tieto dáta mohli nájsť uplatnenie v zmysle distribuovaného systému riadenia v prípade robotického futbalu?

- V rámci platformy ROS existuje modul s názvom "sql_database", ktorý umožňuje ukladanie správ priamo do SQL databázy. Tento modul si na základe typu správy sám vytvorí správnu tabuľku v databáze a následne do nej ukladá údaje, ktoré sú odchýtavané počas behu aplikácie.
- Uplatnenie vidím napríklad v offline strojovom učení, kde by sa mohli roboti učiť taktiky druhých tímov, čo by umožnilo lepšie vyhýbacie manévre a streľby gólu. Ak by bol väčší dopyt po nameraných údajoch, bolo by možné vytvoriť webové rozhranie, ktoré by umožňovalo stiahnuť si namerané dáta podľa požiadaviek.

3. Aký vplyv má hardvérová zmena kamery na modul Vision a jeho výstupy? Konkrétne z pohl'adu rozlíšenia a počtu snímkov za sekundu.

- Kamera je charaktreristická hlavne rozlíšením a počtom snímok za sekundu. Oba tieto hodnoty vedia značne ovplyvniť chovanie rozpoznávania obrazu. Väčšie rozlíšenie spôsobuje spomalenie rozpoznávania obrazu koli väčším maticiam na spracovanie. Vyšší počet snímok nemusíme stíhať analyzovať, no ak tieto údaje zladíme vzhľadom na použitý hardvér, dostaneme dostatočne rýchle a presné rozpoznávanie obrazu a detekciu polôh robotov a loptičky.

4. Popíšte, ako by bolo potrebné upraviť moduly platformy ROS prezentovaného riešenia v prípade použitia iných mobilných robotov, napríklad typu Khepera a podobne.

- V prípade, žeby sme poznali protokol komunikácie, stačilo by aplikovať nastavovanie jednotlivých údajov v robotoch pomocou bluetooth rozhrania, stačilo by napríklad pomocou ROS parametrov v spúštacích skriptoch pozmeniť konštanty zosilnení v module regulátor ak by bolo treba (rôzna dynamika robotov)
- Ak by sme protokol komunikácie nepoznali, bolo by potrebné vytvoriť nový modul komunikácie, ktorý by nahradil ten existujúci. Nový modul komunikácie by sa pripojil už na existujúce správy platformy ROS v rámci tohto riešenia a komunikoval by s robotmi, ktorých si zvolíme. Taktiež by mohol používať iné rozhrania a protokoly.

Ďakujem za pozornosť