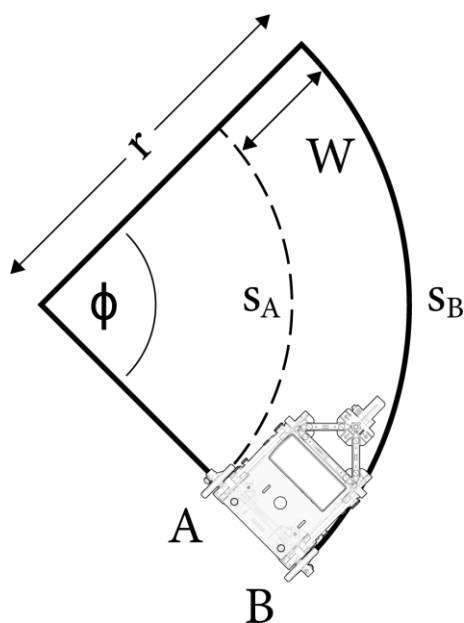

StructureBOT se točí v kruhu

Cíle

- Naprogramujte StructureBOT tak, aby provedl zatáčku, kde se obě kola pohybují vpřed, ale různými rychlostmi.

Materiály a vybavení

Číslo dílu	Popis	mn ožs tví
ME-7039	StructureBOT, sestaveno	1
PS-3232	//control.Node	1
	Software pro sběr dat PASCO Capstone™	



Obrázek 1. Otočení proti směru hodinových ručiček (kladný úhel).

Pozadí

Rychlost otáčení vnějšího kola (B na obrázku 1) bude nastavena na 2 ot/s. Rychlost otáčení vnitřního kola (A) musí být vypočtena tak, aby poskytla požadovaný poloměr otáčení.

Upřesněno: Následující parametry jsou specifikovány uživatelem nebo určeny rozměry BOT.

W = Šířka stopy (cm)
 D = průměr kola (cm)
 $R = D/2$ = poloměr kola (cm)
 r = Poloměr otáčení (cm)
 ϕ = Úhel natočení (rad) = [\pm Úhel (stupeň)] ($\pi/180$)
 ω_B = rychlost otáčení (v rad/s) vnějšího kola na poloměru r

Odvozený: Následující veličiny jsou odvozeny ze zadaných veličin.

θ_B = úhel (rad), o který se otáčí vnější (pravé) kolo
 θ_A = úhel (rad), o který se otáčí vnitřní (levé) kolo
 ω_A = rychlost otáčení (v rad/s) vnitřního kola na poloměru W

Definovat:

s_B = délka oblouku vnější dráhy
 s_A = délka oblouku vnitřní dráhy

Toto odvození je pro otočení proti směru hodinových ručiček (kladný úhel), jak je znázorněno na obrázku 1. Vnější kolo urazí určitou vzdálenost .

Vnitřní kolo ujede kratší vzdálenost $s_A = (r - W)\phi$.

Na vnějším kruhu se kolo otočí o úhel θ_B , když kolo urazí vzdálenost s_B . s_B = (obvod kola) (počet otáček kola)

$$s_B = (2\pi R) \left(\frac{\theta_B}{2\pi} \right) = R\theta_B$$

$$r\phi = R\theta_B$$

Řešení pro θ_B dává vztah úhlu natočení vnějšího kola θ_B k určenému úhlu kružnice ϕ .

$$\theta_B = \left(\frac{r}{R} \right) \phi \quad (1)$$

Podobně pro vnitřní kruh,

s_A = (obvod kola) (počet otáček kola)

$$s_A = (2\pi R) \left(\frac{\theta_A}{2\pi} \right) = R\theta_A$$

$$(r - W)\phi = R\theta_A$$

To dává vztah k úhlu, o který se vnitřní kolo otáčí, θ_A k určenému úhlu kružnice ϕ .

$$\theta_A = \left(\frac{r-W}{R}\right)\phi \quad (2)$$

Dosažením ϕ z rovnice (1) do rovnice (2) dostaneme:

$$\theta_A = \left(\frac{r-W}{R}\right)\left(\frac{R}{r}\right)\theta_B$$

$$\theta_A = \left(\frac{r-W}{r}\right)\theta_B$$

Vydělením obou stran této rovnice časem získáme vztah mezi rychlostí otáčení vnitřního kola ω_A a specifikovanou rychlostí otáčení vnějšího kola ω_B .

$$\omega_A = \omega_B \left(1 - \frac{W}{r}\right) \quad (3)$$

Postup

Založit

1. Sestavte StructureBOT, jak je popsáno v „Sestavení StructureBOT s pohonem 2 kol“.
2. Nabijte //control.Node.
3. Ujistěte se, že je krokový motor na levé straně BOT zapojen do portu A //control.Node a že krokový motor na pravé straně BOT je zapojen do portu B.
4. Připojte //control.Node k softwaru PASCO Capstone přes Bluetooth.

POZNÁMKA: Pokyny týkající se softwarových úloh, jako je připojení bezdrátových zařízení a používání Blockly, naleznete v nabídce Nápověda v PASCO Capstone.

Vytvořte kruhovou funkci

1. Spustíte nový experiment a vytvoříte funkci nazvanou Kruh s parametry Poloměr otáčení (cm) a \pm Úhel (stupně).
2. Vytvořte následující proměnné Blockly:
 - \pm Úhel (rad)
 - Levý úhel (rad)
 - Rychlost doleva (rad/s)
 - R (cm)
 - Pravý úhel (rad)
 - Správná rychlost (rad/s)
 - rychlost (rad/s)
 - rychlost (ot/s)
 - Šířka stopy (cm)
 - Průměr kola (cm)

3. Uvnitř funkce Kruh nastavte proměnné, jak je uvedeno níže:

```

to Circle with: Turn Radius (cm), ± Angle (deg)
  set Wheel Diameter (cm) to 6.35
  set R (cm) to Wheel Diameter (cm) + 2
  set Track Width (cm) to 17.35
  if Turn Radius (cm) < Track Width (cm)
    do set Turn Radius (cm) to Track Width (cm)
  set speed (rev/s) to 2
  set speed (rad/s) to 2 * pi * speed (rev/s)
  set ± Angle (rad) to pi / 180 * ± Angle (deg)
  
```

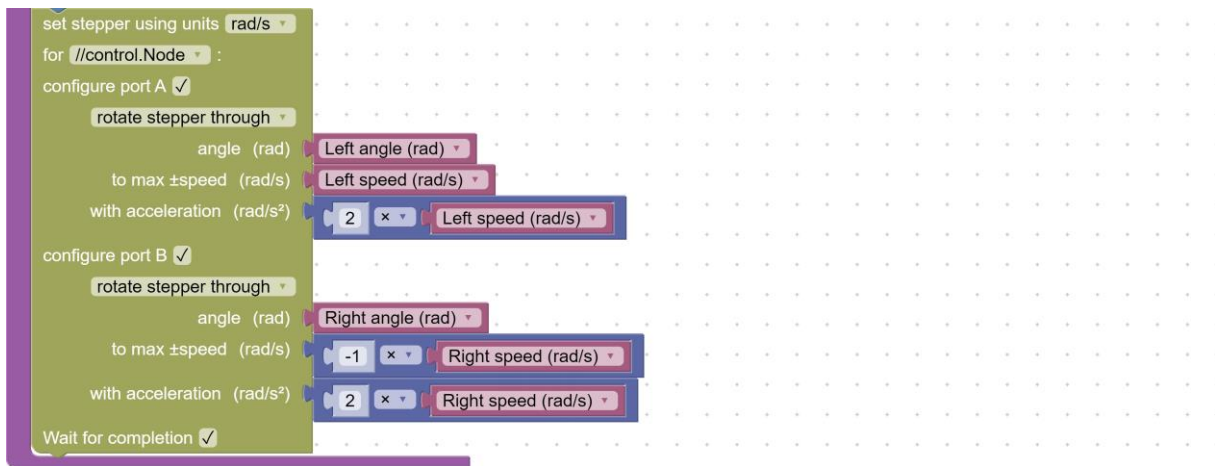
Všimněte si, že čísla pro Průměr kola a Šířku stopy se mohou pro váš BOT lišit. Všimněte si také, že příkaz if zabraňuje uživateli používat záporný poloměr zatačky nebo poloměr zatačky, který je menší než šířka stopy. Když se poloměr otáčení rovná šířce stopy, BOT se otočí kolem vnitřního kola.

4. Vytvořte příkaz if pro správné nastavení parametrů pro kladný ± Úhel (stupeň) nebo záporný ± Úhel (stupně). Tento znak určuje, které kolo (levé nebo pravé) je vnější kolo v kruhu. Odvozené rovnice byly pro případ kladného úhlu v jiné části příkazu if. Rovnice pro případ záporného úhlu (v části do příkazu if) jsou stejné, až na to, že levá a pravá rovnice jsou zaměněny.

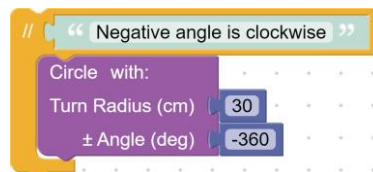
```

if ± Angle (deg) is negative
  do
    set ± Angle (rad) to absolute ± Angle (rad)
    set Right speed (rad/s) to speed (rad/s) * (1 - Track Width (cm) / Turn Radius (cm))
    set Right angle (rad) to (Turn Radius (cm) - Track Width (cm) + R (cm)) * ± Angle (rad)
    set Left speed (rad/s) to speed (rad/s)
    set Left angle (rad) to (Turn Radius (cm) + R (cm)) * ± Angle (rad)
  else
    set Right speed (rad/s) to speed (rad/s)
    set Right angle (rad) to (Turn Radius (cm) + R (cm)) * ± Angle (rad)
    set Left speed (rad/s) to speed (rad/s) * (1 - Track Width (cm) / Turn Radius (cm))
    set Left angle (rad) to (Turn Radius (cm) - Track Width (cm) + R (cm)) * ± Angle (rad)
  
```

5. Vložte krokový blok nakonfigurovaný s portem A pro pravé kolo a portem B pro levé kolo. Všimněte si, že je vybrána možnost Čekat na dokončení.



6. Vytvořte blok Notes pro tuto funkci s výchozími hodnotami pro Poloměr otáčení (cm) a \pm Úhel (stupně).



7. Vyzkoušejte svůj kód pro -360° a $+360^\circ$. Zkuste 180° . Je poloměr kruhu správný? Opravte všechny problémy. Poté sbalte funkční blok a uložte kód jako circle.pcbx.
8. Spustěte nový experiment. Napište kód pomocí funkcí moveADistance a circle, aby BOT šel rovně o 60 cm, otoč se proti směru hodinových ručiček v kruhu o poloměru 30 cm o 180° a šel rovně o 60 cm. Jak daleko je koncový bod tohoto pohybu od počátečního bodu?