TIZIMLASHGAN XARM ROBOTLARINI PYTHON DASTURLASH TILI ORQALI BOSHQARISH

Mustafayev Hasan Baxtiyor o‘g‘li

E-mail:hasanboymustafayev@gmail.com

Andijon Mashinasozlik institute “Intellektual Boshqaruv va Kompyuter Tizimlari” fakulteti “Mexatronika va Robotatexnika“ yo’nalishi K 18-20 guruh 4-kurs talabasi

Annotatsiya. xArm*-xArm robotlari haqida gapiradigan bolsak bu robotlar koplab foydali ishlarni bajara oladi misol uchun imkoniyati cheklanganlarga yordam berish,yullarni kotarish, tushirish kabi bir qancha ishlarni bajarish uchun mo’ljallangan. xArm robotlarini boshqarishda Python dasturlash tilidan foydalanishimiz mumkin. Ushbu maqolada xArm robotlarini dasturlash orqali boshqarishni ko’rib chiqamiz. xArm roboti - UFACTORY korxonasi tomonidan ishlab chiqarilgan sanoat roborlaridir (manipulyatorlari). xArm robotlari koplab foydali ishlarni bajara oladi misol uchun imkoniyati cheklanganlarga yordam berish,yullarni kotarish, tushirish kabi bir qancha ishlarni bajarish uchun mo‘ljallangan. To‘g’ri og’ir bo‘lgan yuklarni ko‘tara olmasligi mumkin lekin bu hozircha faqatgina yengil bo‘lgan narsalar uchun.*

Amalga oshirish

XArm robotini boshqarish uchun Python kodini amalga oshirish bir nechta komponentlarni o'z ichiga oladi, jumladan, xArm SDK ni ishga tushirish, boshqaruv algoritmlarini amalga oshirish, sensor ma'lumotlarini qayta ishlash integratsiyasi va robot bilan o'zaro ishlash uchun foydalanuvchi interfeyslarini ishlab chiqish. Quyida biz kod parchalari va foydalanilgan algoritmlarning tushuntirishlari bilan birga amalga oshirishning asosiy jihatlari haqida umumiy ma'lumot beramiz.

XArm -ni ishga tushirish SDK:

xArm robotini boshqarishda birinchi qadam xArm SDK-ni ishga tushirish va robot bilan aloqa o'rnatishdir. Bu ulanish parametrlarini sozlashni (masalan, IP manzili, port raqami) va xArm kontroller obyektini ishga tushirishni o'z ichiga oladi.

```python
import xarm
# Initialize xArm controller
arm = xarm.Controller('192.168.1.100')
```

Boshqaruv algoritmlari:

Boshqarish algoritmlari qo'shma harakatlar, Kartezian harakatlar yoki yakuniy effektor pozalari kabi kerakli robot harakatlarini belgilash uchun amalga oshiriladi. Masalan, xArm-ni belgilangan qo'shma konfiguratsiyaga o'tkazish uchun ` move_joint ` funksiyasidan foydalanish mumkin.

```python
# Move xArm to specified joint configuration
target_joint_config = [0, -45, 45, 0, 0, 0]
arm.move_joint(target_joint_config)
```

Xuddi shunday, Dekart harakati uchun ` move_line ` funksiyasidan maqsadli yakunlovchi pozani belgilash uchun foydalanish mumkin.

```python
# Move xArm in a straight line to specified pose
target_pose = [300, 200, 500, 180, 0, 0]
arm.move_line(target_pose)
```

Sensor ma'lumotlarini qayta ishlash integratsiyasi :

Sensor ma'lumotlarini qayta ishlash idrokga asoslangan boshqaruv strategiyalarini faollashtirish uchun boshqaruv dasturiga birlashtirilgan. Masalan, robot muhitidagi ob'ektlarni aniqlash uchun kompyuter ko'rish algoritmlari yordamida kamera tasvirlari qayta ishlanishi mumkin.

```python
import cv2
# Capture image from camera
image = arm.get_image()
# Process image using OpenCV for object detection
# (Code for object detection algorithm goes here)
```

Foydalanuvchi interfeyslari:

Foydalanuvchi interfeyslari xArm roboti bilan o‘zaro aloqani osonlashtirish uchun ishlab chiqilgan bo‘lib, foydalanuvchilarga boshqaruv buyruqlarini belgilash, robot holatini kuzatish va sensor ma’lumotlarini vizualizatsiya qilish imkonini beradi. Grafik foydalanuvchi interfeyslari (GUI) Tkinter yoki PyQt kabi kutubxonalar yordamida amalga oshirilishi mumkin .

```python
import tkinter as tk
# Create GUI window
root = tk.Tk()
root.title("xArm Control")
# Add buttons, sliders, and other GUI components for controlling the xArm
# (Code for GUI components goes here)
# Main event loop
root.mainloop()
```

Qiyinchiliklar :

Amalga oshirish jarayonida bir qator qiyinchiliklarga duch kelishi mumkin, jumladan:

1. Aloqa kechikishi: Boshqarish dasturi va xArm roboti o‘rtasidagi past kechikishli aloqani ta’minlash real vaqt rejimida boshqarish uchun zarur.

Tarmoqdagi kechikishlar yoki paketlarning yo'qolishi robotning sezgirligiga ta'sir qilishi mumkin, bu esa aloqa protokollari va buferni boshqarish strategiyalarini

optimallashtirishni talab qiladi.

2. Sensor integratsiyasi: Turli manbalardan (masalan, kameralar, chuqurlik datchiklari) sensor ma’lumotlarini birlashtirish va ularni robot boshqaruvi buyruqlari bilan sinxronlashtirish qiyin bo‘lishi mumkin. Sensorni to'g'ri kalibrlash va ma'lumotlarni moslashtirishni ta'minlash idrokga asoslangan nazorat vazifalari uchun juda muhimdir.

3. Xatolarni hal qilish:Aloqadagi nosozliklar yoki sensorlarning ishlamay qolishi kabi kutilmagan hodisalarni hal qilish uchun ishonchli xatolarni boshqarish mexanizmlarini joriy qilish robot tizimining xavfsizligi va ishonchliligini ta’minlash uchun zarurdir.

4. Algoritmni optimallashtirish: Samaradorlik va aniqlik uchun boshqaruv algoritmlarini optimallashtirish, ayniqsa yo‘lni rejalashtirish yoki mashina o‘rganishga asoslangan boshqaruv kabi hisoblash intensiv vazifalari uchun zarur. Ishlash samaradorligini oshirish uchun algoritmni parallellashtirish va apparat tezlashtirish kabi usullardan foydalanish mumkin.

Ushbu muammolarni hal qilish va Python yordamida boshqaruv dasturini joriy qilish orqali xArm robotini samarali boshqarish va sanoat avtomatizatsiyasidan tortib tadqiqot va ta'limgacha bo'lgan turli robotik ilovalarga integratsiyalash mumkin. Python-ning moslashuvchanligi va kengaytirilishi xArm SDK va tegishli kutubxonalarning imkoniyatlari bilan birgalikda murakkab vazifalarni avtonom va odamlar bilan hamkorlikda bajarishga qodir bo'lgan murakkab robot tizimlarini ishlab chiqish imkonini beradi.

Albatta, bu erda sinov va baholash bo'yicha kengaytirilgan bo'lim:

Xulosa

Xulosa qilib shuni aytish mumkinki hozirgi kunda avtomatlashtirish va ishlab chiqarish sohalarida robotlar keng qo’llanilyapti va insonlarning asosiy yordamchisiga aylanib boryapti.Bunga sabab ishlab chiqarish jarayonida insonlarning havsizligini taminlash, inson hayot faoliyati havfsizligiga tasir ko’rsatishi mumkin bo’lgan jarayonlarda robotlardan foylanib ishlab chiqarish jarayonini sifatini oshirish. Hozirgi kunda ishlab chiqarish jarayonlarida

ishlatilayotgan robotlar turlari juda ham ko’p. xArm robotlari ishlash prinsipi inson qo’li ishlashiga asoslangan bo’lib xArm robotini boshqarish uchun turli usullardan foydalanishimiz mumkin. Men ushbu maqolamda xArm robotlarini Python dasturlash tili orqali boshqarish jarayonini ko’rsatib berdim. Python dasturlashdan tashqari xArm robotlarini xArm studioda live control bo’limi orqali va blocly control bo’limida blokli elemntlari orqali boshqarish mumkin.

FOYDALANILGAN ADABIYOTLAR RO‘YXATI

1.Robot control devices:Circuit design and programming. Predko M.2014, 402p

2.Robotics Experiments for the Evil Genius (TAB Robotics) 1st Edition. ByMvkePredko. 2008. – 296p. ISBN-10: 0071413588.

3.Хартов В. Я. Микропроцессорные системи : учеб. Пособие для вузов / Хартов В. Я – 2-изд., испр. и доп. – М.: Академия, 2014. – 367 с.

5.http://help.ufactory.cc/en/articles/4491842-the-difference-between-ufactory-xarm5-ufactory-xarm6-and-ufactory-xarm7

6.http://download.ufactory.cc/xarm/en/Specification%20for%20xArm_20191021.pdf

7. http://microkontroller.ru/

8.http://help.ufactory.cc/en/articles/4491842-the-difference-between-ufactory-xarm5-ufactory-xarm6-and-ufactory-xarm7