Я-Профессионал

Робототехника

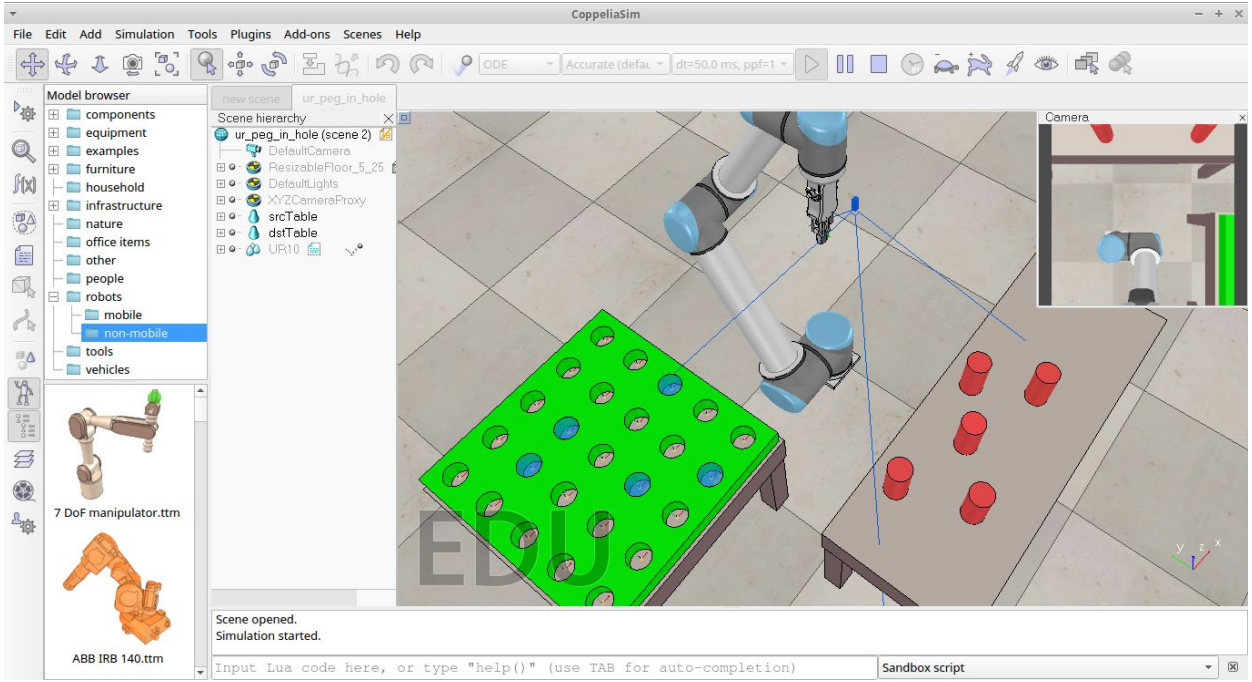
Вебинар 2

Спикеры: Станислав Михель, Альфия Хабибулина

Университет Иннополис

10 февраля 2020

Задание

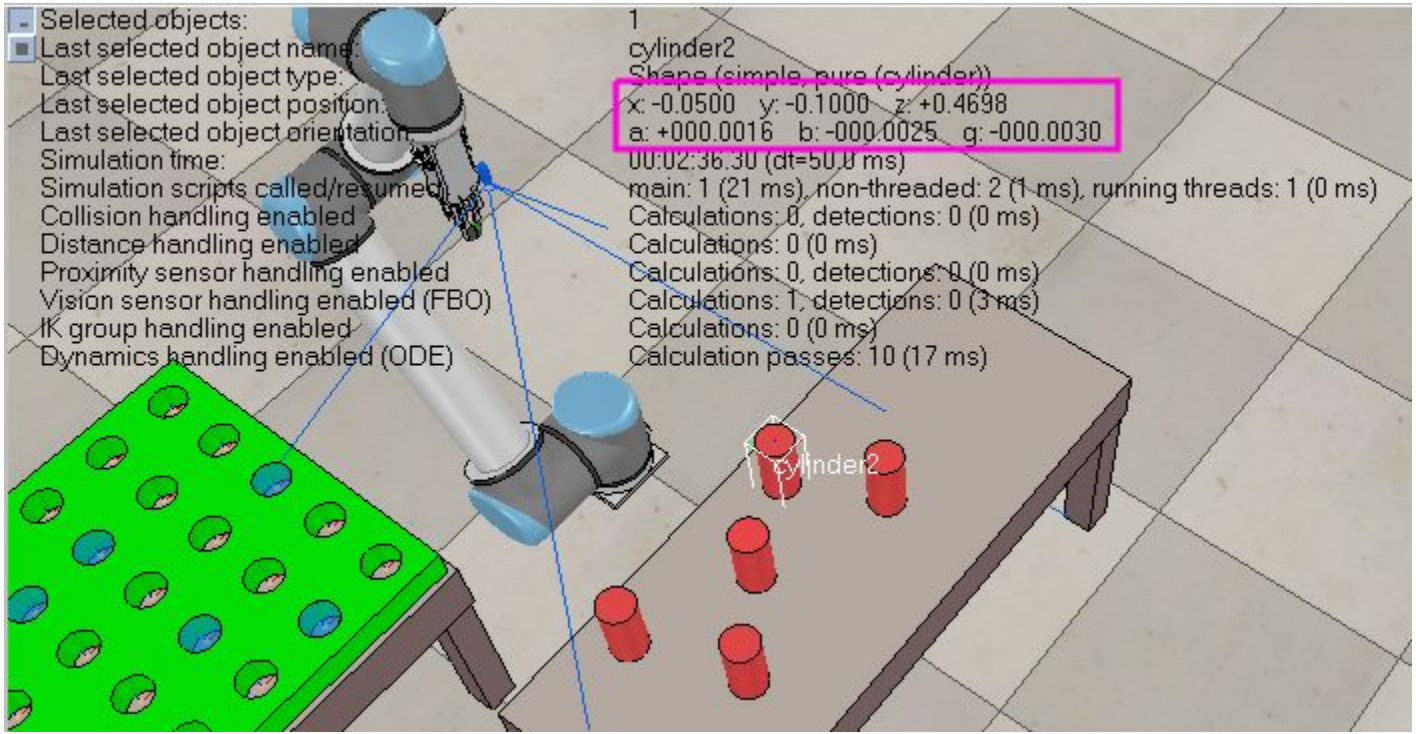


ur_main.py
ur_aux.py
ur_peg_in_hole.ttt

Проверка решения

- файл ur_main.py
- 5 попыток
- перестановка цилиндров и целевых отверстий
- функция ur.check()
- идентификация файла по m5-сумме

Взаимодействие с симулятором



Управление роботом

ptp(q) - движение в пространстве конфигураций

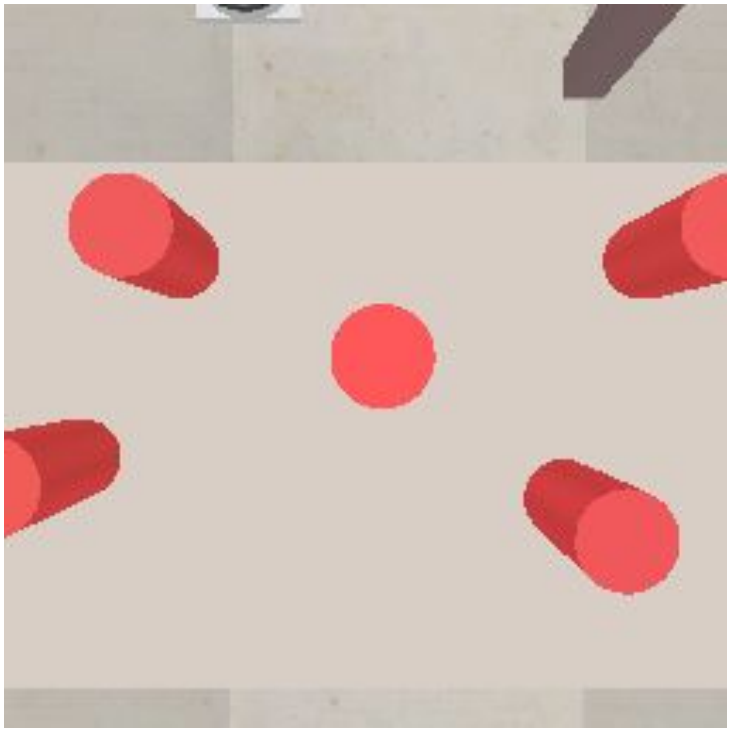
lin(p) - движение в декартовом пространстве с постоянной ориентацией

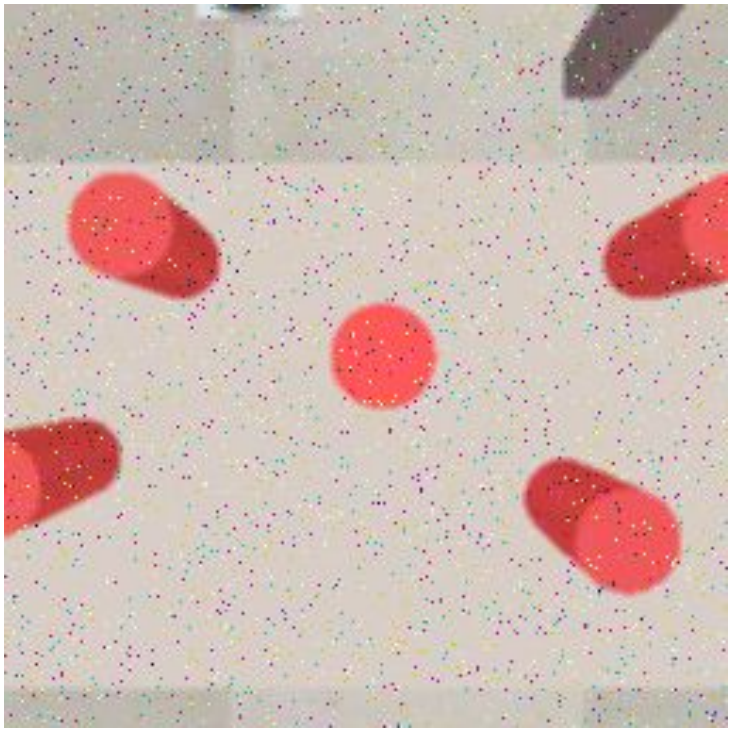
lin(p,o) - движение в декартовом пространстве для достижения требуемого положения и ориентации

gripperOpen() - управление инструментом

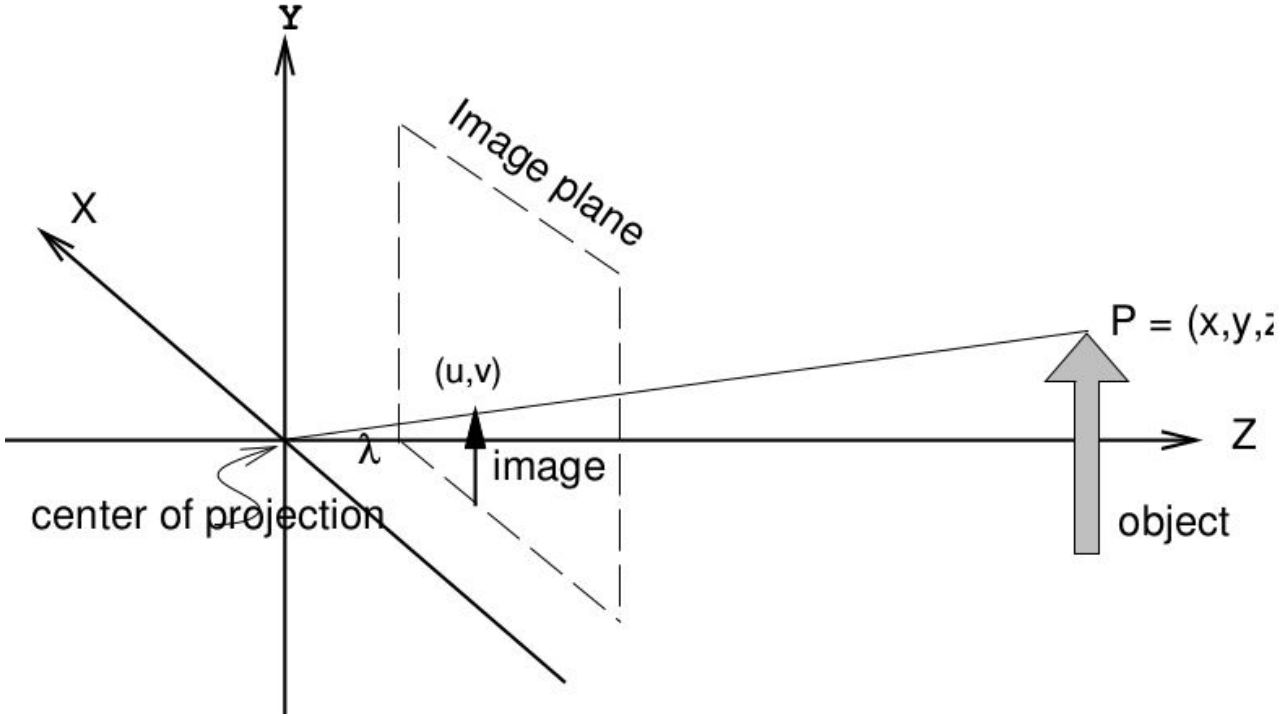
```
# go to some position in joint space
ur.ptp([0,rad(20),rad(-75),rad(-35),rad(90),0])

pos = ur.getToolPosition() # read current position

pos[1] -= 0.1
ur.lin(pos)             # go to new position in Cartesian space

orient = ur.getToolOrientation()
orient[0] += rad(45)
ur.lin(pos,orient)      # go to position and orientation

ur.gripperOpen(True)    # use True/False to open/close the gripper
```

Изображение с камеры





Преобразование координат

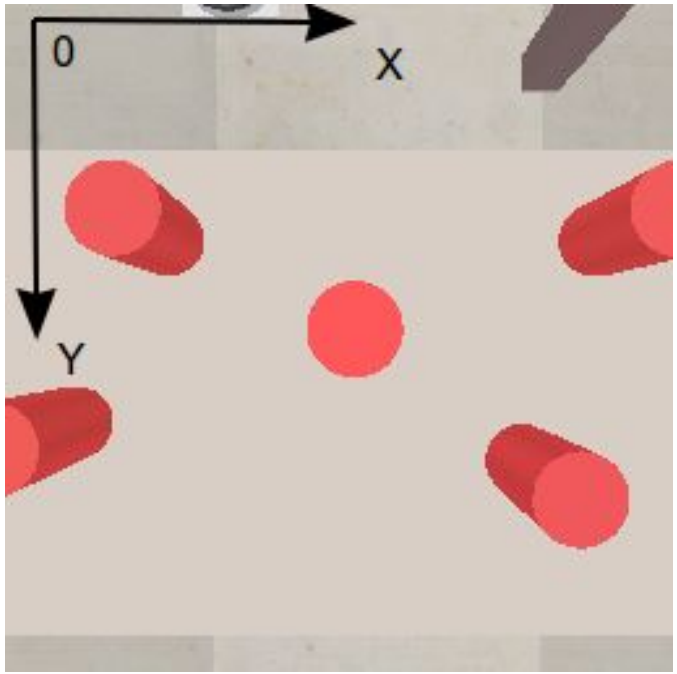


$$u = \lambda \frac{x}{z}$$

$$v = \lambda \frac{y}{z}$$

M. Spong “Robot modeling and control”

Изображение в OpenCV

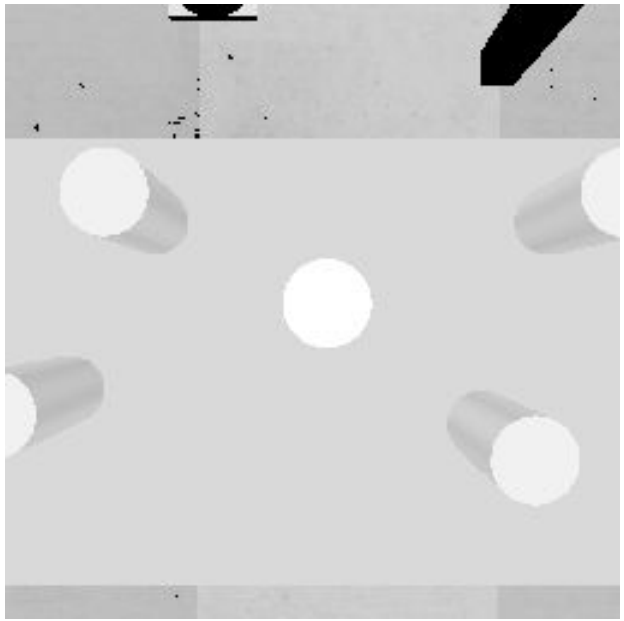


порядок координат: [Y, X]

порядок цветов: [blue, green, red]

Обработка: изображение в оттенках серого

```
_, gray = cv2.threshold(image[:,:,2], 180, 255, cv2.THRESH_TOZERO)
```



Обработка: преобразование Хафа

```
circles = cv2.HoughCircles(gray, cv2.HOUGH_GRADIENT, 1, 50, param1=50, param2=10,
                           minRadius=10, maxRadius=40)
```

результат:

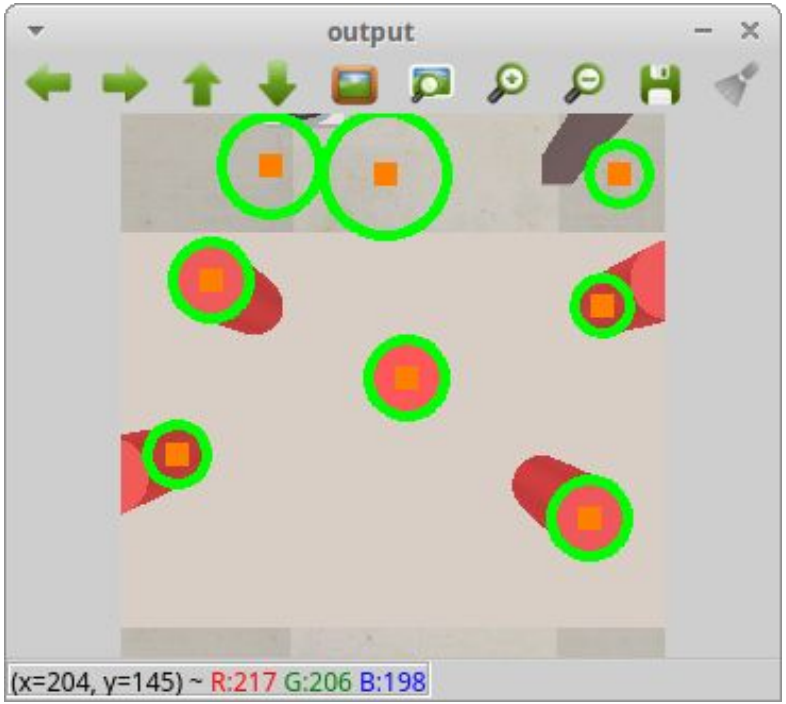
```
[[[133.5 124.5  18.1],  [ 41.5  77.5  18. ],  [220.5 189.5  18. ],  [ 26.5 160.5  13.8],  [225.5  89.5  13.4]
  [124.5  27.5  29. ],  [ 70.5  24.5  23.1],  [233.5  27.5  13.6]]]
```

Функция поиска

```
# Find pegs in the image
# Return: [(x1,y1,rad1),(x2,y2,rad2)...]
def findPegs(image, accumulator=10):
  _,gray = cv2.threshold(image[:,:,2], 180, 255, cv2.THRESH_TOZERO)  # pegs are red
  # find circles
  circles = cv2.HoughCircles(gray, cv2.HOUGH_GRADIENT, 1, 50, param1=50, param2=accumulator,
                             minRadius=10, maxRadius=40)
  if circles is None:
    return []
  else:
    circles = np.round(circles[0,:]).astype('int')
    isRed = lambda p,q: image[q,p,0] < 100 and image[q,p,1] < 100 and image[q,p,2] > 180
    return [x for x in circles if isRed(rng(x[0]),rng(x[1]))]
```

Визуализация результата

visualize(image,circles)



Использование силомоментного датчика

“Наивное” решение для незначительной коррекции положения инструмента при вставке цилиндра

```
pp = ur.getToolPosition()
for i in range(15):
  ff,tt = ur.getForceTorque()
  dx = -ff[0] / 1000.0
  dy =  ff[1] / 1000.0
  dz = -ff[2] / 2000.0
  pp[0] += dx
  pp[1] += dy
  pp[2] += dz
  ur.lin(p2)
```

Ссылки

Mark Spong, “Robot modeling and control”

Kenneth Dawson-Howe, “A practical introduction to computer vision with OpenCV”

https://opencv-python-tutroals.readthedocs.io/en/latest/

http://www.coppeliarobotics.com/helpFiles/en/remoteApiOverview.htm