Промежуточный отчет о проделанной работе

Задача

Внедрить в проект систему для избежания столкновений, подходящую для применения в телеуправляемом режиме в реальном времени

Если коротко, то хотим, чтобы при управлении джойстиком ошибки оператора не приводили к столкновению частей робота с самим собой и окружающим пространством

Основной план действий

Анализ литературы

Изучение кода проекта

Внедрение библиотеки Bullet

Отладка и тест

Основной принцип работы алгоритма избежания столкновений

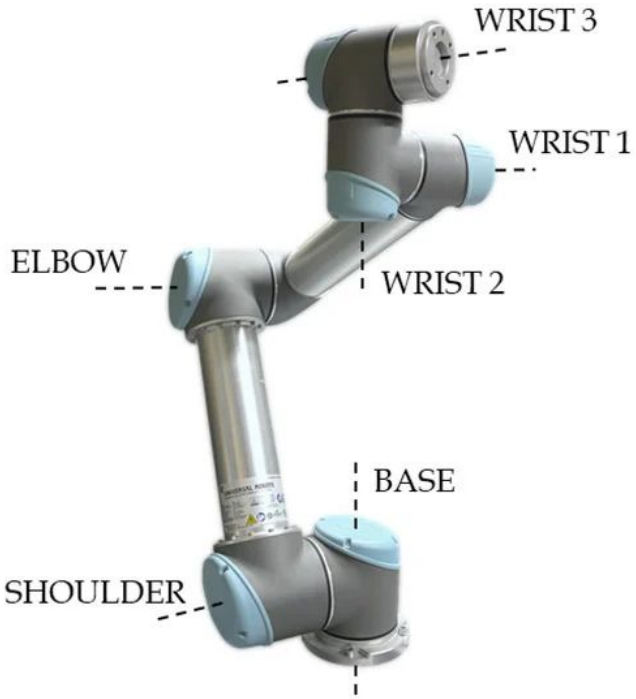
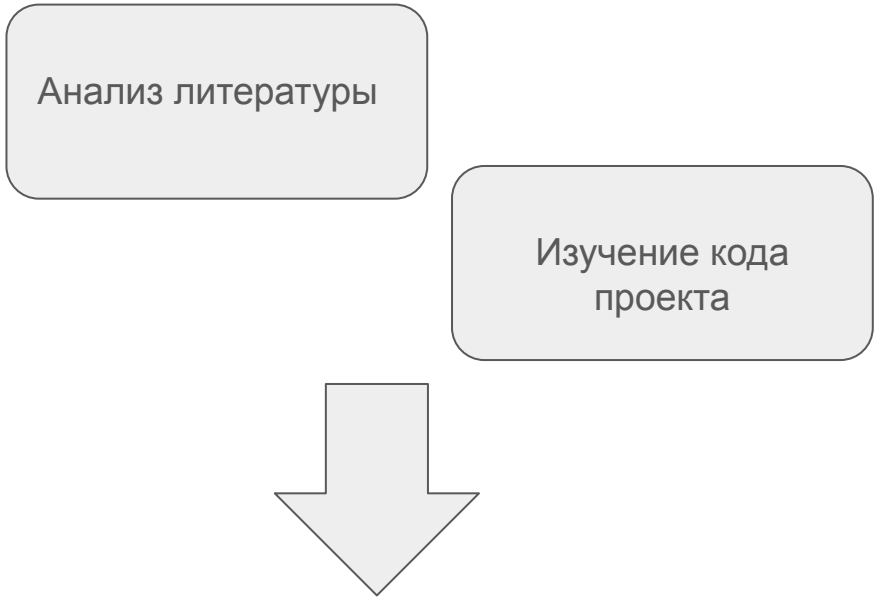


Схема Ur

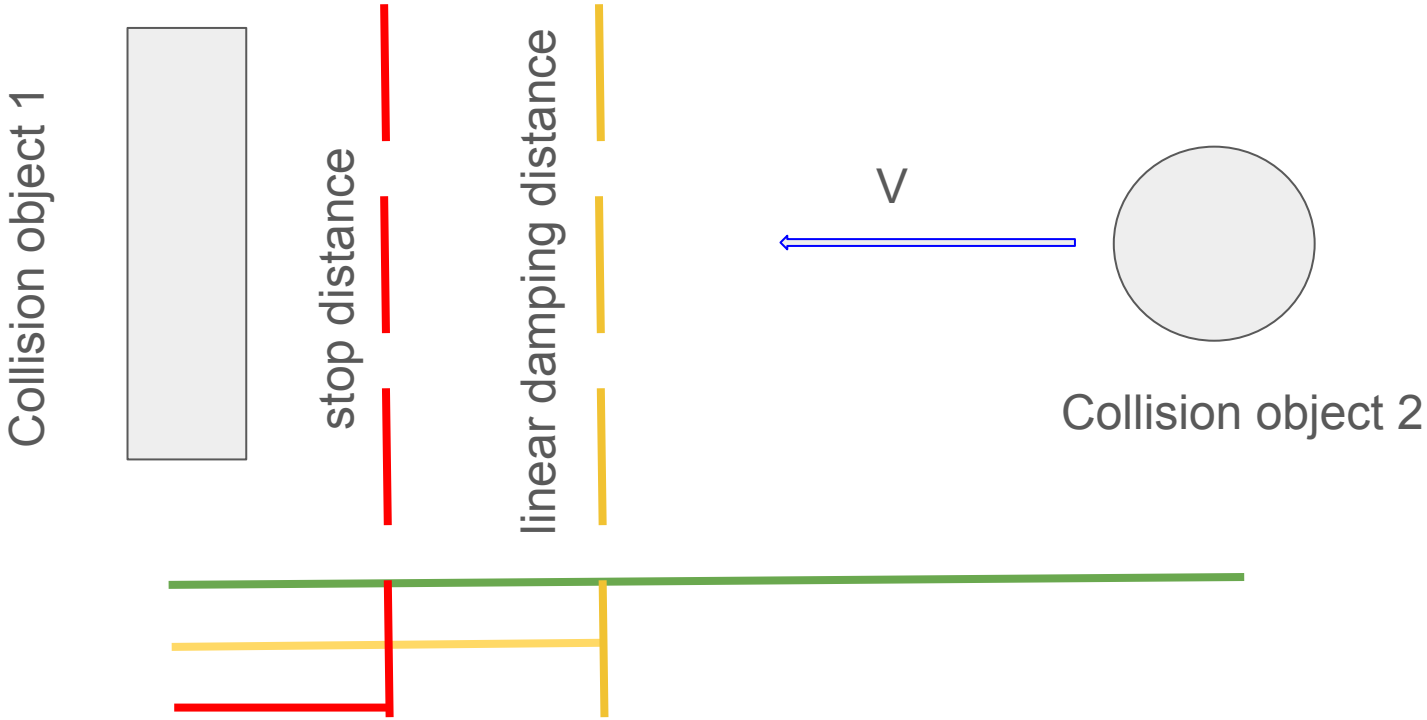
				0	1	2	3	4	5	6	7	8	9	10	11
base_link_inertia	0	-		-	-	-	-	-	-	1	1	-	-	-	-
coil_link	1	-		-	-	1	-	1	1	-	-	-	1	1	1
flange_link	2	-		-	1	-	-	1	1	-	-	-	1	1	1
forearm_link	3	-		-	-	-	-	-	-	-	-	1	1	-	-
ndi_marker_link	4	-		-	1	1	-	-	1	-	-	-	1	1	1
sensor_body	5	-		-	1	1	-	1	-	-	-	-	1	1	1
shoulder_link	6	-		1	-	-	-	-	-	-	1	1	-	-	-
tms_base	7	-		1	-	-	-	-	-	1	-	-	-	-	-
upper_arm_link	8	-		-	-	-	1	-	-	1	-	-	-	-	-
wrist_1_link	9	-		-	1	1	1	1	1	-	-	-	-	1	1
wrist_2_link	10	-		-	1	1	-	1	1	-	-	-	1	-	1
wrist_3_link	11	-		-	1	1	-	1	1	-	-	-	1	1	-

allowed collision matrix



Предлагаемый изначально метод APF равноценен уже реализованному методу Threshold Distance

Threshold Distance: суть метода



Проблема: библиотека FCL (flexible collision library), внедренная “по умолчанию”, работает медленно, вследствие чего не подходит для режима ручного управления

Решение: попробовать внедрить другую библиотеку, обещающую более быстрый рассчет расстояний

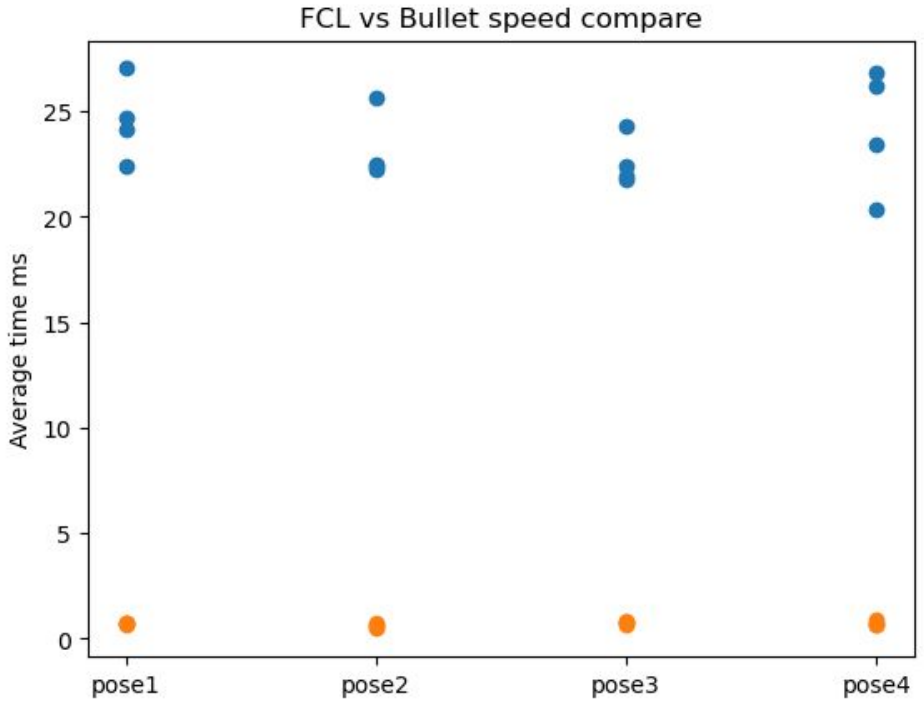
Внедрение
библиотеки Bullet

Был создан виджет на Qt для измерения скорости

Bullet действительно даёт существенный выигрыш в скорости вычисления расстояний

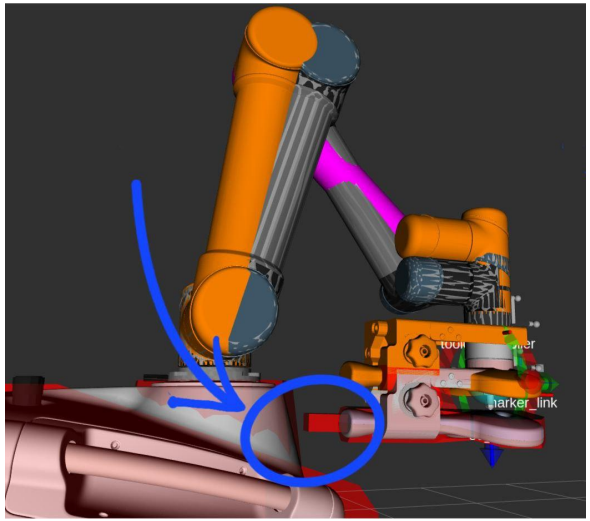
FCL ~ 24 ms
Bullet ~ 0.7 ms

выигрыш в скорости примерно
в 30 раз

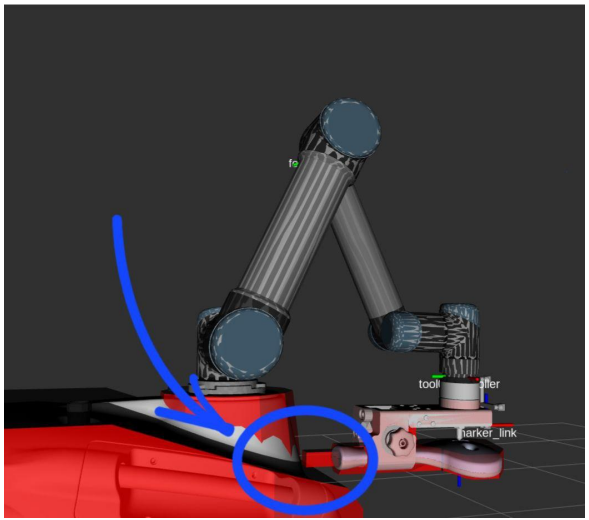


Проблема

При сравнении значений расстояний в одном и том же положении мы получали разные значения



distance = 0

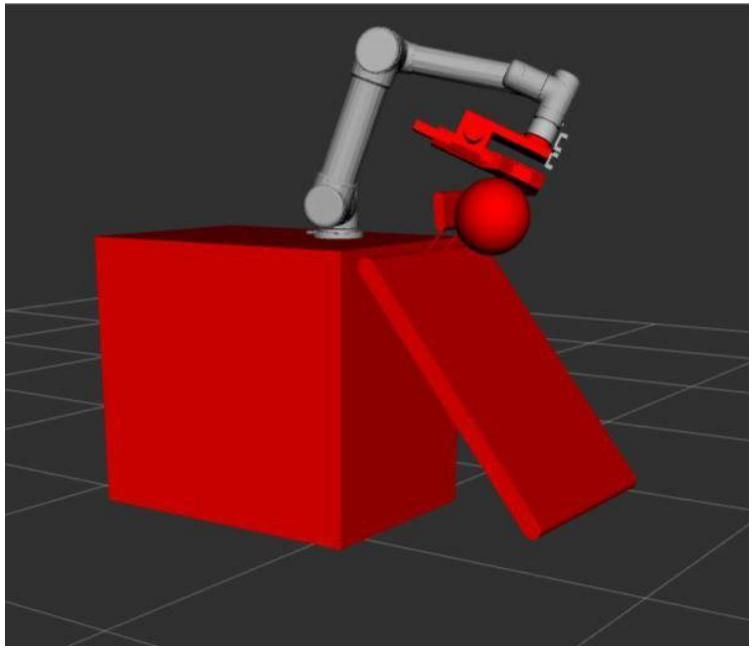


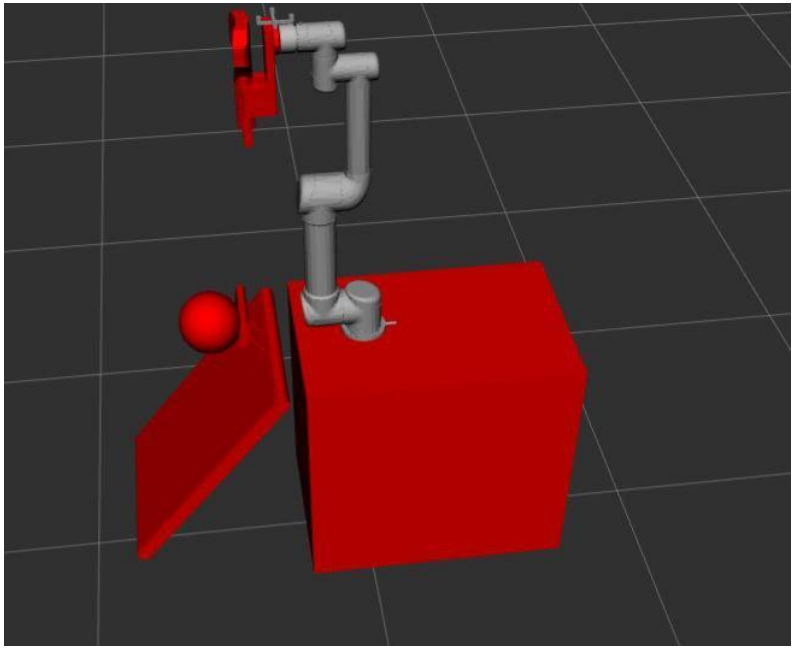
Bullet

FCL

Решение

После довольно долгого поиска выяснилось, что bullet плохо работает с невыпуклыми объектами, всвязи с чем необходимо было изменить некоторые URDF объекты





Оставшаяся работа

1. Подключить джойстик
2. Убрать “заглушки в коде”
3. Выбрать параметры collision shape (хорошо, что у нас есть значение “0”)
4. Протестировать работоспособность

```
double BulletCollisionCheck::computeScale(double current_dist)
{
  double warn=0;//FIXME
  double stop=3;//FIXME
  if (current_dist > warn)
    return 1.0;
  else if (current_dist <= warn && current_dist > stop)
  {
    double k = 1 / (warn - stop);
    double b = -k * stop;
    return k * current_dist + b;
  }
  else
    return 0.0;
}
```

Заглушка 1
подобрать warn и stop

```
if (use_collision_check)
{
  static ros::Duration max_elapsed;
  auto start = ros::Time::now();
  auto scale = 1.0; //calcIteration(now, future); //FIXME

  auto dummy = calcIteration(now, future); //FIXME
  //distance
  std_msgs::Float64 collision_test_distance;//FIXME
  collision_test_distance.data = dummy;//FIXME
  dummy_distance_publisher_.publish(collision_test_distance); //FIXME
```

Заглушка 2
сейчас всегда scale = 1