

# Sprawozdanie Lab 3

---

Andrzej Żaba, gr\_lab 4, nr indeksu: 401490

Do programu zaimportowano przygotowany model robota. Podzielono go na mniejsze elementy używając opcji:

**„Edit -> Grouping/Merging -> Divide selected shapes”.**

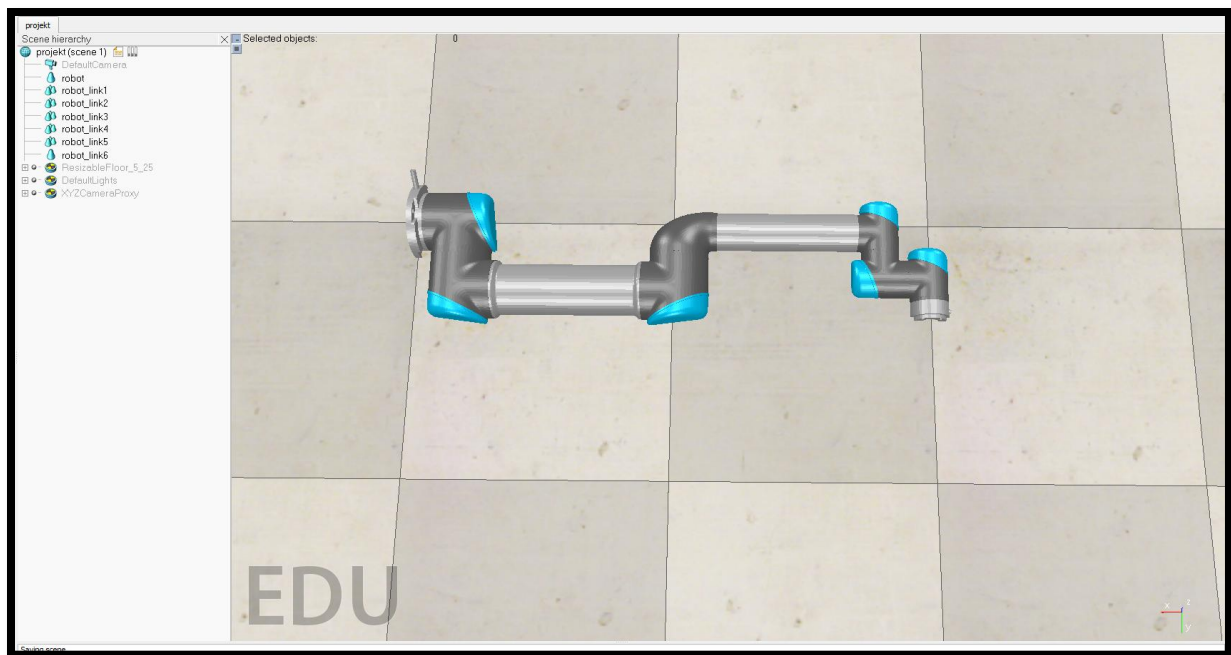
Otrzymano rezultat widoczny poniżej.



Następnie pokolorowano odpowiednio elementy oraz zgrupowano je poleceniem:

**„Edit -> Grouping/Merging -> Merge selected shapes”.**

Finalnie otrzymano 7 elementów.



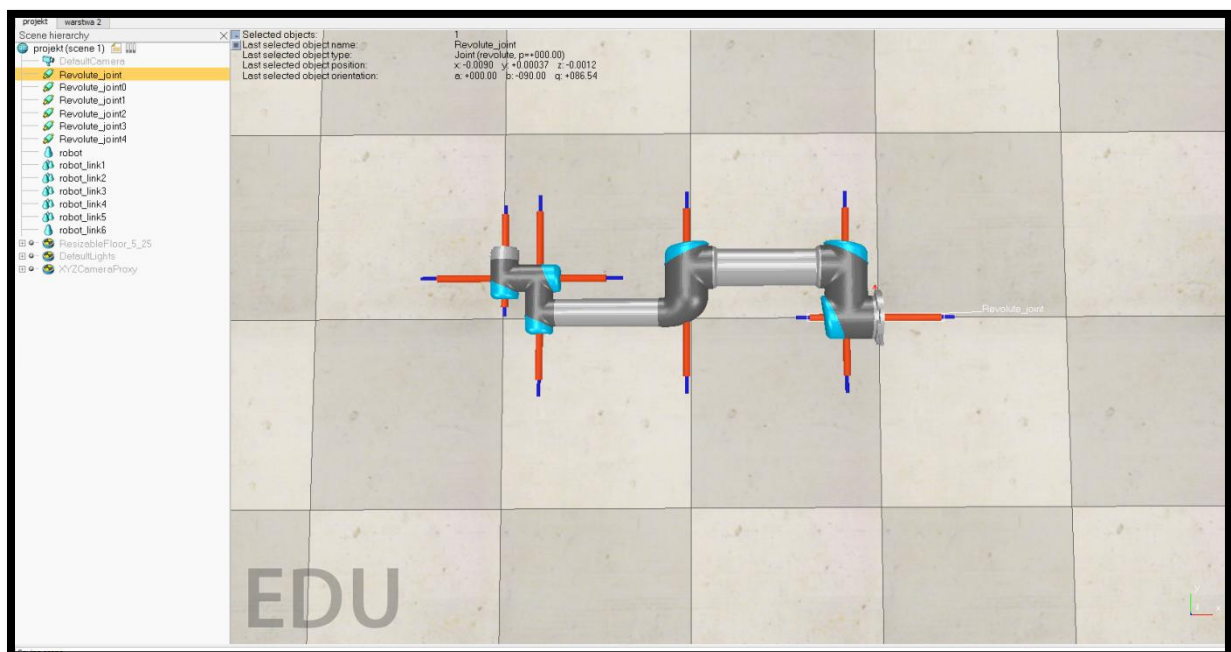
Dalej, pracując w nowej scenie oraz używając trybu edycji trójkątów (ikona po lewej stronie ekranu):

**„Toggle shape Edit mode -> Triangle Edit mode”**

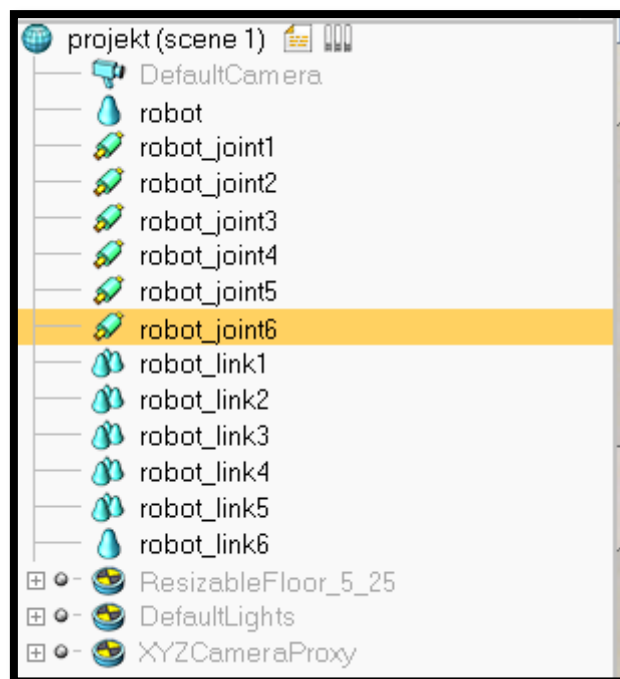
kopiowano poszczególne elementy i w nowej scenie dodawano im połączenia (Menu na górze):

**„Add -> joint -> revolute”.**

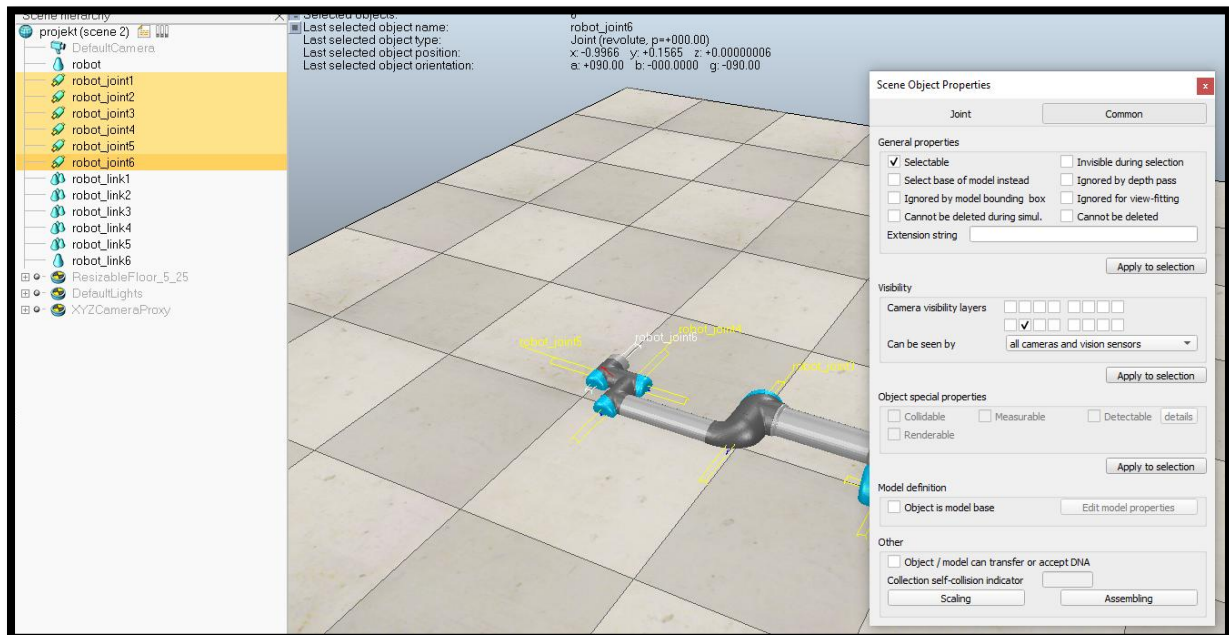
Połączenia dzięki pozycjonowaniu, przesunięciom i rotacjom dopasowano do konkretnych elementów a następnie przekopiowano do oryginalnej sceny.

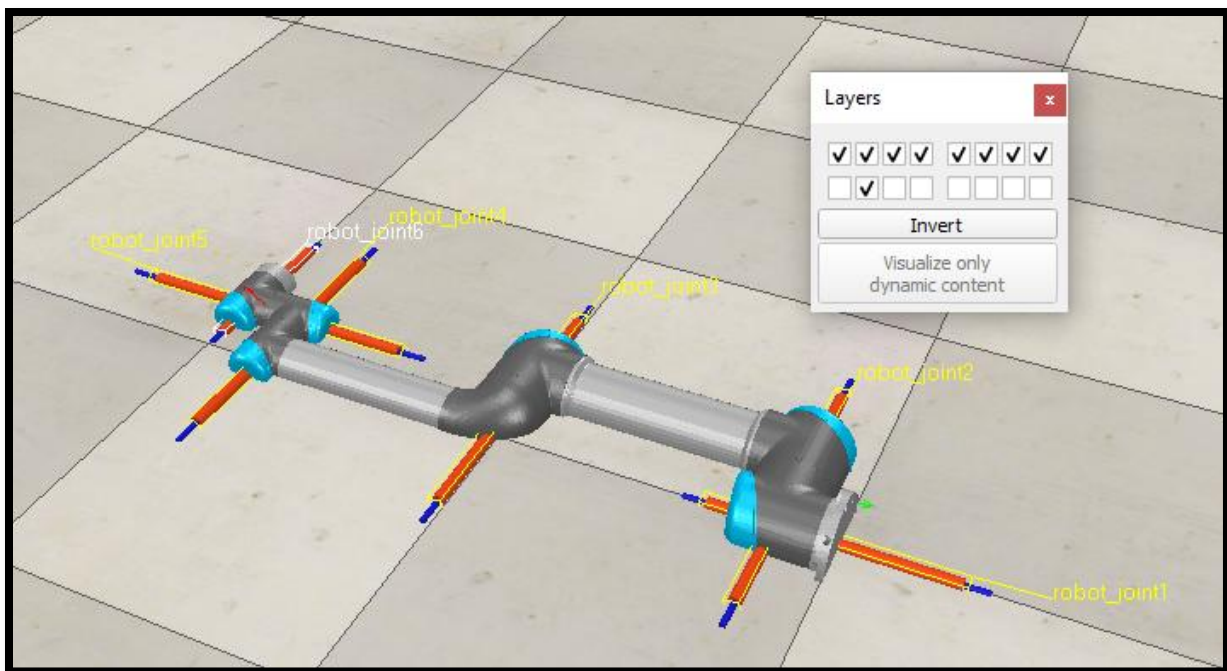


Nadano odpowiednie nazwy poszczególnym elementom.

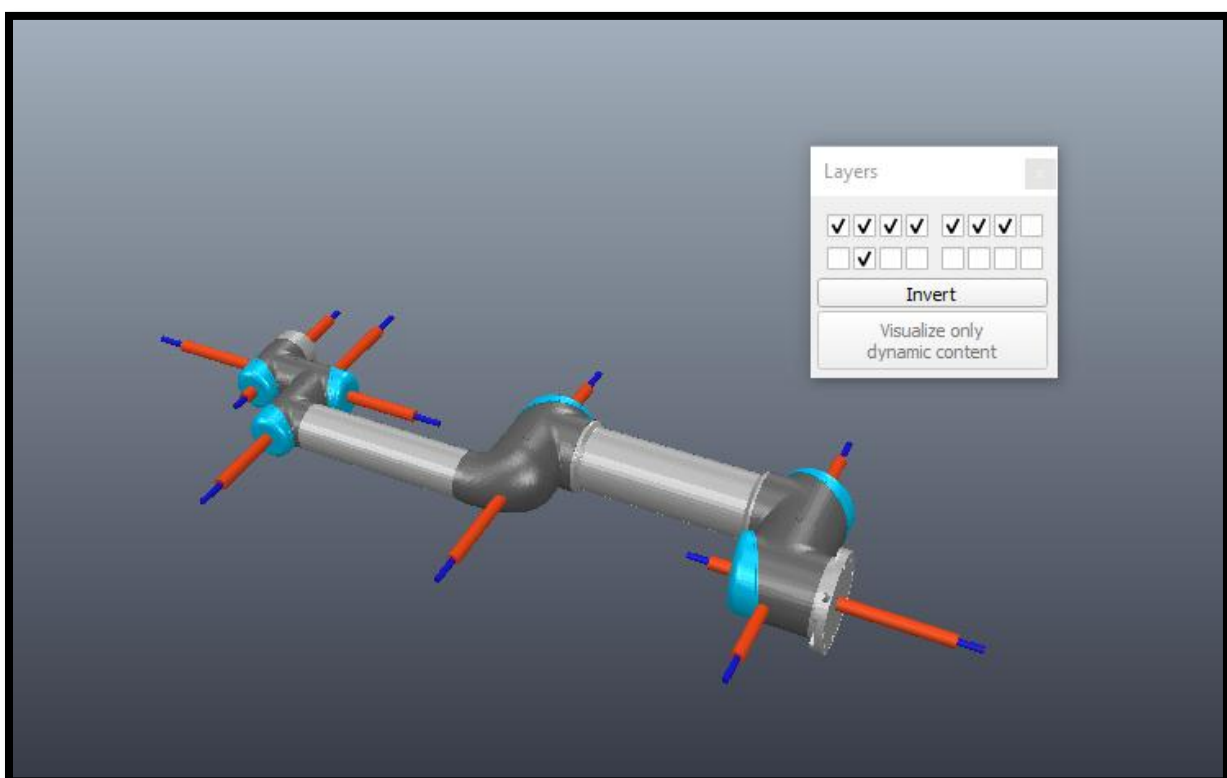


Nadano połączeniom 9 warstwę widoczności (chwilowo pozostawiono jednak również drugą)





Wyłączono widoczność podstawy (warstwa 8)



W kolejnym etapie stworzono dynamiczne części modelu. Używając nowej sceny kopiowano tam poszczególne elementy a następnie korzystając z **trybu edycji trójkątów** stworzono cylindry dla bazy robota, oraz dla elementu robot\_link6.

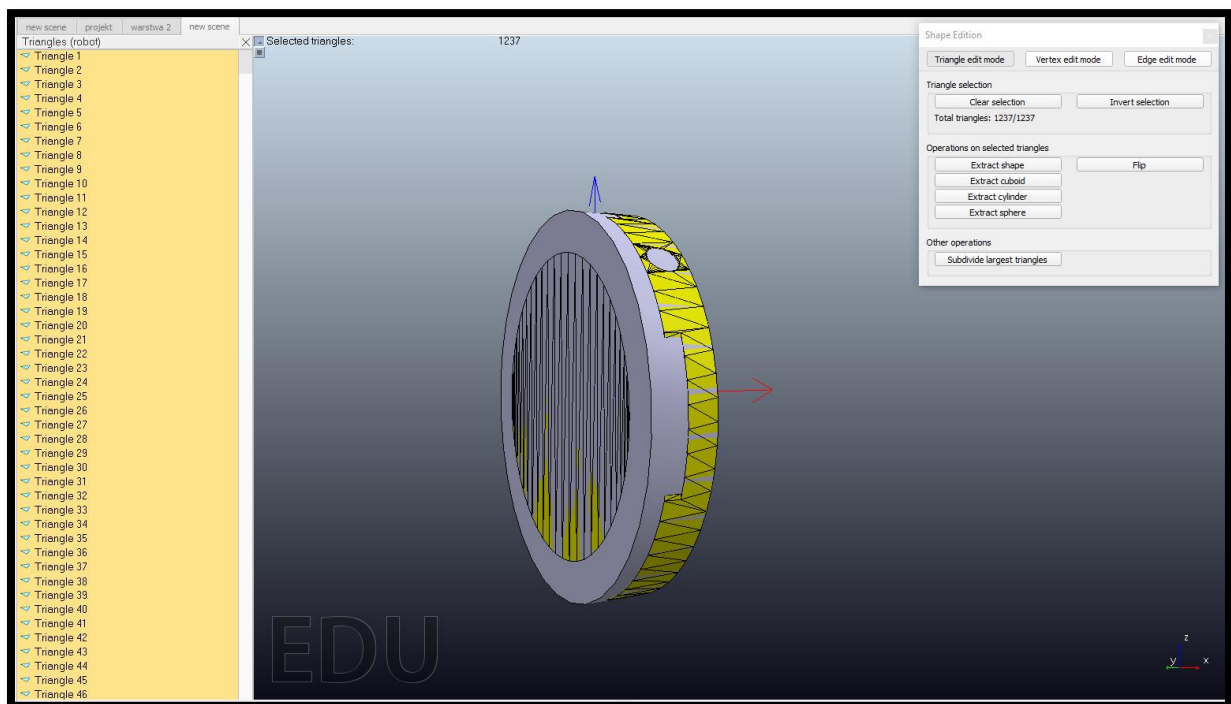
Kształty dynamiczne dla elementów robot\_link1,4,5 utworzono z użyciem funkcji (górne menu):

„Add -> Convex hull of selection”.

Dla elementów robot\_link2 i 3 użyto natomiast funkcji (górne menu):

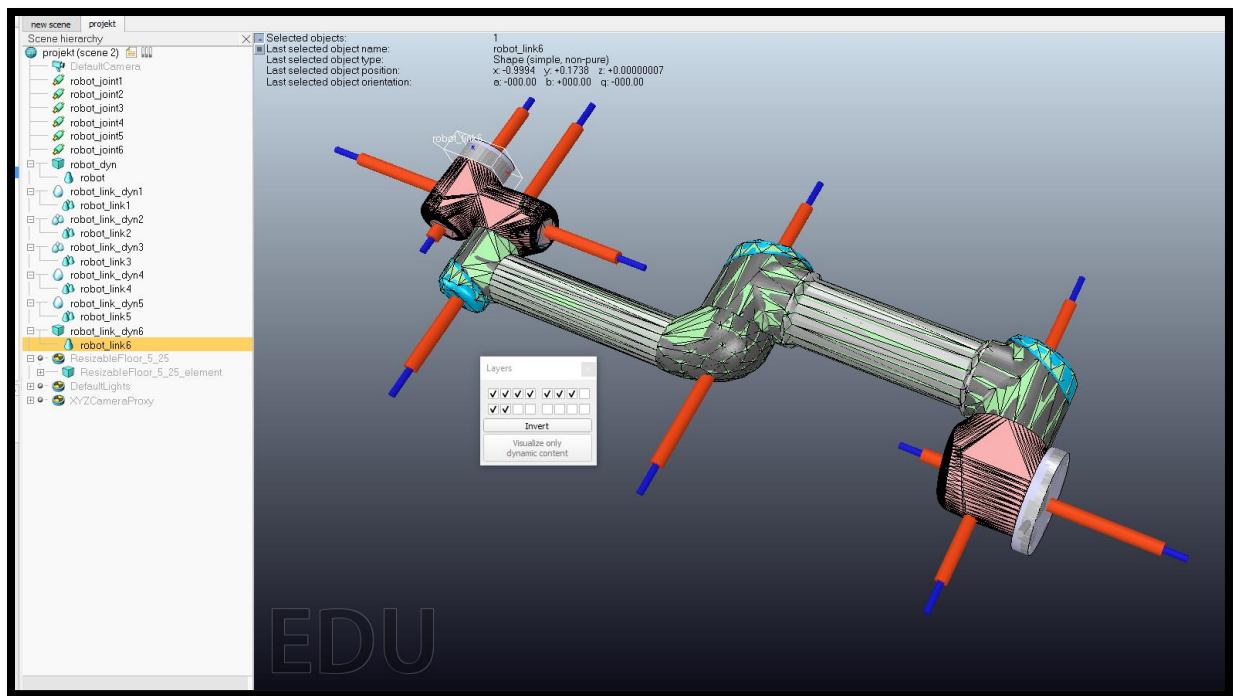
„Add -> Convex decomposition of selection”

ze względu na większe skomplikowanie kształtu oraz lepsze wyniki możliwe do osiągnięcia tą metodą. Ustawiono **Min. nb of clusters** na 1 aby uzyskać wierniejsze odwzorowanie kształtu.



Po ukończeniu tego procesu nadano hierarchię, gdzie element dynamiczny jest rodzicem. Po wszystkim wyłączono widoczność elementów dynamicznych.





We **właściwościach dynamicznych** (dwuklik na ikonę w drzewie hierarchii -> show dynamic properties dialog) dopiero stworzonych elementów ustawiono właściwość:

**„Body is responsible”**

oraz odpowiednio dostosowano lokalne maski:

**„Lokal responsible mask”**

Zaczynając od bazy – 0000 1111.

Natomiast element robot\_link\_dyn1 – 0000 1111. Po kolei na zmianę ustawiono maski wszystkich elementów dynamicznych. Naprzemiennosc na tym etapie jest istotna. W czasie projektowania robotów należy pamiętać, że pierwszy i ostatni człon powinny mieć taką samą maskę lokalną – 0000 1111.

Kolejno wszystkim elementom dynamicznym włączono cechę:

**„Body is dynamic”**

oraz używając funkcji:

**„Compute mass & inertia properties for selected convex shapes”**

automatycznie ustawiono im masę oraz momenty bezwładności.

Na koniec natomiast elementowi robot\_dyn zaznaczono opcję:

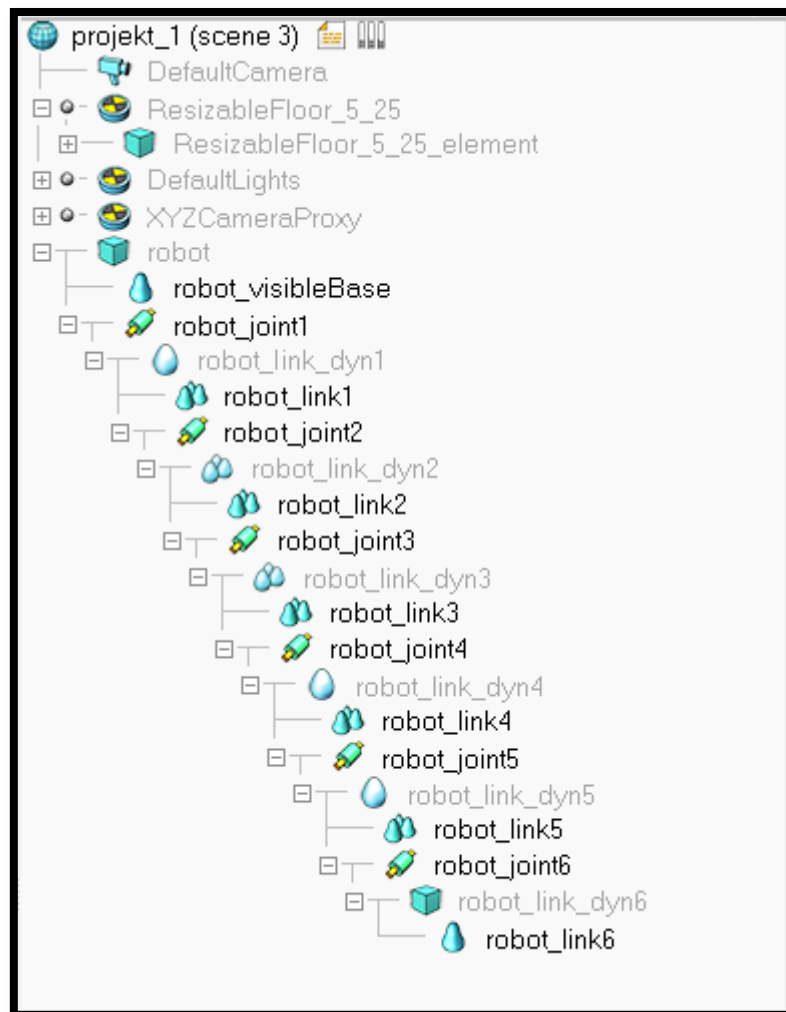
**„Set to dynamic if gets parent”**

oraz wyłączyła opcję:

**„Body is dynamic”**



Utworzono odpowiednią hierarchię elementów według której robot\_link\_dyn6 jest dzieckiem robot\_joint6 a robot\_joint6 jest dzieckiem robot\_link\_dyn5 i tak dalej.



Zmieniono nazwę bazy robota oraz jego elementu dynamicznego. Następnie dla obiektu robot w ustawieniach wspólnych (dwuklik na ikonę w hierarchii -> common) ustawiono właściwość:

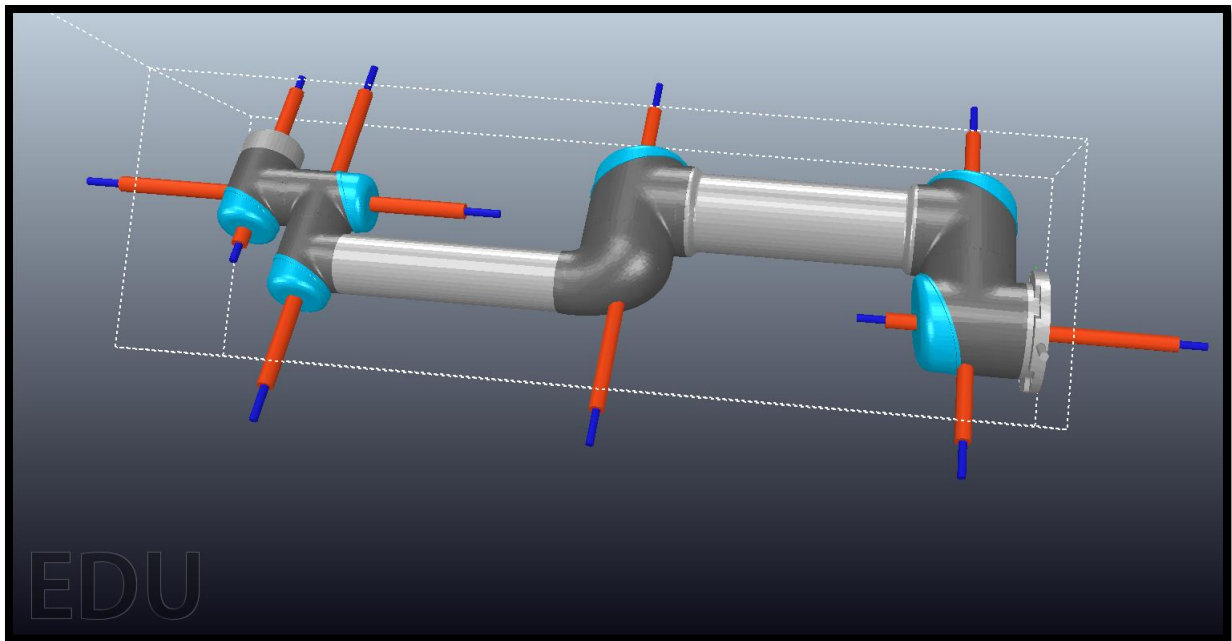
**„Object is model base ”** oraz

**„Object / model can transfer or akcept DNA”.**

Wszystkim wiązaniom natomiast zaznaczono opcję (również w ustawieniach common):

**„Ignored my model bounding box”.**





Następnie ustawiono robota w pozycję pionową. Kolejno wszystkim połączeniom dodano silnik oraz regulator PID  
 „dwuklik na ikonę -> Joint -> Show dynamic properties dialog”

Joint Dynamic Properties

Motor properties

☒ Motor enabled  
 Target velocity [deg/s]  
 Maximum torque [N\*m] 2.5000e+00  
☐ Lock motor when target velocity is zero  
 Edit engine specific properties

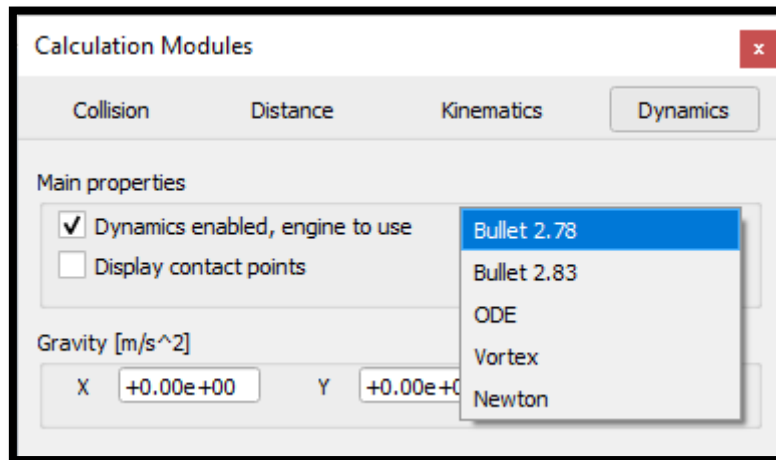
Apply to selection

Control properties

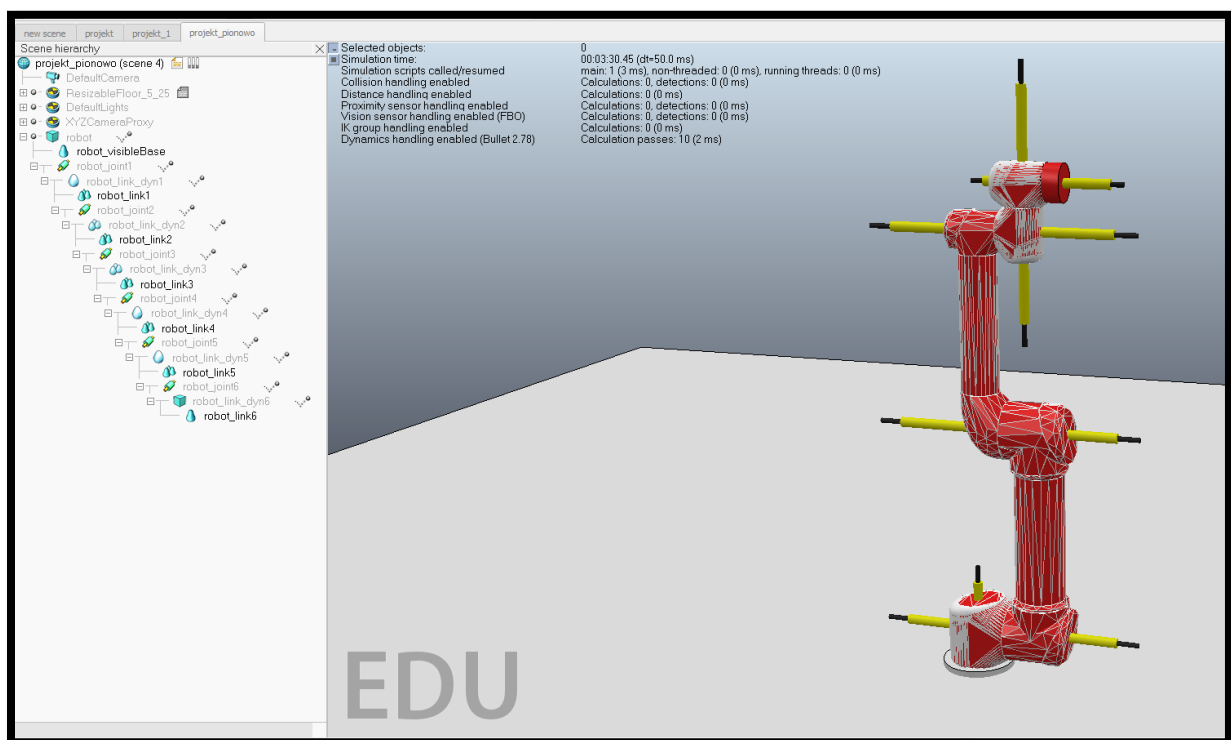
☒ Control loop enabled  
 Target position [deg] +0.0000e+00  
 Upper velocity limit [deg/s] 3.6000e+02  
☒ Position control (PID)  
   Proportional parameter 0.100  
   Integral parameter 0.000  
   Derivative parameter 0.000  
☐ Spring-damper mode  
   Spring constant K [N] 1.000e-01  
   Damping coefficient C [N\*s] 0.000e+00

Apply to selection

Przetestowano działanie na różnych silnikach fizycznych. Z użyciem wszystkich robot zachowywał sztywność i nie rozpadał się jak wcześniej.



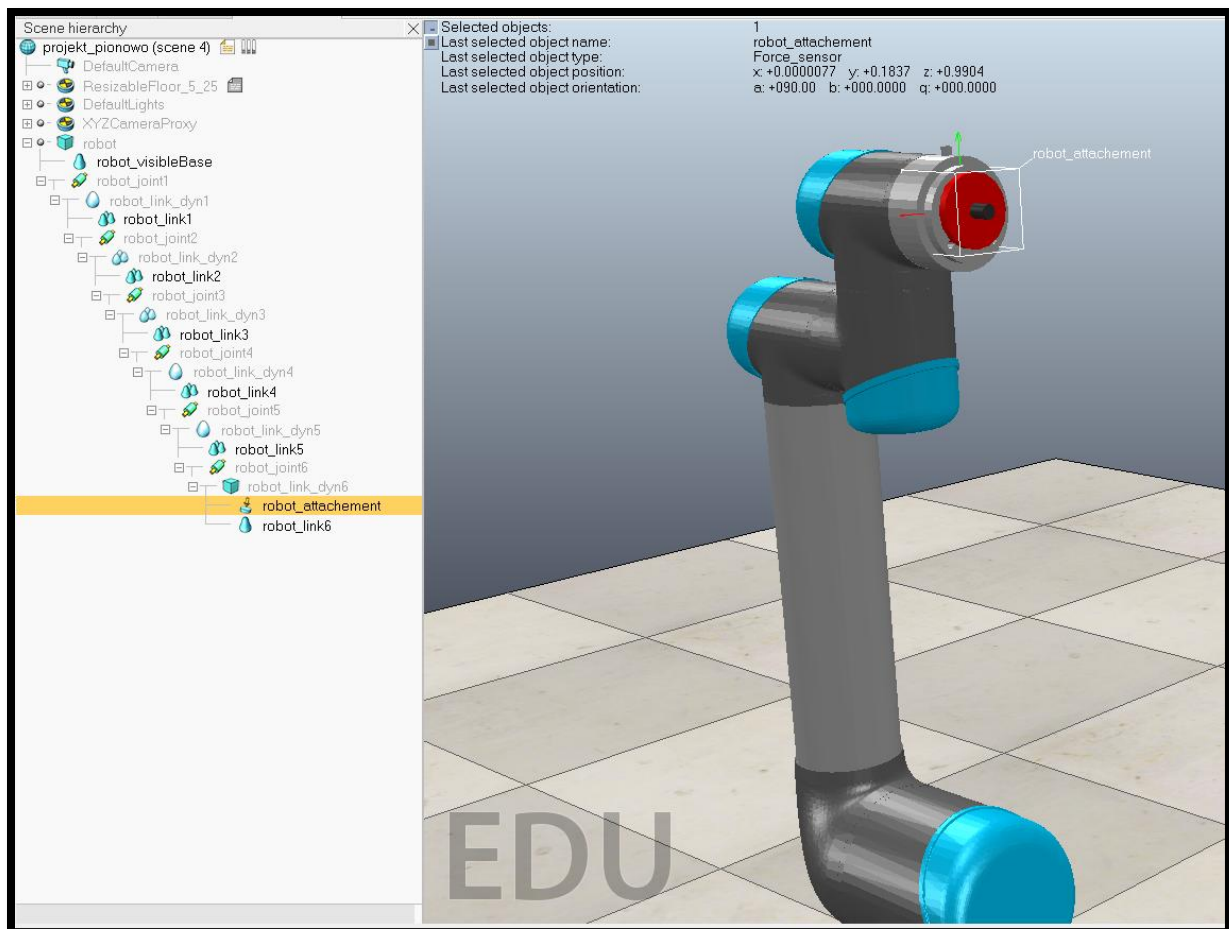
Uruchomiono symulację oraz włączono tryb (górne menu):  
„Visualize and verify dynamic content”.



Dalej dodano czujnik siły:

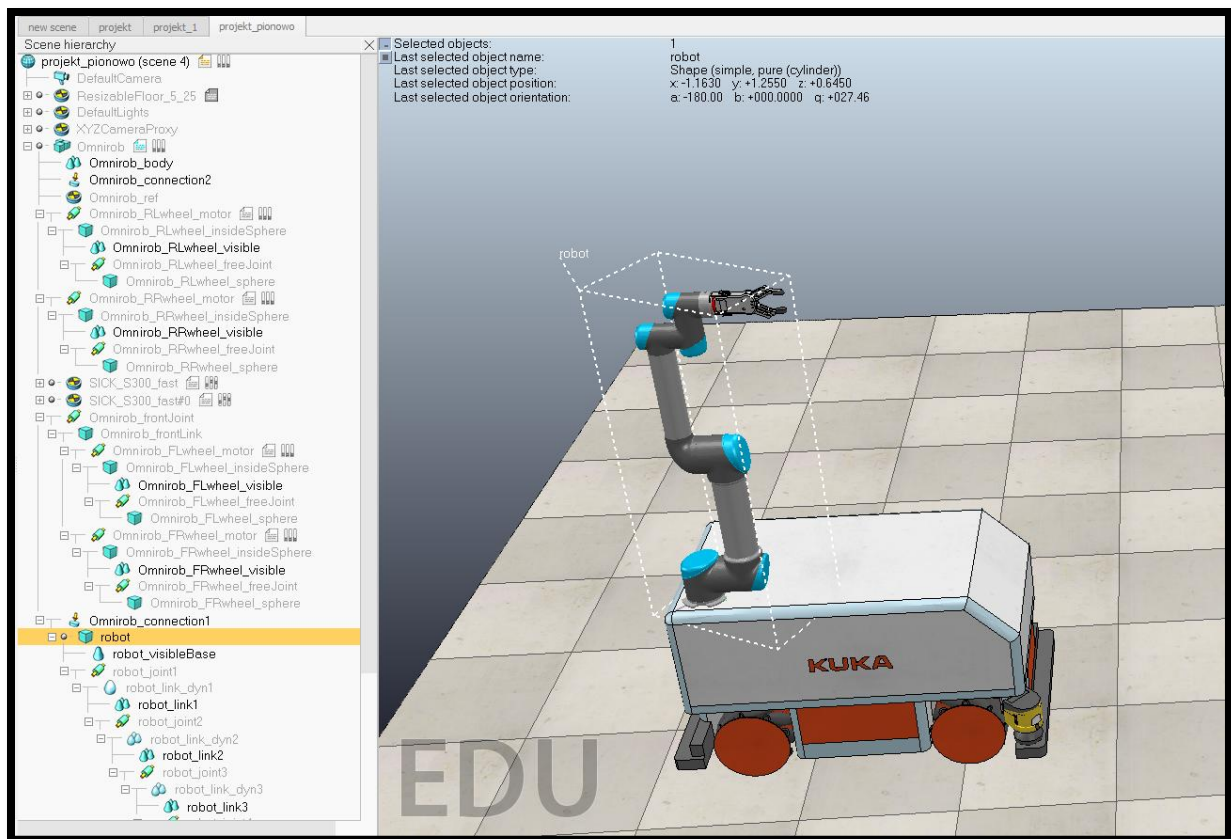
„Add -> Force sensor”

Wypozycjonowano go odpowiednio oraz ustawiono jako dziecko obiektu robot\_link\_dyn6. Nadano mu nazwę robot\_attachement.



Na koniec korzystając z gotowych modeli w programie dodano chwytak. Zaznaczając jednocześnie chwytak i czujnik oraz używając opcji (górne menu): **„Assemble / Disassemble”** automatycznie dopasowano i połączono chwytak z czujnikiem a co za tym idzie, z całym modelem.

Następnie w analogiczny sposób dodano robota mobilnego KUKA z dwoma czujnikami na grzbiecie. Używając tej samej funkcji przyczepiono ramię na dach robota mobilnego.



## Wnioski:

Poznano nową metodę modelowania robotów w programie V Rep – tworzenie „elementów dynamicznych” korzystając z trybu edycji trójkątów. Używamy ich jako uproszczenie modelu zbudowanego w programie CAD i to do nich dodaje się wszystkie właściwości dynamiczne które chcemy zawrzeć. Natomiast oryginalne części służą do wizualizacji całości złożenia.

Poznano użyteczny sposób łączenia elementów przy pomocy czujnika siły oraz opcji „**Assemble / Disassemble**”.