Roboty Humanoidalne grające w piłkę nożną

Grzegorz Ficht, doktorant na Uniwersytecie w Bonn

Autonomiczne Inteligentne Systemy
Uniwersytet w Bonn

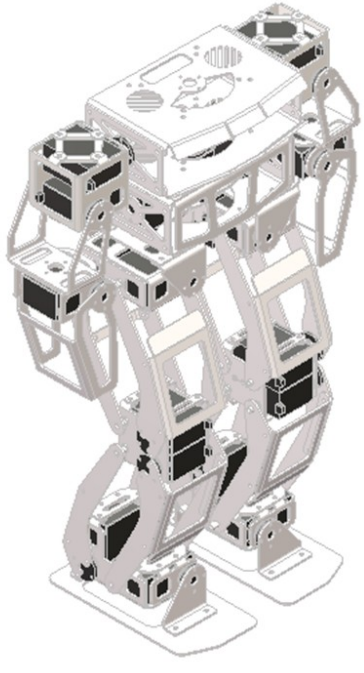
UNIVERSITÄT BONN

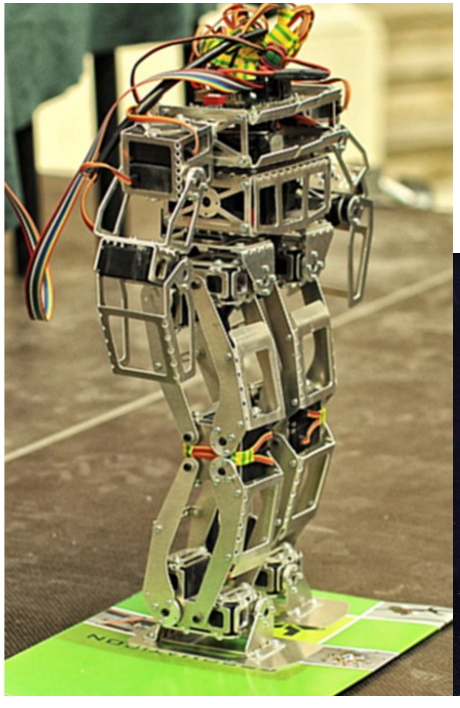
Plan prezentacji

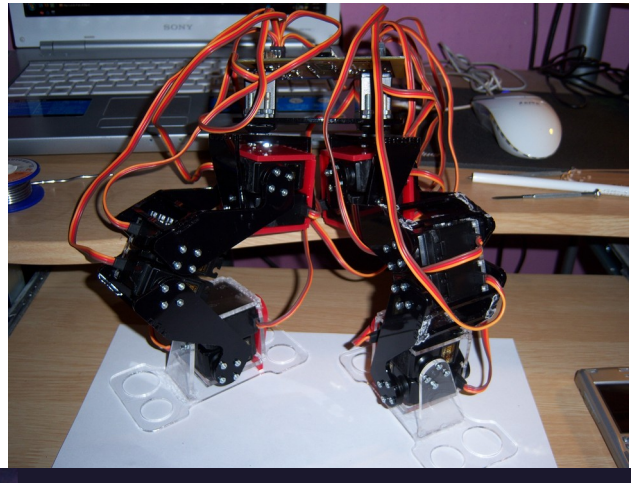
- Trochę kontekstu:
  - O mnie, mojej “katedrze”, grupie i pracy,
  - O zawodach RoboCup – cel, historia i stan aktualny,
  - Nasze występy w RoboCup – roboty i wyniki.
- Opensource’owe roboty NimbRo-OP2(X)
  - Design i Hardware,
  - Framework NimbRo:
    - Oparty na ROS
      - ROS w systemach embedded?
    - Kontrola niskopoziomowa, estymacja stanu, algorytm chodu, sprzężenia zwrotne, przetwarzanie obrazu, autolokalizacja, podejmowanie decyzji.
- Co dalej? Podsumowanie

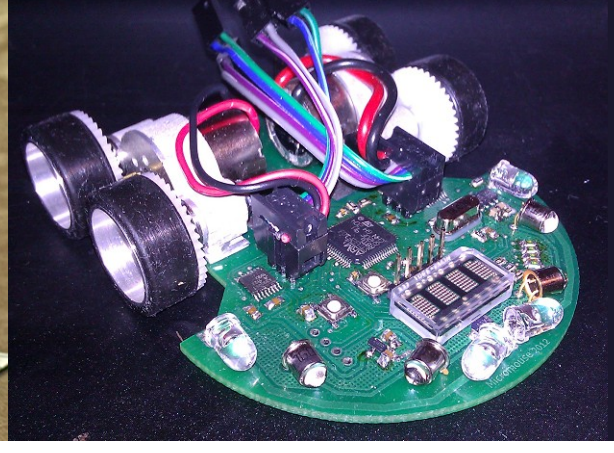
Trochę kontekstu – o mnie

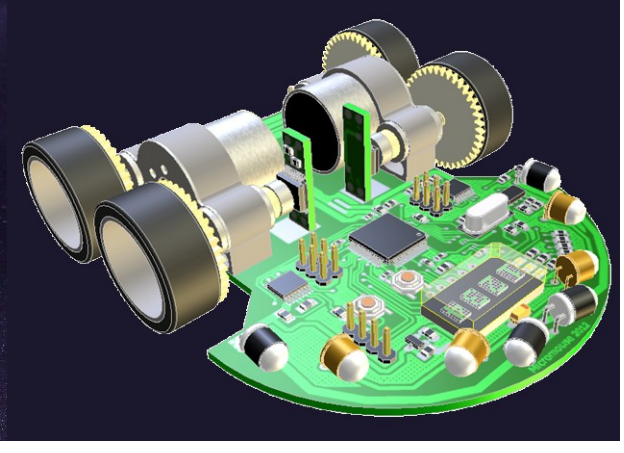
- Robotami interesuję się od Technikum – ROBO-ONE & OmniZero.2
- Studiowałem Automatykę i Robotykę na EiA, PG
  - Aktywność w kole naukowym
  - Praca inżynierska – robot humanoidalny
  - Praca magisterska – robot micromouse
  - Wielokrotne wyjazdy na zawody
  - 2 x złota odznaka absolwenta PG







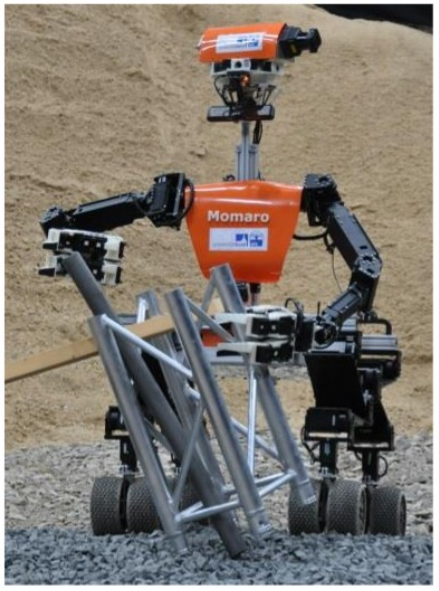




Trochę kontekstu – o katedrze

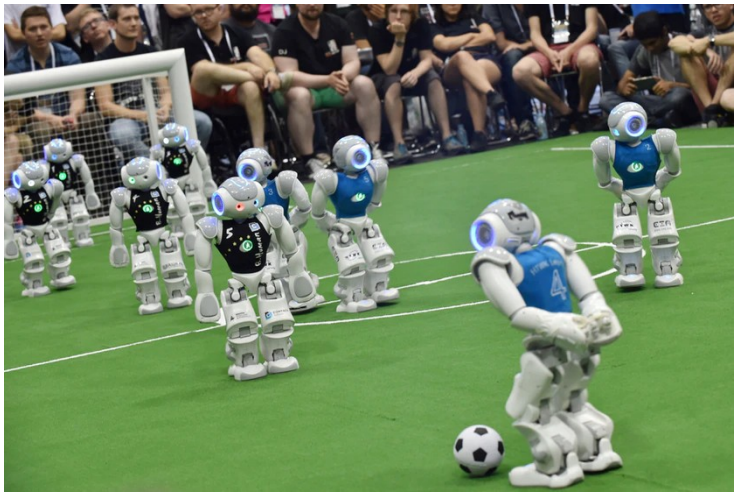
- Grupa Autonomicznych Inteligentnych Systemów
  - Nacisk na: inteligencję i kognitywistykę
  - Wykorzystanie robotów w różnych środowiskach
    - Piłka nożna, usługi, sortowanie/automatyzacja, ratunek/teleoperacja, inwentaryzacja, inspekcja
- Sukcesy w międzynarodowych zawodach:
  - DRC, MBZIRC, APC/ARC, RoboCup, SpaceBot





Trochę kontekstu – cel RoboCup

- Cel RoboCup: “W połowie XXI wieku, drużyna w pełni samodzielnych, humanoidalnych robotów wygra mecz w piłkę nożną, zgodnie ze wszystkimi zasadami FIFA, grając przeciwko aktualnej drużynie mistrzowskiej.”
- Idea: Stworzenie autonomicznie działających robotów, z zamiarem promocji prowadzenia badań naukowych w zakresie sztucznej inteligencji i robotyki.



Trochę kontekstu – historia RoboCup

- Coroczne zawody robotów, od 1997
- Geneza RoboCup – szachy?
- Skala problemu
- Początkowo – symulacja i small size league
- Humanoid League – od 2002
  - Bramki niesymetryczne, pomarańczowa piłka z plastiku
  - Gry rozgrywane na płaskiej powierzchni
  - Linie idealnie widoczne
  - Brak wymagań odnośnie wielkości robotów
  - Znaczne spłycenie zasad rozgrywki

Trochę kontekstu – stan RoboCup

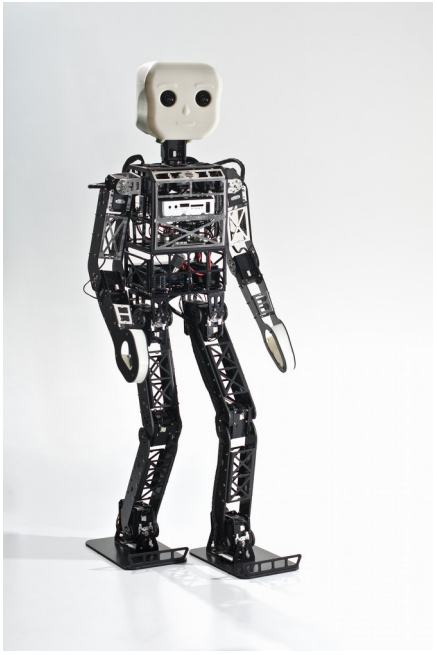
- Stan Humanoid League z RoboCup 2019:
  - Klasy rozmiarowe (KidSize, TeenSize, AdultSize)
  - Bazowanie na zasadach FIFA
  - Całkowita symetria boiska
  - Granie aktualną piłką FIFA
  - Sztuczna murawa
  - Linie boiska się wycierają, krawędzie nie są ostre
  - Kształt robotów oraz ich wyposażenie musi być analogiczne do człowieka
  - Autonomiczne gry (game controller)
  - Ludzie w polu tylko w AdultSize

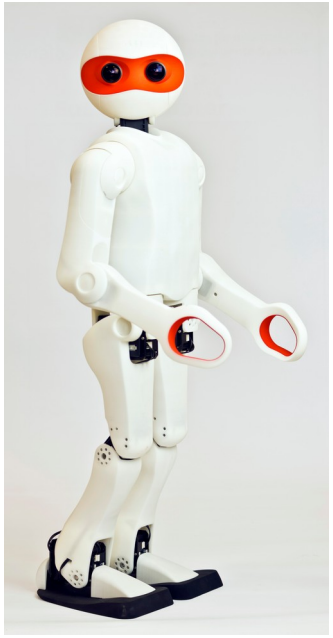
Trochę kontekstu – NimbRo Soccer

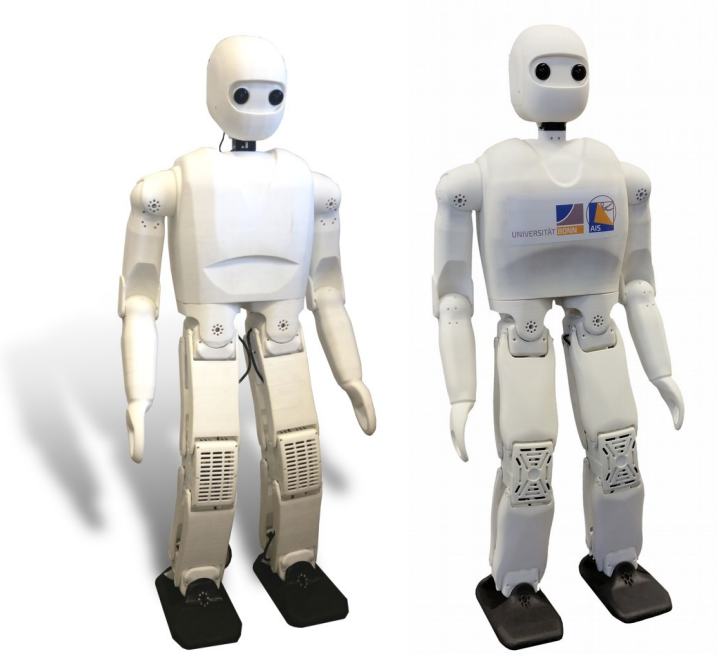
- Projekt NimbRo rozpoczął się na uniwersytecie w Freiburgu ~2004 roku, od 2008 jest w Bonn.
- Zdobyte mistrzostwa w: 2009–2013, 2016–2019, w tym dwa podwójne.
- 4x Best Humanoid Award (2x Louis Vuitton Cup), 2x Design Award, Open-Source Award.

Trochę kontekstu – NimbRo Soccer

- Do ~2015 roboty operowały na pocket PC & Windows (Dynaped, Copedo).
- 2012 – początki wdrażania ROS (NimbRo-OP, igus-OP)
- 2015 – wyłącznie ROS
- 2016 – konwersja Dynapeda
- 2017 – NimbRo-OP2 & konwersja Copedo
- 2018 – NimbRo-OP2X
- 4 opensource'owe roboty:
  - NimbRo-OP, igus Humanoid Open Platform, NimbRo-OP2(X)



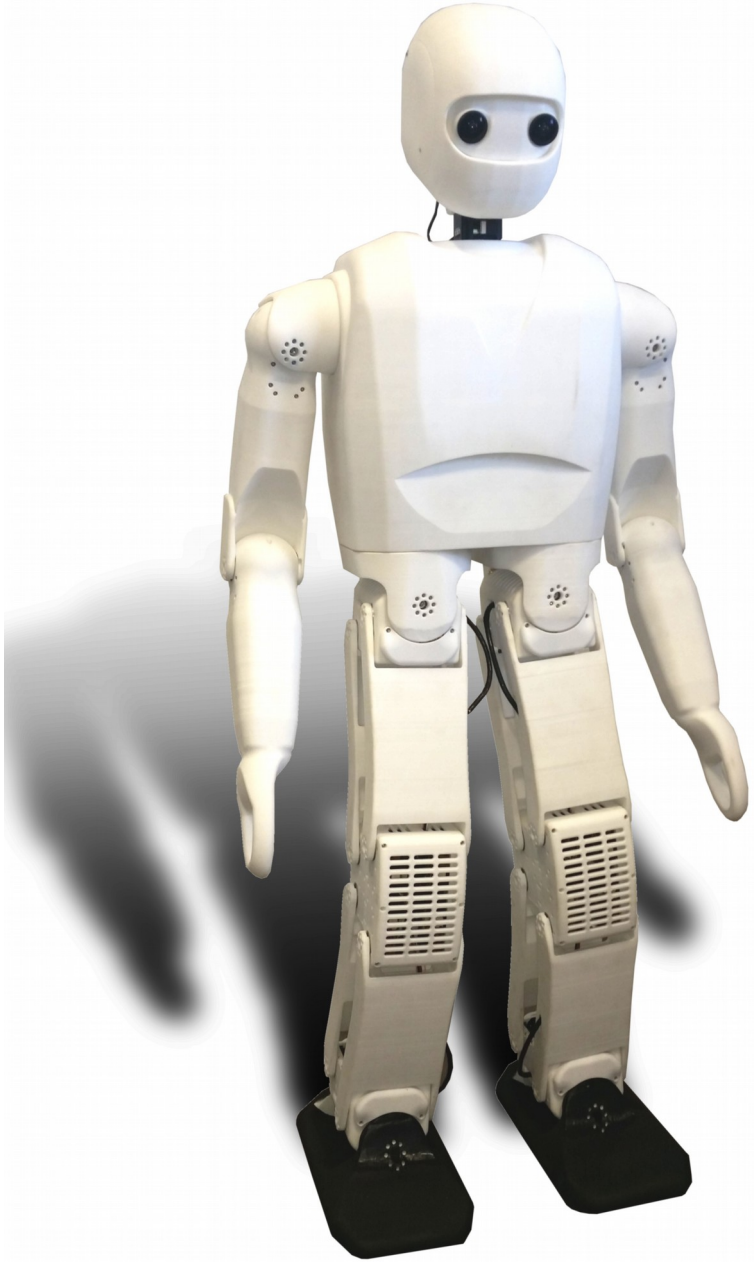




http://nimbro.net/OP/

Roboty NimbRo-OP2(X)

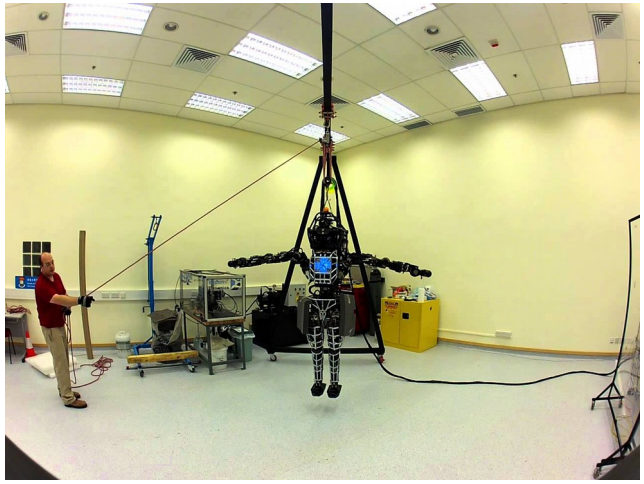
NimbRo-OP2



NimbRo-OP2X

Roboty NimbRo-OP2(X) - design

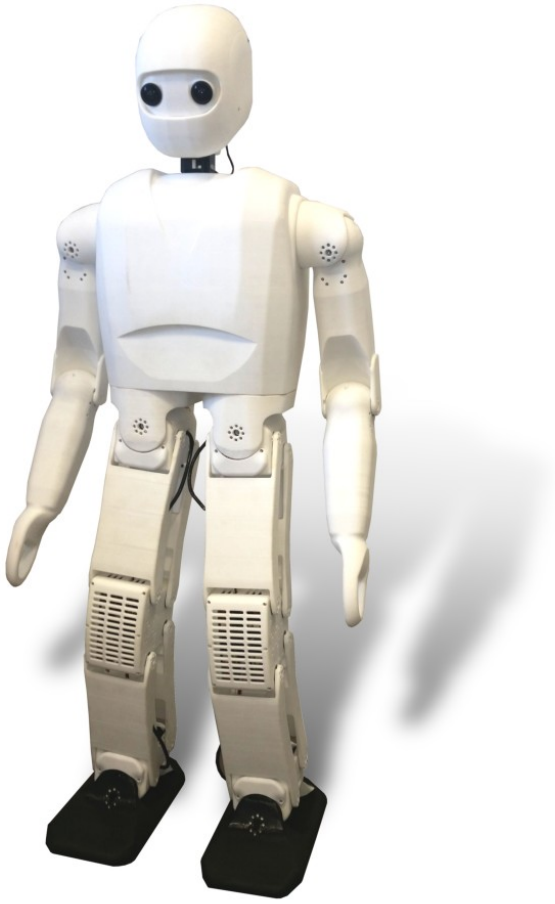
- Motywacja:
  - Ograniczone zastosowanie mniejszych robotów
  - Większe roboty są drogie, trudne w utrzymaniu i operowaniu, czasem niebezpieczne
- Porządane parametry:
  - Stosunkowo tani, w pełni otwarty
  - 130-140cm wysokości
  - Minimalna ilość części
  - Łatwy w montażu
  - Bezproblemowy w utrzymaniu
  - Możliwy do rozbudowy
  - Przyjazny, estetyczny wygląd



Roboty NimbRo-OP2(X) – HW OP2

NimbRo-OP2

Type	Specification	Value
General	Height & Weight	1345 mm, 17.5 kg
	Battery	4-cell LiPo (14.8 V, 6.6 Ah)
	Battery Life	15–30 min
	Material	Polyamide 12 (PA12)
CM730	Microcontroller	STM32F103RE (Cortex M3)
	Memory	512 KB Flash, 64 KB SRAM
	Other	3 × Buttons, 7 × Status LEDs
Actuators	Stall Torque	10.0 Nm
	No load speed	55 rpm
	Total	34 × MX-106R
	Head	2 × MX-106R
	Each Arm	3 × MX-106R
	Each Leg	13 × MX-106R
Sensors	Encoders	4096 ticks/rev
	Gyroscope	3-axis (L3G4200D chip)
	Accelerometer	3-axis (LIS331DLH chip)
	Camera	Logitech C905 (720p)
	Camera Lens	Wide-angle lens with 150° FOV



Roboty NimbRo-OP2(X) – HW OP2X

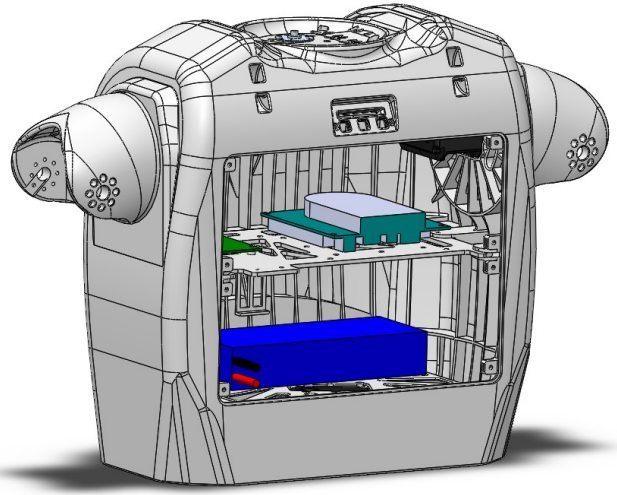
NimbRo-OP2X

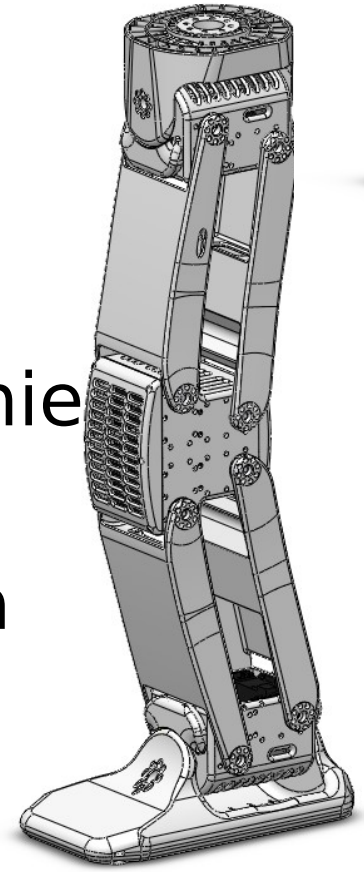
Type	Specification	Value
General	Height & Weight	135 cm, 19 kg
	Battery	4-cell LiPo (14.8 V, 8.0 Ah)
	Battery life	20–40 min
	Material	Polyamide 12 (PA12)
PC	Mainboard	Z370 Chipset, Mini-ITX
	CPU	Intel Core i7-8700T, 2.7–4.0 GHz
	GPU	GTX 1050 Ti, 768 CUDA Cores
	Memory	4 GB DDR4 RAM, 120 GB SSD
	Network	Ethernet, Wi-Fi, Bluetooth
	Other	8 × USB 3.1, 2 × HDMI, DisplayPort
Actuators	Total	34 × Robotis XH540-W270-R
	Stall torque	12.9 Nm
	No load speed	37 rpm
	Control mode	Torque, Velocity, Position, Multi-turn
Sensors	Encoders	12 bit/rev
	Joint current (torque)	12 bit
	Gyroscope	3-axis (L3G4200D chip)
	Accelerometer	3-axis (LIS331DLH chip)
	Camera	Logitech C905 (720p)
	Camera lens	Wide-angle lens with 150° FOV

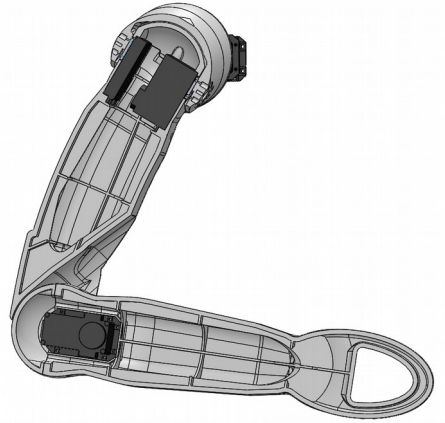
NimbRo-OP2(X) – mechanika

Elementy z druku 3D:

- Funkcjonalne i dla wyglądu
- Selective laser sintering, warstwy 0.1mm
- Żebra obniżają wagę, utrzymując sztywność
- Wielokrotnie wykorzystanie elementów
- Wbudowane “ścieżki” dla przewodów

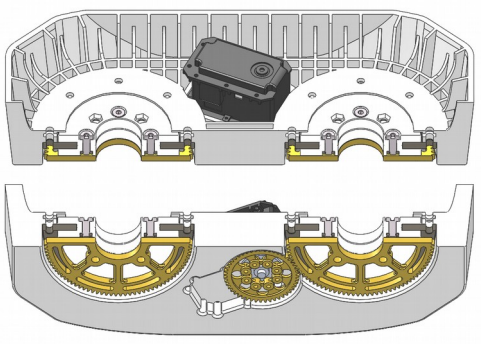


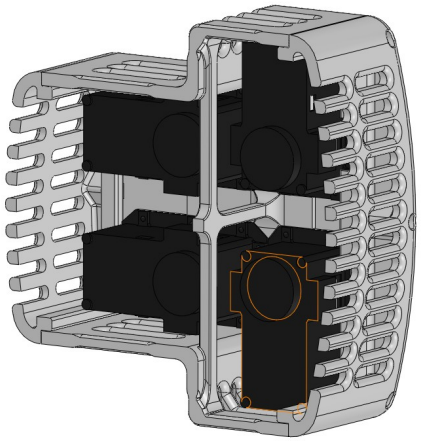


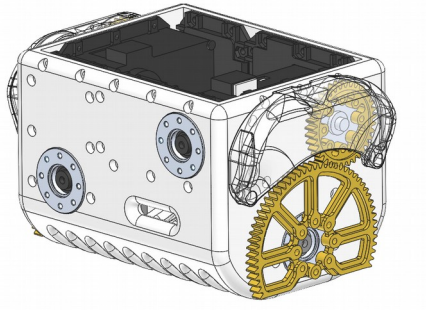


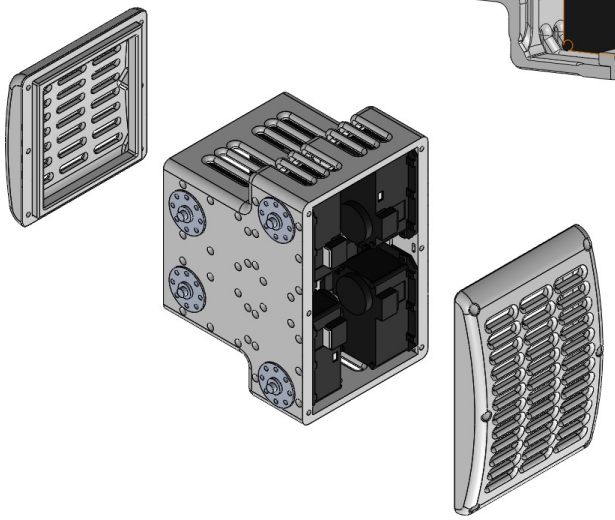
NimbRo-OP2(X) – mechanika

- Kinematyka równoległa, synchronizacja napędów (PWM)
- Zewnętrzne przekładnie:
  - w kostce i biodrze
  - z mosiądzu lub druk 3D
- Łatwy dostęp (naprawy)
- Osłony zębatek









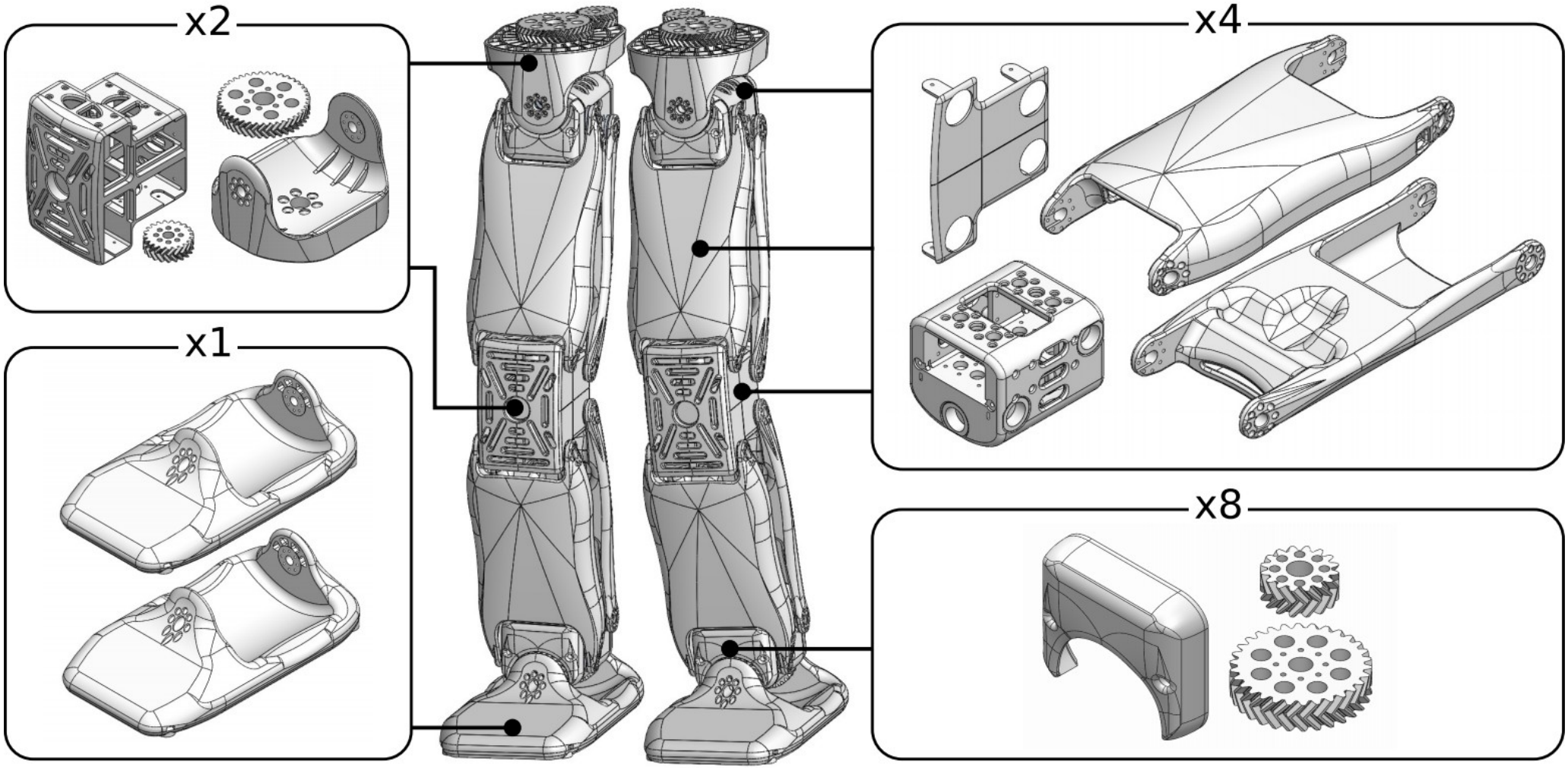
- Całkowicie open-source:
  https://github.com/NimbRo/nimbro-op2

NimbRo-OP2(X) – mechanika

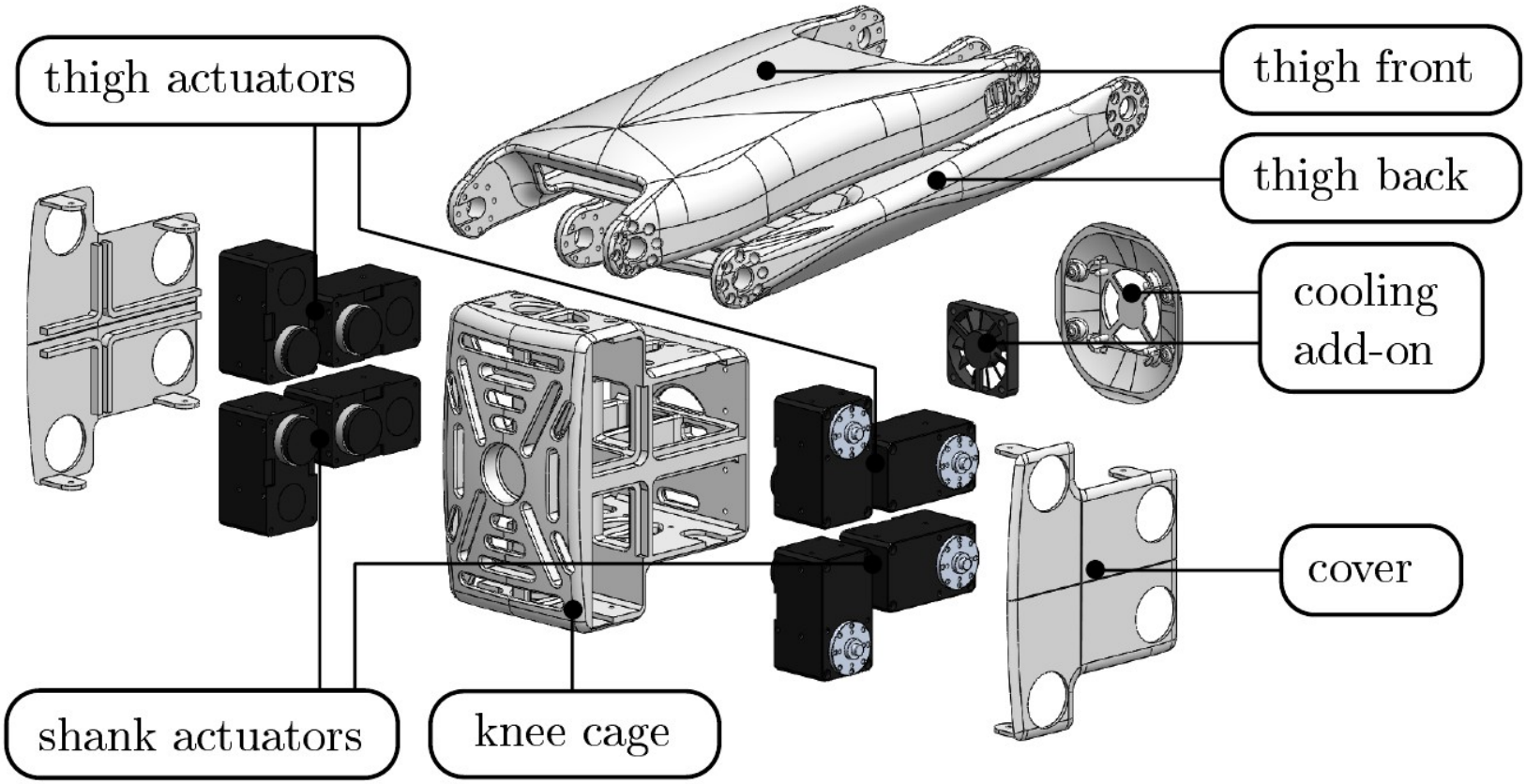
Redesign w OP2X:

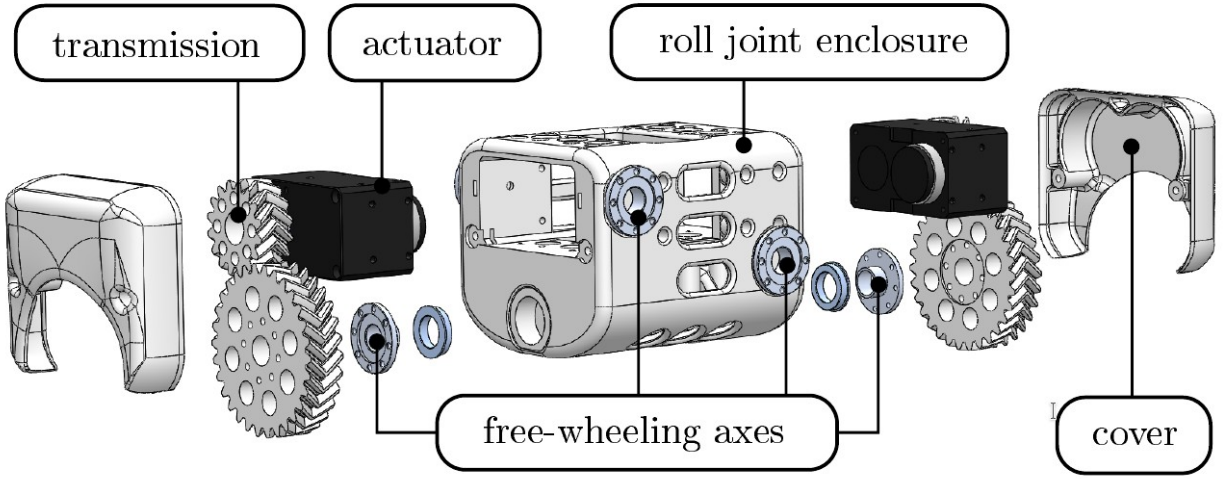
- Nowe napędy: XH540-W270-R
- Obniżenie wagi szkieletu
- Miejsce na MiniITX i GPU
- Nowe przekładnie zewnętrzne:
  - Podwójna helikoida
  - Wyłącznie druk 3D
- Przewidziane chłodzenie
- Dopracowanie szczegółów

NimbRo-OP2(X) – mechanika

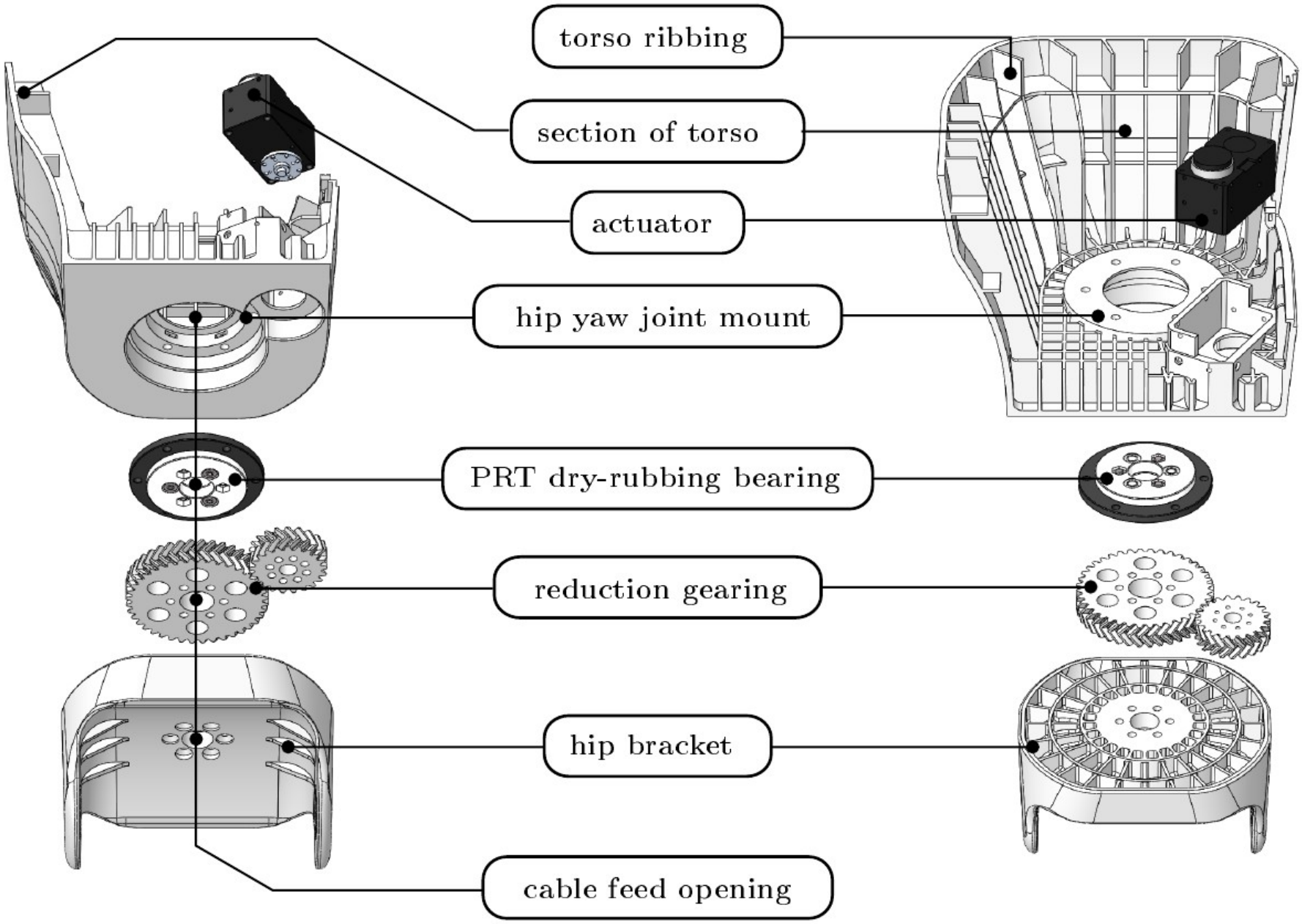


NimbRo-OP2(X) – mechanika

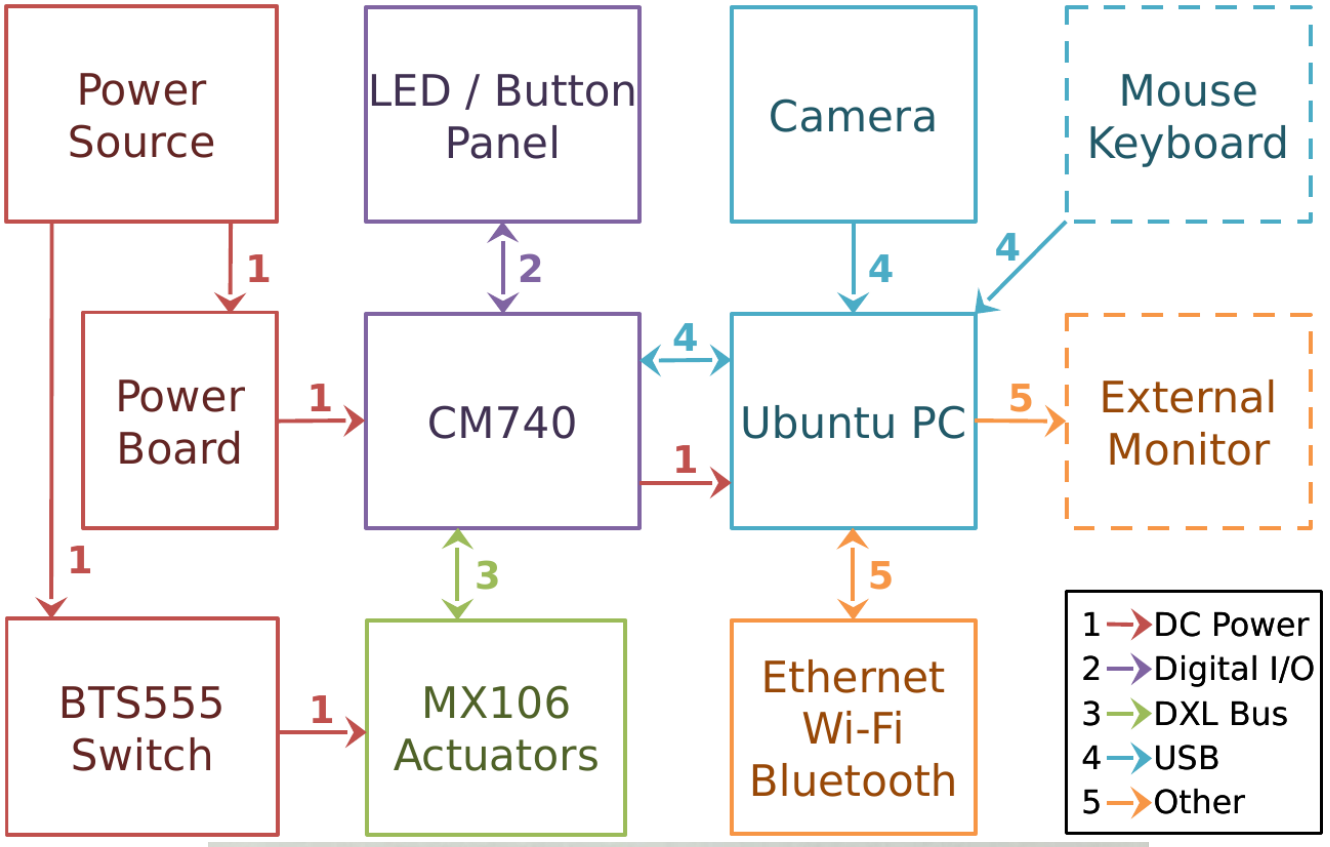


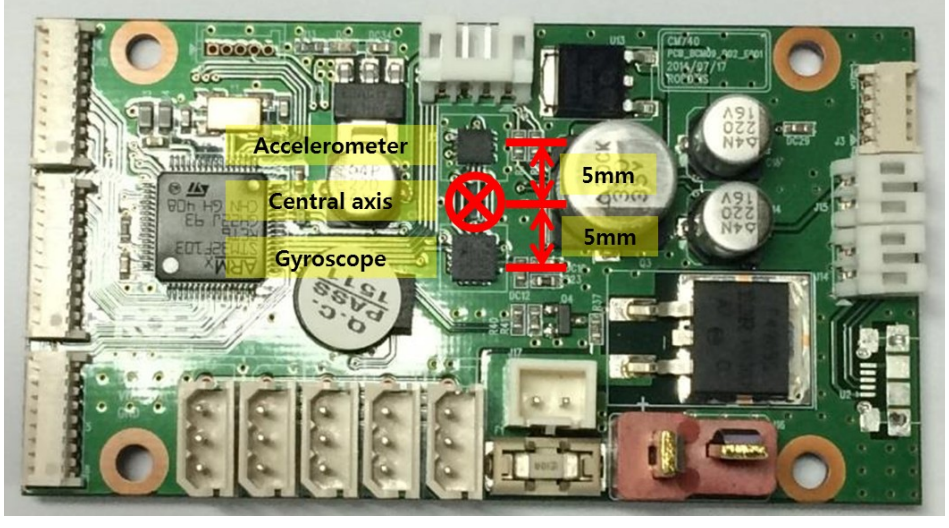


NimbRo-OP2(X) – mechanika



Roboty NimbRo-OP2(X) – Elektronika



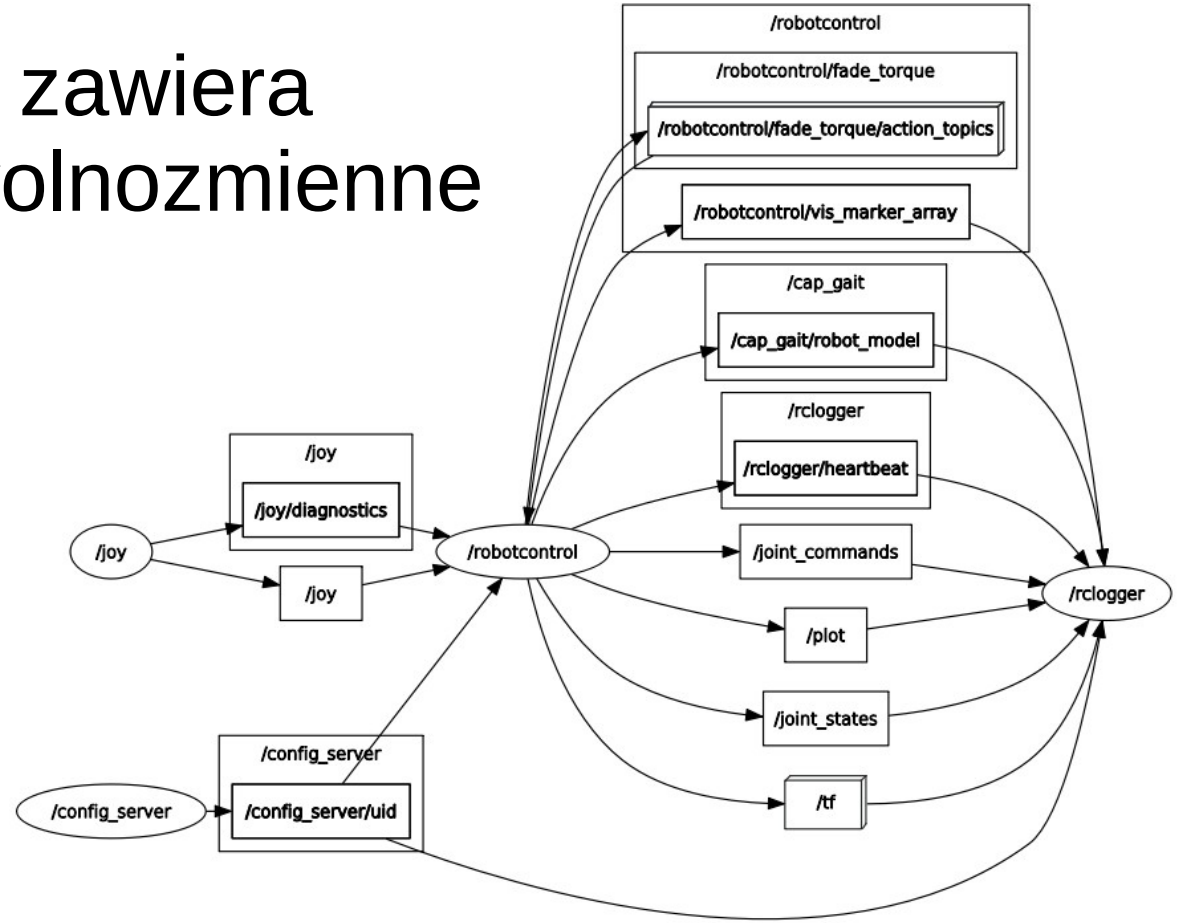


ROS, czyli Robot Operating System

- Wbrew nazewnictwu – ROS nie jest systemem operacyjnym
- Zbiór oprogramowania tworzący “ekosystem”, zapewniający: abstrakcję sprzętową, kontrolę niskopoziomową, podstawową funkcjonalność, komunikację między procesami oraz system zarządzania pakietami
- Znaczna większość oprogramowania jest open-source, na różnych licencjach.
- Oficjalnie wspierany jest tylko Ubuntu, inne dystrybucje linuxa, Windows czy macOS uznawane są jako experymentalne.
- Brak natywnego wsparcia RTOS

ROS – model działania

- Grupa węzłów (nodes) komunikujących się po magistralach zwanych tematami (topics)
- Węzły moga udostępniać usługi – jednorazowe operacje o z góry określonym działaniu
- Serwer parametrów – zawiera dane statyczne, lub wolnozmienne

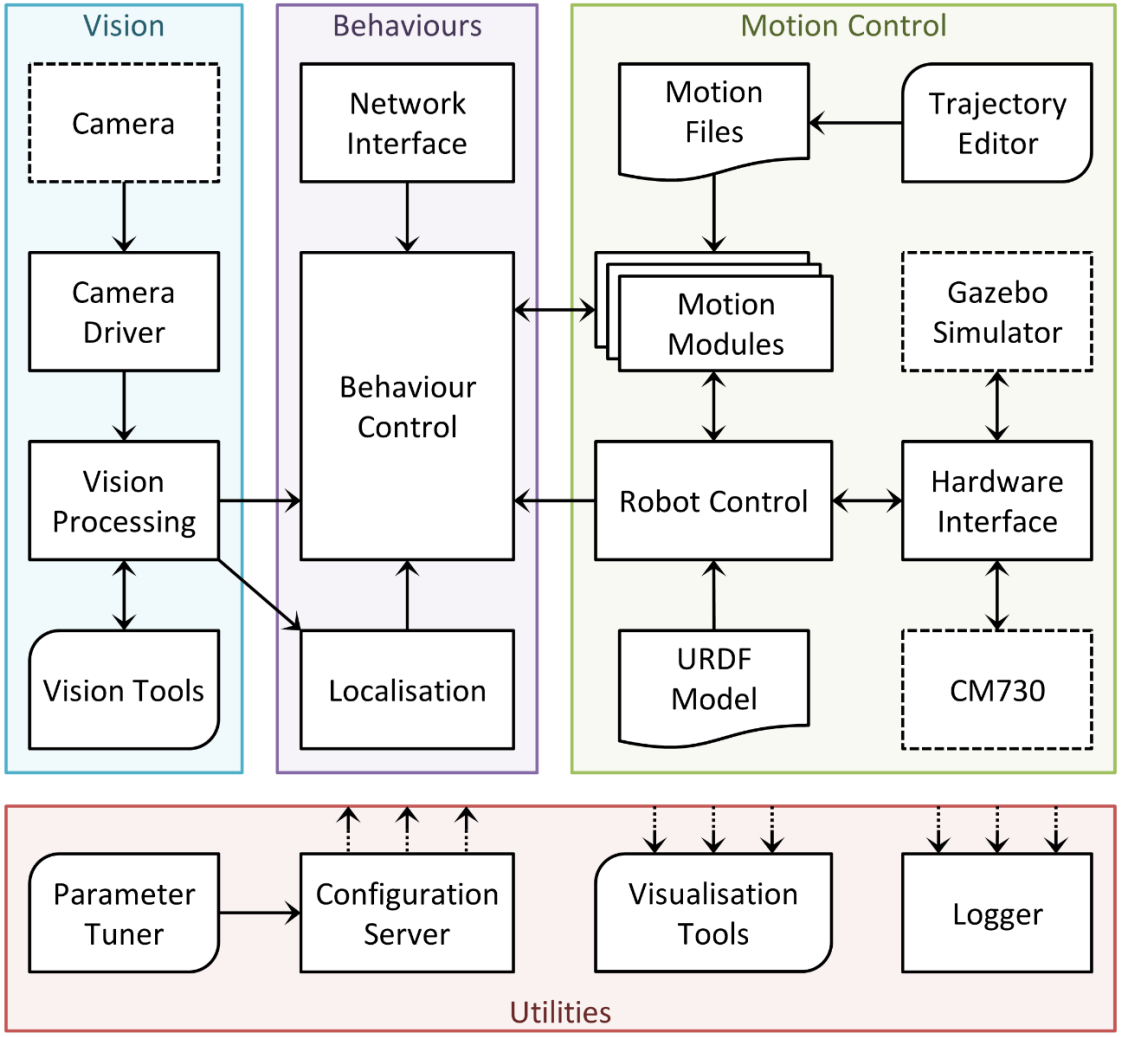


ROS – w systemach embedded

- Rosserial – uniwersalne, lecz ograniczone:
  - Wparcie dla arduino, mbed, stm32, część TI
  - Niska przepustowość danych (baudrate, rozmiar wiadomości)
- Lepsze rozwiązania:
  - Dedykowany osprzęt z USB + node z abstrakcją HW
  - SBC(np. RPI) z modułami rozszerzeń + zewnętrzna komunikacja.

Roboty NimbRo-OP2(X) - Framework

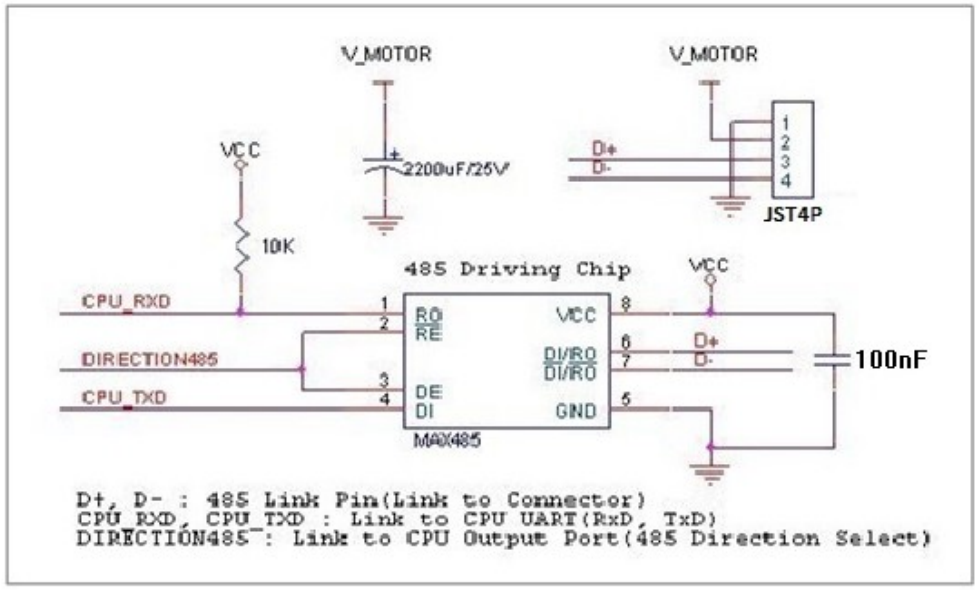
- Współdzielony przez wiele różnych robotów
- Zawiera:
  - Kontrola ruchu
  - Zachowania
  - System wizji
  - Narzędzia

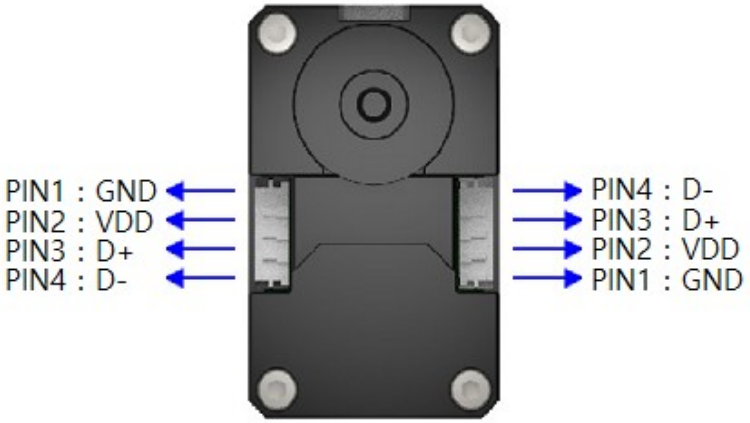


- W pełni open-source:
  - https://github.com/AIS-Bonn/humanoid_op_ros/

Framework – sterowanie napędami

- Napędy Robotis Dynamixel
- Wbudowany reg. PID



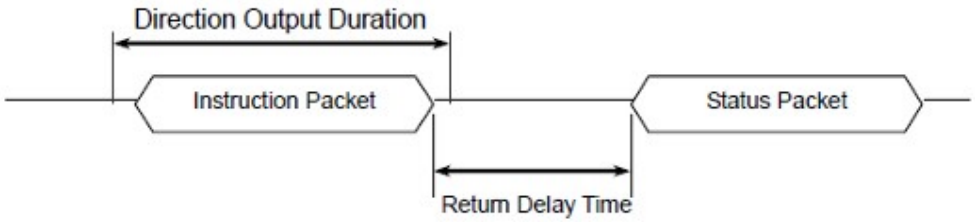


Data

	Unit	data
Dimension	mm (in)	33.5 x 58.5 x 44 (1.32 x 2.30 x 1.73)
Weight	g (oz)	160(5.64)
Operating Voltage	V	12
Gear Ratio	-	272.5 : 1
Stall Torque	N.m (oz.in)	9.9(1402)
Stall Current	A	4.9
No Load Speed	RPM	39
Position Sensor	-	Contactless Absolute Encoder (AMS)
Motor	-	Coreless Motor(Maxon)
Operating Range	°	360
Resolution	steps/turn	4096
Network Interface	-	TTL/RS-485
Material	Casing	Front / Middle : Aluminum Back : Enpla
	Gear	Metal

Framework – sterowanie napędami

- Częstotliwość komunikacji – 100Hz
- Protokół DXL 1.0:
  - Half duplex
  - Pakiet danych:
  - Odpowiedź:
  - Instrukcje:



Header1	Header2	ID	Length	Instruction	Param 1	...	Param N	Checksum	
0xFF	0xFF	ID	Length	Instruction	Param 1	...	Param N	CHKSUM	

Header1	Header2	ID	Length	Error	Param 1	...	Param N	Checksum	
0xFF	0xFF	ID	Length	Error	Param 1	...	Param N	CHKSUM	

Value	Instructions	Description
0x01	Ping	Instruction that checks whether the Packet has arrived to a device with the same ID as Packet ID
0x02	Read	Instruction to read data from the Device
0x03	Write	Instruction to write data on the Device
0x04	Reg Write	Instruction that registers the Instruction Packet to a standby status; Packet is later executed through the Action instruction
0x05	Action	Instruction that executes the Packet that was registered beforehand using Reg Write
0x06	Factory Reset	Instruction that resets the Control Table to its initial factory default settings
0x08	Reboot	Instruction that reboots DYNAMIXEL(See applied products in the description)
0x83	Sync Write	For multiple devices, Instruction to write data on the same Address with the same length at once
0x92	Bulk Read	For multiple devices, Instruction to write data on different Addresses with different lengths at once This command can only be used with MX series.

Framework – sterowanie napędami

- Lista rejestrów XH540

2. 2. Control Table of EEPROM Area

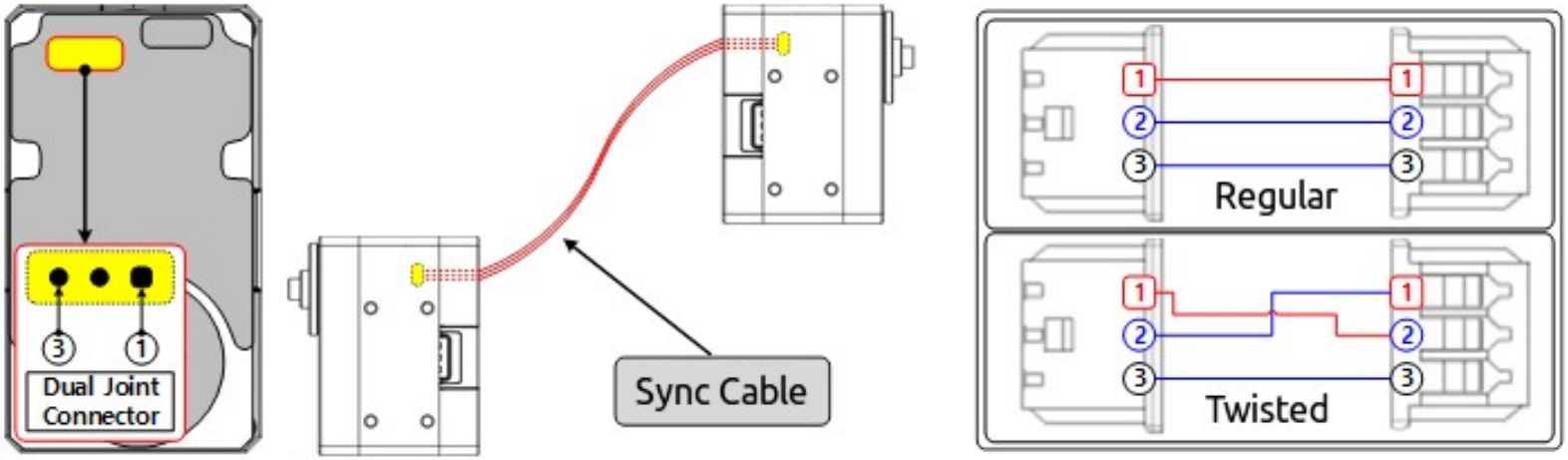
Address	Size (Byte)	Data Name	Access	Default Value	Range	Unit
0	2	Model Number	R	1,100	-	-
2	4	Model Information	R	-	-	-
6	1	Firmware Version	R	-	-	-
7	1	ID	RW	1	0 ~ 253	-
8	1	Baud Rate	RW	1	0 ~ 7	-
9	1	Return Delay Time	RW	250	0 ~ 254	2 [µsec]
10	1	Drive Mode	RW	0	0 ~ 2	-
11	1	Operating Mode	RW	3	0 ~ 16	-
12	1	Secondary(Shadow) ID	RW	255	0 ~ 252	-
13	1	Protocol Type	RW	2	1 ~ 2	-
20	4	Homing Offset	RW	0	-1,044,479 ~ 1,044,479	1 [pulse]
24	4	Moving Threshold	RW	10	0 ~ 1,023	0.229 [rev/min]
31	1	Temperature Limit	RW	80	0 ~ 100	1 [°C]
32	2	Max Voltage Limit	RW	160	95 ~ 160	0.1 [V]
34	2	Min Voltage Limit	RW	95	95 ~ 160	0.1 [V]
36	2	PWM Limit	RW	885	0 ~ 885	0.113 [%]
38	2	Current Limit	RW	2,047	0 ~ 2,047	2.69 [mA]
44	4	Velocity Limit	RW	167	0 ~ 1023	0.229 [rev/min]
48	4	Max Position Limit	RW	4,095	0 ~ 4,095	1 [pulse]
52	4	Min Position Limit	RW	0	0 ~ 4,095	1 [pulse]
56	1	External Port Mode 1	RW	3	0 ~ 3	-
57	1	External Port Mode 2	RW	3	0 ~ 3	-
58	1	External Port Mode 3	RW	3	0 ~ 3	-
63	1	Shutdown	RW	52	-	-

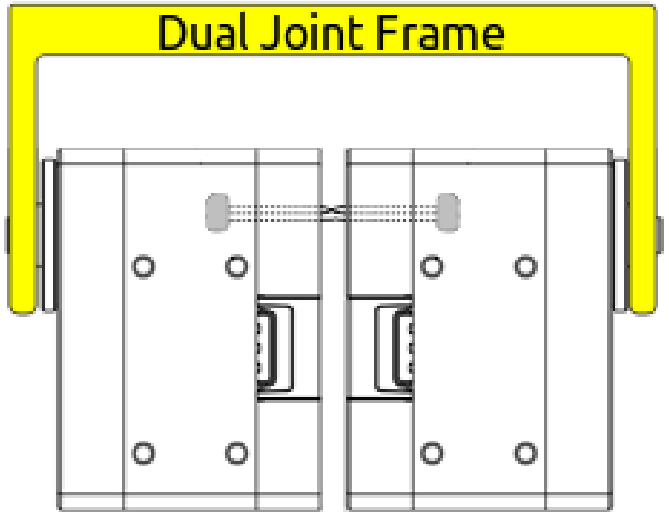
2. 3. Control Table of RAM Area

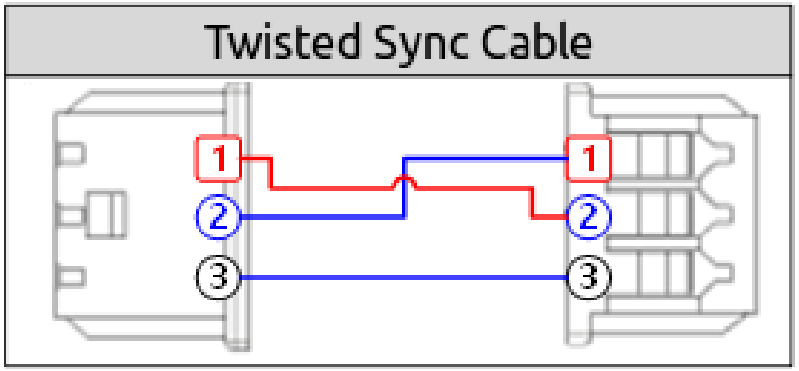
Address	Size (Byte)	Data Name	Access	Default Value	Range	Unit
64	1	Torque Enable	RW	0	0 ~ 1	-
65	1	LED	RW	0	0 ~ 1	-
68	1	Status Return Level	RW	2	0 ~ 2	-
69	1	Registered Instruction	R	0	0 ~ 1	-
70	1	Hardware Error Status	R	0	-	-
76	2	Velocity I Gain	RW	1,920	0 ~ 16,383	-
78	2	Velocity P Gain	RW	100	0 ~ 16,383	-
80	2	Position D Gain	RW	0	0 ~ 16,383	-
82	2	Position I Gain	RW	0	0 ~ 16,383	-
84	2	Position P Gain	RW	800	0 ~ 16,383	-
88	2	Feedforward 2nd Gain	RW	0	0 ~ 16,383	-
90	2	Feedforward 1st Gain	RW	0	0 ~ 16,383	-
98	1	Bus Watchdog	RW	0	1 ~ 127	20 [msec]
100	2	Goal PWM	RW	-	-PWM Limit(36) ~ PWM Limit(36)	-
102	2	Goal Current	RW	-	-Current Limit(38) ~ Current Limit(38)	2.69 [mA]
104	4	Goal Velocity	RW	-	-Velocity Limit(44) ~ Velocity Limit(44)	0.229 [rev/min]
108	4	Profile Acceleration	RW	0	0 ~ 32,767 0 ~ 32,737	214.577 [rev/min^2] 1 [ms]
112	4	Profile Velocity	RW	0	0 ~ 32,767	0.229 [rev/min]
116	4	Goal Position	RW	-	Min Position Limit(52) ~ Max Position Limit(48)	1 [pulse]
120	2	Realtime Tick	R	-	0 ~ 32,767	1 [msec]
122	1	Moving	R	0	0 ~ 1	-
123	1	Moving Status	R	0	-	-
124	2	Present PWM	R	-	-	-
126	2	Present Current	R	-	-	2.69 [mA]
128	4	Present Velocity	R	-	-	0.229 [rev/min]
132	4	Present Position	R	-	-	1 [pulse]
136	4	Velocity Trajectory	R	-	-	0.229 [rev/min]
140	4	Position Trajectory	R	-	-	1 [pulse]
144	2	Present Input Voltage	R	-	-	0.1 [V]
146	1	Present Temperature	R	-	-	1 [°C]
152	2	External Port Data 1	RW	-	-	-
154	2	External Port Data 2	RW	-	-	-
156	2	External Port Data 3	RW	-	-	-

Framework – sterowanie napędami

- Synchronizacja napędów







Framework – sterowanie napędami

- Sterujemy położeniem i wzmocnieniem P w trybie feed-forward
- Zakładamy że napęd potrzebuje wygenerować pewien momen obrotowy

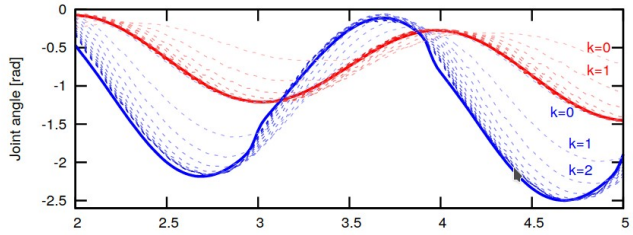
$$\tau = K_c V_B K_p (q_d - q)$$

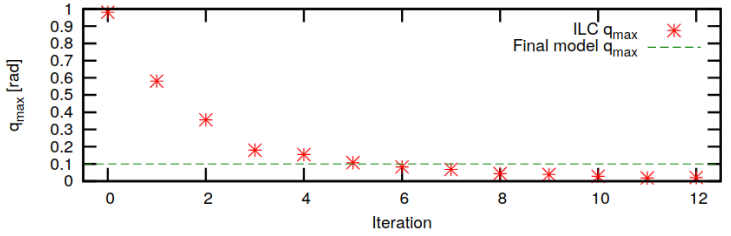
- Wliczając efekty tarcia, obliczamy całkowity moment

$$\tau = \tau_d + \alpha_1 \dot{q} + \alpha_2 \operatorname{sgn}(\dot{q})(1-\beta) + \alpha_3 \operatorname{sgn}(\dot{q})\beta$$

- Używając odwrotnej dynamiki (RBDL), obliczamy wymagany moment napędowy.
- Identyfikując parametry alfa, obliczamy q_d:

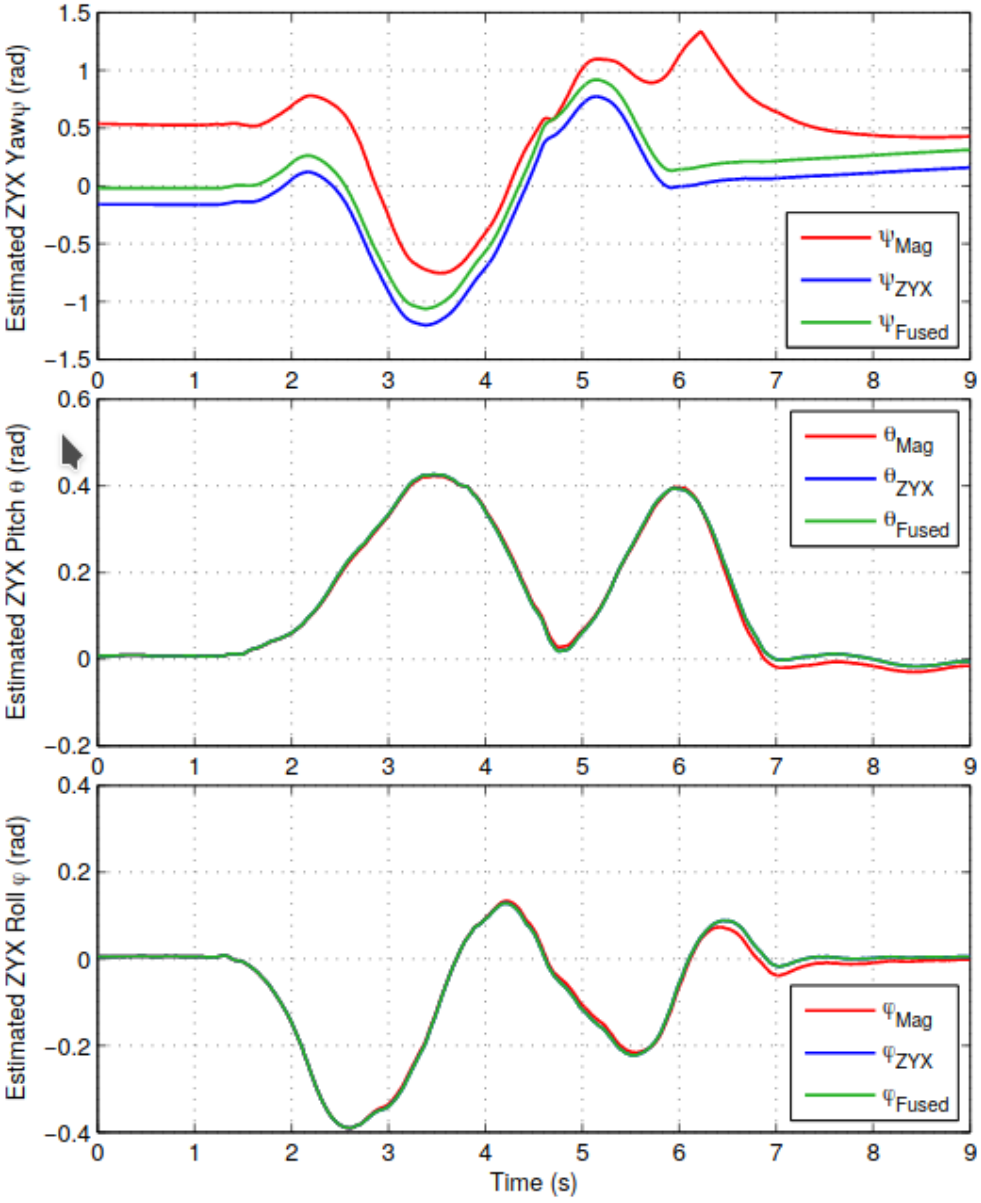
$$q_d = q + \frac{1}{V_B K_p}(\hat{\alpha}_0 \tau_d + \hat{\alpha}_1 \dot{q} + \hat{\alpha}_2 s_{\dot{q}}(1-\beta) + \hat{\alpha}_3 s_{\dot{q}}\beta)$$





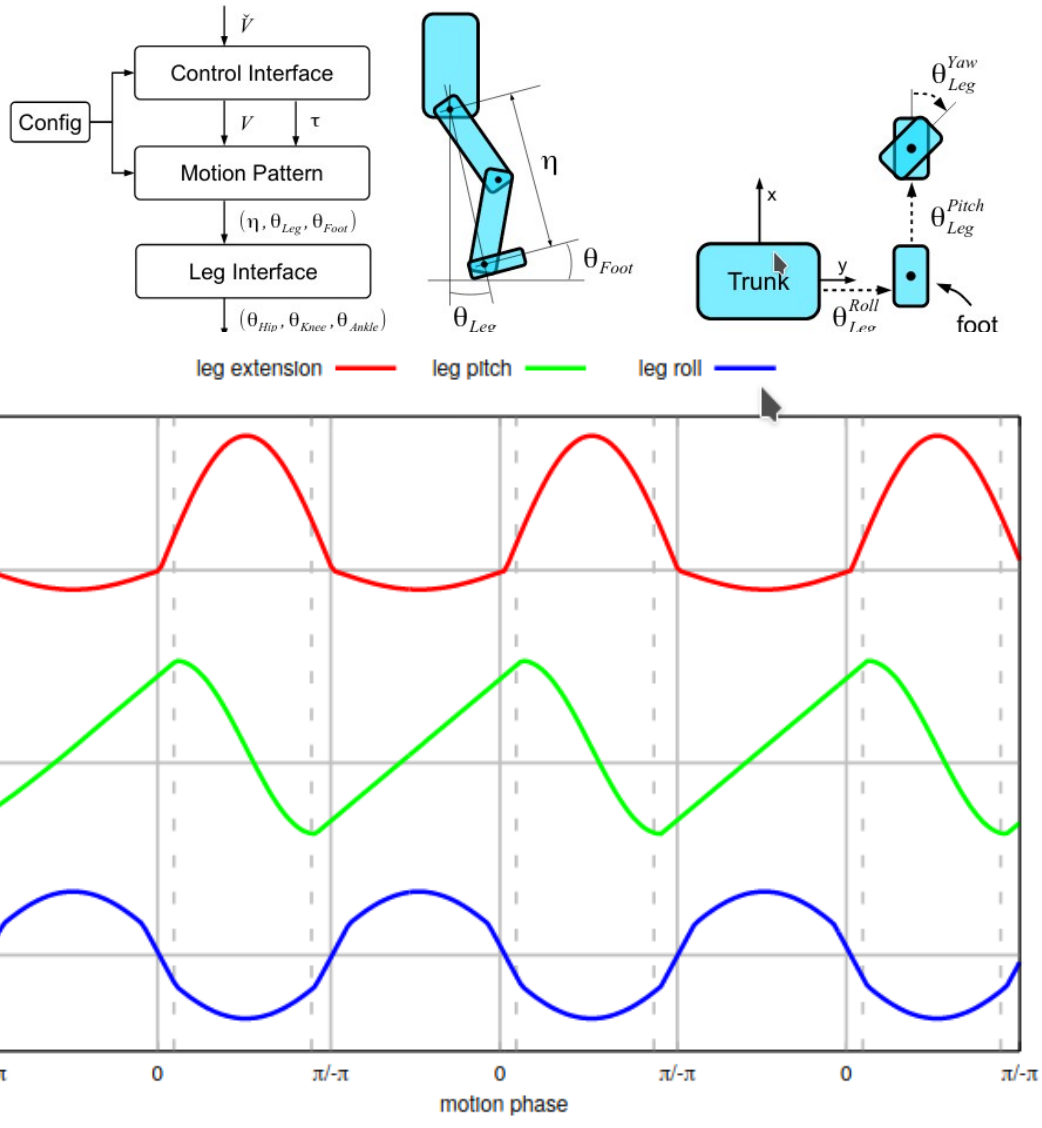
Framework – Estymacja stanu

- Dostępny mamy jedynie akcelerometr i żyroskop (od 2017 brak magnetometru – brak globalnego obrotu wokół osi Z)
- Zmodyfikowany nieliniowy pasywny filtr komplementarny Mahony'ego dostarcza szybkich i dokładnych estymat orientacji/ /wychylenia w przestrzeni



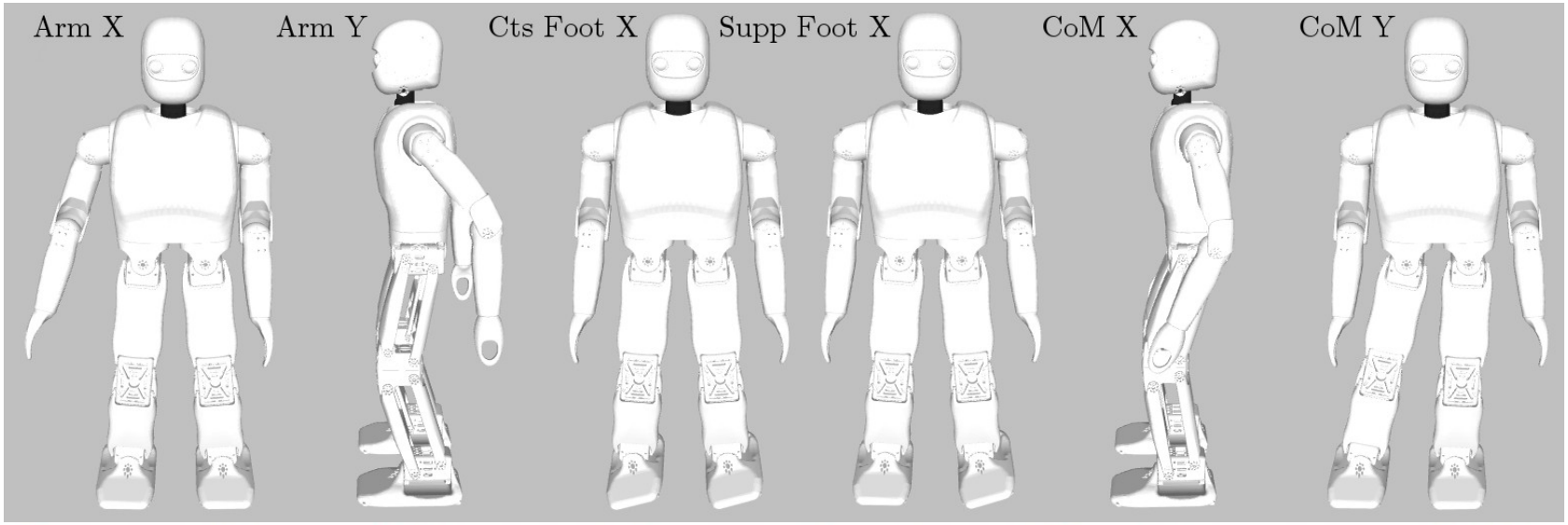
Framework – Gait: algorytm chodu

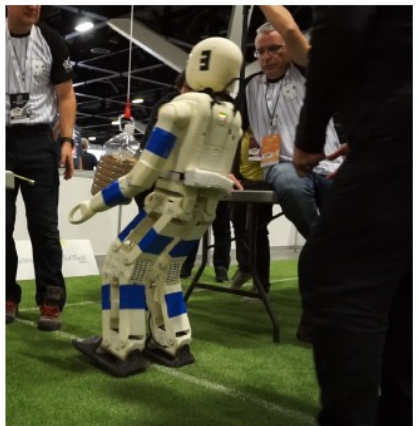
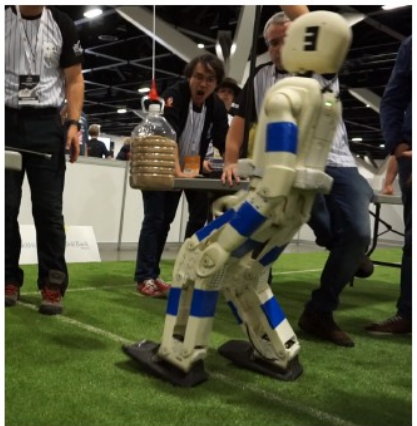
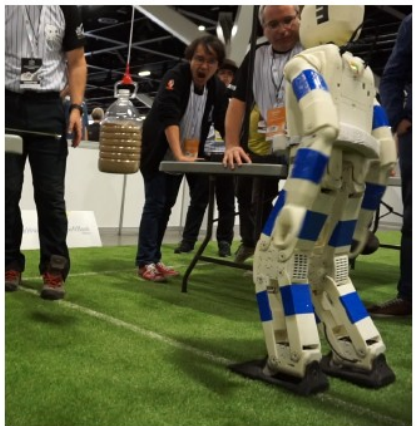
- Chód oparty jest o centralny generator, który na podstawie fazy chodu (-pi,pi] modyfikuje kąty nóg oraz rąk robota.
- Kinematyka odwrotna wyznacza dokładne wartości poszczególnych przegubów
- Nieskomplikowane obliczeniowo
- Szerokie zastosowanie
- Długa konfiguracja
- Brak sprzężenia zwrotnego!



Framework – Gait: stabilizacja chodu

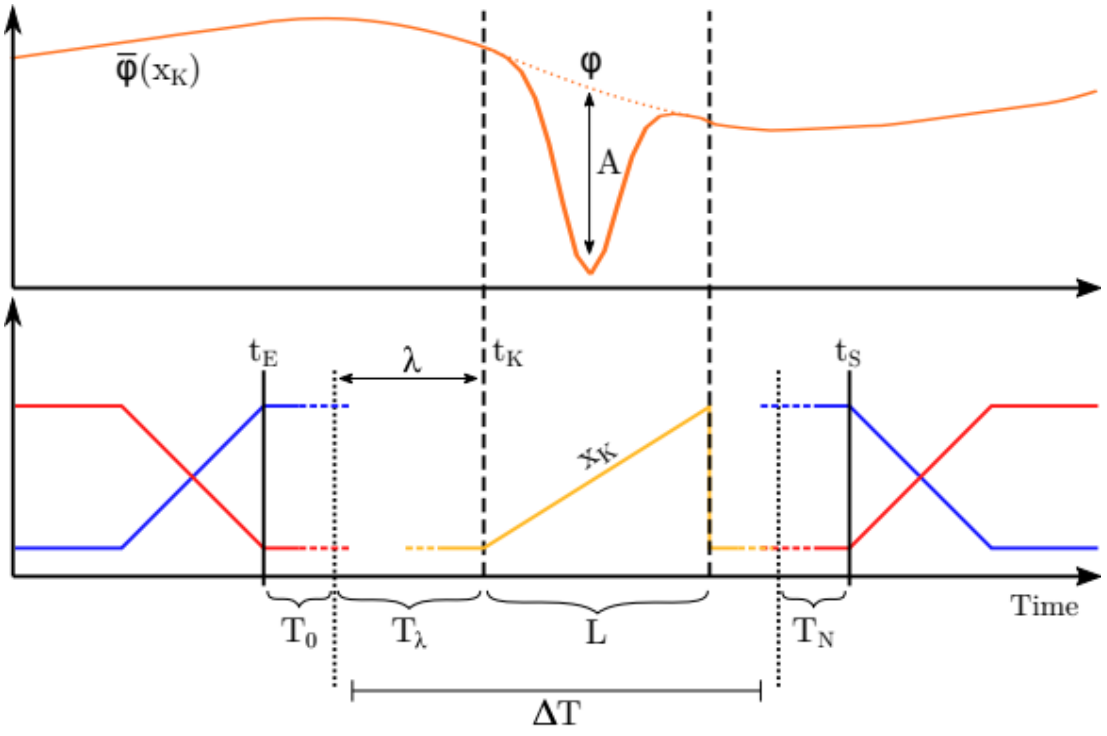
- Zależnie od estymat obrotu i ich zmian:
  - nakładane są modyfikatory do kątów rąk i nóg oraz fazy chodu by uzyskać “zamierzony” efekt
- Dodatkowo: Capture-step



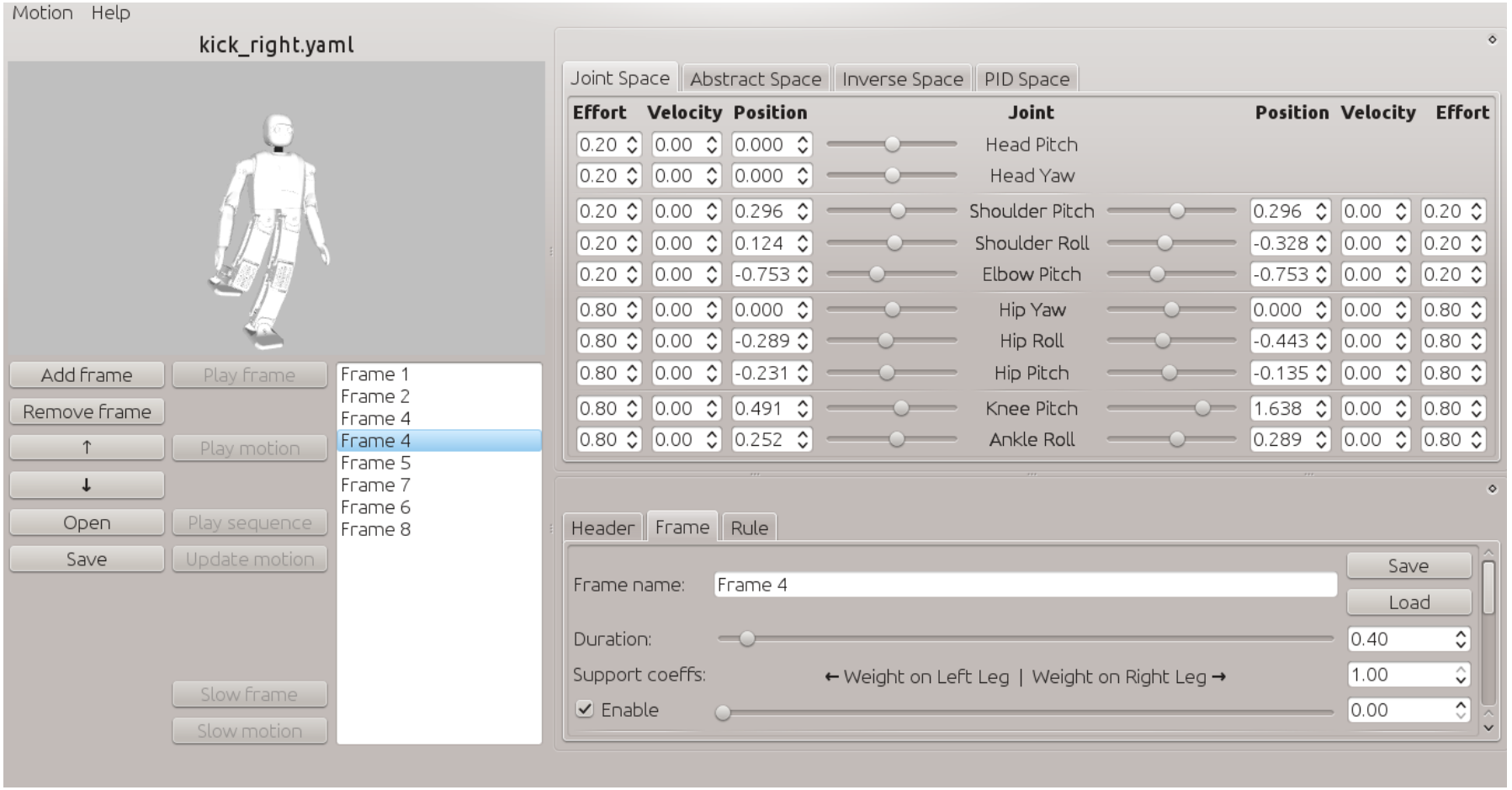
Framework – kopnięcia z chodu

- Moduł chodu udostępnia usługę kopnięcia, która modyfikuje zadane kąty nogi
- Zakolejkowane kopnięcie wykonywane jest gdy chód jest w odpowiedniej fazie



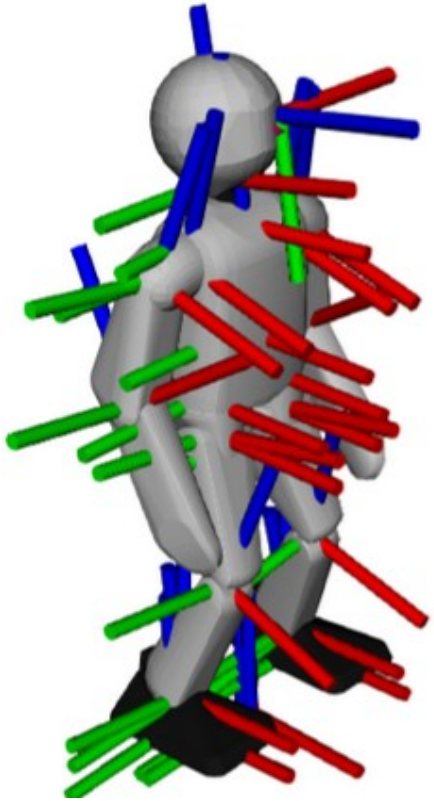
Framework – edytor trajektorii

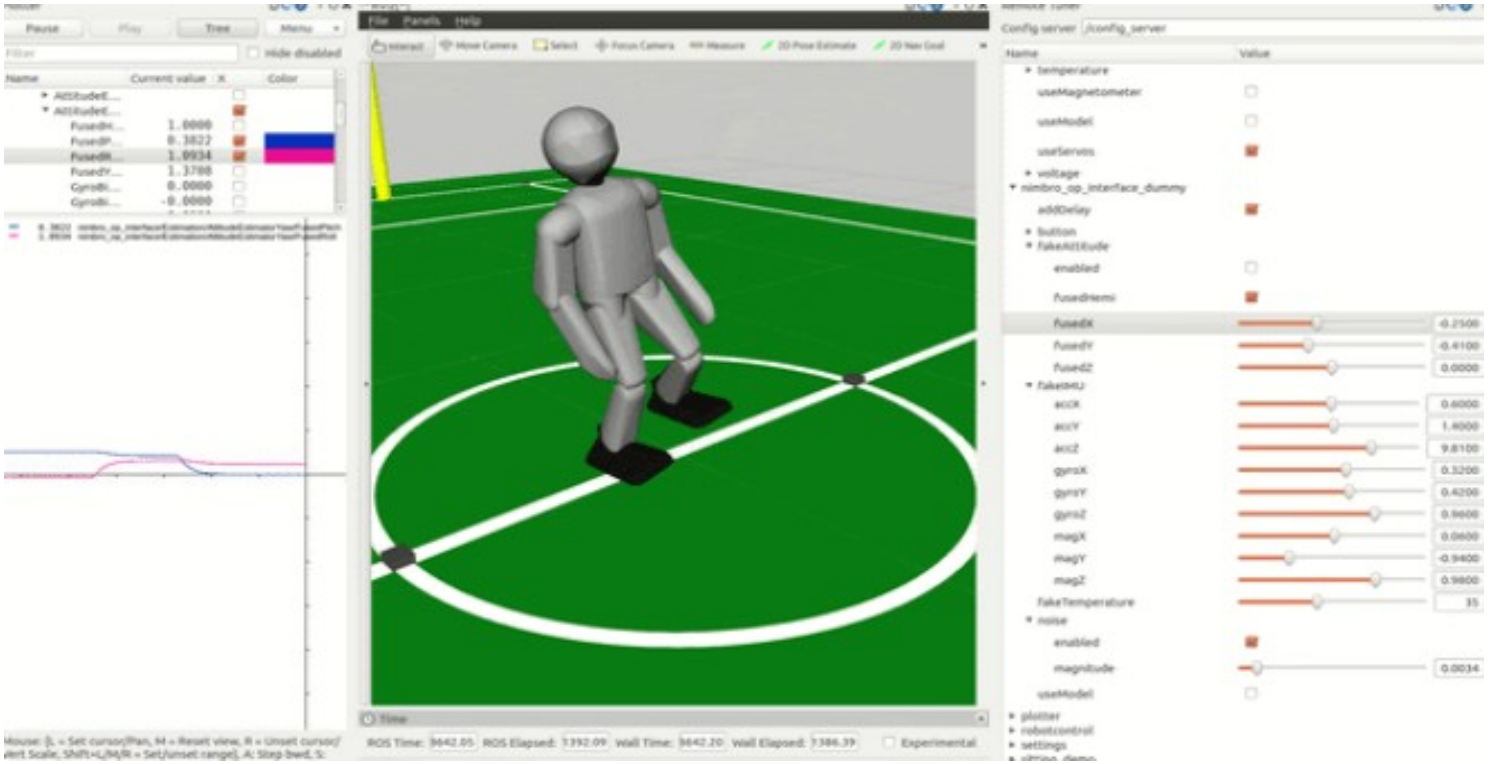
- Dość zaawansowane narzędzie mogące tworzyć dowolne ruchy w sposób kinematyczny
- Interpolacja pozycji, dynamika odwrotna możliwość dodania modyfikatorów PID.



Framework – rviz i tf

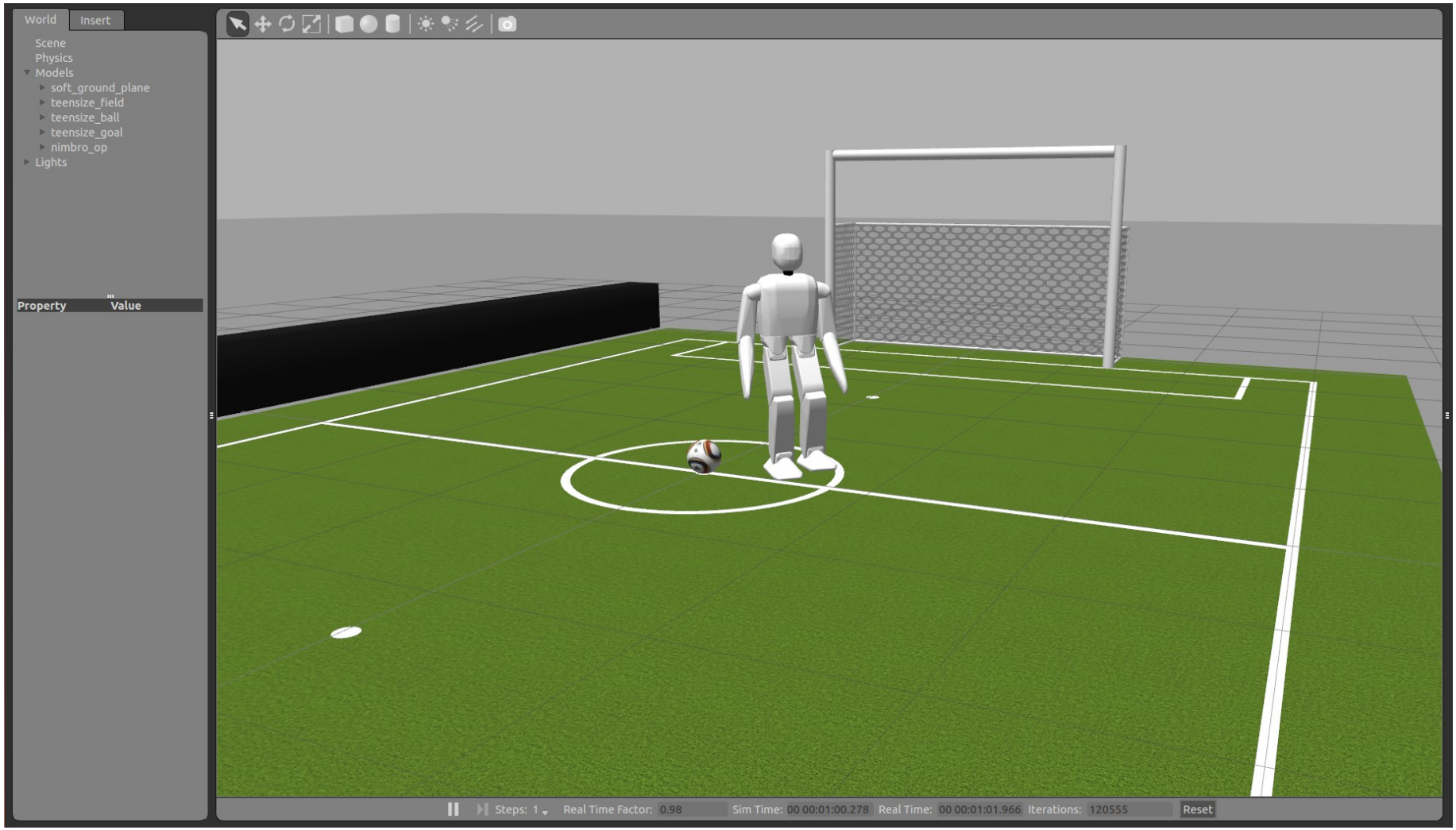
- Rviz to podstawowe narzędzie do wizualizacji robota i jego otoczenia
- Tf to podstawowa paczka określająca transformacje między poszczególnymi przegubami robota, lub jego otoczeniem.





Framework – symulacje

- Symulacja kinematyczna – rviz
- Symulacja fizyczna – Gazebo (& MuJoCo).

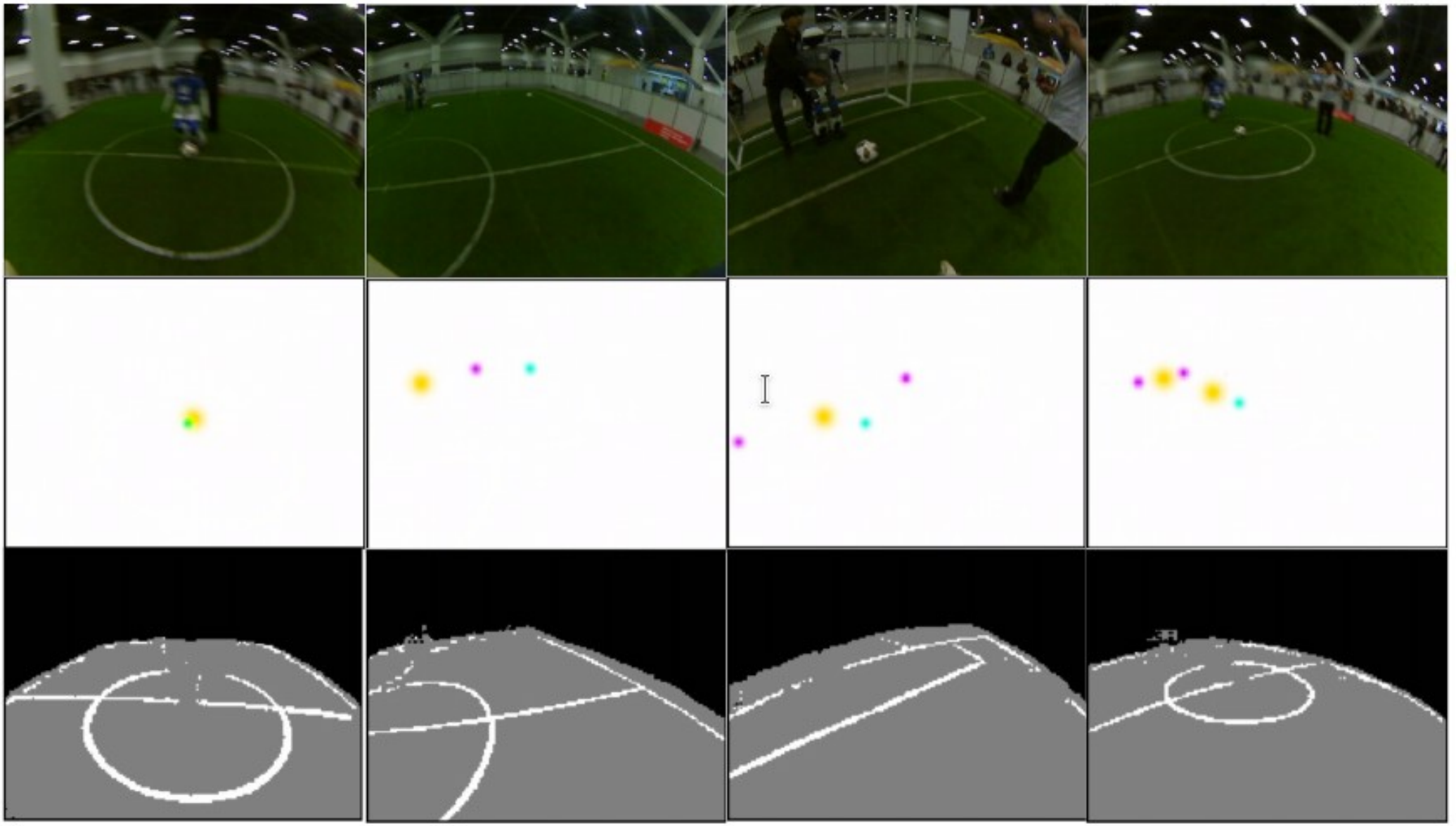


Framework – system wizji na DL

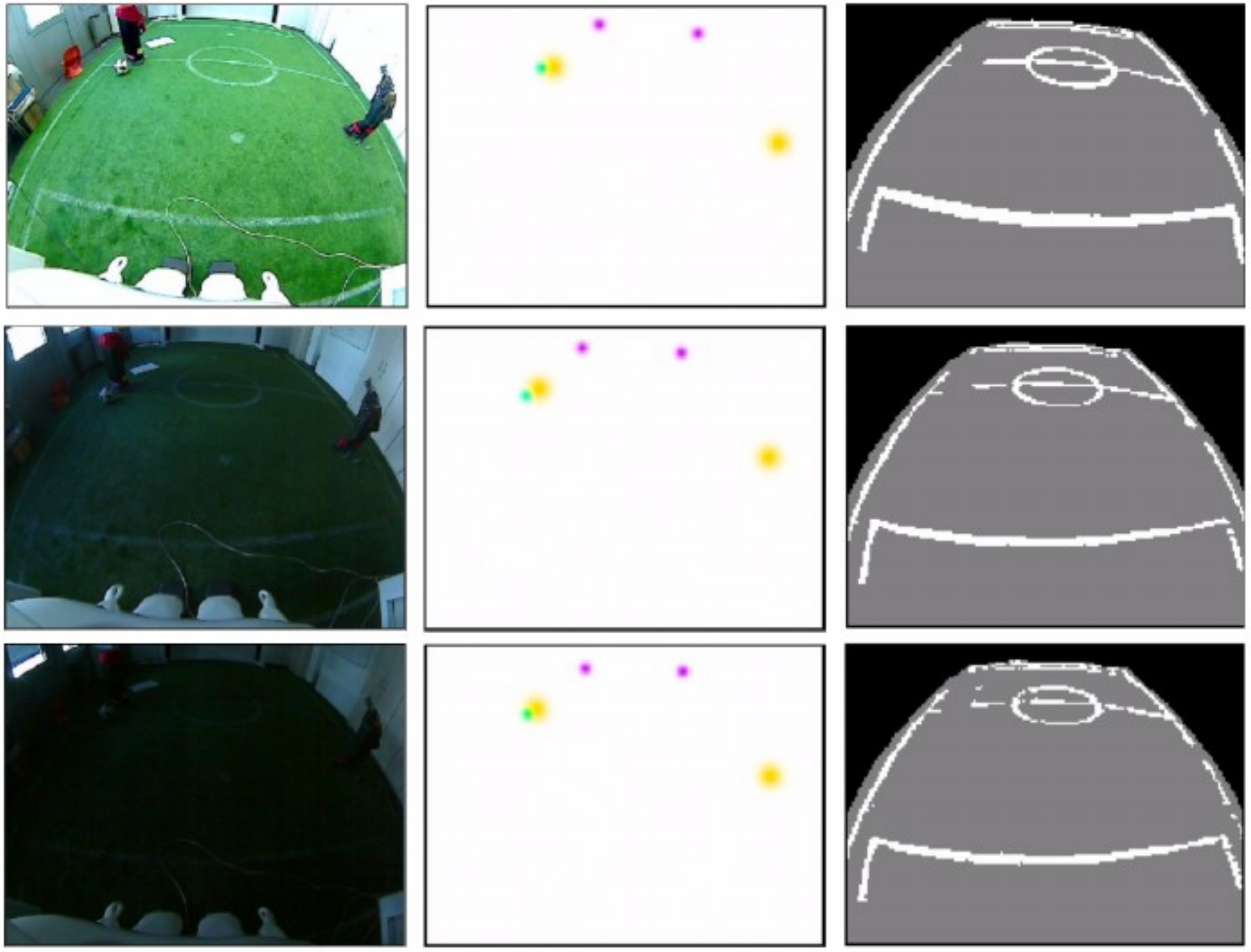
- Wszystkie detekcje wykonuje nasza sieć neuronowa NimbRoNet (splotowa sieć neuronowa - CNN)
  - Początkowo nauczona ResNet jako baza
  - Dwa wyjścia:
    - detekcja obiektów (piłka, słupki, roboty)
    - pikselowa segmentacja pola
  - Odporna na:
    - zmiany oświetlenia
    - zmiany kąta widzenia
    - zniekształcenia soczewki
  - Inferencja ~ 20ms
  - Bardzo stabilne detekcje

Type	F1	Accuracy	Recall	Precision	FDR
Ball (NimbRoNet2)	**0.998**	**0.996**	0.996	**1.0**	**0.0**
Ball (NimbRoNet)	0.997	0.994	**1.0**	0.994	0.005
Ball (SweatyNet-1 [10])	0.985	0.973	0.988	0.983	0.016
Goal (NimbRoNet2)	**0.981**	**0.971**	0.973	**0.988**	**0.011**
Goal (NimbRoNet)	0.977	0.967	**0.988**	0.966	0.033
Goal (SweatyNet-1 [10])	0.963	0.946	0.966	0.960	0.039
Robot (NimbRoNet2)	**0.979**	**0.973**	**0.963**	**0.995**	**0.004**
Robot (NimbRoNet)	0.974	0.971	0.957	0.992	0.007
Robot (SweatyNet-1 [10])	0.940	0.932	0.957	0.924	0.075
Total (NimbRoNet2)	**0.986**	**0.986**	0.977	**0.994**	**0.005**
Total (NimbRoNet)	0.983	0.977	**0.982**	0.984	0.015
Total (SweatyNet-1 [10])	0.963	0.950	0.970	0.956	0.043

Framework – system wizji na DL

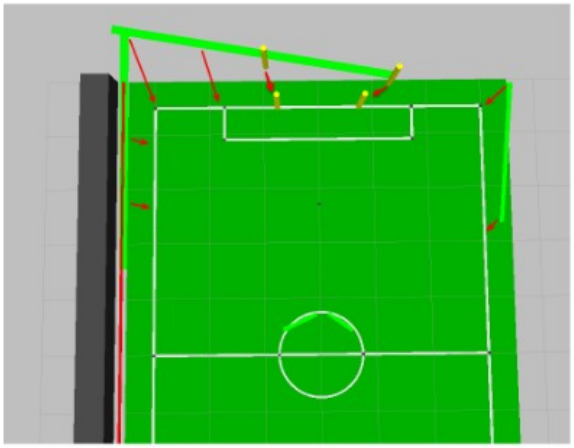


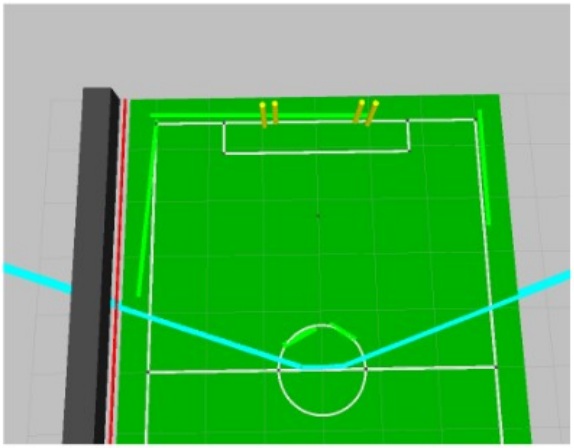
Framework – system wizji na DL



Framework – Lokalizacja

- Idea:
  - projekcja obserwacji z kamery na dwuwymiarowy model boiska
- Obserwacje:
  - Przecięcia linii (+,T,L)
  - Słupki bramek
  - Markery karnych
  - Koło środkowe

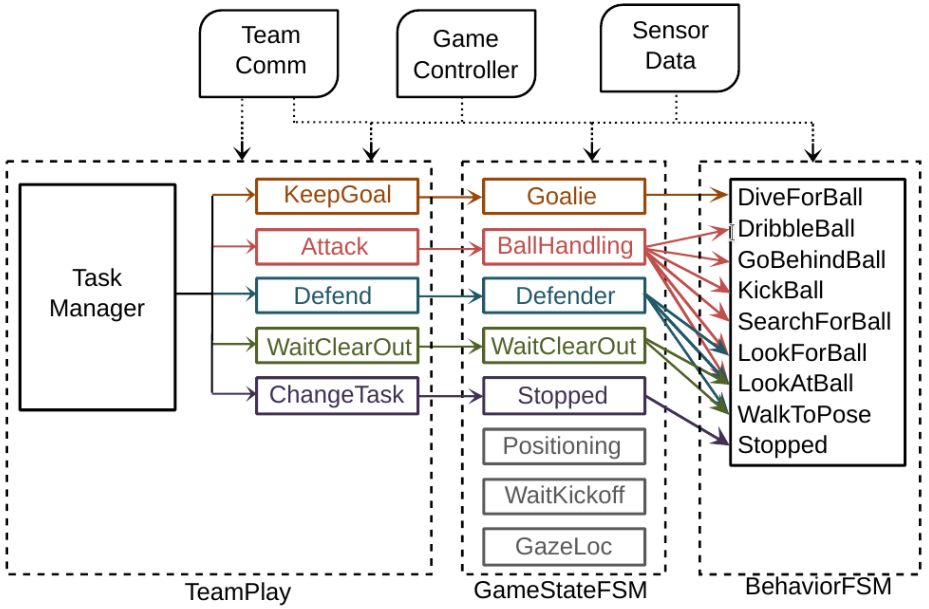


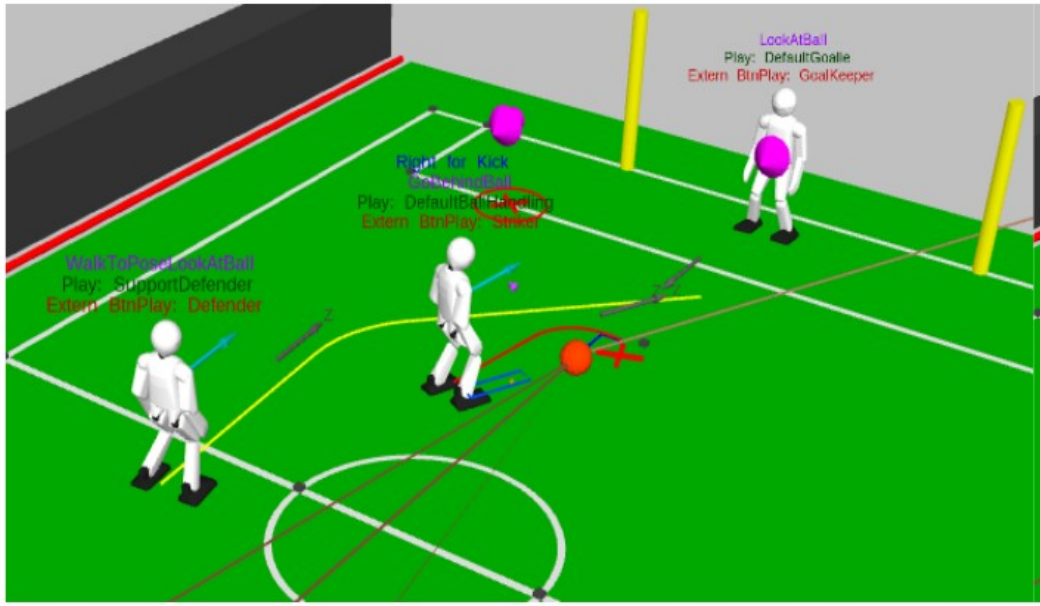


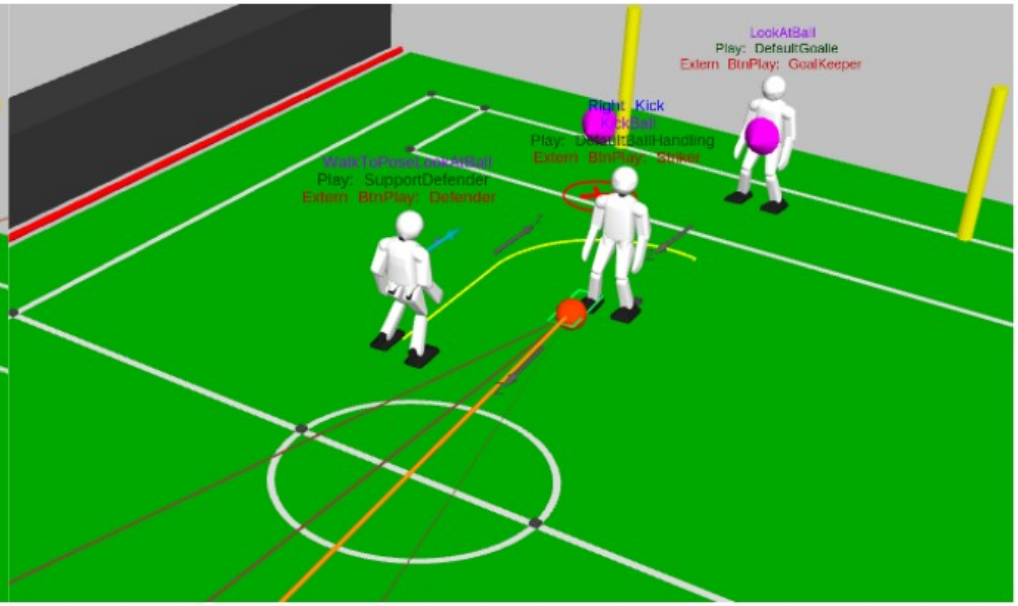
- Autolokalizacja na podstawie odległości do projekcji obserwacji
- Duża zależność od kalibracji tf kamery
- Globalna orientacja uzyskiwana przy wejściu na boisko

Framework – zachowania, teamplay

- FSM sterujący zachowaniem robotów i koordynacją zespołową
- Tylko jeden napastnik
- Możliwa wymiana ról na zasadzie server(napastnik)/ client(reszta)

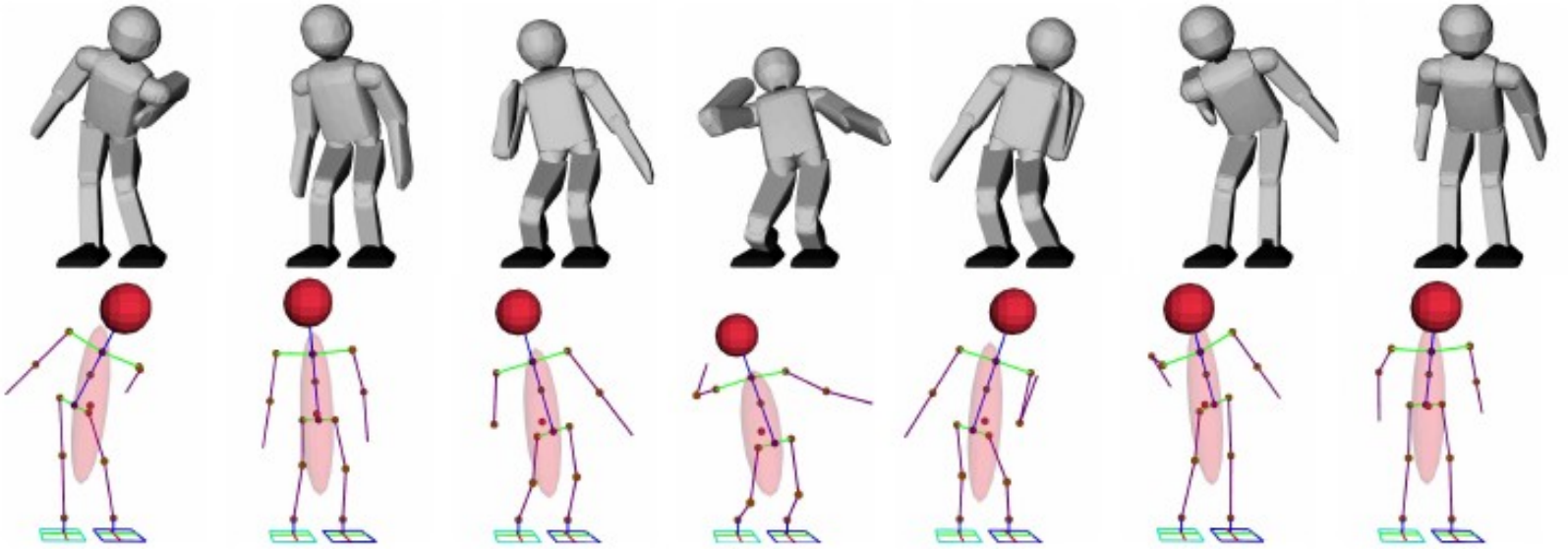






Co dalej?

- Roboty mogące poruszać się dynamicznie – skakanie, bieganie.
- Odporne na zewnętrzne uderzenia (compliant)
- Dynamiczna kontrola z uwzględnieniem momentów robota (centroidal momenta)



Dziękuję za uwagę!

Więcej informacji, źródła:

https://NimbRo.net/OP ; https://ais.uni-bonn.de