

Symulacja inspekcji przestrzennej ekstremalnie modularnym hiper-redundantnym manipulatorem Arm-Z

Jacek Szklarski¹, Ela Zawidzka¹, Tomasz Ponikiewski², Machi Zawidzki^{1,3}

¹ Instytut Podstawowych Problemów Techniki PAN, Adolfa Pawińskiego 5B, 02-106 Warszawa

² Politechnika Śląska, Wydział Budownictwa, ul. Akademicka 5, 44-100 Gliwice

³ Sieć Badawcza Łukasiewicz – Przemysłowy Instytut Automatyki i Pomiarów PIAP, Al. Jerozolimskie 202, 02-486 Warszawa

Streszczenie: Arm-Z to koncepcja hiper-redundantnego manipulatora robotycznego składającego się z identycznych modułów o jednym stopniu swobody (1-DOF) i realizującego dowolne ruchy końcówką w przestrzeni roboczej. Dwie zasadnicze zalety Arm-Z to oszczędność (dzięki masowej produkcji identycznych elementów) oraz odporność na awarie (zepsute moduły mogą być łatwo zastąpione). Z drugiej strony, sterowanie tak bardzo nieliniowym systemem jest znacznie trudniejsze niż typowym manipulatorem przemysłowym i wymaga zastosowania odpowiednich technik optymalizacyjnych. W artykule przedstawiono wstępne wyniki implementacji autorskiego algorytmu opartego na metodzie gradientu prostego do znajdowania trajektorii przejścia między zadanymi stanami Arm-Z. Przykładowym zadaniem dla manipulatora jest bezpieczne wejście do przestrzeni roboczej, inspekcja zadanych pięciu punktów oraz bezpieczne opuszczenie przestrzeni roboczej. Ten eksperyment jest trójwymiarową wersją realizowanej wcześniej inspekcji dwuwymiarowej.

Słowa kluczowe: Arm-Z, manipulator hiper-redundantny, inspekcja

1. Wstęp

Trąba słonia oraz wąż są przykładami niezwykle przystosowania do różnego rodzaju środowisk. Ta cecha wynika z wysokiej redundancji, czyli nadliczbowości stopni swobody tego typu mechanizmów. W przypadkach nieregularnego środowiska pracy, roboty przypominające biologiczne trąby słonia lub węże przewyższają tradycyjne roboty poruszające się na kołach, nogach lub gąsienicach. Badania nad tego typu robotami prowadzone są już od kilku dekad. Wężowy rodzaj lokomocji był już studiowany w latach czterdziestych dwudziestego wieku [2]. Pięćdziesiąt lat później został opracowany ścisły model matematyczny tego typu lokomocji [3]. Od dawna jednym z wyzwań naukowców zajmujących się robotyką było stworzenie hiper-redundantnego ramienia wielostawowego, które, podobnie

jak trąba słonia, mogłoby aktywnie omijać przeszkody wślizgnąć się w trudno dostępne i wąskie przestrzenie [4]. Do tej pory przeprowadzono wiele badań symulacyjnych nad problemami związanymi ze sterowaniem nadmiarowym robotów hiper-redundantnych. Niemniej zaprojektowanie fizycznego mechanizmu ramienia okazało się bardzo trudne i do tej pory wykonano i wykorzystano tylko kilka praktycznych modeli [5]. Większość z nich przeznaczona jest do pełzania [6–10], niektóre do pływania [11, 12], a jeszcze mniej zarówno do pełzania, jak i pływania [13, 14]. Przykład tych ostatnich pokazano na Rys. 1.

W rozmaitych środowiskach, podobnie jak biologiczne węże lub trąby słonia, manipulatory i lokomotory hiper-redundantne mają pewną przewagę nad konwencjonalnymi robotami. Mogą one działać w skomplikowanych geometrycznie środowiskach,



Rys. 1. Fotografia modularnego robota AmphiBot skonstruowanego na EPFL w Szwajcarii (www.epfl.ch/labs/biorob/)

Fig. 1. A photograph of the modular AmphiBot robot constructed in the School of Computer and Communication Sciences, EPFL Switzerland (www.epfl.ch/labs/biorob/)

Autor korespondujący:

Machi Zawidzki, zawidzki@piap.lukasiewicz.gov.pl

Artykuł recenzowany

nadesłany 15.12.2024 r., przyjęty do druku 16.07.2025 r.



Zezwala się na korzystanie z artykułu na warunkach licencji Creative Commons Uznanie autorstwa 4.0 Int.



Rys. 2: Oliver Crispin Robotics Ltd. Manipulatory bioniczne z serii II, systemu X125. Po lewej: manipulator z laserową głowicą tnącą. Po prawej: manipulator inspekcyjny z lampą i kamerą (<http://www.ocrobotics.com>)

Fig. 2. Oliver Crispin Robotics Ltd. Snake-arm robots, series II, X125 System. On the left: sleeved and integrated manipulator with a laser cutting head. On the right: unsleeved, integrated manipulator with inspection camera and light tool (www.ocrobotics.com)

które są niedostępne dla innych metod. W zależności od wymaganego zadania, na takich manipulatorach można instalować różnego rodzaju głowice robocze, np. do: spawania, czyszczenia, monitorowania itp. (Rys. 2).

W ogólności sześć stopni swobody wystarcza do wykonania dowolnego ruchu w przestrzeni trójwymiarowej: przemieszczenie wzdłuż trzech osi kartezjańskich X, Y, Z; oraz trzy obroty: odchylenie, pochylenie i przechylenie. Konwencjonalny manipulator przemysłowy ma niewielką liczbę DOF – zwykle tylko sześć. Z drugiej strony, ludzkie ramię jest biologicznym archetypem kinematycznie redundanтного manipulatora z siedmioma stopniami swobody: trzy w ramieniu, jeden w łokciu i trzy w nadgarstku. W wielu robotach przemysłowych zastosowano ten układ kinematyczny, np. PA-10 firmy Mitsubishi, Lightweight Robot DLR (Deutsches Zentrum für Luft- und Raumfahrt) itp. DEXTER firmy Scienza Macchinale jest przykładem manipulatora o ośmiu stopniach swobody. Systemy z większą liczbą przegubów nazywane są redundanтными, podczas gdy termin hiper-redundantny odnosi się do manipulatorów z bardzo dużą, być może nieskończoną, liczbą stopni swobody [15]. Manipulatory takie można dalej podzielić na: manipulatory sztywne podobne do kręgowców, np. takie jak węże, oraz manipulatory ciągłe podobne do bezkręgowców, takie jak ramiona ośmiornicy lub trąba słońca.

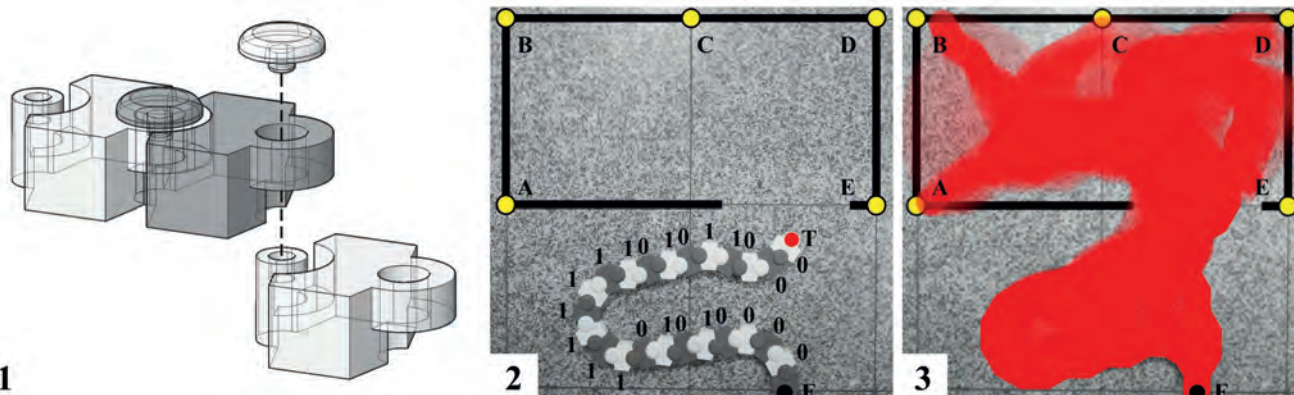
Problem kinematyki odwrotnej dla manipulatora szeregowego polega na określeniu pozycji przegubów z uwzględnieniem pozycji i orientacji końcówki manipulatora. Rozwiązania w postaci zamkniętej są praktyczne, ponieważ łatwo identyfikują wszystkie możliwe rozwiązania szybciej niż metody numeryczne [16]. Problem kinematyki odwrotnej typowego manipulatora przemysłowego można często rozwiązać anali-

tycznie [17], co upraszcza implementację układu sterowania. Manipulatory hiper-redundantne są jednak systemami wysoce nieliniowymi, dlatego problem kinematyki odwrotnej jest dla nich znacznie trudniejszy, a często wręcz nierozwiązalny metodami analitycznymi. Przeto ich sterowanie nie jest łatwe i zwykle wymaga zastosowania metod inteligencji obliczeniowej [18–20].

W pracy [21] zaproponowano metodę rozwiązania problemu kinematyki odwrotnej planarnego manipulatora redundanтного w postaci zamkniętej. Opierała się ona na zastosowaniu różniczki Frecheta pewnej funkcji kryterialnej wprowadzonej w celu rozwiązania redundancji. Model ten nie zawierał jednak żadnych ograniczeń na zakres ruchu przegubów – co w rzeczywistości znacznie komplikuje sformułowanie. Co więcej, już w kinematyce prostej opis analityczny „eksploduje” wraz z liczbą ogniw, nie wspominając o kinematyce odwrotnej, która z natury jest trudniejsza. Dlatego też proponowane w [21] podejście nie jest praktyczne w przypadku manipulatorów hiper-redundantnych. Pionierska praca modelowania matematycznego dyskretnego redundanтного manipulatora planarnego w przestrzeni kartezjańskiej została przedstawiona w 1989 r. [22]. Ten sam autor przedstawił w pracy [23] model kinematyki obrotowego, redundanтного manipulatora w postaci automatu skończonego [24]. Implementacja heurystycznego algorytmu przeszukiwania grafowego w celu znalezienia bezkolizyjnej trajektorii dla pięcio-przegubowego planarnego redundanтного manipulatora została przedstawiona w [25]. Dla problemu omijania przeszkód przez planarne manipulatory hiper-redundantne, tak zwane podejście „tunelowania” zostało zaprezentowane w [15]. W kolejnym artykule [26] ci sami autorzy przedstawili hiper-redundantne mechanizmy robotów i ich zastosowania, w tym hiper-redundantnego robota o trzydziestu stopniach swobody. W pracy [27] dynamika hiper-redundantnych manipulatorów została sformułowana jako problem mechaniki ciągłej. Zaletą przedstawionej metody było to, że można ją łatwo zrównoleglić. Więcej informacji na temat tego typu manipulatorów można znaleźć w pracy [27].

2. Inspekcja planarna Arm-Z (poprzedni eksperyment)

We wcześniejszym eksperymencie [1] przedstawiono koncepcję niezwykle prostego manipulatora planarnego, składającego się z 24 przystających modułów. Każdy moduł ma tylko dwie możliwe dyskretne pozycje w stosunku do poprzedniego modułu: skręt w lewo ($-\pi/6$) lub w prawo ($\pi/6$) (Rys. 3.1). Manipulator jest umieszczony w środowisku eksperymentalnym (Rys. 3.2). Zadaniem jest umieszczenie bez kolizji końcówki manipulatora w pobliżu pięciu zadanych punktów. Pomimo



Rys. 3. 1. Dwa połączone ze sobą moduły i widok rozstrzelony pokazujący połączenie. 2. Konfiguracja eksperymentu. 3. Na czerwono pokazano obwiednię wszystkich pozycji manipulatora podczas eksperymentu

Fig. 3. 1. Two linked modules and an exploded view showing the connection. 2. Experiment setup. 3. The envelope of all manipulator positions during the experiment is shown in red

niezwykłej prostoty, manipulator skutecznie “wpełza” do przestrzeni roboczej, odwiedza wyznaczone punkty i bezpiecznie “wypłza”, co ilustruje Rys. 3.3.

W kolejnym rozdziale przedstawiono trójwymiarową wersję manipulatora Arm-Z oraz analogiczny eksperyment w przestrzeni 3D.

3. Przestrzenna inspekcja manipulatorem Arm-Z

Koncepcję modularnego manipulatora inspekcyjnego jako części orbitalnego systemu konserwacji przedstawiono w artykule [28]. Manipulator ten składał się z siedmiu modułów z obrotowymi złączami nachylonymi pod zadanym kątem, czyli każde ogniwo miało dwa stopnie swobody.

Trójwymiarowy manipulator Arm-Z wprowadzony w pracy [29] składa się z jednego rodzaju modułu, o jednym stopniu swobody – względnym skręcie. Każdy moduł jest określony przez kilka parametrów: wymiar r , odsunięcie d , i kąt ζ między powierzchniami dolną (**B**) i górną (**T**) (Rys. 4).

Kształt i pozycja manipulatora Arm-Z w danej chwili zależy od: liczby modułów, parametrów geometrycznych modułu i sekwencji względnych skrętów między każdą parą modułów. Na Rys. 5 przedstawiono dwa moduły przy kolejnych skrętach od 0 do 180° co 30°.

Tradycyjnie analiza kinematyczna i planowanie ruchu dla manipulatorów redundantnych opierały się na: pseudo-odwrotności [30], uogólnionej odwrotności [31] lub rozszerzonej odwrotności [32] Jakobianu manipulatora. Niektórzy badacze rozważali schematy znajdowania nadmiarowych trajektorii manipulatora, które optymalizują prędkości lub momenty obrotowe przegubów na całej trajektorii zamiast od punktu do punktu, co opisano m.in. w pracach [33, 34]. Inne schematy kinematyki odwrotnej zostały opracowane w oparciu o koncepcję izotropii dynamicznej [35]. Efektywna obliczeniowo metoda wyznaczania optymalnych kinematycznie konfiguracji hiper-redundantnych manipulatorów została przedstawiona w artykule [36]. Wprawdzie w szczególnych przypadkach ist-

nieją rozwiązania w formie zamkniętej dla niektórych rodzajów manipulatorów hiper-redundantnych [37], jednakże w tym przypadku jest to niemożliwe. W następnym podrozdziale przedstawiono hybrydowy algorytm sterowania oparty na numerycznym szukaniu Jakobianu dla znalezienia docelowych stanów poszczególnych modułów oraz wykorzystaniu silnika fizyki do określania wymaganych prędkości kątowych.

3.1. Algorytm wyszukania ścieżki przejścia do stanu docelowego

Ustawienie N modułów reprezentowane jest przez wektor $\mathbf{S} = [\theta_1, \theta_2, \dots, \theta_N]$, gdzie θ_i to kąt obrotu modułu i względem poprzedniego $i - 1$. $\mathbf{S}$ determinuje pozycję środka ostatniego modułu $\mathbf{p}$ i wektora normalnego $\mathbf{n}$ do płaszczyzny zamykającej ostatni moduł w punkcie $\mathbf{p}$. Wektory $\mathbf{p}$ i $\mathbf{n}$ w pełni definiują pozycję ostatniego modułu, na którym może być zamontowany efektor, np. w postaci chwytaka czy kamery inspekcyjnej ($\mathbf{n}$ jest niezbędny by określić kąty ustawienia powierzchni zamykającej ostatni moduł).

Manipulator składa się z szeregu N ogniw połączonych ze sobą złączami, a z każdym ogniwem tego łańcucha kinematycznego wiążemy układ kartezjańskich współrzędnych, którego środek znajduje się w środku koła tworzącego górne zamknięcie modułu. Wektor normalny $\mathbf{n}$ w tym nowym układzie ma współrzędne $[0, 0, 1]$ i stanowi oś obrotu dla kolejnego modułu.

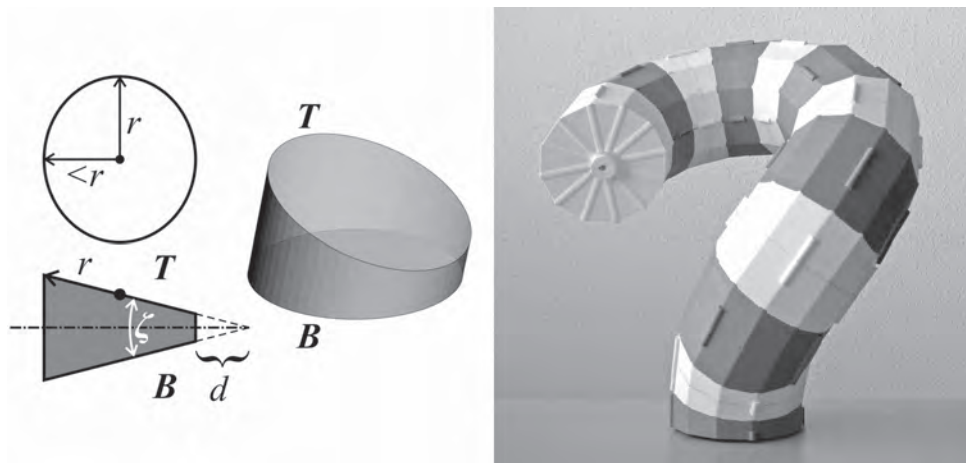
Macierz przejścia między układem złącza $i - 1$ oraz i została określona równaniem:

$${}^{i-1}H_i = \text{Rot}(Z, \theta_i) \text{Trans}(0, \delta_y, \delta_z) \text{Rot}(X, \zeta), \quad (1)$$

gdzie $\text{Rot}(Z, \alpha)$ to macierz obrotu wokół lokalnej osi Z o kąt α i $\text{Rot}(X, \alpha)$ analogicznie dla osi X .

Pozycję efektora (czyli ostatniego modułu) można otrzymać przez złożenie kolejnych transformacji:

$${}^0H_N(\theta_1, \theta_2, \dots, \theta_N) = \prod_{i=1}^N [\text{Rot}(Z, \theta_i) \text{Trans}(0, \delta_y, \delta_z) \text{Rot}(X, \zeta)]. \quad (2)$$



Rys. 4. Po lewej moduł Arm-Z definiowany trzema parametrami geometrycznymi: r , d oraz ζ . Po prawej wczesna makieta Arm-Z

Fig. 4. On the left: The Arm-Z module defined by three geometric parameters: r , d , and ζ . On the right: An early mock-up of Arm-Z



Rys. 5. Dolny (ciemnoszary) moduł jest przymocowany do podstawy i nie obraca się. Oś obrotu, która jest prostopadła do dolnej powierzchni górnego modułu, jest zaznaczona na ciemnoszaro

Fig. 5. The bottom (dark gray) module is fixed to the base and does not rotate. The axis of rotation, which is perpendicular to the bottom face of the top module is shown in dark gray

Wartości stałych $\zeta, \delta_y, \delta_z$, charakteryzujących moduł wynikają z założonej geometrii modułu, tutaj założono $\zeta = 22,5^\circ$, $\delta_y = -0,1397$ m, $\delta_z = 0,7023$ m (Rys. 4).

Moduł 0, pełniący rolę podstawy, pozostaje nieruchomy względem podłoża i w tym konkretnym omawianym przypadku jest zakotwiczony pod kątem $\zeta/2$ (Rys. 4), a oś obrotu modułu 1 wyznacza początkowy układ współrzędnych związany ze środkiem, przekształcony $\text{Trans}(0, \delta_y, \delta_z/2) \text{Rot}(X, \zeta/2)$.

Celem algorytmu jest znalezienie kolejnych stanów $\mathbf{S}$, które przeniosą środek ostatniego modułu $\mathbf{p}$ i wektor normalny $\mathbf{n}$ do zadanej docelowej pozycji $\mathbf{p}_d$ wraz docelowym wektorem normalnym $\mathbf{n}_d$. Musi być to zrealizowane w sposób zapewniający unikanie kolizji manipulatora z samym sobą i statycznym otoczeniem. Dodatkowo stan końcowy $\mathbf{S}_e$ jest nieznany, gdyż w ogólnym przypadku nie można zakładać, że np. $\mathbf{p}_d$ leży w przestrzeni zawierającej punkty możliwe do osiągnięcia przez manipulator. Wynika to z charakterystyki geometrii manipulatora.

Założono, że w czasie t stan manipulatora określić można za pomocą $\mathbf{S}_t$ i prędkości kątowych $\dot{\mathbf{S}}_t$ (prędkości początkowe są równe 0, $\dot{\mathbf{S}}_0 = 0$). Celem jest znalezienie trajektorii $\mathbf{S}_{0,\dots,T}$, która przeniesie pozycję końcowego modułu do pozycji docelowej $\mathbf{p}$, $\mathbf{n}$ w czasie końcowym $t = T$. Dynamika przejść między $\mathbf{S}_t$ i $\mathbf{S}_{t+1}$ jest realizowana w symulatorze.

Ze względu na określone założenia oraz możliwość występowania przeszkód, problem jest traktowany jako optymalizacja wielokryterialna, tzn. w każdym kroku należy minimalizować zarówno odległość do celu, jak i zbliżać bieżący wektor normalny do docelowego. Minimalizacja numeryczna dotyczy funkcji celu:

$$f(\mathbf{S}) = \tau_T d(\mathbf{p}, \mathbf{p}_d) + \tau_N (1 - \mathbf{n} \cdot \mathbf{n}_d)^2 + \lambda(\mathbf{S}), \quad (3)$$

gdzie τ_T i τ_N to wagi nadające znaczenie nagrodzie, odpowiednio za zbliżenie się do celu i osiągnięcie zakładanego wektora normalnego, d to odległość między punktami, $\lambda(\mathbf{S})$ to funkcja osiągająca bardzo dużą wartość (pięć rzędów wielkości więcej niż typowe wartości f) w przypadku, gdy dla konfiguracji $\mathbf{S}$ manipulator jest w kolizji z samym sobą lub otoczeniem. Gdy $f(\mathbf{S}) < \varepsilon$ zakłada się, że cel został osiągnięty.

Zasadniczy algorytm wyszukania ścieżki wygląda następująco:

```

 $\mathbf{S} \leftarrow \mathbf{S}_0$ 
while  $f(\mathbf{S}) < \varepsilon$  do
     $\mathbf{S}_{t+1} \leftarrow \mathbf{S}_t + \gamma \dot{\mathbf{S}}_t$ 
    posctrltowards( $\mathbf{S}_{t+1}$ )
end while
    
```

gdzie $\gamma = 0,05$ to parametr będący miarą prędkości podążania za gradientem, ε przyjęto 0,5 dla wag $\tau_T = \tau_N = 1$.

Stosunkowo duża wartość ε wynika z faktu, że ze względu na swoją naturę, z dużym prawdopodobieństwem manipulator nie tylko nie może dokładnie osiągnąć docelowego punktu $\mathbf{p}_d$, ale także zakładanego wektora normalnego $\mathbf{n}_d$ zależy nam na szybkim rozwiązaniu przybliżonym. Do symulacji wykorzystano model stworzony w realistycznym środowisku symulacyjnym PyBullet [38]. PyBullet rozwiązuje pełne równania dynamiki przy użyciu algorytmu *Featherstone Articulated-Body* w taki sposób, że masy, tensory bezwładności, grawitacja oraz wszystkie człony Coriolisa/odśrodkowe są automatycznie uwzględniane. Sterowanie realizowane jest jako zadawanie kolejnych docelowych kątów $\mathbf{S}_p$, natomiast prędkości, wyliczanie momentów obrotowych dla przegubów i całkowanie równań ruchu realizowane jest już w symulatorze. Żeby było to możliwe, niezbędne jest założenie pewnych maksymalnych momentów obrotowych dla każdego przegubu.

W przytoczonym pseudokodzie funkcja posctrltowards realizuje takie właśnie sterowanie przez zadany okres (tutaj 100 kroków symulacji, gdzie każdy krok odpowiada 1/240 s; ruch taki oczywiście zmienia bieżący stan $\mathbf{S}$ w kierunku oczekiwanego $\mathbf{S}_{t+1}$, chociaż sam stan $\mathbf{S}_{t+1}$ nie musi zostać osiągnięty. Jeśli kąty manipulatora nie osiągną $\mathbf{S}_{t+1}$, nie ma to wpływu na dalsze działanie, gdyż w kolejnych krokach funkcja celu jest minimalizowana, tak więc szukane są kąty $\mathbf{S}$, które zbliżą końcowy moduł do pozycji docelowej.

Symulator został uruchomiony na typowym komputerze typu desktop i działał w czasie rzeczywistym. Wspomniane 100 kroków to kroki realizowane przez symulator, gdzie zadana trajektoria (tzn. kolejny jej wektor $\mathbf{S}$) jest odtwarzana przez wewnętrzny regulator pozycyjny. Między tymi krokami realizowany jest algorytm znalezienia kolejnego stanu $\mathbf{S}$. Można więc takie planowanie określić jako działające w trybie on-line, gdyż zarówno symulator, jak i planowanie mogą działać w czasie rzeczywistym.

Ze względu na fakt, że nie czyni się żadnych założeń dotyczących środowiska, w którym działa manipulator (czyli miejsc, w których pojawia się dodatkowa kara za kolizję) oraz stosunkowo skomplikowane zależności $\mathbf{p}$ i $\mathbf{n}$ od stanu układu, najwygodniej – w przypadku symulacji – obliczyć $\nabla f(\mathbf{S})$ numerycznie. Wykorzystano `approx_fprime` ze `scipy.optimize`, która numerycznie oblicza jacobian funkcji skalarnej za pomocą różnic skończonych – przybliżając pochodne cząstkowe przez perturbację każdej zmiennej niezależnie i mierząc zmiany wartości funkcji celu.

Na aktualnym etapie badań nie zakładamy zbyt szczegółowego realności symulacji, gdyż (1) interesuje nas przede wszystkim zachowanie emergentne manipulatora pod kątem możliwości przejść między różnymi stanami, a nie jego sama dynamika; (2) fizyczny demonstrator układu jest w budowie i jego fizyczne parametry nie są jeszcze ustalone. Przyjmujemy tutaj masę modułu 1 kg i momenty bezwładności przybliżone cylindrem z wartościami tensora momentu bezwładności na diagonalu równymi (0,25; 0,25; 0,5) $\text{kg} \cdot \text{m}^2$. Zakładając maksymalny moment obrotowy na przegubie na poziomie 100 $\text{N} \cdot \text{m}$ otrzymujemy niezbyt realistyczny manipulator, który swoim regulatorem otrzymuje prędkości kątowe, które dosyć łatwo osiągają zadane mu kąty $\mathbf{S}_{t+1}$.

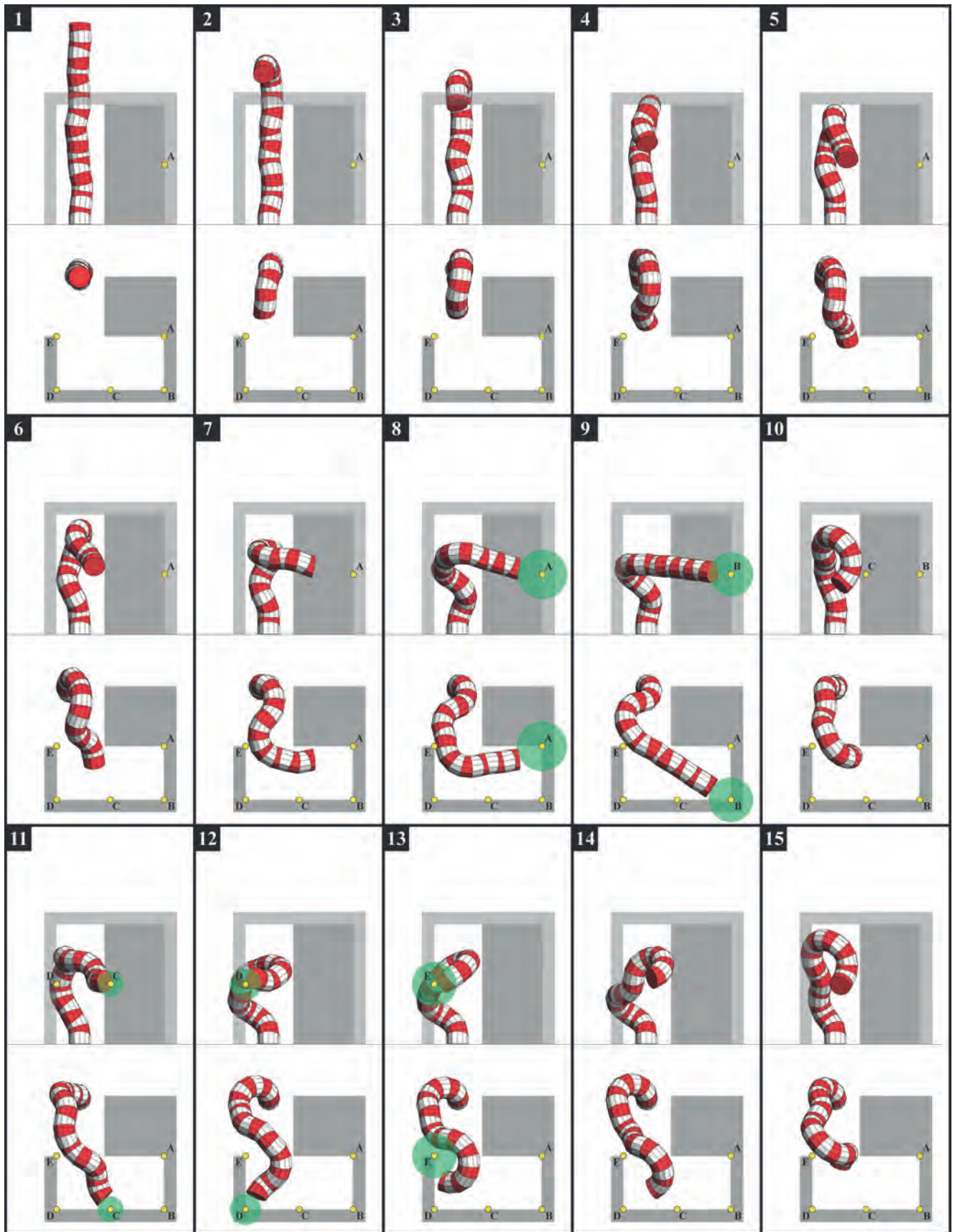
4. Eksperyment numeryczny

Do eksperymentów użyto analogicznego planarnego zadania inspekcyjnego opisanego w [1]. W obu przypadkach manipulator wchodzi w zadany obszar testowy i zbliża końcówkę do pięciu zadanych punktów. Zasadnicza różnica polega na tym, że w niniejszym artykule manipulator Arm-Z dokonuje inspekcji nie na płaszczyźnie, lecz w przestrzeni trójwymiarowej.

Tak jak w przypadku planarnym, tutaj też manipulator Arm-Z składa się z 24 identycznych modułów, co przy nieruchomym module bazowym daje 23 stopnie swobody – w tym przypadku (ciągłe) kąty względnych obrotów poszczególnych modułów.

W przypadku zaprezentowanym na Rys. 6 zdefiniowano przejścia między kolejno następującymi po sobie punktami: $\mathbf{p}_A = [-15/2 + \rho; 5 + \rho; 5]$, $\mathbf{n}_A = [-1; 0; 0]$, $\mathbf{p}_B = [-15/2 + \rho; 10 - \rho; 5]$, $\mathbf{n}_B = [-0,7; 0,7; 0]$, $\mathbf{p}_C = [-2,5; 10 - \rho; 5]$, $\mathbf{n}_C = [0; 10]$, $\mathbf{p}_D = [2,5 - \rho; 10 - \rho; 5]$, $\mathbf{n}_D = [0,7; 0,7; 0]$, $\mathbf{p}_E = [1,5; 6; 5]$, $\mathbf{n}_E = [1; 0; 0]$, ρ to margines wynikający z tego, że moduł nie może fizycznie dotrzeć do samego punktu będącego stykiem dwóch prostopadłych ścian; przyjęto $\rho = 0,8$.

Środowisko dla opisywanego eksperymentu jest określone przez geometrię danej przestrzeni i lokalizację punktów, które podlegają inspekcji, a także umiejscowienie modułu bazowego manipulatora. Manipulator w stanie spoczynku jest umieszczony obok przestrzeni testowej. Na Rys. 6.1 pokazano widok



Rys. 6. Wybrane fazy pokazujące działanie 24-modułowego hiper-redundantnego manipulatora Arm-Z. Na każdym widoku pokazano rzut z przodu oraz odpowiadający mu rzut z góry. Widok 1 pokazuje pozycję wyjściową. Następnie, odpowiednie widoki pokazują inspekcję pięciu kolejnych punktów: 8- p_A , 9- p_B , 11- p_C , 12- p_D , 13- p_E . Widoki 14 i 15 pokazują w skróconej formie proces opuszczania przestrzeni testowej przez manipulator

Fig. 6. Selected stages showing the operation of the 24-module hyper-redundant Arm-Z manipulator. Each view shows: a front and a corresponding top views. View 1 shows the starting position. Next, the corresponding views show the inspection of five consecutive points: 8- p_A , 9- p_B , 11- p_C , 12- p_D , 13- p_E . Views 14 and 15 show selected stages of the manipulator leaving the test space

z przodu, natomiast poniżej korespondujący z nim widok z góry dla sytuacji wyjściowej.

Na tym etapie eksperymentu manipulator tworzy względnie prostą rurę. Zadaniem początkowym dla manipulatora jest bezkolizyjne wprowadzenie końcówki do przestrzeni testowej (Rys. 6.2–6). Następnie końcówka manipulatora zbliża się na zadaną odległość (5 mm) z odpowiednim ustawieniem wektora normalnego kolejno do pięciu zadanych punktów (A–E), co pokazano na (Rys. 6.7–13). W ostatnim etapie zadaniem manipulatora jest bezkolizyjne opuszczenie z przestrzeni testowej, która jest w znacznym stopniu odwróceniem procesu wprowadzania. Na podstawie przeprowadzonego eksperymentu przygotowano animację całej symulacji [39].

5. Podsumowanie

W artykule przedstawiono autorską koncepcję sterowania hiper-redundantnym manipulatorem Arm-Z.

Efektywność zaproponowanej metody zademonstrowano przez symulację zadania, jakim jest bezkolizyjne wejście 24-modułowego manipulatora Arm-Z do niewielkiej ograniczonej przestrzeni w celu dokonania inspekcji kilku zadanych punktów. Zadanie to jest realizowane przez bezkolizyjne zbliżenie końcówki manipulatora do poszczególnych punktów pod odpowiednim kątem z zadaną dokładnością.

Wykazano, że mimo niezwykle prostej budowy, za pomocą zaproponowanego algorytmu manipulator Arm-Z może być sterowany w celu wykonania stosunkowo skomplikowanej czynności w przestrzeni trójwymiarowej. Proponowane rozwiązanie jest konkurencyjne względem wcześniejszych metod [40].

Kolejnym naturalnym krokiem badań będzie wykonanie fizycznego demonstratora i opracowanie realistycznego modelu w środowisku symulacyjnym, który będzie jemu odpowiadał w jak najwierniejszy sposób. Ponadto omówiony algorytm generowanie ścieżki nie gwarantuje, że układ nie utknie w lokalnym minimum, co może zdarzyć się w przypadku bardziej skomplikowanych środowisk. Planuje się rozwiązać to zagadnienie bazując na planowaniu metodami próbkowania (ang. *Sampling-based path planning*).

Podziękowania

Opisane badania stanowią część projektu „Arm-Z: ekstremalnie modularny hiperredundantny ekonomiczny manipulator – opracowanie metod sterowania oraz analiza efektywności” finansowanego przez Narodowe Centrum Nauki w ramach programu OPUS 17, kontrakt nr 2019/33/B/ST8/02791.

Bibliografia

- Zawidzka E., Zawidzki M., *Discrete (Two-State) Modular Hyper-Redundant Planar Manipulator*, „Computer Assisted Methods in Engineering and Science”, Vol. 29, No. 4, 2022, 397–407, DOI: 10.24423/comes.550.
- Gray J., *The mechanism of locomotion in snakes*, „Journal of Experimental Biology”, Vol. 23, No. 2, 1946, 101–120, DOI: 10.1242/jeb.23.2.101.
- Hirose S., *Biologically Inspired Robots: Snake-Like Locomotors and Manipulators*, Oxford University Press, 1993.
- Hirose S., *Design of hyper-redundant arm*, Robotics Research: The Seventh International Symposium, 1996, 548–557, DOI: 10.1007/978-1-4471-0765-1_61.
- Ijspeert A.J., Crespi A., *Online trajectory generation in an amphibious snake robot using a lamprey-like central pattern generator model*, Proceedings of the 2007 IEEE International Conference on Robotics and Automation, 262–268, DOI: 10.1109/ROBOT.2007.363797.
- Chirikjian G.S., Burdick J.W., *Design and experiments with a 30 DOF robot*, Proceedings IEEE International Conference on Robotics and Automation, 1993, DOI: 10.1109/ROBOT.1993.291862.
- Klaassen B., Paap K.L., *GMD-SNAKE2: a snake-like robot driven by wheels and a method for motion control*, Proceedings 1999 IEEE International Conference on Robotics and Automation, Vol. 4, 1999, 3014–3019, DOI: 10.1109/ROBOT.1999.774055.
- Miller G.S.P., *Snake robots for search and rescue*, [In:] Neurotechnology for Biomimetic Robots, 2002, 271–284, DOI: 10.7551/mitpress/4962.003.0023.
- Choi H.R., Ryew S.M., *Robotic system with active steering capability for internal inspection of urban gas pipelines*, „Mechatronics”, Vol. 12, No. 5, 2002, 713–736, DOI: 10.1016/S0957-4158(01)00022-8.
- Tsakiris D.R., Sfakiotakis M., Menciassi A., La Spina G., Dario P., *Polychaete-like undulatory robotic locomotion*, Proceedings of the 2005 IEEE International Conference on Robotics and Automation, 2005, 3018–3023, DOI: 10.1109/ROBOT.2005.1570573.
- Melsaac K.A., Ostrowski J.P., *A geometric approach to anguilliform locomotion: modelling of an underwater eel robot*, Proceedings 1999 IEEE International Conference on Robotics and Automation, Vol. 4, 1999, 2843–2848, DOI: 10.1109/ROBOT.1999.774028.
- Wilbur C., Vorus W., Cao Y., Currie S.N., *A lamprey-based undulatory vehicle*, [In:] Neurotechnology for Biomimetic Robots, 2002, 285–296, DOI: 10.7551/mitpress/4962.003.0024.
- Li B., Ma S., Ye C., Yu S., Zhang G., Gong H., *Development of an amphibious snake-like robot*, 8th World Congress on Intelligent Control and Automation, China, 2010, DOI: 10.1109/WCICA.2010.5553836.
- Crespi A., Badertscher A., Guignard A., Ijspeert A.J., *AmphiBot I: an amphibious snake-like robot*, „Robotics and Autonomous Systems”, Vol. 50, No. 4, 2005, 163–175, DOI: 10.1016/j.robot.2004.09.015.
- Chirikjian G.S., Burdick J.W., *An obstacle avoidance algorithm for hyper-redundant manipulators*, Proceedings IEEE International Conference on Robotics and Automation, 1990, 625–631, DOI: 10.1109/ROBOT.1990.126052.
- Siciliano B., Khatib O., *Springer Handbook of Robotics*, Springer, 2008, DOI: 10.1007/978-3-540-30301-5.
- Murray R.M., Li Z., Sastry S.S., *A Mathematical Introduction to Robotic Manipulation*, CRC Press, 1994, DOI: 10.5555/561828.
- Rolf M., Steil J.J., *Efficient exploratory learning of inverse kinematics on a bionic elephant trunk*, „IEEE Transactions on Neural Networks and Learning Systems”, Vol. 25, No. 6, 2014, 1147–1160, DOI: 10.1109/TNNLS.2013.2287890.
- Melingui A., Escande C., Benoudjit N., Merzouki R., Mbeye J.B., *Qualitative approach for forward kinematic modeling of a Compact Bionic Handling Assistant trunk*, „IFAC Proceedings Volumes”, Vol. 47, No. 3, 2014, 9353–9358, DOI: 10.3182/20140824-6-ZA-1003.01758.
- Falkenhahn V., Hildebrandt A., Neumann R., Sawodny O., *Dynamic Control of the Bionic Handling Assistant*, „IEEE/ASME Transactions on Mechatronics”, Vol. 22, No. 1, 2017, 6–17, DOI: 10.1109/TMECH.2016.2605820.
- Galicki M., *A closed solution to the inverse kinematics of redundant manipulators*, „Mechanism and Machine Theory”, Vol. 26, No. 2, 1991, 221–226, DOI: 10.1016/0094-114X(91)90085-I.
- Jacak W., *A discrete kinematic model of robots in the Cartesian space*, „IEEE Transactions on Robotics and Automation”, Vol. 5, No. 4, 1989, 435–443, DOI: 10.1109/70.88058.

23. Jacak W., *Strategies of searching for collision-free manipulator motions: automata theory approach*, "Robotica", Vol. 7, No. 2, 1989, 129–138, DOI: 10.1017/S0263574700005439.
24. Wang J., *Formal methods in computer science*, Chapman and Hall/CRC, 2019, DOI: 10.1201/9780429184185.
25. Galicki M., Morecki A., *Finding collision-free trajectory for redundant manipulator by local information available*, RoManSy 9, 1993, 61–71, DOI: 10.1007/BFb0031432.
26. Chirikjian G.S., Burdick J.W., *Hyper-redundant robot mechanisms and their applications*, Proceedings IROS '91 of IEEE/RSJ International Workshop on Intelligent Robots and Systems, DOI: 10.1109/IROS.1991.174447.
27. Chirikjian G.S., Burdick J.W., *A hyper-redundant manipulator*, "IEEE Robotics & Automation Magazine", Vol. 1, No. 4, 1994, 22–29, DOI: 10.1109/100.388263.
28. Kimura S., Tsuchiya S., Takegai T., Iizuka N., Moritani K., Nishida S., Kawashima N., Okamura R., *Hyper redundant modular manipulator arm*, 2000 26th Annual Conference of the IEEE Industrial Electronics Society. IECON 2000. 2000 IEEE International Conference on Industrial Electronics, Control and Instrumentation. 21st Century Technologies, Vol. 4, 2000, 2467–2472, DOI: 10.1109/IECON.2000.972384.
29. Zawidzki M., Nagakura T., *Arm-Z: a modular virtual manipulative*, [In:] H.-P. Schröcker, editor, Proceedings of the 16th International Conference on Geometry and Graphics, 2014, 75–80.
30. Hollerbach J., Suh K., *Redundancy resolution of manipulators through torque optimization*, IEEE Journal on Robotics and Automation, Vol. 3, No. 4, 1987, 308–316, DOI: 10.1109/JRA.1987.1087111.
31. Shamir T., Yomdin Y., *Repeatability of redundant manipulators: Mathematical solution of the problem*, "IEEE Transactions on Automatic Control", Vol. 33, No. 11, 1988, 1004–1009, DOI: 10.1109/9.14412.
32. Baillieul J., *Kinematic programming alternatives for redundant manipulators*, Proceedings of 1985 IEEE International Conference on Robotics and Automation, Vol. 2, 1985, 722–728, DOI: 10.1109/ROBOT.1985.1087234.
33. Wang Z., Kazerooni K., *An efficient algorithm for global optimization in redundant manipulations*, "Journal of Mechanical Design", Vol. 111, No. 4, 1989, 488–493, DOI: 10.1115/1.3259026.
34. Suh K., Hollerbach J., *Local versus global torque optimization of redundant manipulators*, Proceedings of 1987 IEEE International Conference on Robotics and Automation, Vol. 4, 1987, 619–624, DOI: 10.1109/ROBOT.1987.1087955.
35. Ma O., Angeles J., *The concept of dynamic isotropy and its applications to inverse kinematics and trajectory planning*, Proceedings of IEEE International Conference on Robotics and Automation, 1990, 481–486, DOI: 10.1109/ROBOT.1990.126024.
36. Chirikjian G.S., Burdick J.W., *Kinematically optimal hyper-redundant manipulator configurations*, "IEEE Transactions on Robotics and Automation", Vol. 11, No. 6, 1995, 794–806, DOI: 10.1109/70.478427.
37. Konietschke R., Hirzinger G., *Inverse kinematics with closed form solutions for highly redundant robotic systems*, 2009 IEEE International Conference on Robotics and Automation, 2009, 2945–2950, DOI: 10.1109/ROBOT.2009.5152706.
38. Coumans E., Bai Y., *Pybullet, a python module for physics simulation for games, robotics and machine learning*, <http://pybullet.org>, 2016–2021.
39. Zawidzka E., Szklarski J., Zawidzki M., *Animation: Arm-Z inspects 5 points inside the box*, 2024, [<https://youtu.be/8ho7o-JeP2c>].
40. Zawidzki M., Szklarski J., *Transformations of Arm-Z modular manipulator with Particle Swarm Optimization*, "Advances in Engineering Software", Vol. 126, 2018, 147–160, DOI: 10.1016/j.advengsoft.2018.05.003.

Simulation of Spatial Inspection with Arm-Z – the Extremely Modular Hyper-Redundant Manipulator

Abstract: Arm-Z is a concept of a hyper-redundant robotic manipulator composed of congruent modules with one degree of freedom (1-DOF) each. The end-effector of Arm-Z is capable of performing any movements in the workspace. The two main advantages of Arm-Z are: economic (due to the potential mass production of the modules) and robustness (since the broken modules can be easily replaced). On the other hand, controlling such a highly non-linear system is much more difficult than a typical industrial manipulator and requires the use of appropriate optimization techniques. This paper presents the preliminary results of the implementation of an algorithm based on the simple gradient method for finding the transition trajectory between the given states of Arm-Z. An example task for the manipulator is presented: Arm-Z is to safely enter the workspace, inspect the five points given, and safely leave the workspace. This experiment is a three-dimensional version of the previously implemented two-dimensional inspection.

Keywords: Arm-Z, hyper-redundant manipulator, inspection

dr Jacek Szklarski

jszklar@ippt.pan.pl
ORCID: 0000-0003-1541-9878

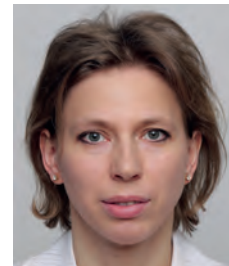
Kierownik Pracowni Systemów Inteligentnych w IPPT PAN. Doktorat w dziedzinie nauk fizycznych uzyskał na Uniwersytecie w Poczdamie. Aktualnie jego główne obszary zainteresowań to wizja maszynowa i zastosowanie metod sztucznej inteligencji w robotyce.



mgr inż. Ela Zawidzka

zawidzka@ippt.pan.pl
ORCID: 0000-0003-1243-9355

Od 10/2020 szkoła doktorska Instytutu Podstawowych Problemów Techniki Polskiej Akademii Nauk. Temat badań: Metody sterowania hiperredundantnym modularnym manipulatorem. W latach 2009–2013 studium doktoranckie na Wydziale Inżynierii Materiałowej, Politechnika Warszawska. Temat badań: Stopy o wysokiej entropii. W latach 2003–2008 studia magisterskie na Wydziale Inżynierii Materiałowej PW. Temat pracy: Wytwarzanie i charakteryzacja stopów z układu Ni-Zr-Ti z dodatkiem Ag, B, Cu lub Nb.



dr hab. inż. Tomasz Ponikiewski

Tomasz.Ponikiewski@polsl.pl
ORCID: 0000-0003-0535-2454

Od 2016 r. profesor nadzwyczajny w Politechnice Śląskiej. Stopień naukowy doktora habilitowanego uzyskał w 2015 r. w dyscyplinie Inżynieria lądowa i transport na podstawie cyklu publikacji „Analiza właściwości i identyfikacja struktur fibrobetonów samozagęszczalnych w wybranych elementach modelowych”. Zakres zainteresowań naukowo-badawczych obejmuje: technologię betonów, zwłaszcza nowej generacji (betony wysokowartościowe, samozagęszczalne i ze zbrojeniem rozproszonym), reologię mieszanek na spoiwach cementowych, technologię druku 3D z mieszanek na spoiwach cementowych i pokrewnych oraz zastosowanie robotyki w technologiach budowlanych.



dr hab. inż. Machi Zawidzki

zawidzki@piap.lukasiewicz.gov.pl
ORCID: 0000-0001-8695-4400

Od 10/2022 kierownik ZBI – Dział Wsparcia Badań Aplikacyjnych w Sieci Badawczej Łukasiewicz – Przemysłowym Instytucie Automatyki i Pomiarów PIAP. Od 2/2017 adiunkt w Zakładzie Technologii Inteligentnych w Instytucie Podstawowych Problemów Techniki Polskiej Akademii Nauk. 7/2020 habilitacja w dziedzinie Nauk Technicznych, w dyscyplinie – Informatyka. Tytuł autoreferatu: Zastosowania metod inteligencji obliczeniowej do optymalizacji architektonicznych Systemów Ekstremalnie Modularnych. W latach 10/2007 – 09/2010 studia doktoranckie na Uniwersytecie Ritsumeikan w Japonii. Temat rozprawy: Application of Computational Intelligence to engineering design problems in Architecture – firmitatis, utilitatis, venustatis.

