

doi:10.15199/48.2022.07.07

# Analiza mocy obliczeniowej platform sprzętowych dla algorytmu sterowania robota hybrydowego

**Streszczenie.** W artykule opisano konstrukcję pięcioosiowego robota hybrydowego składającego się z manipulatora równoległego i szeregowego, do budowy których, wykorzystano napędy PMSM. Zaprezentowano przyjęty model dynamiki robota oraz rozwiązanie zadania odwrotnego kinematyki wraz z pochodnymi. W wybranych platformach sprzętowych, typu FPGA i DSP, zaimplementowano algorytm śledzenia trajektorii układu sterowania robotem. Wyniki przeprowadzonych testów pozwoliły na analizę mocy obliczeniowej układów, i ich porównanie.

**Abstract.** The article describes the assembly of a five-axis hybrid robot consisting of the parallel and the serial manipulator, for the construction of which Permanent Magnet Synchronous Motor (PMSM) were used. The adopted model of robot dynamics and the solution of the task of inverse kinematics with derivatives are presented. In selected hardware platforms, such as FPGA and DSP, the trajectory tracking algorithm of the robot control system has been implemented. The results of the tests carried out enabled the analysis of the computing power of the systems and their comparison. (The assembly of a five-axis hybrid robot consisting of the parallel and the serial manipulator)

**Słowa kluczowe:** układu sterowania ze sprzężeniem w przód metodą wyliczanych momentów CTM, napędy PMSM, układy FPGA, DSP.  
**Keywords:** the trajectory tracking control algorithm of the hybrid robot, PMSM motors, FPGA, DSP.

## Wstęp

Ciągle podnoszenie wymagań technologicznych przez przemysł wytwórczy, wobec maszyn numerycznych, dotyczący możliwości wykonywania precyzyjnych działań (dokładnego odwzorowywania zadanego toru ruchu lub/i pozycjonowania) sprawia, że poszukiwanie rozwiązań w obszarze projektowania konstrukcji, napędów i układów sterowania jest aktualnym trendem prowadzenia badań.

Maszyny sterowane numerycznie są urządzeniami mechatronicznymi, łączącymi elementy elektroniczne i mechaniczne z systemami sterowania i programowania [1]. Parametry poszczególnych elementów i ich poziom synergii determinują możliwości techniczne tego typu maszyn.

Przykładem aktualnego podejścia w projektowaniu maszyn może być użycie napędów typu PMSM (ang. Permanent Magnet Synchronous Motor), oddziałujących bezpośrednio (bez reduktorów mechanicznych) na pozycję efektora robota i zastosowanie układu sterowania ze sprzężeniem w przód (ang. feedforward) [2] metodą wyliczanych momentów – CTM (ang. Computed torque method) [2], [3], [4]. Rozwiązanie to pozwala na znaczącą redukcję wpływu zakłóceń wywołanych interakcjami, co przekłada się na wysoką dokładność odwzorowania zadanej trajektorii [3].

W zastosowaniu metody wyliczanych momentów konieczne jest opracowanie modelu dynamiki manipulatora, który wymaga dużych nakładów obliczeniowych. Ponadto algorytmy: generowania zadanej trajektorii i śledzenia trajektorii, powinny być wykonywane w pojedynczym cyklu pracy sterownika (tzn. w okresie próbkowania) [4].

Wykorzystanie złożonego modelu powoduje problem związany z wpływem błędów obliczeń numerycznych na poziom zakłóceń w układzie sterowania. Ograniczenie wielkości tych błędów można uzyskać poprzez zastosowanie odpowiedniego typu kodowania liczb w programowym opisie algorytmów oraz implementację analitycznej postaci równań matematycznych modelu.

## Robot hybrydowy

Opracowany pięcioosiowy robot hybrydowy, pokazany na rys. 1 i 2, jest przeznaczony przede wszystkim do obróbki materiałów (wiercenia, frezowania, toczenia, cięcia laserowego i cięcia strumieniem wody), ale może być używany, po wyposażeniu w odpowiednie narzędzie, również do prac manipulacyjnych i montażowych.



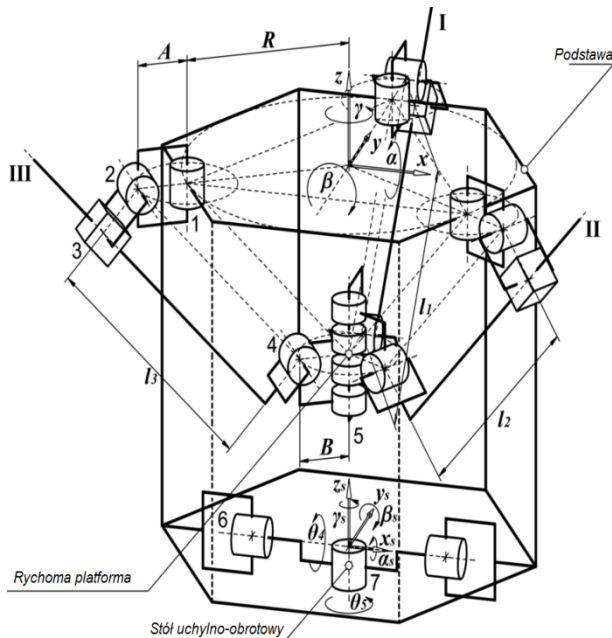
Rys.1. Prototyp pięcioosiowego robota hybrydowego

Robot hybrydowy składa się z dwóch manipulatorów: równoległego trójramiennego oraz szeregowego w postaci stołu uchylno-obrotowego, zamontowanego w podstawie manipulatora równoległego.

Manipulator równoległy, o trzech translacyjnych stopniach swobody, ma strukturę będącą kompromisem pomiędzy dwoma dominującymi trendami kinematyki równoległej: wywodzącą się z robota Delta i platformy Stewarta-Gougha. Charakteryzuje się stosunkowo dużą przestrzenią roboczą, dużą obciążalnością oraz osiągalnymi dużymi przyspieszeniami. Manipulator ma trzy ramiona przymocowane do podstawy i połączone wspólnym przegubem potrójnym (rys. 2).

Główny przegub potrójny służy jako ruchoma platforma, do której mocowane jest narzędzie (np. elektrowrzeciono). Każde z ramion robota tworzy łańcuch kinematyczny o strukturze RRP<sub>RR</sub> (rys. 2, złącza: 1,2,3,4,5), w którym przegub przyrządkowy (rys. 2, złącze 3) jest napędzany przez liniowy napęd bezpośredni typu PMSM, a jego długość (rys.2,  $l_i$  –  $i:1,2,3$ ) mierzona jest za pomocą

liniowego, optycznego, przyrostowego czujnika pozycji o rozdzielczości 0,25  $\mu\text{m}$ . Biegnik prowadzony jest przez prowadnice liniowe przymocowane za pomocą przegubów obrotowych do wspólnej podstawy. Dwa złącza łączące prowadnice liniowe z podstawą oraz elementy potrójnego złącza zostały skonstruowane w taki sposób, że składają się z połączeń o tylko jednym stopniu swobody. Taka konstrukcja zapewnia możliwość łatwego wykonania, montażu i uzyskania wymaganej dokładności.



Rys.2. Schemat kinematyczny pięcioosiowego robota hybrydowego

Stół uchylny-obrotowy (rys. 2) składa się z podstawy, części uchylniej i stołu obrotowego. Część uchylna jest zamontowana w podstawie i może być wychylana o  $\pm 180^\circ$  od poziomej pozycji stołu. W części uchylniej montowany jest stół obrotowy  $\varnothing 400$  mm z nieograniczonym kątem obrotu. Obie osie są napędzane przez obrotowe napędy bezpośrednie PMSM, a położenie katowe mierzone jest za pomocą optycznych, absolutnych, czujników położenia katowego o rozdzielczości 18 bitów.

Docelowa dokładność opracowanego robota wynosi 10  $\mu\text{m}$ . Aby zapewnić tak wysoką skuteczność sterowania, maksymalna odległość pomiędzy dwoma kolejnymi punktami trajektorii musi być co najmniej dwa razy mniejsza. W typowych procesach obróbki (np. aluminium HSM) posuw wynosi 2 m/min. Dlatego ustalona częstotliwość próbkowania układu sterowania wynosi 10 kHz. Skutkuje to odległością między dwoma kolejnymi punktami trajektorii równą 3,3  $\mu\text{m}$  lub 1/3 wymaganej dokładności. Z tego samego powodu dokładność obliczeń generowania trajektorii i transformacji musi zapewniać całkowity błąd mniejszy niż powyższa założona wartość. Taką dokładność można osiągnąć dzięki zastosowaniu silników synchronicznych z magnesami trwałymi (PMSM), które są coraz częściej stosowane w wielu dziedzinach sterowania automatyką jako elementy wykonawcze [5].

### Kinematyka robota

Jako globalny (zewewnętrzny) układ współrzędnych przyjęto prawoskrętny kartezjański układ współrzędnych, skojarzony z bazą robota, którego początek znajduje się na przecięciu osi obrotu stołu uchylny-obrotowego (rys. 2). Oś Z jest skierowana pionowo w górę, a oś Y przechodzi przez oś obrotu złącza 1 w ramieniu I (rys. 2).

Manipulator równoległy posiada trzy identyczne ramiona I, II, III, działające bezpośrednio na ruchomą platformę (rys. 2). Każde ramię tworzy łańcuch kinematyczny składający się z przegubów obrotowych 1, 2, 4 i 5 oraz przegubu pryzmatycznego 3. Początek łańcucha jest przymocowany do podstawy, a koniec ma bezpośredni wpływ na położenie platformy.

Długość przegubu pryzmatycznego 3, oznaczona jako  $l_i$ , zawiera się pomiędzy punktem  $(x_{pi}, y_{pi}, z_p=0)$ , będącym przecięciem osi symetrii przegubu 1 z płaszczyzną XY o współrzędnej  $z=0$ , a środkiem platformy o współrzędnych  $(x, y, z)$ .

W opisywanej konstrukcji odległości A i B są równe [22], co znacznie upraszcza równania kinematyki i pozwala uzyskać rozwiązanie problemów kinematyki w postaci analitycznej. Wówczas wielkość pomocnicza  $K_i$ , będąca rzutem  $l_i$  na płaszczyznę XY, pozwala powiązać współrzędną z platformy z długością  $l_i$ :

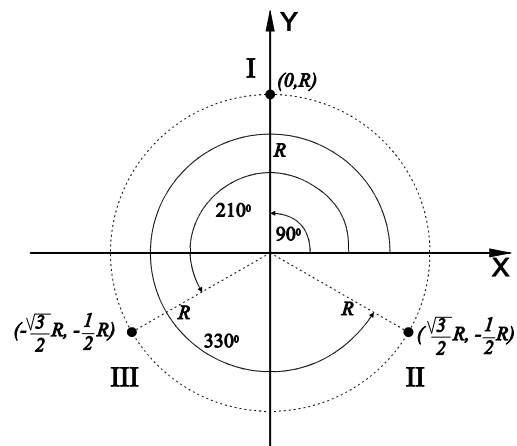
$$(1) \quad l_i^2 = z^2 + K_i$$

Odległość  $K_i$  wyraża się jako:

$$(2) \quad K_i = (x - x_{pi})^2 + (y - y_{pi})^2$$

Po podstawieniu (2) do (1) otrzymujemy równanie (3), które opisuje długość  $l_i$  w zależności od położenia platformy  $(x, y, z)$  oraz parametrów  $x_{pi}, y_{pi}$ , które są współrzędnymi początku ramienia:

$$(3) \quad l_i = \sqrt{(x - x_{pi})^2 + (y - y_{pi})^2 + z^2}$$



Rys.3. Schemat z punktami mocowania ramion manipulatora równoległego w płaszczyźnie XY – rzut z góry

Manipulator posiada trzy ramiona przymocowane do podstawy na promieniu R, zgodnie z rys. 8. Współrzędne  $(x_p, y_p)$  dla wszystkich ramion są równe:

$$(4) \quad x_{p1} = R \cdot \cos(90^\circ) = 0, \quad y_{p1} = R \cdot \sin(90^\circ) = R$$

$$(5) \quad x_{p2} = R \cdot \cos(330^\circ) = \frac{\sqrt{3}}{2} R,$$

$$y_{p2} = R \cdot \sin(330^\circ) = -\frac{1}{2} R$$

$$(6) \quad x_{p3} = R \cdot \cos(210^\circ) = -\frac{\sqrt{3}}{2} R,$$

$$y_{p3} = R \cdot \sin(210^\circ) = -\frac{1}{2} R$$

Po podstawieniu (4)-(6) do (3) otrzymujemy układ równań (7), który jest rozwiązaniem problemu kinematyki odwrotnej dla manipulatora równoległego.

$$(7) \quad l_i = \sqrt{x^2 + (y - R)^2 + z^2},$$

$$l_2 = \sqrt{\left(x - \frac{\sqrt{3}}{2}R\right)^2 + \left(y + \frac{1}{2}R\right)^2 + z^2},$$

$$l_3 = \sqrt{\left(x + \frac{\sqrt{3}}{2}R\right)^2 + \left(y + \frac{1}{2}R\right)^2 + z^2}$$

Różniczkując rozwiązanie zagadnienia kinematyki odwrotnej (7) otrzymujemy równania prędkości (8) i przyspieszenia (9)-(11) w układzie współrzędnych złączowych. Równania (7)-(11) pozwalają przekształcić trajektorie manipulatora ze współrzędnych globalnych, w których jest generowana, we współrzędne złączowe, w których jest realizowana.

$$(8) \quad v_1 = \frac{1}{2} \frac{2 \cdot x \cdot v_x + 2 \cdot (y-R) \cdot v_y + 2 \cdot z \cdot v_z}{\sqrt{x^2 + (y-R)^2 + z^2}},$$

$$v_2 = \frac{1}{2} \frac{2 \cdot \left(x - \frac{\sqrt{3}}{2}R\right) \cdot v_x + 2 \cdot \left(y + \frac{1}{2}R\right) \cdot v_y + 2 \cdot z \cdot v_z}{\sqrt{\left(x - \frac{\sqrt{3}}{2}R\right)^2 + \left(y + \frac{1}{2}R\right)^2 + z^2}},$$

$$v_3 = \frac{1}{2} \frac{2 \cdot \left(x + \frac{\sqrt{3}}{2}R\right) \cdot v_x + 2 \cdot \left(y + \frac{1}{2}R\right) \cdot v_y + 2 \cdot z \cdot v_z}{\sqrt{\left(x + \frac{\sqrt{3}}{2}R\right)^2 + \left(y + \frac{1}{2}R\right)^2 + z^2}}$$

$$(9) \quad a_1 = -\frac{1}{4} \frac{(2 \cdot x \cdot v_x + 2 \cdot (y-R) \cdot v_y + 2 \cdot z \cdot v_z)^2}{(x^2 + (y-R)^2 + z^2)^{\frac{3}{2}}} + \frac{1}{2} \cdot \frac{2 \cdot v_x^2 + 2 \cdot x \cdot a_x + 2 \cdot v_y^2 + 2 \cdot (y-R) \cdot a_y + 2 \cdot v_z^2 + 2 \cdot z \cdot a_z}{\sqrt{x^2 + (y-R)^2 + z^2}}$$

$$(10) \quad a_2 = -\frac{1}{4} \frac{(2 \cdot \left(x - \frac{\sqrt{3}}{2}R\right) \cdot v_x + 2 \cdot \left(y + \frac{1}{2}R\right) \cdot v_y + 2 \cdot z \cdot v_z)^2}{\left(\left(x - \frac{\sqrt{3}}{2}R\right)^2 + \left(y + \frac{1}{2}R\right)^2 + z^2\right)^{\frac{3}{2}}} + \frac{1}{2} \cdot \frac{2 \cdot v_x^2 + 2 \cdot \left(x - \frac{\sqrt{3}}{2}R\right) \cdot a_x + 2 \cdot v_y^2 + 2 \cdot \left(y + \frac{1}{2}R\right) \cdot a_y + 2 \cdot v_z^2 + 2 \cdot z \cdot a_z}{\sqrt{\left(x - \frac{\sqrt{3}}{2}R\right)^2 + \left(y + \frac{1}{2}R\right)^2 + z^2}}$$

$$(11) \quad a_3 = -\frac{1}{4} \frac{(2 \cdot \left(x + \frac{\sqrt{3}}{2}R\right) \cdot v_x + 2 \cdot \left(y + \frac{1}{2}R\right) \cdot v_y + 2 \cdot z \cdot v_z)^2}{\left(\left(x + \frac{\sqrt{3}}{2}R\right)^2 + \left(y + \frac{1}{2}R\right)^2 + z^2\right)^{\frac{3}{2}}} + \frac{1}{2} \cdot \frac{2 \cdot v_x^2 + 2 \cdot \left(x + \frac{\sqrt{3}}{2}R\right) \cdot a_x + 2 \cdot v_y^2 + 2 \cdot \left(y + \frac{1}{2}R\right) \cdot a_y + 2 \cdot v_z^2 + 2 \cdot z \cdot a_z}{\sqrt{\left(x + \frac{\sqrt{3}}{2}R\right)^2 + \left(y + \frac{1}{2}R\right)^2 + z^2}}$$

Przekształcenie pierwszego równania w (7) daje:

$$(12) \quad z^2 = l_1^2 - x^2 - (y-R)^2$$

Podstawienie (12) do pozostałych równań w (7) daje:

$$(13) \quad l_2 = \sqrt{\left(x - \frac{1}{2}\sqrt{3}R\right)^2 + \left(y + \frac{1}{2}R\right)^2 + l_1^2 - x^2 - (y-R)^2},$$

$$l_3 = \sqrt{\left(x + \frac{1}{2}\sqrt{3}R\right)^2 + \left(y + \frac{1}{2}R\right)^2 + l_1^2 - x^2 - (y-R)^2}$$

Rozwiązanie (13) względem x i y przyjmuje postać:

$$(14) \quad x = -\frac{\sqrt{3}(l_2^2 - l_3^2)}{6R}, \quad y = -\frac{2l_1^2 - l_2^2 - l_3^2}{6R}$$

Za pomocą równań (14) i (12) można wyznaczyć zmienną z:

$$(15) \quad z = \sqrt{l_1^2 - \frac{(l_2^2 - l_3^2)^2}{12R^2} - \left(\frac{2l_1^2 - l_2^2 - l_3^2}{6R} + R\right)^2}$$

Orientacja platformy wokół osi X i Y jest stała – platforma przez cały czas pozostaje w płaszczyźnie poziomej:

$$(16) \quad \alpha = -\pi, \quad \beta = -\pi$$

Natomiast orientacja  $\gamma$  wokół osi Z zmienia się w zależności od długości ramienia robota. Ale ze względu na ciągły obrót narzędzia, równania (14)-(16) stanowią rozwiązanie zagadnienia kinematyki prostej dla manipulatora równoległego.

Konstrukcja kinematyczna manipulatora szeregowego, jakim jest stół uchylny-obrotowy, jest mniej złożona, niż w przypadku manipulatora równoległego. Rozwiązaniem problemu kinematyki prostej zostało sformułowane:

$$(17) \quad x = 0, \quad y = -h \sin \theta_4, \quad z = h \cos \theta_4$$

$$\alpha = \theta_4, \quad \beta = 0, \quad \gamma = \theta_5$$

gdzie:  $h$  – odległość powierzchni stołu od osi przechyłu (X).

Rozwiązanie kinematyki odwrotnej ma postać (18), wraz z pochodnymi pierwszego (19) i drugiego (20) rzędu:

$$(18) \quad \theta_4 = \alpha, \quad \theta_5 = \gamma$$

$$(19) \quad \omega_4 = \omega_\alpha, \quad \omega_5 = \omega_\gamma$$

$$(20) \quad \varepsilon_4 = \varepsilon_\alpha, \quad \varepsilon_5 = \varepsilon_\gamma$$

gdzie:  $\omega_4 = \dot{\theta}_4$  - prędkość kątowna części uchylnego stołu w układzie współrzędnych złączowych,  $\omega_5 = \dot{\theta}_5$  - prędkość kątowna części obrotowej stołu w układzie współrzędnych złączowych,  $\omega_{\alpha_s} = \dot{\alpha}_s$  - prędkość kątowna części uchylnego stołu w układzie współrzędnych globalnych,  $\omega_{\gamma_s} = \dot{\gamma}_s$  - prędkość kątowna części obrotowej stołu w układzie współrzędnych globalnych,  $\varepsilon_4 = \ddot{\theta}_4$  - przyspieszenie kątowne części uchylnego stołu w układzie współrzędnych złączowych,  $\varepsilon_5 = \ddot{\theta}_5$  - przyspieszenie kątowne części obrotowej stołu w układzie współrzędnych złączowych,  $\varepsilon_{\alpha_s} = \ddot{\alpha}_s$  - przyspieszenie kątowne części uchylnego stołu w układzie współrzędnych globalnych,  $\varepsilon_{\gamma_s} = \ddot{\gamma}_s$  - przyspieszenie kątowne części obrotowej stołu w układzie współrzędnych globalnych.

### Model dynamiki

Dominującym źródłem nieliniowości manipulatora równoległego jest jego geometria, powodująca duże zmiany, w obrębie przestrzeni roboczej, zredukowanych mas i sił grawitacyjnych. Struktura modelu, obejmująca liniowe i nieliniowe sprzężenia bezwładnościowe (od sił Coriolisa i odśrodkowych) okazała się zbyt złożona dla celów syntezy sterowania. Dlatego przyjęto szereg uproszczeń: pominięto siły Coriolisa, ze względu na małe prędkości obrotowe ramion oraz, pozostałe nieliniowości poza geometrycznymi, otrzymując równania dynamiki manipulatora w postaci:

$$(21) \quad M(l, x) \ddot{l} + G(l, x) = F$$

gdzie:  $M$  - macierz bezwładności (zredukowanych mas i momentów bezwładności),  $G$  - wektor zredukowanych sił grawitacyjnych,  $F$  - wektor sił generowanych przez napędy,  $l = [l_1, l_2, l_3]^T$  - wektor współrzędnych narzędzia w układzie złączowym,  $x = [x, y, z]^T$  - wektor współrzędnych narzędzia w układzie kartezjańskim  $(x, y, z)$ .

Dalsze uproszczenie polegało na wykorzystaniu do wyznaczenia macierzy  $M$  analizy stanów równowagi (kinetostatycznej) dla chwilowego położenia narzędzia w przestrzeni roboczej. Dało to strukturę układu liniowego odprężonego, opisanego równaniami o zmiennych współczynnikach zależnych od położenia we współrzędnych złączowych, który nadaje się do syntezy sterowania.

Błędy wynikające z pominięcia sprzężeń bezwładnościowych zostały skompensowane w sterowaniu przez człon PID. Jest to podejście uproszczone, ale zweryfikowane eksperymentalnie jako skuteczne. Równanie (21) po przyjęciu powyższego uproszczenia można zapisać w postaci:

$$(22) \begin{bmatrix} M_{11}(\mathbf{l}, \mathbf{x}) & 0 & 0 \\ 0 & M_{22}(\mathbf{l}, \mathbf{x}) & 0 \\ 0 & 0 & M_{33}(\mathbf{l}, \mathbf{x}) \end{bmatrix} \cdot \begin{bmatrix} \ddot{\theta}_1 \\ \ddot{\theta}_2 \\ \ddot{\theta}_3 \end{bmatrix} + \begin{bmatrix} G_1(\mathbf{l}, \mathbf{x}) \\ G_2(\mathbf{l}, \mathbf{x}) \\ G_3(\mathbf{l}, \mathbf{x}) \end{bmatrix} = \begin{bmatrix} F_1 \\ F_2 \\ F_3 \end{bmatrix}$$

Korzystając z zasady d’Lamberta, zredukowane masy manipulatora  $M_{ii}$ , można wyrazić jako funkcje długości członów pryzmatycznych, które są mierzone bezpośrednio. Oznaczając numery ramion przez  $i, j, k$ , elementy macierzy mas są wyliczane na podstawie równania:

$$(23) M_{ij}(\mathbf{l}, \mathbf{x}) = m_s + m_p + m_{rij}(\mathbf{l}, \mathbf{x}) + m_{rik}(\mathbf{l}, \mathbf{x}) + m_{jij}(\mathbf{l}, \mathbf{x}) + m_{jik}(\mathbf{l}, \mathbf{x})$$

$m_s$  – masa przesuwnej części i-tego ramienia,  $m_p$  – masa ruchomej platformy,  $m_r$  – masy pozostałych ramion, zredukowane do osi silnika i-tego ramienia,  $m_j$  – masy wynikające z momentów bezwładności pozostałych ramion, zredukowanych do osi silnika i-tego ramienia.

Masy  $m_r$  i  $m_j$  zależą nieliniowo od kątów pomiędzy ramionami, które mogą być wyrażone jako funkcje pozycji narzędzia w złączowym układzie współrzędnych.

Jednak ze względu na wymagania dotyczące mocy obliczeniowej układu wykonawczego, do implementacji korzystniejsze jest wykorzystanie równocześnie współrzędnych złączowych oraz globalnych. Taka sama sytuacja zachodzi podczas obliczania zredukowanych sił grawitacyjnych  $G_i$  w równaniu (22). Elementy macierzy  $G$  są rozwiązaniami układu równań:

$$(24) \begin{bmatrix} w_{x1}(\mathbf{l}, \mathbf{x}) & w_{x2}(\mathbf{l}, \mathbf{x}) & w_{x3}(\mathbf{l}, \mathbf{x}) \\ w_{y1}(\mathbf{l}, \mathbf{x}) & w_{y2}(\mathbf{l}, \mathbf{x}) & w_{y3}(\mathbf{l}, \mathbf{x}) \\ w_{z1}(\mathbf{l}, \mathbf{x}) & w_{z2}(\mathbf{l}, \mathbf{x}) & w_{z3}(\mathbf{l}, \mathbf{x}) \end{bmatrix} \cdot \begin{bmatrix} G_1 \\ G_2 \\ G_3 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ m_z(\mathbf{l}, \mathbf{x}) \cdot g \end{bmatrix}$$

gdzie:  $w_{xi}, w_{yi}, w_{zi}$  – współczynniki zależne nieliniowo od nachylenia ramion,  $g$  – przyspieszenie ziemskie,  $m_z$  – masa wszystkich ruchomych części manipulatora sprowadzona do osi narzędzia.

Do wyznaczenia wartości współczynników  $w_{xi}, w_{yi}, w_{zi}$  w równaniu (24) konieczna jest znajomość położenia narzędzia w kartezjańskim układzie współrzędnych, które można wyznaczyć z rozwiązania problemu kinematyki prostej.

$$(25) m_z(\mathbf{l}, \mathbf{x}) = m_p + m_{z1}(\mathbf{l}, \mathbf{x}) + m_{z2}(\mathbf{l}, \mathbf{x}) + m_{z3}(\mathbf{l}, \mathbf{x})$$

Składowa  $m_{zi}$  i-tej kończyny zawiera rzut ciężaru ramienia robota na oś biegnika napędu liniowego i jego przeciwwagę, sprowadzonych do osi narzędzia. Innym źródłami niedokładności modelu dynamiki jest uproszczony rozkład masy w ruchomych częściach manipulatora wynikający ze skomplikowanych kształtów elementów oraz brak informacji w przypadku części komercyjnych, takich jak napędy, prowadnice itp. Wszelkie niedoskonałości modelu traktowane są jako zakłócenia w sterowaniu manipulatora i kompensowane przez układ PID.

Zakładając symetrię ruchomych części stołu uchylnobrotowego, równanie dynamiczne manipulatora szeregowego przyjmuje postać:

$$(26) M(\theta)\ddot{\theta} + G(\theta) = \tau$$

gdzie:  $M$  – macierz bezwładności (zredukowanych momentów bezwładności),  $G$  – wektor zredukowanych momentów sił grawitacyjnych,  $\theta = [\theta_4, \theta_5]^T$  – wektor współrzędnych złączowych środka powierzchni stołu.

Równanie (26) można przedstawić w postaci rozszerzonej:

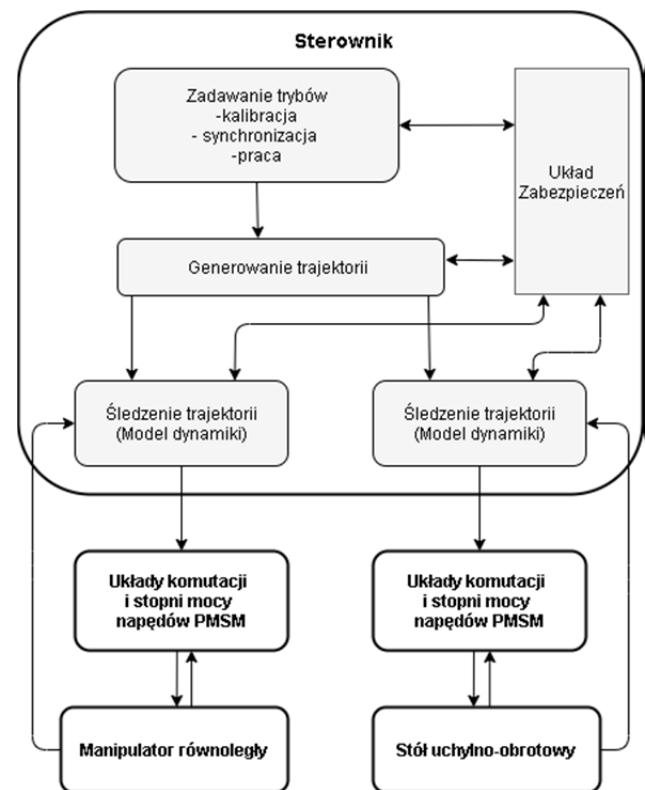
$$(27) \begin{bmatrix} J_{44} & 0 \\ 0 & J_{55} \end{bmatrix} \cdot \begin{bmatrix} \ddot{\theta}_4 \\ \ddot{\theta}_5 \end{bmatrix} + \begin{bmatrix} r_4 m_4 g \cdot \sin \theta_4 \\ 0 \end{bmatrix} = \begin{bmatrix} \tau_4 \\ \tau_5 \end{bmatrix}$$

gdzie:  $J_{44}$  – moment bezwładności części uchylniej (wraz ze stołem obrotowym) względem osi uchyłu,  $J_{55}$  – moment bezwładności części obrotowej stołu względem osi obrotu,  $r_4$  – odległość środka ciężkości części uchylniej (wraz ze stołem obrotowym) od osi uchyłu,  $m_4$  – masa części uchylniej (wraz ze stołem obrotowym), zredukowana do środka ciężkości.

### Architektura sterownika

Ponieważ w konstrukcji manipulatora hybrydowego nie występują sprzężenia mechaniczne między manipulatorem równoległym a manipulatorem szeregowym (stołem uchylnobrotowym), dlatego są one traktowane jak dwa osobne, współpracujące ze sobą urządzenia.

Sterowanie pracą robota można podzielić na kilka głównych części: sterownik, układy komutacji i stopni mocy napędów PMSM (rys.4). Jednym z zadań sterownika, jest dopilnowanie, aby narzędzie podążało za wygenerowaną trajektorią z założoną dokładnością (algorytm śledzenia trajektorii). Należy podkreślić, że sterowanie robotem nałożyło wysokie wymagania na względną (w stosunku do wielkości przestrzeni roboczej) dokładność realizacji ścieżki narzędzia. Aby spełnić te wymagania, generowanie i śledzenie trajektorii odbywa się z tą samą, wysoką częstotliwością próbkowania 10 kHz.



Rys.4. Uproszczony schemat architektury sterownika pięcioosiowego robota hybrydowego

Kolejne zadanie sterownika, nazywane zadawaniem trybów, odpowiada za planowanie pracy robota, unikanie kolizji oraz czasową i przestrzenną synchronizację wykonywanych operacji między sobą oraz z zewnętrznymi urządzeniami lub procesami. Podczas frezowania jest to synchronizacja ruchów manipulatora równoległego z ruchami stołu uchylno-obrotowego, napędu wrzeciona oraz układów chłodzenia.

### Algorytm śledzenia trajektorii

Algorytm śledzenia trajektorii został opracowany w oparciu o metodę wyliczanych momentów (ang. Computed torque method). Charakterystyczną cechą tego typu układu, jest sprzężenie w przód, którego zadaniem jest przewidywanie wymaganych momentów i sił na podstawie opracowanego modelu dynamiki.

Do sterowania położeniem manipulatora równoległego, w celu śledzenia zadanej trajektorii ( $l_d, v_d, a_d$ ) zostało zaproponowane następujące dyskretne prawo sterowania [82], [87]:

$$(28) \quad F(t) = M(l(t), x(t)) [a_d(t) + K_p e(t) + K_v e_v(t) + K_I s_c(t)] + G(l(t), x(t))$$

gdzie:  $K_p, K_v, K_I$  - wektory stałych (wzmocnień sterownika),  $e(t) = l_d(t) - l(t)$  - wektor błędów pozycji,  $e_v(t) = v_d(t) - v(t)$  - wektor błędów prędkości liniowej,  $v(t) = [l(t) - l(t-1)] \frac{1}{T_s}$  - wektor prędkości liniowej,  $s_c(t)$  - wektor błędów pozycji pochodzących od integratora,  $T_s$  - okres próbkowania sterownika.

Badania eksperymentalne przeprowadzone nad układem śledzenia trajektorii, których rezultaty zostały zaprezentowane w publikacjach [11], [85], [86], wykazały że w chwili nawrotu napędu, błąd pozycji (we współrzędnych złączowych), związanej z tym napędem, gwałtownie wzrasta, zmieniając równocześnie znak. Zakumulowana wartość w integratorze ze znakiem przeciwnym, sprzed nawrotu (ang. wind-up), powodowała, że przez pewien czas człon całkujący był źródłem składowej sygnału sterującego, który zwiększał błąd pozycji. Potrzebny był dość długi czas, aby znak wartości całki błędu stał się zgodny z bieżącym znakiem błędu. Dlatego wprowadzono modyfikację członu całkującego, która ograniczyła wartości akumulowane w integratorze oraz przyspieszyła zmianę znaku całki po nawrotach, nie wpływając znacząco na sposób pracy tego członu w pozostałym czasie. Działanie i-tego elementu wektora został przedstawiony w postaci układu równań (29):

$$(29) \quad s_{c,i}(t) = \begin{cases} s_{imax} \leftrightarrow s_{c,i}(t-1) + \varphi_{c,i}(t)e_i(t)T_s > s_{imax} \\ s_{imin} \leftrightarrow s_{c,i}(t-1) + \varphi_{c,i}(t)e_i(t)T_s < s_{imin} \\ s_{c,i}(t-1) + \varphi_{c,i}(t)e_i(t)T_s \leftrightarrow \\ s_{imax} \leq s_{c,i}(t-1) + \varphi_{c,i}(t)e_i(t)T_s \leq s_{imin} \end{cases}$$

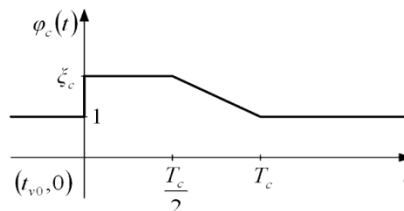
gdzie:  $\varphi_c$  jest nieliniowym współczynnikiem (rys. 5), powodującym zmianę okresu całkowania przez pewien czas po nawrocie napędu.

Jest on określony równaniami (30) i (31):

$$(30) \quad \varphi_{c,i}(t) = \varphi_{t,i}(t) + 1$$

$$(31) \quad \varphi_{t,i}(t) = \begin{cases} 0; & t_{c,i} \leq 0 \\ \xi_c - 1; & 0 < t_{c,i} \leq \frac{T_c}{2} \\ 2(\xi_c - 1) \left(1 - \frac{t_{c,i}}{T_c}\right) \varphi_{e,i}(t); & \frac{T_c}{2} < t_{c,i} < T_c \\ 0; & t_{c,i} \geq T_c \end{cases}$$

gdzie:  $\xi_c$  - maksymalny współczynnik „wzmocnienia” całkowania,  $T_c$  - maksymalny czas „wzmocnienia” całkowania,  $t_{c,i}$  - czas, który upłynął od wykrycia ostatniego nawrotu lub zatrzymania napędu  $i$ ,  $\varphi_{e,i}$  - współczynnik powodujący wcześniejsze wyłączenie „wzmocnienia” całkowania, gdy nie jest ono potrzebne (błąd jest mały).



Rys.5. Wykres w funkcji czasu współczynnika całkowania, w przypadku gdy nie jest on wcześniej wyłączony z powodu małego błędu.

$$(32) \quad \varphi_{e,i}(t) = \begin{cases} 1; & |e_i(t)| \geq e_c \\ 0; & |e_i(t)| < e_c \end{cases}$$

gdzie:  $e_c$  - stała.

$$(33) \quad t_{c,i} = t - t_{v0,i}$$

gdzie  $t_{v0,i}$  jest chwilą czasu (próbka) w której wykryto nawrót lub zatrzymanie napędu  $i$ .

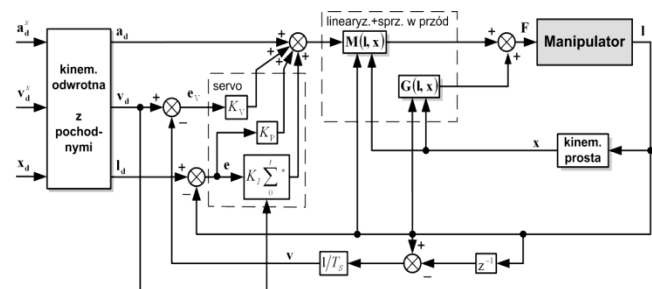
$$(34) \quad t_{v0,i} = \begin{cases} t; & s_{g,i}(t-1) \neq s_{g,i}(t) \\ t_{v0,i}; & s_{g,i}(t-1) = s_{g,i}(t) \end{cases}$$

$$(35) \quad s_{g,i}(t) = \begin{cases} -1; & v_{d,i}(t) < 0 \\ 0; & v_{d,i}(t) = 0 \\ 1; & v_{d,i}(t) > 0 \end{cases}$$

gdzie:  $v_{d,i}$  jest prędkością zadaną we współrzędnych złączowych dla napędu  $i$ .

Pomimo że zapis modyfikacji integratora wydaje się być złożony, to jego implementacja, ze względu na warunkową postać, jest prosta.

Strukturę sterownika pokazuje rys.6. Część linearyzująca ( $M(l,x)+G(l,x)$ ) zawiera zmienne wymagające rozwiązania zagadnienia kinematyki prostej w każdym kroku czasowym. W sterowniku tym model dynamiczny manipulatora pełni rolę predyktora sił potrzebnych do realizacji zadanej trajektorii, a człon PID ma za zadanie kompensację zakłóceń i niedokładności modelu. Sterowanie odbywa się we współrzędnych złączowych, ale sterownik jest przystosowany do pracy z generatorem trajektorii we współrzędnych globalnych, które są transformowane do układu lokalnego przez rozwiązanie odwrotnego kinematyki (7) i jego pochodne, określone związkami (8)-(11).



Rys.6. Schemat blokowy sterowania pozycyjnego manipulatora równoległego

Do sterowania położeniem manipulatora szeregowego (stołu uchylny-obrotowego), w celu śledzenia zadanej trajektorii  $(\theta_d, \omega_d, \varepsilon_d)$  zostało zaproponowane następujące dyskretne prawo sterowania z linearyzacją w pętli sprzężenia zwrotnego [82], [87]:

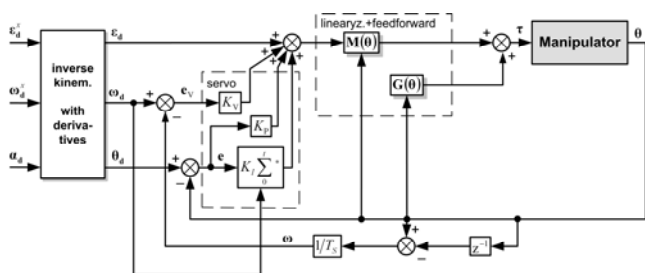
$$(36) \tau(t) = \mathbf{M}(\boldsymbol{\theta}(t))[\boldsymbol{\varepsilon}_d(t) + \mathbf{K}_P \mathbf{e}(t) + \mathbf{K}_V \mathbf{e}_V(t) + \mathbf{K}_I \mathbf{s}_c(t)] + \mathbf{G}(\boldsymbol{\theta}(t))$$

gdzie:  $\mathbf{K}_P, \mathbf{K}_V, \mathbf{K}_I$  - wektory stałych (wzmocnień sterownika),  $\mathbf{e}(t) = \boldsymbol{\theta}_d(t) - \boldsymbol{\theta}(t)$  - wektor błędów pozycji kątowej,  $\mathbf{e}_V(t) = \boldsymbol{\omega}_d(t) - \boldsymbol{\omega}(t)$  - wektor błędów prędkości kątowej,  $\boldsymbol{\omega}(t) = [\boldsymbol{\theta}(t) - \boldsymbol{\theta}(t-1)] \frac{1}{T_s}$  - wektor prędkości kątowej,  $\mathbf{s}_c(t)$  - wektor błędów pochodzący od nieliniowego integratora,  $T_s$  - okres próbkowania sterownika.

Nieliniowy integrator  $\mathbf{s}_c$ , z takimi samymi modyfikacjami, jakie zastosowano w równoległym sterowniku manipulatora. Są one opisane równaniami (30)-(35), a kierunek ruchu jest wykrywany według zależności: (37) gdzie  $\omega_{d,i}$  jest pożądaną prędkością kątową silnika we współrzędnych złączowych.

$$(37) \quad s_{g,i}(t) = \begin{cases} -1; & \omega_{d,i}(t) < 0 \\ 0; & \omega_{d,i}(t) = 0 \\ 1; & \omega_{d,i}(t) > 0 \end{cases}$$

Strukturę zmodyfikowanego sterownika śledzenia trajektorii przedstawiono na rys. 15.



Rys.7. Schemat blokowy sterowania pozycyjnego manipulatora szeregowego

Śledzenie trajektorii odbywa się we współrzędnych złączowych, ale sterownik jest przystosowany do pracy z trajektoriami generowanymi w globalnej przestrzeni kartezjańskiej, które są przekształcane w przestrzeń lokalną (złączową) przez rozwiązanie zagadnienia kinematyki odwrotnej (18) i jej pochodnych (19)-(20)

Wyniki przeprowadzonych testów dotyczących dokładności odwzorowania trajektorii zadanej dla robota zostały przedstawione i omówione w publikacji [6].

### Platformy sprzętowe

Wykorzystanie metody wyliczanych momentów w układzie śledzenia trajektorii sterownika wiąże się z wymogiem obliczenia momentów i sił, na podstawie przyjętego modelu dynamiki, z założoną wysoką częstotliwością próbkowania. Dlatego warunkiem zastosowania platformy sprzętowej do implementacji algorytmów sterowania jest posiadanie odpowiedniej mocy obliczeniowej, zapewniającej stabilną pracę sterownika. Dla wysokiej dokładności odwzorowania zadanej trajektorii, równie istotne jest określenie maksymalnego błędu numerycznego. Dlatego, zdecydowano o zapisie funkcjonalnym (programowym) postaci analitycznych równań matematycznych opisujących modele dynamiki i kinematykę robota, z zastosowaniem kodowania liczb

zmiennoprzecinkowych pojedynczej precyzji (standard IEE-754 ang. float). W tab. 1 przedstawiono typ i ilość operacji algorytmu śledzenia trajektorii sterownika robota.

Tabela 1. Rodzaj i liczba operacji algorytmu śledzenia trajektorii

Opis	Rodzaj operacji	Liczba operacji
dodawanie i odejmowanie	+/-	186
mnożenie	x	216
dzielnie	/	31
pierwiastek kwadraty	sqrt(x)	13
sinus	sin(x)	1
rzutowanie	float(int)	5
saturacja	sat(x,y)	5

Do implementacji algorytmu śledzenia trajektorii dla robota hybrydowego (obu manipulatorów) wraz z równaniami kinematyki odwrotnej (7)-(11) zostały wybrane dwie platformy sprzętowe, wyposażone w układy cyfrowe różnego typu.

Pierwszą stanowił układ reprogramowalny typu FPGA z serii Stratix III [7] (3SL150), firmy Intel-Altera. Drugą platformą użytą do testów był mikrokontroler TMS320F28069 [8] klasy DSP z serii C2000, firmy Texas Instruments.

W rezultacie prowadzonych badań nad architekturą sprzętowo-programową systemu typu SoPC (ang. System on Programmable Chip), dla pierwszej platformy sprzętowej wyposażonej w układ FPGA, uzyskano trzy wersje architektury.

Pierwsza wersja, była oparta na procesorze typu soft-core, Nios II (RISC, 32-bit) [9] firmy Intel-Altera, którego lista instrukcji koprocessorowych została rozszerzona o układ FPU dostarczany przez producenta układów w postaci bibliotek IP-core [10]. W testach tę architekturę oznaczono jako FPGA1.

Kolejną wersją architektury z wykorzystaniem układów reprogramowalnych stanowił system wyposażony w procesor Nios II ze zbiorem autorskich, sprzętowych instrukcji koprocessorowych opisanych w publikacjach [11], przeznaczonych do akceleracji obliczeń zmiennoprzecinkowych. Do zbioru tych instrukcji można zaliczyć: FPU (dodawanie, odejmowanie, mnożenie, dzielenie), funkcje trygonometryczne (sinus, kosinus) oraz dodatkowe funkcje (saturacja, wykrywanie znaku – signum, rzutowania typów – float(int)). W testach tę architekturę oznaczono jako FPGA2.

Trzecią wersję architektury sprzętowo-programowej stanowiła autorska zmiennoprzecinkowa jednostka cyfrowa typu ASIP (ang. Application specific instruction set processor) [12]. Do przeprowadzonych prac dotyczących opracowania dedykowanej jednostki można zaliczyć: opisanie architektury sprzętowej jednostki z wykorzystaniem, wcześniej wykonanych, instrukcji koprocessorowych oraz opracowanie narzędzi programistycznych służących do kompilacji, implementacji i debugowania, autorskiego kodu assembler. Architekturę tę oznaczono w testach jako FPGA3.

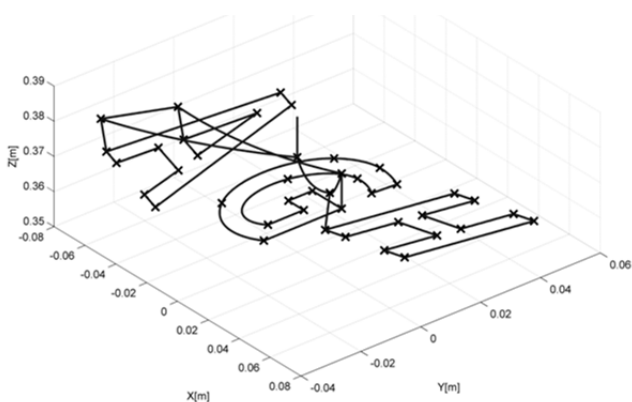
Podstawowym układem drugiej platformy sprzętowej jest mikrokontroler TMS320F28069 klasy DSP, przeznaczony do implementacji złożonych algorytmów sterowania, wyposażony w 32-bitowy procesor C2806x, ze zmodyfikowaną architekturą harwardzką wspomagany przez FPU i CLA (ang. Control law accelerator). Algorytm śledzenia trajektorii został zapisany w języku C, przy użyciu narzędzi programistycznych producenta: Code Composer Studio (CCS), F280xx C/C++ kompilator. Aby uzyskać maksymalną wydajność obliczeniową zastosowano:

- bibliotekę producenta zapewniającą wsparcie dla działań matematycznych z wykorzystaniem FPU - rts2800\_fpu32.lib [13]
- bibliotekę producenta: rts2800\_fpu32\_fast\_supplement.lib [13], będącą rozszerzeniem powyższej biblioteki zapewniające zoptymalizowanie kodu pod kątem wydajności wykonania funkcji: atan, atan2, kosinus, sinus, dzielenie, pierwiastek kwadratowy,
- skonfigurowano kompilator na najwyższym poziomie optymalizacji kodu,
- do implementacji kodu wykorzystano pamięć typu RAM, skonfigurowano pamięć w jeden blok RAM\_EBSS, poprzez połączenie bloków RAML3 o rozmiarze 512B i RAML4 o rozmiarze 1KB.

Architekturę tę oznaczono w testach jako DSP-C28x.

### Wyniki testów

Na rys. 8 zaprezentowano tor narzędzia wykorzystany do testów, składający się z liter "AGH", które zostały rozmieszczone w różnych płaszczyznach. Do programowego zapisu toru, wykorzystano G-kod.



Rys.8. Tor narzędzia użyty w testach

W testach wykorzystano płytę rozwojową Terasic DE3 z układem FPGA Stratix III, oraz płytę Piccollo firmy Texas Instruments z mikrokontrolerem TMS320F28069 (DSP). W powyższych platformach zaimplementowano algorytm śledzenia trajektorii robota (manipulatora równoległego i szeregowego) oraz kinematykę prostą i odwrotną robota wraz z pochodnymi I i II rzędu. Pomiar czasu we wszystkich testach dokonano niezależnym zewnętrznym układem dla każdej iteracji programu. W tab. 2 zaprezentowano wyniki najdłuższych czasów uzyskanych dla każdej z platform.

Tabela 2. Czasy realizacji algorytmu dla testowanych platform sprzętowych

Typ platformy/układu	Częstotliwość taktowania zegara fclk [MHz]	Całkowity maksymalny czas obliczeń [μs]
FPGA1	100	218
FPGA2	100	82
FPGA3	125	21
DSP-C28x	80	43

Kolejne przeprowadzone testy dotyczyły wyznaczenia maksymalnego błędu numerycznego dla poszczególnych platform. Błędy obliczono dla każdej iteracji programu, na podstawie różnicy zbioru wartości wzorcowych, zapisanych w formacie liczb zmiennoprzecinkowych podwójnej precyzji (64-bit, typ double), ze zbiorem otrzymanych wyników pomiarowych, liczb zmiennoprzecinkowych pojedynczej precyzji (32-bit, typ float). Część uzyskanych wyników, w

postaci maksymalnych błędów dla rozwiązania zadania kinematyki odwrotnej robota, wraz z pierwszą i drugą pochodną, przedstawiono w tab. 3. Dla porównania, w kolumnie drugiej tab. 3, umieszczono wyniki błędów maksymalnych, dla tych samych obliczeń, uzyskanych na komputerze klasy PC, pomiędzy wartościami wzorcowymi (podwójnej precyzji, 64-bit, double) i wartościami pojedynczej precyzji (32-bit, float). Jednym z ograniczeń układów FPGA, jest liczba elementów programowalnych zwanych zasobami sprzętowymi układu. W tab.3 zostały zestawione ilości wykorzystanych poszczególnych zasobów sprzętowych układu Stratix III (EP3SL150 F1152), dla wersji architektury wykorzystującej procesor NiosII (tab. 3, FPGA2), z wersją architektury, w której wykorzystano autorską jednostkę zmiennoprzecinkową ASIP (tab. 3, FPGA3).

Tabela 3. Maksymalny błąd numeryczny dla testowanych platform sprzętowych

Wielkość	PC	DSP-C28x	FPGA 1, 2, 3
$\Delta l_1$ [m]	1.8651e-06	1.8651e-06	1.8651e-06
$\Delta l_2$ [m]	1.1658e-06	2.2893e-08	2.2165e-08
$\Delta l_3$ [m]	1.0510e-06	1.0512e-07	1.0510e-06
$\Delta \theta_4$ [rad]		4.4702e-08	
$\Delta \theta_5$ [rad]		4.4702e-08	
$\Delta v_1$ [m/s]	5.9368e-07	5.9377e-07	5.9368e-07
$\Delta v_2$ [m/s]	8.2634e-07	8.2634e-07	8.2634e-07
$\Delta v_3$ [m/s]	7.5938e-07	7.6683e-07	7.6003e-07
$\Delta \omega_4$ [rad/s]		1.0000e-07	
$\Delta \omega_5$ [rad/s]		1.0000e-07	
$\Delta a_1$ [m/s <sup>2</sup> ]	1.8899e-06	2.1942e-07	2.2496e-07
$\Delta a_2$ [m/s <sup>2</sup> ]	2.7877e-06	6.0238e-07	5.8701e-07
$\Delta a_3$ [m/s <sup>2</sup> ]	2.6891e-06	6.0238e-06	2.9990e-06
$\Delta \varepsilon_4$ [rad/s <sup>2</sup> ]		3.2011e-07	
$\Delta \varepsilon_5$ [rad/s <sup>2</sup> ]		3.2011e-07	

Tabela 4. Liczba wykorzystanych poszczególnych zasobów układu FPGA

Wersja systemu	Typ elementu w strukturze FPGA Stratix III 3SL150			
	LUTs	Dedykowane rej. log.	Pamięć wew. [kB]	Bloki DSP
FPGA 2	25 307 (*22%)	8 703 (*8%)	159 (*41%)	28 (*7%)
FPGA 3	12 010 (*11%)	7 489 (*7%)	120 (*21%)	8 (*2%)
Redukcja	13 297	1 217	39	20

\* część zasobów układu [%] wykorzystana do implementacji, w stosunku do całkowitej liczby poszczególnych elementów (zasobów) układu

### Wnioski

Architektury sprzętowe oznaczone w testach jako: FPGA2 i FPGA3 oraz platforma sprzętowa wyposażona w procesor DSP-C28x spełniły wymaganie dotyczące założonego maksymalnego czasu obliczeń koniecznego do sterowania robotem, który wynosił 100 μs. W przypadku procesora klasy DSP wyposażonego w procesor C-28x, wspomaganego przez jednostki FPU i CLA, całkowity czas wykonania algorytmu śledzenia trajektorii, wyniósł 43 μs. W przypadku zastosowania tego mikrokontrolera, do pełnej implementacji algorytmów sterownika, należałoby uwzględnić zapas czasu 57 μs, względem wymaganego, na realizację pozostałych części algorytmu sterownika, w tym algorytm generowania trajektorii. W przypadku architektury FPGA, zastosowanie architektury dwuprocesorowej

pozwała na zrównoleglenie wykonywania algorytmów sterownika (równoległe wykonywanie algorytmu generowania trajektorii i śledzenia trajektorii). Dlatego, uzyskany zapas czasu, wynoszący 79  $\mu$ s, dla wersji FPGA3, może być wykorzystany do rozwinięcia algorytmu sterownika np. o uwzględnienie w modelu dynamiki sił tarcia [14]. Zastosowanie autorskiej jednostki zmiennoprzecinkowej (FPGA3) w porównaniu do architektury z wykorzystaniem soft-procesora NiosII z FPU (FPGA2), zredukowało ilość wykorzystanych zasobów układu FPGA (Stratix III 3SL150) o: 50% elementów LUT, 14% dedykowanych rejestrów logicznych, 24% wbudowanej pamięci oraz o 71% zasobów bloków DSP.

Maksymalne błędy numeryczne pozycji, orientacji, prędkości i przyspieszeń dla wersji platformy FPGA1, FPGA2 i FPGA3 były takie same. Maksymalny błąd numeryczny pozycji (dla wszystkich testowanych platform) nie przekroczył 2  $\mu$ m, a dla orientacji wyniósł 10  $\mu$ rad, co jest dopuszczalnym wynikiem dla przyjętych wymagań.

Przeprowadzone testy pozwoliły na porównanie platform sprzętowych wyposażonych w układy cyfrowe różnego typu. Uzyskane wyniki, dostarczyły informacji dotyczących spełnienia stawianych wymagań (częstotliwości próbkowania 10kHz i dokładności numerycznej) a tym samym, możliwości zastosowania testowanych układów (DSP-C28x, FPGA2, FPGA3) jako platformy docelowej dla układu sterowania robotem.

**Autorzy:** dr inż. Konrad Gac, AGH Akademia Górniczo-Hutnicza, Katedra Robotyki i Mechatroniki, al. Mickiewicza 30, 30-059 Kraków, E-mail: [kgac@agh.edu.pl](mailto:kgac@agh.edu.pl), dr hab. inż. Maciej Petko, AGH Akademia Górniczo-Hutnicza, Katedra Robotyki i Mechatroniki, al. Mickiewicza 30, 30-059 Kraków, E-mail: [petko@agh.edu.pl](mailto:petko@agh.edu.pl).

## LITERATURA

- [1] Suh S.H., Kang S.K., Chung D.H. and Stroud I., "Theory and Design of CNC Systems", Springer Series in Advanced Manufacturing, London: Springer, (2008)
- [2] L. Sciacivco, B. Siciliano, Modelling and Control of Robot Manipulators, Springer Verlag, (2000)
- [3] Schilling R.J., Fundamentals of Robotic, Analysis and Control, Prentice Hall, (1990)
- [4] Kozłowski K., Dutkiewicz P., Wróblewski W., Modelowanie i Sterowanie Robotów, PWN, (2012)
- [5] Karpel G., Petko M., "Triple articulated joint," PL Patent PL 207 396 B1, Dec. 31, (2010)
- [6] Petko M., Karpel G., Gac K., Góra G., Kobus K., Ochoński J Trajectory tracking controller of the hybrid robot for milling Mechatronics, Theoretical and applied aspects of modern mechatronics, (2016), No. 37, s. 100–111
- [7] Intel-Altera, karta katalogowa Stratix III 3SL150, online <http://www.datasheetarchive.com/dlmain/2b564f92343d43dc6eab7400dae06f345e3d0e/M/EP3SL150F1152C2>, 2021.
- [8] Texas-Instruments, datasheet TMS-320-F28069, online, <http://www.ti.com/general/docs/lit/getliterature.tsp?baseLiteratureNumber=sprs698&fileType=pdf>, (2020)
- [9] Intel-Altera, Nios II Software Developer Handbook, online [https://www.intel.com/content/dam/www/programmable/us/en/pdfs/literature/hb/nios2/n2sw\\_nii5v2gen2.pdf](https://www.intel.com/content/dam/www/programmable/us/en/pdfs/literature/hb/nios2/n2sw_nii5v2gen2.pdf), (2021)
- [10] Intel-Altera, Nios II Custom Instruction User Guide, online [https://www.intel.com/content/dam/www/programmable/us/en/pdfs/literature/ug/ug\\_nios2\\_custom\\_instruction.pdf](https://www.intel.com/content/dam/www/programmable/us/en/pdfs/literature/ug/ug_nios2_custom_instruction.pdf), (2021)
- [11] Petko M., Karpel G., Gac K., Góra G., Kobus K., Ochoński J. CNC system of the 5-axis hybrid robot for milling, Mechatronics Theoretical and applied aspects of modern mechatronics, (2016), No. 37, s. 89–99
- [12] Gac K., Petko M., Góra G., Zmiennoprzecinkowa jednostka numeryczna implementowana w układach reprogramowalnych Projektowanie systemów mechatronicznych, zagadnienia wybrane, Akademia Górniczo-Hutnicza w Krakowie, (2020), s. 137–144
- [13] Texas-Instruments, CCS lib C2800, online, [http://www.deyisupport.com/cfsfile.ashx/\\_\\_\\_key/communityserverdiscussionscomponentsfiles/56/4075.C28x\\_2D00\\_FPU\\_2D00\\_LIB\\_2D00\\_UG.pdf](http://www.deyisupport.com/cfsfile.ashx/___key/communityserverdiscussionscomponentsfiles/56/4075.C28x_2D00_FPU_2D00_LIB_2D00_UG.pdf), (2020)
- [14] Kabziński J., Jastrzębski M., Practical implementation of adaptive friction compensation based on partially identified LuGre model, 19th International Conference on Methods and Models in Automation and Robotics (MMAR), IEEE, (2014)