

Sterowanie kobotem za pomocą wirtualnej rzeczywistości dla potrzeb Przemysłu 4.0

Mateusz Salach, Arkadiusz Stęchły, Andrzej Paszkiewicz, Patryk Organiściak, Grzegorz Budzik
Politechniki Rzeszowska, Wydział Elektrotechniki i Informatyki, Katedra Informatyki i Automatyki, ul. Wincentego Pola 2, 35-021 Rzeszów

Streszczenie: Przemysł 4.0 przechodzi ciągłą ewolucję. Systemy zarządzania i monitoringu zaimplementowane w infrastrukturze przemysłowej gromadzą i przetwarzają coraz więcej danych, następnie bazując na dedykowanych algorytmach podejmują decyzje, a w konsekwencji wysyłają komendy sterujące do urządzeń i procesów technologicznych. Równocześnie wzrasta zainteresowanie łączeniem urządzeń z innymi na pozór odmiennymi technologiami. Przykładem tworzenia takiej koegzystencji jest integracja rozwiązań przemysłowych z wirtualną rzeczywistością. Działania takie przyczyniają się do tworzenia cyfrowych bliźniaków urządzeń i procesów, co z kolei umożliwia zdalne, a w wielu przypadkach również automatyczne sterowanie fizycznymi obiektami oraz przebiegiem rzeczywistych procesów. W trakcie prowadzonych badań wykorzystano takie podejście do opracowania rozwiązania umożliwiającego sterowanie robotem z dowolnego miejsca i w dowolnym czasie. Niniejsza praca prezentuje prototyp rozwiązania zapewniającego zdalne sterowanie robotem współpracującym typu kobot.

Słowa kluczowe: wirtualna rzeczywistość, robotyka, przemysł 4.0, cyfrowy bliźniak

1. Wprowadzenie

Rozwiązania oparte na wirtualnej rzeczywistości (VR) stają się coraz bardziej popularne, szczególnie w obszarze szkoleń dedykowanych różnym dziedzinom i zastosowaniom. Takie systemy VR są już implementowane na potrzeby medycyny [1-3], edukacji [4-6], przemysłu [7, 8], lotnictwa [9-11], branży automotive [12, 13], a nawet w zastosowaniach militarnych [14, 15, 20, 21]. Wykorzystanie wirtualnej rzeczywistości doskonale wpisuje się w koncepcję Przemysłu 4.0 w ujęciu inteligentnego sterowania wraz z obsługą informacji zwrotnej w czasie rzeczywistym. Równocześnie rozwój technologii wpływa na zakres i potencjał praktycznego wykorzystania nowych rozwiązań. Dzięki wydajniejszym, bardziej zaawansowanym i zoptymalizowanym urządzeniom w połączeniu z nowoczesną infrastrukturą komunikacyjną możliwe staje się wykorzystanie rozwiązań wirtualnej/rozszerzonej (VR/AR), jak również mieszanej rzeczywistości (MR) do kontroli i sterowania różnych urządzeń w czasie zbliżonym do rzeczywistego. Przykładem takiego rozwiązania może być zdalne sterowanie robotem współpracującym przez operatora znajdującego się w zupeł-

nie innym miejscu, często odległym wiele kilometrów, a nawet tysiące kilometrów od miejsca lokalizacji fizycznego urządzenia.

W wielu przypadkach brak bezpośredniego dostępu do fabryki/linii produkcyjnej zwiększa bezpieczeństwo zarówno operatorów jak i samych urządzeń. Obecność innych osób w zasięgu pracy takiego robota jest zabroniona. Użytkownik sterujący danym urządzeniem ma możliwość podglądu sytuacji z wielu kamer rozmieszczonych przy stanowisku. Zakłada się fizyczne odizolowanie strefy pracy robota. Istnieje również możliwość implementacji systemu lokalizacji personelu, który połączony byłby z systemem sterowania robotem. W momencie pojawienia się jakiegokolwiek osoby w strefie pracy robota byłby on automatycznie wyłączony. W przypadku zapewnienia bezpiecznego połączenia, charakteryzującego się jednocześnie dobrymi parametrami transmisyjnymi (np. małymi opóźnieniami) w odniesieniu do przesyłanego obrazu oraz pakietów sterujących robotem możliwe jest efektywne sterowanie urządzeniem praktycznie z każdego miejsca na ziemi. Automatyzacja oraz wirtualizacja takich procesów pozwoli nie tylko swobodnie przełączać się między różnymi urządzeniami, ale także przyczyni się do ograniczenia kosztów. Istotnym aspektem tworzenia tego typu rozwiązań jest również bezpieczeństwo. Taki system może znaleźć zastosowanie nie tylko w przemyśle, ale także podczas klęsk żywiołowych, pandemii, ograniczonej dostępności pracowników itp.

Mimo licznych zalet, sterowanie za pomocą technologii VR odrębnym, zewnętrznym urządzeniem nie jest jeszcze powszechne. Można stwierdzić, że obszar ten czeka na odkrycie i uznanie w wielu branżach. Sytuacja taka wynika z pewnością z wielu różnych przesłanek, niemniej jednak należy pamiętać, że wykorzystanie jakiegokolwiek nowej technologii w procesach edukacyjnych jest znacznie prostsze od zastosowania ich w rzeczywistych procesach produkcyjnych, gdzie tolerancja i poziom

Autor korespondujący:

Mateusz Salach, m.salach@prz.edu.pl

Artykuł recenzowany

nadesłany 08.12.2023 r., przyjęty do druku 19.08.2024 r.



Zezwala się na korzystanie z artykułu na warunkach licencji Creative Commons Uznanie autorstwa 3.0

akceptacji ewentualnych błędów i niedokładności jest znacznie niższy niż w przypadku szkoleń, czy też kursów instruktażowych. Symulacje szkoleń pozwalają na zapoznanie się z wieloma aspektami i użytkowaniem np. różnego rodzaju ciężkich maszyn [15].

Wychodząc naprzeciw potrzebom przemysłu oraz biorąc pod uwagę olbrzymi potencjał wdrożeniowy autorzy opracowali system zdalnego sterowania robotem współpracującym typu kobot za pomocą kontrolerów dedykowanych wirtualnej rzeczywistości. System integruje wirtualne środowisko z cyfrowym bliźniakiem robota współpracującego UR5e, którego rzeczywisty odpowiednik zlokalizowany został w innej lokalizacji. Dzięki temu użytkownik sterując kontrolerem VR może zdalnie obsługiwać robota współpracującego, który wykonuje komendy przetworzone przez swojego cyfrowego bliźniaka w środowisku VR. Najczęstszym rozwiązaniem stosowanym w robotyce jest sterowanie za pomocą dedykowanych kontrolerów (np. joystick). Zaproponowane rozwiązanie bazuje na kontrolerach VR, implementujące zdalne sterowania, stanowi nowe podejście do zagadnienia kontroli i sterowania urządzeniami przemysłowymi. Zastosowanie kontrolerów VR pozwala w pełni zintegrować i odwzorować ruch, umożliwia użytkownikowi zdalny podgląd otoczenia pracy robota. W drugiej części artykułu przedstawione zostaną rozwiązania oparte o wykorzystanie wirtualnej rzeczywistości zarówno w szkoleniach jak i zdalnym sterowaniu urządzeniami zewnętrznymi. Trzecia część artykułu poświęcona jest koncepcji zdalnego sterowania, zaś czwarta prezentuje wykonanie aplikacji. W ostatniej części prezentowane będą wyniki pomiarowe zaś w sekcji szóstej przedstawione zostanie podsumowanie.

2. Analiza rozwiązań

W ostatnich latach pojawiło się wiele rozwiązań z obszaru VR przeznaczonych dla branży przemysłowej, medycznej, lotniczej oraz automotive. Popularność, coraz szersza dostępność i innowacyjność gogli VR pozwala na przygotowanie atrakcyjnych i realistycznych cyfrowych odpowiedników rzeczywistych obiektów dedykowanych środowisku pracy, jak również znajdujących zastosowanie w szkoleniach. Tworzenie świata wirtualnego wymaga opracowania odpowiednich scenariuszy szkoleń i pracy. W celu uzyskania wyższego poziomu immersji w ramach zdalnego sterowania można zastosować zewnętrzne czujniki wraz z mikrokontrolerami do pozyskiwania dodatkowych informacji np. położenia ciała, bądź ręki. Obecnie przy wykorzystaniu standardowego systemu VR taka integracja jest niedostępna.

Jednakże, środowisko badawczo-rozwojowe umożliwia opracowywanie i wstępne testowanie przyszłych rozwiązań. Przykładem takim jest rozwiązanie, gdzie wykorzystano system VR wraz z mikrokomputerem Raspberry Pi i podłączoną do niej kamerą oraz mikrokontrolerem Arduino wraz z czujnikami [16]. Za pomocą interfejsu Bluetooth dane z czujników urządzeń wearables zamocowanych na ręce były pozyskiwane, a następnie analizowane przez program, który przetwarzał je na odpowiednie położenie człowieka/roboty. Stosując dwa czujniki IMU, dwa czujniki Flex oraz dwa czujniki FSR podłączone do Arduino UNO oraz kamerę podpiętą do RPi, można odczytać położenie całej ręki i przesłać te informacje bezpośrednio do programu, który odpowiadał za sterowanie robotem humanoidalnym. Rozwiązanie to bazowało jedynie na widoku z kamery bez wykorzystania środowiska wirtualnej rzeczywistości.

Kolejne interesujące rozwiązanie polega na wykorzystaniu robota współpracującego KUKA LBR iiwa do szkolenia z zakresu wymiany biodra [17]. W pracy skupiono się głównie na podejściu haptycznym tzn. pozyskaniu impulsów zewnętrznych do określenia możliwości i zasięgu pracy urządzenia. VR wykorzystano do prezentacji szkolenia w wirtualnym

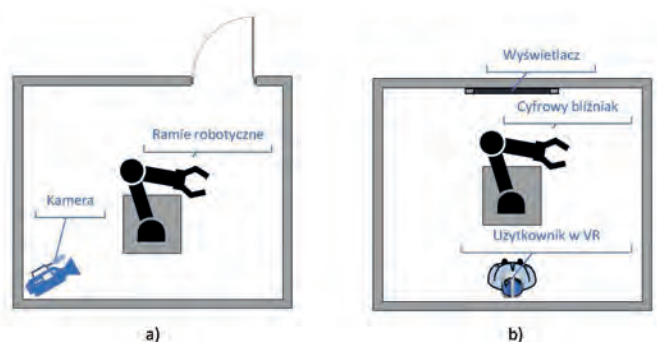
środowisku. Ruchy robota zostały pozyskane i wprowadzone do oprogramowania co pozwalało uczestnikowi odczuć większą immersję.

Inny przykład omawia sterowanie robotem za pomocą gogli Oculus Rift w połączeniu z systemem ROS [20]. System ten pozwala na sterowanie rzeczywistym urządzeniem/robotem mobilnym podczas zawodów. Jednym z celów projektu było sprawdzenie scenariusza obejmującego zamianę ról pomiędzy użytkownikiem a robotem. W tym przypadku to robot dyktował komendy użytkownikowi, który miał założone gogle VR. Konkretna osoba otrzymywała komunikaty wskazujące określone zadania do wykonania, np. przeniesienie obiektu z miejsca A do miejsca B. Wirtualna rzeczywistość może zostać wykorzystana także do współpracy z robotami o innej strukturze i przeznaczeniu aniżeli pojazdy bądź roboty współpracujące. W publikacji [19] autorzy prezentują rozwiązanie, w którym udało im się połączyć urządzenia latające UAV (ang. *Unmanned Aerial Vehicle*) ze środowiskiem VR. Za pomocą kontrolerów mogli zdalnie sterować zarówno dronami w przygotowanym przez nich programie w środowisku Unity, jak i fizycznymi urządzeniami w świecie rzeczywistym. Autorzy opracowali rozwiązanie, w którym połączyli drony z siecią telekomunikacyjną wykorzystując mikrokomputer Raspberry Pi, moduły 5G oraz platformę ROS (ang. *Robotic Operating System*). Wszystkie zaprezentowane rozwiązania skupiają się na robotach dedykowanych, budowanych przez zespoły pod konkretne zastosowanie, zaś w mniejszym bądź żadnym stopniu na urządzeniach przemysłowych. Rozwiązanie zaprezentowane w artykule wykorzystuje implementację środowiska wraz z robotem produkcyjnym.

3. Środowisko testowe

Większość interfejsów umożliwiających zdalne sterowanie robotem współpracującym jest niewygodna w użyciu. Najczęściej jest to nieporęczny kontroler lub aplikacja komputerowa, która pozwala sterować tylko jednym przegubem w danym momencie lub punktem końcowym w mało intuicyjny sposób. Sterowanie za pomocą kontrolera odbywa się za pomocą dżwaka lub przycisków na dotykowym ekranie, a w przypadku aplikacji komputerowej przyciskami lub wpisywaniem wartości kątów obrotu. W przypadku kontrolera ABB możliwe jest poruszanie się jednocześnie w trzech osiach (obrót dżwaka odpowiada za wysokość efektora), lecz jest to trudne i nieprecyzyjne, natomiast w przypadku kontrolera Universal Robots jest ono całkowicie niemożliwe.

Operator poruszający się w aplikacji VR może obserwować fizycznego robota tylko jako obraz z kamery wyświetlony na ekranie. Z uwagi na fakt, że kamery przesyłają obraz dwuwymiarowy, powstaje problem z percepcją głębi. Może to spowodować błędną ocenę pozycji ramienia przez operatora. Aby temu zapobiec w opracowanej aplikacji umieszczony został model 3D robota tj. jego cyfrowy bliźniak (*Digital twin*), którego położenie



Rys. 1. Schemat pomieszczeń, a) rzeczywiste stanowisko, b) aplikacja VR
Fig. 1. Room schematics, a) laboratory schematic, b) VR application

nie jest stale aktualizowane, tak aby było zgodne z położeniem fizycznego robota (rys. 1).

Aplikacja VR stworzona została w środowisku Unity w wersji 2020.3.26f1. Dodatkowo, wykorzystana została biblioteka open-source OpenXR, która umożliwia unifikację aplikacji, dzięki czemu jest ona dostępna na różne platformy VR.

4. Algorytm zdalnego sterowania robotem

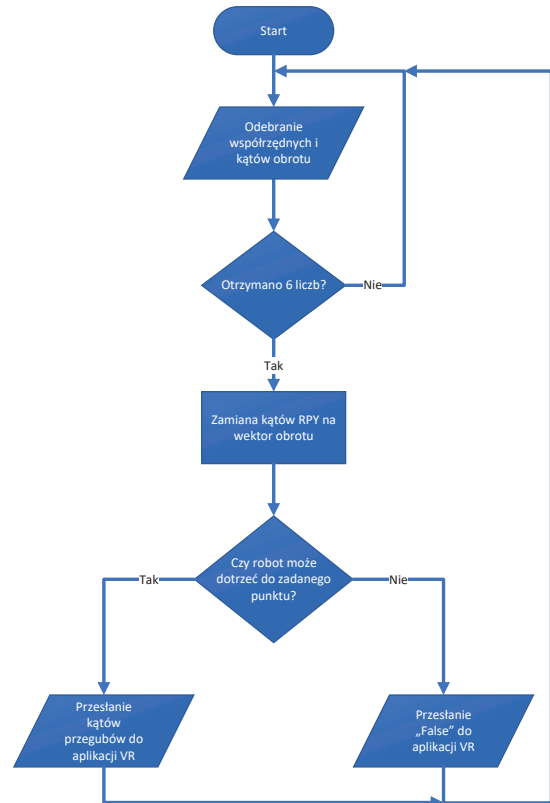
Program napisany w języku URScript podzielony jest na dwa segmenty, przy czym segment „BeforeStart” wykonywany jest jako pierwszy i tylko raz, natomiast na drugi segment składają się cztery wątki, które działają równolegle w pętłach nieskończonych (rys. 2). Komunikacja między aplikacją VR a programem w języku URScript odbywa się za pomocą protokołu TCP/IP. Sterowanie robotem jest realizowane w czterech wątkach wykonujących się w pętłach nieskończonych i jednej instrukcji wykonywanej jednokrotnie. Instrukcja ta otwiera gniazda TCP/IP, pierwszy wątek odbiera dane wysłane przez aplikację Unity i wysyła wizualizację docelowej pozycji ramienia, drugi wątek również odbiera dane przysyłane przez Unity i oblicza kinematykę odwrotną, trzeci wątek odpowiedzialny jest za ruch ramieniem, a ostatni za przesyłanie rzeczywistych kątów obrotu poszczególnych przegubów ramienia do aplikacji Unity.

Z pierwszego gniazda TCP/IP odczytana zostaje wiadomość wysłana przez aplikację VR, wykorzystane zostało do tego celu polecenie `socket_read_ascii_float()`, które jako parametry przyjmują liczbę oczekiwanych wartości parametrów współrzędnych, oraz nazwę gniazda, które odpowiada za tę komunikację (rys. 3). Jak sama nazwa wskazuje polecenie to przyjmuje wiadomość z liczbami zmiennoprzecinkowymi, poszczególne liczby oddzielane są przecinkami, a cała wiadomość znajduje się w nawiasach. Otrzymana wiadomość zostaje zapisana w tablicy, której pierwsza wartość oznacza, ile liczb się w niej znajduje, jeśli ta



Rys. 2. Algorytm komunikacji między aplikacją VR a programem w URScript

Fig. 2. The algorithm of communication between the VR application and URScript program



Rys. 3. Algorytm przetwarzania kątów ruchu dla robota – I wątek

Fig. 3. The algorithm for processing angles of motion for a robot – I thread

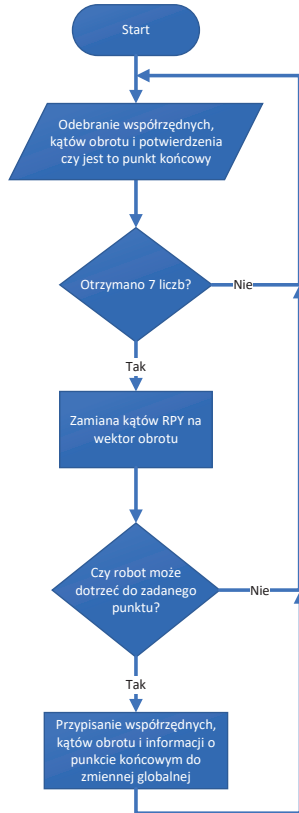
liczba jest inna od oczekiwanej (w tym przypadku „6”), następuje ponowna próba odczytania wartości. Ponieważ silnik Unity korzysta z kątów RPY, czyli kątów Przechylenia (ang. *Roll*), Pochylenia (ang. *Pitch*) i Odchylenia (ang. *Yaw*), a polecenie ruchu robota wymaga wektora obrotu, wykorzystane zostało ogólnodostępne polecenie języka URScript `ropy2rotvec()`, które wykonuje potrzebne przekształcenia. Za pomocą polecenia `get_inverse_kin_has_solution()`, sprawdzone zostaje czy robot może dojechać do wskazanego punktu, jeśli tak za pomocą polecenia, `get_inverse_kin()` obliczona zostaje kinematyka odwrotna robota, a wynikowe kąty obrotu przegubów zostają przesłane do aplikacji VR, w przeciwnym wypadku przesłany zostaje ciąg znaków „False”. Ponieważ cały wątek wykonywany jest w pętli nieskończonej program wraca do odczytywania wiadomości.

Drugi wątek działa analogicznie do pierwszego. Z drugiego gniazda TCP/IP odczytane są współrzędne i kąt obrotu docelowego punktu, oraz czy punkt ten jest punktem końcowym czy pośrednim w trajektorii robota. Ostatnia informacja przedstawiona jest za pomocą cyfry 1 lub 0, gdzie 1 oznacza punkt końcowy. W tym wątku również następuje zamiana kątów RPY na wektor obrotu oraz sprawdzenie czy robot może dotrzeć do zadanego punktu. W przypadku możliwości poprawnego dotarcia do zadanego punktu, do zmiennych globalnych przypisane zostają współrzędne, wektor obrotu oraz informacja o punkcie końcowym, a następnie program wraca do początku. W przeciwnym razie wartości zostają odrzucone i odczytana zostaje kolejna wiadomość (rys. 4).

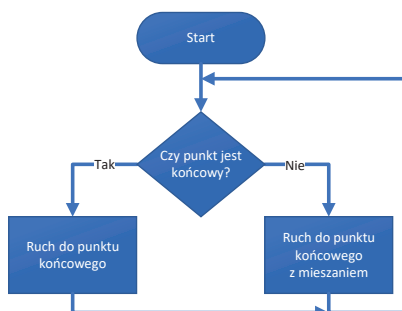
Trzeci wątek (rys. 5) w pierwszej kolejności sprawdza czy punkt jest punktem końcowym ruchu. Jeśli tak, wykonywany jest ruch do zadanego punktu bez blendowania (polecenie `MoveJ`), w przeciwnym wypadku ruch wykonany jest z mieszaniem. Ruch z blendowaniem (ang. *blending*) umożliwia robotowi płynne przejście z jednego ruchu do następnego co przedstawia przykład zaprezentowany na rys. 6. Głównymi cechami ruchu `MoveJ` jest jego szybkość oraz to, że wszystkie przeguby kończą ruch w tym samym momencie. Negatywną cechą tego działania jest zakrzywiony tor ruchu, który w niektórych przypadkach

jest trudny do przewidzenia. Dodatkowymi parametrami, jakie ma polecenie MoveJ jest maksymalne przyspieszenie i prędkość kątowa. Ruch ten może się wykonać tylko wtedy, jeśli robot się nie porusza lub wykonuje poprzedni ruch z blendowaniem (mieszaniem).

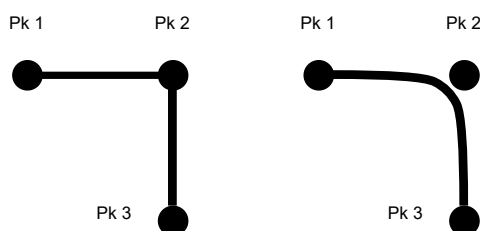
Na rys. 6 przedstawiono przykładowy ruch bez blendowania i z blendowaniem. W przypadku ruchu pokazanego po lewej stronie, robot zaczynając w punkcie „Pk 1” wykonuje ruch do punktu „Pk 2” i się zatrzymuje, następnie rozpoczyna ruch do punktu „Pk 3”, gdzie kończy ruch. Ruch pokazany po prawej



Rys. 4. Algorytm przetwarzania kątów ruchu dla robota – II wątek
Fig. 4. The algorithm for processing angles of motion for a robot – II thread

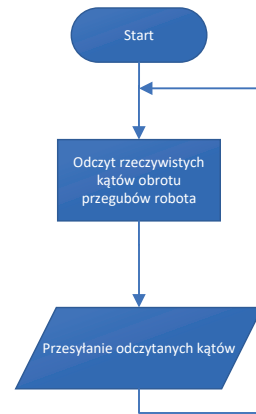


Rys. 5. Algorytm przetwarzania kątów ruchu dla robota – III wątek
Fig. 5. The algorithm for processing angles of motion for a robot – III thread



Rys. 6. Porównanie ruchu bez oraz z mieszaniem
Fig. 6. Comparison of movements without and with blending

stronie jest realizowany z włączoną opcją blendowania, robot nie osiąga punktu „Pk 2”, lecz płynnie przechodzi do punktu „Pk 3”, gdzie się zatrzymuje.



Rys. 7. Algorytm przetwarzania kątów ruchu dla robota – IV wątek
Fig. 7. The algorithm for processing angles of motion for a robot – IV thread



Rys. 8. Widok pomieszczenia w aplikacji VR w silniku Unity
Fig. 8. The view of VR room in Unity software

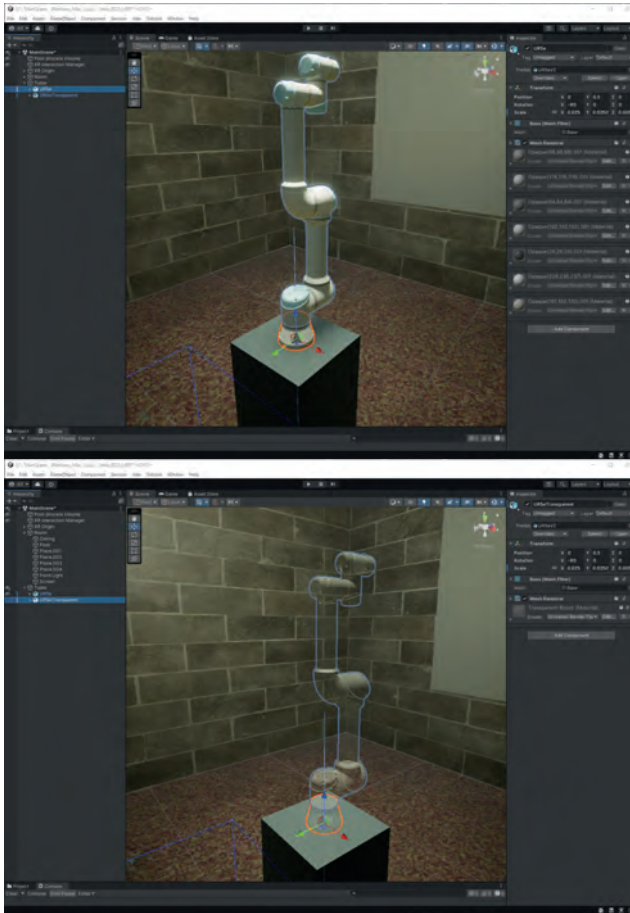
Czwarty wątek (rys. 7) odpowiedzialny jest wyłącznie za odczyt i przesłanie aktualnych kątów odchylenia przegubów robota. Utworzenie aktualnych kątów odchylenia przegubów robota. Aby umożliwić płynną wizualizację robota w aplikacji Unity. Kąty odczytane są za pomocą polecenia `get_actual_joint_positions()`, które to zwraca tablicę liczb zmiennoprzecinkowych. Następnie za pomocą polecenia `socket_send_string()` tablica liczb zostaje zamieniona na ciąg znaków, które przesłane są do Unity.

5. Aplikacja VR sterująca robotem

Aplikacja VR obejmuje pomieszczenie, w którym znajduje się cyfrowy bliźniak robota oraz ekran, na którym wyświetlony zostanie obraz z kamery i punktowe źródło światła (rys. 8).

Do zaprezentowania kolejnych położeń manipulatora zastosowano modele 3D udostępnione przez producenta robota. Model ten został uproszczony w oprogramowaniu Blender, aby ułatwić manipulowanie nim i zmniejszyć stopień skomplikowania siatki wielokątów. Do materiału modelu przedstawiającego docelowe położenie ramienia został przypisany kanał alfa, aby osiągnąć efekt przezroczystości (rys. 9).

Podczas działania aplikacji wychylenie przegubów w modelach zmienia się, co pozwala użytkownikowi ocenić docelowe oraz aktualne położenie manipulatora. Dodatkowo wyświetlony jest obraz z kamery jako tekstura ekranu. Skrypty odpowiedzialne za działanie aplikacji zostały przedstawione za pomocą schema-

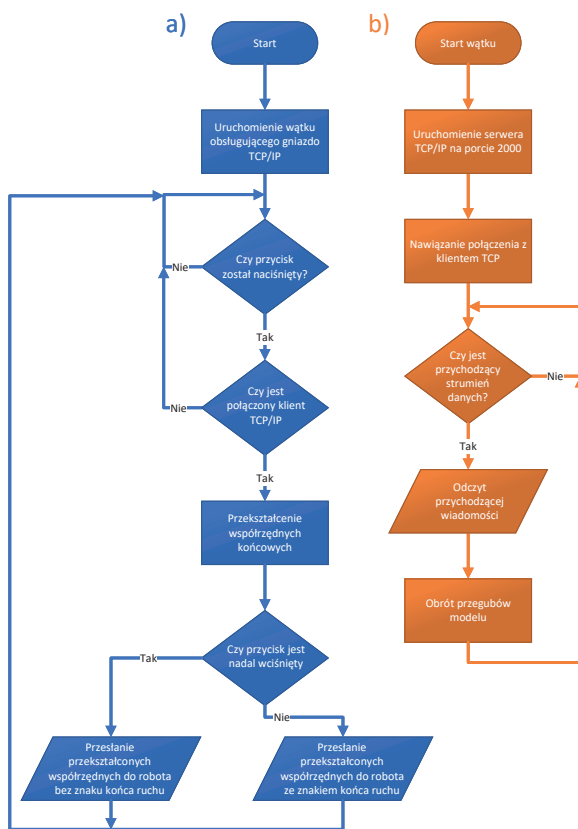


Rys. 9. Widok cyfrowego bliźniaka robota ramieniowego w Unity
Fig. 9. The view of digital twin of robotic arm in Unity software

tów blokowych (rys. 10 i rys. 11). Podzielone zostały na skrypty odpowiedzialne za model docelowego położenia robota i model położenia bieżącego.

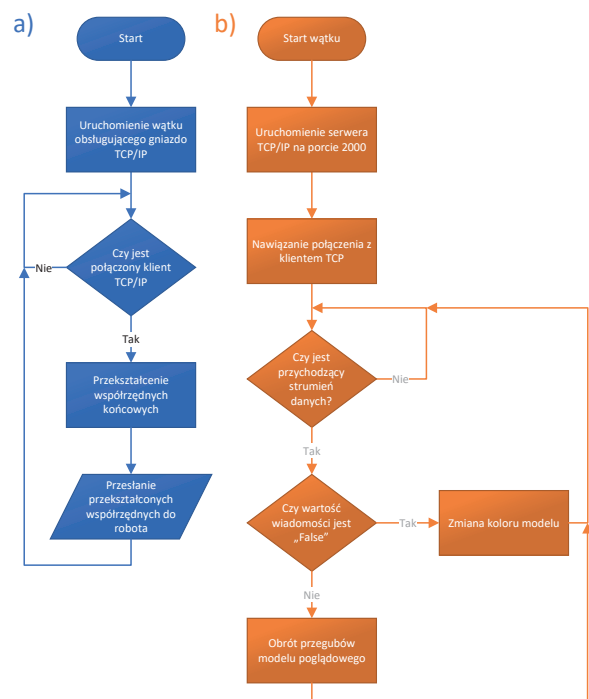
Pierwszy algorytm jest odpowiedzialny za działanie rzeczywistego robota (rys. 10). Skrypt ten działa w dwóch wątkach, gdzie główny wątek jest odpowiedzialny za przesyłanie pozycji końcowej robota, natomiast drugi wątek steruje cyfrowym bliźniakiem. Główny wątek skryptu odpowiada za komunikację z robotem w czasie rzeczywistym. W ramach tego wątku skrypt odczytuje pozycję wskaźnika, czyli jego położenie i obrót. Następnie przekształca te dane, np. zamienia jednostki na milimetry. Przed wysłaniem współrzędnych sprawdzany jest stan przycisku, jeśli jest nadal wciśnięty, robot będzie się poruszał z mieszaniem. Drugi wątek skryptu jest odpowiedzialny za sterowanie cyfrowym bliźniakiem robota. Ten wątek skryptu odpowiada za aktualizowanie pozycji modelu na podstawie odczytanych danych z rzeczywistego robota. Obejmuje to aktualizację położenia i orientacji modelu wirtualnego robota, tak aby był on zawsze zgodny z rzeczywistym robotem. Oba wątki skryptu działają w pętli nieskończonej, a dzięki prostocie obliczeniowej utrzymują zgodność między rzeczywistym robotem a jego wirtualnym modelem.

Drugi algorytm (rys. 11) jest odpowiedzialny za działanie modelu podglądu położenia końcowego. Skrypt ten działa w dwóch wątkach, gdzie główny wątek jest odpowiedzialny za przesyłanie pozycji docelowej robota, natomiast drugi wątek steruje modelem przedstawiającą przyszłą pozycję robota. Główny wątek skryptu odpowiada za przesyłanie do sterownika punktu docelowego, do którego robot ma się przemieścić. Sterownik sprawdza, czy można obliczyć kinematykę odwrotną dla podanego punktu. Jeśli tak, to ją oblicza. Obliczone kąty obrotu podane są do aplikacji VR. Drugi wątek skryptu jest odpowiedzialny za sterowanie modelem, który przedstawia docelową pozycję robota. Jeśli przesłany zostaje ciąg znaków „False” zmieniający jest kolor modelu docelowego, w przeciwnym razie model jest obracany zgodnie z obliczonymi kątami. Drugi wątek stale aktualizuje model docelowej pozycji robota, informując użytkownika o ułożeniu manipulatora po przesłaniu polecenia ruchu.



Rys. 10. Algorytm docelowego położenia robota: a) przesyłanie poprawnych danych, b) ruch fizyczny robota

Fig. 10. Algorithm of the target position of the robot: a) transmission of correct data, b) movement of the physical robot



Rys. 11. Algorytm bieżącego położenia modelu 3D robota:

a) przesyłanie poprawnych danych, b) ruchu i modyfikacja modelu podglądowego

Fig. 11. Algorithm for current position of 3D robot model: a) transmission of correct data, b) movement and modification of preview model

6. Testy sterowania fizycznym robotem z poziomu aplikacji VR

W celu weryfikacji działania aplikacji oraz jej połączenia z fizycznym ramieniem UR5 przeprowadzono szereg testów. Testy polegały na przeniesieniu wybranego elementu z punktu A, zdefiniowanego jako pozycja początkowa do punktu B, zdefiniowanego jako punkt końcowy. W testach wzięli udział zarówno studenci dysponujący doświadczeniem z zakresu poruszania się i obsługi aplikacji wirtualnej rzeczywistości, jak i osoby pierwszy raz pracujące w środowisku VR. Pierwsze pomiary wykonano startując z punktu A o współrzędnych $X = 10$, $Y = 10$, $Z = 10$. Wartości wyrażone są w milimetrach jako jednostka odległości od punktu (0, 0, 0) deklarowanej przez robota. Punkt końcowy (B) miał współrzędne $X = 200$, $Y = 200$, $Z = 200$. Uczestnicy badań mieli za zadanie dotrzeć z punktu A do punktu B za pomocą kontrolera VR, mając podgląd obrazu z kamery skierowanej bezpośrednio na rzeczywistego robota oraz cyfrowego bliźniaka. Położenie cyfrowego bliźniaka odpowiadało szacowanemu końcowemu położeniu robota. Ruch kontrolera w zadanym kierunku wprawiał ramię robotyczne w ruch w tej samej osi. W trakcie pomiarów użyto gogle Oculus Quest 2 wraz z kontrolerami wchodzącymi w skład zestawu. Do przesyłania rzeczywistego obrazu do aplikacji VR wykorzystano kamerę Logitech C920 HD. Po zakończeniu każdego pomiaru z panelu kontrolnego robota UR5 odczytywane były współrzędne (X, Y, Z) zatrzymania ramienia (tab. 1). Wartości te były porównywane z zadanymi współrzędnymi (wartościami startowymi), jednocześnie obliczano wyrażoną w procentach dokładność dotarcia do punktu B (tab. 2).

Na podstawie analizy otrzymanych wyników można stwierdzić, że dokładność wykonania zadań w VR dla pierwszych ośmiu osób jest większa w porównaniu z wynikami uzyskanymi przez kolejne osiem osób. Jest to związane z doświadczeniem

Tab. 1. Współrzędne zatrzymania końca ramienia robota

Tab. 1. The coordinates of the stop of the end of the robotic arm

Próby	Punkt zatrzymania ramienia [mm]		
	X	Y	Z
1	172	161	192
2	167	172	166
3	171	169	159
4	158	187	155
5	149	169	178
6	173	162	182
7	181	171	191
8	178	175	187
9	142	154	179
10	144	160	159
11	159	164	155
12	164	152	173
13	149	143	168
14	155	163	163
15	161	156	171
16	152	163	162

Tab. 2. Dokładność dotarcia do punktu docelowego oraz błąd odległości

Tab. 2. The accuracy of reaching the destination point and distance of the stopped robot

Próby	Dokładność dotarcia do punktu docelowego [%]			Błąd odległości [%]		
	X	Y	Z	X	Y	Z
1	85,26	79,47	95,79	14,74	20,53	4,21
2	82,63	85,26	82,11	17,37	14,74	17,89
3	84,74	83,68	78,42	15,26	16,32	21,58
4	77,89	93,16	76,32	22,11	6,84	23,68
5	73,16	83,68	88,42	26,84	16,32	11,58
6	85,79	80	90,53	14,21	20	9,47
7	90	84,74	95,26	10	15,26	4,74
8	88,42	86,84	93,16	11,58	13,16	6,84
9	69,47	75,79	88,95	30,53	24,21	11,05
10	70,53	78,95	78,42	29,47	21,05	21,58
11	78,42	81,05	76,32	21,58	18,95	23,68
12	81,05	74,74	85,79	18,95	25,26	14,21
13	73,16	70	83,16	26,84	30	16,84
14	76,32	80,53	80,53	23,68	19,47	19,47
15	79,47	76,84	84,74	20,53	23,16	15,26
16	74,74	80,53	80	25,26	19,47	20

pracy w VR poszczególnych uczestników testów. Dla grupy pierwszych ośmiu osób wykonywane ruchy były bardziej naturalne. Kolejne pomiary przeprowadzono z osobami, które nie miały wcześniej styczności z VR. Po raz pierwszy zapoznali się z elementami VR przed realizacją doświadczenia, zasadami pracy w takim środowisku oraz zadaniami, które mieli wykonać. Ich umiejętność poruszania się w świecie wirtualnym były na poziomie adekwatnym dla osób początkujących.

Warto zaznaczyć, że uczestnicy najlepiej radzili sobie z dotarciem do punktu docelowego w osi Z. Jest to oś wysokości od podłoża. Po przedstawieniu uczestnikom wartości docelowej okazało się, że było im znacznie łatwiej dotrzeć do zadanej wysokości niż do wartości punktu docelowego zdefiniowanych w osiach X i Y.

7. Podsumowanie

Rozwój przemysłu wprowadza nowe technologie i rozwiązania umożliwiające efektywniejsze zarządzanie produkcją lub logistyką. Szkolenia przeprowadzane w wirtualnych środowiskach dostarczają nowych możliwości, w wielu przypadkach będąc efektywniejszymi i oszczędniejszymi. Sterowanie maszynami za pomocą własnych ruchów jest jednak elementem uznawanym wciąż jako aspekt teoretyczny. Testy wykazały, iż niezbędne jest dopracowanie w przyszłości algorytmów tak aby ruch robota współpracującego stał się jak najbardziej naturalny, szczególnie ważne jest do dla odczucia naturalnego ste-

rowania przez użytkownika. Przeprowadzone doświadczenia wykazały, że na sprawność pracy w środowisku VR, a co za tym idzie na dokładność wykonanych zadań ma wpływ posiadane doświadczenie i czas spędzony przez użytkowników w VR. Biorąc pod uwagę fakt, że technologia VR może koegzystować z rzeczywistą infrastrukturą przemysłową, a co więcej działania wykonywane w wirtualnym środowisku mogą być automatycznie odwzorowywane na linii produkcyjnej precyzją ruchów operatorów VR ma istotne znaczenie dla jakości wykonywanej pracy. Dlatego też, w dalszych pracach przewidziane jest zaktualizowanie sterowania aby ruch robota odbywał się płynniej i bardziej naturalnie. Planowane jest również zaimplementowanie aplikacji na innych platformach robotycznych oraz umożliwienie sterowania dwoma ramionami w tym samym czasie.

Podziękowania

Projekt finansowany w ramach programu Ministra Edukacji i Nauki pod nazwą „Regionalna Inicjatywa Doskonałości” w latach 2019–2023 nr projektu 027/RID/2018/19 kwota finansowania 11 999 900 zł.

Bibliografia

- Kuehn B.M., *Virtual and Augmented Reality Put a Twist on Medical Education*, “JAMA”, Vol. 319, No. 8, 2018, 756–758, DOI: 10.1001/jama.2017.20800.
- Moro C., Štromberga Z., Raikos A., Stirling A., *The effectiveness of virtual and augmented reality in health sciences and medical anatomy*, “Anatomical Sciences Education”, Vol. 10, No. 6, 2017, 549–559, DOI: 10.1002/ase.1696.
- Rajeswaran P., Varghese J., Kumar P., Vozenilek J., Kesavadas T., *AirwayVR: Virtual Reality Trainer for Endotracheal Intubation*. 2019 IEEE Conference on Virtual Reality and 3D User Interfaces (VR), 2019, 1345–1346, DOI: 10.1109/VR.2019.8797998.
- Khalilia W.M., Gombar M., Palkova Z., Palko M., Valicek J., Harnicarova M., *Using Virtual Reality as Support to the Learning Process of Forensic Scenarios*. “IEEE Access”, Vol. 10, 2022, 83297–83310. DOI: 10.1109/ACCESS.2022.3196471.
- Desselle M.R., Brown R.A., James A.R., Midwinter M.J., Powell S.K., Woodruff M.A., *Augmented and Virtual Reality in Surgery*, “Computing in Science & Engineering”, Vol. 22, No. 3, 2020, 18–26, DOI: 10.1109/MCSE.2020.2972822.
- Pinter C., Lasso A., Choueib S., Asselin M., Fillion-Robin J.-C., Vimort J.-B., Martin K., Jolley M.A., Fichtinger G., *SlizerVR for Medical Intervention Training and Planning in Immersive Virtual Reality*, “IEEE Transactions on Medical Robotics and Bionics”, Vol. 2, No. 2, 2020, 108–117, DOI: 10.1109/TMRB.2020.2983199.
- Davila Delgado J.M., Oyedele L., Demian P., Beach T., *A research agenda for augmented and virtual reality in architecture, engineering and construction*. “Advanced Engineering Informatics”, Vol. 45, 2020, DOI: 10.1016/j.aei.2020.101122.
- Joshi S., Hamilton M., Warren R., Faucett D., Tian W., Wang Y., Ma J., *Implementing Virtual Reality technology for safety training in the precast/ prestressed concrete industry*, “Applied Ergonomics”, Vol. 90, 2021, DOI: 10.1016/j.apergo.2020.103286.
- Wang H., Zhang B., Zhang T., Jakacky A., *Tele-operating a Collaborative Robot for Space Repairs with Virtual Reality*, IEEE 9th Annual International Conference on CYBER Technology in Automation, Control, and Intelligent Systems (CYBER), 2019, 175–180, DOI: 10.1109/CYBER46603.2019.9066493.
- Clifford R.M.S., Khan H., Hoermann S., Billingham M., Lindeman R.W., *The Effect of Immersive Displays on Situation Awareness in Virtual Environments for Aerial Fire-fighting Air Attack Supervisor Training*. IEEE Conference on Virtual Reality and 3D User Interfaces (VR), 2018, DOI: 10.1109/VR.2018.8446139.
- Li Y., *Cabin Operation and Management Model Based on VR Technology*, 13th International Conference on Measuring Technology and Mechatronics Automation (ICMTMA), 2021, DOI: 10.1109/ICMTMA52658.2021.00013.
- Sun X., Wang X., Wang X., *Design of Vehicle Engineering Teaching System Based on VR Technology*. 5th Asian Conference on Artificial Intelligence Technology (ACAIT), 2021, 414–418, DOI: 10.1109/ACAIT53529.2021.9730892.
- Ciprian Firu A., Ion Tapîrdea A., Ioana Feier A., Drăghici G., *Virtual reality in the automotive field in industry 4.0*. “Materials Today: Proceedings”, Vol. 45, Part 5, 2021, 4177–4182, DOI: 10.1016/j.matpr.2020.12.037.
- Dourado A.O., Martin C.A., *New concept of dynamic flight simulator, Part I*, “Aerospace Science and Technology”, Vol. 30, No. 1, 2013, 79–82, DOI: 10.1016/j.ast.2013.07.005.
- Knerr B.W., *Immersive Simulation Training for the Dismounted Soldier*, University of Michigan Library, 2007.
- Moon B.-H., Choi J.-W., Jung K.-T., Kim D.-H., Song H.-J., Gil K.-J., Kim J.-W., *Connecting motion control mobile robot and VR content*, 14th International Conference on Ubiquitous Robots and Ambient Intelligence (URAI), 2017, 355–359, DOI: 10.1109/URAI.2017.7992750.
- Knopp S., Lorenz M., Pelliccia L., Klimant P., *Using Industrial Robots as Haptic Devices for VR-Training*, IEEE Conference on Virtual Reality and 3D User Interfaces (VR), 2018, 607–608, DOI: 10.1109/VR.2018.8446614
- Inamura T., Mizuchi Y., *Robot Competition to Evaluate Guidance Skill for General Users in VR Environment*. 14th ACM/IEEE International Conference on Human-Robot Interaction (HRI), 2019, 552–553, DOI: 10.1109/HRI.2019.8673218.
- Chen H., Liu F., Yang Y., Meng W., *MultiVR: Digital Twin and Virtual Reality Based System for Multi-people Remote Control Unmanned Aerial Vehicles*, 17th International Conference on Control, Automation, Robotics and Vision (ICARCV), 2022, 647–652, DOI: 10.1109/ICARCV57592.2022.10004244.
- Losey S., United States Air Force Academy, [www.airforcetimes.com/news/your-air-force/2019/10/08/academy-brings-vr-pilot-training-to-cadets].
- Volkswagen AG, [www.volkswagen-group.com/en/press-releases/what-does-a-dot-dot-dot-vr-developer-actually-do-at-volkswagen-16225].

Cobot Control Using Virtual Reality for the Needs of Industry 4.0

Abstract: Industry 4.0 is undergoing continuous evolution. Management and monitoring systems implemented in the industrial infrastructure are able to collect and process more and more data, then, based on dedicated algorithms, make decisions, and consequently send specific control commands to devices and technological processes. At the same time, there is growing interest in combining devices with other seemingly disparate technologies. An example of creating such coexistence is the integration of sensory solutions with virtual reality. Such activities contribute to the creation of digital twins of devices and processes, which in turn enables remote and in many cases automatic control of physical objects and the course of real processes. In the course of this research, such an approach has been used to develop a solution that allows a robot to be controlled from anywhere and at any time. This work presents a prototype of a solution providing remote control of a cobot arm robot.

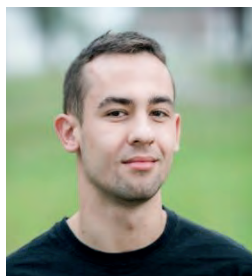
Keywords: virtual reality, robotics, Industry 4.0, digital twin

mgr inż. Mateusz Salach

m.salach@prz.edu.pl

ORCID: 0000-0002-9199-3460

Asystent w Zakładzie Systemów Złożonych Wydziału Elektrotechniki i Informatyki Politechniki Rzeszowskiej. Zajmuje się rozwiązaniami i badaniami z zakresu Internetu Rzeczy, VR oraz Smart City.



mgr inż. Patryk Organiściak

org@prz.edu.pl

ORCID: 0000-0002-5277-4038

Pracownik naukowo-dydaktyczny na Politechnice Rzeszowskiej, gdzie prowadzi zajęcia z technologii informacyjnych, programowania, przetwarzania danych oraz cyberbezpieczeństwa. Jednocześnie realizuje projekty komercyjne w ramach własnego software house, specjalizując się w tworzeniu oprogramowania dla biznesu.



dr inż. Andrzej Paszkiewicz

andrzejp@prz.edu.pl

ORCID: 0000-0001-7573-3856

Adiunkt w Zakładzie Systemów Złożonych Politechniki Rzeszowskiej. Stopień doktora nauk technicznych w zakresie informatyki uzyskał w 2009 r. na Politechnice Łódzkiej. Autor około 100 publikacji w krajowych i międzynarodowych czasopismach naukowych i materiałach konferencyjnych. Organizował warsztaty i konferencje z zakresu sieci komputerowych, inżynierii oprogramowania oraz Przemysłu 4.0. Jego zainteresowania badawcze skupiają się na procesach zachodzących w strukturach sieciowych, VR/AR, a także na rozwoju technologii dla przemysłu.



mgr inż. Arkadiusz Stęchły

arkadiusz.stechly@kneiti.prz.edu.pl

ORCID: 0009-0006-5200-761X

Student Politechniki Rzeszowskiej, gdzie ukończył studia pierwszego stopnia na kierunku Automatyka i Robotyka. Kontynuuje studia drugiego stopnia na kierunku Informatyka na specjalności Cyberbezpieczeństwo i technologie chmurowe. Realizuje projekty naukowe w ramach koła naukowego, specjalizując się w tworzeniu oprogramowania wykorzystującego technologię wirtualnej i rozszerzonej rzeczywistości.



Prof. dr hab. inż. Grzegorz Budzik

gbudzik@prz.edu.pl

ORCID: 0000-0003-3598-2860

Profesor w Katedrze Konstrukcji Maszyn na Wydziale Budowy Maszyn i Lotnictwa Politechniki Rzeszowskiej. Obszar jego badań obejmuje inżynierię mechaniczną, wytwarzanie przyrostowe, systemy CAD, rozwój nanomateriałów przy użyciu druku 3D i technik szybkiego prototypowania. Był prodziekanem ds. badań naukowych na Wydziale Budowy Maszyn i Lotnictwa (2012–2016) oraz prorektorem ds. badań naukowych (2016–2020) Politechniki Rzeszowskiej. Wniósł obszerny wkład w światową literaturę dotyczącą komputerowego wspomaganie inżynierii i wytwarzania przyrostowego w Rapid Prototyping Journal, Journal of Cleaner Production, Polymers, Sensors, Nanomaterials. Jest współautorem 7 książek i 15 rozdziałów. Twórca 19 patentów i 20 licencjonowanych technologii dla przemysłu oraz start-upów. Jest ekspertem Polskiego Komitetu Normalizacyjnego oraz Międzynarodowej Organizacji Normalizacyjnej ISO sekcja TC 261/WG 2.

