Wiele celów, jedno rozwiązanie – o projektowaniu nowoczesnej automatyki

Paweł Skruch

Akademia Górniczo-Hutnicza im. Stanisława Staszica w Krakowie

Streszczenie

W artykule przedstawiono proces projektowania nowoczesnych systemów automatyki jako sztukę godzenia wielu, często sprzecznych celów. Pokazano, że wybór rozwiązania nie wynika wyłącznie z kryteriów technicznych, lecz jest silnie uwarunkowany konkretnym zastosowaniem przemysłowym, w którym kluczową rolę odgrywają także koszty, niezawodność, bezpieczeństwo czy łatwość wdrożenia. W przeciwieństwie do podejścia typowego dla badań naukowych, koncentrującego się na demonstracji wybranych metod, praktyka inżynierska wymaga uwzględnienia wielu dodatkowych ograniczeń i kompromisów. W artykule omówiono te właśnie czynniki, które w rzeczywistych projektach często decydują o sukcesie rozwiązania, jednocześnie znacząco zwiększając złożoność procesu projektowego.

Wprowadzenie

Podczas projektowania systemów automatyki istotna część nakładu pracy koncentruje się na zapewnieniu tak zwanej zamierzonej funkcjonalności (ang. *intended functionality*). Przez zamierzoną funkcjonalność należy rozumieć zdolność systemu do realizacji określonych zadań i świadczenia usług zgodnie z wymaganiami użytkownika. Uwaga badaczy, inżynierów i projektantów jest skupiona na tym właśnie aspekcie, gdyż stanowi on podstawową cechę definiującą projektowany układ. Współcześnie jednak, w warunkach dynamicznego rozwoju technologii oraz rosnącej konkurencji rynkowej, sama funkcjonalność okazuje się niewystarczająca do osiągnięcia sukcesu komercyjnego. Wielu producentów jest w stanie zaoferować rozwiązania spełniające założone wymagania funkcjonalne, jednak jedynie nieliczni osiągają dominującą pozycję na rynku.

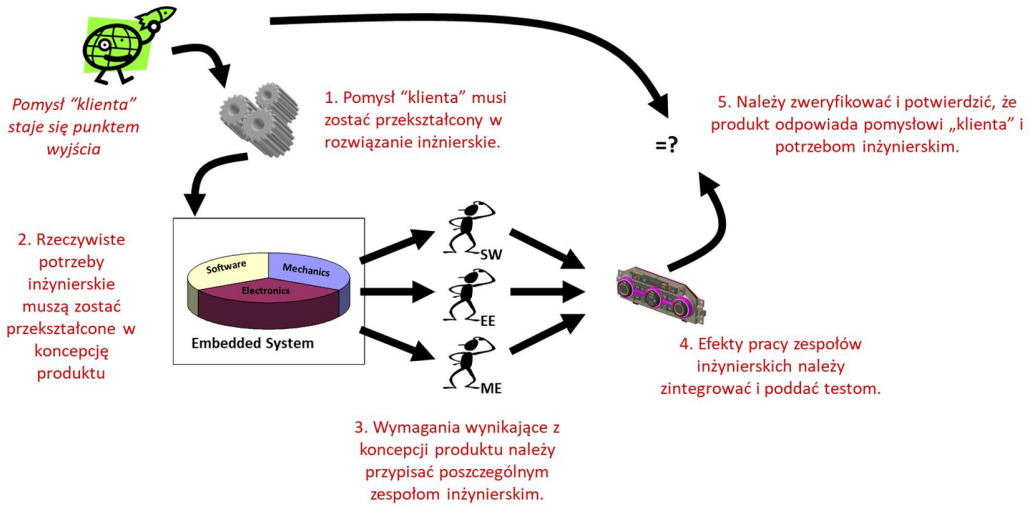
W artykule przedstawiono analizę czynników istotnych z punktu widzenia projektowania inżynierskiego, które powinny być uwzględnione i odpowiednio zaadresowane w procesie projektowania, aby zwiększyć szanse powodzenia wdrożenia w warunkach przemysłowych. Pokazano, że wybór ostatecznego rozwiązania rzadko wynika

wyłącznie z przesłanek technicznych, lecz jest silnie uwarunkowany kontekstem zastosowania i realiami przemysłowymi.

Charakterystyka współczesnych układów automatyki

Współczesny układ automatyki stanowi integrację komponentów sprzętowych, programowych oraz mechanicznych, zorganizowanych w postaci w pełni funkcjonalnego systemu. Zintegrowany system powinien spełniać określone wymagania funkcjonalne i niefunkcjonalne, które należy jednoznacznie zdefiniować na wczesnym etapie projektu (zob. rys. 1). Pomimo że system obejmuje warstwę sprzętową, mechaniczną oraz programową, to właśnie oprogramowanie coraz częściej odgrywa dominującą rolę w kształtowaniu funkcjonalności całego układu. W praktyce oznacza to, że funkcjonalności możliwe do realizacji na poziomie oprogramowania są implementowane właśnie w tej warstwie. Jeszcze stosunkowo niedawno projektanci stawali przed dylematem podziału funkcjonalności pomiędzy sprzęt a oprogramowanie. Obecnie obserwuje się wyraźną zmianę tego podejścia – dominującym trendem jest maksymalizacja zakresu funkcji realizowanych programowo. Tendencja ta jest dodatkowo wzmacniana przez rozwój narzędzi opartych na dużych modelach językowych, umożliwiających automatyczne generowanie kodu.

Realizacja funkcjonalności w postaci oprogramowania wiąże się z istotnymi korzyściami, takimi jak możliwość relatywnie łatwej modyfikacji oprogramowania, jego aktualizacji oraz rozszerzania systemu o nowe funkcje. Równolegle obserwuje się zmianę modeli biznesowych w kierunku podejścia usługowego, w którym oprogramowanie podlega ciągłemu rozwojowi i dystrybucji w formie subskrypcji. Jest to też zgodne z kierunkiem wyznaczonym przez czwartą rewolucję przemysłową określaną mianem Przemysłu 4.0 [1].



Rysunek 1. Schemat procesu projektowania systemów automatyki [2].

Dostępność zasobów sprzętowych

Funkcjonalność realizowana za pomocą dedykowanego oprogramowania jest najczęściej uruchamiana na dedykowanej platformie sprzętowej skonstruowanej na potrzeby danego urządzenia. Każda platforma tego typu (ang. *embedded system*) dysponuje ograniczonymi zasobami, co w kontekście oprogramowania przekłada się na limity dostępnej pamięci operacyjnej oraz mocy obliczeniowej. Naturalną konsekwencją tych ograniczeń jest konieczność projektowania algorytmów w taki sposób, aby wynikowy program mieścił się w dostępnej pamięci, zapotrzebowanie na pamięć operacyjną (np. liczba i rozmiar zmiennych) nie przekraczało dostępnych zasobów pamięci RAM, a wymagania obliczeniowe mogły zostać spełnione bez przeciążania procesora lub systemu wieloprocesorowego. W zastosowaniach przemysłowych dodatkowo uwzględnia się tzw. bufor bezpieczeństwa zasobów, istotny z punktu widzenia dalszego rozwoju produktu. Oznacza to, że dostępne zasoby nie powinny być wykorzystywane w całości – przykładowo, przy 100 MB dostępnej pamięci jedynie około 75 MB przeznacza się na implementację bieżącej funkcjonalności, pozostawiając pozostałą część na przyszłe rozszerzenia systemu. Ponadto, szczególnie w systemach krytycznych pod względem bezpieczeństwa, nakłada się ograniczenia na maksymalne obciążenie procesora, które zazwyczaj nie powinno przekraczać określonego progu (np. 80%). Przekroczenie tej wartości może prowadzić do niepożądanych efektów, takich jak wzrost temperatury układu elektronicznego, niestabilność działania czy nawet reset systemu, co jest niedopuszczalne w warunkach przemysłowych.

Powyższe ograniczenia oznaczają, że niektóre, nawet bardzo dobre rozwiązania algorytmiczne, ze względu na wysokie wymagania sprzętowe, nie są możliwe do wdrożenia przemysłowego lub ich implementacja wiąże się z kosztami, które z punktu widzenia biznesowego są nieuzasadnione.

Praca w czasie rzeczywistym

Czas rzeczywisty w układach sterowania oznacza zdolność systemu do wykonania wszystkich niezbędnych operacji związanych z przetwarzaniem danych wejściowych oraz generowaniem sygnałów wyjściowych w ściśle określonych ramach czasowych. Wymaganie to ma kluczowe znaczenie w systemach krytycznych z punktu widzenia bezpieczeństwa. Zapewnienie pracy w reżimie czasu rzeczywistego zależy zarówno od doboru odpowiedniej platformy sprzętowej i systemu operacyjnego, jak i od takiego projektowania algorytmów, które w sposób świadomy uwzględnia ograniczenia czasowe. Operacje obliczeniowe o wysokiej złożoności, stanowiące element projektowanych algorytmów, mogą nie spełniać wymagań czasowych, co prowadzi do poważnych konsekwencji. Ten problem jest szczególnie widoczny w systemach przetwarzających duże ilości danych (np. dane wizyjne, chmury punktów lidarowych), gdzie wymagana jest wysoka moc obliczeniową.

Przykładowo, współczesne systemy akwizycji obrazu pracują z częstotliwościami rzędu 60, a nawet 120 klatek na sekundę, co oznacza, że pojedyncza ramka musi zostać przetworzona w ściśle określonym oknie czasowym (np. około 20 ms). Niedotrzymanie tych ograniczeń może skutkować generowaniem sygnałów sterujących na podstawie nieaktualnych danych, co bezpośrednio wpływa na pogorszenie jakości sterowania oraz bezpieczeństwa systemu. Stosowane rozwiązania polegające na redukcji liczby przetwarzanych danych (np. pomijaniu części ramek lub skanów pomiarowych) w celu spełnienia wymagań czasu rzeczywistego są zazwyczaj niewystarczające i mogą prowadzić do utraty istotnych informacji (np. błędnej estymacji trajektorii, utraty krótkotrwałych zdarzeń), a tym samym do degradacji funkcjonalności systemu.

Kompaktowość konstrukcji

Ograniczenia gabarytowe projektowanego układu automatyki stanowią istotne wyzwanie inżynierskie, często nieuwzględniane na etapie koncepcyjnym. Każdy fizyczny układ automatyki jest projektowany jako element większego systemu, co implikuje konieczność spełnienia określonych wymagań dotyczących wymiarów przestrzennych. Niespełnienie tych wymagań może uniemożliwić integrację układu, np. z powodu braku wystarczającej przestrzeni montażowej w docelowym miejscu instalacji. Problem ten jest szczególnie widoczny w aplikacjach takich jak robotyka czy systemy motoryzacyjne, gdzie dostępna przestrzeń jest ściśle ograniczona, a rozmieszczenie komponentów musi być precyzyjnie zaplanowane. W konsekwencji projektanci elektroniki i mechaniki napotykają istotne ograniczenia w doborze komponentów elektronicznych oraz elementów konstrukcyjnych, takich jak obudowy czy systemy chłodzenia. Przykładowo, ograniczenia dotyczące wysokości obudowy układu mogą wykluczyć zastosowanie określonych komponentów, takich jak kondensatory o większych gabarytach, jeżeli ich użycie prowadziłoby do przekroczenia dopuszczalnych wymiarów całego urządzenia. Tego rodzaju ograniczenia wymagają ścisłej współpracy między obszarami projektowania mechanicznego i elektronicznego oraz uwzględnienia kompromisów już na wczesnych etapach procesu projektowego.

Efektywność energetyczna

Algorytmy o wysokiej złożoności obliczeniowej są bezpośrednio powiązane z dużym zapotrzebowaniem na energię [3]. Aspekt ten ma szczególne znaczenie w systemach zasilanych bateryjnie, w których czas pracy na jednym cyklu ładowania stanowi jeden z kluczowych parametrów determinujących sukces komercyjny oraz akceptację użytkowników. W konsekwencji obserwuje się wyraźny trend w kierunku opracowywania rozwiązań algorytmicznych o wysokiej efektywności energetycznej. Zagadnienie to jest szczególnie istotne w kontekście zaawansowanych metod uczenia maszynowego, zwłaszcza dużych modeli, których implementacja często wymaga wykorzystania akceleratorów sprzętowych, takich jak jednostki GPU (ang. *Graphics Processing Unit*), charakteryzujących się znacznym poborem mocy. W rezultacie wiele algorytmów

osiągających wysokie wskaźniki wydajnościowe nie znajduje szerokiego zastosowania praktycznego ze względu na nadmierne wymagania energetyczne, co istotnie ogranicza ich potencjał aplikacyjny.

Niezawodność

Każde oprogramowanie zawiera błędy, co wynika z natury procesu jego wytwarzania [4]. Dotyczy to zarówno oprogramowania tworzonego manualnie, jak i generowanego w sposób zautomatyzowany. Całkowite wyeliminowanie błędów, a więc stworzenie oprogramowania w pełni wolnego od defektów, jest w praktyce niemożliwe. W systemach opartych na oprogramowaniu dąży się zatem do osiągnięcia poziomu liczby i znaczenia błędów, który pozostaje akceptowalny z punktu widzenia funkcjonalności i bezpieczeństwa systemu. Kluczową rolę w tym procesie odgrywa testowanie oprogramowania. Jego celem nie jest jedynie wykazanie poprawności działania systemu, lecz przede wszystkim identyfikacja istniejących błędów. W tym ujęciu testowanie można traktować jako problem optymalizacyjny, polegający na zaprojektowaniu możliwie niewielkiego zbioru testów, który umożliwia wykrycie maksymalnej liczby defektów. Testowanie oprogramowania, mimo swojej kluczowej roli, jest często jednak postrzegane jako kosztowna i czasochłonna aktywność projektowa, co w praktyce prowadzi do prób jego ograniczania kosztem jakości systemu.

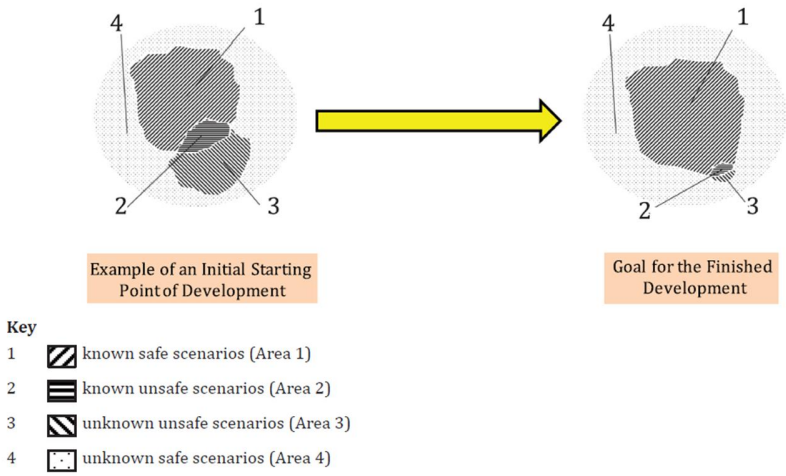
W systemach opartych na oprogramowaniu niezawodność jest ściśle powiązana z jego jakością, a testowanie stanowi podstawowe narzędzie umożliwiające jej ilościową ocenę. Znaczenie błędów oraz ich wpływ na działanie systemu mają również kluczowe znaczenie z perspektywy użytkowników końcowych. Wadliwe oprogramowanie sterujące może istotnie pogorszyć odbiór produktu, a w konsekwencji wpływać na decyzje zakupowe i preferencje klientów.

Bezpieczeństwo

Systemy automatyki przemysłowej, ze względu na swoje zastosowanie, muszą spełniać rygorystyczne wymagania w zakresie bezpieczeństwa, rozumianego jako ograniczanie ryzyka wystąpienia sytuacji niebezpiecznych mogących prowadzić do strat materialnych lub zagrożenia zdrowia i życia [5]. Ponieważ funkcjonalność tych systemów w dużej mierze opiera się na oprogramowaniu, a każde oprogramowanie zawiera błędy, stworzenie systemu całkowicie wolnego od takiego ryzyka jest w praktyce niemożliwa. W konsekwencji bezpieczeństwo analizuje się w kategoriach akceptowalnego poziomu ryzyka [6].

Redukcja ryzyka (mitygacja) realizowana jest poprzez zastosowanie odpowiednich mechanizmów i warstw bezpieczeństwa, które pełnią rolę niezależnych komponentów nadzorczych, monitorujących stan systemu oraz reagujących na sytuacje niepożądane (zob. rys. 2). W praktyce przemysłowej oznacza to, że znaczna część nakładu pracy projektowej i implementacyjnej poświęcana jest obsłudze błędów, stanów wyjątkowych

oraz scenariuszy awaryjnych. W rezultacie implementacja podstawowej funkcjonalności systemu stanowi często jedynie część całkowitego wysiłku inżynierskiego.



Rysunek 2 Cele bezpieczeństwa w procesie rozwoju systemu zgodnie z normą [6].

Odporność cybernetyczna

Obecność oprogramowania w układach automatyki wiąże się nie tylko z rozszerzeniem funkcjonalności, lecz również z pojawieniem się zagrożeń wynikających z nieautoryzowanych ataków cybernetycznych [7]. Systemy te, jako elementy infrastruktury cyber-fizycznej, są szczególnie narażone na takie incydenty jak ataki typu „man-in-the-middle", manipulacja danymi telemetrycznymi, nieautoryzowany dostęp do systemów sterowania czy zakłócenia komunikacji. Dodatkowym wyzwaniem są zagrożenia specyficzne dla systemów wykorzystujących sztuczną inteligencję, w tym zanieczyszczanie danych uczących, ataki na modele (np. *adversarial attacks*) oraz podatności w łańcuchu dostaw oprogramowania.

Ponieważ każde oprogramowanie zawiera błędy, nie jest również możliwe zagwarantowanie pełnego, stuprocentowego poziomu cyberbezpieczeństwa. Z tego względu ochrona cybernetyczna – analogicznie do bezpieczeństwa funkcjonalnego – powinna być rozpatrywana w kategoriach zarządzania ryzykiem, obejmującego zarówno prawdopodobieństwo wystąpienia ataku, jak i jego potencjalne skutki dla działania systemu.

Z technicznego punktu widzenia zapewnienie odporności cybernetycznej wymaga uwzględnienia odpowiednich mechanizmów już na etapie projektowania architektury systemu [8]. Obejmuje to m.in. segmentację sieci, uwierzytelnianie i autoryzację dostępu, szyfrowanie komunikacji, monitorowanie i rejestrowanie zdarzeń, a także separację funkcjonalną pomiędzy warstwami operacyjnej, decyzyjnej i bezpieczeństwa.

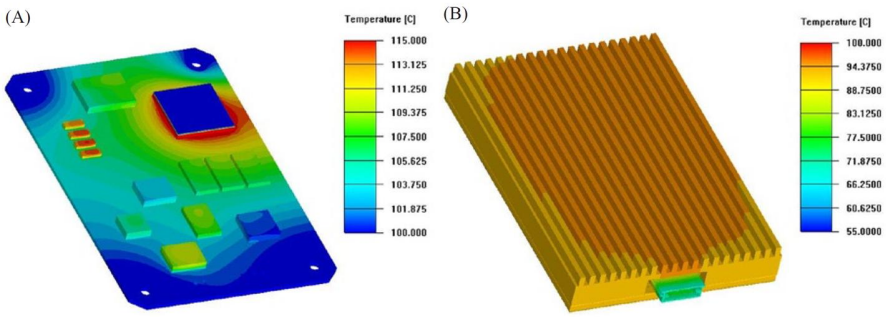
Masa urządzenia

Masa docelowego urządzenia jest istotnym parametrem, szczególnie w systemach mobilnych zasilanych bateryjnie, gdzie wpływa bezpośrednio na zużycie energii oraz czas pracy. Choć parametr ten nie jest bezpośrednio związany z warstwą programową, lecz przede wszystkim ze sprzętem i konstrukcją mechaniczną, to jednak decyzje projektowe dotyczące doboru komponentów, materiałów oraz obudowy mają kluczowy wpływ na całkowitą masę urządzenia. W konsekwencji oddziałują one również na efektywność energetyczną, dynamikę ruchu oraz koszty eksploatacji systemu.

Znaczenie tego aspektu jest szczególnie widoczne w przemyśle motoryzacyjnym, gdzie masa pojazdu bezpośrednio przekłada się na jego osiągi, a w przypadku pojazdów elektrycznych – na zasięg, stanowiący jedno z podstawowych kryteriów oceny przez użytkownika końcowego. Analogiczne zależności występują w systemach robotycznych gdzie większa masa oznacza wyższe zapotrzebowanie na energię, większe zużycie komponentów napędowych oraz potencjalne ograniczenia w zakresie bezpieczeństwa i dynamiki pracy. W rezultacie optymalizacja masy stanowi istotny element procesu projektowego, wymagający kompromisu pomiędzy wytrzymałością konstrukcji, bezpieczeństwem, kosztem a efektywnością energetyczną systemu.

Termiczna stabilizacja układu

W praktyce inżynierskiej aspekt ten bywa często pomijany w procesie tworzenia oprogramowania, jako niezwiązany bezpośrednio z jego implementacją. Takie uproszczenie jest jednak nieuzasadnione, gdyż współczesne systemy automatyki i przetwarzania danych generują znaczne ilości energii cieplnej, głównie w jednostkach obliczeniowych realizujących intensywne przetwarzanie informacji. To właśnie te komponenty elektroniczne ulegają największemu nagrzewaniu podczas normalnej pracy systemu (zob. rys. 3). Zapewnienie odpowiedniego chłodzenia jest kluczowe dla uniknięcia przegrzewania i awarii jednostek procesorowych [9]. O ile odprowadzanie ciepła w warunkach laboratoryjnych lub serwerowych jest stosunkowo dobrze rozwiązanym problemem, o tyle w rzeczywistych warunkach pracy urządzeń przemysłowych staje się ono istotnym wyzwaniem. Systemy te często pracują w środowisku o podwyższonej temperaturze, ograniczonej wentylacji oraz utrudnionych możliwościach efektywnego odprowadzania ciepła. W konsekwencji układ, który osiąga zbyt wysoką temperaturę, może ulec automatycznemu resetowi, ograniczeniu wydajności lub przejściu w tryb awaryjny, co bezpośrednio wpływa na niezawodność i ciągłość działania całego systemu.



Rysunek 3. Wyniki symulacji termicznej pracy elektronicznego układu sterowania.

Warunki pracy systemu

Warunki środowiska docelowego, w których ma pracować urządzenie sterujące, w istotny sposób wpływają na projekt elektroniki oraz konstrukcję mechaniczną, w tym na dobór materiałów i architekturę całego systemu. W zależności od zastosowania, warunki eksploatacyjne mogą znacząco odbiegać od warunków laboratoryjnych. Obejmują one m.in. szeroki zakres temperatur i wilgotności, silne zapylenie, niestabilne zasilanie, przepięcia, wyładowania elektrostatyczne oraz oddziaływanie silnych pól elektromagnetycznych. Dodatkowo systemy te często muszą zapewniać nieprzerwaną pracę przez długi okres czasu, bez konieczności restartów oraz bez zawieszania się oprogramowania, co stawia wysokie wymagania zarówno wobec warstwy sprzętowej, jak i programowej. W praktyce projektowanie urządzeń automatyki przemysłowej przeznaczonych do pracy w takich warunkach wymaga kompleksowego podejścia, znacząco odmiennego od metod stosowanych przy projektowaniu urządzeń laboratoryjnych lub prototypowych.

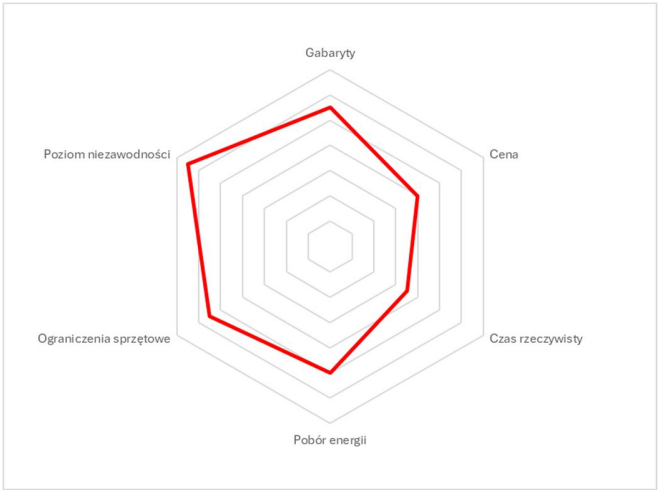
Koszt

Spośród wszystkich wcześniej wymienionych czynników kluczowym, który bardzo często z perspektywy biznesowej ma charakter dominujący, jest cena finalna produktu. Wielu producentów oferuje rozwiązania o zbliżonej funkcjonalności, jednak o sukcesie rynkowym i ekonomicznym – szczególnie w przypadku produkcji seryjnej – decyduje w dużej mierze koszt jednostkowy oraz wynikająca z niego cena końcowa. W praktyce oznacza to konieczność ciągłej optymalizacji nie tylko samej funkcjonalności i parametrów technicznych, ale również struktury kosztów, obejmującej zarówno komponenty sprzętowe, jak i złożoność oprogramowania, proces wytwarzania oraz utrzymania systemu. W efekcie konkurencyjność rozwiązania na rynku jest bezpośrednio powiązana z jego efektywnością kosztową, a nie wyłącznie z poziomem zaawansowania technologicznego.

Podsumowanie

Analiza przedstawionej charakterystyki współczesnych systemów automatyki, w których oprogramowanie stanowi główny czynnik kształtujący funkcjonalność, prowadzi do wniosku, że proces ich projektowania ma charakter wielokryterialnej optymalizacji (zob. rys. 4). W tak sformułowanym problemie należy uwzględnić szereg, często konkurencyjnych, kryteriów, takich jak efektywność energetyczna, masa urządzenia, niezawodność, poziom bezpieczeństwa funkcjonalnego, odporność cybernetyczna oraz zdolność systemu do efektywnego odprowadzania ciepła. Dodatkowo projektowanie ograniczają czynniki techniczne i środowiskowe, w tym dostępność zasobów obliczeniowych i pamięciowych platformy sprzętowej, ograniczenia gabarytowe, warunki pracy (temperatura, wilgotność, zakłócenia elektromagnetyczne), a także wymagania dotyczące pracy w czasie rzeczywistym. W praktyce oznacza to konieczność podejmowania świadomych kompromisów projektowych, ponieważ poprawa jednego z parametrów często prowadzi do pogorszenia innego.

Istotną rolę odgrywają również uwarunkowania ekonomiczne i regulacyjne, które wpływają na końcowy kształt rozwiązania oraz jego możliwość wdrożenia przemysłowego [10], [11]. W rezultacie projektowanie nowoczesnych układów automatyki nie sprowadza się do wyboru najlepszego rozwiązania z punktu widzenia pojedynczego kryterium, lecz stanowi proces poszukiwania rozwiązania kompromisowego, spełniającego jednocześnie wymagania techniczne, środowiskowe, bezpieczeństwa oraz efektywności kosztowej.



Rysunek 4. Ilustracja zagadnienia projektowania układów automatyki jako zagadnienie optymalizacji wielokryterialnej.

Referencje

[1] Cañas, H., Mula, J., Díaz-Madroñero, M., Campuzano-Bolarín, F.: “Implementing Industry 4.0 principles,”, Compters & Industrial Engineering, vol. 158, no. 8, 107379, 2021, doi: https://doi.org/10.1016/j.cie.2021.107379.

[2] Oprzędkiewicz, K., Pawłuszewicz, E., Bartoszewicz, A., Byrski, W., Jaroszewski, K., Pawłuszewicz, G., Skruch, P.: „Dydaktyka automatyki i robotyki, ” Wydawnictwo AGH, Kraków, Polska, 2025, doi: https://doi.org/10.7494/978-83-68219-79-1.

[3] Sze, V., Chen, Y-H., Yang, T-J., Emer, J. : ”Efficient processing of deep neural networks: a tutorial and survey,” arxiv.org, 2017, doi: https://doi.org/10.48550/arXiv.1703.09039.

[4] Myers, G.J.: “The art of software testing,” John Wiley & Sons, New York, NY, USA1979.

[5] Skruch, P., Szelest, M., Długosz, M., Cieślar, D.: “Safety of perception systems in vehicles of high-level motion automation,” Proc. of the 20th Anniversary of IEEE International Conference on Emerging eLearning Technologies and Applications, 20-21.10.2022, Grand Hotel Stary Smokovec, High Tatras, Slovakia, doi: 0.1109/ICETA57911.2022.9974838.

[6] International Organization for Standardization: ISO 21448:2022 Road vehicles – Safety of the intended functionality. Edition 1, 2022.

[7] Alqudhaibi, A., Albarrak, M., Jagtap, S., Williams, N., Salonitis, K.: “Securing industry 4.0.: assessing cybersecurity challenges and proposing strategies for manufacturing management,” Cyber Security and Applications, vol. 3, no. 12, 100067, 2025, doi: https://doi.org/10.1016/j.csa.2024.100067.

[8] Kohler, C.: “The EU cybersecurity act and european standards: an introduction to the role of European standardization,” International Cybersecurity Law Review, vol. 1, no. 1–2, pp. 7–12, 2020, doi: 10.1365/s43439-020-00008-1.

[9] Korta, J., Skruch, P., Hołoń, K.: “Reliability of automotive multi-domain controllers: advancements in electronics cooling technologies,” IEEE Vehicular Technology Magazine, vol. 16, no. 2, pp. 86-94, 2021, doi: 10.1109/MVT.2020.3038588.

[10] European Parliament and Council of the European Union: “Regulation (EU) 2024/1689 (Artificial Intelligence Act),”Official Journal of the European Union, 2024.

[11] European Parliament and Council of the European Union: ”Regulation (EU) 2023/1230,” Official Journal of the European Union, 2023.