

## Projektowanie i ocena systemów autonomicznych

\* Opisz pięć pytań projektowych, które wpływają na stopień zaawansowania algorytmicznego wymaganego dla określonej zdolności autonomicznej (Jakie funkcje musi zapewniać robot? Jakiego horyzontu planowania wymagają te funkcje? Jak szybko algorytmy muszą aktualizować dane wyjściowe? Jakiego rodzaju model świata potrzebny robotowi i dokąd wchodzi człowiek?)

\* Podaj przykłady każdego z trzech źródeł awarii robota: awarie zewnętrzne, awarie fizyczne i błąd ludzki.

\* Mając cel oceny robota, wybierz i uzasadnij odpowiedni rodzaj eksperymentu: eksperyment kontrolowany, ćwiczenia, eksperyment z koncepcją i uczestnik-obszernik.

\* Wymień sześć kategorii danych do zebrania podczas eksperymentów (dziennik aktywności, kontekst, widok robota, stan robota, widok robota z zewnątrz, widok interakcji człowiek-robot) i omów strategie pozyskiwania danych.

## Przegląd

Zaczęliśmy od Części dotyczących autonomii, automatyzacji i architektur, aby zapewnić ramy do poznawania elementów sztucznej inteligencji wykorzystywanej w robotyce. Tu powracamy do pojęć wprowadzonych w tych Częściach, aby powrócić do pytania: Jak zaprojektować inteligentnego robota? Opowiadamy się za ekologicznym podejściem do projektowania, w którym robot i jego inteligencja wpisują się w ekologiczną niszę systemu. Przypomnijmy, że podejście ekologiczne uwzględnia wpływ robota, środowiska i zadania na udany projekt. Projektant musi zastanowić się, co robot może zrobić: Co może wyczuć bezpośrednio lub pośrednio, na przykład zasięg? Jakie są jego zdolności lokomocyjne i manipulacyjne? Jaka moc obliczeniowa jest potrzebna? Co może zrobić, czego człowiek nie może lub co musi zrobić, ponieważ człowiek nie może? Projektant musi również wziąć pod uwagę środowisko: czy można je zamodelować jako zamknięty świat lub zaprojektować tak, aby wspierać automatyzację? Na co pozwala środowisko? Czy te afordancje są wystarczające, czy też wymagane jest uznanie? Jaka jest skala i przejeźdźność każdego regionu środowiska? Wreszcie, samo zadanie wpływa na projekt: Czy nawigacja po zadaniach czy coś innego? Czy system wielorobotowy może lepiej służyć celom? Jakie są ukryte cele, takie jak monitorowanie zaufania i zdrowia? Czy cele wymagają namysłu? Ekologiczne podejście można zastosować do projektowania konkretnych możliwości, a także do całego projektu systemu robota. Rozważ uczenie, gdzie stworzenie skutecznego algorytmu uczenia się wymaga zrozumienia, co robot próbuje zrobić, a także tego, czego robot może wyczuwać i z czego się uczyć; przywołuje to ekologiczne podejście do projektowania, które uwzględnia zadanie, robota i środowisko. Przypomnijmy również, że w Części 4 opisano, że architektura hybrydowa jest czymś więcej niż tylko stylem programowania; pomaga projektantowi określić, ile sztucznej inteligencji jest potrzebne, zadając pięć kluczowych pytań projektowych. Są: Jakie funkcje musi zapewniać robot? Jakiego horyzontu planowania wymagają funkcje? Jak szybko algorytmy muszą aktualizować dane wyjściowe? Jakiego modelu świata potrzebuje robot? i Gdzie wchodzi człowiek? Część kończy się próbą zestawienia poprzednich Częściach w praktycznym kontekście projektowym. Rozpoczyna się od nakreślenia, jak zaprojektować konkretną autonomiczną zdolność, począwszy od ogólnej filozofii projektowania, przeglądu pięciu pytań, które projektant musi zadać z perspektywy architektonicznej, oraz przedstawienia studium przypadku pomyślnego wykorzystania tego procesu projektowania. Następnie przypomina czytelnikom, że różne taksonomie i metryki wprowadzone w książce, a zwłaszcza w części 3, nie zastępują dobrego projektu systemu inteligencji. Część kończy się sekcją dotyczącą całościowej oceny projektów systemów, takiej jak identyfikacja awarii, typy eksperymentów do przeprowadzenia dla określonego celu oceny oraz zalecane dane do zebrania.

## **Projektowanie określonych zdolności autonomicznych**

Kanoniczna architektura operacyjna przedstawiona w Części 4 jest czymś więcej niż ramą, może być również wykorzystana jako pomoc w projektowaniu autonomicznej zdolności, ponieważ przedstawia pięć pytań, na które projektant musi odpowiedzieć. Te pytania kierują projektanta w kierunku spełnienia wyraźnych, a zwłaszcza ukrytych oczekiwań związanych z umiejętnościami. Projektowanie zdolności autonomicznych może być mylące, jeśli projektant traktuje warstwy w architekturze operacyjnej jako mapę drogową. Mierniki, takie jak Poziomy autonomii systemów bezzałogowych (ALFUS),<sup>95</sup> które próbują porównywać zdolności autonomiczne, mogą skłaniać projektantów do myślenia, że bardziej złożona sytuacja wymaga bardziej wyrafinowanej inteligencji, ale jak pokazano w części 6, zwierzęta o niższej inteligencji są wysoce skuteczne w niezwykle złożonych środowiskach. Ostatecznie strategia powinna określać niezbędne możliwości, a następnie badać, w jaki sposób można je osiągnąć, biorąc pod uwagę ograniczenia sprzętowe, programowe, środowiskowe i ludzkie.

## **Filozofia projektowania**

Filozofia projektowania zaczyna się od założenia, że autonomiczna zdolność obejmuje wszystkie funkcje i warstwy architektury kanonicznej. Niektóre z tych funkcji mogą być obsługiwane przez człowieka, często z powodu ograniczeń uniemożliwiających komputerowi wykonywanie tych funkcji. Zapominanie o wyjaśnieniu, w jaki sposób te funkcje są obsługiwane, może spowodować nieoczekiwane zapotrzebowanie na siłę roboczą, co widać po zaskakująco wysokich kosztach siły roboczej, jakie napotyka armia amerykańska przy rozmieszczaniu bezzałogowych systemów latających. Z drugiej strony projektant może uznać, że zdolność autonomiczna nie wymaga namysłu i że kruchość „efektu „fly-at-the-window” jest akceptowalna. Podobnie projekt autonomicznego pojazdu, który jest używany do oportunistycznego eksplorowania zagrazonego środowiska, może kłaść nacisk na dodanie autonomicznej deliberacji zamiast nawigacji, gdzie robot zapewnia mapowanie i monitorowanie stanu zdrowia, podczas gdy użytkownik decyduje, gdzie się udać i co jest interesujące do dalszego zbadania. Chodzi o to, aby projektant świadomie decydował, jakie autonomiczne zdolności są potrzebne do realizacji misji i jak te zdolności będą wdrażane z perspektywy systemowej. Misja prawdopodobnie będzie wymagała wielu autonomicznych zdolności. Na przykład nawigacja po punktach orientacyjnych może obejmować cztery możliwości: planowanie trasy, tworzenie wystąpień zachowań umożliwiających podążanie drogą i omijanie przeszkód, ustawianie monitora lub licznika czasu, który uruchamia ponowne planowanie, jeśli robot nie dotrze do punktu nawigacyjnego w określonym czasie oraz udostępnianie interfejsu użytkownika który wyświetla postęp i przedstawia wyświetlacz diagnostyczny w przypadku wystąpienia problemu. Drugą autonomiczną funkcją robota może być wyszukiwanie intruzów za pomocą kamery na podczerwień, która wykorzystuje oprogramowanie do ignorowania grup małych sygnałów cieplnych zgodnych ze zwierzętami i ostrzega operatora o obecności człowieka. Trzecią autonomiczną zdolnością tego samego robota może być wykrywanie i usuwanie usterek, które monitoruje stan podsystemów i powoduje zastąpienie równoważnego czujnika lub komponentu. System z wieloma możliwościami autonomicznymi będzie prawdopodobnie miał różne potrzeby w zakresie modeli światowych, monitorowania i interfejsów. Architektura operacyjna jest wystarczająca, aby przypomnieć projektantowi, że te funkcje powinny być brane pod uwagę, ale architektura nie pomaga projektantowi w rozważaniu, jak skoordynować te różne wymagania. Jak zarządzać różnymi wymaganiami projektowymi, to rola systemów i widoków architektury technicznej; opisują, jak zaprojektować funkcję monitorowania lub interfejs użytkownika w celu obsługi różnych możliwości.

## **Pięć pytań dotyczących projektowania autonomicznego robota**

Istnieje pięć ogólnych pytań, które projektant powinien zadać, projektując każdą konkretną autonomiczną zdolność.

Jakie funkcje wewnętrzne robot musi wykonywać, aby uzyskać tę zdolność? Czy trzeba generować plan? Czy będzie monitorować realizację planu? Czy będzie musiał wybierać zasoby lub strategie na podstawie różnych warunków pogodowych lub typów terenu? W jaki sposób te strategie zostaną przekształcone w zachowania oraz włączone i wyłączone (wdrożone)? Jak wykona zachowania? Czy podczas misji będzie jakaś nauka online? Czy po misji będzie nauka, od operatora lub offline?

Jakiego horyzontu planowania (teraźniejszość; teraźniejszość i przeszłość; teraźniejszość, przeszłość, przyszłość) wymaga zdolność? Horyzont planowania pomaga kształtować oczekiwania. Zdolności reaktywne, takie jak odruchowe omijanie przeszkód, działają w teraźniejszości i są bardzo szybkie, ale podobnie jak odruch kolanowy ludzkiej rzepki, mogą nie być optymalne. Optymalne algorytmy na ogół wymagają zachowania stanu przeszłości i często prognozowania wyników w przyszłość, co może wiązać się z konstruowaniem modeli. Każdy horyzont planowania jest domyślnym zobowiązaniem do wewnętrznych modeli i struktur.

Jak szybko muszą działać algorytmy umożliwiające tę zdolność? Z pewnością wiele algorytmów sterowania lotem UAV musi działać z bardzo wysokimi prędkościami, chociaż pamiętaj, że zwykle nie są one uwzględnione w sformułowaniu „sztucznej inteligencji” autonomicznego robota. Ale jeśli robot musi omijać poruszającą się przeszkodę lub pocisk, może to wymagać bardzo szybkiego wykonania. Ta szybkość wykonania wskazuje, że zdolność może znajdować się wyłącznie w warstwie reaktywnej, ponieważ funkcjonalność deliberatywna jest wolniejsza z powodu dodatkowego czasu obliczeniowego wymaganego do utrzymania modeli stanu i świata. Często pożądana percepcja zdolności trwa dłużej niż jest to pożądane i, jak pokazano w Części 6, dane z czujników są często przetwarzane zarówno w sposób reaktywny, jak i celowy; na przykład robot może wykorzystać dane z czujnika, aby uniknąć kolizji, podczas gdy inny proces jednocześnie wykorzystuje dane z czujnika, aby określić, czy obiekt był kulą osłabienia i czy można bezpiecznie wznowić jego tor.

Jakich modeli potrzebuje konkretny algorytm? Lokalny? Światowy? Obydwa? Powszechnie przyjmuje się, że możliwości autonomiczne wymagają wyrafinowanych, szczegółowych modeli. Pierwsze modele iRobot Roomba odniosły sukces właśnie dlatego, że nie stworzyły modelu świata; nie musieli planować pokoju, aby go posprzątać. Podobnie wiele systemów tworzy modele w oparciu o możliwości czujnika w porównaniu z obszarem zainteresowania; jeśli obwiednia robocza robota jest określona przez promień trzech metrów, po co budować szczegółowy model, który obejmuje promień dziewięciu metrów? Czy reprezentacje 3D są niezbędne dla robota, czy są po prostu atrakcyjne do wizualizacji? Modelowanie znacznie zwiększa złożoność obliczeniową i tworzy wąskie gardła, a ponadto może autonomicznie tworzyć martwe punkty — jeśli obiektu lub warunku nie ma w modelu, oznacza to, że nie istnieje.

Gdzie wchodzi człowiek? Jak widać w Części 18, człowiek zawsze jest w jakiś sposób związany z robotem i należy to uwzględnić w projekcie od samego początku. Czy człowiek tylko zleca robotowi, ale robot jest w pełni odpowiedzialny za tworzenie instancji, monitorowanie i rozwiązywanie wszelkich problemów całej misji? Jest to mało prawdopodobne ze względu na obecny stan sztucznej inteligencji i kształtowanie misji. Dlatego algorytmy są ograniczone, a zrozumienie domeny jest niepełne, że operatorzy muszą być dostępni, aby poradzić sobie z nieuniknionymi awariami autonomii i niespodziankami. Czy misje naprawdę są całkowicie oddelegowane do robota, to jest agentów wykonujących zadania, czy też są to misje polegające na zdalnej obecności lub pomocy ludzkiemu operatorowi? Jeśli człowiek jest zaangażowany w jedną lub więcej zdolności, który agent - robot czy człowiek — wykonuje jakie funkcje; na przykład, czy człowiek monitoruje postępy robota? Czy któreś z

funkcji są wspólne dla człowieka i robota, a jeśli tak, to w jaki sposób? Na przykład robot może zalecić ścieżkę lub alokację zasobów, ale ostateczną decyzję podejmuje operator.

### **Studium przypadku: Konkurs bezzałogowej robotyki naziemnej**

#### **PROCES PROJEKTOWANIA EKOLOGICZNEGO**

Zgłoszenie Colorado School of Mines (CSM) w konkursie Unmanned Ground Robotics Competition w 1994 roku stanowi przykład procesu projektowania. Mimo że wydarzenie ma już dziesiątki lat, prostota pomaga zobaczyć, jak zasady projektowania łączą się z wynikami. Celem konkursu było, aby mały pojazd bezzałogowy (nie większy niż wózek golfowy) poruszał się autonomicznie po odkrytym torze z białych linii namalowanych na trawie. Wpis do CSM zdobył pierwsze miejsce i nagrodę w wysokości 5000 \$. Zespół zastosował ekologiczny proces projektowania składający się z siedmiu kroków. Każdy etap projektowania jest najpierw przedstawiany pogrubioną czcionką i omawiany. To, co faktycznie zostało zrobione przez zespół CSM, zaznaczono kursywą. To studium przypadku ilustruje efektywne wykorzystanie niezwykle niewielu zachowań oraz wykorzystanie afordancji w połączeniu ze zrozumieniem niszy ekologicznej. Podkreśla również, że nawet prosty projekt może wymagać wielu iteracji, aby był wykonalny.

**Krok 1: Opisz zadanie.** Celem tego kroku jest określenie, co robot musi zrobić, aby odnieść sukces. Zadanie polegało na tym, aby pojazd robota podążał ścieżką zawierającą zakręty, nieruchome przeszkody na ścieżce i piaskownicę. Robot, który posunąłby się najdalej bez całkowitego wykroczenia poza granice, byłby zwycięzcą, chyba że dwa lub więcej robotów przebyło tę samą odległość lub ukończyło trasę, wówczas zwycięzcą byłby robot z najszybszym czasem. Maksymalna prędkość wynosiła pięć mil na godzinę. Jeśli robot częściowo wyszedł poza granice (ale jedno koło lub część bieżnika pozostało w środku), odjęto karę za odległość. Jeśli robot uderzył w przeszkodę wystarczająco mocno, aby ją przesunąć, naliczana jest kolejna kara dystansowa. Dlatego konkurencja faworyzowała wjazd, który mógłby ukończyć trasę bez naliczania kar, w porównaniu z szybszym wjazdem, który mógłby przekroczyć linię graniczną lub uderzyć w przeszkodę. Uczestnicy otrzymali trzy przejazdy w ciągu jednego dnia i dwa dni, aby przygotować się i przetestować na torze w pobliżu toru; czasy rozgrywek wyznaczała loteria.

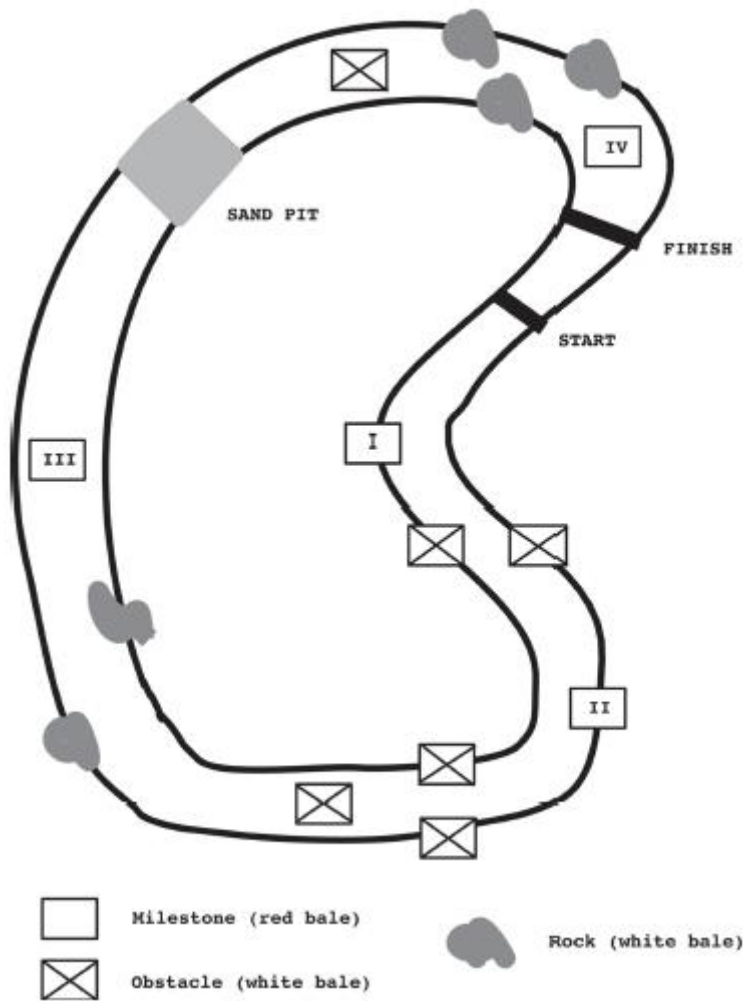
**Krok 2: Opisz robota.** Celem tego kroku jest określenie podstawowych zdolności fizycznych robota oraz wszelkich ograniczeń. Teoretycznie można by się spodziewać, że projektant miałby kontrolę nad projektem samego robota, co mógłby zrobić, jakie czujniki nosi i tak dalej. W praktyce większość robotyków pracuje albo z dostępną na rynku platformą badawczą, która może mieć ograniczenia co do tego, jaki sprzęt i czujniki można dodać, albo ze stosunkowo niedrogim zestawem typu platformy, w którym ograniczenia masy i mocy mogą mieć wpływ na to, co może rozsądnie zrobić. Dlatego projektantowi zwykle podaje się pewne stałe ograniczenia na platformie robota, które będą miały wpływ na projekt. W tym przypadku konkurencja stwierdziła, że pojazd robota musiał mieć ślad co najmniej 3 stopy na 3,5 stopy, ale nie większy niż wózek golfowy. Co więcej, robot musiał mieć własne źródło zasilania i wykonywać wszystkie obliczenia na pokładzie (nie była dozwolona komunikacja radiowa z zewnętrznym procesorem) oraz mieć 20-funtowy ładunek.

Rysunek pokazuje wpis CSM, Omnibot. Materiały przekazała firma Omnitech Robotics, Inc. Bazą pojazdu był dziecięcy jeep Fisher Price Power Wheels na baterie zakupiony w sklepie z zabawkami. Podstawa dokładnie spełniała minimalną powierzchnię. Zastosowano w nim układ kierowniczy Ackerman (podobny do samochodu), z silnikiem napędowym napędzającym koła z tyłu i silnikiem sterującym z przodu. Pojazd miał kąt skrętu 22°. Komputer pokładowy był obsługiwany przez komputer PC 33 MHz 486 przy użyciu sterowników silników Omnitech CANAMP. Zestaw czujników składał się z trzech urządzeń: enkoderów wału na silnikach napędowych i sterujących do martwego naliczenia,

kamery wideo zamontowanej na maszcie w pobliżu środka pojazdu oraz sonaru panoramującego zamontowanego pod kratą z przodu. Wyjście z kamery wideo zostało zdigitalizowane przez czarno-biały framegrabber. Sonar był przetwornikiem ultradźwiękowym Polaroid klasy laboratoryjnej. Urządzenie do panoramowania może przesuwać się o 180 stopni. Całe kodowanie zostało wykonane w C++.



Dzięki silnikom i przekładniom Omnibot mógł poruszać się z prędkością zaledwie 1,5 mil na godzinę. To ograniczenie oznaczało, że mógł wygrać tylko wtedy, gdyby posunął się dalej z mniejszą liczbą punktów karnych niż jakikolwiek inny wpis. Oznaczało to również, że system sterowania musiał mieć częstotliwość aktualizacji co najmniej 150 ms, w przeciwnym razie robot mógł wyjechać poza granice, nie zauważając, że zbacza z kursu. Czarno-biały framegrabber wyeliminował użycie koloru. Co gorsza, częstotliwość aktualizacji framegrabbera wynosiła prawie 150 ms; każdy algorytm przetwarzania obrazu musiałby aktualizować się bardzo szybko, w przeciwnym razie robot poruszałby się szybciej, niż mógłby wyczuć i zareagować. Odbicia od nierównej trawy zmniejszyły standardowy zasięg sonaru z 25,5 stóp do około 10 stóp. Krok 3: Opisz środowisko. Ten krok jest krytyczny z dwóch powodów. Po pierwsze, jest to kluczowy czynnik określający umiejscowienie robota. Po drugie, identyfikuje percepcyjne możliwości zachowań: w jaki sposób zdarzenie percepcyjne stworzy instancję nowego zachowania i jak będzie funkcjonować schemat percepcyjny danego zachowania. Przypomnij sobie z Części 4, że paradygmat reaktywności faworyzuje percepcję bezpośrednią lub percepcję opartą na afordancji, ponieważ ma szybki czas wykonania i nie wymaga rozumowania ani pamięci. Pole zostało wytyczone na trawiastym polu o łagodnych zboczach. Trasa składała się z pasa o szerokości 3 stóp, oznaczonego białą farbą Departamentu Transportu Stanów Zjednoczonych i miało mniej więcej kształt nerki.



Dokładna długość trasy i rozmieszczenie przeszkód na trasie nie były znane aż do dnia zawodów, a drużynom nie wolno było mierzyć trasy ani przeprowadzać na niej prób. Wszystkie przeszkody były nieruchome i składały się z bel siana owiniętych w biały lub czerwony plastik. Bele miały około 2 stopy na 4 stopy i nigdy nie sięgały dalej niż 3 stopy w głąb pasa. Sonar był w stanie niezawodnie wykryć bele pokryte plastikiem pod większością kątów zbliżenia z odległości 8 stóp. Pojazdy miały kursować 22 maja między 9:00 a 17:00, niezależnie od pogody czy zachmurzenia. Oprócz wizualnych wyzwań związanych ze zmianą oświetlenia z powodu chmur, bele wprowadziły cienie na białych liniach w godzinach od 9:00 do 11:00 i od 15:00 do 17:00. Piaskownica miała zaledwie 4 stopy długości i była umieszczona na prostym odcinku toru. Analiza otoczenia pozwoliła na uproszczenie zadania. Umieszczenie przeszkód pozostawiło otwartą przestrzeń o szerokości 4 stóp. Ponieważ Omnibot miał tylko 3 stopy szerokości, tor mógł być traktowany jako pozbawiony przeszkód, gdyby robot mógł pozostać na środku pasa z tolerancją 0,5 stopy. Ta zależność między wielkością robota a skalą otoczenia wyeliminowała potrzebę unikania zachowania przeszkód. Analiza otoczenia pozwoliła również zidentyfikować afordancję sterowania robotem. Jedynym obiektem zainteresowania robota była biała linia, która powinna ukazywać wysoki kontrast z zieloną (ciemnoszarą) trawą w aparacie. Ale dokładna wartość oświetlenia białej linii zmieniała się wraz z pogodą. Jednak po dalszych rozważaniach zespół zdał sobie sprawę, że gdyby kamera była skierowana bezpośrednio na jedną linię, zamiast próbować zobaczyć obie linie, większość najjaśniejszych punktów na obrazie należałaby do tej linii. Jest to zmniejszenie stosunku sygnału do szumu, ponieważ większa część obrazu zawiera linię. Niektóre jasne punkty byłyby spowodowane odbiciami, ale zakładano, że są one rozmieszczone losowo. Dlatego też,

gdyby robot próbował utrzymać środek ciężkości białych punktów na środku obrazu, pozostałby na środku pasa.

Krok 4: Opisz, jak robot powinien zachowywać się w odpowiedzi na otoczenie. Celem tego kroku jest zidentyfikowanie zestawu jednego lub więcej kandydujących zachowań pierwotnych; kandydaci ci zostaną później udoskonaleni lub wyeliminowani. Gdy projektant opisuje, jak robot powinien się zachowywać, zachowania zwykle stają się widoczne. Należy podkreślić, że celem tego kroku jest skoncentrowanie się na tym, co robot powinien zrobić, a nie jak to robi, chociaż często projektant widzi jednocześnie co i jak. W przypadku wpisu CSM początkowo zaproponowano tylko jedno zachowanie: follow-line. Schemat percepcyjny wykorzystałby białą linię do obliczenia różnicy między położeniem środka ciężkości białej linii a tym, gdzie powinien być, podczas gdy schemat silnika przekształca tę różnicę w polecenie dla silnika sterującego.

## TABELA ZACHOWANIA

Jeśli chodzi o wyrażanie zachowań dla zadania, często korzystne jest skonstruowanie tabeli zachowań jako jednego ze sposobów na zebranie wszystkich zachowań na jednej kartce papieru. Zwolnienie dla każdego zachowania jest pomocne w potwierdzeniu, że zachowania będą działać poprawnie, bez konfliktów (pamiętaj, że przypadkowe zaprogramowanie robota odpowiadającego męskim ciernikom z Części 7 jest niepożądane). Często projektantowi przydaje się sklasyfikowanie schematu motorycznego i percepcji. Zastanówmy się na przykład, co się stanie, jeśli implementacja ma czysto refleksyjny schemat motoryczny przejścia do celu i zachowanie polegające na unikaniu przeszkód. Co się stanie, jeśli zachowanie polegające na unikaniu przeszkód spowoduje, że robot straci percepcję celu? Ups. Schemat percepcyjny nie zwraca żadnego celu, a zachowanie polegające na przejściu do celu zostaje zakończone! Prawdopodobnie projektant założył, że zachowanie to będzie wzorcem o ustalonym działaniu, a tym samym będzie trwało w kierunku ostatniego znanego położenia celu. Jak widać z Tabeli Zachowań w tabeli , zespół CSM początkowo zaproponował tylko jedno zachowanie, follow-line.

Behavior Table

Releaser	Behavior	Motor Schema	Percept	Perceptual Schema
always on	follow-line()	stay-on-path(c_x)	c_x	compute-centroid(image,white)

follow-line składał się ze schematu motorycznego, pozostania na ścieżce (centroid), który był refleksyjny (bodziec-odpowiedź) i taksówek (ukierunkowywał robota względem bodźca). Schemat percepcyjny, computecentroid (obraz, biel), wyodrębnił z obrazu afordancję centroidu bieli jako linii. Użyto tylko składowej x lub położenia poziomego środka ciężkości, c\_x.

Krok 5: Popraw każde zachowanie. W tym momencie projektant ma ogólną koncepcję organizacji systemu reaktywnego i jego pożądanych działań i może teraz skoncentrować się na projektowaniu każdego indywidualnego zachowania. Podczas konstruowania podstawowych algorytmów dla schematów motorycznych i percepcyjnych ważne jest, aby projektant wziął pod uwagę normalny zakres warunków środowiskowych, w których oczekuje się, że robot będzie działał (np. stan stacjonarny) oraz warunki, w których zachowanie będzie ponieść porażkę. Zachowanie podążania za linią oparto na analizie, że jedynymi białymi obiektami w środowisku były białe linie i bele siana pokryte białym, zgodnie z opisem w regulaminie. Chociaż było to dobre założenie, doprowadziło to do humorystycznego wydarzenia podczas drugiego biegu zawodów. Gdy robot podążał wzdłuż białej linii, jeden z sędziów wszedł w pole widzenia kamery. Niestety sędzia miał na sobie białe buty, a Omnibot

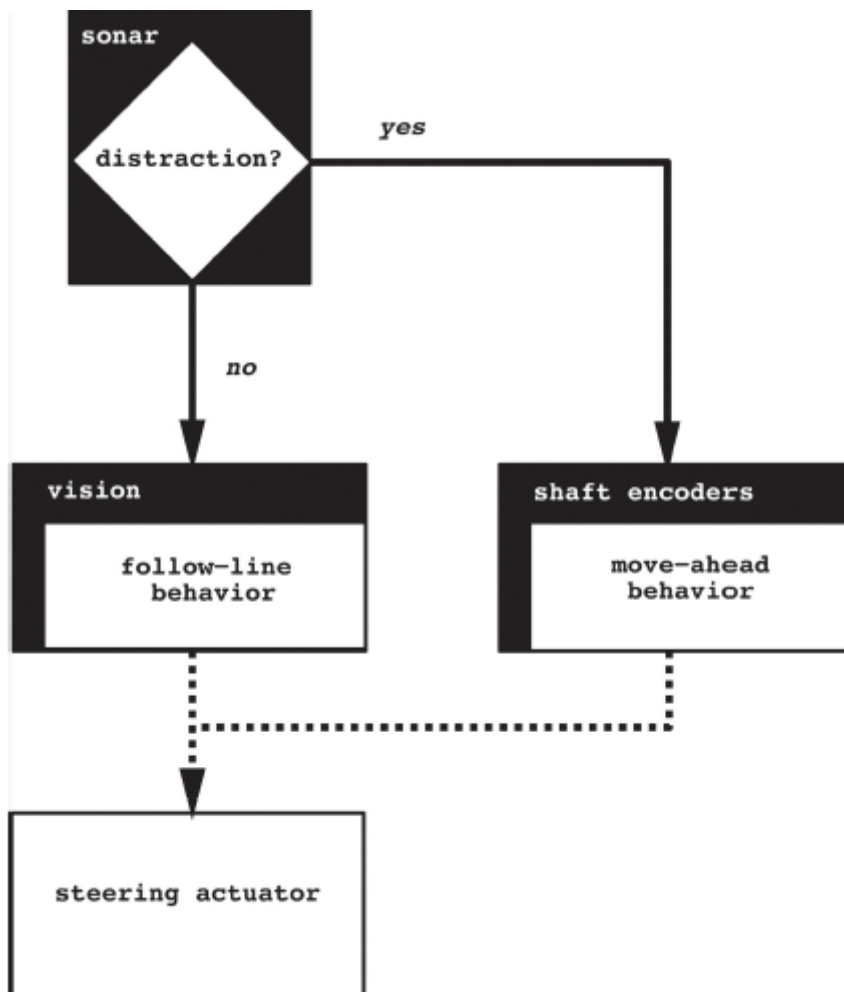
odwrócił się mniej więcej pomiędzy butami a linią. Kapitan drużyny CSM, Floyd Henning, zorientował się, co się dzieje i krzyknął do sędziego, aby się ruszył. Ostrzeżenie nadeszło za późno; przednie koła robota przekroczyły już linię. Jego kamera była teraz skierowana poza linię i nie było szans na wyzdrowienie. Nagle, tuż przed tym, jak skrajne lewe tylne koło miało opuścić granicę, Omnibot wyprostował się i zaczął jechać równolegle do linii! Ścieżka skręciła w prawo, a Omnibot wszedł z powrotem na ścieżkę i ponownie pozyskał linię. Robot w końcu wyszedł poza granice na szpilce do włosów w dalszej części kursu. Tłum oszalał, a ekipa CSM wymieniła zdezorientowane spojrzenia. Co się stało, że Omnibot wrócił do normy? Schemat percepcyjny używał 20% najjaśniejszych pikseli obrazu do obliczenia centroidu. Kiedy Omnibot wędrował po trawie, szedł prosto, ponieważ odbicie na trawie było w dużej mierze przypadkowe, a pozycje znikwały, pozostawiając środek ciężkości zawsze na środku obrazu. Dozorcy kosili trawę tylko w miejscach, gdzie była ścieżka. Obok ścieżki znajdowała się równoległa próbka nieskoszonej trawy naładowana chwastami z nasion mniszka lekarskiego. Rząd białych puchów działał jak biała linia, a gdy mleczce znalazły się w zasięgu widzenia, Omnibot posłusznie skorygował swój kurs, aby był równoległy do nich. Tylko szczęściem było, że ścieżka zakrzywiła się tak, że kiedy mleczce się skończyły, Omnibot szedł prosto i przecinał się ze ścieżką. Chociaż Omnibot nie został zaprogramowany do reagowania na buty i mleczce, zareagował poprawnie, biorąc pod uwagę swoją niszę ekologiczną.

Krok 6: Przetestuj każde zachowanie niezależnie. Podobnie jak w przypadku każdego projektu inżynierii oprogramowania, moduły lub zachowania są testowane indywidualnie. Najlepiej byłoby, gdyby testowanie odbywało się w symulacji przed testowaniem modułów robota działającego w jego środowisku. Wiele dostępnych na rynku robotów ma imponujące symulatory lub symulatory wbudowane w pakiet Robot Operation System (ROS). Należy jednak pamiętać, że symulatory często modelują jedynie mechanikę robota, a nie zdolności percepcyjne. Symulacja jest przydatna do potwierdzenia, że kod schematu silnika jest poprawny, ale często jedynym sposobem weryfikacji schematu percepcyjnego jest wypróbowanie go w świecie rzeczywistym.

Krok 7: Przetestuj z innymi zachowaniami. Ostatnim etapem projektowania i wdrażania systemu reaktywnego jest wykonanie testów integracyjnych; testowanie integracyjne to etap, na którym zachowania są łączone. Testowanie integracyjne obejmuje również testowanie zachowań w rzeczywistym środowisku. Chociaż zachowanie podążania za linią działało dobrze w przypadku testowania z białymi liniami, nie działało dobrze w przypadku testowania z białymi liniami i przeszkodami. Przeszkody, lśniące białe bele siana stojące w pobliżu linii, często były jaśniejsze niż trawa. (Zauważ, że bele owinięte na czerwono były używane tylko do oznaczania obszaru początkowego i nie były widoczne dla robota.) Dlatego schemat percepcyjny dla kolejnych linii uwzględniał piksele należące do beli podczas obliczania centroidu. Niezmiennie robot zafiksował się na beli i podążał za jej obwodem, a nie po linii. Bele były określane jako „rozproszenia wzrokowe”. Na szczęście bele były stosunkowo małe. Gdyby robot mógł „zamknąć oczy” na około dwie sekundy i po prostu jechać w linii prostej, w większości pozostawałby na kursie i byłby w stanie ponownie przejść linię. Nazywało się to zachowaniem ruchu naprzód. Zachowanie wykorzystywało kierunek robota (kąt skrętu, dir) do wytworzenia jednolitego pola potencjalnego, mimo że nie było sumowania wektorów. Problem polegał na tym, że robot wiedziałby, kiedy zignorować sygnał wizji i wykonać ruch do przodu. Rozwiązaniem problemu, kiedy należy zignorować sygnał wejściowy z kamery, było użycie sonaru jako wyzwalacza ruchu naprzód. Echosonda była skierowana na linię i za każdym razem, gdy zwracała odczyt zasięgu, przesuwanie do przodu zajmowało kontrolę na dwie sekundy. Ze względu na trudności w pracy z DOS, wpis CSM musiał używać stałego harmonogramu dla wszystkich procesów. Byłoby łatwiej i bardziej niezawodnie, gdyby każdy proces wykonywał każdy cykl aktualizacji, nawet jeśli wyniki zostały odrzucone. W rezultacie wyzwalacz sonaru dla ruchu do przodu zasadniczo hamował podążanie za linią, podczas gdy brak wyzwalacza sonaru hamował ruch do przodu. Oba zachowania występowały



cały czas, ale tylko jedno miało wpływ na to, co robił robot. Rysunek pokazuje to hamowanie, podczas gdy nowa tabela behawioralna jest pokazana w tabeli.



Releaser	Inhibited by	Behavior	Motor Schema	Percept	Perceptual Schema
always on	near-read_sonar()	follow_line()	stay-on-path(c_x)	c_x	compute_centroid(image,white)
always on	far-read_sonar()	move_ahead(dir)	uniform(dir)	dir	dead_reckon(shaft-encoders)

Ostateczna wersja projektu działała na tyle dobrze, że zespół CSM zajął pierwsze miejsce. Robot okrążył tor, aż znalazł się około 10 jardów od mety i natknął się na płytkę piaskownicę, która miała przetestować przyczepność platformy. Piaskownica budziła pewne obawy, ponieważ piasek ma jasny kolor i może być interpretowany jako część linii. Ponieważ piasek znajdował się na poziomie gruntu, odczyt zasięgu nie mógł być użyty jako inhibitor. Ostatecznie zespół zdecydował, że skoro piaskownica ma tylko połowę długości beli, nie będzie to miało wystarczającego wpływu na robota, aby warto było zmieniać delikatny harmonogram istniejących procesów. Zespół miał rację, twierdząc, że piaskownica okazała się zbyt mała, aby stanowić znaczącą wizualną rozrywkę. Zapomnieli jednak o kwestii trakcji. Aby uzyskać lepszą przyczepność, zespół wsunął prawdziwe opony na śliskie plastikowe koła, ale zapomnieli przymocować opony do kół. W piasku robot zakręcił kołami w oponach. Po upływie limitu czasu zespołowi pozwolono popchnąć robota (zrobiono to sfrustrowanym kopnięciem przez głównego programistę), aby sprawdzić, czy ukończyłby cały kurs. Rzeczywiście tak było. Żaden inny robot nie dotarł aż do piaskownicy. Oczywiście jest, że do tego zastosowania wystarczył system reaktywny. Wykorzystanie prymitywnych zachowań reaktywnych było niezwykle tanie obliczeniowo, dzięki czemu robot mógł aktualizować siłowniki niemal z szybkością aktualizacji framegrabbera wizyjnego. Zespół

CSM opracował robota, który pasuje do jego wąskiej niszy ekologicznej. Zachowania nie działałyby w podobnej domenie, takiej jak podążanie chodnikami, a nawet ścieżka z białych linii ze skrzyżowaniem, ponieważ nisza nie zawierała tych atrybutów. Jednak proces projektowania ekologicznego nie ogranicza się do wąskich nisz. Podobny proces projektowania można by wykorzystać do opracowania robota, który spełniałby wyraźne wymagania dla szerszego zbiegu środowisk i zadań z dodatkową odpornością. Ten robot prawdopodobnie wymagałby rozważnej funkcjonalności.

### **Taksonomie i metryki a projektowanie systemu**

Częstym błędem w projektowaniu jest mylenie taksonomii i metryk do porównywania systemów autonomicznych przedstawionych w Części 3 z wytycznymi projektowymi i mapami drogowymi. W szczególności taksonomia delegacji między człowiekiem a maszyną, stworzona przez Sheridana dla NASA i wykorzystywana jako źródło stylów architektury poziomu autonomii (LOA), jest często traktowana jako mapa drogowa programowania. Skupienie się na taksonomii kieruje projektantów do ewolucyjnego podejścia do mapy drogowej: najpierw zaprogramuj roboty do obsługi wspólnej kontroli, następnie zaprogramuj wsparcie decyzji i tak dalej, a ostatecznie zostaną dodane wystarczające możliwości, aby baza kodu dla w pełni autonomicznych robotów istniała. Filozofia projektowania promowana w tej książce zaczyna się od określenia pożądaných możliwości konkretnego robota, niezależnie od ilości potrzebnej sztucznej inteligencji. Następnie projektant określa, który agent – robot, człowiek czy oprogramowanie – jest odpowiedzialny. Na przykład nisza ekologiczna dla jednego robota może polegać na wspieraniu decyzji w przypadku niektórych zadań, ale ten sam robot może wymagać wspólnej kontroli nad innymi zadaniami. Taksonomie i wizualizacja organizacji oprogramowania jako warstw są często traktowane jako rozwojowe mapy drogowe oparte na idei, że należy najpierw zaprogramować systemy o „najniższych” inteligentnych możliwościach, a następnie dodać programowanie, aby umożliwić robotowi wykonanie kolejnego zestawu funkcji lub możliwości. Z takim podejściem wiążą się dwa problemy. Po pierwsze, podważa wartość projektowania systemów. Na przykład mały dom będzie potrzebował tych samych podstawowych funkcjonalnych przestrzeni, takich jak łazienka, kuchnia, sypialnie i przestrzeń życiowa, co dwór; dom na ogół nie jest domem, jeśli zawiera tylko łazienkę lub tylko kuchnię. Taksonomie i warstwy mają tendencję do zachęcania do podejścia „najpierw buduj domy tylko z sypialniami, następnie buduj domy z sypialniami i łazienkami, a następnie buduj domy z sypialniami, łazienkami i kuchniami” zamiast „zbuduj mały dom z sypialnią, łazienką, i kuchnię i dodaj drugie piętro lub skrzydło później”. Dom musi od samego początku zapewniać pewną funkcjonalność, aby był wartościowy. Po drugie, podejście warstwowe hamuje rozszerzalność. Jeśli dana osoba budowała dom stopniowo przez lata i mieszkała tam, gdy go budowała, pierwsza część domu normalnie miałaby miejsce na kuchnię, łazienkę i jakąś przestrzeń życiową, która może być również wykorzystana jako sypialnia, ale dom zostałby zaprojektowany tak, aby ułatwić przyszłe dobudowy. Na przykład można zbudować większy fundament lub niektóre ściany lub części dachu skonstruować w taki sposób, aby ułatwić dobudowanie do domu. Dodanie do domu, który nie został zaprojektowany z myślą o przyszłych dobudowach, może być drogie i prowadzić do brzydkiego domu z niezręcznym wykorzystaniem przestrzeni. Niedawne dyskusje w społecznościach zajmujących się robotyką doprowadziły do prób stworzenia schematów klasyfikacji, które porównywałyby autonomię jednego systemu robota z innym. Jednym z przykładów takiego schematu klasyfikacji są poziomy autonomii systemów bezzałogowych (ALFUS) Narodowego Instytutu Standardów i Technologii Stanów Zjednoczonych (ALFUS).<sup>95</sup> Mierniki służące do porównywania lub szacowania autonomii systemu mogą wprowadzać w błąd, ponieważ ilość pracy potrzebnej do zbudowania domu nie jest miarą tego, czy dana konstrukcja jest domem. Weź pod uwagę, że budowa budynku z ładną kuchnią, ale bez miejsca do spania, może zająć tyle samo czasu, co dom z sypialnią i minimalistyczną kuchnią, ale piękna kuchnia to nie dom. Budowa domu z XIX wieku prawdopodobnie wymagała większego wysiłku fizycznego niż obecny podmiejski dom, który został zoptymalizowany pod

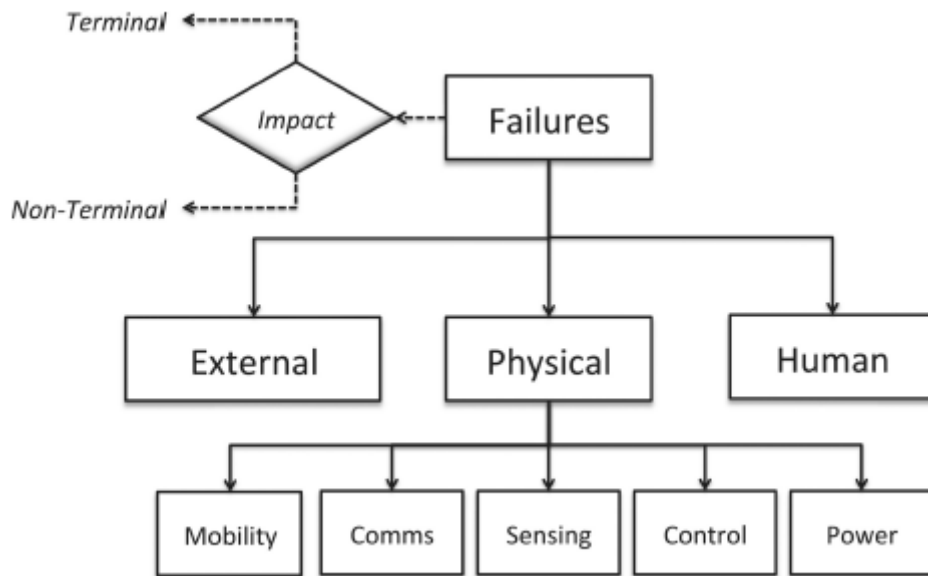
kątem łatwości budowy. Jednak oba są funkcjonalnymi odpowiednikami. System robota powinien mieć określoną autonomiczną zdolność, ponieważ cele systemu wymagają tej zdolności, a nie dlatego, że ta konkretna zdolność była łatwiejsza do zaprogramowania niż inna zdolność. Ekologiczne podejście zachęca projektanta do myślenia poza uproszczoną wizualizacją warstw i wpływem na projektowanie określonych podstawowych funkcji. Dom z trzema sypialniami, zbudowany na zboczu wzgórza w podatnej na trzęsienia ziemi Kalifornii, zostanie zbudowany z innego rodzaju fundamentów i materiałów budowlanych niż dom z trzema sypialniami o podobnej wielkości, położony wzdłuż plaży na Florydzie. Podobnie w robocie na wybór platform, czujników i algorytmów ma wpływ złożoność środowiska. Ale dom w Kalifornii nie jest z natury „lepszy” niż dom na Florydzie. Wracając do przykładu małego domu, podczas gdy zarówno mały dom, jak i rezydencja będą miały te same podstawowe funkcjonalne przestrzenie, rezydencja zaprojektowana z myślą o bardziej złożonym stylu życia rozrywkowych gości może mieć większą kuchnię, więcej sypialni i łazienek oraz dodatkowe funkcje, takie jak basen i kino domowe. Podobnie złożoność misji ma wpływ na projekt robota. Dom z pięcioma sypialniami może mieć małe podwórkę, ponieważ właściciele nie chcą spędzać czasu na ogrodnictwie, podczas gdy domek z jedną sypialnią może mieć znacznie większe podwórkę, ponieważ właściciele chcą spędzać czas na ogrodnictwie. Podobnie na projektowanie robotów ma wpływ pożądana przez człowieka niezależność.

### **Holistyczna ocena inteligentnego robota**

W Części 3 przedstawiono wyzwanie niedeterminizmu w testowaniu autonomii. Kolejne Części dostarczyły dowodów na to, że autonomiczna zdolność może w rzeczywistości być wynikiem współdziałania wielu algorytmów lub struktur danych. Te poszczególne algorytmy można przetestować, ale w pewnym momencie ich zdolność zostanie oceniona w kontekście całościowego systemu robota. Pytanie brzmi: „Czy zwiększona inteligencja zadziałała i ogólnie dodała wartość do systemu?” W tej sekcji nie omówiono konkretnych metod testowania, ponieważ są one w dużym stopniu zależne od algorytmu, sprzętu i aplikacji. Zamiast tego oferuje ogólne ramy do przeprowadzania oceny inteligentnego robota. Najpierw zapewnia taksonomię błędów, aby ostrzec projektanta o typach błędów, których należy szukać. Następnie opisuje cztery rodzaje eksperymentów lub ustawień, w których ocenia się roboty. Sekcja zaleca określone kategorie danych do zebrania w celu udowodnienia hipotez, debugowania problemów i oportunistycznego poznania nowych rzeczy o robocie. Kończy się studium przypadku, w jaki sposób zebranie dużego zestawu danych pomogło w ustaleniu problemu z czujnikiem.

### **Taksonomia niepowodzeń**

Rysunek przedstawia taksonomię typów awarii robotów przedstawionych w141 jako rozwinięcie wcześniejszej wersji autorstwa Carlsona i Murphy. Awarie robotów, zgodnie z taksonomią, mają trzy źródła.



### **AWARIA ZEWNĘTRZNA**

1. Awaria zewnętrzna, która jest poza bezpośrednią kontrolą robota lub obsługi. Przykładem jest robot zniszczony w wybuchu miny.

### **FIZYCZNE AWARIE**

2. Fizyczna awaria samego robota. Awarie fizyczne można dalej klasyfikować według podsystemów: mobilność, komunikacja, wykrywanie, sterowanie lub obliczenia oraz moc. Awarie mobilności są rzadko spotykane, podczas gdy przerwy w zasilaniu (np. zwarcia) i łączności są częstsze.<sup>38</sup> Niski wskaźnik awarii mobilności i wysoki wskaźnik awarii zasilania i łączności może nie być zaskakujący, ponieważ projektanci często koncentrują się na doskonaleniu efektorów mobilności i zignoruj systemy „drugorzędne”.

### **LUDZKI BŁĄD**

3. Błąd ludzki. Niedawne badanie robotyki katastrof wskazuje, że ponad 50% awarii robotów było spowodowanych błędami ludzkimi. Tak wysoki wskaźnik błędów ludzkich sugeruje, że źródłem błędu jest projektant, a nie operator. Awarie mogą mieć jeden z dwóch skutków. Awaria terminala oznacza zakończenie misji; albo misja została przerwana, albo robot zginął w miejscu. Nieterminowa awaria oznacza, że misja jest kontynuowana, ale z obniżonymi możliwościami.

### **Cztery rodzaje eksperymentów**

Tabela przedstawia cztery rodzaje eksperymentów, które są wykorzystywane przez robotyków do testowania zdolności autonomicznych.

Type of Fieldwork	Driver	Venue	Composer	Outcomes
Controlled Experimentation	hypothesis or performance	Laboratory	Robotacist	statistically valid inferences
Exercise	successful demonstration	Staged World	Stakeholders and Robotacists	familiarity, favorable opinion, accelerated adoption, some feedback
Concept Experimentation	mission suitability	Staged World or Natural World	Stakeholder	gaps analysis, possible new uses, concepts of operations
Participant-Observer	authenticity	Natural World	Stakeholder	case studies of actual uses and concepts of operations, failures and bottlenecks

Każdy typ ma siłę napędową, ostateczny cel lub motywację, miejsce, w którym odbywa się eksperyment, oraz ogólną klasę oczekiwanych wyników. Ponadto autorem ogólnej struktury eksperymentu może być robotnik lub interesariusz korzystający z robota. Eksperyment może odbyć się w realistycznych warunkach, zwanych również wysokiej wierności. Należy pamiętać, że realistyczne warunki odtwarzają zarówno fizyczne właściwości zamierzonego obszaru roboczego, jak i właściwości operacyjne obszaru roboczego; na przykład wszystko jest zgodne ze standardowymi procedurami bezpieczeństwa i procedurami operacyjnymi stosowanymi w tej aplikacji lub przez tego interesariusza. Warunki wahają się od ustawień laboratoryjnych, które poświęcają realizm w zamian za statystyczną powtarzalność, po wykorzystanie robota w prawdziwej misji.

#### **EKSPERYMENT KONTROLOWANY**

W kontrolowanych eksperymentach celem jest przetestowanie hipotezy lub ilościowe ustalenie wydajności określonej zdolności robota. Eksperyment odbywa się w warunkach laboratoryjnych (w pomieszczeniu lub na zewnątrz), gdzie robotnik wyraźnie komponuje eksperyment i kontroluje warunki w celu uzyskania statystycznie ważnych wniosków. Nacisk kładziony jest na powtarzalność.

#### **ĆWICZENIE**

Zgodnie z definicją podaną w Alberts i Hayes, celem ćwiczenia jest przetestowanie systemu w bardziej realistycznym środowisku, zgodnie z oskryptowanym zestawem czynności. Ćwiczenie odbywa się w „zainscenizowanym świecie”, w którym środowisko robota jest zainscenizowane tak, aby było realistyczne, ale powtarzalne, co pozwala na testy fizyczne o wyższej wierności. Oczekuje się, że skrypt będzie miał wyższą wierność operacyjną niż w laboratorium. Ćwiczenia mogą być skomponowane przez robotnika lub interesariusza. Ćwiczenia są zwykle wykorzystywane przez robotyków, aby wypróbować robota w bardziej realistycznych warunkach lub uzyskać informacje zwrotne od interesariuszy. Ćwiczenia, takie jak Top Official National Internal Counterterrorism Exercise (TOPOFF) lub demonstracje zaawansowanej koncepcji technologicznej Departamentu Obrony USA, są wykorzystywane przez interesariuszy w celu promowania znajomości praktycznego korzystania z technologii i zachęcania innych interesariuszy do wyrażenia pozytywnej opinii, co z kolei przyspieszy przyjęcie tej technologii. Przestrzeganie skryptu i chęć zaimponowania uczestnikom i obserwatorom robotem często prowadzi do inżynierii skryptu i środowiska, aby podkreślić mocne strony robota. W związku z tym ćwiczenia nie są tak cenne, jak można by oczekiwać, jeśli chodzi o ocenę rzeczywistej wydajności robota na poziomie systemu lub uzyskanie wglądu w interakcje człowiek-robot.

#### **EKSPERYMENT KONCEPCYJNY**

Ekspertymentowanie z koncepcjami jest być może celem większości robotów jako celem ćwiczenia; że rezultatem ćwiczenia będzie analiza luk w tym, czego brakowało w projekcie systemu, zidentyfikowanie możliwych nowych zastosowań oraz ogólna koncepcja operacji. Ekspertymenty koncepcyjne mają na celu zbadać przydatność systemu technologicznego do misji, od początku do końca, co określa się mianem koncepcji operacji. Koncepcja operacji obejmuje nie tylko misję, ale także logistykę przemieszczania robota do pozycji, ograniczenia dotyczące rozmiaru i wagi, brak prądu do ładowania akumulatorów podczas normalnej zmiany i tak dalej. Scenariusz ćwiczenia zostałby skomponowany przez interesariuszy, aby zaoferować możliwość wprowadzenia technologii, ale nie zmieniania niczego innego w normalnych operacjach. Technolodzy musieliby zbierać dane bez ingerencji w działanie robota lub przebieg ćwiczenia. W ramach eksperymenów koncepcyjnych uczestnicy kontynuują misję, nawet jeśli robot ulegnie awarii, dostarczając w ten sposób cennych informacji zwrotnych na temat słabych punktów systemu. Ekspertymenty koncepcyjne są wykorzystywane w robotyce i interakcji człowiek-maszyna od 2002 roku, w szczególności w serii Summer Institute poświęconej robotyce katastrof.

### **UCZESTNIK-OBSERWATOR**

Warunki fizyczne i operacyjne o najwyższej wierności do oceny robota występują podczas rzeczywistego użytkowania. W tym przypadku robotnik działa jako uczestnik-obszrwator w świecie przyrody (w przeciwieństwie do świata inscenizowanego), podobnie jak w przypadku, gdy etnografowie osadzają się w odległym plemieniu. Dane uzyskane z tych rzeczywistych aplikacji są z natury mniej ilościowe niż inne formy testowania i oceny, ale studia przypadków są przydatne do określenia, w jaki sposób robot będzie rzeczywiście używany w praktyce, co należy zrobić, aby ułatwić te zastosowania i zidentyfikować kiedy robot zawodzi i dlaczego.

### **Dane do zebrania**

Ocena systemu robota wymaga danych. O ile zbieranie danych jest oczywiste, łatwo zapomnieć o zebraniu pełnego zestawu danych. Ekspertymenty na ogół podają zmienną zależną, zmienną niezależną i warunki pracy, robota i inny sprzęt używany do eksperymenru, a także metodę eksperymenralną, a także wyniki. Jednak niedawna analiza 18 badań autonomicznego startu i lądowania w mikro-UAV wykazała, że jedyną wspólną miarą zgłoszoną we wszystkich badaniach był typ pojazdu, dlatego warto przejrzeć, jakie dane należy zebrać. W przypadku trzech z czterech typów eksperymenów, ćwiczeń, eksperymenów koncepcyjnych i eksperymenów uczestnik-obszrwator, książka Disaster Robotics identyfikuje sześć kategorii danych eksperymenralnych do zebrania, jeśli to możliwe: dziennik aktywności, kontekst, widok robota, stan robota, dane zewnętrzne widok robota i widok interakcji człowiek-robot. Książka opisuje również podstawowe elementy protokołu zbierania danych dla robotyki terenowej, w tym zarządzanie danymi. W przypadku czwartego typu eksperymenów, eksperymenów kontrolowanych, celem zbierania danych jest udowodnienie lub obalenie hipotezy. Hipoteza może być bardzo szeroka, często w stylu „działa autonomiczna zdolność”, ale musi istnieć definicja „działań”, a także dane potwierdzające lub obalające hipotezę oraz wystarczająca ilość informacji o danych, aby zapewnić zaufanie do wyniku. Na przykład hipoteza kontrolowanych eksperymenów z możliwością autonomicznego startu i lądowania mikro-UAV może polegać na wykazaniu, że robot może niezawodnie wylądować w każdych warunkach pogodowych lub że robot może lądować tak samo jak człowiek. lub że pilot może przejąć kontrolę podczas przerwanej lądowania. Pierwsza hipoteza sugeruje, że w eksperymencie będą powtarzane starty i lądowania w różnych warunkach pogodowych i będą miały metryki do pomiaru błędu. Tymi wskaźnikami mogą być odległość od celu, odległość i różnica w nagłówkach lub jakakolwiek inna miara, która odzwierciedla definicję „działań”. Druga hipoteza zakładałaby statystyczne porównanie ludzkich pilotów i autopilotów, gdy człowiek i robot lądują w zestawie prawie identycznych sytuacji. Miernik może obejmować

sparowany test T, aby potwierdzić, że osiągi człowieka i robota są różne. Oba eksperymenty przypominałyby klasyczne eksperymenty laboratoryjne z zajęć z chemii i fizyki. Trzecia hipoteza wymagałaby zupełnie innego rodzaju eksperymentu, zgodnie z psychologicznym projektem eksperymentalnym opisanym w Części 18.

## **DANE EKSPERYMENTALNE**

Dane zebrane dla klasycznego eksperymentu, za Duncanem i Murphym, składałyby się z:

\* Zmienne zależne, zmienne niezależne i stałe. Zmienne niezależne to zmienne, które są zmieniane lub manipulowane w celu wytworzenia mierzalnej zmiany zmiennej zależnej. W przypadku lądowania autonomicznego zmienną zależną może być średni błąd między rzeczywistym miejscem lądowania a miejscem docelowym, a zmiennymi niezależnymi algorytmy (np. autonomiczny, ręczny, autonomiczny o różnych parametrach itp.), różni operatorzy, początkowa wysokość zniżania, powierzchnia lądowania itp. Wszystko inne powinno być stałe. Jeśli istnieją aspekty eksperymentu, które nie są stałe, na przykład wiatr, temperatura, pora dnia, zachmurzenie lub inne warunki operacyjne, muszą być mierzone i ich wpływ na wszelkie wyniki jest wyraźnie uwzględniony i omówiony. Te niekontrolowane warunki pracy mogą być niewielkie lub mogą być mylącymi czynnikami, które podważają wszelkie wnioski dotyczące hipotezy.

\* Dane operacyjne. Mogą istnieć dane, które są rejestrowane, co do których oczekuje się, że nie zostaną wykorzystane w hipotezie, ale mogą mieć później wartość. Rejestrowanie całkowitego czasu lotu może odkryć ważne informacje. Na przykład, jeśli UAV nie jest w stanie wylądować w ciągu 2,5 minuty, a operator musi przejąć kontrolę, można to włączyć do algorytmu lub procedur operacyjnych. Podobnie rejestrowanie toru lotu może być przydatne, ponieważ prostszy lub gładniejszy tor może wskazywać na lepsze osiągi lub identyfikować przebiegi, w których wystąpiła anomalia.

\* Sprzęt lub aparatura. Eksperyment musi rejestrować, jakiego robota, ładunku, systemu operacyjnego i tak dalej użyto. Eksperymenty informują również o tym, jaki sprzęt został użyty do zebrania danych na wypadek, gdyby miało to znaczenie. Na przykład użycie kamery do przechwytywania ruchu dałoby inną rozdzielczość śledzenia ruchu opadania niż kamera w telefonie komórkowym.

\* Metoda eksperymentalna. Należy zbierać dane o całym procesie. W przypadku testowania autonomicznego lądowania metoda eksperymentalna może wyglądać następująco. UAV startuje 30 razy z losowej wysokości między 10 a 30 metrów nad poziomem gruntu i w losowej względnej lokalizacji i orientacji kompasu w promieniu 10 metrów od celu lądowania. Operator zainicjowałby autonomiczne lądowanie. Eksperymentalny bieg zostanie przerwany, gdy a) UAV przyziemi i wyłączy się, b) gdy operator przerwie lądowanie z powodu pogody lub nieprzewidzianego zachowania, lub c) lot zostanie przerwany po pięciu minutach bez pomyślnego przyziemienia.

## **SZEŚĆ KATEGORII DANYCH**

Książka Disaster Robotics identyfikuje sześć kategorii danych, które należy zbierać podczas ćwiczeń terenowych, eksperymentów koncepcyjnych i sytuacji uczestnik-obszernik. Są one przydatne do ogólnych eksperymentów, a także do wstępnych testów:

\* Dziennik aktywności. Dziennik aktywności zawiera sekwencję zdarzeń. Dziennik aktywności może informować o udoskonaleniach projektu, takich jak przewidywany czas trwania lotu. Na przykład prace nad inspekcją strukturalną wykazały, że bezzałogowe statki powietrzne mogą wykonać mapę boku wielopiętrowego budynku w ciągu 8–12 minut, a załoga generalnie musiała wylądować i poruszać się, aby zachować bezpieczne operacje podczas mapowania następnej strony. Bez względu na to, ile stron ma budynek, maksymalny czas lotu wynosił 12 minut. Dlatego też bezzałogowiec z dłuższym czasem

pracy na baterii, a co za tym idzie dłuższym czasem lotu, nie był konieczny. Projektanci mogliby wykorzystać te informacje do stworzenia tańszego bezzałogowego statku powietrznego o krótszym czasie trwania na rynku kontroli strukturalnej. Dziennik aktywności może naświetlić interesujące zalety robotów. Na przykład rozmieszczenie UGV w przypadku zdarzenia radiologicznego zajmuje średnio tyle samo czasu, co przeszkolony ratownik, aby się ubrać, ale UGV może szybciej zejść z zasięgu i pozostać dłużej.

\* Kontekst. Kontekst może pomóc wyjaśnić nietypowe zachowanie, a także dlaczego robot czasami działa dobrze, a dlaczego czasami nie. W jednym przypadku śledzony UGV, opracowany do inspekcji rur i kanalizacji, został wykorzystany do przeszukania domu mieszkalnego zmiążdżonego przez lawinę błotną. Gąsienice zerwały się z UGV i misję trzeba było przerwać. Kontekst domu mieszkalnego, w szczególności kontekst, w którym dom miał kudłatą wykładzinę, był niezbędny do zrozumienia, dlaczego robot zawiódł. W przypadku wrażliwych sytuacji interakcji człowieka z robotem kontekst tego, ile snu zespół ma, ogólny stres i presja, którą odczuwa, itd., zapewnia kontekst psychologiczny.

\* Widok oka robota, stan robota i widok zewnętrzny robota. Te zestawy danych są przydatne do debugowania i analizy błędów. Na przykład, jeśli robot zaczyna lekko oscylować w czasie T przed zderzeniem, nagranie wideo pokazujące oscylację służy dwóm celom. Jest to zarówno wskazówka, co mogło pójść nie tak, a także daje znacznik czasu, aby zacząć przeglądać dzienniki stanu robota (Co robił?) i widok oka robota (Co widział?). Widok zewnętrzny można uzyskać z kamery na kasku obserwatora bezpieczeństwa lub dedykowanej kamery śledzącej do działań w pobliżu. Może warto mieć samolot pościgowy lub drugi robot podążający za pierwszym robotem.

\* Widok interakcji człowieka z robotem. Niezależnie od kamery rejestrującej widok robota z zewnątrz, pomocne może być oddzielne nagranie rejestrujące to, co mówią i robią członkowie zespołu. Może to być pomocne przy debugowaniu lub ustaleniu przyczyny awarii, zwłaszcza w identyfikacji interfejsu użytkownika i wąskich gardeł interakcji. Etnograf pełniący rolę obserwatora i sporządzającego notatki jest również pomocny, ponieważ może zidentyfikować problemy z interakcją człowiek-robot i możliwe przyczyny, które mogą być trudne do wydobycia z filmu.

### **Studium przypadku: Eksperymenty koncepcyjne**

Wartość eksperymentowania nad koncepcją i zbierania danych ilustruje następujący przypadek, w którym analityk odkrył, że problem z wydajnością systemu, początkowo przypisywany błędowi człowieka, był w rzeczywistości spowodowany nieoczekiwanym zachowaniem czujnika. Odkrycie zaskakującego zachowania czujnika było wynikiem wysokiej wierności eksperymentów koncepcyjnych jako metody, wykorzystania etnografa, który zauważył anomalię w zachowaniu operatora, oraz wszechstronnego gromadzenia danych, które umożliwiły analizę post hoc.

Eksperyment koncepcyjny z 2013 roku w Disaster City symulował reakcję na trzęsienie ziemi w skrzydle medycyny nuklearnej dużego szpitala. Częściowo zawałony wielopiętrowy budynek, zwany Prop 133, w Disaster City symulował szpital, a źródło radiologiczne cezu było ukryte gdzieś w budynku. Zobacz rysunki a i b.

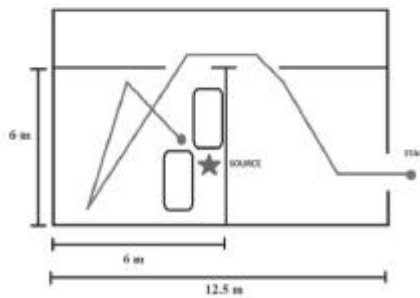
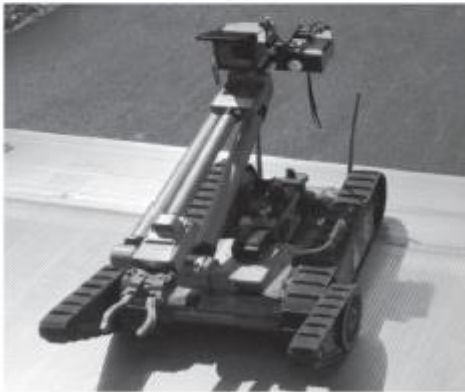




a.



b.



d.

Zespoły ratownictwa pożarowego z materiałami niebezpiecznymi (HazMat), przeszkolone w zakresie zdarzeń chemicznych, biologicznych, radiologicznych, nuklearnych i wybuchowych (CBRNE), musiały zareagować na trzęsienie ziemi, przeszukać obszar i szpital pod kątem promieniowania i osób w niebezpieczeństwie oraz zlokalizować wszelkie materiały radiologiczne i usunąć je. Eksperyment koncepcyjny wprowadził robota naziemnego, bezałogowego statku powietrznego i narzędzia do wizualizacji, z których każdy miał skrócić czas potrzebny na podjęcie decyzji. Hipotezy zostały potwierdzone, ale robot naziemny nie działał zgodnie z oczekiwaniami.

Wdrożenie robota naziemnego było następujące: iRobot Packbot 510 z czujnikiem Canberra Radiac, małym czujnikiem powszechnie używanym przez zespoły HazMat, pokazany na rysunku c. został rozpakowany i skonfigurowany w tym samym czasie, co tradycyjne zespoły HazMat rozpoczynające zakładanie odzieży ochronnej. Konfiguracja robota nie była znacząco szybsza niż zespołów ręcznych, więc zarówno robot, jak i zespół ludzi rozpoczęli poszukiwania źródła promieniotwórczego w tym samym czasie. Ale gdy robot zaczął poruszać się w dół zasięgu, zapewniał ciągłe odczyty radiologiczne, czego zespół manualny nie był w stanie wykonać. Dowódca incydentu ustalił, że promieniowanie nie było tak złe, jak się obawiano, i zaczął przenosić personel i sprzęt bliżej miejsca. W międzyczasie robot dotarł do zawalonego budynku szybciej, niż respondenci mogli przejść w dół, zbadał zewnętrzną część budynku, a następnie wszedł do budynku w poszukiwaniu materiału radiologicznego. Chociaż pomieszczenia były małe, operatorowi i specjaliście od HazMat prawie 30 minut zajęło znalezienie jasnyniebieskiego cylindra zawierającego cez na podłodze między dwoma biurkami. Podczas spotkania po akcji po ćwiczeniu niespodziewanie długi czas lokalizacji był początkowo przypisywany błędowi operatora, ale etnograf stacjonujący z robotem zaobserwował i poinformował, że operator stale narzekał, że odczyty radiologiczne były niespójne i wydawały się bardziej stabilne i bardziej czytelne kiedy robot cofał się, aby ominąć przeszkody. Student Dexter Duckworth z zespołu badawczego przejrzał zebrane obszernie dane i odkrył, że problem nie wynikał z błędu ludzkiego. Student potrafił

zrekonstruować ścieżkę robota względem źródła (rysunek d.) i powiązać odczyty promieniowania z położeniem wzdłuż ścieżki. Dane pokazały, że odczyty z robota były rzeczywiście wyższe, gdy robot był odwrócony od źródła. Było to bardzo zaskakujące, ponieważ czujnik Radiac był dookólny i przeszedł wszystkie testy kalibracyjne i jest de facto standardem dla czujników radiologicznych w Stanach Zjednoczonych. Zespół skontaktował się z Japońską Agencją Energii Atomowej (JAEA), która brała udział w wykorzystaniu robotów podczas wypadku jądrowego w Fukushima Daiichi i jego późniejszej likwidacji, aby sprawdzić, czy agencja była świadkiem tego dziwnego zachowania czujnika, jako najczęściej używanego robota w Fukushima. Był iRobot Packbot 510. JAEA odpowiedziała, że pracownicy nigdy nie próbowali używać Packbota do lokalizacji źródeł promieniotwórczych, używali go tylko do ogólnych badań radiologicznych. Agencja miała jednak wyjaśnienie tego, co się dzieje. Czujnik Radiac został zaprojektowany do noszenia przez człowieka, zwykle przypięty do paska. Na tej wysokości czujnik rejestruje promieniowanie pod dowolnym kątem, ponieważ ludzkie ciało pochłania stosunkowo mało. W scenariuszu Disaster City Radiac został zamontowany z przodu wnęki ładunkowej Packbota, z jednej strony na ciężkiej metalowej ramie. Metalowa rama blokowała promieniowanie, zmieniając Radiac w czujnik kierunkowy wycelowany w tył robota. Zespoły JAEA i HazMat z przyjemnością dowiedziały się o tym zachowaniu, aby mogły być tego świadome, gdyby używały robotów do lokalizacji źródła radiologicznego.

## **Podsumowanie**

Ten Część skoncentrował się na odpowiedziach Jak zaprojektować inteligentnego robota? poprzez zapewnienie zasad projektowania całego systemu robota. Pomimo częstotliwości prób ilościowego określenia, o ile bardziej autonomiczny jest jeden system w porównaniu z innym, za pomocą „warstw” lub metryk, porównania te mają tendencję do odwracania uwagi od ekologicznego podejścia do projektowania. Podejście ekologiczne wymaga od projektanta rozważenia współzależności między robotem, środowiskiem i zadaniem. Wyraźne uwzględnienie tych współzależności może doprowadzić projektanta do odkrycia afordancji i zachowań, które mogą uprościć programowanie i zwiększyć niezawodną wydajność. Niezależnie od proaktywnego wysiłku projektowego, projekt może być niedoskonały lub roboty mogą zawieść z innych powodów. Awarie mogą wynikać z jednego z pięciu zorientowanych sprzętowo podsystemów samego robota, które nazywane są awariami fizycznymi. Mogą wynikać z błędu ludzkiego, który może stanowić 50% awarii. Robot może zawieść z powodu zdarzenia lub warunków w świecie zewnętrznym, takich jak zmiążdżenie podczas eksploracji zawalonego budynku. Sprawdzenie, czy system będzie działał inteligentnie, może być trudne, ponieważ robot zwykle pracuje w otwartym świecie. Jak zauważono w Części 3, algorytmy mogą być niedeterministyczne i z pewnością otwarty świat wzmacnia spektrum nieoczekiwanych zdarzeń, które mogą być zbyt liczne lub trudne do jednoznacznego modelowania. Oczywiście ocena systemu robota jest bardziej złożona niż testowanie algorytmu. W zależności od celów lub motywów oceny, projektant może chcieć wybrać między formalnie kontrolowanym eksperymentem laboratoryjnym, ćwiczeniem, eksperymentem koncepcyjnym lub wprowadzeniem systemu robota do świata przyrody. Szczególnie ważne jest, aby nie mylić ćwiczenia z eksperymentowaniem koncepcji, ponieważ ćwiczenia gwarantują, że cele zostaną pomyślnie osiągnięte, ale eksperymentowanie z koncepcją bada potencjalne problemy na poziomie systemu i zachęca do niepowodzeń. Niezależnie od rodzaju eksperymentu użytego do oceny systemu robota, projektant powinien starać się zebrać wiele różnych ogólnych danych: dziennik aktywności, kontekst, widok oka robota, stan robota, widok zewnętrzny robota i człowiek-robot widok interakcji. Nie zawsze można zbierać wszystkie rodzaje danych, ale im więcej, tym lepiej. Teraz, gdy czytelnik jest już zaznajomiony z projektowaniem autonomicznych zdolności i systemu robotów, kolejny, ostatni Część zajmie się etyką. W pewnym sensie ten Część przygotowuje grunt pod etykę, ponieważ projektant robotów jest zobowiązany etyką zawodową do stosowania najlepszych zasad w

projektowaniu autonomicznych zdolności i powinien być świadomy większych problemów etycznych związanych z używaniem robotów.