

Opis funkcjonalności do spełnienia przez oprogramowanie obliczeniowe. Są to m.in.:

- Reprezentacja systemów liniowych za pomocą transmitancji, równań stanu, postaci wzmocnienia zer i biegunów (*zero-pole-gain*) oraz odpowiedzi częstotliwościowej.
- Odpowiedź skokowa, charakterystyka Nyquista, a także inne narzędzia do badania stabilności i wydajności w dziedzinie czasu i częstotliwości.
- Metoda root locus, charakterystyka Bodego, LQR, LQG, a także inne techniki projektowania systemów: klasyczne oraz wykorzystujące równania stanu.
- Automatyczne strojenie regulatorów PID oraz harmonogramowanie wzmocnienia.
- Konwersja reprezentacji modeli, dyskretyzacja modeli czasu ciągłego, aproksymacja niskiego rzędu dla systemów wysokiego rzędu.
- Algorytmy estymacji stanu (filtracja Kalmana) oraz narzędzia do projektowania regulatorów LQR/LQG.
- Współpraca z wybranymi narzędziami do identyfikacji systemów w celu modelowania obiektów za pomocą danych eksperymentalnych.
- Interaktywny interfejs graficzny użytkownika, unifikujący podstawowe zadania dopasowywania krzywych i płaszczyzn.
- Liniowe i nieliniowe równania regresji z możliwością użycia równań niestandardowych.
- Biblioteka modeli regresji ze zoptymalizowanymi punktami startowymi i parametrami solvera.
- Dopasowywanie nieparametryczne, za pomocą interpolacji, funkcji sklepanych i średnich ruchomych.
- Procedury wstępnego przetwarzania danych: skalowanie danych, podział na podgrupy, wygładzanie, usuwanie błędnych punktów.
- Procedury po przetworzeniu: interpolacja, ekstrapolacja, estymacja przedziałów ufności, obliczanie całek i pochodnych.
- Funkcje i aplikacje do obsługi przemysłowych urządzeń do akwizycji danych.
- Wsparcie dla urządzeń DAQ wykorzystujących połączenie USB, PCI, PCI Express, PXI, PXI Express od różnych producentów, w tym National Instruments.
- Obsługa analogowych wejść i wyjść, liczników, timerów oraz cyfrowych wejść i wyjść.
- Bezpośredni dostęp do parametrów sprzętowych urządzeń pomiarowych, takich jak pomiar jedno- i wielokanałowy, pojedynczy pomiar punktowy i pomiary buforowane.
- Automatyczne uruchamianie procedur poprzez zdarzenia sprzętowe i programowe.
- Możliwość tworzenia własnych interfejsów dla nieobsługiwanej sprzętu.
- Aplikacje graficzne do konfiguracji ustawień akwizycji i generacji danych.
- Zestaw narzędzi do głębokiego uczenia zapewnia możliwość łatwego projektowania i implementowania rozwiązań opartych o głębokie sieci neuronowe, udostępniając narzędzia, algorytmy oraz wstępnie wytrenowane modele.
- Za pomocą konwolucyjnych sieci neuronowych (ConvNet, CNN) i sieci LSTM (Long Short-Term Memory) można przeprowadzać klasyfikację i regresję danych obrazów, szeregów czasowych i tekstowych.

- Możliwość tworzenia zaawansowanych architektur sieci, takich jak generatywne sieci przeciwstawne (GAN) i sieci syjamskie, przy użyciu algorytmów automatycznego różnicowania, niestandardowych pętli trenowania i współdzielonych wag.
- Możliwość wizualizacji aktywacji warstw i graficznego monitorowania postępu trenowania.
- Zwiększanie efektywności uczenia się dzięki funkcjom przetwarzającym dane przed i po uczeniu sieci.
- Graficzny interfejs użytkownika (Deep Network Designer) do tworzenia, uczenia i symulowania sieci neuronowych.
- Aplikacja (Experiment Manager) ułatwiająca zarządzanie wieloma eksperymentami uczenia głębokiego, śledzenie parametrów trenowania, analizowanie wyników i porównywanie kodu z różnych eksperymentów.
- Narzędzia do importowania modeli sieci z innych bibliotek takich jak: TensorFlow 2, TensorFlow-Keras i PyTorch, formatu ONNX (Open Neural Network Exchange) oraz Caffe.
- Możliwość wyeksportowania modeli sieci i warstw do biblioteki TensorFlow 2 i formatu ONNX.
- Obsługa uczenia transferowego (ang. *transfer learning*) wstępnie wytrenowanych modeli.
- Eksportowanie wytrenowanych modeli do Simulinka.
- Algorytmy regresji, klasyfikacji, grupowania i modelowanie nieliniowych układów dynamicznych przy użyciu płytkich sieci neuronowych (ang. *shallow networks*).
- Modelowanie systemów DSP czasu rzeczywistego w aplikacjach komunikacji, radarów, audio, urządzeń medycznych, IoT i innych.
- Wbudowane metody do projektowania filtrów, w tym zaawansowanych filtrów wielostopniowych, filtrów typu *multirate* i filtrów adaptacyjnych.
- Szybka transformacja Fouriera, estymacja widmowa, wykorzystanie okien, statystyki sygnałów i algebra liniowa.
- Graficzne analizatory widma sygnałów oraz dynamiczne wykresy sygnałów w domenie czasu.
- Algorytmy wspierające zmiennoprzecinkowe, stałoprzecinkowe i całkowite typy danych.
- Wsparcie dla modelowania stałoprzecinkowego oraz generacji kodu HDL.
- Możliwość projektowania systemów rozmytych typu Mamdani i Sugeno oraz systemów złożonych.
- Wsparcie dla systemów rozmytych typu 2 z funkcjami przynależności uwzględniającymi niepewność.
- Wsparcie dla narzędzi pozwalających na dostrajanie algorytmów rozmytych metodami optymalizacyjnymi jak np. algorytmy genetyczne.
- Możliwość tworzenia algorytmów adaptacyjnych w postaci neuro-rozmytych.
- Wsparcie dla narzędzi do generacji kodu C/C++ oraz kodu Structured Text.
- Aplikacja w Live Editor z graficznym interfejsem użytkownika do definiowania (typ zadania, zmienne decyzyjne, funkcja celu, ograniczenia, parametry metody) i rozwiązywania problemów optymalizacji oraz monitorowania procesów z tym związanych.
- Optymalizacja nieliniowa i wielokryterialna.

- Solvery dla regresji nieliniowej metodą najmniejszych kwadratów, dopasowywania danych i równań nieliniowych.
- Rozwiązywania zadań programowania liniowego, także mieszanych (z ciągłymi i całkowitoliczbowymi zmiennymi decyzyjnymi) oraz zadań programowania kwadratowego.
- Narzędzia do automatycznego różnicowania funkcji celu oraz ograniczeń, mające za zadanie uzyskanie dokładniejszych rozwiązań w krótszym czasie.
- Możliwość rozwiązywania intensywnych obliczeniowo zadań z wykorzystaniem procesorów wielordzeniowych, kart GPU oraz klastrów komputerowych.
- Zrównoleglone pętle for (parfor) do uruchamiania równoległych zadań na wielu procesorach.
- Wsparcie dla procesorów graficznych NVIDIA z obsługą architektury CUDA.
- Pełne wykorzystanie procesorów wielordzeniowych.
- Interaktywne i wsadowe wykonywanie aplikacji.
- Tablice rozproszone oraz równoległe wykonywanie identycznych podzadań dla różnych danych (przetwarzanie współbieżne) do obsługi dużych zestawów danych.
- Możliwość implementacji kontrolerów i algorytmów podejmowania decyzji dla złożonych aplikacji, takich jak alokacja zasobów, robotyka i systemy autonomiczne.
- Możliwość ewaluacji jedno- lub wieloagentowych algorytmów uczenia przez wzmacnianie udostępnianych przez moduł lub opracowywania własnych.
- Możliwość eksperymentowania i parametryzacji procesu trenowania poprzez ustawienia hiperparametrów oraz możliwość monitorowania postępów trenowania i symulacji wytrenowanych agentów w sposób interaktywny (przy pomocy narzędzi modułu) lub programowo.
- Narzędzia i algorytmy do projektowania, symulacji i testowania manipulatorów, robotów mobilnych i robotów humanoidalnych.
- Algorytmy sprawdzania kolizji, generowania trajektorii, kinematyki prostej i odwrotnej oraz dynamiki bryły sztywnej.
- Algorytm mapowania, lokalizacji, planowania ścieżki, śledzenia ścieżki i sterowania ruchem.
- Biblioteka modeli komercyjnych robotów przemysłowych.
- Możliwość kosymulacji z symulatorem Gazebo.
- Sprawdzanie kolizji, definiowanie kształtów kolizji i wykrywanie kolizji pomiędzy siatkowymi geometriami.
- Funkcje i bloki do analizy i strojenia systemów sterownia pod kątem ich wydajności i niezawodności.
- Tworzenie modeli z niepewnościami (niepewność parametrów, niezamodelowana dynamika).
- Analizy najgorszych przypadków zapasu stabilności i wrażliwości na zakłócenia.
- Generowanie kontrolerów ze zredukowaną czułością na wariację parametrów i błędy modelowania.

- Automatyczne strojenie regulatorów SISO i MIMO, zdecentralizowane, o stałej strukturze oraz z wieloma pętlami sprzężenia zwrotnego.
- Interfejs komunikacyjny dla narzędzia do obliczeń inżynierskich oraz symulacji dynamicznych systemów w domenie czasu z systemami ROS oraz ROS 2.
- Możliwość obustronnej wymiany danych z systemami ROS oraz ROS 2.
- Wsparcie dla wybranych narzędzi do automatycznej generacji kodu dla tworzenia aplikacji do sieci ROS (ROS Nodes).
- Wsparcie dla funkcjonalności ROS jak: Publishers, Subscribers, Services, Actions, Parameter Server, rosbag i niestandardowe wiadomości (messages) ROS.
- Wsparcie dla systemów ROS/ROS 2 działających pod różnymi systemami operacyjnymi, takimi jak Windows, Mac OS i Linux.
- Funkcje i aplikacje do analizy, preprocessingu i ekstrakcji cech z jednostajnie i niejednostajnie próbkowanych sygnałów.
- Transformacje sygnałów, m.in. szybka transformacja Fouriera (FFT), krótkoczasowa transformacja Fouriera (STFT), transformacja Hilberta.
- Metody projektowania filtrów FIR i IIR, ich analiza i implementacja.
- Możliwość projektowania filtrów analogowych: Butterwortha, Czebyszewa, Bessla i eliptycznych oraz ich konwersji na postać cyfrową za pomocą metod transformacji biliniowej i niezmienności odpowiedzi impulsowej.
- Aplikacja Filter Designer do interaktywnego projektowania i analizy filtrów o określonych charakterystykach.
- Funkcje do generacji sygnałów takich jak sinus, prostokąt, piła, delta Kroneckera.
- Pomiar i analizy statystyczne sygnałów.
- Algorytmy estymacji widmowej gęstości mocy, m.in. periodogram, funkcje Welch, Burga, Yule-Walkera.
- Pomiar widma mocy sygnału i parametrów takich jak SNR, THD i SINAD.
- Narzędzia do analizy okien czasowych.
- Modelowanie parametryczne i predykcyjne systemów liniowych.
- Narzędzia do etykietowania fragmentów sygnałów do celów trenowania i walidacji modeli uczenia maszynowego.
- Możliwość przeprowadzenia analizy modalnej oraz analizy rzędu sygnałów wibracyjnych.
- Wsparcie generacji kodu C/C++ oraz zoptymalizowanego kodu CUDA.
- Jedno środowisko do modelowania i symulacji systemów mechanicznych, elektrycznych, dynamiki płynów, termicznych, a także innych wielodomenowych systemów fizycznych.
- Biblioteki bloków do modelowania fizycznego oraz elementy matematyczne dla opracowania własnych komponentów.
- Jednostki fizyczne dla parametrów i zmiennych, z automatyczną obsługą konwersji wszystkich jednostek.
- Automatyczna redukcja zmiennych w równaniach symbolicznych oraz metody numeryczne do rozwiązywania równań różniczkowo-algebraicznych (DAE), w tym obsługa zdarzeń.

- Specjalne solvery pozwalające na symulację w czasie rzeczywistym oraz testy hardware-in-the-loop (HIL).
- Możliwość symulacji modeli, które zawierają bloki pochodzące z innych produktów związanych z modelowaniem fizycznym, bez konieczności zakupu tych produktów.
- Automatyczne wykorzystywanie funkcji interpolujących zdefiniowanych w oparciu o wprowadzony lub importowany zbiór punktów dyskretnych
- Możliwość wyboru parametrów płynów w zależności od typu symulacji przepływu: izotermicznego, nieizotermicznego lub wielofazowego
- Możliwość symulacji potencjalnych awarii układów
- Jedno środowisko do symulacji systemów mechanicznych w 3D razem z wielodomenowymi systemami fizycznymi i algorytmami sterowania.
- Definiowanie bryły sztywnej za pomocą standardowej geometrii oraz niestandardowych profili.
- Możliwość importu modeli brył utworzonych w oprogramowaniu typu CAD
- Połączenia i ograniczenia mechaniczne dotyczące standardowych i niestandardowych relacji kinematycznych.
- Modele symulacyjne pozwalają na obliczanie ruchu i obliczanie siły.
- Animacje 3D dynamiki systemów wielocłonowych.
- Wsparcie dla generacji kodu C.
- Środowisko graficzne do tworzenia i zarządzania hierarchicznymi schematami blokowymi.
- Biblioteki predefiniowanych bloków do modelowania systemów dyskretnych oraz ciągłych.
- Silnik symulacji ze stało- i zmiennokrokovymi solverami ODE.
- Bloki do wizualizacji wyników symulacji.
- Narzędzia zarządzania projektem i danymi.
- Narzędzia importu kodu C i C++ do modeli.
- Implementacja algorytmów na tanich platformach sprzętowych (*low-cost hardware platforms*), takich jak Arduino, Raspberry Pi, LEGO Mindstorms EV3.
- Możliwość tworzenia projektów ułatwiających zarządzanie pracą w środowisku oraz integrację z systemami kontroli wersji (np. SVN, GIT, Github)
- Interfejs graficzny do wyszukiwania utworzonych przez użytkownika schematów blokowych
- Mechanizmy debugowania symulacji tworzonych modeli, wykonywania krokowego oraz pauzowania symulacji
- Możliwość pracy w chmurze za pośrednictwem przeglądarki WWW ze stałym dostępem do internetu
- Użycie wygenerowanego kodu w aplikacjach czasu rzeczywistego i nierzeczywistego.
- Wsparcie dla całkowitych, zmiennie- i stałoprzecinkowych typów danych.
- Strojenie parametrów oraz monitoring sygnałów w trybie symulacji zewnętrznej.
- Możliwość implementacji kodu na wybranych platformach wbudowanych
- Generacja niezależnych komponentów FMU (*Functional Mockup Units*) zgodnych ze standardem FMI (*Functional Mockup Interface*) z modeli Simulinka po zainstalowaniu dodatkowego modułu FMU Builder

- Automatyczne strojenie systemów o architekturze regulacji SISO i MIMO, w tym regulatorów PID.
- Strojenie regulatorów PID zaimplementowanych na docelowej platformie.
- Znajdowanie punktów pracy i linearyzacja modeli.
- Oparte na symulacji obliczenia częstotliwościowej odpowiedzi modelu.
- Funkcje do tworzenia skryptów automatycznej linearyzacji.
- Możliwość strojenia regulatorów poprzez ręczne kształtowanie charakterystyk amplitudowo-fazowych
- Estymacja parametrów modelu na podstawie testowych danych.
- Możliwość przeprowadzania analiz typu parameter sweep w celu oceny wpływu parametrów modelu na jakość optymalizacji i przeprowadzanie optymalizacji z wyrażeniem parametrów w postaci zakresu możliwych wielkości.
- Graficzna specyfikacja wymagań nałożonych na odpowiedź modelu i wizualne monitorowanie postępu optymalizacji.
- Interfejs skryptowy dla programistycznego projektowania problemów optymalizacji.
- Środowisko do graficznego modelowania i symulacji algorytmów opartych o diagramy przepływu sygnału, maszyny stanowe, tablice przejść stanowych oraz tablice prawdy.
- Wsparcie dla rozbudowanych funkcjonalności wprowadzających do modelowanego systemu hierarchii, wykonywania równoległego, operatorów logiczny oraz zdarzeń.
- Możliwość projektowania standardowych maszyn stanowych w konwencji Mealy'ego oraz Moore'a wraz z regułami, które wprowadzają.
- Narzędzia ułatwiające debugowanie algorytmów logicznych jak np. breakpointy, wykonywanie sekwencyjne, analiza przejść i warunków oraz wskazywanie konfliktów przejść, martwej logiki, niespójności stanów czy też przeddefiniowania/niedodefiniowania tablic prawdy.
- Posiada dedykowany interfejs do analizy pracy maszyny stanowej (liczba przejść, czas aktywności stanów, zliczanie liczby wywołań funkcji)
- Wsparcie dla wybranych narzędzi do automatycznej generacji kodu, testowania i analizy pokrycia testami, zarządzania wymaganiami, weryfikacji algorytmów pod względem zgodności z wybranymi normami IEC/ISO/EN, analizy formalnej algorytmów oraz statycznej analizy kodu źródłowego.
- Moduł udostępnia funkcje i zestaw narzędzi do opisywania, analizowania i modelowania danych.
- Zestaw narzędzi zapewniający nadzorowane, częściowo nadzorowane i nienadzorowane algorytmy uczenia maszynowego, w tym maszyny wektorów nośnych (SVM), wzmocnione drzewa decyzyjne, k-średnie i inne metody grupowania.
- Możliwość użycia statystyk opisowych, wizualizacji i klastrowania do eksploracyjnej analizy danych, dopasowywania rozkładów prawdopodobieństwa do danych, generowania liczb losowych dla symulacji Monte Carlo i wykonywania testów hipotez statystycznych.
- Algorytmy regresji i klasyfikacji umożliwiają wyciąganie wniosków na podstawie danych i tworzenie modeli predykcyjnych interaktywnie przy użyciu dedykowanych narzędzi środowiska (Classification Learner, Regression Learner) lub programowo przy użyciu rozwiązania AutoML.

- Do wielowymiarowej analizy danych i wyodrębniania cech zestaw narzędzi zapewniający analizę głównych składowych (PCA), regularyzację, redukcję wymiarowości i metody wyboru cech, które pozwalają zidentyfikować zmienne o najlepszej mocy predykcyjnej.
- Możliwość stosowania technik interpretacji danych, takich jak wykresy zależności częściowych i LIME.
- Wsparcie analizy zbiorów danych, które są zbyt duże, aby można je było przechowywać w pamięci (Big Data).
- Identyfikacja transmitancji, modelu procesu oraz modelu w postaci równań stanu z wykorzystaniem odpowiedzi w dziedzinie czasu i częstotliwości.
- Estymacja online parametrów modelu.
- Modelowanie szeregów czasowych (AR, ARMA) i prognozowanie.
- Identyfikacja nieliniowych modeli ARX i modeli Hammerstein-Wienera z nieliniowościami wejścia-wyjścia, takimi jak nasycenie i martwa strefa.
- Identyfikacja liniowych i nieliniowych systemów w postaci „szarej skrzynki”.
- Estymacja opóźnienia, usuwanie trendu, filtrowanie, resampling oraz rekonstrukcja brakujących danych.
- Aplikacja do interaktywnej estymacji modeli liniowych i nieliniowych systemu na bazie zmierzonych danych wejściowych i wyjściowych.
- Ciągła transformacja falkowa (CWT), skalogram i falki koherentne do analizy cech spektralnych w czasie, identyfikacji powtarzalnych w czasie wzorów w dwóch sygnałach oraz filtracji zlokalizowanej w dziedzinie czasu.
- Dyskretna analiza falkowa, w tym decymowana, podwójnego drzewa i transformat falkowych pakietowych – do celów analizy sygnałów i obrazów o różnych rozdzielczościach i znajdowania punktów zmiany, nieciągłości i innych artefaktów.
- Zastosowanie transformacji stałego Q (CQT) oraz transformacji falkowej z przestrajalnym współczynnikiem Q (TQWT) pozwalającej na dyskretną analizę sygnałów z wykorzystaniem wybranego współczynnika Q dla dekompozycji i rekompozycji.
- Kompresja i rekonstrukcja sygnałów i obrazów, w tym dopasowujące algorytmy poszukiwań.
- Interaktywne aplikacje, w tym narzędzie Wavelet Signal Denoiser pozwalające na wizualizację i odszumianie rzeczywistych sygnałów jednowymiarowych.
- Banki rekonstrukcyjnych filtrów ortogonalnych i nieortogonalnych, w tym Daubechies, Coiflet, Haar, Fejer-Korovkin.