Omówienie NVIDIA Isaac Sim: architektura, symulacja i integracja

Podsumowanie

NVIDIA Isaac Sim to zaawansowana platforma symulacyjna dla robotyki, zbudowana na fundamencie NVIDIA Omniverse™ Kit i standardzie Universal Scene Description (USD). Dokument ten syntetyzuje kluczowe aspekty platformy, obejmując mechanizmy kontroli robotów, zaawansowane struktury danych sensorycznych, integrację z systemem ROS 2 oraz narzędzia do generowania danych syntetycznych (SDG). Isaac Sim oferuje dwa główne tryby pracy: **Extension Workflow** (interaktywny, z obsługą "hot-reloading") oraz **Standalone Workflow** (skryptowy, z pełną kontrolą nad krokiem czasowym symulacji). Platforma umożliwia precyzyjne modelowanie fizyki, importowanie złożonych złożeń CAD oraz optymalizację tras za pomocą NVIDIA cuOpt, stanowiąc kompletne środowisko do rozwoju i testowania inteligentnych maszyn.

---

1. Architektura Systemu

Isaac Sim bazuje na obiektowym podejściu do zarządzania światem wirtualnym poprzez Core API, które abstrahuje złożoność niskopoziomowych operacji USD.

Kluczowe Klasy Core API

- **World:** Klasa typu Singleton, stanowiąca centralny punkt interakcji z symulatorem. Zarządza zdarzeniami czasowymi, fizyką, renderowaniem i resetowaniem sceny.
- **Scene:** Zarządza zasobami symulacyjnymi (assetami) na scenie USD. Umożliwia dodawanie, inspekcję i manipulację obiektami.

Tab.1. Porównanie trybów pracy

Funkcja	Extension Workflow	Standalone Workflow
Uruchamianie	Z poziomu GUI Isaac Sim	Z terminala przez skrypt Python (python.sh)
Kontrola czasu	Asynchroniczna (zarządzana przez Kit)	Synchroniczna (jawne wywołanie world.step())
Zastosowanie	Tworzenie interaktywnych narzędzi i GUI	Zadania batchowe, uczenie maszynowe (RL)

Edycja kodu	Obsługuje "hot-reloading" (zapis pliku aktualizuje symulację)	Wymaga ponownego uruchomienia skryptu

---

2. Modelowanie i inspekcja robotów

Isaac Sim umożliwia precyzyjną kontrolę nad strukturą mechaniczną i właściwościami fizycznymi robotów.

Importowanie modeli

- **URDF i MJCF:** Wsparcie dla standardowych formatów opisu robotów. Modele są konwertowane na format USD, przy czym Mesh są instancjonowane dla optymalizacji wydajności.
- **Onshape Importer:** Umożliwia importowanie złożeń CAD bezpośrednio z platformy chmurowej Onshape, zachowując relacje kinematyczne (mates) i przekształcając je w stawy (joints) fizyczne.
- **Konwencje Nazewnictwa:** Znaki specjalne w nazwach są zastępowane podkreśleniami, aby zachować zgodność ze standardem USD.

Napędy i właściwości fizyczne

Platforma oferuje zaawansowaną konfigurację stawów:

- **Drive Type:** Wybór między napędem siłowym (Force – model sprężyna-tłumik) a akceleracyjnym (Acceleration – niezależny od masy).
- **Joint Control:** Możliwość sterowania pozycją (stiffness) lub prędkością (damping).
- **Physics Inspector:** Narzędzie GUI do badania limitów stawów, sztywności oraz tłumienia w czasie rzeczywistym.

---

3. Struktury danych sensorycznych i GenericModelOutput

Kluczowym elementem percepcji w Isaac Sim jest ustandaryzowany format wyjściowy sensorów – **GenericModelOutput**.

Standard ISO8855

Wszystkie sensory (Lidar, Radar, Ultrasonic, IDS) operują w układzie współrzędnych ISO8855:

- **Front:** Oś +x, **Lewo:** Oś +y, **Góra:** Oś +z.
- **Kąty:** Azymut [-180, 180]°, Elewacja [-90, 90]°.

Charakterystyka GenericModelOutput

Jest to ciągły bufor pamięci zawierający:

- **Magic Number:** 0x4E474D4F ("NGMO") dla identyfikacji.
- **Motion Compensation:** Obsługa danych kompensowanych i niekompensowanych (istotne przy ruchomych sensorach, np. rotacyjnych Lidarach).
- **Auxiliary Data:** Dane specyficzne dla modalności, np. emitterId dla Lidaru, radial velocity dla Radaru czy objectId dla czujników głębi (IDS).

---

4. Integracja z ROS 2

Isaac Sim oferuje natywny mostek (bridge) dla systemów ROS 2 Humble i Jazzy, działający na Pythonie 3.11.

Możliwości ROS 2 Bridge

- **Publikowanie Percepcji:** Przesyłanie danych RGB, głębi, chmur punktów oraz etykiet segmentacji semantycznej.
- **Simulation Control:** Możliwość sterowania stanem symulacji (Play/Pause/Stop/Step) oraz manipulacji obiektami (Spawn/Delete) poprzez serwisy ROS 2.
- **Odometria i Transformacje:** Automatyczne generowanie drzew transformacji (TF) i danych odometrycznych z robotów mobilnych.

Wdrażanie polityk Reinforcement Learning (RL)

Isaac Sim umożliwia uruchamianie wyuczonych polityk (np. dla H1) poprzez ROS 2:

1. **Node ROS 2:** Zbiera obserwacje (dane IMU, stany stawów) i przesyła je do polityki.
2. **Akcje:** Wynikowe pozycje stawów są przesyłane z powrotem do symulatora jako komendy sterujące (JointState).
3. **Synchronizacja:** Wykorzystanie On Physics Step w OmniGraph zapewnia, że logika kontroli wykonuje się dokładnie w krokach fizyki.

---

5. Generowanie danych syntetycznych (Replicator)

Narzędzie **Isaac Sim Replicator** służy do tworzenia wysokiej jakości zbiorów danych do uczenia sieci neuronowych (SDG – Synthetic Data Generation).

Komponenty Replicatora

- **Semantics Schema Editor:** Narzędzie do etykietowania obiektów (np. "klasa: robot", "klasa: przeszkoda"), co jest niezbędne dla adnotatorów (Bounding Boxes, Segmentation).
- **Randomizacja:** Możliwość losowania oświetlenia, tekstur, pozycji obiektów oraz parametrów kamer w każdym klatce symulacji.
- **Synthetic Data Recorder:** GUI do nagrywania danych w różnych formatach z możliwością pisania własnych modułów zapisu (Custom Writers).

---

6. Funkcje optymalizacyjne (NVIDIA cuOpt)

Isaac Sim integruje się z usługą **NVIDIA cuOpt**, umożliwiając rozwiązywanie problemów marszrutyzacji (VRP) w czasie rzeczywistym.

- **Waypoint Network:** Tworzenie grafów punktów nawigacyjnych w środowisku magazynowym.
- **Semantic Zones:** Definiowanie obszarów o wysokim koszcie (np. strefy ograniczonego ruchu), które cuOpt uwzględnia przy optymalizacji trasy.
- **Zastosowanie:** Zarządzanie flotą robotów mobilnych (AMR) w celu minimalizacji czasu transportu przy uwzględnieniu pojemności pojazdów i okien czasowych.

---

7. Kontekst technologiczny: Spatial Computing i DDS

Platforma wpisuje się w szerszy trend **Spatial Computing** (obliczeń przestrzennych), który definiuje interakcję człowiek-komputer w trójwymiarowej przestrzeni świata rzeczywistego.

- **Middleware Komunikacyjne:** Isaac Sim wykorzystuje standard DDS (Data Distribution Service) do wymiany danych. Implementacje takie jak **Eclipse Cyclone DDS** pozwalają na wydajne przesyłanie komunikatów między różnymi węzłami systemu (Publisher/Subscriber) z wykorzystaniem mechanizmów takich jak DomainParticipant i silnego typowania danych przez IDL (Interface Definition Language).
- **Universal Scene Description (USD):** Jako fundament Isaac Sim, USD pozwala na nieniszczącą współpracę wielu twórców nad jedną sceną, oferując mechanizmy warstwowania (layering) i wariantów.