

章節:External controller tutorial

在 CoppeliaSim 中，有幾種方法可以控制機器人或仿真：

- 方法一是編寫一個子腳本來處理給定機器人或模型的行為。這是最方便的方法，因為子腳本直接附加到場景對象，它們將與相關的場景對象一起復制，它們不需要使用外部工具進行任何編譯，它們可以在線程或非線程模式下運行，它們可以通過自定義 Lua 函數或 Lua 擴展庫進行擴展。使用子腳本的另一個主要優點是：與本節中提到的後 3 種方法（即使用常規 API）一樣，沒有通信延遲，並且子腳本是應用程序主線程的一部分（固有的同步操作）。
- 方法二是編寫插件。插件機制允許使用回調機制，自定義 Lua 函數註冊，當然還可以訪問外部函數庫。插件通常與子腳本結合使用（例如，插件註冊自定義的 Lua 函數，當從子腳本中調用時，該 Lua 函數將回調特定的插件函數）。使用插件的主要優勢還在於，與本節中提到的後 3 種方法（即使用常規 API）一樣，沒有通信延遲，並且插件是應用程序主線程的一部分（固有的同步操作）。插件的缺點是：它們的編程更加複雜，並且也需要使用外部編譯。
- 方法三是編寫依賴於遠程 API 的外部客戶端應用程序。如果您需要從外部應用程序，機器人或另一台計算機運行控制代碼，這是一種非常便捷的方法。這也使您可以使用與運行真實機器人完全相同的代碼來

控制仿真或模型（例如虛擬機器人）。

- 方法四是通過 ROS 節點。ROS 與遠程 API 相似，是使多個分佈式進程相互通信的便捷方法。儘管遠程 API 非常輕巧且快速，但它僅允許與 CoppeliaSim 通信。另一方面，ROS 允許幾乎將任意數量的進程相互連接，並且提供了大量兼容的庫。但是，它比遠程 API 重並且更複雜。
- 方法五是通過 BlueZero (BØ) 節點。與 ROS 類似，BlueZero 是使多個分佈式進程相互通信的一種便捷方法，並且是一種輕量級的跨平台解決方案。
- 方法六是編寫一個外部應用程序，該應用程序通過各種方式（例如管道，套接字，串行端口等）與 CoppeliaSim 插件或 CoppeliaSim 腳本進行通信。選擇編程語言（可以是任何一種語言）和靈活性是兩個主要優點。同樣，控制代碼也可以在機器人或其他計算機上運行。

