

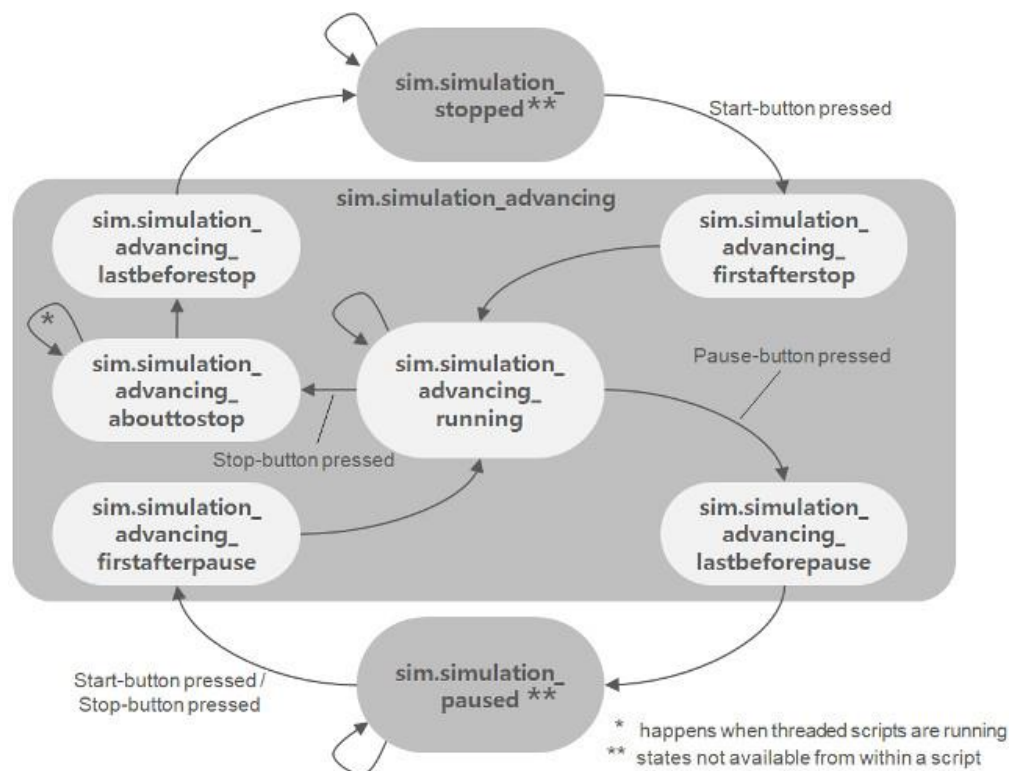
章節:Simulation

模擬

使用[菜單欄->模擬->開始/暫停/停止模擬]或通過相關的工具欄按鈕來啟動，暫停和停止 CoppeliaSim 中的模擬：



在內部，模擬器將使用其他中間狀態，以正確告知腳本或程序接下來將發生的情況。



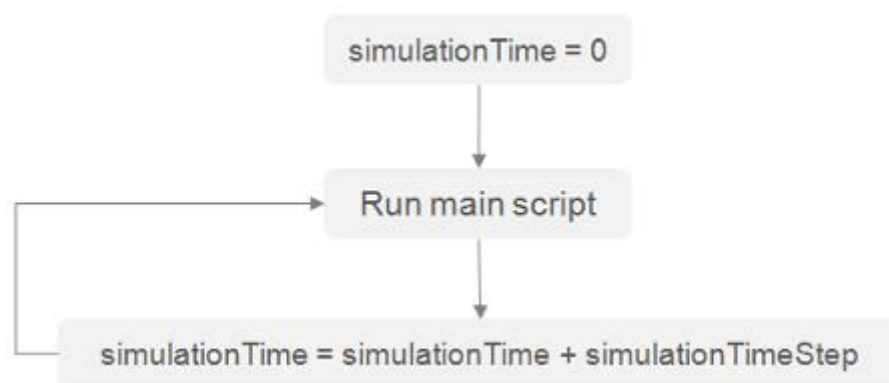
腳本和程序應始終根據當前系統調用功能以及可能的模擬狀態進行反應，

以便正確運行。優良作法是將每個控制代碼至少分為 4 個系統調用函數

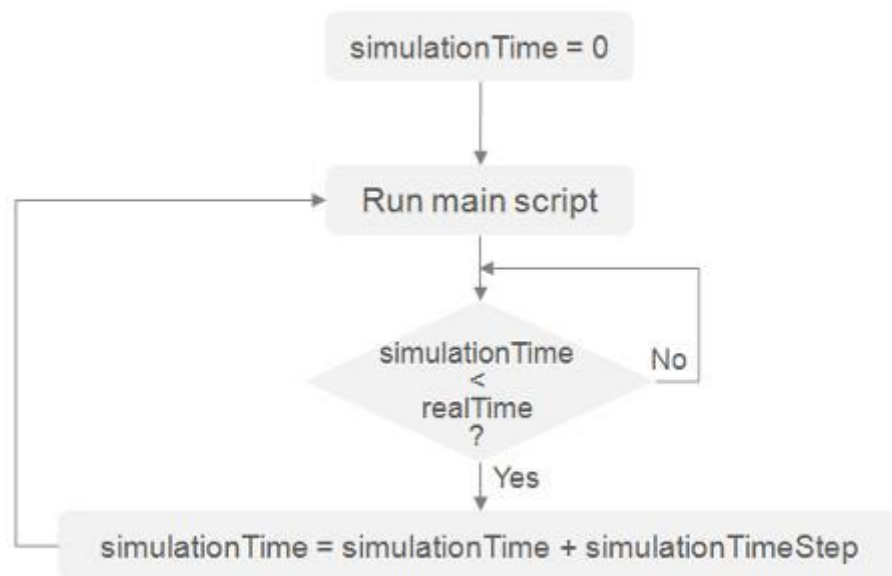
- **初始化函數**：僅在腳本初始化時調用該函數
- **激活函數**：應在發生激活時調用該函數。
- **傳感功能**：應在傳感發生時調用此函數。
- **清理函數**：該函數在腳本未初始化之前被調用。

仿真循環

模擬器通過以恆定的時間步長推進模擬時間來進行操作。



通過嘗試使仿真時間與實時保持同步來支持實時仿真：



以下是一個非常簡化的主客戶端應用程序:

```
void initializationCallback
{
    // do some initialization here
}

void loopCallback
{
    if ( (simGetSimulationState() & sim_simulation_advancing) != 0 )
    {
        if ( (simGetRealTimeSimulation() != 1) || (simIsRealTimeSimulationStepNeeded() == 1) )
        {
            if ( (simHandleMainScript() & sim_script_main_script_not_called) == 0 )
                simAdvanceSimulationByOneStep();
        }
    }
}

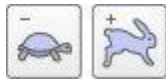
void deinitializationCallback
{
    // do some clean-up here
}
```

取決於模擬的複雜性，計算機的性能和模擬設置，實時模擬可能並不總是可能的。

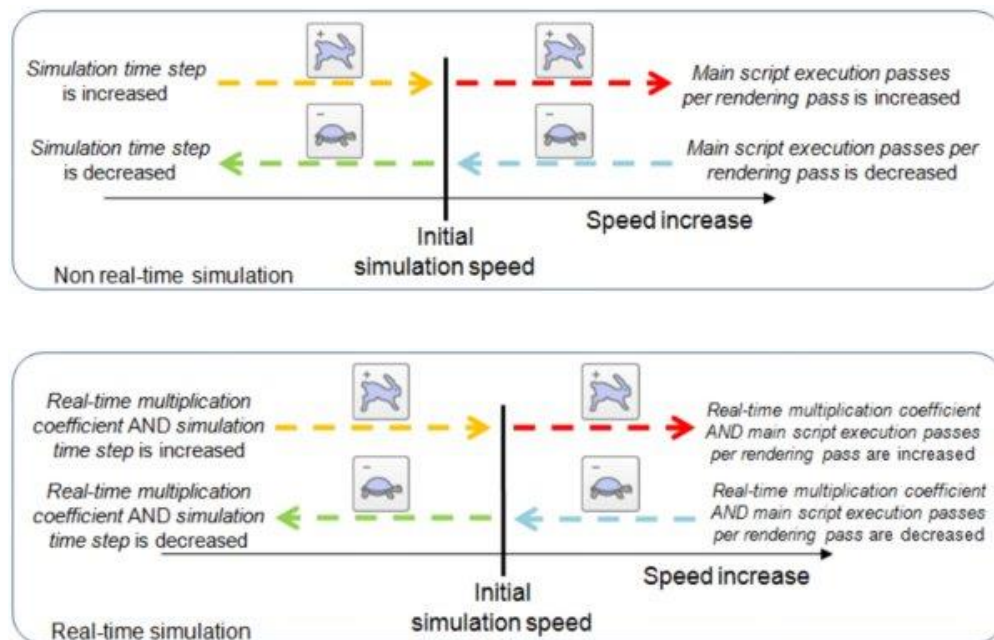
仿真速度

在非實時仿真中，仿真速度主要取決於兩個因素，仿真時間長和一個渲染通道的仿真通道數量。在實時仿真的情況下，仿真速度主要取決於實時乘法係數，而且在一定程度上取決於仿真時間步長。由於計算機的計算能力

有限，因此無法進行仿真。在模擬過程中，可以使用以下工具欄按鈕來調整模擬速度：



以某種方式調整模擬速度，以使初始模擬時間步長永遠不會增加。以下兩個圖說明了仿真速度調整機制：



默認情況下，每個模擬週期由以下順序操作組成：

- 執行主腳本
- 渲染場景

螺紋渲染

渲染操作將始終增加仿真週期的持續時間，從而也降低了仿真速度。可以定義每個場景渲染的主腳本執行次數，但這在某些情況下還不夠，因為渲染仍然會減慢每個第 x 個模擬週期的時間。在這種情況下，可以通過用戶設置或以下工具欄按鈕激活線程渲染模式：



激活線程渲染模式後，模擬週期將僅包含在執行主腳本中，因此模擬將以最大速度運行。渲染將通過不同的線程進行，並且不會減慢模擬任務的速度。然而，必須考慮缺點。

- 渲染將與模擬循環異步進行，並且可能會出現視覺故障。
- 錄像機將無法以恆定速度運行。
- 應用程序的穩定性可能會降低。
- 某些操作需要等待渲染線程完成工作才能執行，反之亦然。在那些情況下，循環可能比順序渲染模式花費更多的時間。