Ein Bild, das Reihe, Diagramm, Steigung, parallel enthält.

KI-generierte Inhalte können fehlerhaft sein.

Ein Bild, das Diagramm, Reihe, Text, parallel enthält.

KI-generierte Inhalte können fehlerhaft sein.

Ein Bild, das Text, Diagramm, Reihe, Screenshot enthält.

KI-generierte Inhalte können fehlerhaft sein.

**Projekt: TUC Drillbotics Trajectory Optimization Tool**

**1. Fragestellung**

Im Rahmen des internationalen Drillbotics-Wettbewerbs steht das TUC-Team vor der Herausforderung, eine bohrbare Trajektorie durch eine Reihe von 3D-Zielpunkten zu planen, die nicht geradlinig angeordnet sind. Diese Aufgabe erfordert die automatisierte Erstellung einer Flugbahn, die sowohl die Zielkoordinaten erreicht als auch die mechanischen und physikalischen Grenzen der Bohrlochgarnitur (BHA) berücksichtigt.

Das Kernproblem ist die Entwicklung eines systematischen Bewertungsrahmens zur quantitativen Bewertung und zum Vergleich verschiedener Algorithmen zur Trajektorienplanung.

**Ziel:** Entwurf und Implementierung eines **Tools zur Trajektorienoptimierung und -bewertung**, das die Ergebnisse verschiedener Planungsansätze quantitativ bewertet.

**2. Methode**

Um die optimale Flugbahn zu finden, wird **das Problem als Optimalsteuerproblem (OCP) formuliert** und mit **dem rockit-Toolkit in Python gelöst**, das auf casadi aufbaut.

* **Zustandsvariablen:** Die Trajektorie wird durch 6 Dimensionen definiert:
  + Position: x\_pos, y\_pos, z\_pos
  + Blickrichtung: azimuth, inclination
  + Zeit: t
* **Steuervariablen:** Die Änderungsraten der Winkel und die Bohrgeschwindigkeit (ROP) werden als Steuereingänge verwendet:
  + u\_azimuth\_rate\_ds: Änderungsrate des Azimuts in Bezug auf die Bogenlänge.
  + u\_inclination\_rate\_ds: Änderungsrate der Neigung in Bezug auf die Bogenlänge.
  + inverse\_rop: Der Kehrwert der Bohrgeschwindigkeit (1/Geschwindigkeit), der die Bohrzeit darstellt.
* **Zielfunktion:** Das Hauptziel ist die **Minimierung der gesamten Bohrzeit**. Dies wird durch die Minimierung des Integrals von inverse\_rop über die gesamte Bogenlänge erreicht.
* **Beschränkungen:**
  + **Anfangsbedingungen:** Start im Ursprung (0,0,0) mit einer geraden Ausrichtung nach unten.
  + **Endbedingungen:** Die Trajektorie muss den vordefinierten Zielpunkt erreichen ([5, 2, -30]).
  + **Pfadbeschränkungen:** Um die mechanische Machbarkeit zu gewährleisten, werden die maximal zulässige Krümmung (max\_bur = 0.05 rad/m) und die Bohrgeschwindigkeit (5 <= ROP <= 30 m/min) begrenzt. Die Neigung ist so eingeschränkt, dass nur nach unten gebohrt wird.

**3. Stand der Arbeit**

* **Implementierung:** Das Tool zur Lösung des OCP wurde erfolgreich in einem Jupyter Notebook implementiert. Alle wesentlichen Komponenten – Zustands- und Steuervariablen, Zielfunktion und Randbedingungen – sind definiert.
* **Lösungsversuch:** Der ipopt-Solver wurde zur Lösung des Problems eingesetzt. Die Solver-Ausgabe zeigt, dass das Problem formuliert und an den Solver übergeben wurde.
* **Aktuelle Herausforderung:**
  + Der Solver findet immer noch Schwierigkeiten die beste Lösung zu finden bei manschen Punkten.
* **Nächste Schritte:**
  + **Analyse der Solver-Ausgabe:** Die Zwischenergebnisse des Solvers müssen genauer untersucht werden, um die Ursache für die Konvergenzprobleme zu identifizieren.
  + **Anpassung der Parameter:** Die Anfangsschätzungen für die Variablen und die Problemformulierung (z.B. die Gewichtung der Zielfunktion oder die Skalierung der Variablen) müssen möglicherweise angepasst werden, um die Konvergenz zu verbessern.
  + **Visualisierung:** Trotz des Fehlers können die letzten vom Solver berechneten Werte verwendet werden, um eine vorläufige 3D-Trajektorie zu visualisieren und das Verhalten des Algorithmus zu verstehen.