Feedback control over constrained robotic systems through the Udvadia-Kalaba approach

February 13, 2024

Contents

1 Introduction **7**

2 Literature Review **10**

3 Methodology **11**

3.1 Essentials of Udvadia-Kalaba approach 11

4 Implementation **14**

5 Evaluation and Discussion **15**

6 Conclusion **16**

Bibliography cited **17**

Chapter 1

Introduction

The advancement of microelectronics and sensor technologies has catalyzed a widespread integration of robotic systems into various domains. Manipulators, delivery robots, and flying drones have become ubiquitous, presenting developers and researchers with novel challenges in terms of robustness, adaptiveness, and synchronization. Among these challenges, synchronization stands out as a critical issue, especially given the proliferation and increasing complexity of such systems. In real-world applications, synchronization of robotic systems is crucial across various industries. For instance, in manufacturing assembly lines, synchronized robots ensure smooth production flow and maximize throughput. Similarly, in warehouse logistics, coordinated robotic systems optimize order fulfillment times and enhance overall productivity. Furthermore, collaborative robotics scenarios in construction projects benefit from synchronized robotic systems for tasks like concrete pouring and steel beam placement, improving efficiency and minimizing delays. A common challenge in this realm involves effectively controlling robots constrained by a shared object. This paper proposes a straightforward methodology to address such problems through the Udvadia-Kalaba[1] approach.

This proposed methodology gains particular significance within the context of modern physics simulation, derivation, and optimization libraries. These tools offer substantial computational speed, enabling efficient problem-solving. In this article, we leverage MuJoCo[2], Pinocchio[3], and ProxSuite[4] for simulation, robotic dynamics computation, and optimization, respectively. Pinocchio, in particular, emerges as a key tool for computing the dynamics of robotic systems. However, its limitation to open-loop physics models poses challenges for controlling and synchronizing multiple robots.

Previous solutions have often involved constructing models with pre-existing constraints or employing the KKT (Karush-Kuhn-Tucker) approach. However, these methods suffer from computational complexity and issues with constraint prioritization. The proposed methodology combines the strengths of both approaches, integrating physical grounding from the former and simplicity of application from the latter. Notably, it allows for manual fine-tuning of constraint priorities and boasts superior computational speed through auto code generation.

The implementation of the proposed methodology demonstrates significant advancements in the control and synchronization of robotic systems constrained by shared objects. Through the integration of robust physics simulation, precise robotic dynamics computation, and efficient optimization techniques, the method achieves remarkable outcomes across various simulation experiments. By leveraging advanced simulation, computation, and optimization techniques, the method has improved levels of efficiency, accuracy, and adaptability in robotic operations, paving the way for further advancements in automation and robotics technology.

In subsequent chapters, we delve deeper into various aspects of the proposed method. Chapter 2 provides an exhaustive review of recent literature, highlighting the existing landscape of solutions and their limitations. Chapter 3 elucidates the methodology underlying the proposed approach, offering insights into its theoretical underpinnings. Implementation details and code snippets are presented in Chapter 4, demonstrating the practical application of the method. Chapter 5 undertakes a comparative analysis, pitting technique against established methods to gauge its efficacy. Finally, Chapter 6 summarizes findings, discusses implications, and outlines avenues for future research.

Chapter 2

Literature Review

Chapter 3

Methodology

This chapter will present a detailed elucidation of the principles underpinning the proposed methodology. The initial section will offer a concise overview of the Udvadia-Kalaba's approach is accompanied by a comprehensive exposition of the physical rationales employed in this technique. Subsequently, these principles will be harnessed in the development of the intended methodology. Furthermore, this section will scrutinize the limitations and distinctive characteristics inherent in the proposed methodology.

3.1 Essentials of Udvadia-Kalaba approach

Let $\mathbf{q}$ be a vector of generalized coordinates of some physical system. Also let agree that $\dim \mathbf{q} = n_q$ and $\dim \dot{\mathbf{q}} = \dim \ddot{\mathbf{q}} = n_v$, where $\dot{\mathbf{q}}$ and $\ddot{\mathbf{q}}$ are generalized velocity and acceleration respectevly. Notably, that in the general case $n_q \neq n_v$ because $\dot{\mathbf{q}}$ and $\ddot{\mathbf{q}}$ can be located in different spaces. Thus, the dynamics of the given system can be expressed:

$$M(\mathbf{q})\ddot{\mathbf{q}} + C(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + g(\mathbf{q}) = \boldsymbol{\tau} \tag{3.1}$$

where $M(\mathbf{q}) \succeq 0$ is known as general inertia matrix, $C(\mathbf{q}, \dot{\mathbf{q}})$ is Coriolis-Centrifugal matrix, and $g(\mathbf{q})$ is potential forces impact (usually gravitational impact). For further convenience, I would omit the parameters of matrix functions. In the case of Coriolis-Centrifugal and potential forces, it is common to combine them in one term $\mathbf{Q}(\mathbf{q}, \dot{\mathbf{q}}) = \boldsymbol{\tau} - C(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} - g(\mathbf{q})$. Hence, I can rewrite 3.1 in the following manner

$$M\ddot{\mathbf{q}} = \mathbf{Q} \tag{3.2}$$

equation 3.2 is compact form of equation 3.1. It would be highly utilized in further investigations.

Let's consider the same physical system again with imposed constraints at this time. All holonomic constraints would be presented by $\varphi(\mathbf{q}, t) : \mathbb{S}_{n_q} \times \mathbb{R} \to \mathbb{R}^m$, and all non-holonomic by $\psi(\mathbf{q}, \dot{\mathbf{q}}, t) : \mathbb{S}_{n_q} \times \mathbb{R}^{n_v} \times \mathbb{R} \to \mathbb{R}^k$, where $\mathbb{S}_{n_q}$ generalized coordinates space in the most common case. All constraints are satisfied if and only if the above functions are equal to zero.

These constraints can be rewritten in the form $A(\mathbf{q}, \dot{\mathbf{q}}, t)\ddot{\mathbf{q}} - b(\mathbf{q}, \dot{\mathbf{q}}, t)$ by differentiation of holonomic constraints twice and non-holonomic once. According to Udwadia-Kalaba the constrained force that would satisfy can be written

$$Q_c = M^{1/2}(AM^{-1/2})^{+}(b - AM^{-1}Q) \tag{3.3}$$

where $M^{\pm 1/2} = W\Lambda^{\pm 1/2}W^T$ (W, Λ gain by eigendecomposition), $[*]^{+}$ is the Moore-Penrose inverse. Under these forces, the solution of the forward dynamics takes the form

$$\ddot{\mathrm{q}} = M^{-1}Q + M^{-1/2}(AM^{-1/2})^{+}(b - AM^{-1}Q) \tag{3.4}$$

The equation 3.3 according to F. Udvadia and R. Kalaba [1] is the analytical solution to the minimization problem based on Gauss's principle of least constraint

$$\begin{aligned} \min_{\ddot{\mathrm{q}}} \quad & [\ddot{\mathrm{q}} - a]^T M [\ddot{\mathrm{q}} - a] \\ \text{s.t.} \quad & A\ddot{\mathrm{q}} - b = 0 \\ & a(\mathrm{q}, \dot{\mathrm{q}}, t) = M^{-1}Q \end{aligned} \tag{3.5}$$

While the solution proposed by F. Udvadia and R. Kalaba demonstrates precision, it is not without its drawbacks. Instability points and computationally intensive operations, such as the computation of matrix roots, present significant challenges. Moreover, a notable deficiency lies in the absence of prioritization within the methodology.

Chapter 4

Implementation

Chapter 5

Evaluation and Discussion

Chapter 6

Conclusion

Bibliography cited

[1] F. Udwadia and R. Kalaba, “A new perspective on constrained motion,” *Proceedings of the Royal Society of London. Series A: Mathematical and Physical Sciences*, vol. 439, no. 1906, pp. 407–410, Nov. 1992. DOI: 10.1098/rspa.1992.0158.

[2] E. Todorov, T. Erez, and Y. Tassa, “Mujoco: A physics engine for model-based control,” *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 5026–5033, Oct. 2012. DOI: 10.1109/IROS.2012.6386109.

[3] J. Carpentier, G. Saurel, G. Buondonno, *et al.*, “The Pinocchio C++ library – A fast and flexible implementation of rigid body dynamics algorithms and their analytical derivatives,” *International Symposium on System Integration SII*, Jan. 2019. DOI: 10.1109/SII.2019.8700380.

[4] S. Diamond and S. Boyd, “CVXPY: A Python-embedded modeling language for convex optimization,” *Journal of Machine Learning Research*, vol. 17, no. 83, pp. 1–5, 2016.