Investigation of Energy Transfer in Coupled Electro-mechanical Oscillators using Parallel Computing

Kairvi Lodhiya

Department of Mechanical, Aerospace, and Nuclear Engineering Rensselaer Polytechnic Institute, 110 8th St. Troy, NY 12180

Abstract

This work aims to investigate the energy exchange mechanisms within an idealized conservative system, with the goal of maximizing mechanical-to-electrical energy conversion. The system under study consists of a spring-mass mechanical oscillator coupled to an electrical LC circuit via a variable capacitor, where the moving electrode serves as the mechanical mass. To solve the coupled non-dimensional ordinary differential equations (ODEs), two numerical methods were implemented and evaluated. First, a self-developed classical fourth-order Runge-Kutta (RK4) solver was designed to provide a baseline solution, offering good computational efficiency and moderate accuracy for the system evolution. Second, the open-source SUNDIALS (Suite of Nonlinear and Differential/Algebraic Equation Solvers) library was employed, specifically utilizing the CVODE solver for initial value problems. Third, the combination of advanced numerical integration, custom solver benchmarking, efficient parallel data handling using ADIOS2, and flexible post-processing constituted the complete solution strategy for investigating the energy transfer in the coupled electromechanical oscillators system.

Keywords: Electrostatic Energy Harvesting, Coupled Electromechanical Oscillators, Nonlinear Dynamics, Ordinary Differential Equations (ODEs), Parallel Computing, Adaptive Time-Stepping, SUNDIALS, Runge-Kutta Methods, ADIOS2, Python Visualization

Developers Repository:

Link for the repo: `https://github.com/KairviLodhiya/Project-for-computing-at-scale`

Licensing Provision

BSD-3 clause

Programming language

C++

Research Problem

The Internet of Things (IoT) has become integral to modern technology. As IoT devices proliferate, identifying a sustainable power source becomes crucial [1]. Ambient energy harvesting offers a promising solution by enabling autonomous operation and reducing environmental impacts associated with battery disposal. Among ambient sources, mechanical vibrations are particularly attractive for powering low-power sensors [2].

Mechanical vibrations can be converted into electrical energy via electromagnetic, piezoelectric, or electrostatic methods. While electromagnetic and piezoelectric techniques are mature and commercially deployed, electrostatic harvesters remain under investigation. Their compatibility with micro-electro-mechanical systems (MEMS) fabrication offers the potential for cost-effective production [3]. However, their energy output currently falls short of application requirements.

In electrostatic converters, vibrational energy is harvested by modulating a variable capacitor charged and discharged in synchronization with external vibrations [4]. The relative motion between capacitor electrodes alters capacitance, enabling mechanical work against electrostatic forces and facilitating energy transfer [5].

Despite advances, a fundamental understanding of energy exchange between mechanical and electrical domains remains limited. This work aims to systematically explore the energy transfer mechanisms within an idealized conservative system, with the goal of maximizing mechanical-to-electrical energy conversion.

The system under study consists of a spring-mass mechanical oscillator coupled to an electrical LC circuit via a variable capacitor, where the moving electrode serves as the mechanical mass. The model, depicted in Figure 1, is idealized and conservative, reflecting core principles of electrostatic energy harvesters.

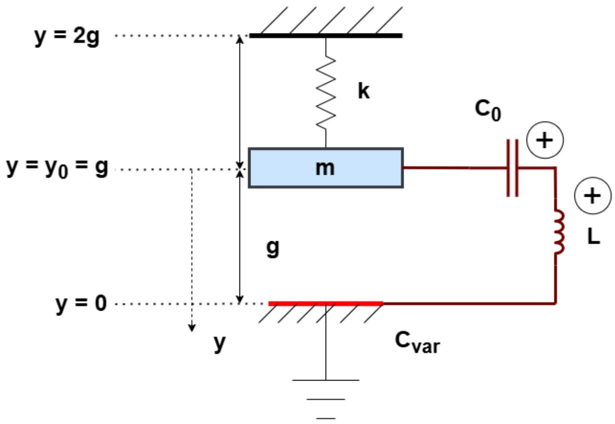

Figure 1: Schematic of the coupled electro-mechanical oscillator under study.

The mechanical oscillator consists of a mass M connected to a rigid frame by a linear spring of stiffness k. It is coupled to an electrical oscillator composed of an inductance L and a fixed capacitance C_0, with the variable capacitor C_{var} acting as the coupling element. The motion of mass M modulates C_{var}, facilitating bidirectional energy transfer.

Applying Newton's second law and Kirchhoff's voltage law yields the governing equations:

$$m\ddot{y} + ky + \frac{1}{2}\frac{q^2}{\varepsilon A} = kg \tag{1}$$

$$L\ddot{q} + \frac{q}{C_0} + \frac{qy}{\varepsilon A} = 0 \tag{2}$$

Here, y represents the relative displacement of mass m from its initial position y_0, taken as the origin. The nominal electrode gap is g, q is the instantaneous charge, ε is the dielectric permittivity, and A is the area of the capacitor plate. The non-dimensional form of the governing equations for the coupled mechanical-electrical system is given by:

$$\frac{d^2y^*}{dt^{*2}} + y^* + \frac{1}{4}\frac{E_{c,\max}}{E_{m,n}}q^{*2} = \frac{1}{2} \tag{3}$$

$$\frac{d^2q^*}{dt^{*2}} + \frac{\omega_0^2}{\omega_m^2}q^* + 2\frac{\omega_{n,c}^2}{\omega_m^2}q^*y^* = 0 \tag{4}$$

Here, y^* and q^* represent the non-dimensional displacement and non-dimensional charge, respectively, and t^* is the non-dimensional time. Their first derivatives with respect to time, $y'^* = \frac{dy^*}{dt^*}$ and $q'^* = \frac{dq^*}{dt^*}$, correspond to the non-dimensional velocity and current in the mechanical and electrical domains, respectively. These four variables - y^*, y'^*, q^*, and q'^* - must evolve simultaneously during the solution process to fully characterize the dynamics of the system.

Solving these two coupled, non-linear, second-order differential equations is computationally demanding due to the complexity of the system dynamics. The state of the system at any given instant is not determined solely by the instantaneous values of the displacement y^* and charge q^*, but also critically depends on their time derivatives, $y'^* = \frac{dy^*}{dt^*}$ and $q'^* = \frac{dq^*}{dt^*}$, which represent the non-dimensional velocity and current, respectively.

Solving these two coupled, non-linear, second-order differential equations is computationally demanding due to the complexity of the system dynamics. The state of the system at any given instant is not determined solely by the instantaneous values of the displacement y^* and charge q^*, but also critically depends on their time derivatives, $y'^* = \frac{dy^*}{dt^*}$ and $q'^* = \frac{dq^*}{dt^*}$, which represent the non-dimensional velocity and current, respectively.

Thus, a complete description of the system requires simultaneous tracking of four interdependent variables: y^*, y'^*, q^*, and q'^*. These variables evolve continuously with time, and their interactions define the mechanical-electrical energy exchange within the system. Since y^* and q^* are governed by second-order ordinary differential equations (ODEs), they must first be reformulated as a system of first-order ODEs to enable numerical integration. This involves treating each original second-order equation as two coupled first-order equations, effectively doubling the dimensionality of the system.

As a result, the numerical solution process becomes substantially more intensive, requiring careful synchronization of the evolution of all four variables while maintaining numerical stability and accuracy over a potentially large number of time steps. The coupled and non-linear nature of the equations further increases the computational burden, as small changes in one variable can produce significant nonlinear effects on the others, necessitating fine-time discretization and dense data management.

Solution Approach

To solve the coupled non-dimensional ordinary differential equations (ODEs) presented in Equations (3) and (4), two numerical methods were implemented and evaluated.

First, a self-developed classical fourth-order Runge-Kutta (RK4) solver was created. This fixed-step RK4 method was designed to provide a baseline solution, offering good computational efficiency and moderate accuracy for the system evolution. The RK4 method discretizes the ODEs over small, constant time increments and iteratively updates the system states y^*, y'^*, q^*, and q'^*.

In parallel, the open-source SUNDIALS (Suite of Nonlinear and Differential/Algebraic Equation Solvers) library was employed, specifically utilizing the CVODE solver for initial value problems. CVODE offers advanced time-integration techniques, including adaptive time-stepping and higher-order variable-coefficient methods, which are particularly effective for stiff and nonlinear systems [6]. Adaptive time-stepping allowed the solver to dynamically adjust the time step based on local error estimates, significantly improving the accuracy and stability of the solution, especially during regions of rapid system variation.

Through comparative evaluation, the SUNDIALS-based solution was found to deliver superior accuracy compared to the self-coded RK4 implementation. The built-in adaptivity and robust error control mechanisms of CVODE contributed to more reliable resolution of the coupled system dynamics, particularly during transient phases where fixed-step RK4 struggled to maintain precision without excessively small time steps.

RK4 was mainly used to implement Kokkos as SUNDIALS is not very friendly with GPU parallelization [7]. RK4 (GPU version) worked fairly well with Kokkos parallelization. This will help implementation on parallel computers to fasten up the process and handle longer data production and management.

To manage the large volume of data generated during the simulation, the ADIOS (Adaptable IO System) library was integrated for efficient parallel data storage [8]. ADIOS enabled scalable and high-throughput writing of solution snapshots to disk during the simulation runtime, minimizing I/O bottlenecks and ensuring the reproducibility of computational results.

Subsequent data visualization and analysis were performed using Python, leveraging libraries such as NumPy, Matplotlib, and SciPy. Python's flexibility allowed for detailed post-processing, including trajectory plotting, energy trend analysis, and frequency estimation, facilitating deeper insights into the system behavior [9].

Thus, the combination of advanced numerical integration (via SUNDIALS), custom solver benchmarking (via RK4), efficient parallel data handling (via ADIOS), and flexible post-processing (via Python) constituted the complete solution strategy for investigating the energy exchange in the coupled electromechanical oscillator system.

The implementation of all these codes were tested using Catch2 technique of testing [10].

Finally the execution time for each of the executables was:

	SUNDIALS - CPU	RK4 - CPU	RK4 - GPU	MATLAB
Simulation time (s)	3.34742	5.66233	2.2467	15.9758
Writing time (s)	1.19959	1.83962	2.17343	1.0990
Total time (s)	4.54702	7.50195	4.42013	17.0760

Table 1: Performance comparison of different simulation methods.

References

[1] X. Liu and N. Ansari, "Toward Green IoT: Energy Solutions and Key Challenges," *IEEE Communications Magazine*, vol. 57, no. 3, pp. 104–110, 2019.
`https://doi.org/10.1109/MCOM.2019.1800175`

[2] J. Li, X. Tong, J. Oxaal, Z. Liu, M. Hella, and D.-A. Borca-Tasciuc, "Investigation of Parallel-Connected MEMS Electrostatic Energy Harvesters for Enhancing Output Power over a Wide Frequency Range," *Journal of Micromechanics and Microengineering*, vol. 29, no. 9, p. 094001, 2019.

[3] F. U. Khan and M. U. Qadir, "State-of-the-Art in Vibration-Based Electrostatic Energy Harvesting," *Journal of Micromechanics and Microengineering*, vol. 26, no. 10, p. 103001, 2016.
`https://doi.org/10.1088/0960-1317/26/10/103001`

[4] M. El-Hami, P. Glynne-Jones, N. M. White, M. Hill, S. Beeby, E. James, A. D. Brown, and J. N. Ross, "Design and Fabrication of a New Vibration-Based Electromechanical Power Generator," *Sensors and Actuators A: Physical*, vol. 92, no. 1–3, pp. 335–342, 2001.
`https://doi.org/10.1016/S0924-4247(01)00569-6`

[5] S. Boisseau, G. Despesse, and B. A. Seddik, "Electrostatic Conversion for Vibration Energy Harvesting," in *Small-Scale Energy Harvesting*, edited by M. Lallart, InTech, 2012.

[6] A. C. Hindmarsh, P. N. Brown, K. E. Grant, S. L. Lee, R. Serban, D. E. Shumaker, and C. S. Woodward, "SUNDIALS: Suite of Nonlinear and Differential/Algebraic Equation Solvers," *ACM Transactions on Mathematical Software (TOMS)*, vol. 31, no. 3, pp. 363–396, 2005.
`https://doi.org/10.1145/1089014.1089020`

[7] Trott, C. R., Sunderland, D., Adelstein-Lelbach, B., Finkel, H., Heroux, M. A. (2022). Kokkos 3: Programming Model Extensions for the Exascale Era. *In Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis (SC '22)*. IEEE Press.
`https://doi.org/10.5555/3571885.3571945`

[8] K. S. Ibrahim, J. Liu, A. S. Bland, Q. Liu, and M. Wolf, "ADIOS 2: The Adaptable Input Output System," in *Proceedings of the 2018 IEEE International Conference on Big Data (Big Data)*, pp. 473–482, 2018.
`https://doi.org/10.1109/BigData.2018.8622142`

[9] G. van Rossum and F. L. Drake (eds.), *Python 3 Reference Manual*, Python Software Foundation, 2009.
`https://www.python.org/doc/`

[10] Catch2 Authors, "Catch2: A Modern, C++-Native, Header-Only, Test Framework for Unit-Tests, TDD, and BDD," GitHub repository, 2024.
`https://github.com/catchorg/Catch2`