2026 ASEE ANNUAL CONFERENCE June 21–24, 2026 | Charlotte Convention Center, NC
Host Sponsor UNC Charlotte
Paper ID #50757

Learning Principal Component Analysis through an Embodied Classroom Experience

Dr. Abbas Attarwala, California State University, Chico

I am currently serving as an Associate Professor in Computer Science at California State University, Chico. With 14 years of extensive teaching experience in the field of Computer Science, I have taught at both the University of Toronto and Boston University. My passion for teaching and utilizing technology in the classroom has been recognized with the prestigious Gerald and Deanne Gitner Family Award for Innovation in Teaching with Technology, which I received in 2020 at Boston University. I received the International Wildcat Outstanding Faculty of 2022-23 at California State University, Chico for my teaching.

Prof. Jaime Raigoza, California State University, Chico

Learning Principal Component Analysis through an Embodied Classroom Experience

Abstract

In computer science education, eigenvectors and eigenvalues are typically introduced in abstract linear algebra contexts and later reinforced through practical data analysis tasks, including Principal Component Analysis (PCA) applications such as image compression. While such PCA based activities correctly present eigenvectors as a coordinate system for re expressing data, students frequently experience them as abstract mathematical artifacts rather than as representations whose structure can be justified through physical experience. In this experience paper, we reflect on an instructional sequence implemented in a summer linear algebra course with a small student cohort, using LEGO Education SPIKE to reposition eigenvectors as learned coordinate systems that validate, rather than replace, student intuition.

Students first collect distance sensor data from a mobile robot interacting with a physical environment and apply PCA to uncover dominant and secondary patterns of variation. The resulting eigenvectors align with quantities students already find meaningful, such as overall distance and left right imbalance, yet emerge without being specified in advance. In a second phase, the robot interprets new sensor readings using these learned eigencoordinates, transforming raw measurements into action relevant quantities governing forward motion and rotation.

From an instructor's perspective, the most powerful learning moments occur when students recognize that PCA does not produce something surprising but rather justifies what they intuitively expected through a principled mathematical process. We argue that this embodied two phase use of PCA helps students understand eigenvectors as meaningful coordinate systems grounded in experience, rather than as arbitrary geometric constructs.

1 Introduction

Eigenvectors and eigenvalues are foundational concepts in computer science and engineering, appearing prominently in signal processing algorithms, machine learning, and in the analysis and control of dynamical systems [1, 2]. However many students struggle to explain what eigenvectors *mean* beyond formal definitions and computation steps [3, 4]. In our own teaching, we repeatedly see students who can carry out eigenvector and eigenvalue calculations correctly but still experience eigenvectors as arbitrary geometric objects whose relevance is confined to textbook exercises.

In instructional settings, eigenvectors are frequently re-visitted through PCA, often using static data analysis tasks such as image compression [5, 6]. Image compression using PCA effectively demonstrates variance, dimensionality reduction, and reconstruction error and correctly presents eigenvectors as an orthogonal coordinate system learned from the data. However, these activities often emphasize a retrospective use of representation: eigenvectors are learned from a fixed dataset and then used to re-express that same dataset after the fact. As a result, students can complete the activity without developing an intuition for why the learned directions should be meaningful or what it means to learn a representation from experience.

We introduce an alternative instructional framework for PCA, inspired by robotics and interactive control, that emphasizes the prospective use of learned representations. In this setting, students learn a coordinate system from past experience and then use it to interpret new sensor readings in real time as the robot moves. A change of basis is therefore not merely a descriptive tool applied after the fact; it determines which aspects of the world become legible to the system and which actions are natural or stable. This experience paper reflects on an instructional sequence designed to make this distinction tangible using LEGO Education SPIKE. Students work with a simple mobile robot that produces paired distance readings (d_L, d_R) through embodied interaction with a physical environment, apply PCA to learn an orthogonal basis from that experience, and then use the resulting eigencoordinates as action relevant quantities governing `speed` and `steering`.

The remainder of the paper details how the activity is staged and taught. Section 2 presents related work. Section 3 presents the classroom implementation, including data collection, PCA computation, and robot control. Section 4 addresses practical considerations and sensor robustness for the LEGO Education SPIKE platform. Section 5 discusses key pedagogical points and Section 6 concludes.

2 Literature Review

Teaching eigenvectors and PCA has inspired a wide range of pedagogical approaches aimed at reducing abstraction and improving conceptual understanding. Previous work spans interactive visualization tools, hands-on, and context-based projects, robotics-supported mathematics education, and the integration of computational tools in undergraduate linear algebra curricula. Together, these efforts seek to help students interpret eigenvectors as meaningful structures rather than purely algebraic artifacts.

A common strategy in this literature is the use of visual representations to connect algebraic definitions with geometric intuition. Dynamic geometry software and plotting tools allow students to explore how eigenvectors orient within linear transformations and how variance aligns with principal components. For example, Aytekin and Kıymaz used GeoGebra to help students visualize linear independence and spanning sets in low-dimensional spaces, reporting improved connections between formal definitions and geometric meaning [7]. In PCA instruction, similar visualization-based approaches employ scatter plots, animations, and interactive applets to show how data variance is captured along the principal axes. Across these studies, visualization has been consistently shown to lower cognitive barriers and provide a foundation for understanding eigenvectors as directions that organize structure in data.

Beyond screen-based visualization, researchers have introduced hands-on labs and application-driven projects to further ground the meaning of eigenvectors and PCA. Anderson and McCusker developed a physical lab in which students built a spring-coupled pendulum system and analyzed its vibration modes using eigenvalue methods, allowing eigenvectors to be experienced as physical modes of motion [8]. Other context-based interventions embed linear algebra into real-world modeling tasks, such as population migration analysis using Markov models. Other work has demonstrated the effectiveness of authentic open-ended linear algebra projects, ranging from population migration modeling [9] to data analytics pipelines centered on PCA and dimensionality reduction [10]. These studies report increased motivation and evidence of improved conceptual engagement, with students interpreting eigenvectors as equilibrium states, dominant patterns, or data features rather than as purely symbolic computations.

A parallel line of work has focused on integrating computation and programming into undergraduate linear algebra courses. Recent curriculum reforms emphasize the implementation of algorithms such as PCA on authentic datasets rather than relying solely on hand calculation and proof. Silva et al. reported that a computationally focused engineering linear algebra course allowed students to engage with realistic data analysis tasks, including image compression and exploratory PCA, earlier in their studies [11]. Li described a complementary approach using MATLAB and automated feedback to promote experimentation with eigendecomposition and basis changes [12]. Consistent with these findings, Chen et al. showed that students in computationally enriched linear algebra courses performed better in subsequent numerically intensive courses [13]. Together, these efforts reflect a broader trend in engineering and computer science education toward blending linear algebra with computation, data analysis, and context-rich applications.

Robotics has also been widely used as a context for mathematics education, particularly at the K–12 level, where embodied interaction and collaboration can make abstract ideas more tangible. Previous work has shown that building and programming robots can improve engagement, support multiple representations, and highlight the gap between ideal mathematical models and real-world execution [14]. However, most robotics-based mathematics interventions focus on foundational topics such as arithmetic, geometry, or proportional reasoning, rather than advanced linear algebra concepts. As a result, while robotics is well established as an effective medium for hands-on mathematical learning, its use to teach eigenvectors and PCA in undergraduate classrooms remains largely unexplored.

Despite this broad body of pedagogical work, we are not aware of classroom instructional approaches that use physical robotics as a unified platform for learning both an eigenspace from data and subsequently enacting behavior through that learned representation. The central novelty of our work lies in this dual use of embodiment. A low-cost LEGO Education SPIKE robot serves first as a data-generating system, producing sensor measurements whose covariance structure can be analyzed, and then as an acting system whose behavior is governed by the learned eigencoordinates. By coupling PCA computation with physical actuation, the approach creates an immediate feedback loop between mathematical representation and observable behavior. This allows eigenvectors to be experienced not only as descriptive directions in data, but as coordinate systems that directly shape perception and action, distinguishing the approach from prior methods that emphasize visualization or post hoc data analysis without embodied consequence.

3 Classroom Implementation and Pedagogical Flow

This section describes how the activity is implemented in the classroom, with particular attention to the deliberate sequencing of student intuition, embodied interaction, and formal linear algebra. The instructional design is informed by Vygotsky's theory of learning [15], in which understanding is first constructed through social interaction and mediated activity and only later internalized through formal abstraction. Accordingly, the activity is staged in three steps: students first perceive structure in the data through shared observation and discussion, then give that structure a name using linear algebraic language, and only later compute it formally. This sequencing allows PCA to function as an explanatory tool rather than as a black-box algorithm.

Overview of the Two-Phase Learning Design:

The activity begins with students building a simple mobile robot using LEGO Education SPIKE components. This construction step is intentional: by assembling the robot themselves, students develop immediate ownership over the system that will later generate the data they analyze. The use of LEGO reduces the barrier to entry, requiring no specialized equipment or prior robotics experience, while the collaborative nature of the build emphasizes shared problem-solving rather than instructor-led explanation.

Once assembled, the robot is equipped with two forward-facing distance sensors positioned to measure the left and right sides of the environment. In the first phase of the activity, the students drive the robot through a physical space and collect paired distance readings (d_L, d_R) as it interacts with its surroundings. At this stage, the data are neither interpreted nor controlled; the instructional goal is for students to experience these measurements as raw sensor observations that reflect what the robot encounters in the world.

This first phase plays a critical pedagogical role. Rather than starting with predefined features or control variables, students treat the collected sensor data as an experience record from which to learn the structure. Applying PCA to this data set produces an orthogonal coordinate system whose axes are not specified in advance but emerge from patterns of variation present in the robot's interaction with the environment. Students discover that the learned eigenvectors align with quantities they already find meaningful: one corresponds to overall distance, while the other captures the left–right imbalance. Importantly, these interpretations arise only after the data have been analyzed, reinforcing the idea that the representation is learned from experience rather than imposed by design.

In the second phase of the activity, the role of the learned representation changes fundamentally. The robot is again driven through the environment, but no longer acts directly on the raw sensor readings (d_L, d_R). Instead, new sensor measurements are projected into the learned eigenspace, and the resulting coordinates are used as action-relevant variables that govern forward motion and steering correction. The robot now navigates by attempting to maintain or return to the dominant patterns identified during the first phase, effectively perceiving and acting through the learned coordinate system.

This two-phase structure is central to the learning objective. Students see that PCA does not merely re-express data after the fact, but produces a coordinate system that can be used prospectively to guide behavior. Eigenvectors are no longer experienced as abstract geometric objects, but as meaningful directions that determine how the robot interprets its environment and how it moves within it. In this way, the activity culminates in a concrete understanding of PCA as a mechanism for discovering and acting upon the structure present in physical experience.

From a learning-theoretic perspective, this sequence is designed to support the' movement of students through Vygotsky's Zone of Proximal Development (ZPD), defined as the gap between what learners can do independently and what they can achieve with appropriate social support and mediation. The early phases emphasize collaborative construction, shared prediction, and embodied exploration, placing students in situations where intuitive reasoning is accessible but incomplete. The instructor's role during these phases is to mediate the discussion and pose guiding questions rather than providing formal explanations. Only after students encounter limitations in their informal reasoning is formal linear algebra introduced, allowing eigenvectors and eigenvalues to be internalized as sense-making tools that resolve problems students have already encountered through experience.

Classroom Setup and Group Structure:

Students work in teams of four per LEGO Education SPIKE robot, allowing all group members to participate directly while supporting discussion and shared decision-making. Each team builds a simple two-wheeled mobile robot using the LEGO Education SPIKE kit. The robot is controlled by the SPIKE Prime hub, which serves as the central processing unit and executes student-written programs in Python. Figure 1 shows one such mobile robot built by students. Two LEGO distance sensors are mounted symmetrically on the left and right sides of the front bumper at the same height, facing forward, and connected directly to the hub. These sensors estimate distance by emitting a pulse and measuring its return time, providing real-time proximity readings that serve as the robot's primary sensory input. While a reference build is provided, students are explicitly encouraged to experiment with alternative or playful designs, provided the symmetry constraint is maintained. Before any linear algebra is introduced, students explore the sensors using either the Scratch-based SPIKE environment or Python. This exploratory phase allows students to observe sensor noise, range limitations, and asymmetries, and to develop intuition about how physical placement affects readings through collaborative experimentation and discussion. The combination of a programmable hub, modular construction, and readily accessible sensors allows students to modify both the physical configuration and the control logic, grounding the final eigenvector-based control logic implemented in Python in lived, embodied experience.

Round 1: Data Collection, Prediction, and Emergence of Structure:

At the start of Round 1, students position their robot directly facing a large flat wall and program it to drive forward at a constant, moderate speed. Before running the program, students are asked to predict how distance sensor readings (d_L, d_R) will evolve over time and to justify their predictions within their groups.

Many students initially expected the two readings to be identical. When students run the robot

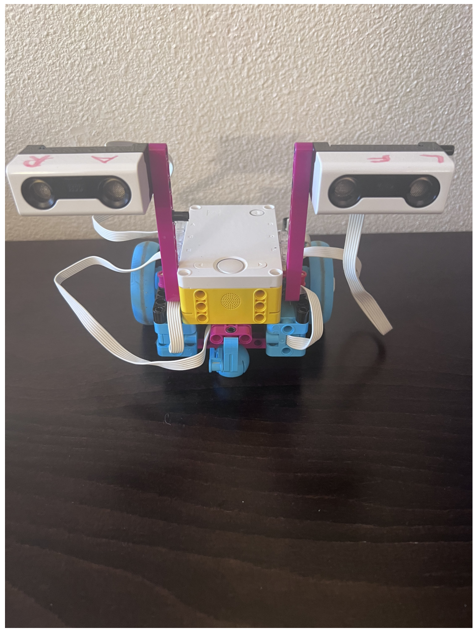

Figure 1: LEGO Education SPIKE mobile robot used in the activity, featuring a two-wheeled differential-drive base and two symmetrically mounted front-facing distance sensors.

and record the sensor values, they immediately observe that while the two signals are strongly correlated, they are not equal. These discrepancies prompt group discussion and revision of initial hypotheses, with students attributing the differences to sensor noise, mounting asymmetry, or slight angular offsets. At this stage, the instructor mediates the discussion by asking questions—for example, about the difference between correlation and equality—rather than providing explanations. Through this guided discussion, the distinction between *correlated* and *equal* becomes a shared conceptual reference point that is revisited throughout the activity.

As students examine their collected data, they begin to reason about which combinations of sensor readings appear to matter the most. Through observation and discussion, they identify the sum $d_L + d_R$ as capturing the overall distance to the wall, which decreases steadily as the robot approaches, and the difference $d_R - d_L$ as capturing the left–right imbalance, which remains near zero when the robot moves straight. By comparing their runs between groups, students argue that variation in $d_L + d_R$ dominates their experience, while variation in $d_R - d_L$ is minimal over the same time interval. These ideas emerge entirely from student discussion and comparison, without reference to eigenvectors or covariance.

During Round 1, each team records distance sensor readings (d_L, d_R) in fixed time intervals while the robot moves forward without trying to steer or correct itself. The environment, starting position, and orientation are kept constant throughout the runs to ensure comparability. From a physical standpoint, the dominant phenomenon is simple: as the robot approaches the wall, both distance sensors decrease together over time. When students plot their collected readings in the (d_L, d_R) plane, this shared behavior produces a visibly elongated cloud of points as seen in the blue markers in the left plot of Figure 3. Students are asked to draw—by eye—the direction along which the data appear to vary the most and to explain their reasoning to their peers. Most groups correctly identify a dominant direction corresponding to simultaneous increases or decreases in both d_L and d_R, consistent with their earlier observation that the robot primarily moved straight toward the wall.

At this point, the instructor acts as a mediator by posing a follow-up question: if one direction captures overall change in distance, is there another direction that captures how the two sensor readings differ from one another? Students are then asked to draw a second vector that increases as one sensor reading increases while the other decreases. Through discussion and comparison, they recognize that variation along this direction is minimal in their data and that it corresponds to a left–right imbalance. The two drawn directions are observed to be approximately orthogonal, reinforcing the idea that the captured experience varies strongly along $d_L + d_R$ and only weakly along $-d_L + d_R$. This geometric reasoning reinforces students' initial embodied observations of the robot's motion and provides a visual explanation for why one direction dominates the data while the other does not. These observations are later formalized using linear algebra, allowing eigenvectors and eigenvalues to be understood as precise mathematical descriptions of patterns students have already identified through experience.

The instructor continues to encourage groups to compare proposed directions and to articulate why certain directions capture more variation than others. In this process, the plots and sensor traces support collective geometric reasoning, allowing students to refine hypotheses about structure and variation in the data. This moment is deliberately highlighted: students are already reasoning about eigenvectors intuitively, using geometric structure grounded in shared visual experience rather than algebra.

The instructor then reframes the discussion around a concrete question: *How can we systematically find the direction along which the data vary the most, especially when the answer is not visually obvious?* Earlier work has shown that in simple and strongly correlated datasets, dominant directions can often be qualitatively identified by inspection alone [5, 6]. Building on this observation and the work of these researchers, the instructional goal at this point is to show students how such intuitive reasoning can be formalized and generalized using linear algebra, by quantitatively identifying the direction along which the variance of the data is maximized. To do so, students are asked to imagine projecting the two-dimensional sensor data onto a unit vector v. Each projection produces a one-dimensional signal whose variance depends on the chosen direction. The problem then becomes one of optimization: *find the direction v that maximizes the variance of the projected data*. This formulation leads naturally to the covariance matrix, since the variance of the projection onto v can be expressed in terms of v and the data covariance. Maximizing this variance subject to the constraint that v is a unit vector yields the eigenvector of the covariance matrix associated with the largest eigenvalue, which therefore points in the direction of maximum variance. Additional directions of variation arise by solving the same optimization problem under the constraint that v be orthogonal to previously identified directions, producing subsequent eigenvectors associated with smaller eigenvalues.

At this point, the instructor presents this result as the formal resolution of the earlier geometric reasoning: the directions students identified by eye correspond to the eigenvectors of the covariance matrix, ordered by their associated eigenvalues. Students then use MATLAB to compute the covariance matrix of their collected sensor data and obtain its eigenvectors and eigenvalues, allowing the focus to remain on interpretation rather than arithmetic. The dominant eigenvector emerges aligned with the direction of maximal variance, closely matching the direction identified earlier by eye and corresponding to the intuitive sum direction of the sensors, while the second eigenvector aligns with left–right imbalance. The relative magnitudes of the eigenvalues formalize this distinction, making the idea that the dominant direction is simply the one along which the data changed most during the experience.

In this way, mathematics provides a quantitative justification for earlier qualitative reasoning rather than replacing it. The transition from visual inspection and mediated discussion to formal computation mirrors the progression from socially mediated reasoning to individual mathematical internalization emphasized in Vygotsky's theory.

From Covariance to Coordinates:

At this point, the instructor clarifies that PCA has already been used to *learn* an orthogonal basis from the data, and that the next step is to *use* this basis as a coordinate system. Given the eigenvectors v_1 and v_2 of the covariance matrix, each sensor reading $\mathbf{d} = \begin{bmatrix} d_L \\ d_R \end{bmatrix}$ is expressed in eigencoordinates by projection: $\alpha_1 = \mathbf{d}^\top v_1$ and $\alpha_2 = \mathbf{d}^\top v_2$. PCA was applied to mean-centered distance sensor data collected during an initial approach to a planar wall to identify the dominant direction of the correlated variation between the left and right sensors. While PCA was computed using centered data to estimate the eigenvectors, the resulting eigenvectors were subsequently used as a rotation matrix to define physically anchored coordinates. In these coordinates, the origin corresponds to wall contact $(d_L, d_R) = (0, 0)$, the first coordinate α_1 represents progress toward the wall, and the second coordinate α_2 captures left–right imbalance. α_1 was used as a proxy for forward speed, while α_2 was used to correct angular misalignment, driving the system toward the dominant correlated manifold.

When students expand these expressions algebraically, they see that α_1 corresponds to a weighted version of $d_L + d_R$, while α_2 corresponds to a weighted version of $d_R - d_L$. Importantly, these quantities are not imposed by design or specified in advance; they emerge naturally from the data collected during interaction with the environment. This moment is emphasized as a key pedagogical payoff. The dot products that define α_1 and α_2 are no longer abstract operations, but familiar physical quantities expressed in a principled coordinate system. Rather than introducing new ideas, PCA provides a mathematical explanation for patterns students have already identified through experience, reinforcing the connection between intuition, geometry, and formal linear algebra.

At this point, students implement this change of coordinates directly in code using the LEGO Education SPIKE Python environment. Working in teams, they write a small function as seen in Figure 2 that takes the raw distance readings (d_L, d_R) as input and computes the corresponding eigencoordinates α_1 and α_2 via dot products with the learned eigenvectors. The resulting values are then passed directly into the robot's motor control routine, with α_1 used as the forward velocity parameter and α_2 used as the steering correction. In doing so, students see that eigencoordinates are not merely descriptive quantities, but variables that directly govern behavior.

Because the raw eigencoordinates can be large or noisy, students quickly discover the need to scale α_1 and α_2 by an appropriate constant to produce smooth and stable motion. This tuning is

```python
def compute_eigenspace_coordinates(left_dist, right_dist):
    pc1_v1, pc1_v2 = 0.7370, 0.6759 #this is eigenvector v1 (dominant one i.e. along d_l + dr)
    pc2_v1, pc2_v2 = 0.6759, -0.7370 #this is eigenvector v2 (weaker one i.e. along dl - dr)
    alpha_1 = pc1_v1 * (left_dist) + pc1_v2 * (right_dist)
    alpha_2 = pc2_v1 * (left_dist) + pc2_v2 * (right_dist)
    return alpha_1, alpha_2

```

(a) Student-written Python function that computes eigencoordinates α_1 and α_2 from raw distance sensor readings using dot products with the learned eigenvectors.

```python
            alpha_1, alpha_2 = compute_eigenspace_coordinates(left_dist, right_dist)
            motor_pair.move(motor_pair.PAIR_1, int(alpha_2), velocity=int(alpha_1))
```

(b) Using the computed eigencoordinates to control motion: α_1 is passed as the forward velocity parameter and α_2 as the steering correction in the SPIKE motor control API.

Figure 2: From eigenspace to action. Students implement the learned coordinate system in code and directly map eigencoordinates to robot behavior, completing the transition from data analysis to embodied control.

not prescribed in advance: teams experiment with different scaling factors to balance responsiveness against stability. These adjustments reinforce the interpretation of α_1 and α_2 as action-relevant signals whose magnitudes matter, further grounding the abstract notion of projection in embodied control.

Round 2: Acting in Eigenspace

In Round 2, students run the robot again in the same environment and from the same starting position as in Round 1. The physical setup is unchanged; the key difference is how sensor readings are interpreted. Instead of operating directly in sensor coordinates (d_L, d_R), the robot now operates entirely in the learned eigenspace using the coordinates (α_1, α_2).

At each time step, students program the robot to compute α_1 and α_2 from the incoming sensor readings. The unnormalized value of α_1 is used to control forward speed: when α_1 is large (far from the wall), the robot moves quickly, and as α_1 decreases, the robot slows down smoothly. The value of α_2 is used to control steering corrections: deviations from zero cause the robot to rotate until symmetry is restored. In this phase, the robot no longer reasons about individual sensors; instead, motion is governed directly by the learned eigencoordinates.

In each of the following plots, the blue points show the sensor data collected during Round 1, which was used to compute the eigenvectors and define the eigenspace. The red points show the robot's behavior during Round 2 under eigenspace control, plotted both in the original sensor coordinates (d_L, d_R) (left) and in eigencoordinates (α_1, α_2) (right). This paired visualization allows students to directly relate physical motion in the environment to changes in representation.

Variant A: Straight Approach. In Variant A, students place the robot facing the wall straight ahead as seen in Figure 3. In the left plot, the red points lie almost entirely along the dominant direction identified by the blue Round 1 data cloud, indicating that the robot's motion aligns with the primary mode of variation learned from experience. In the corresponding eigenspace plot, α_1

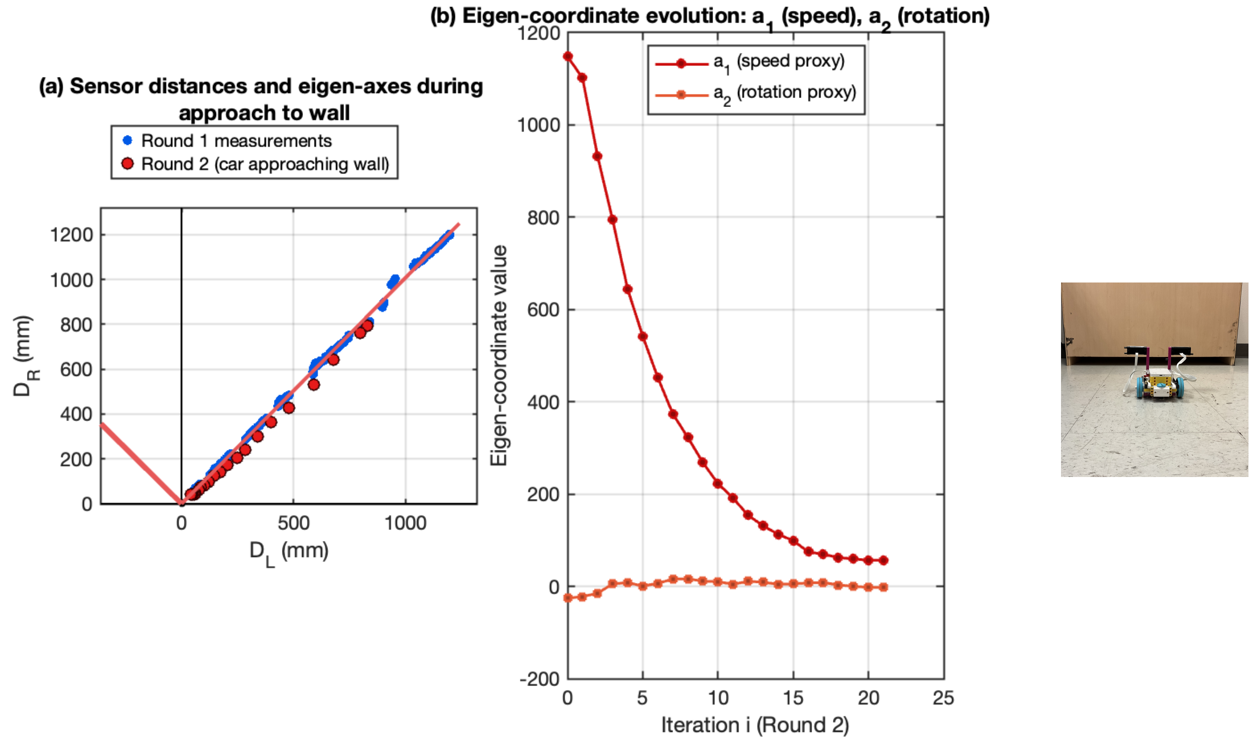

Figure 3: Variant A: The robot moves straight toward the wall while operating in eigenspace. The plot shows the system response, and the image depicts the robot facing the wall.

decreases smoothly as the robot approaches the wall, while α_2 remains near zero. Students interpret this behavior as confirmation that α_1 captures forward progress, while α_2 plays little role when the robot is well aligned.

Variant B: Left Sensor Closer, Right Sensor Farther. In Variant B, students start the robot at a slight angle as seen in Figure 4 so that $d_L < d_R$ initially. In the (d_L, d_R) plot, the red points initially deviate from the dominant direction of the blue data cloud, reflecting the imposed left–right asymmetry. In eigenspace, this deviation appears as a nonzero α_2, while α_1 continues to decrease as the robot moves forward. As the robot rotates to restore symmetry, students observe α_2 returning toward zero, reinforcing the interpretation of α_2 as a correction or rotation coordinate orthogonal to the dominant mode.

Variant C: Right Sensor Closer, Left Sensor Farther. Variant C mirrors Variant B with the asymmetry reversed as seen in Figure 5, so that $d_L > d_R$ initially. In the sensor-coordinate plot, the red points deviate from the dominant Round 1 direction in the opposite sense. In the eigenspace plot, α_2 now has the opposite sign, while α_1 again captures forward progress toward the wall. Students recognize that the sign of α_2 determines the direction of rotation, while its convergence back toward zero signals stabilization around the dominant learned mode.

Across all three variants, students record the raw sensor readings (d_L, d_R), the corresponding eigencoordinates (α_1, α_2), and the number of iterations required for stabilization. Together, these

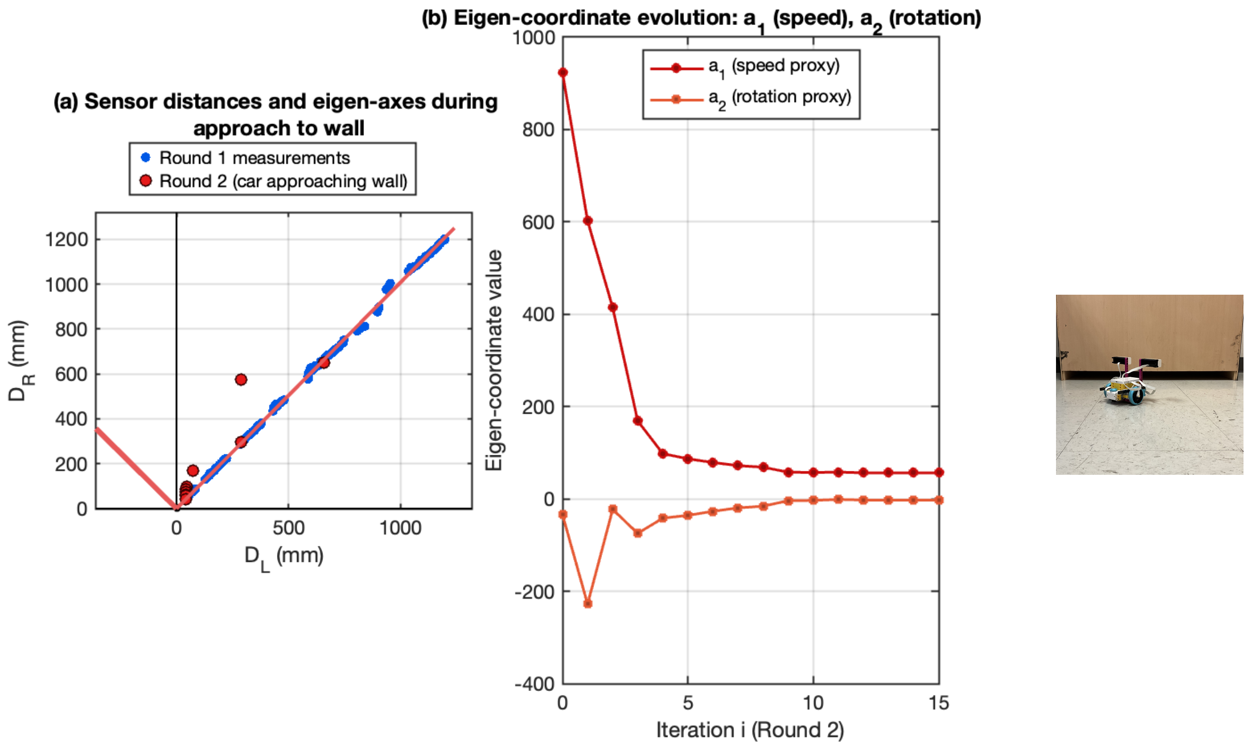

Figure 4: Round 2 Variant B. The robot starts with $d_L < d_R$, producing a nonzero α_2 and triggering an anti-clockwise steering correction and the image depicts the robot's starting orientation facing the wall.

observations make explicit that the same learned eigenspace governs behavior across different initial conditions. The activity reinforces the idea that eigenvectors are not merely descriptive artifacts of past data, but coordinate systems through which the robot perceives and acts in the world.

Some students initially argue that facing the wall itself constitutes the dominant configuration, interpreting the learned behavior as a preference for a particular spatial orientation. Rather than correcting this interpretation directly, the instructor introduces a simple physical intervention designed as seen in Figure 6 to create productive cognitive conflict: a white envelope is placed in front of only one distance sensor.

With one sensor blocked, the robot initially reports a short distance on one side and a much larger distance on the other, producing a nonzero α_2 and inducing rotation. As the robot turns, both sensors eventually face the envelope, at which point their readings become correlated again. This new orientation becomes stable, and the robot begins to follow the envelope as it is moved. This moment is pedagogically critical. Students observe that the system's learned behavior is not tied to a specific direction in space, such as "facing the wall," but to a correlation structure learned from experience. The robot seeks configurations in which its sensor readings vary together, because that pattern dominated the data collected during Round 1. From a learning-theoretic perspective, the envelope intervention is deliberately positioned within students' ZPD. The task is close enough to students' existing understanding to be interpretable, yet sufficiently surprising to

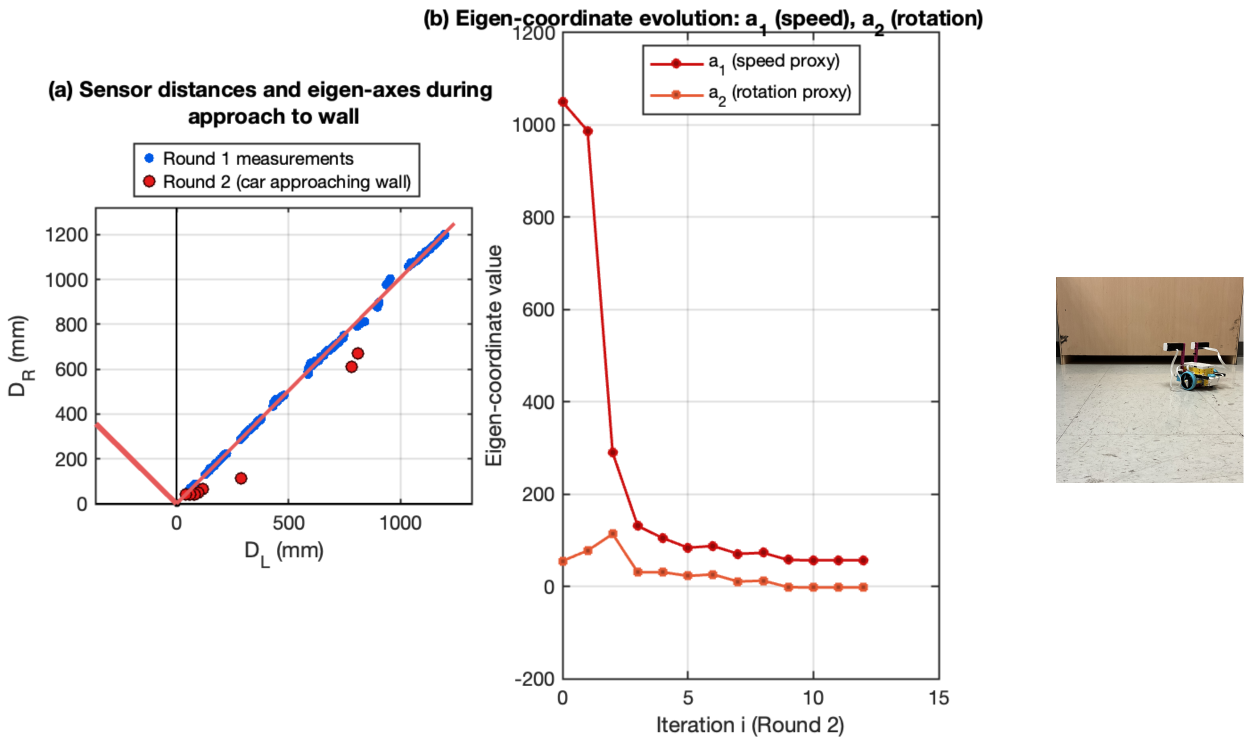

Figure 5: Variant C: The robot starts with $d_L > d_R$, producing a steering correction in the opposite direction and the image depicts the robot's starting orientation facing the wall.

expose a limitation in their current explanation. Rather than being told that their interpretation is incomplete, students reconcile the unexpected behavior by revisiting the learned eigencoordinates and recognizing that correlation—not orientation—is the invariant being maintained. In this way, the instructor's role remains mediational, and the formal interpretation of eigenvectors as learned coordinate systems is internalized through experience rather than imposed by authority.

To further consolidate the idea that eigenvectors are learned representations whose meaning depends on embodiment, students are asked to consider an alternative, intentionally misaligned coordinate system. Working on paper, students draw two orthogonal axes in the (d_L, d_R) plane, choosing one direction to represent anti-correlated sensor readings and the other to represent correlated readings. Specifically, the instructor asks students to define $v_1 = \begin{bmatrix} -0.7 \\ 0.7 \end{bmatrix}$ and $v_2 = \begin{bmatrix} 0.7 \\ 0.7 \end{bmatrix}$ and to designate v_1 as the dominant direction. At this stage, students are asked to predict what would happen if the robot were controlled using these eigenvectors instead of those learned from Round 1. They then modify the Python control code to replace the learned eigenvectors with v_1 and v_2, while keeping the same eigenspace control logic. When the robot is run, students observe that α_1 remains near zero and the robot fails to move forward, resulting in little or no motion.

The instructor mediates discussion by asking why α_1, designated as the dominant coordinate, does not change. Through reasoning about sensor geometry and physical constraints, students

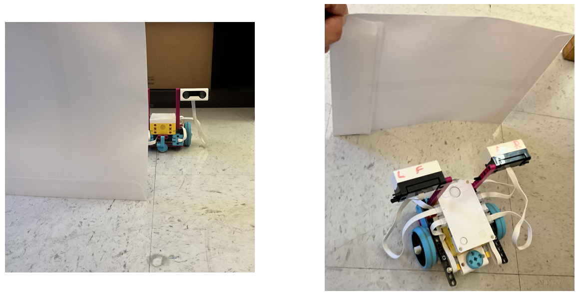

Figure 6: Envelope intervention. Left: an envelope blocks one distance sensor, producing asymmetric readings. Right: as the robot rotates, both sensors face the envelope, the readings become correlated again, and the configuration stabilizes.

recognize that forward motion toward a wall necessarily produces correlated sensor readings. There is no physical way for a robot with two forward-facing distance sensors to generate sustained anti-correlated readings while moving forward. As a result, the projection of the sensor data onto v_1 remains near zero, and the control law interprets this as zero speed.

This moment is pedagogically significant. Students come to understand that declaring a direction to be dominant does not make it so; dominance must be grounded in the structure of experience and the constraints of embodiment. The only way to make motion along the anti-correlated direction physically realizable would be to reconfigure the system itself, for example by relocating one sensor to the back of the robot. In this way, students see that eigenvectors are not arbitrary mathematical choices, but representations whose validity depends on how a system can interact with the world.

From a learning-theoretic perspective, this intervention sits squarely within students' ZPD. Students are not told that their chosen representation is incorrect; instead, they encounter a controlled failure that can only be explained by revisiting the relationship between data, representation, and embodiment. By reconciling this failure through reasoning rather than correction, students internalize a deeper understanding of what it means for a coordinate system to be meaningful, dominant, and actionable.

4 Practical Considerations and Sensor Robustness

Implementing this activity with LEGO Education SPIKE exposed several practical issues related to the behavior of the distance sensors, which, in turn, informed both the instructional design and the supporting software. Although these challenges are not unique to SPIKE, they are important to document in order to distinguish pedagogically relevant complexity from incident technical friction. The distance sensors used in this activity are not LiDAR-based sensors but rather rely on emitting a pulse and inferring distance from its return. In practice, this leads to several characteristic failure modes. In particular, sensors occasionally become `stuck` reporting a maximum distance (approximately 200 cm) even when an obstacle is clearly present at a distance less than 200 cm, or fluctuate intermittently between a correct reading and the maximum value. These behaviors were observed more frequently when the robot was far from the wall or when both sensors were queried in rapid succession, suggesting interference or timing effects between the sensors.

Through experimentation, we observed that the system behaved most reliably when the robot was within approximately 100 cm of a large flat surface. At greater distances, sensors were more likely to return persistent maximum readings. In addition, we found that unplugging and reconnecting a sensor often restored correct behavior. However, manual intervention is not appropriate in a classroom setting and would distract from the conceptual goals of the activity. To address this, we deliberately handled sensor unreliability in software so that students could focus on eigenvectors, representation, and interpretation rather than debugging hardware. Our Python implementation includes several robustness mechanisms. First, distance readings are sampled multiple times in rapid succession and aggregated using a median-based strategy, which mitigates the effect of transient fluctuations between valid and invalid readings. Second, readings from the left and right sensors are staggered with a short delay, reducing interference when both sensors are active.

In cases where a sensor repeatedly reports invalid or maximum-range values, the system enters a brief `wiggle` recovery routine. This routine applies a sequence of small, low-speed steering motions designed to slightly change the sensor angle relative to the environment. Empirically, this often causes a previously stuck sensor to resume reporting valid distances, replicating in software the same effect observed when manually repositioning the robot or reconnecting a sensor cable. If recovery is successful, normal operation resumes automatically; if not, the system returns to the last known valid reading to maintain continuity of behavior. Importantly, these mechanisms were not introduced to optimize robot performance, but to preserve the pedagogical integrity of the activity. By handling sensor instability transparently in the code, students were not required to reason about low-level hardware quirks to understand the core ideas of PCA, eigencoordinates, and representation. Calibration, sampling, and recovery were treated as engineering necessities rather than learning objectives. All code used in this activity, including sensor monitoring logic, sampling strategy, and recovery routines, is publicly available on the author's GitHub repository [1] to support replication and adaptation by other instructors. Although more precise sensors could reduce the need for such measures, we found that addressing these limitations in software allowed the activity to remain focused on its primary conceptual goal: helping students understand how eigenvectors emerge from experience and how eigencoordinates transform raw measurements into interpretable action variables.

5 Discussion

Across multiple offerings of linear algebra and applied linear algebra courses at the University of Toronto, Boston University, and California State University, Chico, we have repeatedly observed a common pattern in how students engage with eigenvectors and eigenvalues. Even when students can compute eigenpairs correctly, they often struggle to articulate why a particular direction should be meaningful or how eigenvectors relate to phenomena outside of textbook exercises. This experience paper suggests that the difficulty is not primarily computational, but

[1] https://github.com/attarwal/lego

interpretive.

Our prior instructional work has addressed this gap by embedding eigenvectors in applied computer science contexts, including PCA based image compression and the use of low pass and high pass filtering to connect eigenvectors to signal structure. These approaches helped students see eigenvectors as tools for organizing data, but still largely operated in static, retrospective settings. The LEGO Education SPIKE activity extends this trajectory by positioning PCA as a way to *learn* a coordinate system from embodied experience and then use it to structure ongoing interpretation and action. In this implementation, the most salient learning moments occurred when students treated eigenvectors not as instructor imposed geometry, but as evidence of which patterns the world made salient for the robot.

In this activity, the students' initial intuitions were often directionally correct. Students readily recognized that overall distance to the wall mattered more than small left right discrepancies and that steering corrections should be secondary to forward progress. What the students lacked was not intuition, but a principled account of *why* some combinations of measurements matter more than others and how this prioritization can be justified from data rather than design.

The staged structure of the activity is central here. Because students first commit to qualitative predictions and only later compute PCA, eigenvectors are experienced less as answers and more as mathematical explanations for patterns already observed. In this setting, PCA functions as a bridge between informal reasoning (for example, reasoning in terms of the sum and difference of the readings of the left and right distances d_L and d_R) and a general method to discover such a structure systematically, since the learned eigencoordinates α_1 and α_2 correspond to weighted versions of the overall distance ($d_L + d_R$ and the imbalance $(-d_L + d_R)$. This reframes eigenvectors from arbitrary directions to coordinates that can be defended by experience.

The embodied robot highlights a constraint on eigenvectors that is often obscured in purely computational PCA activities: not every mathematically valid change of basis yields a representation that can be meaningfully used by a physical system. Even when a change of basis is algebraically correct, it may fail to correspond to any achievable sensorimotor interaction given the geometry of the sensors and the structure of the environment. Experiencing this mismatch helps students recognize that representations are evaluated not only by mathematical correctness but also by how well they support interpretation and action in a given context. The Round 2 variants make this constraint legible to students. When students propose dominant directions that conflict with the geometry of the sensors and the structure of the environment, the resulting representation fails to support stable behavior. This failure is pedagogically productive precisely because it is not a bug to debug away, but a concrete instance of representational mismatch. In this sense, embodiment does not merely motivate PCA; it supplies an external standard against which learned representations can be evaluated. A second learning payoff is that the eigenvectors become a *lens* rather than a static output. In this activity, both the native sensor coordinates (d_L, d_R) and the learned eigencoordinates (α_1, α_2) are two dimensional, making the change of coordinates concrete and directly experienceable. When the robot operates in eigencoordinates, the same raw sensor readings are transformed into action relevant quantities with distinct roles, where α_1 serves as a proxy for forward progress and α_2 for rotation and correction. Because this transformation occurs in real time and governs the robot's behavior, students can easily connect the learned eigenspace to the robot's movement in the physical world. As a result, students' reasoning often shifts away from individual sensor values to concepts such as motion, stability, and relative strength of competing signals. This immediacy stands in contrast to high dimensional PCA applications such as image compression, where the relationship between original coordinates and eigencoordinates can be mathematically correct yet difficult for students to interpret intuitively. This reframing also provides a natural interpretation for eigenvalues. In many PCA based activities, eigenvalues are experienced as bookkeeping quantities that must be sorted. Here, eigenvalues are tied to reliability. Larger eigenvalues correspond to patterns that dominated the robot's experience and therefore provide stable signals for behavior, whereas smaller eigenvalues correspond to weaker effects and noise. This distinction becomes particularly salient when students intentionally disrupt the learned ranking and observe degraded behavior. Together, these observations suggest a complementary way to frame PCA in undergraduate courses. Alongside its common presentation as a tool for analyzing or compressing static datasets, PCA can be introduced as a method to learn representations that will later be used to interpret new data. For students, a key realization is that PCA does not learn a representation for the sake of controlling the mobile robot in Round 2; it learns a representation from experience in Round 1. The usefulness of that representation becomes apparent only when it is later used to interpret new sensor readings and guide behavior. In environments where dominant variance arises from noise, lighting changes, or irrelevant motion, PCA would not be expected to produce action relevant coordinates. In this way, PCA becomes neither a universally applicable solution nor a purely abstract technique, but a method whose usefulness depends on how patterns of variation align with the geometry of the environment, the configuration of the sensors, and the way those sensor readings are later used for action. For instructors seeking to adapt the approach, LEGO Education SPIKE plays an important enabling role by providing a low cost, accessible platform for embodied interaction. At the same time, the core design principles are not specific to LEGO or to distance sensors. The activity is structured so that students first make qualitative predictions that surface their informal hypotheses and then engage in an experience gathering phase in which structure is not imposed in advance. This is followed by a representational learning step, such as PCA, which produces an interpretable change of basis grounded in that experience.

A natural objection is that the robot can be controlled directly from $d_L + d_R$ (distance) and $d_R - d_L$ (imbalance) without PCA. We include PCA not for control optimality but for interpretation: PCA makes these combinations discoverable from experience, quantifies their relative reliability via eigenvalues, and exposes when a mathematically valid basis is physically unrealizable (e.g., forcing anti-correlated "dominant" motion with two forward-facing sensors). In this way, PCA functions as representational learning rather than a post hoc re-expression step.

6 Conclusion

This paper described an embodied instructional sequence for teaching PCA that foregrounds eigenvectors as learned coordinate systems rather than abstract mathematical artifacts. Using a LEGO Education SPIKE robot, students collected paired distance readings (d_L, d_R), used PCA to learn directions of variation, then operated in learned eigencoordinates (α_1, α_2) that transformed sensor readings into action-relevant quantities. A central contribution is the prospective role of representation: rather than re-expressing fixed datasets retrospectively, the activity treats PCA as learning coordinates from past experience to interpret new data online, connecting eigenvectors to intuitive quantities like overall distance and imbalance that emerge without advance specification. Following Vygotsky's view of conceptual development through socially mediated activity, students began with collaborative prediction and interpretation before formal linear algebra was introduced, allowing eigenvectors and eigenvalues to emerge as tools for naming and acting on structure already encountered through embodied interaction. This sequence provides a replicable model for teaching PCA as representational learning, with a two-phase structure—learn from experience, then act in learned coordinates—that may generalize to other linear algebra topics where students compute successfully but struggle to explain meaning.

Acknowledgments

We gratefully acknowledge the use of Anthropic's Claude for proofreading, grammatical checks, and other editing tasks.

References

[1] R. André and X. Luciani, "Sparsity constraint implementation for the joint eigenvalue decomposition of matrices," in *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP).* IEEE, 2023, pp. 1–5.

[2] J. M. Philot, B. L. Bianchini, E. Gomes, G. L. de Lima, and O. M. Neto, "Elaboration of a contextualized event for teaching eigenvalues and eigenvectors in the control and automation engineering course," in *2023 ASEE Annual Conference & Exposition*, 2023.

[3] M. Piroi and F. Arzarello, "Students' difficulties with eigenvalues and eigenvectors. an exploratory study," *Italian Journal of Pure and Applied Mathematics*, p. 153.

[4] M. Wawro, K. Watson, and M. Zandieh, "Student understanding of linear combinations of eigenvectors," *ZDM – Mathematics Education*, vol. 51, pp. 1111–1123, 2019, investigates how students reason about the conceptual structure of eigenspaces and eigenvectors.

[5] A. Attarwala and E. Lindoo, "An integrated pedagogical approach to teaching convolution filtering and principal component analysis," in *Proceedings of the 26th ACM Annual Conference on Cybersecurity & Information Technology Education*, 2025, pp. 240–246.

[6] A. Attarwala, "Rethinking linear algebra for computer science: Applying vygotsky's theory of learning," *Journal of Computing Sciences in Colleges*, vol. 39, no. 10, pp. 23–32, 2024.

[7] C. Aytekin and Y. Kıymaz, "Teaching linear algebra supported by geogebra visualization environment," *Acta Didactica Napocensia*, vol. 12, no. 2, pp. 75–96, 2019.

[8] J. A. Anderson and M. V. McCusker, "Make the eigenvalue problem resonate with our students," *PRIMUS*, vol. 29, no. 6, pp. 625–644, 2019.

[9] M. T. López-Díaz and M. P. na, "Encouraging students' motivation and involvement in stem degrees by the execution of real applications in mathematical subjects: The population migration problem," *Mathematics*, vol. 10, no. 8, p. 1228, 2022.

[10] K. P. Bennett, J. S. Erickson, A. Svirsky, and J. C. Seddon, "A mathematics pipeline to student success in data analytics through course-based undergraduate research," *The Mathematics Enthusiast*, vol. 19, no. 3, pp. 730–750, 2022.

[11] M. Silva, P. Hieronymi, M. West, N. Nytko, A. Deshpande, J. Chuang, and S. Hilgenfeldt, "Innovating and modernizing a linear algebra class through teaching computational skills," in *Proceedings of the 2022 ASEE Annual Conference & Exposition*, 2022.

[12] M. Li, "Developing active learning of linear algebra in engineering by incorporating matlab and autograder," in *Proceedings of the 2023 ASEE Annual Conference & Exposition*, 2023.

[13] H. Chen, M. West, S. Hilgenfeldt, and M. Silva, "Measuring the impact of a computational linear algebra course on students' exam performance in a subsequent numerical methods course," in *Proceedings of the 54th ACM Technical Symposium on Computer Science Education (SIGCSE)*, 2023.

[14] R. T. Shankar, D. Ploger, A. Nemeth, and S. A. Hecht, "Robotics: Enhancing pre-college mathematics learning with real-world examples," in *Proceedings of the 2013 ASEE Annual Conference & Exposition*, 2013.

[15] L. S. Vygotsky, *The collected works of LS Vygotsky: Problems of the theory and history of psychology*. Springer Science & Business Media, 1987, vol. 3.