Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

Design and computational evaluation of an eye gaze wheelchair retrofit with temporal command supervision

Hasan Fayyad-Kazan

h.kazan@kcst.edu.kw

Kuwait College of Science and Technology

Aseel Alenezi

Kuwait College of Science and Technology

Latifah Alsulaili

Kuwait College of Science and Technology

Aisha Hadi

Kuwait College of Science and Technology

Raghad Alrasheedi

Kuwait College of Science and Technology

Research Article

Keywords: eye gaze control, powered wheelchair, assistive technology, temporal supervision, fault injection, stopping distance

Posted Date: October 30th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-11019085/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: No competing interests reported.

Abstract

Eye gaze offers an alternative control channel for powered mobility, but reliable operation requires both deliberate command selection and timely withdrawal of motion when input becomes invalid. This study presents iMove, a wheelchair retrofit comprising a commercial eye tracker, a tablet interface, an Arduino Uno, and two servomotors that mechanically actuate the wheelchair joystick. We combined preliminary functional assessment with executable controller replay, fault injection, component ablation, Monte Carlo sensitivity analysis, and a kinematic stopping model. Preliminary testing yielded eight successful command selections in ten attempts, corresponding to 80.0% success with a 95% Wilson interval of 49.0–94.3%. The baseline controller activated on target entry and retained its motion command after target exit or interrupted communication. We therefore developed and computationally evaluated a temporal supervisor with a 0.60s motion dwell, a 0.15s input-freshness threshold, and a 0.30s receiver timeout. In deterministic fault scenarios, the supervisor neutralized the command 0.12s after input freezing and 0.24s after host or serial-link interruption; the baseline remained active throughout the 3.5s post-fault window. Ablation demonstrated that receiver timeout alone could not detect stale input accompanied by continuing messages. At an assumed speed of 0.50 m/s, a linear-deceleration model using the preliminary 3–4s stopping interval produced a conditional stopping envelope of 0.78–1.13m. The findings establish complementary roles for selection dwell, input freshness, receiver expiry, and braking characterization in gaze-controlled mobility. Physical implementation of the supervisor and evaluation with intended users remain necessary to establish operational performance.

1 Introduction

Powered mobility can support independence when an appropriate control interface matches the user's residual motor abilities. The World Health Organization estimates that 1.3 billion people experience significant disability [1]. Within this diverse population, some individuals cannot reliably use a conventional wheelchair joystick but retain sufficient visual and oculomotor function to interact through gaze. For these users, gaze can provide access to directional control without requiring hand movement. Its suitability nevertheless depends on individual abilities, viewing conditions, calibration stability, and the interaction demands imposed during navigation.

Previous research has demonstrated the feasibility of eye-controlled wheelchairs. Elliott et al. evaluated an eye-controlled power wheelchair with participants with amyotrophic lateral sclerosis [2], while Dahmani et al. integrated camera-based gaze recognition with proximity-based motion interruption [3]. These studies establish a basis for gaze-enabled mobility and motivate closer examination of the command path between the user interface and the wheelchair. An intended direction must be selected, transmitted, converted into an actuator command, and withdrawn when the input no longer authorizes movement. Failure at any of these stages can produce behavior that is poorly represented by a single selection-accuracy measure.

A particular difficulty is that gaze serves both environmental observation and interface interaction. Looking at a directional target may reflect an intention to move, a brief visual scan, or an involuntary transition. Dwell selection can suppress short entries, but it also delays intended activation. Moreover, a stable pointer does not necessarily indicate a valid fixation: it may remain stationary after eye tracking or upstream processing stops. Similarly, periodic messages can continue even when the user input used to generate them is stale. These conditions require separate treatment in the controller.

We investigate this problem using iMove, a mechanical joystick retrofit that connects a gaze-operated tablet interface to an existing powered wheelchair (Fig. 1). The study evaluates a baseline command controller and develops a temporal supervisor that combines motion dwell, neutralization on target exit, input-freshness checking, and receiver-side command expiry. We address three questions: how transient input and interrupted updates affect the baseline command state; which supervisory components handle each modeled fault; and how command timing contributes to travel before rest. The contribution is a reproducible integration of prototype assessment, executable fault analysis, controller ablation, and stopping-distance sensitivity. The resulting evidence concerns preliminary functionality and computational control behavior rather than clinical effectiveness.

2 Related works

Eye-controlled mobility studies evaluate several distinct outcomes, including gaze recognition, task completion, interface usability, and navigation performance. Elliott et al. [2] studied 12 participants with amyotrophic lateral sclerosis, each completing three indoor-course trials involving starting, stopping, turning, reversing, and obstacle negotiation. All participants completed the trials, and eight completed the component tasks without errors. Dahmani et al. [3] reported 99.3% classification accuracy for a convolutional gaze-recognition approach and evaluated their system with eight subjects. Their architecture included proximity sensors capable of overriding movement. These results provide relevant precedents, although classification accuracy and course completion are not directly comparable with individual command-selection success.

Research on gaze interaction also addresses unintended activation. EyeTAP separates gaze pointing from activation through an acoustic input [4]. Riou et al. investigated crossing-based interaction as an alternative to dwell [5]. Such methods illustrate how activation can be distinguished from visual attention, but their suitability depends on the user's available input capabilities and the surrounding task. In the present study, dwell provides a single-channel selection mechanism whose timing can be explicitly varied. We examine its sensitivity computationally without assuming a physiological distribution for unintended gaze entries.

Shared control adds a further layer by combining user commands with environmental information. Xu et al. evaluated manual, shared, and autonomous modes of the CoNav Chair with 21 unimpaired adults in a 2025 preprint [6]. That work concerns navigation assistance within the built environment. Our focus is the lower-level validity and lifetime of a motion request, which remains relevant whether a wheelchair

uses direct steering or higher-level navigation assistance. The supervisor evaluated here does not implement obstacle sensing or path planning.

Together, these studies motivate an architecture in which gaze estimation, intentional selection, command validity, and physical motion are evaluated as distinct functions. We use a simple retrofit to investigate the interfaces between these functions. The study's contribution lies in their combined evaluation and the resulting fault-coverage analysis; dwell and watchdog timers are established control mechanisms.

3 Materials and methods

3.1 Study design

The study combined a preliminary prototype assessment with computational evaluation of command supervision. The prototype assessment comprised ten command-selection attempts, approximate actuator and stopping-time observations, and a ten-minute continuous-operation check. These observations characterized basic functionality. Aggregate outcomes were available for analysis, without participant-level records, direction-specific counts, or synchronized timing traces. We therefore used the selection data descriptively and treated the timing ranges as approximate inputs to a conditional kinematic analysis.

The computational evaluation comprised four parts. First, we executed the baseline interface logic under controlled pointer and communication conditions. Second, we compared three supervisor configurations through component ablation. Third, we quantified the effect of dwell on assumed transient target entries using Monte Carlo simulation and numerical integration. Finally, we calculated stopping-distance envelopes to distinguish command neutralization from physical rest. The supervisor was evaluated in software; no hardware improvement is inferred from its simulated behavior.

Table 1
Preliminary prototype outcomes and the quantities they characterize.

Evaluation	Outcome	Quantity characterized
Command selection	8 successful attempts out of 10	Aggregate selection success
Unsuccessful selection	2 omissions	Failure to complete the intended selection
Click to actuator response	Approximately 0.1–0.2 s	Response after a click, excluding gaze acquisition
STOP actuator response	Less than 0.5 s	Actuator response to neutral selection
STOP to wheelchair rest	Approximately 3–4 s	Overall stopping interval, dependent on speed
Continuous operation	10 min without an observed hardware malfunction	Short-duration functional operation

3.2 Hardware and interface architecture

The iMove architecture comprises a commercial Tobii eye-tracking stack, a Microsoft Surface Pro 6 running Windows and a Python graphical interface, a USB serial connection, an Arduino Uno, and two MG996R servomotors. The servomotors mechanically displace the existing wheelchair joystick along its two control axes. The wheelchair's drive controller translates this displacement into motor operation. Consequently, servo setpoints specify joystick actuation rather than wheel velocity or braking torque (Fig. 2).

The interface arranges eight directional targets around a central STOP target. The directional set comprises forward, backward, left, right, and four diagonal combinations. Dark-mode and camera-display options support visual presentation; the camera view does not perform obstacle detection. The host sends one-character commands at 9600 baud, and the receiver maps recognized characters to paired servo setpoints.

Table 2 gives the complete mapping.

Table 2
Directional command mapping and nominal servo setpoints in degrees.

Target	Serial character	Pin 11 servo	Pin 9 servo
Forward	W	180	90
Backward	S	0	90
Left	A	90	0
Right	D	90	180
Forward left	Q	180	0
Forward right	E	180	180
Backward left	Z	0	0
Backward right	C	0	180
STOP	Space	90	90

Both servomotors initialize at 90°, corresponding to the nominal neutral command. The endpoint values require calibration to the joystick and mounting geometry. They are software setpoints rather than measured linkage angles. In particular, simultaneous endpoint actuation on both axes does not establish equal resultant speed for diagonal and axial motion. The present analysis operates at the command level and does not infer a mechanical safety margin from the servo specifications.

3.3 Baseline command controller

The baseline interface contains button-click callbacks and a periodic pointer-processing loop. The loop obtains desktop pointer coordinates through pyautogui.position() [7], determines whether they fall within a target, and transmits the associated character when it differs from the last command. The nominal rescheduling delay is 30 ms. The actual interval can be longer because GUI processing, camera display, and operating-system scheduling share the host execution environment.

In this command path, target entry can activate motion without a completed click or dwell event. Leaving all target rectangles clears the host's last-command marker but does not transmit STOP. On the receiver, recognized characters update the paired servo setpoints; otherwise, the previous setpoints remain unchanged. The baseline firmware has no command-age timeout. Let u_k denote the held joystick command and $M(c_k)$ the mapping of an accepted character. A recognized character assigns $u_{k+1} = M(c_k)$. In the absence of a recognized character,

$$u_{k+1} = u_k. \quad (1)$$

This state-retention behavior is the baseline against which supervision is evaluated. It assumes a powered receiver that does not reset during a communication interruption. It does not predict physical

behavior following a power loss, mechanical release, or intervention by an independent wheelchair protection function.

The pointer interface supplies position but does not expose a gaze-acquisition timestamp or tracker-validity flag to the controller. This distinction matters because a frozen pointer and a valid stationary fixation can have identical coordinates. A change in position is therefore unsuitable as the sole test of input validity.

3.4 Temporal command supervisor

We designed a temporal supervisor that authorizes motion only after a direction remains continuously selected for dwell duration d. A direction change restarts dwell and requests neutral during the transition. Target exit, STOP selection, and stale upstream input each request neutral at the next host evaluation. STOP bypasses motion dwell so that deliberate activation filtering does not impose an additional stopping delay.

Input validity depends on the age of the most recent upstream sample. For current time t, sample time t_g, and freshness threshold F, a sample is fresh when

$$t - t_g < F. \quad (2)$$

The timestamp represents acquisition or validated input production, rather than repeated access to the same pointer coordinate. Freshness information is supplied explicitly in the computational model. A physical implementation would require access to suitable timestamps and validity information from the deployed tracking interface.

During authorized motion, the host refreshes its command every 60 ms. Independently, the receiver requests neutral when the time since its last accepted command reaches timeout W. The modeled receiver evaluates this condition at period $\Delta_m = 1$ ms. With $W = 0.30$ s, the resulting upper neutralization bound is $W + \Delta_m = 0.301$ s after the last accepted command, provided the receiver and its clock remain operational. Servo movement and wheelchair braking occur after this command-level decision and are outside the bound.

The full supervisor uses $d = 0.60$ s and $F = 0.15$ s. These are explicit design settings for evaluation, rather than user-optimized thresholds. Simulations begin armed with a neutral command. Arming, rearming, receiver failure recovery, packet-integrity checks, and mechanical jam detection are outside the evaluated implementation.

3.5 Executable fault injection and ablation

We executed the baseline Python run_loop method through an abstract-syntax-tree harness that preserved its method body. Mock widgets, controlled pointer positions, and an inert serial interface replaced hardware-dependent inputs and outputs. The receiver was modeled using Eq. (1). This arrangement reproduces the command-generation logic without starting the desktop application or

accessing a wheelchair. Two nonoverlapping target rectangles represented forward and STOP; a third location represented target exit.

Seven scenarios were evaluated over 5 s on a 1 ms grid: sustained valid input, exit from all targets, frozen pointer with stale upstream samples, stalled host processing, serial-link interruption, explicit STOP, and a brief target entry from rest. Faults began at 1.50 s. The brief entry occupied the forward target from 0.30 to 0.42 s, giving a duration of 120 ms. Host evaluations occurred every 30 ms, and available-link transport was idealized as instantaneous. The serial-loss scenario retained receiver power and excluded reset.

To isolate the contribution of each mechanism, we compared the baseline with three supervisory configurations: dwell with neutral-on-exit; dwell and neutral-on-exit with receiver timeout; and the full configuration including freshness checking. All configurations received identical input traces. The primary outcomes were activation during the brief entry and elapsed time from fault onset to a neutral command. A command still active at 5 s was classified as held at the observation horizon, rather than assigned an infinite duration.

Table 3
Computational parameters and their roles.

Parameter	Value or range	Role in the analysis
Host evaluation period	0.030 s	Idealized interface scheduling
Receiver evaluation period	0.001 s	Timeout evaluation resolution
Motion refresh period	0.060 s	Host command renewal
Motion dwell	0.60 s; sweep 0–0.90 s in 0.30 s steps	Intentional-selection threshold
Input freshness threshold	0.15 s	Maximum accepted sample age
Receiver timeout	0.30 s	Command expiry threshold
Transient median duration	0.08, 0.15, 0.30 s	Assumed sensitivity scenarios
Transient log standard deviation	0.60	Assumed distribution shape
Monte Carlo episodes	100,000 per combination	Numerical sampling precision
Random seed	20260909	Deterministic reproduction
Total STOP-to-rest interval	3–4 s	Approximate prototype observation
Delay before deceleration	0.10–0.50 s	Assumed part of total stopping time
Initial speed in tabulation	0.25, 0.50, 0.75, 1.00 m/s	Hypothetical speed scenarios

3.6 Statistical analysis and transient sensitivity

For preliminary command selection, success was estimated as $\hat{p} = x/n$, with $x = 8$ and $n = 10$. We calculated a 95% Wilson score interval using the NIST formulation [8] and an exact Clopper–Pearson interval using SciPy [9]. These intervals assume independent Bernoulli attempts with a common probability. They quantify uncertainty under that model; the aggregate data do not support participant-level inference or adjustment for learning and repeated observations.

For transient sensitivity, target-occupancy durations L followed assumed lognormal distributions with median $m \in \{0.08, 0.15, 0.30\}$ s and log-scale standard deviation 0.60. The phase ϕ between target entry and the next host evaluation was uniformly distributed over $[0, \Delta_h)$, where $\Delta_h = 0.03$ s. A transient activated the motion command if

$$L \geq \phi + q\Delta_h, \quad (3)$$

where q is the smallest nonnegative integer satisfying $q\Delta_h \geq d$. Each episode began neutral with valid upstream input and an available communication link. All evaluated dwell durations were integer multiples of the host period.

We generated 100,000 episodes for each of the twelve median–dwell combinations, using shared standardized random draws and evaluation phases. Monte Carlo activation probabilities were compared with numerical integration of the lognormal survival function over the phase distribution. The numerical acceptance tolerance was six binomial Monte Carlo standard errors plus one sample unit. This cross-check tests the implementation of the assumed model. The duration distributions were not fitted to human gaze data and do not provide an empirical false-activation rate.

3.7 Stopping-distance model and component cost

We used the preliminary 3–4 s stopping observation as a total interval T from STOP receipt to rest. To separate command processing from braking, we assumed an initial speed v_0, a pre-deceleration delay τ, and constant deceleration over the remaining interval $T - \tau$. Integrating this speed profile gives

$$D_{\text{linear}} = v_0\tau + \frac{v_0\,(T-\tau)}{2} = \frac{v_0\,(T+\tau)}{2}. \quad (4)$$

The envelope uses $3 \leq T \leq 4$ s and $0.1 \leq \tau \leq 0.5$ s. The delay range is a modeling assumption informed by approximate actuator timing, rather than a measured brake-engagement distribution. Because T is the total stopping interval, actuator delay is not added a second time. The envelope endpoints are deterministic sensitivity limits, not confidence limits.

For comparison, if speed remains nonnegative and does not exceed v_0 before rest, then $D \leq v_0 T$. An additional wait W before STOP contributes at most $v_0 W$ under the same speed assumption. We calculated both terms to distinguish travel during command persistence from travel after STOP. The analysis concerns longitudinal motion and assumes that the preliminary stopping-time range applies at

the hypothetical speeds evaluated. It excludes slopes, turning, tire slip, payload variation, and braking failure.

Component expenditure was summarized in Kuwaiti dinars (KWD), separating the wheelchair and tablet from the retrofit components. Costs represent the prototype acquisition configuration and exclude labor, validation, maintenance, and protective components added in a subsequent implementation.

4 Results

4.1 Preliminary prototype performance

Eight of ten command-selection attempts were successful, giving an aggregate success proportion of 80.0%. The two unsuccessful attempts were omissions associated with leaving the intended target before completing selection. The 95% Wilson interval was 49.0–94.3%, and the exact binomial interval was 44.4–97.5% (Fig. 3). Both intervals were broad, reflecting the small number of attempts. Direction-specific performance could not be estimated from the aggregate outcomes.

Approximate click-to-actuator response was 0.1–0.2s. Actuator response to STOP was below 0.5s, whereas wheelchair rest required approximately 3–4s, depending on speed. The ten-minute continuous-operation check produced no observed hardware malfunction. These observations establish short-duration functionality and distinguish actuator responsiveness from complete stopping; they do not provide an endurance estimate or a resolved braking trajectory.

Total component expenditure was 844 KWD, including 299 KWD for the wheelchair and 400 KWD for the tablet. Excluding both platforms gives 145 KWD for the itemized retrofit components. Excluding only the already-owned tablet gives 444 KWD. These totals describe acquisition expenditure and do not constitute a cost-effectiveness comparison.

4.2 Command behavior under injected faults

The baseline controller activated during the 120 ms target entry without a click event and retained the forward command after target exit. The full supervisor rejected the entry because it did not satisfy the 0.60s dwell requirement. Under sustained valid input, the baseline activated at the first host evaluation, while the supervisor activated at 0.60s. Both continued to authorize motion while the input remained valid.

For target exit, pointer freezing, host stall, and serial interruption, the baseline command remained active throughout the 3.5s interval after fault injection. The full supervisor neutralized at the host evaluation coincidence with target exit, 0.12s after pointer freezing, and 0.24s after host or serial interruption. Both controllers responded to explicit STOP at the coincident host evaluation. The apparent zero-delay responses arise from the selected fault phase and ideal transport; they are not measurements of instantaneous physical stopping.

Table 4 summarizes the ablation. Dwell and neutral-on-exit blocked brief activation and handled a detectable departure from the targets. Adding a receiver timeout handled host and serial interruptions. However, the timeout-only configuration continued to refresh the motion command when the pointer remained frozen over the forward target. Input-freshness checking was required to withdraw motion in that scenario.

Table 4
Controller ablation under identical simulated inputs. Delays are measured from fault onset to a neutral command. Held denotes a command still active at the 5s horizon.

Controller	120 ms entry	Target exit	Frozen pointer	Host stall	Serial loss
Baseline	Activated	Held	Held	Held	Held
Dwell and target exit	Blocked	0.00 s	Held	Held	Held
Dwell plus timeout	Blocked	0.00 s	Held	0.24 s	0.24 s
Full supervisor	Blocked	0.00 s	0.12 s	0.24 s	0.24 s

The trace-specific delays follow directly from the scheduling phase. The last active refresh before host or link failure occurred at 1.44 s and expired at 1.74 s, producing a 0.24s post-fault delay. The last fresh sample before pointer freezing occurred at 1.47s and reached the 0.15s age threshold at 1.62s. Other fault phases change these delays. For communication cessation, the general modeled receiver bound remains 0.301s after the last accepted command.

4.3 Sensitivity to motion dwell

For the assumed transient median of 0.15 s, numerical integration gave activation probabilities of 99.943%, 10.848%, 0.937%, and 0.129% at dwell durations of 0, 0.30, 0.60, and 0.90 s, respectively. Corresponding Monte Carlo estimates were 99.941%, 10.750%, 0.982%, and 0.134%. All twelve parameter combinations met the numerical consistency tolerance against quadrature. This agreement supports the reproducibility of the probability calculations under the stated assumptions.

At 0.60s dwell, modeled transient activation ranged from 0.034% for the 0.08s median to 11.587% for the 0.30s median. Thus, the same threshold produced markedly different rejection performance as the assumed occupancy distribution changed. For a stable intended target, the activation interval was 0.60–0.63s before transport and actuation. Increasing dwell reduced transient activation at the cost of increased command-entry delay. The simulation does not identify an optimal threshold for a particular user population.

4.4 Stopping-distance sensitivity

At 9600 baud and assumed 8N1 framing, transmission of one command byte requires approximately 1.04 ms of serial line time. This accounts for only one part of the command path. USB scheduling, host processing, servo movement, and wheelchair dynamics also contribute to response. Gaze acquisition and dwell precede that path and are excluded from the preliminary click-to-actuator measurement.

At an assumed speed of 0.50 m/s, Eq. (4) produced a stopping envelope of 0.775–1.125 m. The central case, with $T = 3.5$ s and $\tau = 0.3$s, was 0.950 m. At 1.00 m/s, the envelope increased to 1.550–2.250 m. Under the less specific nonaccelerating-speed assumption, the 4 s upper comparison was 2.00 m at 0.50 m/s. Table 5 separates these post-STOP distances from travel accumulated during the modeled receiver timeout.

Table 5
Conditional stopping distances and additional travel before timeout-triggered STOP. The timeout term uses 0.301 s and is excluded from the stopping envelope.

Assumed speed in m/s	Linear envelope in m	Central case in m	Nonaccelerating 4 s bound in m	Additional timeout term in m
0.25	0.388–0.563	0.475	1.000	0.075
0.50	0.775–1.125	0.950	2.000	0.151
0.75	1.163–1.688	1.425	3.000	0.226
1.00	1.550–2.250	1.900	4.000	0.301

The 0.301s receiver bound corresponds to an additional travel term of approximately 0.151m at 0.50 m/s. Applying a 0.60s dwell to STOP would add 0.30 m at the same constant speed before STOP was issued. The latter calculation supports bypassing motion dwell for STOP, although the user still needs time to acquire the stop target visually. Neither calculation establishes a validated clearance distance.

5 Discussion

5.1 Complementary roles of selection and command expiry

The combined results show that successful selection and timely command withdrawal address different aspects of gaze-controlled mobility. Preliminary testing demonstrated that the prototype could translate intended selections into commands. Fault injection revealed that the baseline command could persist after input exit or interrupted updates. In a controller that holds the last motion request, an omitted selection during movement can leave an earlier direction active. Aggregate selection accuracy therefore needs to be complemented by explicit measures of command persistence and neutralization.

The ablation identifies the mechanisms required by the evaluated control policy. Dwell handles brief target occupancy, neutral-on-exit handles a visible departure from the targets, freshness checking handles invalid upstream input, and receiver expiry handles communication silence. Their effects are complementary. In particular, an active communication channel can repeatedly transmit an outdated request, preventing a receiver timeout from expiring. Preserving sensing age separately from message age is therefore essential to the frozen-input response demonstrated here.

These findings concern the modeled command state. The supervisor did not alter the eye tracker’s recognition performance or the wheelchair’s braking dynamics. Its computational advantage was the

withdrawal of motion authorization under specified faults that left the baseline active. Hardware testing is needed to determine how this command-level behavior propagates through servo movement, joystick neutralization, and wheel motion.

5.2 Interaction tradeoffs and implementation requirements

Motion dwell provides a tunable means of rejecting short target entries, but its appropriate value depends on actual user behavior. The sensitivity analysis produced more than two orders of magnitude of variation in activation probability across the assumed transient medians at 0.60 s dwell. This dependence argues for measuring target-occupancy distributions and intended activation times before selecting operational thresholds. The synthetic distributions provide controlled comparisons rather than a substitute for those measurements.

Neutral-on-exit also changes the interaction experience. Users must look away from control targets to inspect their environment, and a strict exit policy can produce frequent stopping. Longer dwell can then increase the time required to resume movement. These usability effects should be evaluated alongside unintended activation, workload, fatigue, and task completion. Alternative interaction methods, including those studied in EyeTAP and crossing-based selection [4, 5], may be relevant where the user's abilities permit them.

Freshness checking requires an upstream interface that exposes sample validity and acquisition time. A stationary valid fixation must remain distinguishable from a frozen pointer. The physical implementation should define the response to missing eyes, invalid samples, calibration loss, and application focus changes. Receiver-side expiry should run independently of the host, while arming and controlled rearming should govern the transition back to motion after a fault. These functions extend the computational supervisor into an operational control system.

5.3 Mechanical response and travel before rest

A neutral servo command does not verify that the joystick has physically returned to neutral. Linkage binding, calibration error, servo saturation, and supply faults can prevent the requested displacement. Consequently, command logs should be complemented by measurements of joystick position and wheel motion. An independently accessible stop path and a validated neutral-return mechanism are also relevant to faults outside the host's control.

The stopping model illustrates why subsecond actuator response can coexist with substantial travel before rest. At the hypothetical speed of 0.50 m/s, the linear model yielded approximately 0.78–1.13 m after STOP receipt, with a further timeout term before STOP. The difference between the linear envelope and the less specific nonaccelerating bound demonstrates sensitivity to the assumed braking trajectory. Synchronized speed and position measurements are needed to replace that assumption with a fitted physical model.

Risk evaluation must therefore connect input failures, communication faults, actuator faults, and stopping behavior to explicit controls and verification outcomes. ISO 14971 provides a general framework for medical-device risk management [10]. The present study evaluates selected control properties within that broader problem and does not establish conformity to a product or risk-management standard.

5.4 Limitations and further validation

The preliminary functional assessment is limited by ten aggregate selection attempts, approximate timing ranges, and a short continuous-operation observation. Participant count, participant characteristics, trial order, and direction-specific outcomes were unavailable for analysis. The confidence intervals assume independent attempts and cannot characterize population-level performance. Tracker and software configuration also require complete specification for independent reproduction of the physical prototype.

The computational harness preserves the baseline pointer-processing method but idealizes GUI geometry, host timing, communication transport, and receiver execution. It does not execute the Arduino libraries, tracker driver, or physical wheelchair. The supervisor assumes valid timestamps and a functioning receiver clock. Its scenarios isolate selected faults and do not represent combined failures, malformed packets, receiver malfunction, or mechanical jams. The dwell distributions are hypothetical, and the stopping model assumes a deceleration profile and transfer of the preliminary time range across hypothetical speeds.

Further validation should first characterize the linkage, neutral tolerance, supply behavior, and command response on the bench. Instrumented wheelchair trials should then measure command-to-neutral and command-to-rest behavior under defined speed, payload, and surface conditions. After appropriate institutional review and engineering verification, evaluation with intended users should examine selection outcomes, unintended activation per exposure time, stopping distance, task completion, caregiver intervention, and usability. Repeated observations should be analyzed with participant-level dependence accounted for. These stages would establish whether the command-level benefits translate into usable and dependable mobility.

6 Conclusion

This study presented iMove, a gaze-operated mechanical joystick retrofit, and evaluated temporal supervision of its motion commands. Preliminary testing yielded eight successful selections in ten attempts. Executable fault injection showed that the baseline controller could activate on brief target entry and retain motion after input or communication interruption. The full supervisor rejected a 120 ms entry and neutralized the command after stale-input, host, and serial faults in the constructed scenarios. Ablation established that input freshness and receiver timeout provide different fault coverage, because continuing messages can conceal stale user input. Dwell sensitivity quantified the tradeoff between transient rejection and intended activation delay, while the kinematic analysis demonstrated the

contribution of command timing and braking to travel before rest. These results provide a reproducible computational basis for integrating deliberate selection with bounded command persistence. Instrumented hardware testing and intended-user evaluation are required to establish physical performance and usability.

Declarations

Funding: This research received no external funding and was conducted as part of an undergraduate graduation project at Kuwait College of Science and Technology, Kuwait.

Clinical Trial Number: Not Applicable.

Consent to Publish: Not Applicable.

Ethics and Consent to Participate: Not Applicable.

Ethics Statement: All prototype testing, including operation of the eye-tracking interface and generation of wheelchair control commands, was performed by the developers themselves. No individuals outside the development team were recruited or involved as test participants.

Consent to participate: Prototype testing was conducted exclusively by members of the development team. No individuals outside the team were recruited or involved as test participants. All team members involved in testing were informed of its purpose, procedures, and potential risks and freely provided informed consent before participating.

References

1. World Health Organization. Disability. Fact sheet, 7 March 2023. WHO source. Accessed 9 September 2026.
2. Elliott MA, Malvar H, Maassel LL, et al. Eye-controlled, power wheelchair performs well for ALS patients. Muscle Nerve. 2019;60(5):513–9. Author institution publication record.
3. Dahmani M, Chowdhury MEH, Khandakar A, Rahman T, Al-Jayyousi K, Hefny A, Kiranyaz S. An intelligent and low-cost eye-tracking system for motorized wheelchair control. Sensors. 2020;20(14):3936. 10.3390/s20143936. . Accessible author manuscript.
4. Parisay M, Poullis C, Kersten M. EyeTAP: A novel technique using voice inputs to address the Midas touch problem for gaze-based interactions. arXiv preprint, version 2, 2020. 10.48550/arXiv.2002.08455
5. Riou S, Schwab D, Bérard F. Interactions par franchissement grâce a un système de suivi du regard. arXiv preprint. Author manuscript; 2025.
6. Xu Y, Wang Q, Lillie J, Kamat V, Menassa C, D'Souza C. CoNav Chair: Development and evaluation of a shared control-based wheelchair for the built environment. arXiv preprint. 2025.

10.48550/arXiv.2507.11716.

7. PyAutoGUI contributors. PyAutoGUI documentation. Mouse-position API and fail-safe behavior. Official documentation. Accessed 9 September 2026.
8. NIST/SEMATECH. e-Handbook of Statistical Methods. Section 7.2.4.1, Confidence intervals. NIST methods reference. Accessed 9 September 2026.
9. SciPy community. scipy.stats.binomtest. SciPy API reference. Official documentation. Accessed 9 September 2026. Computations used SciPy 1.17.0.
10. International Organization for Standardization. ISO 14971:2019. Medical devices — Application of risk management to medical devices. Geneva: ISO; 2019. Official standard record.

Figures

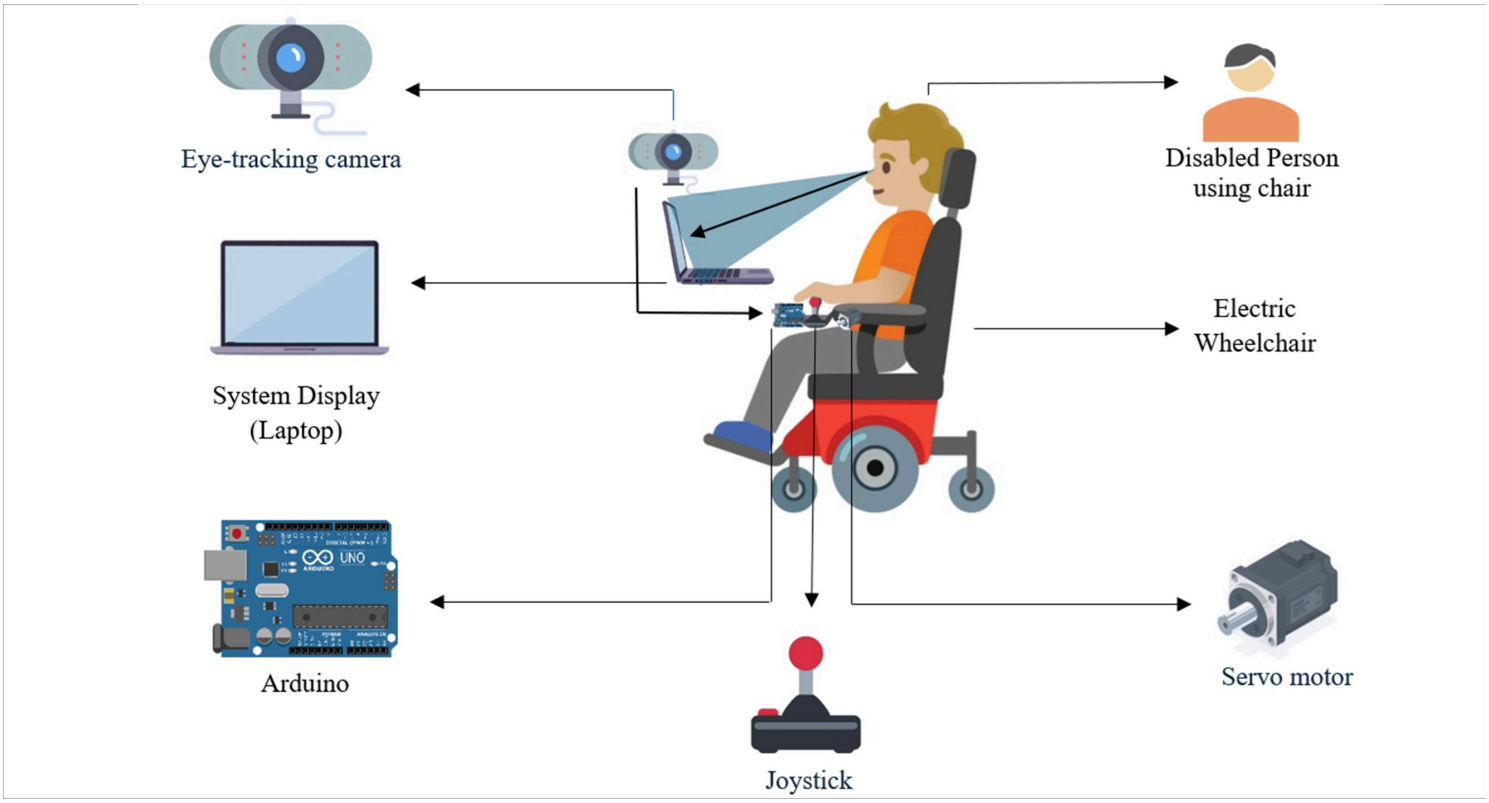


Figure 1

Proposed System Design

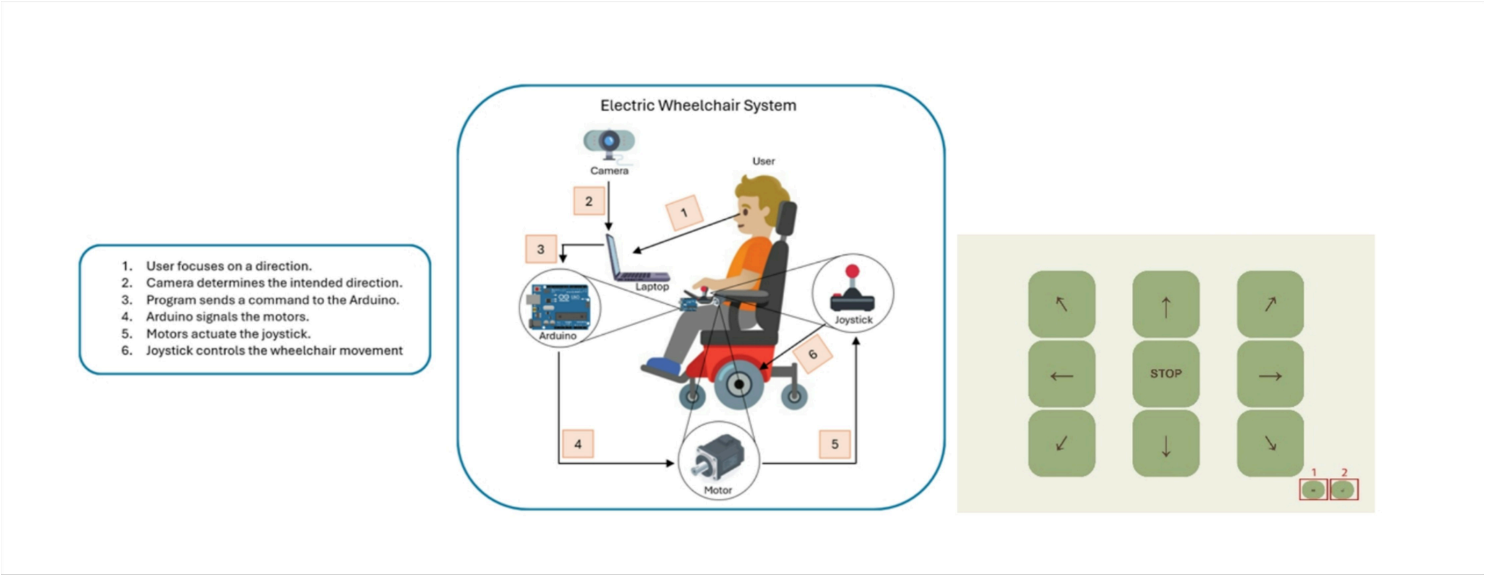


Figure 2

The step-by-step usage process and the user-interface

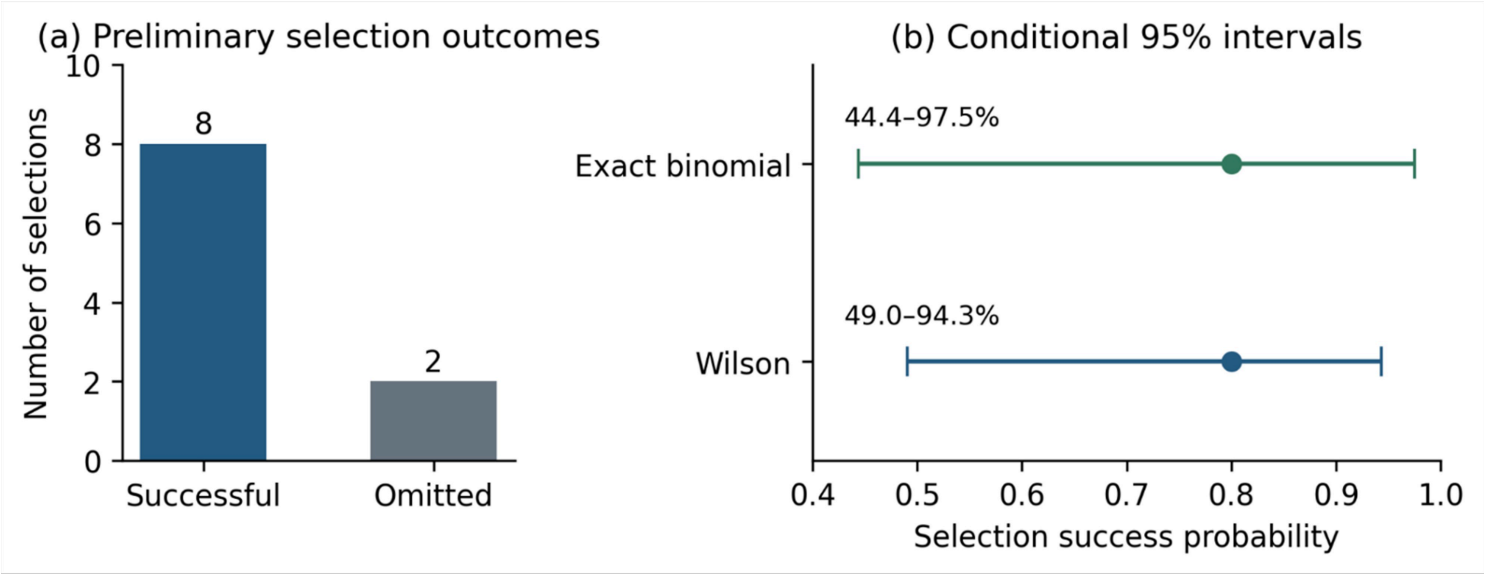


Figure 3

Preliminary command-selection outcomes. Panel a shows eight successful selections and two omissions. Panel b gives 95% Wilson and exact binomial intervals under an independent-attempt model; participant clustering and learning are not represented.

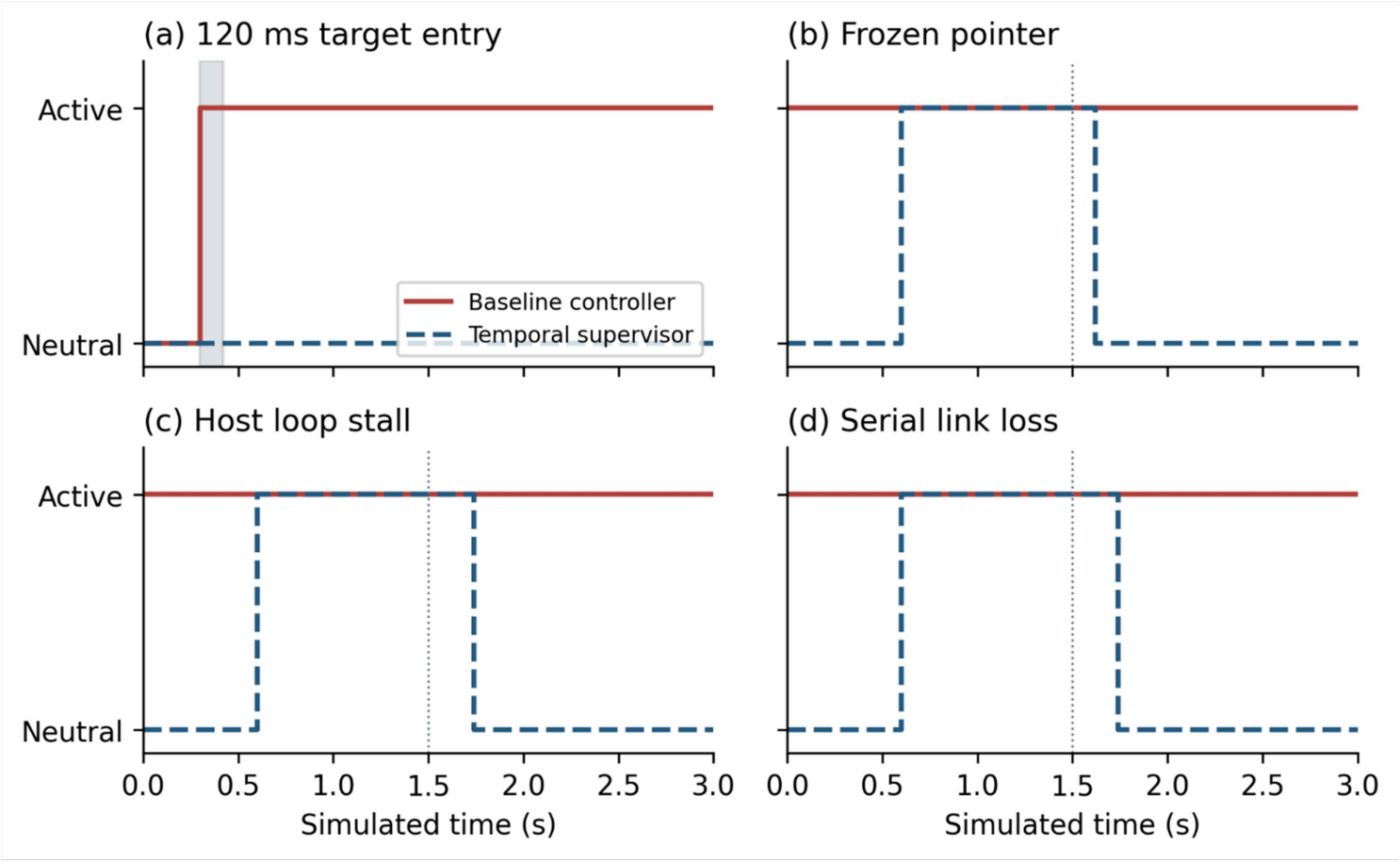


Figure 4

Simulated command states for the baseline and temporal supervisor. The shaded region in panel a denotes a 120 ms target entry. Dotted lines in panels b–d mark fault onset at 1.50s. Active denotes a nonneutral software command. Traces are displayed to 3 s; each simulation continued to 5s.

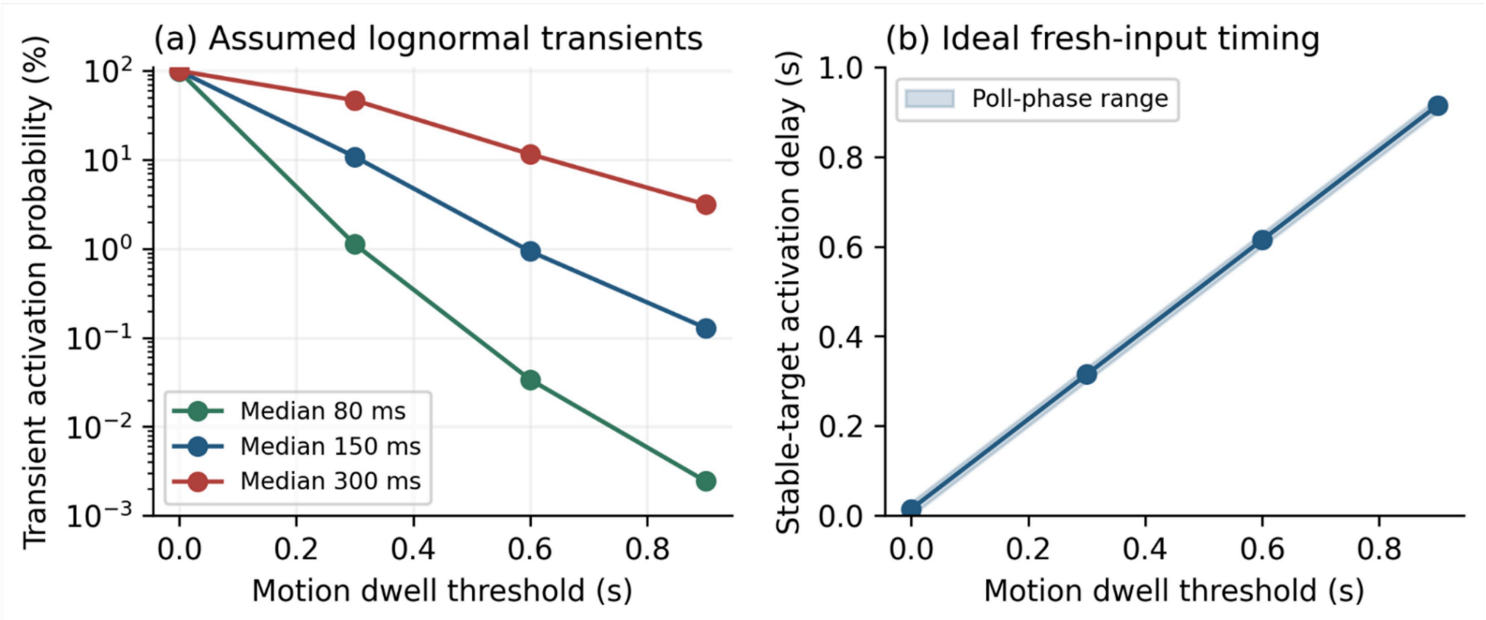


Figure 5

Modeled tradeoff between transient rejection and activation delay. Panel a shows quadrature activation probabilities for three assumed lognormal occupancy distributions on a logarithmic scale. Panel b shows the mean and host-evaluation phase range of stable-target activation time, excluding transport and actuator response.

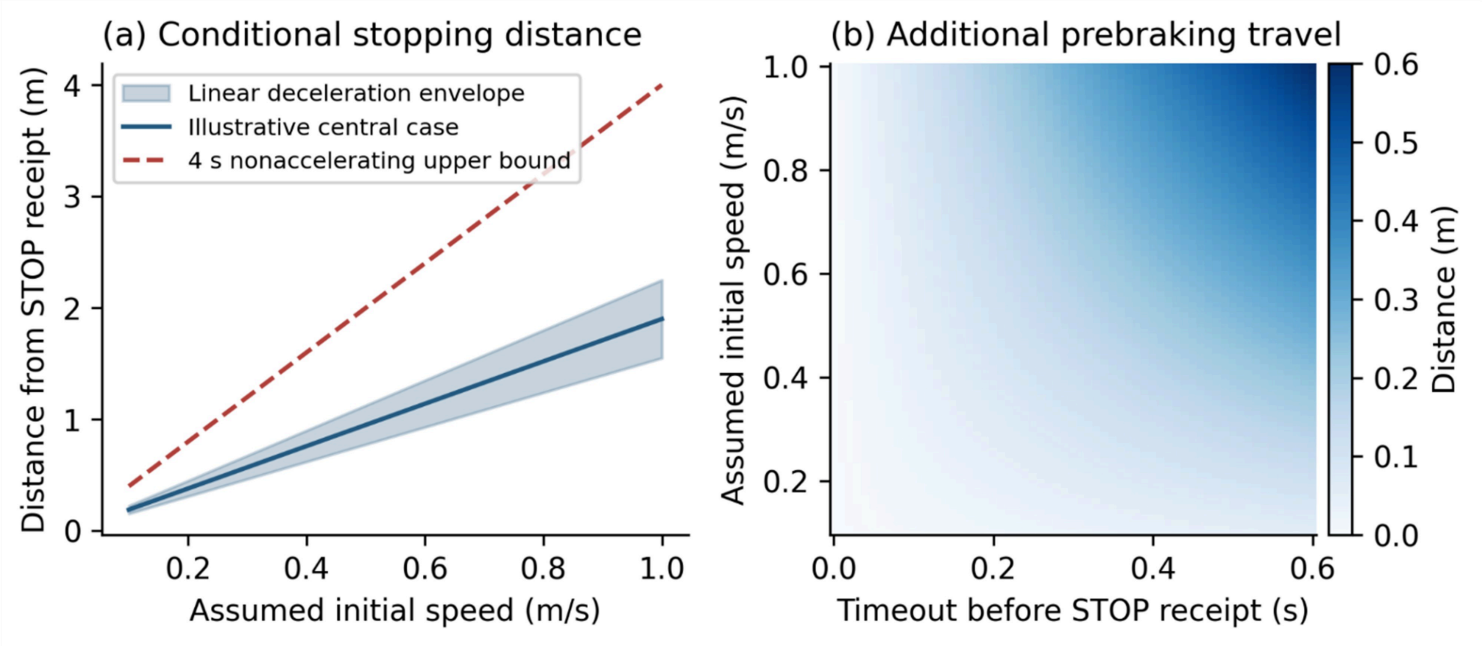


Figure 6

Kinematic sensitivity of travel before rest. Panel a combines the approximate 3–4 s stopping interval with hypothetical speeds and pre-deceleration delays. Panel b isolates additional travel while awaiting timeout-triggered STOP at constant initial speed. The curves are conditional calculations, not measured stopping trajectories.