Supplementary Information for

From Adaptive Locomotion to Predictive Action Selection – Cognitive Control for a Six-Legged Walker

Malte Schilling,1* Jan Paskarbeit,2 Helge Ritter,1 Axel Schneider,2 Holk Cruse3.
(1) Neuroinformatics Group, Faculty of Technology, Bielefeld University, D-33501 Bielefeld, Germany
(2) Bielefeld University of Applied Sciences, D-33615 Bielefeld, Germany
(3) Biological Cybernetics, Faculty of Biology, Bielefeld University, Bielefeld, Germany
* - corresponding author: Malte Schilling

Email: mschilli@techfak.uni-bielefeld.de
ORCID: Malte Schilling – 0000-0002-0849-483X

The supplement shows further results and provides more information concerning the cognitive expansion. First, stable locomotion is shown in dynamic simulation at different velocities demonstrating the adaptivity and robustness of the controller. Second, on the real, physical robot different disturbances are applied and it is analyzed how the cognitive expansion resolves problematic postures. Third, a systematic analysis of different starting postures (done in dynamic simulation) is provided. Last, a schematic of the controller is provided showing the cognitive expansion and how the dynamics of the network converge during selecting an action out of its original context.

This PDF file includes:

Supplementary text
Figures S1 to S7
Tables S1 to S3
Legends for Movies S1 to S3

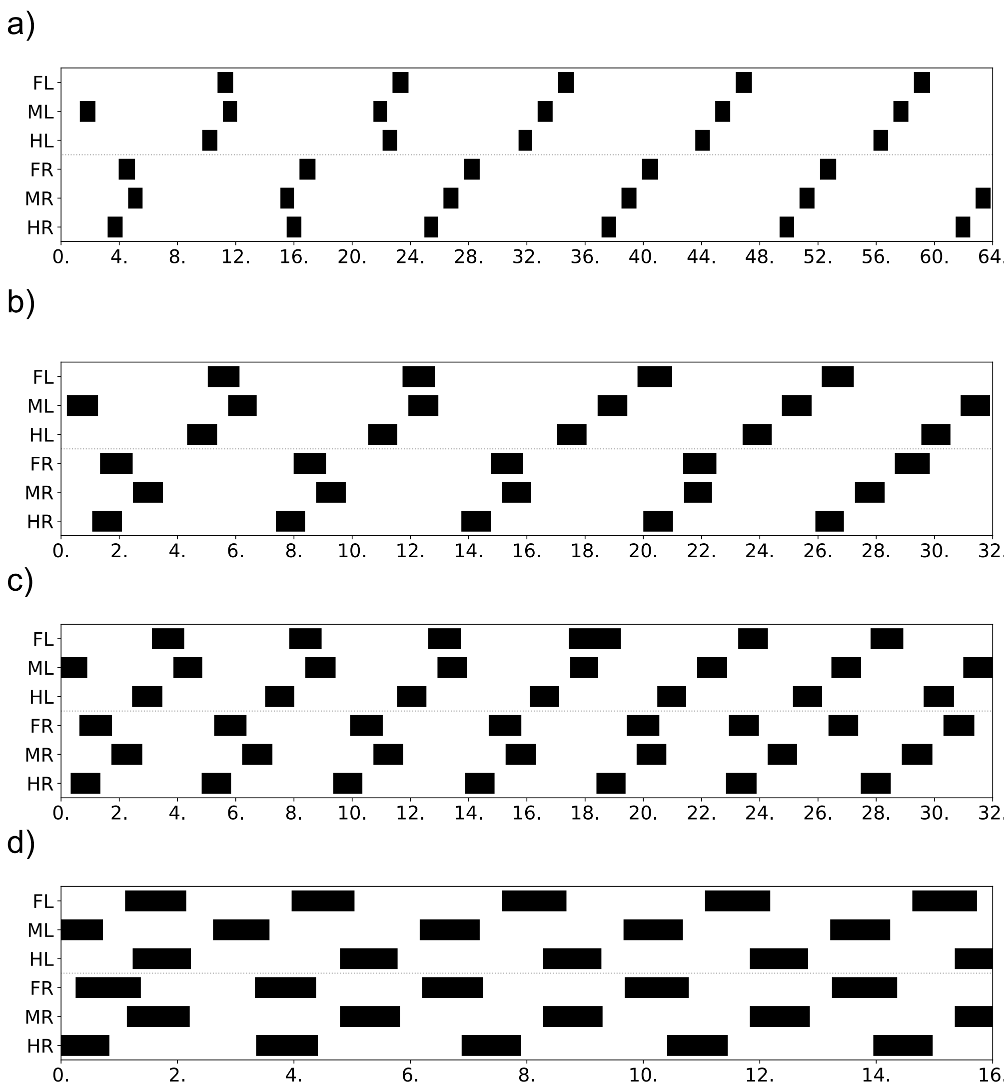
Other supplementary materials for this manuscript include the following:

Movies S1 to S3

Results

1) Stable emergent locomotion

Fig. S1 shows results from the dynamical simulator of the Hector robot. Different velocities are applied (this affects only the stance velocity, swing movements do not depend on walking velocity). Different walking patterns emerge as can be observed in locomotion of insects. See Video S2 for example simulation runs. To illustrate a more complex case, Fig. S2 shows an example of curve walking performed in an earlier simulation (Schilling et al 2013b), which involves intersegmental drives between the different body segments not realized on the current robot platform and its simulator.



e)

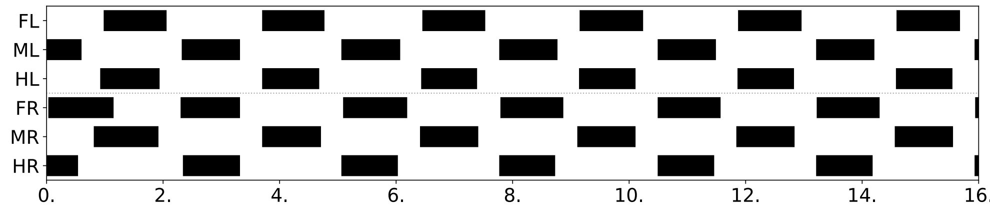


Fig. S1: Footfall pattern for simulated Hector walking straight with different velocities: ranging from very slow to fast walking. Velocity factor (parameter default_speed WalknetSettings.py) in a) = 0.004, b) = 0.008 (wave gait emerges), c) = 0.012, d) = 0.016 (tetrapod gait pattern), e) = 0.020 (tripod gait). Black bars indicate swing movement of the respective leg: front, middle and hind left leg, front, middle and hind right leg, from top to bottom. Abscissa is simulation time in seconds.

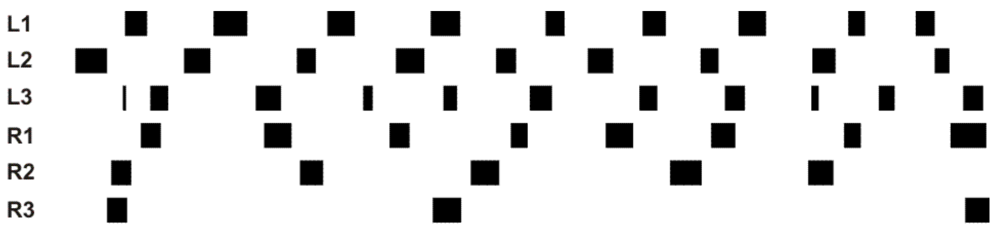


Fig. S2: Footfall pattern for curve walking (turn towards the right), black bars indicate swing movement of the respective leg, abscissa: simulation time, the lower bar indicates 500 iterations corresponding to 5s real time. Simulation data is taken from [Schilling et al. 2013b].

2) Disturbance of the physical robot walking by inducing long steps

As illustrated in Fig. 3, searching movements in the real robot Hector induced long steps far towards the front. This forced movement disturbed coordination and lead to instable walking situations. Table S1 and Fig S3 provide results on how the system coped with such cases. We systematically varied prolongation of middle leg and hind leg swing duration by 0.0 – 0.10 m (normal step amplitude is 0.16 m, which is quite large for legged robots of this size) and observed how the system adapts. For each condition, we produced two robot runs (Table S1, see also video S3 showing two example runs).

Table S1: Robot Hector's reaction to disturbances.

Leg being disturbed	Disturbance amplitude [m]	Adaption of the system
ML	0.05	In both runs, stable gait pattern emerged immediately
	0.06	In both runs, stable gait pattern emerged immediately
	0.07	Run 1: After the disturbance, the first swing movement of HL produced an instability. This was resolved by a step backwards of ML. Run 2 led immediately to stable gait.
	0.08	Run 2: After the disturbance, the first swing movement of HL produced an instability. This was resolved by a step backwards of ML. Run 1 immediately led to stable gait.
	0.09	Both cases: After the disturbance, first swing movement of HL produced an instability. This was resolved using a step backwards of ML.
	0.10	In both cases, after the disturbance, the first swing movement of HL produced an instability. This was resolved using a step backwards of either HR (run 1, Fig S3a) or ML (run 2, Fig S3b).
MR	0.10	Both runs: stable gait pattern emerged immediately
HL	0.10	Run 1: After the disturbance, the next swing movement of HL produced an instability. This was resolved by a step backwards of ML. Run 2 led immediately to stable gait.
HR	0.10	After the disturbance, in both cases the first swing movement of HL produced an instability and the system applied a step backwards of ML. However, this lead to a further disturbance followed by further internal simulations. In run 1, the robot struggled with this task and, after every second step, started a new internal simulation (observed for three more planning phases, see Fig. S3c). In run 2, the system chose by coincidence the ML step backwards again and this resulted in stable walking pattern afterwards (Fig S3d).

The results show that for small disturbances the system was not affected and stable walking patterns emerged. For larger disturbances the robot eventually became unstable and the cognitive expansion had to take over (for examples see Fig. S3). As the problem detector induces a search close to where the problem occurred, usually a preferred solution was found and only rarely another solution was selected. The case of disturbing the hind right leg underlines the difficulty of this problem: while a disturbance is resolved through applying an action out of context, this necessarily interrupts the coordination pattern that has emerged up to that point in time. In most other cases, this was not problematic as the system converged very fast again towards a walking pattern that allowed for stable walking. This is due to the adaptivity of the underlying decentralized control architecture.

Figs. S3a) and b) show two different solutions for a long step (disturbance 0.10 m) of the middle left leg. The robot became instable when trying to lift the hind left (HL, briefly before the grey shaded area). Therefore, walking was interrupted and a searching procedure was started (grey area). In a) it tried three unsuccessful actions (red area) before resorting to move the hind right leg backwards (green area) which unloaded the hind left leg and normal walking could continue (blue area). Here, a difference between internal simulation (green area) and testing behavior (blue area) is visible: the swing movement is much shorter on the physical robot, but the emerging pattern afterwards appears the same. In b) the robot runs into the same problem, but instead, in the second trial, moved the middle left leg (ML) backwards. This solved the problem. Therefore this solution could be applied by the real robot (blue area). Figs. S3 c) and d) show a long step induced in the hind right leg (HR). The robot became instable at the next lift-off of the hind left leg (beginning of first grey area, note the different time scales in c) and d)). The solution found when the first problem (in both figures from 5 to 12 seconds) was detected (step backwards of the middle left leg, in the second trial in c, in the third trial in d)) led a couple of seconds later to another instable posture. While in c) the robot always selected actions that solved only the immediate problem at hand and did not find a long term solution (at least in the time window observed), in d) the system repeated the middle left leg step backwards, but, in the second internal simulation, selected an early step to the front of the middle left leg (purple bar, initiated by the cognitive expansion). This reestablished a stable gait pattern.

To summarize, walks with the physical robot showed considerable variations, even for the same starting positions. Nonetheless, in nine cases the reactive system overcame the disturbance. For the remaining nine cases the cognitive expansion was able to solve the problem.

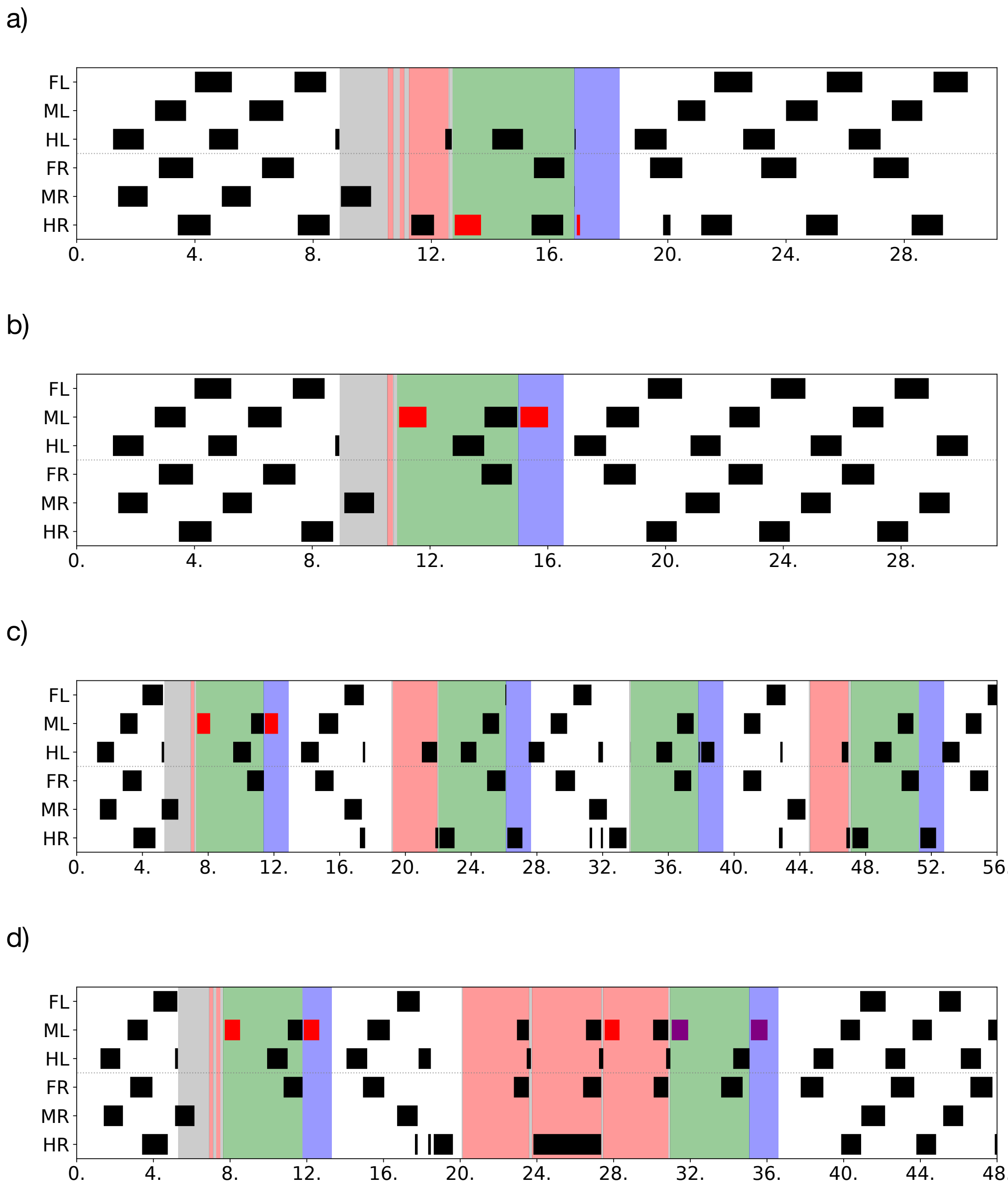


Fig. S3: Footfall pattern for robot Hector walking with a forced long step (velocity 0.016). Black bars indicate swing movements of the different legs over time; dark red bars indicate swing movements backwards during internal simulation (green and red shaded area) or testing of behavior on real robot (blue shaded area). In color shaded areas, the cognitive expansion took over: grey represents initially stopping and selecting an action, red shows unsuccessful testing of that action in internal simulation and green shows a successful internal simulation which finally was applied on the real robot shown in the blue area. Note that during internal simulation, i.e. areas colored grey, red, or green, the real robot is not moved at all. At the end of each internal simulation (either red or green area) the internal model is reset to the real posture of the robot. This means that for each subsequent red, green and blue area (when switching back to the real robot), the initial posture is the same.

3) Variation of starting posture

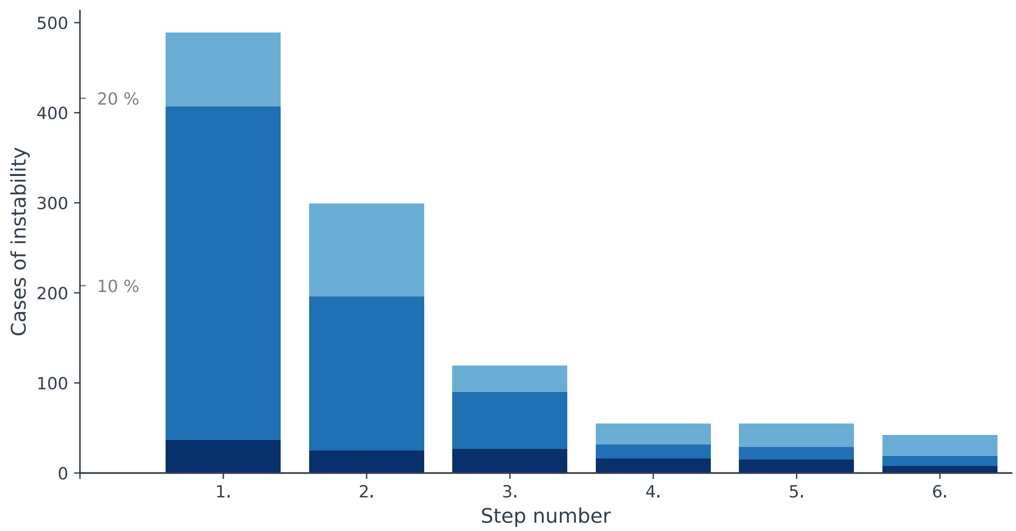
We tested systematic variations of the starting posture, at first for the decentralized architecture without using the cognitive expansion to obtain a coarse estimate of how many different starting positions lead to a stable walking pattern. In a second step, we compared this to the architecture that includes the cognitive expansion and analyzed how the cognitive expansion dealt with the problematic cases.

For each leg four different starting postures were assumed that were equally spaced from the front (anterior extreme position) towards the back (directly in front of the posterior extreme position). This poses a quite challenging task for a controller, as in many cases phases between leg controllers initially differed substantially from a typical, stable walking pattern. Overall, we ended up with 2080 different starting postures (4^6 minus all symmetric configurations with respect to the body axis). The simulated robot was set into the different starting postures and the controller took over control with a defined velocity. A posture was determined instable when the center of gravity left the polygon spanned by the standing legs.

To illustrate the difficulty of the task: in normal walking (at a fast velocity) neighboring controllers are assumed to be in anti-phase relation. In contrast, from the 2080 initial postures, 1216 are defined with in phase relations between neighboring legs (even when excluding middle leg symmetries still 928 initial postures are characterized by phase relations that would cause instabilities when maintained during walking).

The number of instabilities, and correspondingly the durations of instabilities decreased strongly during the first couple of steps (Fig. S4 and Table S2, S3; data from 2080 different starting positions). After three steps, mostly only brief static instabilities could be observed. For an intermediate walking velocity (Fig. S4 b, Table S3), there are less frequent instabilities, but the same trend can be observed: over time the controller emerges towards stable gaits.

a)



b)

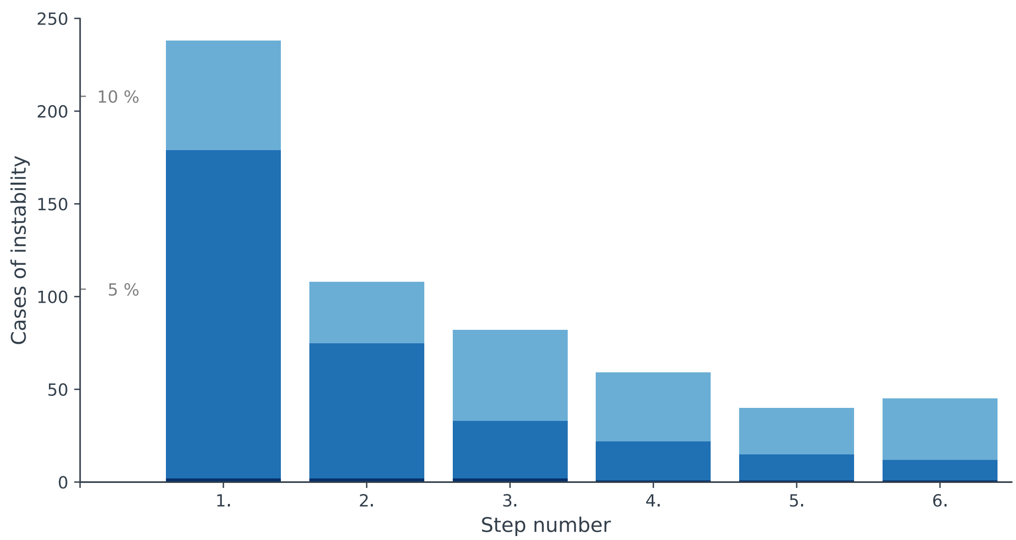


Fig. S4: **Instabilities of the control architecture** (without using the cognitive extension) when forced into systematically varied starting postures. Abscissa: number of robot steps. Light Blue: duration of instabilities between 10 ms and 100 ms; blue: duration longer than 100 ms, but only appearing during a single step cycle of that particular leg; dark blue: long instabilities (longer than 100 ms) and found in subsequent steps of the walking robot. a) high velocity (0.020) b) intermediate velocity (0.016).

Table S2: Cases of instability for tripod (fast) walking.

Step number Duration of instability	1	2	3	4	5	6
10–90 ms (light blue)	82	103	29	23	26	23
Instability >= 100 ms (but single step) (blue)	370	171	63	16	14	11
instability spanning mult. steps (dark blue)	37	25	27	16	15	8

Table S3: Cases of instability for intermediate velocity.

	1. Step	2. Step	3. Step	4. Step	5. Step	6. Step
10–90 ms Instability	59	33	49	37	25	33
Longer Instability (but single step)	177	73	31	21	14	11
Longer instability spanning mult. steps	2	2	2	1	1	1

As a next step in this third series of experiments, the instable cases were analyzed in more detail using the complete system that includes the cognitive expansion using dynamic simulation. In Fig S5 five examples are shown that illustrate the impressive adaptivity of the approach.

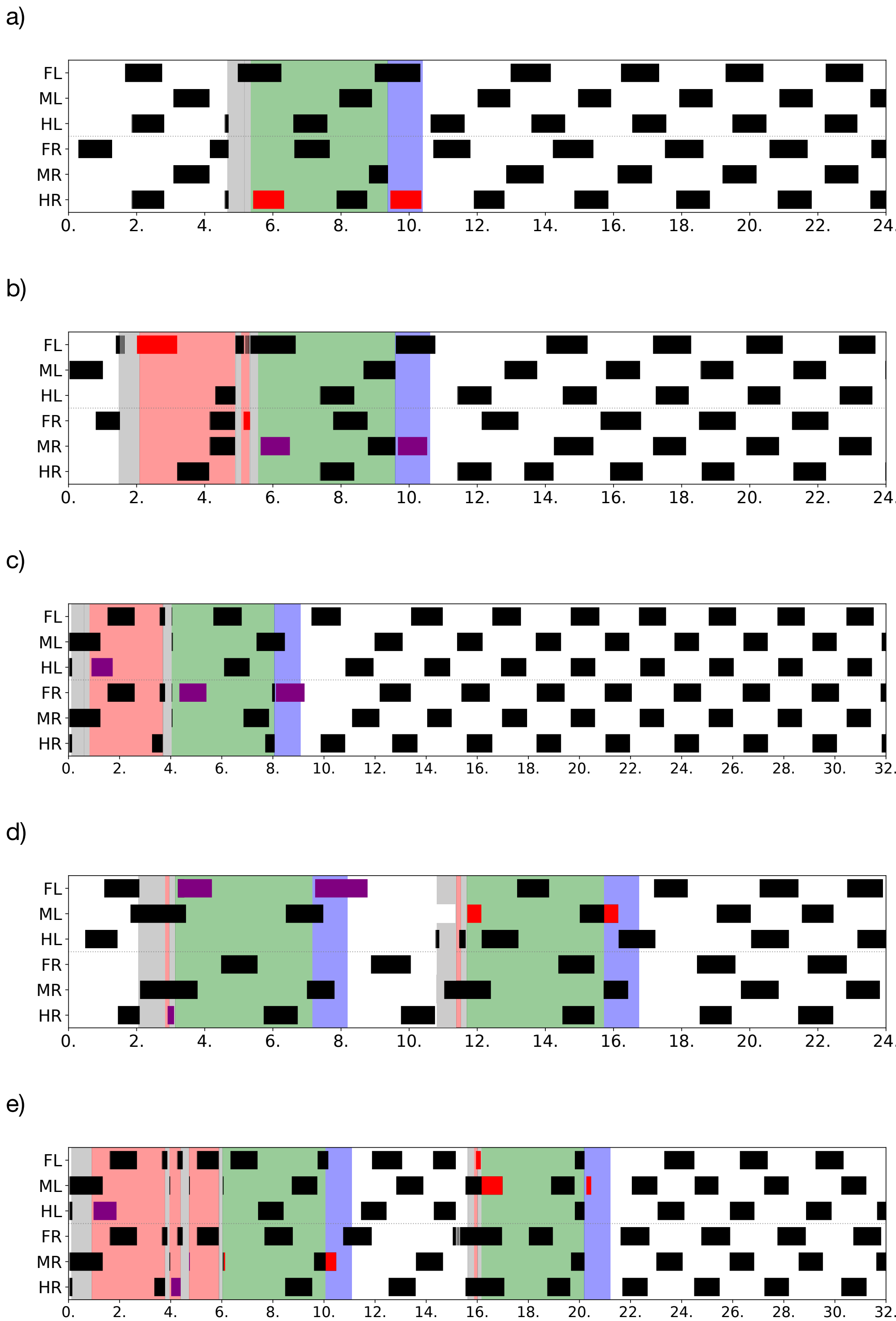


Fig. S5: Examples of difficult starting postures. In the color shaded areas, the cognitive expansion took over: grey represents initially stopping and selecting an action, light red shows unsuccessful testing of that action in internal simulation applied on the body model and green shows a successful internal simulation which was finally applied on the dynamically simulated robot (blue area). Backward swing movements in red. Forward swing movements selected by cognitive expansion in purple. For details see text.

In Fig. S5 a), hind legs were initially in phase which would result in a short instability as the robot tries to lift both hind legs at the same time during the second step. Immediately, the cognitive expansion tested a swing backward of HR, which solved the problem (successful internal simulation is shown in green area) and therefore the solution was applied to the simulated robot (blue area).
b) A long instability occurred as the robot tried to lift the front left leg (FL) while front right (FR) already performed swing movement. Therefore, the robot stopped walking and started search (grey area). But the swing movement of the front right leg was finished immediately (within grey area). A first trial (swing backward of FL, red) was not successful as it led hind left leg, front right leg and middle right leg to lift simultaneously (light red area). In a second trial, cognitive expansion tried to reposition FR backwards, again without success. The third trial (swing movement to the front at MR, marked purple, selected by cognitive expansion) was successful (green area). As can be seen, the front left leg also performs a swing movement which is still triggered by the front left leg being far to the back of its working range making this movement necessary. This solved the instability, which did not occur later-on again as both front legs were moved in a more or less anti-phase relation by the intervention. The solution was applied to real walking (blue area).
c) When this case had first been tested without access to the cognitive expansion, long instabilities occurred over several steps (not shown) because, immediately after initialization, swing movements for middle left leg, hind left leg, middle right and hind right leg were started. During a run of 16 s altogether instabilities for 4.56 s have been observed. If, however, as shown in Fig. S5c, the test was repeated with the cognitive expansion being available, the system was triggered by the problem detector in the hind right leg and the cognitive expansion was activated. In the first trial (left grey area followed by light red area), the cognitive expansion tried unsuccessfully to reposition the hind left leg (swing movement to the front, marked by purple). In a second trial (starting in second grey area, followed by green area), the problem was resolved by a swing movement to the front of FR (marked by purple). As the FR leg does not itself contribute to stability at the back, this appears a surprising solution: but when looking at the footfall patterns and data more closely, it becomes apparent that this solution emerges from the decentralized structure and coordination rules: selecting a swing movement for FR inactivated coordination rule 3 from FR to MR which would have elicited the early swing movement of MR. Last, this solution is applied for real walking (blue area).
d) Multiple internal simulations: During the first steps the robot tried to lift MR, while HR and ML have already been lifted, which led to an instability (grey area followed by light red area). Importantly, during stopping of the simulated robot, MR and ML did not finish their swing movements, but the HR finished its swing movement (as it was already nearly finished). This allowed the cognitive expansion to select HR (first trial, grey area followed by light red area) to try another swing movement of HR to the front (marked purple) which caused another instability. In the second trial, the cognitive expansion selected a swing movement of FL to the front (marked purple) which provides a solution. Although this first instability could be solved, after about 4 s of walking (at about 11 s) the same instability occurred during the next step of the robot. This is not surprising as the phase relations between the hind legs and middle legs had not been altered by the intervention of the cognitive expansion, and, in this case, the decentralized structure could not resolve the problem over the short time span. But during this second internal simulation the cognitive expansion selected a swing backwards of ML (green shaded area, swing shown in red) which resolved the instability and was then applied on the robot (blue shaded area).
e) In the initial leg configuration, both middle and both hind legs were placed very far back and, therefore, immediately tried to produce swing movements leading to instability. Initially, the cognitive expansion tried, without success, multiple different actions (first trial, swing movement to the front of HL, marked purple; second trial, swing movement to the front of HR, marked purple; third trial, short swing movement to the front of MR at about 5 s, marked purple), but none of these solved the problem. The system found as a solution swinging the right middle leg backwards (at about 6 s, as the leg was already in swing, but far to the back of the working range, this short swing movement, marked red, is difficult to recognize). This successful movement (green area) is then applied on the robot (blue area, about 10 – 11 s). At this point, a difference is

notable between internal simulation and the (dynamically simulated) robot: switching direction of the ongoing swing movement of MR takes much more time in the dynamic simulation (blue area, MR, red). Nonetheless, this provides a solution for the problem at hand. Some steps later-on, at about 16 s, a further planning stage is required to resolve the difficult initial leg configuration. During this planning stage, first, the cognitive tried to move the front left leg backwards (shown as a red bar at around 16 seconds in small, red shaded area) which lead again to an instability with the robot threatening to tilt to the front. Secondly, the middle left leg (ML, red, in light red area) was moved backwards. This was then successfully tested also on the robot leading to the emergence of a stable walking pattern.

Cognitive Expansion

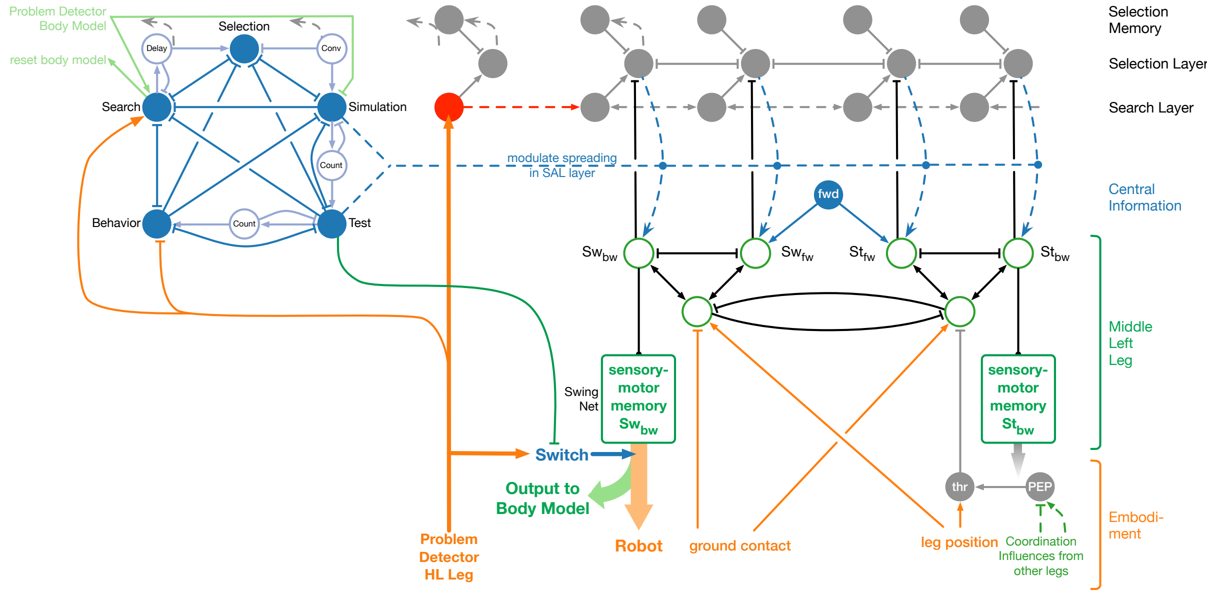


Fig. S6: Detailed leg control network (middle left leg) and cognitive expansion. The blue units (upper left) show the small global network that regulates the state of the whole architecture and represents the different stages. The right section shows a problem detector of the HL (orange), active when an instability is detected, and the controller of the left middle leg in more detail. Lower part (green, orange): local controller of ML. Upper part (grey units): three layers that form the local cognitive expansion, which selects an appropriate new sensory-motor memory. The Switch (blue) decides if the motor output is routed to the body model (for predictive internal model/simulation, green arrow) or to the motors to run the robot (orange arrow). Normally, behavior is switched on and the robot is walking. When a problem is detected the network switches towards search (orange, inhibition of behavior and activation of search). This initiates selection of another action: first, information on where the problem occurred is briefly spread in the Search Layer (grey units, top right). Second, a single activation is selected in the WTA selection layer (grey units). These units are inhibited by the Selection Memory which remembers which behavior was already tried in internal simulation. After a selection, internal simulation is run for a given time which is regulated by the small state network (shown at the top left, blue unfilled circles represent units that measure timing). When a problem occurs during internal simulation, simulation is inhibited and a new search is initiated (light green arrows are fed back from internal model and affect the current state of the system in the network shown in blue at the top left). After successful internal simulation, it is switched to the stage “Test”: the switch is operated again and commands of the successfully simulated action are now routed to the robot. When successful, normal walking continues (stage “Behavior”). Motivation units for swing back (SWbw), swing forward (SWfw), stance back (STbw), stance forward (STfw). These units are centrally activated by other motivation units (blue, fwd). Other motivation units (SW, ST) are stimulated by sensory input (e.g. ground contact, leg position, and coordination influences) passing a threshold (thr). Embodiment refers to both physical robot (first order embodiment) and body model (second order embodiment). Connection between units: arrows indicate excitatory influences, bars indicate inhibitory influences.

In case a problem is detected, the cognitive expansion will be activated. This, first, leads to an interruption of walking and a selection of a new sensory-motor memory including its motivation unit, which is not used in the current context. This is followed by an internal simulation to test the suitability of the new action selected. If a sensible solution is found, it will be applied to the robot.

In the following this will briefly be illustrated (see Fig S6, Figs. 2 and 3, for details the reader may be referred to Schilling and Cruse, 2017). The control system forming the cognitive expansion has different control states (called stages): During walking, the system is in the behavioral stage (Fig. S6, Behavior, lower left blue unit). When a problem is detected, the system switches to the next stage, Search. A “problematic situation” is characterized here as an unstable posture that occurs during walking and that could be detected if the leg still experiences load although it is actively unloaded (for example Fig 3, beginning of grey space). The problem detector is realized simply as a mathematical calculation of static stability (see section above). If the problem detector is activated, it stops the execution of the behavior. It further activates the switch decoupling the body from the control process (Fig. S6, right part, lower left, green bolt arrow). Therefore, the behavior causing the problem is stopped. At the same time, the problem detector is activating the search stage for a solution, i.e., it selects a new action out of its current context: this is realized in three sub-stages that are related to three sub-networks (Fig. S6 and S7, Fig. 2). For each motivation unit (green units), that represents a sensory-motor memory, there is a corresponding unit in each of the three added layers (search – selection – selection memory layer, grey units, upper right; for convergence of these networks see Fig. S7). In the “search” layer an activation is induced starting at the problem detector that caused the internal simulation. This activation spreads slowly through the layer (modulated by the stage neuron “Search”) and at first activates units close by. This favors actions that are morphologically close to the cause of the problem. Each problem detector is connected to at least one unit of the search layer there initiating the spreading activation.

After some units of the search layer became activated in a next step one unit shall be selected in the selection layer forming a winner-take-all net. The two layers are connected in a one to one fashion, i.e., for each unit in the search layer there is a corresponding unit in the selection layer. As long as the spreading activation is running, the search units’ activation transfers directly to the corresponding selection unit (again, this connection is modulated by the unit representing the search stage, modulation is not shown). Currently active behaviors and motivation units are excluded from the selection through inhibiting connections from the motivation units towards the corresponding units in the selection layer. Similarly, an active “search memory” unit inhibits activation and selection of an action. “Search memory” units become activated once an action was selected and run in internal simulation. This circumvents applying the same action twice and realizes a form of working memory (as assumed required for such processes and already present in insects (Giurfa & Menzel, 2013)).

In the winner-take-all network (selection layer) only the unit with the highest activation stays active and inhibits all other activation after the network has converged for some time (after about 10 to 20 iteration steps). When the WTA has converged it has selected a new behavior which is morphologically close to the origin of the problem and should be now applied in internal simulation. The active unit of the selection layer is activating the corresponding motivation unit (green circle) and in this way initiates the behavior. This activation of a behavior may affect not only the explicitly selected behavior, but may have direct effects on the selection of other behaviors. Crucially, during internal simulation the behavior is not carried out on the robot itself, as the motor commands are rerouted towards the body model instead of the motor system (such a decoupling is assumed in mental simulation in humans (Hesslow, 2002)). Instead it is applied on the internal model (Fig. S6, Fig. 2, green bold arrow) and the model predicts the consequences of the simulation of this behavior. Internal simulation runs for a given time: after search and selection of a behavior, the control structure switches to a “Simulation” stage (shown top left, blue unit “Simulation”). This stage is kept active for a specified time (450 iteration steps, simulating at least two steps of the robot). The stage is, however, aborted when a problem occurs in internal simulation. Importantly, the internal model is equipped with the problem detectors as given in the real agent. Only in this way the internal model can decide if the problem is still present and the search has to be started again by letting the spreading activation start over.

As soon as a sensible solution has been found during simulation, the test stage (Fig S6, top left and blue unit “Test”) is activated and the switch turns back to reroute the output of the sensory-motor memory to the motors of the robot (orange bolt arrows), so physical walking can resume again. For further details see Schilling and Cruse (2017).

In Fig. 10 an example illustrates the dynamics of some units forming the three layers (search, selection, memory) for six sensory motor memories of the cognitive expansion (Fig. S7, left). This network is modulated by the global stage units (blue units, top left in Fig. S6): First, during the search stage activation is spread out in the network (red units on the left, initial activation is induced by problem detector). Second, in the selection stage a single unit is selected in the middle layer (Fig. S7, green units that form a winner-take-all network). Activation is driven by activation of the corresponding units in the search layer. Further, the selection is inhibited by the third unit (selection memory, blue unit on right) that gets activated after an action was selected once. In this way, this layer remembers which action should not be selected a second time. In the search layer (Fig. S7, left, red units) activity spreads only during search stage, in between activity is maintained (but there is some noise). In the selection layer (green units), selection of an action is only considered during selection stage.

The right section illustrates the activation of the three units of each of the six memory elements depicted here (search, red line, selection, green line). At the beginning, memory element 2 (from above) is activated by the problem detector (search on, but no selection). Via spreading activation the neighboring units (element 1, 3, to a lesser extent 4 – 6) are activated. Due to WTA properties, unit 1 wins (red) wins and search is activated (green). This is apparently not successful, which elicits a further spreading activation, which now activates unit 4 (red, note that earlier units are inhibited via activation of selection memory (not shown). As a consequence, selection is activated (green). This apparently is not successful, too, which leads to activation of unit 5 (green).

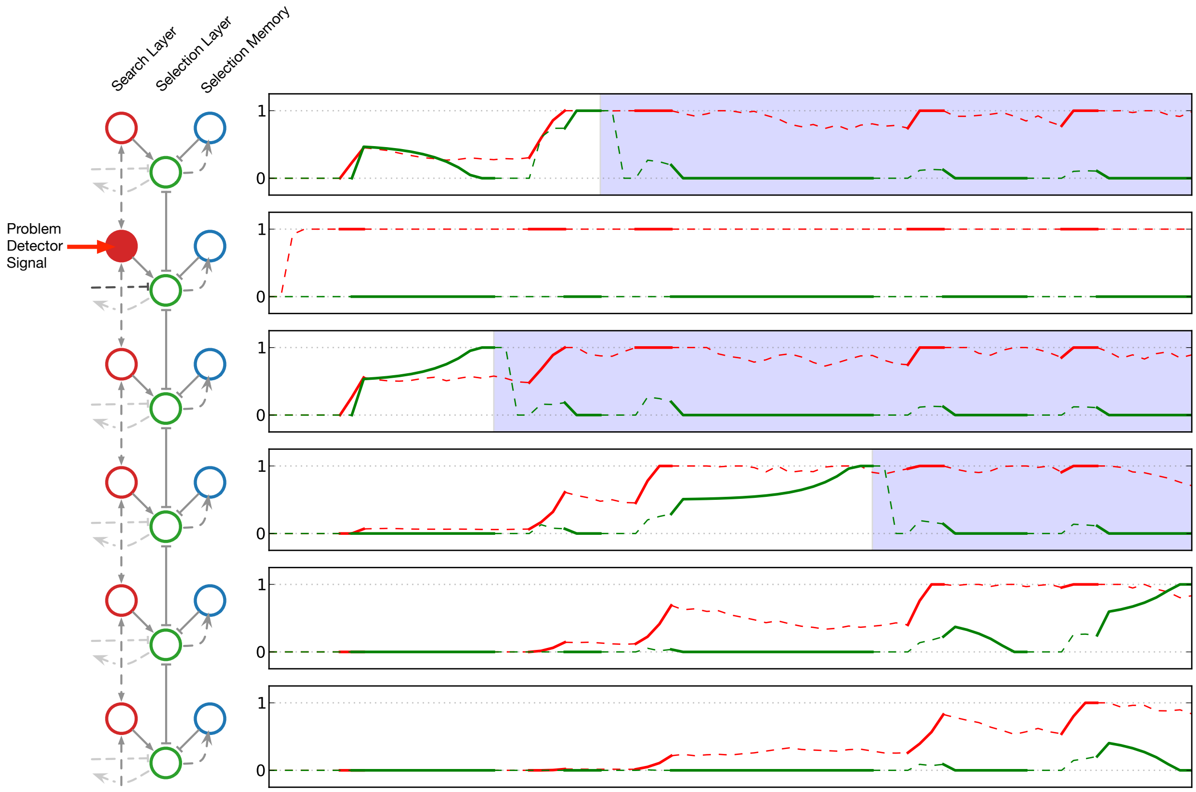


Fig. S7: Convergence of search and selection process. Illustration of the local part of the cognitive expansion that consists of three units for each behavior (sensory-motor memory) of the control network (Fig S6, grey units, upper right). On the right, the corresponding temporal activations (between 0 and 1) of the network are shown during a run of the cognitive expansion (time from left to right). Red lines show activity of search unit, green of the selection unit). Solid parts of line indicate activity during the respective stage, dashed lines show activity of the network that does not affect the behavior of the system. Blue area indicates activity of the selection memory unit after the specific behavior (shown in one row) has been selected. This inhibits the subsequent reselection (shown in green) of this action.

Simulator and Controller Implementation

The control framework and the simulator (next section) are publicly available (dynamical simulation environment is realized in C++ and based on the Open Dynamics Engine library, see https://github.com/malteschilling/hector ; the controller has been implemented in python (version 3), see https://github.com/malteschilling/cognitiveWalker). For more details on the body model see (61); for the cognitive expansion see above and (62), which showed a proof of concept in simulation, and for the neural network for action selection see (88).

Movie S1 (separate file). Robot climbing over hole in walkway requires rearranging foot position by the cognitive expansion. Hector walks on a walkway containing a hole. Shown is a perspective view (right) and a side view showing a closeup of the hole (left). Top row shows coordination of foot movements. When a leg reaches into the hole, a searching movement to the front is initiated until ground contact has been established. As a consequence, the middle left leg was moved far to the front. When the hind left leg had to be lifted, the robot got instable. Therefore, the robot was stopped and started to test different alternative behaviors in internal simulation. As a solution, the middle left leg performed a step backwards which unloaded the hind leg and it can perform a normal swing. Afterwards normal walking is resumed.

https://www.dropbox.com/s/g06gwai46igudc3/1_CognitiveSystem_Hector.mp4?dl=0

Movie S2 (separate file). Emergence of stable gait patterns for different velocities: When the decentralized control structure is run with different velocities, different gait patterns emerge adaptively from the interaction fo the local control centers. Shown are tripod patterns, tetrapod patterns, and a wave gait.

https://www.dropbox.com/s/dlb7hmmt7dmjihl/2_EmergentGaits_Simulation.mp4?dl=0

Movie S3 (separate file). Disturbance of the physical robot walking by inducing long steps: As further experiments, we systematically induced searching movements and long steps on the real robot Hector. This forced movement disturbed coordination and leads to instable walking situations. Shown is one run, where the cognitive expansion had to take over in order to solve an unstable situation. Again, the cognitive expansion found a behavior applied out of context which allowed to resolve the situation and later to continue stable walking.

https://www.dropbox.com/s/prq9xe8fg24v60y/3_PlanningFlat_Hector.mp4?dl=0