

Optimal Leg Sequencing for a Hexapod Subject to External Forces and Slopes*

Georgios Rekleitis, *Member, IEEE*, Menelaos Vidakis, and Evangelos Papadopoulos, *Fellow, IEEE*

Abstract— An optimal leg sequence selection method is developed, which maximizes hexapod robot stability, considering feasible gaits, motion modes, and terrain slope. A novel and fast search method is employed to find the most stable leg sequence for a given gait; if no such sequence exists, the next fastest stable gait is chosen and the most stable leg sequence for this gait is selected. The method can be based on any stability measure; here the Force-Angle Stability Margin criterion is employed that is sensitive to top-heaviness, and inertial and external forces. Results show that the developed method senses instabilities accurately and selects the best leg sequence for maximum stability far faster than exhaustive searches, offering distinct advantages when varied external forces are applied.

I. INTRODUCTION

Legged robots offer advantages over wheeled ones such as discontinuous terrain adaptability and off-road mobility, in the expense of speed and power efficiency. Hexapods are more stable than wheeled robots [1]-[3], and robots with fewer legs. Hexapod studies have dealt with terrain adaptability and locomotion stability [4]-[6], aiming at gaits and/or leg sequences that compensate for lack of foothold positions [8], [6], or for leg failures [5], [9] - [11].

A random leg sequence selection for a hexapod robot motion results in motion stability characteristics ranging from optimally stable motion to instability and failure. Despite the importance of leg sequence selection, little has been done in sequence optimization, especially in a generic way that includes a wide range of gaits, and motion modes, in the presence of sloped terrain and external forces. Usually a gait and/or leg sequence selection is performed among a small, pre-defined group of choices, [12], [13], while no stability optimization may be performed, since other criteria may be used, such as energy consumption minimization, [7]. When stability and leg sequence is studied, only the static stability is considered; the effects of dynamic external forces on robot stability are usually neglected [14], [15]. The latter is very important to underwater legged robots subject to current disturbances or walking on variable slope terrains. External disturbances are considered in deriving stable leg sequences, but no stability optimization is undertaken [16], [23].

The authors have introduced a generic method, which employs exhaustive search to determine a hexapod stable leg sequence for the fastest stable gait, given external conditions, such as terrain slope, and disturbances [18]. The method can be based on any stability criterion; in [18] the Force-Angle

Stability Margin (FASM) was used, considering the effect of top-heaviness and of external forces [17]. The underwater hexapod HexaTerra [20], see Figure 1, simulated on slopes with severe external forces, showed that the method detects instabilities accurately, yielding stable leg sequences [18].

In this paper, in contrast to the time-consuming exhaustive search in [18], a novel search method is presented, yielding an optimal, or in few known cases, suboptimal solution, significantly faster. The method can be used off-line to study hexapod motion stability, e.g. for design purposes, or using toe force sensors, on-line to monitor robot stability and set the best gait and leg-sequence. Extensive simulations for the underwater robot HexaTerra and simulations and experimental tests for the Phantom X Hexapod Mark II robot [19] illustrated the usefulness of the developed Leg-Sequence Selection Algorithm (LSSA), and its enhanced applicability due to the intelligent leg-sequence searching method.

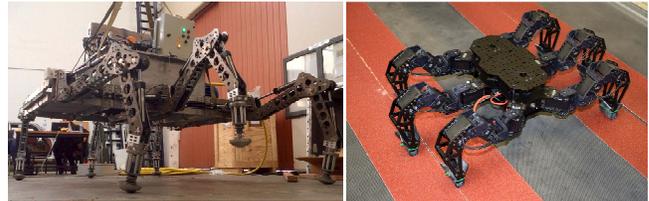


Figure 1. The hexapods HexaTerra (left), [20], and Phantom X (right) [19].

II. HEXAPOD ROBOT REPRESENTATION

In this work, hexapod robots with three degrees of freedom per leg are studied, such as the underwater HexaTerra or the smaller Phantom X, Figure 1. The motion modes for such robots include *curved motion* (rotation and translation) and *pure translation*, with pure rotation just a special case of curved motion with zero translation, see Figure 2.

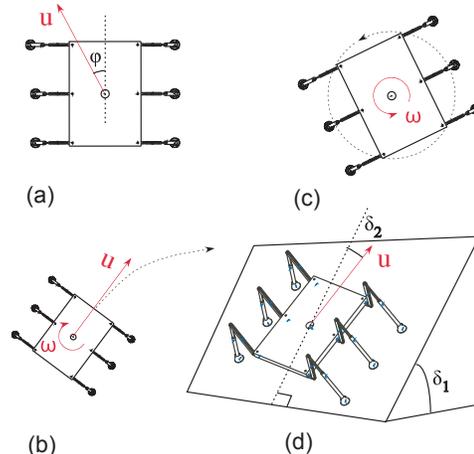


Figure 2. Hexapod motion modes: (a) crab motion, (b) curved motion, and special cases: (c) pure rotation, (d) straight motion (slopped terrain).

* Support from the European Commission through contract FP7-SME-2013 / 605420 (HexaTerra), is acknowledged.

The authors are with the School of Mechanical Engineering, National Technical University of Athens, Greece (e-mail: [georek, egpapado]@central.ntua.gr; menelaosvidakis@gmail.com, phone: +30-210-772-1440).

Figure 3 shows a simplified 2D top view of hexapod toe workspaces [21]. The leg toes are labelled 1, 2, 3 on the left-hand side and 4, 5, 6 on the right-hand side. The toes are within their workspace, defined by rectangles of dimensions P and Q , see Figure 3. Point C denotes the projection of the robot center of mass (CM) on the locomotion plane.

For simplicity and without loss of generality, the following assumptions are made: (a) the hexapod is symmetric, (b) there is point toe-ground contact, (c) feet do not slip, (d) leg masses are lumped into the hexapod body, and the CM is at its centroid, (e) hexapod body speed and leg average speed during the transfer phase are constant, (f) the initial foothold positions are known, (g) toe force sensors are available, (h) each leg has a safe workspace, accessible to itself only, and (i) the hexapod robot moves statically stable.

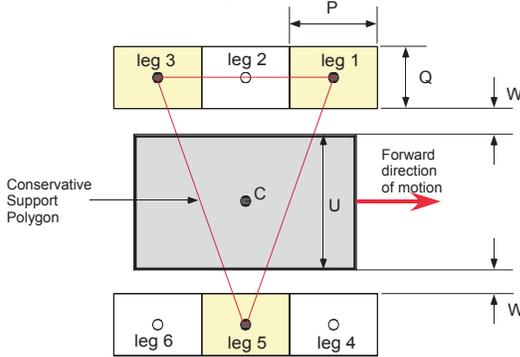


Figure 3. Kinematic parameters of a hexapod robot, including the CSP.

III. LEG-SEQUENCE SELECTION ALGORITHM

(a) Stability Criteria for Walking Robots. To monitor robot stability and avoid tip-over, a criterion must be employed. Examples include the static stability criterion Stability Margin (SM), and the Force-Angle Stability Margin (FASM) criterion, which considers the effects of external forces and top heaviness, and is therefore superior to SM when such forces are applied [17], [18]. Most criteria also use some variation of the Conservative Support Polygon (CSP) [22], see Figure 3, in which black circles denote supporting leg foothold positions and white circles the currently lifted leg previous positions. Using FASM with a hexapod, all external forces applied to the body and legs are equal to the sum of ground reaction forces, and can be measured using toe force sensors. In analytical studies, if ground reactions are not provided by the simulated environment, they can be obtained using models and assumption (d) is not required.

(b) Hexapod Gaits. The six legs of a hexapod allow for three possible gaits: (a) tripod gait (3 feet in the air, 3 supporting), (b) tetrapod gait (2 feet in the air, 4 supporting), and (c) pentapod gait (1 foot in the air, 5 supporting). The tripod gait is the fastest but the least stable, (min CSP), the tetrapod is the second fastest and second most stable, and the pentapod is the slowest but most stable gait. For each motion mode and gait combination, several leg sequences exist, resulting in different CSPs with different stability results, from the optimally stable case to even unstable cases, depending also on the external conditions. Leg sequences are shown in brackets, with the leg groups to be lifted separated by commas; for example, for a quadruped gait, {1–4, 3–5, 2–

6} means that first legs 1, 4 are lifted and moved, then 3, 5 etc. To maintain robot static stability during locomotion, a decision must be made regarding the optimal sequence for lifting and positioning of the robot legs, and for the fastest stable gait available that maintains a desired stability margin; this is the task of the *Leg-Sequence Selection Algorithm (LSSA)* to be developed. Regarding the motion modes, the crab mode is defined by the leg stride displacement $d > 0$ at angle φ , Figure 4a, while the curve mode, by $d > 0$ along a curve, with orientation change by φ , Figure 4b.

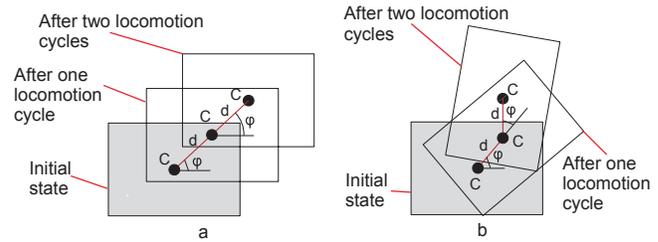


Figure 4. Motion mode parameters for (a) crab motion, (b) curve motion.

(c) Stability enhancement. To maximize leg displacement, and hence speed, d is not preset, but is bounded by toe workspace limits. Therefore, d is a function of φ and toe workspace. In crab mode, if the desired d and φ result in toe workspace violation, d is adjusted to keep the toe in its workspace. In curve mode, toe workspace violation leads in d , and φ trimming, reducing the motion speed, but maintaining the motion direction.

Lowering the robot CM height h reduces the robot top heaviness, while the more apart the legs are, the larger the CSP is; both increase the stability margin independent of the stability criterion employed. Height h is bounded by h_{min} , depending on the terrain or the task. Note that if $\varphi \neq 0$, extending leg positions can lead to partial, or even total loss of motion capability. As shown in Figure 5, if the toe starts from its workspace center, the available displacement $d = s_1$, while if the legs have moved outwards for increased stability, d is reduced to s_2 . Thus, leg extension is used when increased stability is required without affecting the desired motion, e.g. if $\varphi = 0$, otherwise a compromise regarding leg extension must be made. Therefore, to enhance robot stability without a gait change, first the CM is lowered to h_{min} . If instability is still an issue, initial leg toe positions are extended, possibly resulting in reduced speed, but less than that due to a gait change (in pentapod gait this is the only option).

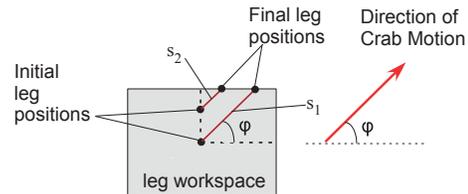


Figure 5. Reduction of crab motion capabilities when the initial leg positions are extended sideways for increased stability.

(d) Algorithm development. The LSSA inputs include robot parameters, external conditions, such as the terrain, or disturbances, and the desired motion mode and gait. Its output is the optimal leg sequence for maximum stability, which is determined by employing some stability criterion, and perhaps a change in the desired gait. Note that although

simple criteria like the SM can be used, with LSSA providing a corresponding leg sequence solution, stability *will not be guaranteed* if top heaviness and external disturbances are non-negligible, since the criterion is not appropriate. In such a case, the LSSA would require a criterion like the FASM.

(i) *LSSA main loop.* For a specific motion mode, the algorithm starts with the desired gait and checks for possible leg sequences, by calling the corresponding routine, e.g. *3podRoutine* for a tripod desired gait. If no stable leg sequence is returned, then the algorithm choses the next most stable gait, i.e. from tripod to tetrapod and/or from tetrapod to pentapod, see Figure 6. The LSSA exits either by returning the most stable leg sequence for the desired or fastest stable gait, or no sequence, if no stable sequence exists for any gait.

```

if mode == crab
  if gait == tripod
    LSeq = 3podRoutine
    if LSeq == []
      LSeq = 4podRoutine
      if LSeq == []
        LSeq = 5podRoutine
      end
    end
  elseif gait == tetrapod
    LSeq = 4podRoutine
    if LSeq == []
      LSeq = 5podRoutine
    end
  else % gait = pentapod
    LSeq = 5podRoutine
  end
else % mode = curve
  LSeq = LegSeqRoutine
end

```

Figure 6. The main loop of the LSSA.

(ii) *The 3podRoutine, 4podRoutine and 5podRoutine.* For the tripod gait, the *3podRoutine* calls the 3pod function, see Figure 7, to obtain the most stable tripod gait leg sequence. If the 3pod function returns no feasible solution, the robot CM is lowered once, and the process repeated. If no sequence is obtained by lowering the CM, the routine extends the leg initial positions once, and calls the 3pod function for a last time. If no sequence is obtained, the *3podRoutine* ends by returning an empty result; in such a case, the next more stable gait is tried by the main loop, i.e. the *4podRoutine*. Else, the leg sequence obtained is used to command the legs. The *4*, and *5podRoutine* have similar structure to the *3podRoutine*.

```

LSeq = 3pod
if LSeq == []
  Lower_CM
  LSeq = 3pod
  if LSeq == []
    Expand_Legs_Pos
    LSeq = 3pod
    Undo(Expand_Legs_Pos)
  end
  Undo(Lower_CM)
end

```

Figure 7. The 3podRoutine.

(iii) *The 3pod, 4pod and 5pod functions.* The 3pod, 4pod and 5pod functions are called by the corresponding routine to test the available leg sequences and identify the optimal one for the chosen gait, using some stability criterion. For each of the *i* CSPs resulting from the groups of legs lifted during the

*j*th leg sequence, the corresponding stability values (e.g. β_{ij} for FASM, [17]) are computed. The overall minimum stability value characterizes the sequence (e.g. $\beta_j = \min(\beta_{ij})$). Negative value means unstable leg sequence and is rejected. Selecting the maximum stability value among all stable leg sequences (e.g. $\max(\beta_j)$; max-min criterion), yields the optimal leg sequence in stability terms, for the given gait/ motion mode.

(iv) *Smart search.* An important question is whether one can avoid testing all leg sequences. Depending on external conditions, a randomly chosen leg sequence may result in reduced stability or even in instability. An exhaustive search yields the optimal motion parameters but is rather slow [18]. The search can be accelerated significantly using simple observations, enabling the use of the LSSA in real time.

Independent of the stability criterion employed, to optimize static stability for any leg sequence it is best to maximize each CSP and the intersection of consecutive CSPs. The latter is achieved by having CSPs maximizing the area around C. Thus, for the tripod gait (*3pod function*), simultaneously lifting legs 1-3-5 or 2-4-6, see Figure 3, results in consecutive CSPs with far larger intersection areas than any feasible alternative, yielding far more stable motions. The 3pod function, thus, checks the stability of only two leg sequences: {1-3-5, 2-4-6} and {2-4-6, 1-3-5}. For the tetrapod gait (legs lifted in pairs), it is undesirable to either lift two consecutive legs, since this diminishes the CSP area leading to smaller CSP intersections (see also Figure 8), or lift two non-consecutive legs from the same side (e.g. 1-3), since the resulting CSP is equal to the tripod gait, wasting the tetrapod stability advantage. Thus, only seven possible leg pairs to be lifted remain, shown in blue lines in Figure 9a. Since each leg is lifted only once in each cycle, the combinations of those leg pairs are further reduced to a total of 24 possible tetrapod gait leg sequences. The pentapod gait has no such limitations, leaving 720 possible leg sequences. To avoid an exhaustive search for all possible sequences each time the 4pod or 5pod functions are called by the 4pod or 5pod Routine (similar to 3pod function calls by 3podRoutine in Figure 7), a smart search strategy is developed.

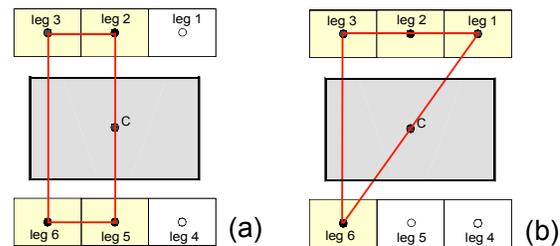


Figure 8. Undesired tetrapod gait lifting (a) 1-4 and (b) 4-5, leading to small CSPs, with low coverage of the area around C.

The first two legs to be moved initially are found first, i.e. the first pair in tetrapod, or the first and second legs in pentapod gaits. The remaining combinations are 4 for the tetrapod and 24 for the pentapod gait, leading to 83.3% and 96.7% smaller search respectively, each time the corresponding function is called. If this search yields no stable leg sequence, then the 4pod and 5pod functions try the second most desirable initial leg pair and so on, until either a stable leg sequence is obtained, or all possible leg sequences have been checked. Thus, the algorithm is robust in that it *always*

yields a stable leg sequence *if at least one exists*. Due to the way the critical first two legs are chosen, see (v), this solution is the optimal in almost all cases (see also Section IV), while in a few known cases, not discussed here for brevity, it yields at most the third best solution (sub-optimal result).

(v) *The initial two legs.* Several possible CSPs exist, see Figure 9a, depending on the lifted legs, e.g. lifting leg 4 leaves the line between legs 1 and 5 (line 1-5) as a CSP edge. Lines 1-6, 3-4 and 2-5 cannot be CSP edges, since this requires simultaneously lifting two consecutive legs. Thus, possible CSP edges for tetrapod and pentapod gaits are those shown as red and blue lines in Figure 9b. Lines 1-5, 2-4, 2-6 and 3-5 (red lines in Figure 9b) define the *critical potential edges* (CPE). If projection point C in Figure 9 moves towards some CSP edge, stability is reduced, while the opposite happens if it moves away from it. So, if the robot and thus C move towards some CPE, the leg which if lifted will make this CPE a CSP edge, must be lifted and moved immediately, before this CPE becomes more critical. If C moves away from a CPE, lifting the legs that will make it a CSP edge can be delayed. For example, in Figure 9b if C moves towards CPE 2-4 and away from CPE 3-5, then leg 1 that, when lifted, makes CPE 2-4 a CSP edge, must be lifted before CPE 2-4 becomes more critical, while leg 6 that exposes CPE 3-5 when lifted, can be delayed letting 3-5 become less critical.

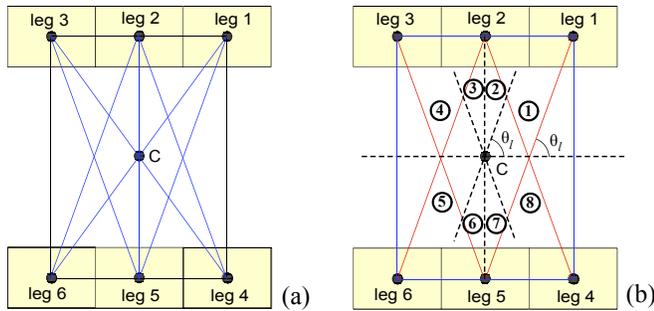


Figure 9. (a) possible leg pairs in tetrapod gait (blue lines) and resulting CSPs (blue and black lines) and (b) most critical edges of the CSPs (red lines) in tetrapod and pentapod gaits and point C motion regions.

As a search strategy first step, the direction of the robot motion and of point C is checked against eight regions and bounding CPEs, see Figure 9b. Weights $w_4 < w_3 < 0 < w_2 < w_1$ are attributed to the legs defining each CPE. For example, robot motion towards region 1 in Figure 9b suggests that C is moving towards CPE 2-4, and the highest weight w_1 is attributed to legs 2 and 4, less directly towards CPE 1-5, and w_2 is attributed to legs 1 and 5, away from CPE 3-5, and the lowest, negative weight w_4 is attributed to legs 3 and 5, and less directly away from CPE 2-6, with w_3 attributed to legs 2 and 6. Legs common in two CPEs are attributed the algebraic sum of the corresponding weighting points (e.g. leg 2 is attributed $w_1 + w_3$). Point C in Figure 9 is obtained with the robot weight as the only external force on even terrain. In general, point C can be anywhere inside the blue rectangle of Figure 9b, and then the motion direction regions are twelve instead of eight (not shown here for brevity and simplicity).

Note that the cases in which the LSSA yields at most the third-best sequence depend on the direction of the robot motion, the position of C in the CPE-defined regions, and on what stage of the leg sequence C changes region, which

cannot be determined a priori in some cases, resulting in three possible solutions. The optimal sequence can be obtained in these cases too, by checking all three best initial leg motion cases. For pentapod gait, this leads to 72 sequence searches instead of 24. Although 72 is still far less than the exhaustive search of 720 sequences, all three sequences are very close; therefore not much is gained with the 72 searches.

The search strategy outlined above requires computing the location of point C. When taking into account external disturbances (e.g. when using the FASM), the projection yielding point C is performed along the line of the resultant external force acting on the robot CM. The external torques effect is also considered, by including an equivalent force, using an approach similar to the one used by the FASM [17].

Figure 10 shows the LSSA flowchart (FASM), with the smart leg sequence search omitted for presentation simplicity.

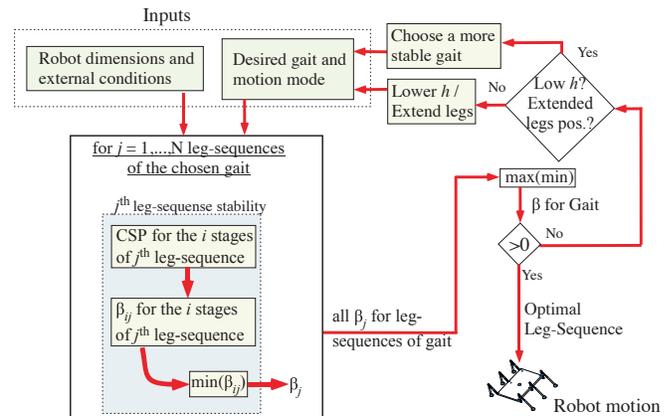


Figure 10. The LSSA flowchart.

The LSSA can be used either prior to a simulated robot motion with preset external conditions, or during robot motion with measured forces, tracking the optimal leg sequence for a commanded motion mode and gait (see also Section IV). In both cases, if no stable leg sequence exists for the desired gait, the LSSA yields the optimal leg sequence for the fastest possible stable gait. Moreover, the LSSA can be called at any time to re-calculate the optimal leg sequence after the motion of each leg, or pair/ triplet of legs, interrupting the leg sequence if necessary, and quickly adjusting the remaining legs motion accordingly, searching only the legs that have not yet been moved. The importance of the smart search and its very low computational time (see Section IV), becomes even more apparent in such cases.

IV. RESULTS

(a) **Simulations.** We study the LSSA validity and the novel smart search efficiency, initially by simulations. Two hexapod robots are simulated; the underwater hexapod HexaTerra (Figure 1 left, [20]) and the smaller Phantom X robot (Figure 1 right, [19]), with properties shown in Table I.

Table I. Dimensions and total mass of the two hexapods (see Figure 2).

P (m)	Q (m)	W (m)	U (m)	h (m)	h_{min} (m)	M (kg)
HexaTerra						
1	1	0.433	1.2	0.9	0.7	660
Phantom X						
0.12	0.024	0.144	0.12	0.12	0.09	2.055

The LSSA employs the smart search method and the FASM stability criterion, while, for safety reasons, it is assumed that accepted stable leg sequences are the ones with FASM value $\beta \geq \beta_m$ where β_m provides a safety margin, defined as 10% of the β obtained for the robot on even terrain and with all six legs on the ground, with the normal leg pose. As the β is not normalized to size, β_m is equal to 54 for the Hexaterra and 0.17 for the Phantom X. In simulations, external forces are estimated using models. To compare simulation results to experiments, for the Phantom X, it is assumed that only its weight acts as an external force, (see Section IV (b)).

Without loss of generality, for the underwater HexaTerra simulations buoyancy force \mathbf{A} is assumed to be 25% of the total robot weight W and both are applied to the robot CM. Also, HexaTerra is equipped with a trenching tool of constant maximum trenching force \mathbf{f}_t (see Table II, and Figure 11b), and moment $\|\mathbf{n}_t\| = \|\mathbf{f}_t\| * h$. A water drag force \mathbf{R}_w acts on the HexaTerra at $\theta = 180^\circ$ (i.e. water current from back to front, see Figure 11a), and is modeled as

$$\mathbf{R}_w = 0.5 \cdot C_D \rho A_r \mathbf{u}_{r_max} \|\mathbf{u}_{r_max}\| \quad (1)$$

where C_D is the drag coefficient, ρ is the seawater density, \mathbf{u}_{r_max} is the maximum relative speed between robot and seawater, and A_r is the robot area normal to \mathbf{u}_{r_max} . Note that

$$\mathbf{u}_{r_max} = \mathbf{u}_{wv_max} - \mathbf{u}_{max} \quad (2)$$

where \mathbf{u}_{max} is the maximum robot speed, and \mathbf{u}_{wv_max} is the maximum sea current speed. External force parameters are shown in Table II, and Figure 11. The smart search weights were chosen empirically as $w_1=2$, $w_2=1$, $w_3=-0.5$ and $w_4=-2$. Different weight sets may result in a solution loss, yielding a less optimal one, but not in a total solution loss, if one exists.

Table II. HexaTerra external forces calculation parameters during motion.

\mathbf{f}_t	C_D	ρ	\mathbf{u}_{max}	\mathbf{u}_{wv_max}	\mathbf{R}_w
800 N	0.8	1025 kg/m ³	0.05 m/s	8/5 kn	68.91 N

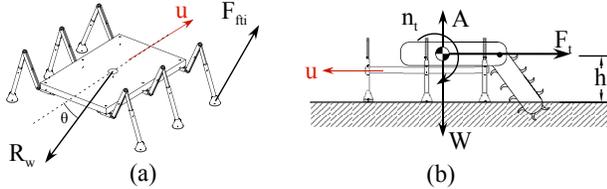


Figure 11. External forces acting on the HexaTerra robot.

Simulations were run for both robots, climbing up a slope that is getting steeper, gradually for the HexaTerra, Figure 12a, and continuously for Phantom X, Figure 13a.

The terrain provides all required friction, i.e. there is no slip. In both cases, external conditions are $\varphi = \delta_2 = 0^\circ$, i.e. a straight up motion with step trimmed to 0.5m for HexaTerra and 0.04m for Phantom X. Initial inclination was set at $\delta_1 = 5^\circ$ for HexaTerra and $\delta_1 = 0^\circ$ for Phantom X. In both cases, the robots start from nominal CM height and leg extension, adapting to the slope increasing inclination accordingly, see Figure 12b, c and d, and Figure 13b, c and d, respectively.

The requested gait for both cases is the tripod one with nominal CM position and leg extension. The response of the LSSA to this request and the resulting FASM values β are shown in Figure 12e and Figure 13e, for the HexaTerra and

Phantom X simulations, respectively. As expected, as the slope becomes steeper, the LSSA resorts to increasingly more stable, but also slower gaits, in both cases. For the Phantom X case, at slope angle $\delta_1 = 25^\circ$ the LSSA provides the stable leg sequence $\{1, 4, 3, 2, 6, 5\}$ in the pentapod gait, with $\beta = 0.466$. If the inclination keeps on rising, then the β keeps on dropping, until the inclination becomes $\delta_1 = 35^\circ$, in which case no stable gait / leg sequence can be found.

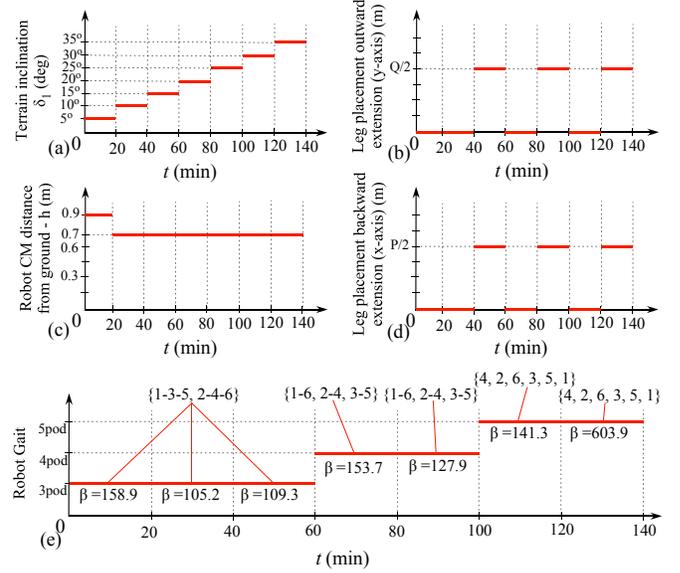


Figure 12. HexaTerra motion on a slope with increasing inclination. (a) Terrain inclination. Leg placement extensions: (b) outward, along y-axis and (d) backward, along the x-axis. (c) Robot CM distance from the ground. (e) Chosen gaits and leg sequences, with corresponding FASM values.

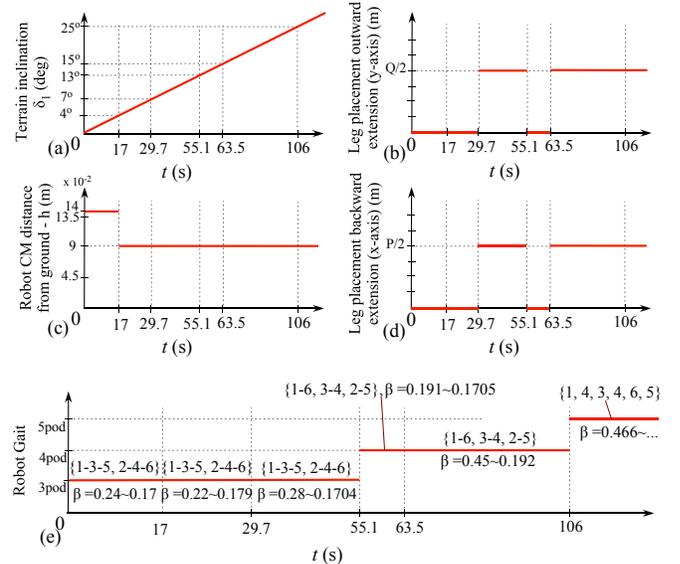


Figure 13. Phantom X motion on a slope with increasing inclination. (a) Terrain inclination. Leg placement extensions: (b) outward, along y-axis and (d) backward, along the x-axis. (c) Robot CM distance from the ground. (e) Chosen gaits and leg sequences, with corresponding FASM values.

For both robots, the effect of the extension of the initial leg placement is quite significant on stability. As a result, the configuration with lowered CM and extended initial leg positions, and with tripod gait, is more stable than the one with the initial robot pose and tetrapod gait. Thus, the latter does not appear during the slope climbing motion. The same

is observed between tetrapod and pentapod gaits. Nevertheless, even with enhanced stability initial pose (leg placement and lowered CM), there is a limit to which slope inclination each robot can climb with tripod gait. Turning to tetrapod (and then pentapod) gait pushes this limit further on. Moreover, there is also a limit to how much the initial leg position can be extended and the robot CM can be lowered (and thus how much these techniques can affect walking stability), depending not only on the specific robot geometric characteristics, but also on the environment (e.g. obstacles over which the walking robot has to pass), while, the change from tripod to tetrapod and then pentapod gait, is always a stability enhancement option. However, even by changing to a more stable gait, unstable leg sequences were observed in all cases, (e.g. the unstable pentapod sequence {3,6,2,5,1,4} at 34° inclination in the Phantom X simulations). These observations demonstrate the importance of searching for the most stable leg sequence (as opposed to randomly choosing a leg sequence) for a given gait and, failing that, for the fastest, stable gait/leg sequence combination.

In terms of the computational time, the LSSA yielded the above mentioned three tripod gait solutions in just about 11ms, for the HexaTerra simulations and in about 5ms, for the Phantom X simulations. For the tetrapod and pentapod gaits, LSSA computation times are shown in Table III, with the tripod initial gait and the initial robot pose. As shown in Table III, both the LSSA and the exhaustive search resulted with the same leg sequences and same stability margin, a fact also observed for all cases shown in Figure 12 and Figure 13. The table shows that the smart search described in Section III vastly improves the LSSA computational time by 43% to 72%, without missing stable leg sequences, allowing LSSA to be used on-the-fly gait/leg sequence selection. These times were obtained with LSSA running on a on an i7 PC, with compiled C-code called by MATLAB.

Table III. Computational time of the LSSA for both simulation test cases, using the smart and the exhaustive search.

		HexaTerra		Phantom X		
Ground inclination		25°	35°	13°	20°	30°
Resulting gait		4pod	5pod	4pod	4pod	5pod
Initially required gait		3pod	3pod	3pod	3pod	3pod
LSSA-FASM (smart search)	Comp. time	16 (ms)	53 (ms)	8 (ms)	13 (ms)	16 (ms)
	Stability (β)	127.9	603.9	0.1913	0.2947	0.2951
LSSA-FASM (exhaust search)	Comp. time	35 (ms)	170 (ms)	14 (ms)	26 (ms)	58 (ms)
	Stability (β)	127.9	603.9	0.1913	0.2947	0.2951
Computational efficiency improvement		54.3%	68.8%	42.9%	50%	72.4%

(b) Experiments. To demonstrate the validity of the LSSA for a real robot walking on dynamically changing environment, the Phantom X hexapod was used in several experiments, climbing a ramp with increasing inclination. The first test run is identical to the aforementioned Phantom X hexapod simulation. Since the robot is not yet equipped with toe force sensors, the external forces effect (i.e. only the effect of gravity in this case) is calculated based on the known robot properties (see Table I) and the measured ramp inclination. This is fed as input to the LSSA running on an external i7 PC, which then provides the optimal leg sequence, and transmits it to the climbing robot using an XBee. The

compiled C code also includes, except the LSSA itself, the conversion of the LSSA output to the corresponding leg motor angle commands, so as not to burden the hexapod onboard Arduino with this conversion.

Due to height limitations, the ramp inclination does not reach the simulated 25°, but only 13.4°, starting from an initial inclination of 1.4°. Nevertheless, this is enough to observe the change from tripod with initial robot pose, to tripod with lowered CM, then to tripod with lowered CM and extended leg placement, and finally to tetrapod with non-extended leg placement but with lowered CM, just as was the case in the simulations (see also produced video, and experiment snapshot in Figure 1, right). The result is that the LSSA successfully provides the robot with the fastest stable gait/leg sequence combination, even though the initially required gait is tripod at all times. At the end of the motion, at inclination $\delta_1 = 13.4^\circ$, the robot is commanded on purpose to move with tripod gait and the initial robot pose, violating the LSSA output. As expected, PhantomX tips over (see attached video), demonstrating the leg sequence selection importance.

Additional experiments included climbing up the ramp with non-zero δ_2 angle, see also Figure 2d, and going down the ramp, all of which discussed here briefly for brevity. Note that, violating the LSSA output (e.g. keeping on requesting tripod gait/leg sequence when the LSSA predicts instability) may not automatically result in actual instability, depending on how much the LSSA output is violated with the requested motion, due to the introduced safety margin (0.17 for Phantom X). Nevertheless, when the ramp keeps on lifting, the robot tips over in every case, soon after the LSSA first prediction. This demonstrates the use of the safety margin, in recognizing imminent instability on time.

V. CONCLUSION

An optimal leg sequence selection method for hexapod robots in terms of robot stability and for a combination of various gaits, motion modes and sloped terrains, called the LSSA, was developed and analytically presented. The LSSA can be based on any measure of stability; in this work the FASM was employed that is sensitive to dynamically changing environment and disturbances. The LSSA finds a stable leg sequence for the required motion mode and gait; if no such stable case exists, the method reconfigures the robot pose in favor of stability and tries again; if even that fails to provide a stable leg sequence, the gait is changed to the fastest stable one and a stable leg sequence for this gait is obtained. The LSSA searches for stable leg sequences in a robust, intelligent way, resulting in faster and optimal results, except in a few known cases, and never fails to provide a stable leg sequence, if one exists. Simulations of the underwater hexapod HexaTerra moving on slopes with severe external forces, as well as both simulations and experiments with the actual Phantom X hexapod, show that the developed LSSA with the FASM criterion and the introduced smart search, quickly and accurately predicts instabilities. The LSSA has significant practical value, since it can be used off-line, to study robot motion stability or for design purposes, and in real time in selecting gaits and leg sequences, maintaining and monitoring stability.

REFERENCES

- [1] Song, S. M and K. J. Waldron, *Machines That Walk: The Adaptive Suspension Vehicle*, MIT Press, Cambridge, MA, 1989.
- [2] Todd, D. J., *Walking Machines, An Introduction to Legged Robots*, Koran Page, London, 1985.
- [3] Raibert, M. H. *Legged Robots That Balance*, MIT Press, 1986.
- [4] McGhee, R. B., "Vehicular legged locomotion," in *Advances in Automation and Robotics*, Ed. Greenwich, CT: JAI, 1985.
- [5] Yang, J. M. and J. H. Kim, "A Fault Tolerant Gait for a Hexapod Robot over Uneven Terrain," *IEEE Trans. Syst., Man, and Cybernetics—Part B: Cybernetics*, vol. 30, no. 1, February 2000.
- [6] Loc, Vo-Gia, et al., "Improving traversability of quadruped walking robots using body movement in 3D rough terrains," *Robotics and Autonomous Systems* 59, pp. 1036-1048, 2011.
- [7] Luneckas, M., T. Luneckas, D. Udriš, N. M. F. Ferreira, "Hexapod Robot Energy Consumption Dependence on Body Elevation and Step Height," *Elektronika Ir Elektrotechnika*, ISSN 1392-1215, 20,7, 2014.
- [8] Manjanna S. and Dudek, G., "Autonomous Gait Selection for Energy Efficient Walking" *2015 IEEE International Conference on Robotics and Automation (ICRA)* Seattle, Washington, May 26-30, 2015.
- [9] Yang, J. M., "Fault-tolerant gait generation for locked joint failures," *IEEE Int. Conf. Syst., Man, Cyb.*, v.3, Oct. 5-8, 2003, pp. 2237–2242.
- [10] Wang, Z.-Y., X.-L. Ding and A. Rovetta, "Analysis of typical locomotion of a symmetric hexapod robot," *Robotica*, Vol. 28, No 06, pp. 893 – 907, October 2010.
- [11] Manglik, A., Gupta, K., Bhanot, S., "Adaptive Gait Generation for Hexapod Robot using Genetic Algorithm," *IEEE International Conference on Power Electronics, Intelligent Control and Energy Systems (ICPEICES)*, Delhi, India, 4 – 6 July, 2016.
- [12] Zhang, C. D. and S. M. Song, "Turning gait of a quadrupedal walking machine," in *Proc. IEEE International Conference on Robotics & Automation*, vol. 3, no. 4, pp. 2106–2112, 6-8 Apr. 1991.
- [13] Bien, Z., M. G. Chun, H. S. Son, "An optimal turning gait for a quadruped walking robot," in *Proc. IROS '91 IEEE/RJSJ International Workshop on Intelligent Robots and Systems '91. Intelligence for Mechanical Systems*, vol. 3, no. 4, pp. 1511–1514, 3-5 Nov. 1991.
- [14] Seljanko, F., "Hexapod walking robot gait generation using genetic-gravitational hybrid algorithm," *The 15th International Conference on Advanced Robotics*, Tallinn, Estonia, 20-23 June, 2011.
- [15] Raheem, F. A., Khaleel, H. Z., Raoof, A., Noree, M., "Hexapod robot static stability analysis using genetic algorithm simulation and experimental work," *Proc. 15th Scientific Conf.*, 23 - 24 April, 2016.
- [16] Moosavian, S. A. A., Dabiri, A., "Dynamics and planning for stable motion of a hexapod robot," *IEEE/ASME International Conference on Advanced Intelligent Mechatronics*, Montreal, Canada, July 6-9, 2010.
- [17] Papadopoulos, E. and D. Rey, "The Force - Angle Measure of Tipover Stability Margin for Mobile Manipulators," *Vehicle System Dynamics*, Vol. 33, No. 1, 2000.
- [18] Aggelopoulou, E., Rekleitis, G., and Papadopoulos, E., "Optimal Leg-Sequence Selection for an Underwater Hexapod Robot in the Presence of Slopes and External Forces," *25th Mediterranean Conference on Control and Automation (MED '17)*, Valletta, Malta, July 3-6, 2017.
- [19] <https://www.trossenrobotics.com/hex-mk2>
- [20] http://cordis.europa.eu/project/rcn/109873_en.html
- [21] Yang, J. M. and J. H. Kim, "Fault-Tolerant Locomotion of the Hexapod Robot," *IEEE Trans. Systems, Man, and Cybernetics—Part B: Cybernetics*, vol. 28, No. 1, February 1998.
- [22] Nagy, P. V., S. Desa, and W. L. Whittaker, "Energy based stability measures for reliable locomotion of statically stable walkers: Theory and application," *Int. J. Robotics Res.*, v. 13, n. 3, pp. 272–287, 1994.
- [23] Agheli, M., Nestinger, S. S., "Force-based stability margin for multi-legged robots," *Robotics and Autonomous Systems*, Vol. 83, pp. 138-149, September 2016.