安全な四足歩行ロボットの実用化に向けた STAMP/STPA 分析

大野佳寿馬 [1,*] 山本貴志 [1] 吉中尊信 [1]
Princelove Smith Herbert [1] 谷口勝巳 [2] 金子朋子 [1]

概要：近年，ロボット技術の急速な進展により，様々な分野でロボットの実用化が進んでいる．中でも四足歩行ロボットは，不整地での移動性能を活かし，防犯，防災において危険地など，人には立ち入れない現場で，SAR などの人命に関わる情報提供やレスキュー目的での活用が期待されている．一方で，四足歩行ロボットにおける安全性が十分に明確化されておらず，多様なサイバー攻撃の標的になる可能性があるなどの諸課題があるため，本格的な実用化には至っていない．本稿では，STAMP/STPA 分析にて，明らかとなった四足歩行ロボットに関する様々なリスクの中から，複数のホスト間で行われる通信における脆弱性に対する要因を分析し，安全な実用方法の検討をした．

キーワード：四足歩行ロボット，STAMP/STPA，セキュリティ，通信における脆弱性，SAR，STRIDE

Ensuring the Safe and Practical Use of Quadruped Robots Using STAMP/STPA Analysis

Kazuma OHNO[1,*] Takashi YAMAMOTO[1] Takanobu YOSHINAKA[1]
Princelove Smith Herbert[1] Masami TANIGUCHI[2] Tomoko KANEKO[1]

Abstract: Recent advancements in robotics have accelerated the practical deployment of robots across a wide range of fields. Among them, quadruped robots are drawing significant attention due to their superior mobility over rough and uneven terrain. While quadruped robots are expected to be utilized in life-critical applications such as Search and Rescue (SAR) operations and the provision of vital situational information in hazardous or inaccessible environments, their practical use remains limited. This is largely due to the insufficient clarification of their safety requirements and the potential exposure to various types of cyberattacks, particularly in network-based operational settings. This study employs the STAMP/STPA framework to identify and analyze potential risks associated with the operation of quadruped robots. Focusing specifically on vulnerabilities within inter-host communication systems, we examine contributing factors to these risks and propose practical strategies for ensuring safer deployment in real-world scenarios.

Keywords: Quadruped robots, STAMP/STPA, security, vulnerabilities in communication, SAR, STRIDE

1. はじめに

近年，ロボット技術の急速な進展により，様々な分野でロボットの実用化が進んでいる．中でも四足歩行ロボットは，不整地での移動性能を活かし，製造，物流，医療，農業など幅広い現場での応用が期待されている．一方で，これらのロボットがネットワークを介して制御されるようになるにつれ，安全性やセキュリティに関する新たな課題も顕在化してきている．中間者攻撃や DoS 攻撃，IP スプーフィングを初めとする多くのサイバー攻撃は，ロボットの制御信号を妨害・改ざんする可能性があり，結果として誤作動や機能停止を引き起こすリスクがある．こういったリスクは，防災目的での実用化において大きな障害になる可能性を秘めており，より安全な実用化が求められている．

本研究における目的は，簡易な仕様書のみしか提示されていない状況において，複雑な機能をもつ既製品に対して，どのように安全性を担保させるかを明確化し，対策を講じることはできるのかを検証することである．これは四足歩行ロボットにおける安全性を通信機能の観点から分析し，IoT 機器（具体的には四足歩行ロボット）を対象に，IoT 機器の機能，デバイス，システム，運用に関して，STAMP[1][2] の安全評価手法に基づいた IoT 機器のリスクや対策の明確化を目指すことに繋がる．

本稿では，ロボットの通信におけるセキュリティに起因したリスクを挙げるために，STAMP/STPA 手法によるリスク分析を行い，検討・考察をした．第２章では関連知識として STAMP/STPA と四足歩行ロボットの説明を行い，第３章に，分析内容を提示し，最後に考察を述べる．

1 創価大学
Soka University
2 (株)グローバル・ネット・アドベンチャーズ
Global Net Adventures Inc.

* e25m5340@soka-u.jp

2. 関連知識

2.1 STAMP / STPA

STPA(System-Theoretic Process Analysis)とは，STAMP(STAMP: System Theoretic Accident Model and Processes) に基づく安全性分析手法である．事故発生後だけではなく，システムの概念設計の段階から適用でき，相互作用に潜むハザード要因を識別することが特徴である(表 1,2)．制御構造を用いてシステム全体の振る舞いを確認しながら，安全でないコントロールアクション(UCA: Unsafe Control Action)やハザードを誘発するシナリオを特定することが可能である． STAMP/STPA は宇宙，航空，鉄道などの分野でインフラの安全性分析に使用されることが多い．IT システムやネットワークにも活用されているが，四足歩行ロボットに適用された公開例は希少であり，新たな視点での分析が期待できる．以下に手順を示す[1].

Step0-1 アクシデント・ハザード・安全制約の識別
分析対象のシステムにおいてアクシデント，ハザード，安全制約の識別を行う．用語の定義は以下の通り．
・アクシデント
「望ましくない事象，事故・損失」のこと
・ハザード
「アクシデントが潜在している具体的な状態」のこと
・安全制約
「システムを安全に保つための要件もしくは制約」のこと

Step0-2 コントロールストラクチャー図（CS 図）の構築
次に，コントロールストラクチャー図を構築する．コンポーネントとその役割，また他のコンポーネントにどのようなコントロールアクション(CA)を行っているかを入力し図の整理を行う．

Step1 非安全なコントロールアクション(UCA)の抽出
次に，非安全なコントロールアクション(UCA:Unsafe Control Action)を抽出する．UCA は以下の四つの観点で分類される．
・与えられないとハザード(Not Providing)
・与えられるとハザード(Providing causes hazard)
・早過ぎ，遅過ぎ，誤順序でハザード
(Too early/too late, wrong order causes hazard)
早過ぎる停止，長過ぎる適用でハザード(Stopping too soon / applying too long causes hazard)

図 1 STAMP/STPA 手順

2.2 脅威モデル STRIDE

STAMP / STPA はハザード分析手法であり，本稿ではハザード要因（HCF: Hazard Causal Factor)特定の際に，ヒントワードとして脅威モデルである STRIDE[3]を活用し，考察での分析評価に役立てている．また本分析では STRIDE の他に，新たに以下の安全性のカテゴリを HCP として定義した．H: Human Error, C: Communication Error, P: Physical Error を定義し，STRIDE 同様，ヒントワードとして用いている．

表 1 STRIDE 識別子一覧

Threat	属性	内容
Spoofing identity(なりすまし)	Authentication 認証	コンピュータに対し，他のユーザを装う．
Tampering（改ざん）	Integrity 完全性	データを意図的に操作する．
Repudiation（否認）	Nonrepudiation 否認防止	ユーザがあるアクションを行ったことを否認する．
Information Disclosure（情報の暴露）	Confidentially 機密性	アクセス権限のない相手に情報公開
Denial of Service（サービス不能）	Availability 可用性	攻撃により正規ユーザへのサービス中断
Elevation of Privilege（権限の昇格）	Authorization 認可	悪用可能な不正アクセス権限を得る

2.3 Deep Robotics Lite 3

関連知識として，本研究で対象とする四足歩行ロボット DeepRobotics Lite3[4]について紹介する． Lite3 は，DeepRobotics 社によって開発された研究・教育用途向けの

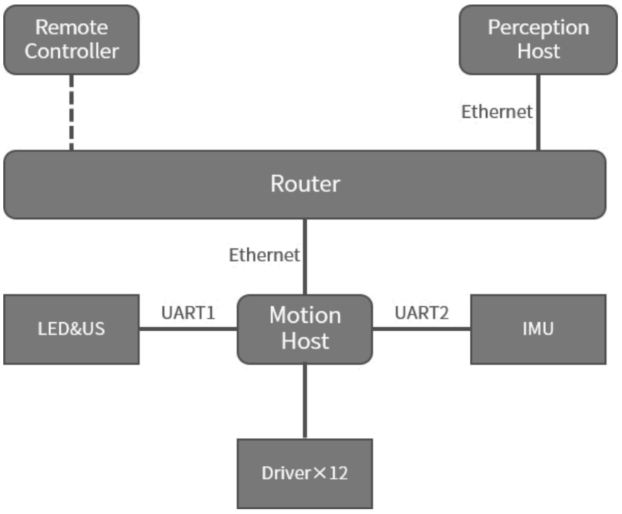

図 2 Deep Robotics Lite3 構成図

表２　アクシデント・ハザード・安全制約の識別（一部抜粋）

アクシデント、ハザード、安全制約の識別					
ID	STPA アクシデント	ハザードID	ハザード	安全制約ID	安全制約
A1	ロボットとの通信に問題があったため、人や障害物に衝突する。	H9	通信帯域が著しく低下しており、リアルタイムの映像や制御コマンドが遅延している。	SC9	ロボットの制御コマンドを正確かつタイムリーに生成し、ロボットへ送信すること。
		H10	通信が頻繁に途絶しており、ロボットの断続的な制御しかできない。	SC10	ロボットの制御コマンドを正確かつタイムリーに生成し、ロボットへ送信すること。
		H11	電波干渉、ノイズ、遮蔽物により、通信信号が極めて弱い。	SC11	電波やノイズ、信号の電波強度を高める。
		H12	有線ケーブルが、ロボットの移動や作業を物理的に妨げる位置にある	SC12	ロボットの可動範囲内には有線ケーブルを置かない。
A2	コントローラーからの制御コマンドを正確に受信できずに、人や障害物に衝突する。	H16	無線通信が暗号化されていない、または脆弱な暗号方式を使用している。	SC16	無線通信を暗号化し、脆弱な暗号方式を使用しない。
		H17	通信プロトコルにセキュリティ上の欠陥が存在する。	SC17	通信プロトコルにセキュリティ上の欠陥が存在しないようにする。
		H18	受信したコマンドが、通信エラーにより誤って解釈されている。	SC18	通信エラーがあった場合でも、コマンドを誤って解釈しない。

軽量な四足歩行ロボットであり，高精度な関節制御，IMU や深度カメラなどの各種センサ，およびモジュール構成された制御系を備えている．本機は，外部との通信を行う Router の他に, Perception Host, Motion Host, IMU, LED&US, Driver を搭載しており，各々Ethernet または UART で通信を行っている．また Motion Host に含まれる Motion SDK が外部と内部情報の取得・制御を仲介する役割を果たしており，外部 PC との連携を通じて高精度な動作制御やデータ収集を可能としている．各コンポーネントの詳細な責務と通信の詳細は，3.2 節で明らかにする．

3. 分析内容

3.1 Step0 アクシデント・ハザード・安全制約の識別

本分析では，主に通信における脆弱性に着目し，アクシデントの発生要因として以下の 2 点を想定する．第一に，コントローラからロボットへの通信障害によって生じる制御ミス，第二に，それに起因する人や障害物との衝突である．それぞれに対するハザードとしては，通信帯域の低下や，電波干渉，プロトコルエラーなどが挙げられた．また安全制約としては，これらハザードを満たせる状態に設定した．

3.2 Step0 各コンポーネントの役割・責務

分析対象の登場人物の抽出を行う．抽出したコンポーネントから，そのコンポーネントが安全にシステム運用していくための役割及び責務を抽出する．Remote Controller では，ユーザーが直接ロボットに指令を出すことができる．実際には，ロボット内に搭載された Router に接続を行う．接続後は，Motion SDK にログインし，認証完了後に，ロボットに対して指令を出すことができる．アクションから前後左右の移動まで様々行うことができる．Motion SDK は，Motion Host を通じて，各モジュールに情報要求を行っている．ここでは，認証等は行わずに，やり取りが行われる．

表３　各コンポーネントの役割・責務

STPA 登場人物	責務	コントロールアクション	フィードバック
Remote Controller (Andorid)	ユーザーがロボットを遠隔操作するためのインターフェース。ユーザーから得た指令をロボット内のRouterに送信する。	IDとPasswordを入力し、Routerに接続 →Router	データを、Motion Hostに送信する。 →Motion Host
Router	超音波センサーは、ロボットの前方と後方にある物体までの距離を計測する。Motion Hostから送信された情報要求に対し、障害物の位置や接近情報をMotion Hostに返送する。	User IDとPasswordを入力し、Routerに接続 →MotionSDK	データを、Motion Hostに送信する。 →Motion SDK
Motion Host	Motion SDKと外部コンポーネントとの通信を繋げる役割を担っている。	情報要求を発出 →各モジュール（CAN 又は UART）	情報受信 ←各モジュール（CAN 又は UART）
Motion SDK	ユーザーからの指示を解析し、各モジュールに情報要求を発出する。要求をMotion Hostへ送信する。	情報要求を発出 →Motion Host（UDP）	情報受信 ←Motion Host（UDP）
各モジュール	IMUや、Battery、MPC Controller、US、CPU、Cameraなどが含まれる。	情報を送信 →Motion Host（CAN 又は UART）	情報を送信 ←Motion Host（CAN 又は UART）

3.3 Step0 コントロールストラクチャ図の構築

対象のシステムにおいて安全を保つために存在するコントロールストラクチャー図（以下 CS 図）を作成する．また CS 図を作成するにあたり，以下の３つの通信路に着目した．Remote Controller から Router，Router から Motion SDK, Motion SDK から各トピックまでの３つの通信である．次節からこの３つの通信路をそれぞれコントロールアクションとし，UCA，HCF を導出する．本分析では，３つの通信に着目している．通信１は，RemoteController から Router までの通信である．ここでは Wi-Fi 経由で自身の
デバイスを, ロボット内に搭載された Router に接続する．

この時，ネットワークセグメントと，それに対応した IP アドレスが決定する．また通信２では SSH で Motion SDK に接続を試みる．ここでは，ユーザー名とパスワードを入力する必要があり，各々デフォルトとして GitHub に公開されているものを使用する．通信３では，Motion SDK から IMU，US，Battery などの各モジュールに情報を送信するように要求を発信し，各モジュールから返信として情報を printed data として受け取る．Motion SDK から Motion Host までは UDP で通信を行っており，Motion Host から各モジュールまでは，CAN または UART で通信を行っている．

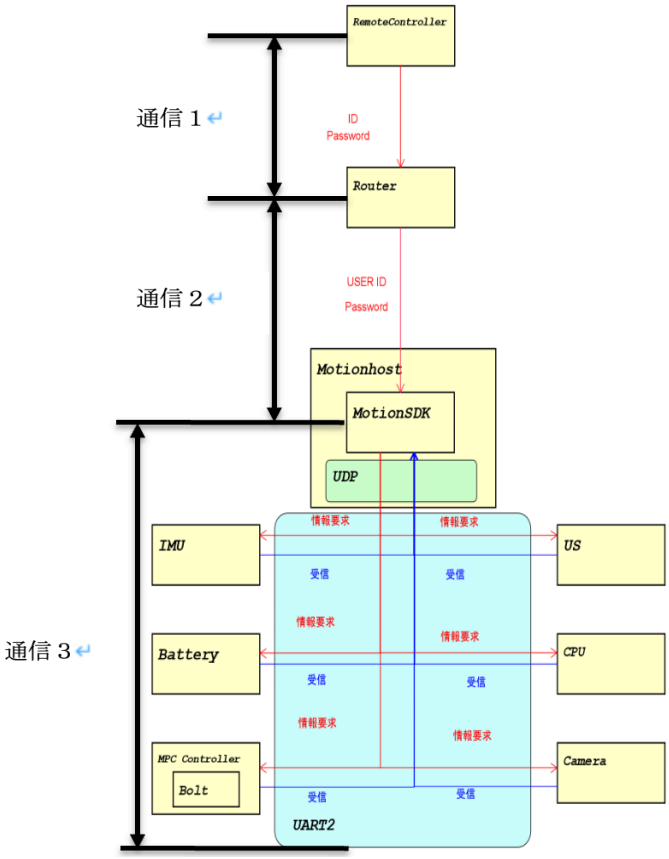

図３　通信機能コントロールストラクチャ
※太い黒線・矢印はコントロールストラクチャに含まない

3.4 Step1 UCA の抽出

3.4.1 Controller と Router での通信における UCA

１つ目の提供されないことでハザードになりうる問題(Not Providing)に関して，ユーザからの指示が与えられないことが挙げられる．２つ目の，原因となり得る危険因子の提供（Providing causes hazard）に関して，Router に対して誤情報の送信や，改ざん，誤った Router への接続など３つを識別できた．

3.4.2 Router と Motion SDK での通信における UCA

１つ目の提供されないことでハザードになりうる問題(Not Providing)に関して，以下の２つを識別できた．①コマンドやデータが Motion SDK に到達しない，②ルーティングを行うための情報提供がされていないことが挙げられる．２つ目の，原因となり得る危険因子の提供（Providing causes hazard）に関して，以下の２つを識別できた．①誤った宛先へのルーティングをして，機密性の高いコマンドやデータが，本来の Motion SDK ではない別の Motion SDK にルーティングされてしまう．②SDK に意図していないコマンドやデータが送信される．３つ目の，早すぎる提供または遅すぎる提供（Too early/Too late）は５つを識別できた．①Motion SDK がコマンドやデータを受信する準備ができていない段階でデータが到達する．②複数のコマンドやデータが特定の順序で処理される必要がある場合，早すぎる到着によって順序が乱れる．③Motion SDK が必要なコマンドやデータを提供の時間内

表４　通信機能コントロールストラクチャーにおける UCA

No		STPA CA	From	To	Not Providing	Providing causes hazard	Too early / Too late	Stop too soon / Applying too long
1	Remote Controller	ユーザーの操作（移動方向、速度、特定のモーションコマンドなど）をRouterまたは直接Motion Hostに送信する。	Remote Controller (Andorid)	Router	(UCA2-N-1)ユーザーが指示した情報が到達しない。	(UCA2-P-1)Routerに対して誤情報を送信する。 (UCA2-P-2)Routerに対して情報を送信している最中に、情報が改ざんされる。 (UCA2-P-3)誤ったRouterに接続してしまう。	(UCA2-T-1)データ送信が遅れてしまう。 (UCA2-T-2)膨大なデータが一気に送信される。	(UCA2-D-1)ControllerからRouterまでの通信時間が長すぎる。 (UCA2-D-2)ControllerからRouterまでの通信が終わる前にセッションが終了する。
2	Router	受信したコマンドやデータをそれぞれの宛先ホストにルーティングする。	Router	MotionSDK	(UCA1-N-1)コマンドやデータがMotionSDKに到達しない。 (UCA1-N-2)ルーティングを行うための情報が提供されていない。	(UCA1-P-1)誤った宛先へのルーティングして、機密性の高いコマンドやデータが、本来のMotionSDKではない別のMotionSDKに到達してしまう。 (UCA1-P-2)SDKに意図していないコマンドやデータが送信される。	(UCA1-T-1)MotionSDKがコマンドやデータを受信する準備ができていない段階でデータが到着する。 (UCA1-T-2)複数のコマンドやデータが特定の順序で処理される必要がある場合、早すぎる到着により順序が乱れる。 (UCA1-T-3)MotionSDKが必要なコマンドやデータを規定の時間内に受信できない場合、タイムアウトが発生し、処理が中断されたり、再試行が繰り返されたりする可能性がある。 (UCA1-T-4)コマンドやデータの到着が遅れることで、MotionSDKの処理開始が遅延し、システム全体の応答性が低下する。	(UCA1-D-1)ルーティングによって確立された通信セッションが、必要なデータ転送が完了する前に切断される。 (UCA1-D-2)不完全な通信により、システムが再試行を繰り返す。 (UCA1-D-3)必要がなくなったルーティング設定が長時間適用され続ける。
3	Motion Host	各モジュールとの通信(各トピックからの情報を要求)	Motion Host	各モジュール (IMU UART1) (LED UART2)	(UCA3-N-1)Motion Hostから各モジュールに対して、情報要求が送られない。	(UCA3-P-1)パケットに記載された宛先アドレスを第３者に書き換えられ、各モジュールに要求が到達しない。 (UCA3-P-2)パケットの内容を改ざんされる。	(UCA3-T-1)要求を各モジュールに送信するのが遅れてしまう。	(UCA3-D-1)各モジュールへの要求が到達する前に、セッションが終了する。
4	Motion SDK	UDP通信	MotionSDK	MotionHost	(UCA4-N-1)MotionSDKからMotionHostにデータが到達しない。	(UCA4-P-1)第３者からの改ざんにより、MotionSDKから意図していないMotionHostにデータが到達する。 (UCA4-P-2)MotionSDKからMotionHostに第３者からの改ざんされたデータが到達する。	(UCA4-T-1)MotionSDKからMotionHostにデータが到達するのが遅すぎる。	(UCA4-D-1)Motion SDKからMotion Hostにデータが到達する前にセッションが終了する。

に受信できない場合，タイムアウトが発生し，処理が中断されたり，再試行が繰り返されたりする可能性がある．④コマンドやデータの到着が遅れることで，Motion SDK の処理開始が遅延し，システム全体の応答性が低下する．⑤データが古くなってから Motion SDK に到着した．４つ目の，短すぎる実行または遅すぎる実行（Stop too soon/Applying too long）は３つを識別した．①ルーティングによって確立された通信セッションが，必要なデータ転送が完了する前に切断される．②不完全な通信により，システムが再試行を繰り返す，③必要がなくなったルーティング設定が長時間適用され続ける．これらの UCA は Motion SDK が UDP 通信を行っていることを基に脆弱性の観点から起こり得ると考えられる．

3.4.3 Motion Host/SDK と各モジュール通信における UCA

Motion SDK から IMU や MCU Controller，CPU などの各モジュールに情報要求を送るには，Motion Host を介して行うことになり，２つのコントロールアクションが生じる．

１つ目の提供されないことでハザードになりうる問題(Not Providing)に関して，２つのコントロールアクションとも，情報要求が送信されないことが挙げられる．また，それぞれの通信において，２つ目の，原因となり得る危険因子の提供（Providing causes hazard）に関しては，改ざんされたことで誤りを含んだ情報が送信されたり，誤った宛先に送信されるなどの各々２つを識別できた．３つ目の，早すぎる提供または遅すぎる提供（Too early/Too late）は要求をモジュールや Host に送信することが遅れてしまうことが挙げられ，４つ目の短すぎる実行または遅すぎる実行（Stop too soon/Applying too long）では，要求が到達する前に通信セッションが終了することなど各々１つが識別できた．

3.5 Step2 HCF の特定

本節では抽出した UCA 毎にハザードの要因（Hazard Causal Factor）を特定する．HCF を特定した後に，その HCF が起こり得るシナリオを想定した．HCF(Hazardous Control Factors)とは Unsafe Control Action(UCA)がハザードに結びつく原因となる要因を指し,システム構造や運用環境に内在するリスクを構造的に明らかにするものである．

3.5.1 Controller と Router での通信における HCF

異なった Router に接続することや，入力が早すぎることでの入力エラーなどといった人為的エラーの他に，攻撃による宛先改ざんや，ログイン等の認証機能における問題を起因とした HCF の特定を行った．一度既定のパスワードを利用し，ルーターにログインすると，パスワードを変更することなく利用することができるため，重大なリスクを招く可能性があると考える．

3.5.2 Router，Motion SDK の通信

四足歩行ロボットは UDP 通信を行っているのでデータの信頼性に欠けることが分かった．想定されるシナリオをもとに，主に DoS 攻撃，DDoS 攻撃，中間者攻撃によるセキュリティ上の問題を特定した．DoS 攻撃，DDoS 攻撃を受けることで四足歩行ロボットのシステム全体の応答性は低下し，正常な行動を阻害される．また，中間者攻撃ではパケットの内容を傍受されコマンドが書き換えられることにより四足歩行ロボットは本来の行動とは異なる挙動をとる．また，重大なデータの盗聴によってデバイス情報を窃取され，攻撃の手がかりにされることなどが考えられる．これらは STAMP/ STPA で，四足歩行ロボットに対する脆弱性を突き止める分析ができたことになる．

表５　Controller と Router での通信における HCF

STPA ID	UCA	HCF	シナリオ	STRIDE
HCF1-N	ユーザーが指示した情報が到達しない。	異なったRouterに接続してしまう。	接続したRouterが意図したものとは違っていた。	H
HCF1-P-1	MotionSDKに対して誤情報を送信する。	意図していないユーザーが同じルーターにログインしてしまう	2回目以降は、パスワードを入力せずともログインできることから、正規でないユーザーが同じデバイスからログインを試みる。	S,R
		意図していないユーザーが同じルーターにログインしてしまう	正規ではないユーザーが公開されたパスワードを用いて違うデバイスからログインを試みる。	S
		リモートコントローラーを使う人間の入力した内容が違った	使う人間がMotionSDKに送る入力内容を間違えてしまった	H
HCF1-P-2	MotionSDKに対して情報を送信している最中に、情報が改ざんされる。	情報を送信するまでの間に、何者かにより攻撃を受ける	リモートコントローラーからMotionSDKの間に中間者攻撃を受ける	I
HCF1-P-3	誤ったRouterに接続してしまう。	同一のIPアドレスを持つ別のルーターに接続してしまう	複数のロボットを使用していた	H
HCF1-T-1	データ送信が遅れてしまう。	ネットが混線している	一度に複数の指示を出していた	C
HCF1-T-2	膨大なデータが一気に送信される。	1つのMotionSDKに複数のリモートコントローラーをつなぐことができる	悪意があってもなくても、たまたま1つのMotionSDKに複数のリモートコントローラーが繋がってる状況になり、膨大なデータが送られる	D
HCF1-D-1	ControllerからMotionSDKまでの通信時間が長すぎる。	MotionSDK周りに障害物が乱立している	周りに障害物が多くあることにより、通信に支障が出てコントローラーからの入力内容が反映されない	P
HCF1-D-2	ControllerからMotionSDKまでの通信が終わる前にセッションが終了する。	ユーザーの入力が早すぎる	入力内容を送信した時に、一部の入力内容だけが動きに反映されていなかった	H

表６　Router と Motion SDK での通信における HCF 一部抜粋

STPA ID	UCA	HCF	シナリオ	STRIDE
HCF1-N-1	Routerからコマンドやデータが MotionSDKに到達しない。	物理的な障害物によって通信が遮断する。	厚い壁や金属等に電波が吸収または反射してしまう	P
HCF1-N-2		Routerの電源が落ちる。	Routerが作動していなかった。	P
HCF1-N-3		別のMotionSDKにコマンドやデータを送信してしまう。	送信中に宛先が改竄されていた。	T
			宛先欄に異なるMotionSDKのIPアドレスが設定されていた。	H
HCF1-N-4	ルーティングを行うための情報が提供されていない。	管理者が初期設定においてルート設定を怠っていた。	ルーティングテーブルの情報が確認できない。	H
HCF1-N-5		ソフトウェアのバグ等に、ルーティング情報が失われる。	ルーティングテーブルの情報が確認できない。	P
HCF1-P-1	誤った宛先へルーティングして、機密性の高いコマンドやデータが、本来のMotionSDKではない別のMotionSDKに到達してしまう。	送信中に宛先が改竄されていた。	コマンドの履歴から、送信先設定時には、適切なIPアドレスに設定されていたことがわかった。	T
HCF1-P-2		宛先欄に異なるMotionSDKのIPアドレスが設定されていた。	コマンドの履歴から、送信先設定時には、適切ではないIPアドレスに設定されていたことがわかった。	H
HCF1-P-1	SDKに意図していないコマンドやデータが送信される。	第３者が送信したコマンドやデータを改竄し、意図していないデータに書き換える。	コマンドの履歴から認可していない送信元からデータが発信されていることがわかった。	T
HCF1-P-2		第３者が窃取した宛先アドレス情報を利用し、コマンドやデータに送信する。	コマンドの履歴から認可していない送信元からデータが発信されていることがわかった。	S,I

表７　Motion SDK と各モジュールでの通信における HCF 一部抜粋

STPA ID	UCA	HCF	シナリオ	STRIDE

HCF3-N-1	(UCA3-N-1)Motion Hostから各モジュールに対して、情報要求が送られない。	異なったMotionSDKにデータを送信する。	返信スレッド作成時に異なった宛先を設定してしまう。
			
			〈本来〉
			Sender* send_cmd = new Sender("192.168.1.120",43893); ///< Create send thread
			〈実際〉
			Sender* send_cmd = new Sender("192.168.1.121",43893); ///< Create send thread
				H

HCF3-N-2		CAN通信の通信条件が一致していない。	終端抵抗が未設置であった。	C
HCF3-N-3		受信側でバッファが溢れてしまい、パケットが破損する。	データが連続して高速に送られていた。またはOSの割り込みにより、読み出しが間に合わなかったため、パケットが破損していた。	D
HCF3-P-1	(UCA3-P-1)要求が誤ったモジュール(ノード)に伝達される。	別のノード（モジュール）が同じCAN IDを使って別のデータを送信。	意図した宛先とは異なった宛先に届く。	S
HCF3-P-2	(UCA3-P-2)パケットの内容を改ざんされる。	パケット内のデータが第３者によって改竄された。	攻撃者が12JointsとMotionSDK間の通信経路に割り込み、正規のUDPパケットを傍受した。	T
HCF3-T-1	(UCA3-T-1)要求を各モジュールに送信するのが遅れてしまう。	タイムスタンプのずれや処理の不整合によって、データが期待される順序やタイミングよりも遅く処理されてしまう。	クロック同期のずれやソフトウェアのバグで、新しいデータが古いタイムスタンプを与えられ、受信側で早く到着した古いデータとして処理される。	C
HCF3-D-1	(UCA3-D-1)トピックへの要求が到達する前に、セッションが終了する。	電源供給の問題（瞬断など）による機器のシャットダウンする。	ネットワーク機器の電源が落ちていたことがわかった。	P

表８　Motion Host と Motion SDK での通信における HCF 一部抜粋

STPA ID	UCA	HCF	シナリオ	STRIDE
HCF4-N-1	(UCA4-N-1)MotionSDKからMotionHostにデータが到達しない。	異なったMotionHostにデータを送信する。	返信スレッド作成時に異なった宛先を設定してしまう。	H
HCF4-N-2		UDPの特性上、セッション確立時に衝突の危険性を確認しないため、パケット同士が衝突した	通信ログを確認したところ、パケットが消えていた。	C
HCF3-P-1	(UCA4-P-1)第３者からの改ざんにより、MotionSDKから意図していないMotionHostにデータが到達する。	宛先IPアドレスが、第３者によって改竄された。	パケットに付与されていたIPアドレスを確認した際に、アドレスが192.168.1.120になっていた。	T

3.5.3 Motion Host/SDK と各モジュール通信における HCF

ここでは，Motion Host と各モジュール間での通信における HCF を可視化した．CAN や UART によって通信を行っているため，CAN ID によって宛先が指定される．よって同じ ID または ID を改ざんすることで，情報要求が到達しない．または，誤った宛先に要求が届くことも考えられる．また Motion Host と各モジュール間での通信における HCF も可視化した．UDP によって通信を行っているため，宛先アドレスによって宛先が指定される．よって同じアドレスまたはアドレスを改ざんすることで，情報要求が到達しない．または，誤った宛先に要求が届くことも考えられる．

4. 分類からの考察

セーフティ・セキュリティ両面に渡って，STRIDE その他 Human Error や Communication Error，Physical Error など，多岐にわたる要因に基づくシナリオが抽出されていることが確認できた．STPA 分析は，各々のシナリオが，コントロールストラクチャのどのアクションに起因しているかが明確になり，どの安全制約に違反するから，結果的にどのようなアクシデントを招くのか特定できることが特徴である．本項では，STPA 分析にて明らかとなった HCF を STRIDE の識別子ごとに区別し，その結果から推測できることを述べる．

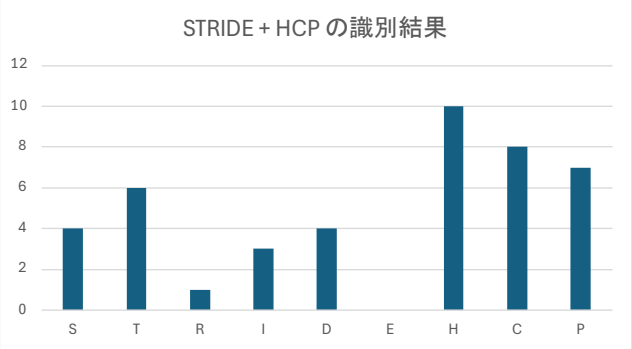

図４　シナリオ識別結果

本分析から特定できた要因とシナリオに対する共通の対策として以下のような策が挙げられる．

表９　対策一覧

S:Spoofing	**通信１　ログイン/初期接続** 公開鍵または，PSK 方式でのクライアント認証を行う．また Diffie-Hellman 方式等でセッション鍵を生成し，セッション確立を行う． **通信１ 通信２ 指令(コマンド送信) 通信３UDP** セッション鍵を用い，コマンドを AES 等で暗号化．
T:Tampering	**通信１ 通信２ 通信３** MAC 等の認証タグを付与する．
R:Repudiation	**通信１　通信２　通信３** 現在収集しているログに含まれる時刻，宛先，アクションに加えて，指令元のデバイス情報，ユーザ ID 等も要素として追加する．
I:Information Disclosure	**通信１　通信２** S(なりすまし)で提案したように，セッションを確立した上で暗号化を行う． **通信３** リアルタイムでの通信を行なっている上，軽量での暗号化が最適であり AES で軽量暗号化(UART)
D:Denial of Service	**通信１　通信２** ファイアウォールで不審な IP アドレスをパケットフィルタリングすることで制限を設けることや，信頼できる IP アドレスの UDP 通信のみを許可．また通信間の監視をした結果，短時間に大量のデータを送信し続けている IP アドレスを検知した場合受信を許可しない． **通信３** IDS 等で異常な通信を遮断(CAN)
E:Elevation of Privilege	**該当なし**
H:Human Error	**通信１　通信２　通信３** 詳細な設定については GitHub で確認することができるが，文字のみで構成されており，画像等で実際の処理画面も用いるなど，誤解の生じにくい説明に変える必要がある．
C:Communication Error	**通信１　通信２** 管理者側がデータの到達を確認し，もし正確に到達しない場合は管理者がデータの再送を行い，データの順序の信頼性の損失を防ぐ． **通信３** シーケンス番号やタイムスタンプを付与し，欠落を瞬時に判断(UDP)
P:Physical Error	システム内部に，予備電源を装備し，緊急時に切り替えられるようにする．

Spoofing(なりすまし)に対しては，ロボットに搭載されている Router にログインするための IP アドレス及びパスワードが公開されているため第３者が不正にアクセスする脅威がある．パスワードをユーザごとに変え，非公開にする

ことも一案ではあるが，通信中に第３者がログイン情報を搾取する可能性を考慮し，クライアント認証，セッションの確立，AES の暗号化等を行い，より頑健なシステム構築を行う必要がある．しかしながら，これらのシステムによる認証および暗号化はリアルタイム性を重視する通信には向かない．そのためこれらの認証はログイン及び初期接続の場合のみ発生し，ログイン時に確立したセッションでコマンドを送信するための通信を行う．

Repudiation(否認)に対しては，現在下記のようなログ収集を行なっているが，指示の内容や，各モジュールの挙動については明示されているが，送信元ユーザやデバイスに関する情報は記されていない．これらの要素を追加することで否認防止につながる．

```
[2025-07-18 09:17:55.516.204] [INFO  ] [Deeprcs] Robot ID: JY-S5-200
[2025-07-18 09:17:55.516.268] [INFO  ] [Deeprcs] Robot Version: standard
[2025-07-18 09:17:55.516.271] [INFO  ] [Deeprcs] DeepRCS is working
[2025-07-18 09:17:55.516.282] [INFO  ] [Deeprcs] Deeprcs Version: 2.0.153(153)
[2025-07-18 09:17:55.516.285] [INFO  ] [Deeprcs] DeepROS Version: 0.2.24(39)
[2025-07-18 09:17:55.516.288] [INFO  ] [Deeprcs] DeepRAS Version: 4.1.7(166)
[2025-07-18 09:17:55.578.868] [INFO  ] [Motor] Hello, CAN Send
[2025-07-18 09:17:55.578.869] [INFO  ] [RT-CAN] Hello, CAN Receive
[2025-07-18 09:17:55.579.054] [INFO  ] [CPU] Hello, CPU Monitor
[2025-07-18 09:17:55.579.091] [INFO  ] [MpvVoice] Hello, MpvVoice
[2025-07-18 09:17:55.579.111] [INFO  ] [UltrasonicSensor] Hello, Ultrasonic sensor
[2025-07-18 09:17:55.579.181] [INFO  ] [Controller] Hello, MPC Controller
[2025-07-18 09:17:55.579.300] [INFO  ] [Battery] Hello, Battery BMS
[2025-07-18 09:17:55.582.608] [INFO  ] [RT-CAN] Set motor timeout threshold = 200 ms

[2025-07-18 09:17:56.061.749] [INFO  ] [Battery] SoC = 100%, Volt = +33.100V, Temp 27
[2025-07-18 09:17:56.579.229] [INFO  ] [Deeprcs] Algorithm is start
[2025-07-18 09:17:56.588.078] [INFO  ] [Controller] !!!Origin_jumplong_left_bias = +0.000
[2025-07-18 09:17:56.588.132] [INFO  ] [basicmethod]
```

図５　収集ログ

Human Error に対しては，本分析がロボットのシステムを対象としているため人を介したシナリオ等がなく限定した範囲での分析であるため，人とロボットに関するリスクを考慮することができていない．今後分析範囲を広げ，人を交えたシナリオを増やして再度分析する必要がある．

最後に Elevation of Privilege(権限昇格)についてである．本分析では，E に該当する HCF は発見されなかった．しかしながら，そもそも権限管理をシステムが行なっていないことが今後の真の脅威になりうるのではないかと考える．通信１，２におけるログイン処理においても，各々のユーザを識別はしておらず，権限そのものに上下関係はないため，当然認証や認可等の処理も行えていない．権限管理を行うことは本分析で特定した全てのリスクに対して効果的な対策となる．

5. おわりに

本研究では，四足歩行ロボットのアーキテクチャにおける潜在的なハザードおよび安全制約を明らかにするため，STAMP/STPA を用いた構造的かつ体系的な安全分析を実施した．Router，リモートコントローラ，Motion SDK 間のホスト間通信に着目し，サブシステムを横断する脆弱性を検討した結果，Unsafe Control Action（UCA）およびそれに関連する Hazard Causal Factor（HCF）を特定した．その結果，遅延，セキュリティプロトコルの制約，データ改ざんといった重大なリスクが明らかとなった．さらに，DDoS 攻撃や中間者攻撃などのサイバー脅威に対する脆弱性という深刻なセキュリティ上の欠陥も明らかになった．結果として本研究の目的であったリスクの導出と，対策の検討を行うことができた．

今後の研究課題として，防災・防犯用途として活用できる四足歩行ロボットへの検討を行う予定である．四足歩行ロボットは，上述したように，移動性能が優れていることから防災用途での使用が期待されているが，本分析には防災を想定したリスク分析は行なっていない．STRIDE での区別においても Human Error はいくつか発見できているが，その多くが四足歩行ロボットを操作する上でのリスクのみになっており，人がロボットを利用することによるリスクは挙げられていない．したがって，今後は防災シナリオを具体的に設定した上で，人間とロボットとの相互作用や，災害現場特有の不確実性を踏まえたリスク要因の抽出が必要である．また，Human Error をより広義に捉え，操作者だけでなく被災者や第三者による意図しない介入なども考慮することで，より包括的な安全・セキュリティ要求を導出できると考えられる．また防犯・防災用途での活用検討を行う際，被災地などの環境は，動的に変化しやすく，STAMP/STPA などの静的な分析だけでは，対応しきれない．災害発生時の動的に変化する被害状況（例: 道路寸断，建物の損壊，ライフライン停止）をテーマとし，我々が提案する STAMP S&S: Layered Modeling for the complexed system in the society of AI/IoT[7]を具体化して，動的にリスクを識別し，対策を明確化する方法論の確立を目指していきたい．

参考文献

[1] IPA 独立行政法人情報処理推進機構. (2016). はじめての STAMP/STPA ~システム 思考に基づく新しい安全性解析手法~. https://www.ipa.go.jp/digital/stamp/ug65p90000001lxs-att/000055009.pdf. (参照 2025-1-12) .

[2] 金子朋子. (2021). セーフティ&セキュリティ入門 AI,IoT 時代のシステム安全. 日科技連出版社.

[3] Tomoko Kaneko, Yuji Takahashi, Takao Okubo, Ryoichi Sasaki. (2018). Threat analysis using STRIDE with STAMP/STPA, APSEC2018

[4] DeepRoboticsLab, ”DEEPRoboticsLab” ,GitHub. https://github.com/DeepRoboticsLab, （参照 2025 年 8 月 4 日）.

[5] Leveson, N. G. (2024). システム理論による安全工学 ―想定外に気づくための思考法 STAMP― (兼本 茂, 福島 祐子, 青木 善貴, 石井 正悟, 岡本 圭史, 沖汐 大志, 片平 真史, 金子 朋子, 日下部 茂, 野本 秀樹, 橋本 岳男, 向山 輝, 山口 晋一, 吉岡 信和, 余宮 尚志). 共立出版.

[6] Princelove Smith Harvard. (2025). Securing Quadruped Robotics in Search and Rescue: Identifying Vulnerabilities and Enhancing Decision Intelligence. SOMET 2025: The 24th International Conference on Intelligent Software Methodologies, Tools, and Techniques.

[7] Tomoko Kaneko, Nobukazu Yoshioka. (2020). STAMP S&S: Layered Modeling for the complexed system in the society of AI/IoT. 13th International Joint Conference on Knowledge-Based Software Engineering. (JCKBSE2020) 122-131.