LLM-based scene graph for Outdoor Human-Robot Teaming in Society 6.0 – Design, applications, and evaluation

Trung Kien La and Eric Guiffo Kaigom

Abstract**——[Society 5.0/6.0 here?]. . . This article describes the design, application and implementation of a multimodal assistance system to support [visually impaired] persons. To this end, a mobile robot is equipped with a depth camera, object detection and large language models (LLMs). The goal is to enable the robot to understand its environment, i.e., to perceive, understand, and respond to it. Additionally, the robot should describe its perceived environment in natural language. A graphical user interface has been developed to bundle all asynchronous processes and act as a central audiovisual control unit. Visual perception is achieved through data fusion of the depth camera and object recognition models. The resulting 3D object data is then used to implement robot navigation. Human-machine interaction takes place via a voice interface that uses speech recognition and a text-to-speech system for speech output. To enrich the scene description, local multimodal LLMs are used. For this purpose, a client-server architecture was established. [mention of results and conducted experiments?] . . . [reference/benefits to Society 6.0?]. . . .**

Index Terms**—Article submission, IEEE, IEEEtran, journal, LaTeX, paper, template, typesetting.**

I. Introduction

THE society of the future will integrate emerging technologies to promote human well-being and enhance quality of life in sustainable and equitable ways. A key objective of this transformation is the establishment of accountable corporate leadership that responsibly affects stakeholders and their environments. Equally important are collective intelligence and the self-determination of individuals in their increasingly mobile professional and social activities. However, the effectiveness of distributed, interconnected, and inclusive autonomy requires perceptual and interactive capacities that extend beyond human limitations, particularly those related to endurance, precision, and feedback in motor skills, vision, and cognition. AI-powered technologies are thus designed to integrate data processing, intelligent interfaces, and adaptive actuators to inform, augment, and extend human abilities. These systems enhance situational awareness while reducing cognitive load, strengthen personalized responses to societal challenges, and promote flexible, natural interaction with minimal effort. The use of environmentally sustainable and everyone-accessible technologies—as personal assistants in home care, adaptive interfaces in rehabilitation, and immersive tools in education and training—can mitigate cultural, physical and cognitive barriers while fostering new individual capacities and professional opportunities. Service robots constitute a prominent class of such technologies. They increasingly demonstrate their capacity to simplify and streamline daily human activities. Robots can transport heavy or critical payloads at customizable speeds with high precision. These capabilities can be achieved in a climate-friendly manner when powered by renewable energy sources such as solar and wind systems. Despite this societal and environmental potential, robot adoption remains constrained by factors beyond cost and safety. First, current systems lack functionalities that enable a transition from prescripted task execution to broader perception, interpretation, and real-time knowledge sharing with humans. Second, non-specialists often face difficulties in translating high-level task intentions into robotic actions. Third, existing designs seldom support synergy and role switching between humans and robots during joint task execution. Fourth, robots typically fail to communicate their internal states and feedback in accessible and intelligible ways. To address these limitations, this work presents an AI-driven infrastructure for human–robot teaming across a wide spectrum of societal applications. These include indoor assistance for the elderly—in which case robots act as followers or commissioned agents for payload transport—and outdoor guidance for individuals with visual impairments. Large Language Models (LLMs) are leveraged to establish a bidirectional, natural-language communication between humans and robots. By exploiting linguistic subtleties, the system dynamically acquires contextual understanding and supports high-level task planning in interchangeable languages, thereby fostering and accelerating inclusion, participation, and collective intelligence. The robot operates on a scene graph that delivers a multimodal, structured representation of its environment and internal behavior, encompassing potential obstacles and energy consumption. This comprehensive model enables the robot to explain its decisions and offer quantitative insights into their underlying causes upon request. Such feedback facilitates adaptive task refinement and energy optimization. Furthermore, the robot is equipped with master and follower capabilities that support dynamic role switching. Humans retain strategic decision-making authority, guided by information provided by the robot. The resulting extended machine intelligence demonstrates utility across diverse applications,

Trung Kien La was with the Frankfurt Industrial Robotics and Digital Twin Laboratory (FriiDA) at the Frankfurt University of Applied Sciences, Hungener Str. 6 Building C, 60389 Frankfurt am Main, Germany (Orcid: 0009-0007-9239-3268).

Eric Guiffo Kaigom is with the Frankfurt Industrial Robotics and Digital Twin Laboratory (FriiDA) at the Frankfurt University of Applied Sciences, Hungener Str. 6 Building C, 60389 Frankfurt am Main, Germany (e-mail: kaigom@fra-uas.de).

from assistive domestic environments to outdoor navigation and cooperative task execution.

II. Related Work

- First short definition of scene graph
- Comparison with other scene graph generation tools and frameworks such as [Concept Graph, OpenIN, HOVSG, Graph2Nav, 3DGraphLLM, SG-Nav, Interaction-Driven Updates: 3D Scene Graph Maintenance During Robot Task Execution, VLMaps] (need mapping beforehand and/or need powerful and additional hardware for models such as Grounding DINO, CLIP, GLIP, SAM, BLIP-2 and even additional drones for mapping, which does not suit our case [real-time, dynamic and unseen environment, reactive, portable/mobile, cost-effective or free/local models, audio interaction, suitable outdoors]). Nearly all the frameworks are used indoors.
- Most similar:
- Can an Embodied Agent Find Your –Cat-shaped Mug? LLM-Based Zero-Shot Object Navigation (uses depth information, but needs mapping)
- VISIONGPT (YOLO World, ChatGPT, voice input and feedback)
- Cobots & Service Robots & Society 5.0/6.0

A scene graph (SG) serves to represent a scene with objects, their attributes, and relationships between other objects in a structured way. Scene graph generation (SGG) can be based on images, text, and videos. Such a hierarchical graph data structure consists of nodes (O) and edges (E). Nodes (O) are objects such as persons, animals, places or things (bicycle, car, etc.), but also parts of other objects (e.g., limbs such as a person's arms). Each object node is usually equipped with attributes that describe the state of the object (e.g., colors, size, or pose). The edges (E) represent the relationships between the object pairs. These connections between objects describe actions (man holds cup) or positions (man stands in front of a bench). Typically, such a relationship is expressed as a ⟨subject − predicate − object⟩ triple. An SG can be defined as a tuple:

$$SG = (\mathcal{O}, R, E) \tag{1}$$

Where:

- $\mathcal{O} = \{o_1, \ldots, o_n\}$ is the set of objects recognized in the images, where n is the number of objects. Each object o_i can also be described as a tuple:

  $$o_i = (c_i, A_i) \tag{2}$$

  where c_i denotes the category (e.g., person, soccer stadium, animal) and A_i denotes the attributes (e.g., shape, color, pose) of the object.
- R stands for the set of relationships between the nodes (objects).
- $E \subseteq \mathcal{O} \times R \times \mathcal{O}$ represents the edges between the object instance nodes and the relationship nodes. The edge set E consists of exactly those triples that are considered relationships between the object pairs.

Using the following formulation, scene graphs can be expressed from an image I using SGG methods:

$$SG(\mathcal{O}, R, E) = SGG(I) \tag{3}$$

SGG employs diverse models and technologies across robotics, computer vision, autonomous navigation, and human-machine interaction. These are summarized in Table I.

III. Methodology

This section presents the developed methods and their implementation.

A. Overview of Features and Components

Our framework pAIrSEEption is a multimodal voice assistance system for environmental detection and navigation with a responsive graphical user interface (GUI) based on PySide6, which serves as the interface between humans and the machine. Fig.2 illustrates the features of the user interface. The GUI can be installed across various platforms. In this case, the interface was installed on the Jetson AGX Orin (32GB version), an edge device and on a standard laptop with an integrated Nvidia graphics card. Subsequently, a ZED 2i stereo camera can be connected to these devices. The mobile Husky robot is utilized as the mounting platform for the components as shown in Fig.1.

Fig. 1. Placeholder for Husky Robot as a mounting platform for the ZED 2i camera.

The GUI acts as a visualization and control center for the user. All data streams from the various components are bundled here. The interface displays the following visual information:

- A live, annotated video stream of the stereo camera from the perception data pipeline.
- System performance statistics, such as frames per second (FPS) and the percentage utilization of the central processing unit (CPU), graphics processing unit (GPU), video random access memory (VRAM), and random-access memory (RAM).
- Status and system messages for active and inactive processes
- Transcribed user voice commands

TABLE I
TECHNICAL OVERVIEW OF MODELS AND TECHNOLOGIES FOR 3D SCENE GRAPH GENERATION, PERCEPTION AND NAVIGATION

System	**Detection & Segmentation (Nodes)**	**Semantic Features & Embeddings**	**Relation Inference & Planning (Edges)**	**Alternatives & Additional Models**
ConceptGraphs	SAM (Segment Anything)	CLIP, DINO	GPT-4 (LLM), LLaVA	Grounding DINO, RAM
3DGraphLLM	Mask3D, OneFormer3D	DINOv2 (2D), Uni3D (3D)	VL-SAT, CLIP	LLaMA3-8B, Vicuna-1.5
Graph2Nav	PSGFormer, Mask2Former	Panoptic Labels	2D-PSG Inference, GPT-3.5	LIO-SAM, Whisper
HOV-SG	SAM	CLIP	Hierarchical Heuristics	GPT-3.5, GPT-4
OpenIN	CropFormer	SBERT, CLIP	GPT-4o (Commonsense)	Tokenize Anything
SG-Nav	Online 3D Instance Seg.	VLMs, LLMs	LLaMA-7B, GPT-4, LLaVA	Fast Marching Method
Interaction Updates	Existing 3DSG (e.g., Hydra)	Visual Transformer (ViT)	GPT-4, BLIP-2	DBSCAN (Denoising)
VisionGPT	YOLO-World	Proportional 2D Coordinates	GPT-3.5 Turbo, GPT-4	H-Pattern Splitter
LGX	GLIP, YOLO	BLIP (Captions)	GPT-3	OWL-ViT, Detectron2
VLMaps	LSeg	CLIP	Codex (Code-writing LLM)	RTAB-Map

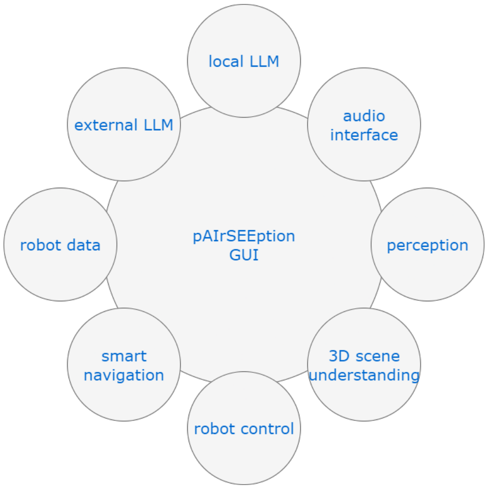

Fig. 2. Husky Robot as a mounting platform for the ZED 2i camera.

- Scene descriptions and casual chat as text generated by the LLM, which are created using object data and images from the current scene
- Navigation control overview to select a recognized object to navigate to and/or to follow
- In addition, internal robot data such as motor temperatures can be queried via voice input and are displayed in the GUI

All data and information displayed on the interface, as well as user (voice) input, can be saved locally in common file formats via the GUI, facilitating any necessary fine-tuning or training of the models (YOLO and LLM) in the future.

B. Setup and Applications

The interface provides a selection of different versions and sizes of YOLO(e) models for object detection and segmentation as well as LLMs for scene interpretation, which can be expanded as needed. In addition, specific YOLO settings can be configured, such as showing and hiding bounding boxes, classification, and confidence. With an additional confidence slide bar, objects that are not well recognized can be filtered, which are below the set threshold. The YOLO object detection is leveraged for two functions. First, it is used for data fusion with the ZED 2i depth camera to provide object data (see Fig. **??**), which is then fed to the LLM as a prompt. [The ZED SDK provides high-resolution stereo images and a depth map, which shows the distance of each pixel in the image. It also offers position tracking and a velocity estimation of the objects via the Python API, pyzed. This API can be used to configure camera parameters, such as resolution, FPS, depth mode, and depth measurement units. The resolution is set to 1280 x 720 pixels by default, with 60 FPS and performance mode for depth detection enabled. The unit of measurement is set to meters. The origin of the stereo camera is set to the coordinates (x=0, y=0, z=0). For compatibility reasons, the camera's 3D coordinate system has been set to match the one used in Robot Operating System 2 (ROS2), a middleware framework for robots, since the mobile robot is based on it.] Second, it is used to navigate and follow a recognized object. To this end, the distance from each detected object to the camera and the pairwise inter-object distances within a frame are computed using the 3D Euclidean distance. The selection of the target object is initiated through its unique identification (ID), which is facilitated by either vocal input or interface selection.

In pAIrSEEption, the objects $\mathcal{O} = \{o_1, \dots, o_n\}$ in the scene are schematically defined as follows (for example, a recognized person standing still):

$$o_1 = \left(\text{person}, \begin{Bmatrix} \text{ID} = 1, \\ \text{confidence value} = 0.95, \\ \text{3D position} = (x, y, z), \\ \text{3D speed} = (v_x = 0, v_y = 0, v_z = 0), \\ \text{speed classification} = \text{static} \end{Bmatrix}\right)$$

Where the object o_1 receives the attributes ID, confidence value, 3D positions, 3D velocity, and the classification *static* if the velocity is zero. Thus, *static* is an attribute derived from velocity, and the attribute set *A* is expanded with this additional information. The selection of LLMs can be divided into two categories:

The first category is local and offline models from Ollama such as Gemma 3 or Qwen 2.5 VL, hosted on consumer PCs in the laboratory or on an edge device like the Jetson AGX Orin attached to the mobile Husky UGV Robot (see Fig. **??**). This solution offers the advantage of control and data protection. The second category is online, hosted models via the OpenRouter API. With a single API key, it can

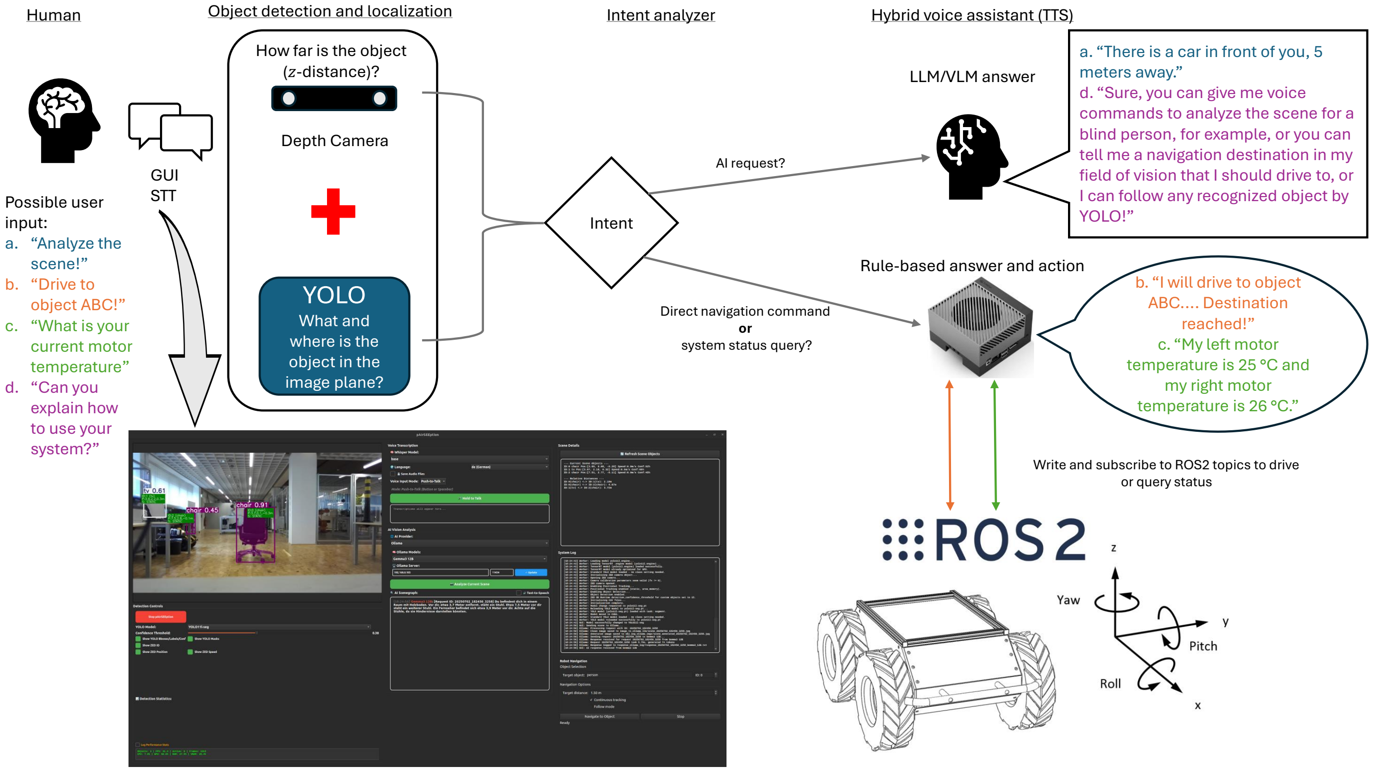
Human
Object detection and localization
Intent analyzer
Hybrid voice assistant (TTS)
How far is the object (z-distance)?
Depth Camera
YOLO
What and where is the object in the image plane?
GUI
STT
Possible user input:
a. "Analyze the scene!"
b. "Drive to object ABC!"
c. "What is your current motor temperature"
d. "Can you explain how to use your system?"
Intent
AI request?
LLM/VLM answer
a. "There is a car in front of you, 5 meters away."
d. "Sure, you can give me voice commands to analyze the scene for a blind person, for example, or you can tell me a navigation destination in my field of vision that I should drive to, or I can follow any recognized object by YOLO!"
Rule-based answer and action
Direct navigation command
or
system status query?
b. "I will drive to object ABC.... Destination reached!"
c. "My left motor temperature is 25 °C and my right motor temperature is 26 °C."
Write and subscribe to ROS2 topics to drive or query status
ROS 2
z
Yaw
y
Pitch
Roll
x
chair 0.45
chair 0.91

Fig. 3. Overview of the system architecture.

connect to many different models from OpenAI, Google, Meta, Anthropic, etc. Most of these are subject to a fee per query and have a request limit. In this case, factors such as server availability and data security are not within the user's purview.

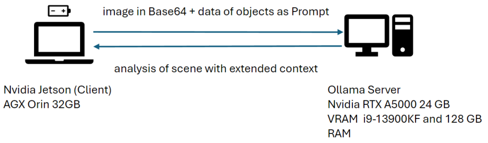
image in Base64 + data of objects as Prompt
analysis of scene with extended context
Nvidia Jetson (Client)
AGX Orin 32GB
Ollama Server
Nvidia RTX A5000 24 GB
VRAM i9-13900KF and 128 GB
RAM

Fig. 4. Client-server architecture between application device and Ollama server.

If a request for a description of the scene is initiated, the Jetson or laptop transmits an image of the current scene to the Ollama server—or, depending on the selection, also to OpenRouter—with the previously merged object data as a prompt to the server for interpretation and evaluation. The following text-prompt instructs the LLM on processing data. The goal is to enable safe navigation for a visually impaired person:

```
Describe the image briefly and
precisely using the object data
provided.

{object_description}

Please describe what can be seen in
the image, where important objects are
located, and provide information about
possible obstacles.
```

The object data, defined as *object_description* is provided in the prompt and has been translated into concise bullet points during the preprocessing stage. Initially, a template for the raw data was employed. Fig. **??** presents a schematic representation of the result of this transformation using the template. The integration of image data with readable object data through the utilization of an LLM (Language Model) facilitates the creation of a comprehensive and cohesive representation of the subject matter. The following description provides a detailed interpretation of a 3D scene that is accessible to humans. The result of the scene interpretation is displayed as text in the output window of the GUI and can optionally be played back via an audio file generated by activating the text-to-speech function.

C. *Navigation and motion control*

IV. Other Resources

See [1]–[5] for resources on formatting math into text and additional help in working with LaTeX.

V. Text

For some of the remainer of this sample we will use dummy text to fill out paragraphs rather than use live text that may violate a copyright.

Itam, que ipiti sum dem velit la sum et dionet quatibus apitet voloritet audam, qui aliciant voloreicid quaspe volorem ut maximusandit faccum conemporerum aut ellatur, nobis arcimus. Fugit odi ut pliquia incitium latum que cusapere perit

Fig. 5. Placeholder.

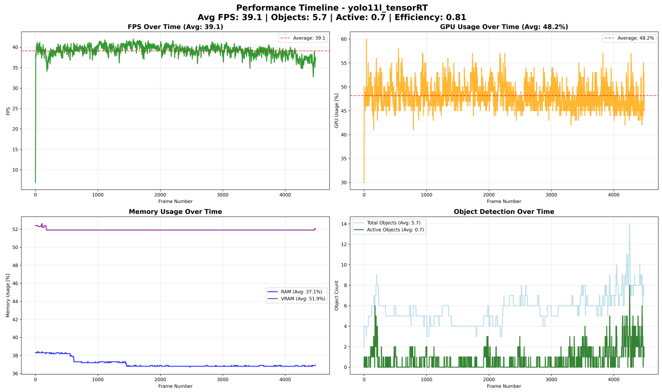

Fig. 6. Client-server architecture between application device and Ollama server.

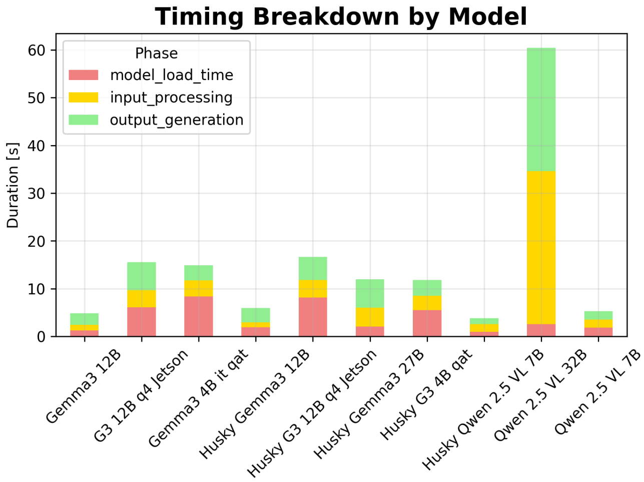

Fig. 7. Client-server architecture between application device and Ollama server.

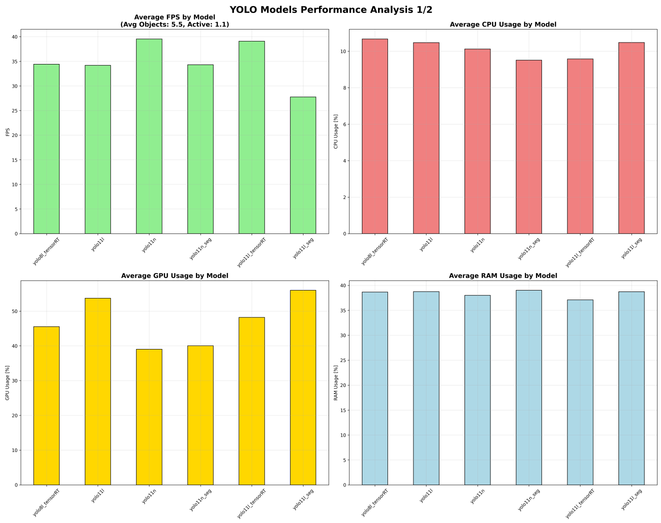

Fig. 8. Client-server architecture between application device and Ollama server.

molupta eaquaeria quod ut optatem poreiur? Quiaerr ovitior suntiant litio bearciur?

Onseque sequaes rectur autate minullore nusae nestiberum, sum voluptatio. Et ratem sequiam quaspername nos rem repudandae volum consequis nos eium aut as molupta tectum ulparumquam ut maximillesti consequas quas inctia cum volectinusa porrum unt eius cusaest exeritatur? Nias es enist fugit pa vollum reium essusam nist et pa aceaqui quo elibusdandis deligendus que nullaci lloreri bla que sa coreriam explacc atiumquos simolorpore, non prehendunt lam que occum [6] si aut aut maximus eliaeruntia dia sequiamenime natem sendae ipidemp orehend uciisi omnienetus most verum, ommolendi omnimus, est, veni aut ipsa volendelist mo conserum volores estisciis recessi nveles ut poressitatur sitiis ex endi diti volum dolupta aut aut odi as eatquo cullabo remquis toreptum et des accus dolende pores sequas dolores tinust quas expel moditae ne sum quiatis nis endipie nihilis etum fugiae audi dia quiasit quibus. Ibus el et quatemo luptatque doluptaest et pe volent rem ipidusa eribus utem venimolorae dera qui acea quam etur aceruptat. Gias anis doluptaspic tem et aliquis alique inctiuntiur?

Sedigent, si aligend elibuscid ut et ium volo tem eictore pellore ritatus ut ut ullatus in con con pere nos ab ium di tem aliqui od magnit repta volectur suntio. Nam isquiante doluptis essit, ut eos suntionsecto debitiur sum ea ipitiis adipit, oditiore, a dolorerempos aut harum ius, atquat.

Rum rem ditinti sciendunti volupiciendi sequiae nonsect oreniatur, volores sition ressimil inus solut ea volum harumqui to see(4) mint aut quat eos explis ad quodi debis deliqui aspel earcius.

$$x = \sum_{i=0}^{n} 2iQ. \tag{4}$$

Alis nime volorempera perferi sitio denim repudae pre ducilit atatet volecte ssimillorae dolore, ut pel ipsa nonsequiam in re nus maiost et que dolor sunt eturita tibusanis eatent a aut et dio blaudit reptibu scipitem liquia consequodi od unto ipsae. Et enitia vel et experferum quiat harum sa net faccae dolut voloria nem. Bus ut labo. Ita eum repraer rovitia samendit aut et volupta tecupti busant omni quiae porro que nossimodic temquis anto blacita conse nis am, que ereperum eumquam quaescil imenisci quae magnimos recus ilibeaque cum etum iliate prae parumquatemo blaceaquiam quundia dit apienditem rerit re eici quaes eos sinvers pelecabo. Namendignis as exerupit aut magnim ium illabor roratecte plic tem res apiscipsam et vernat untur a deliquaest que non cus eat ea dolupiducim fugiam volum hil ius dolo eaquis sitis aut landesto quo corerest et auditaquas ditae voloribus, qui optaspis exero cusa am, ut plibus.

VI. Some Common Elements

A. Sections and Subsections

Enumeration of section headings is desirable, but not required. When numbered, please be consistent throughout the article, that is, all headings and all levels of section headings in the article should be enumerated. Primary headings are designated with Roman numerals, secondary with capital letters, tertiary with Arabic numbers; and quaternary with lowercase letters. Reference and Acknowledgment headings are unlike all other section headings in text. They are never enumerated. They are simply primary headings without labels, regardless of whether the other headings in the article are enumerated.

B. Citations to the Bibliography

The coding for the citations is made with the LaTeX `\cite` command. This will display as: see [1].

For multiple citations code as follows: `\cite{ref1,ref2,ref3}` which will produce [1]–[3]. For reference ranges that are not consecutive code as `\cite{ref1,ref2,ref3,ref9}` which will produce [1]–[3], [9]

C. Lists

In this section, we will consider three types of lists: simple unnumbered, numbered, and bulleted. There have been many options added to IEEEtran to enhance the creation of lists. If your lists are more complex than those shown below, please refer to the original "IEEEtran_HOWTO.pdf" for additional options.

A plain unnumbered list:

bare_jrnl.tex
bare_conf.tex
bare_jrnl_compsoc.tex
bare_conf_compsoc.tex
bare_jrnl_comsoc.tex

A simple numbered list:

1) bare_jrnl.tex
2) bare_conf.tex
3) bare_jrnl_compsoc.tex
4) bare_conf_compsoc.tex
5) bare_jrnl_comsoc.tex

Fig. 9. Simulation results for the network.

A simple bulleted list:

- bare_jrnl.tex
- bare_conf.tex
- bare_jrnl_compsoc.tex
- bare_conf_compsoc.tex
- bare_jrnl_comsoc.tex

D. Figures

Fig. 1 is an example of a floating figure using the graphicx package. Note that `\label` must occur AFTER (or within) `\caption`. For figures, `\caption` should occur after the `\includegraphics`.

Fig. 2(a) and 2(b) is an example of a double column floating figure using two subfigures. (The subfig.sty package must be loaded for this to work.) The subfigure `\label` commands are set within each subfloat command, and the `\label` for the overall figure must come after `\caption`. `\hfil` is used as a separator to get equal spacing. The combined width of all the parts of the figure should do not exceed the text width or a line break will occur.

Note that often IEEE papers with multi-part figures do not place the labels within the image itself (using the optional argument to `\subfloat`[]), but instead will reference/describe all of them (a), (b), etc., within the main caption. Be aware that for subfig.sty to generate the (a), (b), etc., subfigure labels, the optional argument to `\subfloat` must be present. If a subcaption is not desired, leave its contents blank, e.g.,`\subfloat`[].

VII. Tables

Note that, for IEEE-style tables, the `\caption` command should come BEFORE the table. Table captions use title case. Articles (a, an, the), coordinating conjunctions (and, but, for, or, nor), and most short prepositions are lowercase unless they are the first or last word. Table text will default to `\footnotesize` as the IEEE normally uses this smaller font for tables. The `\label` must come after `\caption` as always.

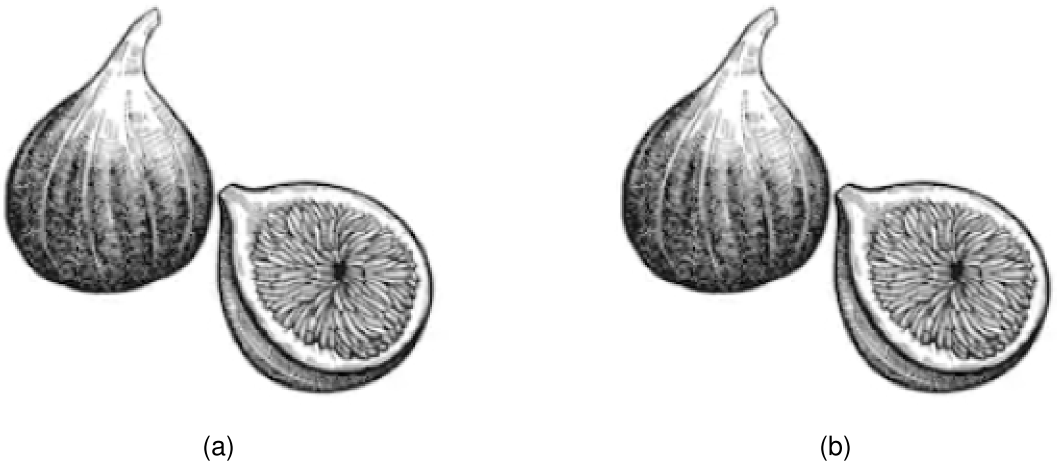

Fig. 10. Dae. Ad quatur autat ut porepel itemoles dolor autem fuga. Bus quia con nessunti as remo di quatus non perum que nimus. (a) Case I. (b) Case II.

TABLE II
AN EXAMPLE OF A TABLE

One	Two
Three	Four

VIII. ALGORITHMS

Algorithms should be numbered and include a short title. They are set off from the text with rules above and below the title and after the last line.

Algorithm 1 Weighted Tanimoto ELM.

```
TRAIN($\mathbf{XT}$)
    select randomly $W \subset \mathbf{X}$
    $N_{\mathbf{t}} \leftarrow |\{i : \mathbf{t}_i = \mathbf{t}\}|$  for  $\mathbf{t} = -1, +1$
    $B_i \leftarrow \sqrt{\textsc{max}(N_{-1}, N_{+1})/N_{\mathbf{t}_i}}$  for  $i = 1, ..., N$
    $\hat{\mathbf{H}} \leftarrow B \cdot (\mathbf{X}^T\mathbf{W})/(\nVdash\mathbf{X} + \nVdash\mathbf{W} - \mathbf{X}^T\mathbf{W})$
    $\beta \leftarrow \left(I/C + \hat{\mathbf{H}}^T\hat{\mathbf{H}}\right)^{-1}(\hat{\mathbf{H}}^T B \cdot \mathbf{T})$
    return $\mathbf{W}, \beta$

PREDICT($\mathbf{X}$)
    $\mathbf{H} \leftarrow (\mathbf{X}^T\mathbf{W})/(\nVdash\mathbf{X} + \nVdash\mathbf{W} - \mathbf{X}^T\mathbf{W})$
    return $\textsc{sign}(\mathbf{H}\beta)$
```

Que sunt eum lam eos si dic to estist, culluptium quid qui nestrum nobis reiumquiatur minimus minctem. Ro moluptat fuga. Itatquiam ut laborpo rersped exceres vollandi repudaerem. Ulparci sunt, qui doluptaquis sumquia ndestiu sapient iorepella sunti veribus. Ro moluptat fuga. Itatquiam ut laborpo rersped exceres vollandi repudaerem.

IX. MATHEMATICAL TYPOGRAPHY AND WHY IT MATTERS

Typographical conventions for mathematical formulas have been developed to **provide uniformity and clarity of presentation across mathematical texts**. This enables the readers of those texts to both understand the author's ideas and to grasp new concepts quickly. While software such as LaTeX and MathType® can produce aesthetically pleasing math when used properly, it is also very easy to misuse the software, potentially resulting in incorrect math display.

IEEE aims to provide authors with the proper guidance on mathematical typesetting style and assist them in writing the best possible article. As such, IEEE has assembled a set of examples of good and bad mathematical typesetting [1]–[5].

Further examples can be found at http://journals.ieeeauthorcenter.ieee.org/wp-content/uploads/sites/7/IEEE-Math-Typesetting-Guide-for-LaTeX-Users.pdf

A. Display Equations

The simple display equation example shown below uses the "equation" environment. To number the equations, use the `\label` macro to create an identifier for the equation. LaTeX will automatically number the equation for you.

$$x = \sum_{i=0}^{n} 2iQ. \tag{5}$$

is coded as follows:

```
\begin{equation}
\label{deqn_ex1}
x = \sum_{i=0}^{n} 2{i} Q.
\end{equation}
```

To reference this equation in the text use the `\ref` macro. Please see (5)
is coded as follows:

```
Please see (\ref{deqn_ex1})
```

B. Equation Numbering

Consecutive Numbering: Equations within an article are numbered consecutively from the beginning of the article to the end, i.e., (1), (2), (3), (4), (5), etc. Do not use roman numerals or section numbers for equation numbering.

Appendix Equations: The continuation of consecutively numbered equations is best in the Appendix, but numbering as (A1), (A2), etc., is permissible.

Hyphens and Periods: Hyphens and periods should not be used in equation numbers, i.e., use (1a) rather than (1-a) and (2a) rather than (2.a) for subequations. This should be consistent throughout the article.

C. Multi-Line Equations and Alignment

Here we show several examples of multi-line equations and proper alignments.
A single equation that must break over multiple lines due to length with no specific alignment.

$$\begin{multline}\text{The first line of this example}\\ \text{The second line of this example}\\ \text{The third line of this example} \tag{6}\end{multline}$$

is coded as:

```
\begin{multline}
\text{The first line of this example}\\
\text{The second line of this example}\\
\text{The third line of this example}
\end{multline}
```

A single equation with multiple lines aligned at the = signs

$$\begin{align} a &= c+d \tag{7}\\ b &= e+f \tag{8}\end{align}$$

is coded as:

```
\begin{align}
a &= c+d \\
b &= e+f
\end{align}
```

The `align` environment can align on multiple points as shown in the following example:

$$\begin{align} x &= y & X &= Y & a &= bc \tag{9}\\ x' &= y' & X' &= Y' & a' &= bz \tag{10}\end{align}$$

is coded as:

```
\begin{align}
x &= y & X & =Y & a &=bc\\
x’ &= y’ & X’ &=Y’ &a’ &=bz
\end{align}
```

D. Subnumbering

The amsmath package provides a `subequations` environment to facilitate subnumbering. An example:

$$\begin{align} f &= g \tag{11a}\\ f' &= g' \tag{11b}\\ \mathcal{L}f &= \mathcal{L}g \tag{11c}\end{align}$$

is coded as:

```
\begin{subequations}\label{eq:2}
\begin{align}
f&=g \label{eq:2A}\\
f’ &=g’ \label{eq:2B}\\
\mathcal{L}f &= \mathcal{L}g \label{eq:2c}
\end{align}
\end{subequations}
```

E. Matrices

There are several useful matrix environments that can save you some keystrokes. See the example coding below and the output.
A simple matrix:

$$\begin{matrix} 0 & 1 \\ 1 & 0 \end{matrix} \tag{12}$$

is coded as:

```
\begin{equation}
\begin{matrix} 0 & 1 \\
1 & 0 \end{matrix}
\end{equation}
```

A matrix with parenthesis

$$\begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix} \tag{13}$$

is coded as:

```
\begin{equation}
\begin{pmatrix} 0 & -i \\
 i & 0 \end{pmatrix}
\end{equation}
```

A matrix with square brackets

$$\begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} \tag{14}$$

is coded as:

```
\begin{equation}
\begin{bmatrix} 0 & -1 \\
1 & 0 \end{bmatrix}
\end{equation}
```

A matrix with curly braces

$$\begin{Bmatrix} 1 & 0 \\ 0 & -1 \end{Bmatrix} \tag{15}$$

is coded as:

```
\begin{equation}
\begin{Bmatrix} 1 & 0 \\
0 & -1 \end{Bmatrix}
\end{equation}
```

A matrix with single verticals

$$\begin{vmatrix} a & b \\ c & d \end{vmatrix} \tag{16}$$

is coded as:

```
\begin{equation}
\begin{vmatrix} a &  b \\
c &  d \end{vmatrix}
\end{equation}
```

A matrix with double verticals

$$\begin{Vmatrix} i & 0 \\ 0 & -i \end{Vmatrix} \tag{17}$$

is coded as:

```
\begin{equation}
\begin{Vmatrix} i &  0 \\
0 & -i \end{Vmatrix}
\end{equation}
```

F. Arrays

The `array` environment allows you some options for matrix-like equations. You will have to manually key the fences, but there are other options for alignment of the columns and for setting horizontal and vertical rules. The argument to `array` controls alignment and placement of vertical rules.

A simple array

$$\left(\begin{array}{cccc} a+b+c & uv & x-y & 27\\ a+b & u+v & z & 134 \end{array} \right) \tag{18}$$

is coded as:

```
\begin{equation}
\left(
\begin{array}{cccc}
a+b+c & uv & x-y & 27\\
a+b & u+v & z & 134
\end{array} \right)
\end{equation}
```

A slight variation on this to better align the numbers in the last column

$$\left(\begin{array}{cccr} a+b+c & uv & x-y & 27\\ a+b & u+v & z & 134 \end{array} \right) \tag{19}$$

is coded as:

```
\begin{equation}
\left(
\begin{array}{cccr}
a+b+c & uv & x-y & 27\\
a+b & u+v & z & 134
\end{array} \right)
\end{equation}
```

An array with vertical and horizontal rules

$$\left(\begin{array}{c|c|c|r} a+b+c & uv & x-y & 27\\ \hline a+b & u+v & z & 134 \end{array} \right) \tag{20}$$

is coded as:

```
\begin{equation}
\left(
\begin{array}{c|c|c|r}
a+b+c & uv & x-y & 27\\
a+b & u+v & z & 134
\end{array} \right)
\end{equation}
```

Note the argument now has the pipe ”|” included to indicate the placement of the vertical rules.

G. Cases Structures

Many times cases can be miscoded using the wrong environment, i.e., `array`. Using the `cases` environment will save keystrokes (from not having to type the `\left\lbrace`) and automatically provide the correct column alignment.

$$z_m(t) = \begin{cases} 1, & \text{if } \beta_m(t) \\ 0, & \text{otherwise.} \end{cases}$$

is coded as follows:

```
\begin{equation*}
{z_m(t)} =
\begin{cases}
1,&{\text{if}}\ {\beta }_m(t),\\
{0,}&{\text{otherwise.}}
\end{cases}
\end{equation*}
```

Note that the “&” is used to mark the tabular alignment. This is important to get proper column alignment. Do not use `\quad` or other fixed spaces to try and align the columns. Also, note the use of the `\text` macro for text elements such as “if” and “otherwise.”

H. Function Formatting in Equations

Often, there is an easy way to properly format most common functions. Use of the \ in front of the function name will in most cases, provide the correct formatting. When this does not work, the following example provides a solution using the `\text` macro:

$$d_{R}^{KM} = \underset {d_{l}^{KM}} {\text{arg min}} \{ d_{1}^{KM}, \ldots,d_{6}^{KM}\}.$$

is coded as follows:

```
\begin{equation*}
 d_{R}^{KM} = \underset {d_{l}^{KM}}
 {\text{arg min}} \{ d_{1}^{KM},
 \ldots,d_{6}^{KM}\}.
\end{equation*}
```

I. Text Acronyms Inside Equations

This example shows where the acronym “MSE” is coded using `\text{}` to match how it appears in the text.

$$\text{MSE} = \frac {1}{n}\sum _{i=1}^{n}(Y_{i} - \hat {Y_{i}})^{2}$$

```
\begin{equation*}
 \text{MSE} = \frac {1}{n}\sum _{i=1}^{n}
(Y_{i} - \hat {Y_{i}})^{2}
\end{equation*}
```

X. Conclusion

The conclusion goes here.

Acknowledgments

This should be a simple paragraph before the References to thank those individuals and institutions who have supported your work on this article.

Appendix
Proof of the Zonklar Equations

Use \appendix if you have a single appendix: Do not use \section anymore after \appendix, only \section*. If you have multiple appendixes use \appendices then use \section to start each appendix. You must declare a \section before using any \subsection or using \label (\appendices by itself starts a section numbered zero.)

References Section

You can use a bibliography generated by BibTeX as a .bbl file. BibTeX documentation can be easily obtained at: http://mirror.ctan.org/biblio/bibtex/contrib/doc/ The IEEEtran BibTeX style support page is: http://www.michaelshell.org/tex/ieeetran/bibtex/

Simple References

You can manually copy in the resultant .bbl file and set second argument of \begin to the number of references (used to reserve space for the reference number labels box).

References

[1] *Mathematics Into Type*. American Mathematical Society. [Online]. Available: https://www.ams.org/arc/styleguide/mit-2.pdf

[2] T. W. Chaundy, P. R. Barrett and C. Batey, *The Printing of Mathematics*. London, U.K., Oxford Univ. Press, 1954.

[3] F. Mittelbach and M. Goossens, *The LaTeXCompanion*, 2nd ed. Boston, MA, USA: Pearson, 2004.

[4] G. Grätzer, *More Math Into LaTeX*, New York, NY, USA: Springer, 2007.

[5] M. Letourneau and J. W. Sharp, *AMS-StyleGuide-online.pdf,* American Mathematical Society, Providence, RI, USA, [Online]. Available: http://www.ams.org/arc/styleguide/index.html

[6] H. Sira-Ramirez, “On the sliding mode control of nonlinear systems,” *Syst. Control Lett.*, vol. 19, pp. 303–312, 1992.

[7] A. Levant, “Exact differentiation of signals with unbounded higher derivatives,” in *Proc. 45th IEEE Conf. Decis. Control*, San Diego, CA, USA, 2006, pp. 5585–5590. DOI: 10.1109/CDC.2006.377165.

[8] M. Fliess, C. Join, and H. Sira-Ramirez, “Non-linear estimation is easy,” *Int. J. Model., Ident. Control*, vol. 4, no. 1, pp. 12–27, 2008.

[9] R. Ortega, A. Astolfi, G. Bastin, and H. Rodriguez, “Stabilization of food-chain systems using a port-controlled Hamiltonian description,” in *Proc. Amer. Control Conf.*, Chicago, IL, USA, 2000, pp. 2245–2249.

Biography Section

If you have an EPS/PDF photo (graphicx package needed), extra braces are needed around the contents of the optional argument to biography to prevent the LaTeX parser from getting confused when it sees the complicated \includegraphics command within an optional argument. (You can create your own custom macro containing the \includegraphics command to make things simpler here.)

If you include a photo:

Michael Shell Use \begin{IEEEbiography} and then for the 1st argument use \includegraphics to declare and link the author photo. Use the author name as the 3rd argument followed by the biography text.

If you will not include a photo:

John Doe Use \begin{IEEEbiographynophoto} and the author name as the argument followed by the biography text.